

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматики
Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів Вінницького національного медичного університету»

Виконав: студент 2-го курсу, групи
ІСТ-23м, спеціальності 126 –
Інформаційні системи та технології

Сергій МИРОНЮК

(ім'я ПРІЗВИЩЕ студента)

Керівник: к.т.н., доц. каф. АІТ

Ярослав КУЛИК

(науковий ступінь, вчене звання / посада, ім'я ПРІЗВИЩЕ керівника)

« 13 » грудня 2024 р.

Опонент: Тетяна ГРИЩУК

(науковий ступінь, вчене звання / посада, ім'я ПРІЗВИЩЕ опонента)

« 16 » грудня 2024 р.

Допущено до захисту
Завідувач кафедри АІТ

д.т.н., проф. Олег БІСІКАЛО

(науковий ступінь, вчене звання)

« 16 » грудня 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Освітньо-кваліфікаційний рівень - магістр
спеціальність 126 Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСІКАЛО

« 19 » вересня 2024 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Миронюку Сергію Михайловичу

1. Тема роботи: Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів Вінницького національного медичного університету

Керівник роботи: к.т.н., доцент каф. АІТ Кулик Я.А.

Затвердженні наказом ВНТУ від « 17 » вересня 2024 року № 310.

2. Термін подання роботи студентом: до « 12 » грудня 2024 року.

Вихідні дані до роботи: Апарат УВЧ-30 з технічними характеристиками: частота генератора 40,68 МГц, вихідна потужність для першого ступеня (15+-3), для другого (30+-5), напруга 220В, маса 10кг, габарити корпусу апарата 425x275x230мм; Гелій-неоновий лазер Екран, Дифракційна Ришівка, Штатив з лінійкою, Операційна система не нижче 7, Оперативна пам'ять не нижче 4, Доступ до інтернету, Мова програмування JavaScript, Середовище розробки VSCode, Бібліотека для розробки ReactJS.

3. Зміст текстової частини: Вступ; Основні відомості лабораторних робіт. Теоретичні відомості та хід виконання; Вибір засобів та середовища для розробки

програмного забезпечення; План та розробка програмного забезпечення;
номічна частина; Висновки, Список використаних джерел.

4. Перелік ілюстративного (або графічного) матеріалу: Алгоритм виконання
лабораторної роботи, UML діаграма послідовностей системи, UML діаграма
тивностей системи, UML діаграма класів програми, UML діаграма розгор-
тання системи, UML діаграма об'єктів системи.

5. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-3	Ярослав КУЛИК к.т.н., доцент кафедри АІТ	12.09.24	13.12.24
4	Олександр ЛЕСЬКО к.е.н., проф. каф. ЕПтаВМ	12.10.24	27.10.24

6. Дата видачі завдання: « 18 » вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примі
1	Аналіз предметної області	20.09.24 - 30.09.24	Вик.
2	Вибір оптимальних інформаційних технологій	30.10.24 - 05.10.24	Вик.
3	Вибір мови програмування та середовища розробки	05.10.24 - 08.10.24	Вик.
4	Програмна реалізація	15.10.24 - 01.11.24	Вик.
5	Експериментальна перевірка роботи програми	02.10.24 - 04.11.24	Вик.
6	Оформлення економічного розділу	05.10.24 - 07.11.24	Вик.
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	08.10.24 - 11.11.24	Вик.
8	Попередній захист роботи	20.11.24	Вик.
9	Захист роботи	20.11.24	Вик.

Студент

(підпис)

Сергій МИРОНЮК

(прізвище та ініціали)

Керівник роботи

(підпис)

Ярослав КУЛИК

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.057.7 + 371.315.2

Миронюк С. М. Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для Вінницького національного медичного університету. Магістерська кваліфікаційна робота зі спеціальності 126 – Інформаційні системи та технології, освітньо-професійна програма – Інформаційні технології аналізу та зображень. Вінниця: ВНТУ, 2024. 157 сторінок.

Українською мовою: Бібліогр.:30 назв; рис.: 29 ; табл.:15;

У магістерській кваліфікаційній роботі було розглянуто основні відомості про лабораторні роботи, їх мету роботи, а також матеріали та обладнання для успішного виконання. Для реалізації розробки програмного забезпечення було обрано мову програмування JavaScript, яка ідеально підходить для веб-розробок, середовище розробки VSCode, та бібліотеку ReactJS. Розроблений проект є інформативним, зрозумілим, без навантажень та цілком задовільняє потреби замовника.

Ключові слова: фізика, фізичні процеси, лабораторні роботи, УВЧ-30, гелій-неоновий лазер, емулятор, програмне забезпечення.

ABSTRACT

Myroniuk S. M. Information Technology for Optimizing the Trajectory Planning of UAV Groups for Air Pollution Assessment. Master's Thesis in specialty 126 – Information Systems and Technologies, educational-professional program – Information Technologies for Data and Image Analysis. Vinnytsia: VNTU, 2024. 157 pages.

In Ukrainian language. References: 30 titles; Figures: 29; Tables: 15.

In the master's qualification work, the basic information about laboratory work, its purpose, as well as materials and equipment for successful implementation were considered. To implement the software development, the JavaScript programming language, which is ideal for web development, the VSCode development environment, and the ReactJS library were chosen. The developed project is informative, understandable, load-free, and fully satisfies the needs of the customer.

Keywords: physics, physical processes, laboratory work, UHF-30, helium-neon laser, emulator, software.

ЗМІСТ

ВСТУП.....	8
1 ОСНОВНІ ВІДОМОСТІ ЛАБОРАТОРНИХ РОБІТ. ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ХІД ВИКОНАННЯ	10
1.1 Характеристика об'єкту досліджень	10
1.2 Опис лабораторних робіт для розробки програмного забезпечення	14
1.2.1 Лабораторна робота №21 «Вивчення апаратів і методів УВЧ-терапії електричним і магнітним полями та ВЧ-струмом»	15
1.2.2 Лабораторна робота №26 «Вивчення гелій-неонового лазера»	26
1.3 Огляд існуючих аналогів	36
1.3.1 PhET Interactive Simulations	38
1.3.2 Vascak	40
1.3.3 Algodoo	42
1.4 Оцінка та порівняння існуючих аналогів	44
1.5 Висновки до першого розділу	49
2 ВИБІР ЗАСОБІВ ТА СЕРЕДОВИЩА ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	52
2.1 Вибір мови програмування	52
2.2 Вибір середовища розробки	61
2.3 Вибір бібліотеки для розробки програмного забезпечення	67
2.4 Висновки до другого розділу	75
3 ПЛАН ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	77
3.1 Розробка головної сторінки	78
3.2 Розробка сторінки «Теоретичні відомості»	80
3.3 Розробка сторінки «Тестування»	83
3.4 Розробка сторінки «Практична частина»	86
3.5 Висновки до третього розділу	89
4 ЕКОНОМІЧНА ЧАСТИНА	90
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	90

4.2 Розрахунок узагальненого коефіцієнта якості розробки	95
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	97
4.3.1 Витрати на оплату праці.....	97
4.3.2 Відрахування на соціальні заходи	101
4.3.3 Сировина та матеріали.....	101
4.3.4 Розрахунок витрат на комплектуючі.....	103
4.3.5 Спецустаткування для наукових (експериментальних) робіт	103
4.3.6 Програмне забезпечення для наукових робіт.....	104
4.3.7 Амортизація обладнання, програмних засобів та приміщень	105
4.3.8 Паливо та енергія для науково-виробничих цілей	107
4.3.9 Службові відрядження.....	108
4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	109
4.3.11 Інші витрати.....	109
4.3.12 Накладні (загальновиробничі) витрати.....	109
4.4 Висновки до четвертого розділу.....	112
ВИСНОВКИ	113
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	116
ДОДАТКИ.....	120
Додаток А (Обов'язковий) Технічне завдання.....	121
Додаток Б (Обов'язковий) Ілюстративна частина	124
Додаток В (Обов'язковий) Лістинг коду	131
Додаток Д (Обов'язковий) Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень.....	157

ВСТУП

Актуальність. У сучасному світі, де дистанційне навчання набуває все більшого поширення, для забезпечення якісної освіти потрібні сучасні підходи та інструменти. Одним з головних предметів, який вимагає не тільки теоретичного знання, але й практичного застосування, є фізика. Застосування сучасних підходів та інструментів у дистанційному навчанні може значно покращити якість освіти з фізики. Різноманітні онлайн-ресурси, симулятивні програми, відеолекції можуть забезпечити студентам набувати практичні навички у віртуальному середовищі. Це дозволяє студентам спостерігати, експериментувати та аналізувати фізичні процеси та явища навіть без присутності в фізичних лабораторіях та кабінетах [1].

Збільшення актуальності сучасних підходів та інструментів у дистанційному навчанні з фізики може значно покращити якість освіти та сприяти кращому засвоєнню знань студентами.

Метою дослідження є покращення умов дистанційного навчання, які спрощують та полегшують процес навчання з фізики, а також дозволяють набутти практичні навички без перебування в фізичних лабораторіях та зменшити час навчання на вивчення фізичного процесу.

Основним завданням є створення інтерактивного середовища, в якому студенти зможуть відтворювати, спостерігати та аналізувати фізичні явища та моделювати експерименти.

Для вирішення поставленої мети сформульовані наступні задачі:

- вивчення фізичних явищ та принципів, які необхідні для симуляції фізичних процесів;
- вибір засобів та середовища для розробки програмного забезпечення - розробка програмного забезпечення, яке забезпечуватиме симуляцію фізичних процесів;
- створення плану для розробки програмного забезпечення та розробка;

Об'єктом дослідження є процес дистанційного навчання з фізики. Враховуючи обмежені можливості студентів для виконання лабораторних робіт у звичайних умовах та необхідність забезпечення доступу до практичного досвіду, програмне забезпечення для симуляції фізичних процесів є важливим інструментом для забезпечення якісного навчання фізики.

Предмет дослідження: дистанційне навчання з фізики та розробка програмного забезпечення для симуляції різних фізичних процесів. Предметом є процес навчання студентів фізики за допомогою віртуальних лабораторних робіт та програмного забезпечення, яке відтворює фізичні процеси.

Джерелом дослідження слугував лабораторний практикум з медичної і біологічної фізики, а також активно використовувались електронні ресурси з інформацією

Науково технічним результатом роботи – є програмне забезпечення для лабораторних робіт з фізики яке полягає в наданні студентам отримувати практичний досвід при виконанні робіт. Використання програмного забезпечення може забезпечити зручний графік виконання лабораторних робіт з фізики і розширити взаємодію між студентами та викладачами.

Апробація результатів та публікації: За результатами даної роботи опублікована доповідь на всеукраїнській науково-практичній конференції «Молодь в науці: дослідження, проблеми, перспективи (2024)» [2].

1 ОСНОВНІ ВІДОМОСТІ ЛАБОРАТОРНИХ РОБІТ. ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ХІД ВИКОНАННЯ

1.1 Характеристика об'єкту досліджень

Об'єктом дослідження у розробці емуляторів фізичних процесів є створення математичних і програмних моделей, що дають змогу з високою точністю відтворювати поведінку різноманітних фізичних явищ та процесів у віртуальному середовищі. Це включає аналіз фізичних закономірностей, розробку алгоритмів і створення програмних рішень, які можуть бути інтегровані в різні прикладні та наукові області.

Об'єктом автоматизації є спрощена можливість виконувати лабораторні роботи без відвідування фізичних лабораторій в умовах конкретних ситуацій.

Основною метою цих досліджень є надання можливості моделювати складні фізичні системи без необхідності проведення реальних експериментів, які можуть бути дорогими, ризикованими або технічно складними для реалізації. Завдяки емуляторам стає можливим не лише зекономити ресурси, але й розширити межі досліджень, моделюючи умови, які неможливо відтворити в лабораторії, наприклад, поведінку матеріалів в екстремальних умовах або процеси в космосі.

Області застосування емуляторів:

1. Механіка твердих тіл, рідин і газів: використовується для аналізу руху, деформацій і взаємодії тіл. Наприклад, симуляція землетрусів або моделювання аеродинаміки літальних апаратів.
2. Теплопередача та термодинаміка: моделювання розподілу температур у матеріалах, що важливо для розробки енергетичного обладнання чи вивчення кліматичних змін.
3. Електромагнетизм: емуляція електричних та магнітних полів для створення і тестування електроніки або вивчення явищ, таких як індукція чи розповсюдження хвиль.

4. Квантова фізика: моделі для вивчення поведінки частинок на мікроскопічному рівні, що критично важливо для розробки квантових комп'ютерів чи аналізу поведінки матеріалів.
5. Біофізика та медицина: використання емуляторів для моделювання процесів у людському організмі, таких як циркуляція крові, реакція тканин на медикаменти або робота внутрішніх органів.

Технічні аспекти:

1. Числові методи: для точного розрахунку фізичних процесів використовуються методи кінцевих елементів, Монте-Карло, інтегральні та диференціальні рівняння.
2. Паралельні обчислення: завдяки сучасним суперкомп'ютерам і графічним процесорам моделювання складних явищ стало швидшим і доступнішим.
3. Оптимізація: створення моделей вимагає знаходження балансу між точністю та швидкістю обчислень. Це досягається шляхом спрощення певних аспектів явищ, зберігаючи їхню фізичну реалістичність.

Прикладні переваги емуляторів фізичних процесів:

1. Інженерія та проектування
 - Оптимізація конструкцій: емулятори дозволяють випробовувати різні варіанти конструкцій, наприклад, будівель, мостів або літальних апаратів, без витрат на створення фізичних прототипів. Це знижує ризик помилок і пришвидшує процес проектування.
 - Розробка нових матеріалів: моделювання поведінки матеріалів у різних умовах, таких як високий тиск, температура чи корозійні середовища. Це дозволяє створювати більш міцні, легкі та довговічні матеріали.
 - Моделювання аварій: емулятори допомагають аналізувати наслідки аварій або катастроф (наприклад, падіння автомобіля чи землетрусу), дозволяючи вдосконалювати безпеку інфраструктури та техніки.

2. Енергетика

- Розробка відновлюваних джерел енергії: моделювання вітрових турбін, сонячних панелей або систем гідроелектростанцій дозволяє покращувати їхню ефективність і знижувати витрати на виробництво енергії.
- Теплоенергетика: емулятори використовуються для оптимізації систем передачі тепла, таких як теплообмінники чи котельні установки, забезпечуючи максимальну ефективність і зменшуючи викиди.

3. Наука та дослідження

- Астрономія: моделювання орбіт планет, формування галактик або чорних дір. Наприклад, симуляції можуть допомогти зрозуміти, як утворився Всесвіт або як виглядають екзопланети.
- Фізика елементарних частинок: емулятори допомагають вивчати взаємодію частинок у колайдерах, таких як Великий адронний колайдер, що дозволяє передбачати нові фізичні явища без додаткових експериментів.

4. Медицина та біотехнології

- Віртуальні тіла: моделювання органів, систем кровообігу або кісток для аналізу їхньої реакції на травми, хірургічні втручання чи медикаменти. Наприклад, симуляція операції дозволяє хірургам відпрацювати техніку перед реальною процедурою.
- Розробка ліків: емулятори дають змогу передбачити, як нові лікарські препарати взаємодіятимуть із клітинами організму або вірусами, скорочуючи час та витрати на їх розробку.
- Моделювання поширення епідемій: аналіз розповсюдження вірусів і розробка ефективних стратегій стримування епідемій, таких як COVID-19.

5. Освіта

- Віртуальні лабораторії: студенти можуть проводити фізичні експерименти в симуляціях, де немає ризику пошкодження обладнання

чи шкоди для здоров'я. Наприклад, симуляція хімічних реакцій або експериментів у фізиці.

- Зменшення вартості освіти: використання емуляторів скорочує витрати на матеріали та обладнання, що робить освіту доступнішою.

6. Авіація та космічні дослідження

- Аеродинамічні тести: емулятори дозволяють аналізувати аеродинаміку літаків або космічних апаратів до їх фізичного виготовлення, що зменшує час і витрати на їх розробку.
- Космічні місії: моделювання умов на інших планетах, таких як Марс, допомагає планувати місії та створювати обладнання, здатне працювати в екстремальних умовах.

7. Екологія та охорона природи

- Аналіз кліматичних змін: емулятори дозволяють прогнозувати зміни клімату, враховуючи викиди парникових газів, танення льодовиків і вплив людської діяльності.
- Моделювання екосистем: аналіз екосистем допомагає зрозуміти, як змінюється природа під впливом урбанізації чи інших чинників, і розробляти стратегії їхнього збереження.

8. Розробка та тестування нових технологій

- Робототехніка: симуляція руху роботів у різних середовищах дозволяє вдосконалювати їхні алгоритми управління та дизайн.
- Автономні транспортні засоби: емулятори дозволяють тестувати системи автопілота віртуально, знижуючи ризики для людей і техніки.

- 9. Розважальна індустрія

- Ігрова індустрія: використання емуляторів для створення реалістичної фізики в іграх, наприклад, моделювання води, вітру або руху персонажів.
- Кіноіндустрія: емуляція фізичних ефектів дозволяє створювати видовищні сцени з мінімальними витратами, наприклад, обвал будівель чи вибухи.

Отже, емулятори фізичних процесів є універсальним інструментом, який знаходить застосування у найрізноманітніших сферах, від науки і техніки до медицини та розваг. Їх використання дозволяє економити ресурси, знижувати ризики, прискорювати інновації та забезпечувати безпечне середовище для досліджень і навчання.

1.2 Опис лабораторних робіт для розробки програмного забезпечення

Основною задачею є розробка двох емуляторів для реалізації лабораторних робіт з фізики, зокрема, на теми «Вивчення апаратів і методів УВЧ-терапії електричним і магнітним полями та ВЧ-струмом» і «Вивчення гелій-неонового лазера». Ці емулятори мають забезпечити студентам можливість детально ознайомитися з теоретичними основами та практичними аспектами роботи з відповідним обладнанням, не потребуючи доступу до реальних установок. Завдяки цьому навчання стане доступнішим, безпечнішим і менш залежним від наявності фізичних лабораторій.

Для створення емуляторів насамперед необхідно ретельно вивчити специфіку кожної з лабораторних робіт. У випадку з УВЧ-терапією важливо розібратися в фізичних принципах впливу електричних і магнітних полів високої частоти на біологічні тканини. Це включає аналіз параметрів електричних сигналів, способів генерування високочастотних полів, а також особливостей взаємодії цих полів із різними середовищами. Розуміння роботи апаратів, які використовуються в УВЧ-терапії, допоможе точно відтворити всі аспекти їхньої поведінки у віртуальному середовищі.

Щодо лабораторної роботи з вивчення гелій-неонового лазера, акцент слід зробити на принципах його роботи, включаючи процеси генерації когерентного випромінювання, роль активного середовища та лазерного резонатора. Дослідження особливостей лазерного випромінювання, таких як довжина хвилі, когерентність і стабільність пучка, є важливими для створення

моделі, яка дозволить студентам вивчати характеристики цього пристрою у віртуальній лабораторії. Аналіз роботи лазера також включає розуміння взаємодії лазерного випромінювання з матеріалами, що може бути інтегровано в емулятор для демонстрації його практичного застосування.

1.2.1 Лабораторна робота №21 «Вивчення апаратів і методів УВЧ-терапії електричним і магнітним полями та ВЧ-струмом»

Метою лабораторної роботи є детальне ознайомлення з будовою, принципом функціонування та методикою використання апарата УВЧ-терапії, а також дослідження впливу електричних і магнітних полів ультрависокої частоти (УВЧ) на середовища, що імітують електропровідні та діелектричні властивості біологічних тканин. Використання таких моделей дозволяє аналізувати фізичні процеси, що відбуваються в тканинах під впливом полів УВЧ, у контрольованих умовах, забезпечуючи точність і безпеку експериментів [3].

Для успішного проведення дослідження використовується спеціалізоване обладнання, серед якого основним є апарат УВЧ-30, що генерує електричні та магнітні поля ультрависокої частоти. Цей апарат дозволяє досліджувати особливості впливу полів УВЧ на різні середовища, забезпечуючи високу стабільність і точність параметрів випромінювання. Для вивчення вихрових струмів застосовується аплікатор ЕВС-1, який дозволяє наочно демонструвати взаємодію полів із матеріалами, що моделюють біологічні тканини.

Індикатори у вигляді люмінесцентних або неонових ламп слугують інструментом для візуалізації наявності електричних і магнітних полів у досліджуваній зоні, полегшуючи розуміння фізичних принципів їх дії. Моделювання біологічних тканин здійснюється за допомогою таких середовищ, як електроліт, касторове масло або гліцерин, які володіють

відповідними електропровідними та діелектричними властивостями. Для точного вимірювання тривалості впливу полів на середовища застосовується секундомір, що дозволяє проводити експерименти з високим рівнем точності та повторюваності. Комплексне використання цих приладів і матеріалів забезпечує всебічне вивчення досліджуваних фізичних процесів, сприяючи формуванню практичних навичок роботи з медичними апаратами та закріпленню теоретичних знань.

У першій лабораторній роботі досліджується принцип дії апарата УВЧ-30, основні елементи якого представлені у вигляді спрощеної блок-схеми (рисунок 1.1). Апарат складається з кількох функціональних компонентів, кожен із яких виконує ключову роль у створенні та передачі електричних і магнітних полів ультрависокої частоти.

Блок живлення (ДЖ) забезпечує необхідну напругу для роботи всіх компонентів апарата, підтримуючи стабільну роботу системи. Перешкодоусуваючий пристрій (ПП) призначений для зменшення впливу електромагнітних завад, що дозволяє підвищити точність та ефективність терапевтичного впливу. Ламповий генератор (ЛГ) виконує функцію джерела електромагнітних коливань, генеруючи високочастотні сигнали, які є основою для створення полів УВЧ.

Терапевтичний контур (ТК) відповідає за формування вихідного сигналу та його адаптацію до параметрів, необхідних для лікувального застосування. Локальний терапевтичний вплив на біологічні тканини здійснюється за допомогою двох основних пристосувань: електродів пацієнта (ЕП), які передають електричне поле, та аплікаторів вихрових струмів (ЕВС-1), які створюють магнітне поле. Обидва пристрої забезпечують цілеспрямовану дію на вибрані ділянки тіла, дозволяючи досягти терапевтичного ефекту завдяки точній концентрації полів на заданій зоні.

Таким чином, апарат УВЧ-30 є складним, багатofункціональним пристроєм, який завдяки своїм компонентам і адаптованим методам передачі

полів УВЧ дозволяє здійснювати ефективний вплив на тканини організму пацієнта з метою терапії.

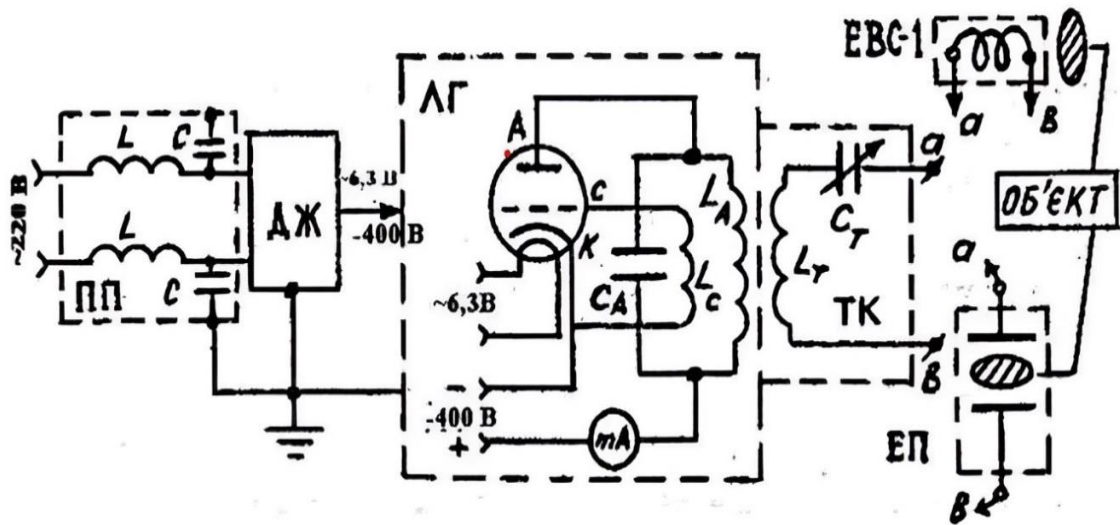


Рисунок 1.1 – Спрощена схема апарату УВЧ-30

Ламповий генератор є ключовим елементом апарату УВЧ-30, відповідальним за створення електромагнітних коливань, необхідних для генерації полів ультрависокої частоти. Принцип його дії базується на використанні коливального контуру, параметри якого визначають частоту генерованих коливань. Коливальний контур складається з індуктивності (котушки) і ємності (конденсатора), які з'єднані в анодному колі генераторної лампи. (рис.1).

Анодне коло генераторної лампи виконує функцію активного елемента, що підсилює коливання в контурі. Лампа забезпечує стабільність і достатню амплітуду електромагнітних коливань, необхідних для формування високочастотного поля. Завдяки цьому ламповий генератор здатний ефективно забезпечувати роботу апарату УВЧ-30, створюючи умови для точного та контрольованого впливу на біологічні тканини під час терапії.

Терапевтичний контур $L_t C_t$, зв'язаний індуктивно з коливальним контуром $L_a C_a$, сприймає від нього енергію електромагнітних коливань і передає її на пристосування ЕП чи ЕВС-1, які підключаються до його виходу.

При цьому найбільша передача енергії досягається при резонансі контурів, що здійснюється шляхом регулювання ємності конденсатора C_m . Момент появи резонансу в ламповому генераторі фіксується за максимальним відхиленням стрілки індикатора, що включений в анодне коло генераторної лампи. Це відхилення сигналізує про досягнення максимальної амплітуди електромагнітних коливань у коливальному контурі. Резонансна частота генерується тоді, коли індуктивність і ємність коливального контуру налаштовані таким чином, що система досягає свого максимального енергетичного ефекту. Це дозволяє досягти високої стабільності та ефективності при генеруванні ультрависокочастотного поля, необхідного для терапевтичних цілей.

Блок живлення апарата відповідає за перетворення напруги змінного струму мережі в необхідні значення напруги, які забезпечують нормальну роботу лампового генератора. Це важливий етап, оскільки для стабільної роботи генератора потрібна напруга, яка відповідає технічним вимогам конкретної лампи та її анодного кола. Блок живлення виконує функцію перетворення і стабілізації напруги, що гарантує безперервну та якісну генерацію електромагнітних коливань.

Перешкодоусуваючий пристрій є важливим компонентом, який виконує функцію захисту апарата та зовнішнього середовища від можливого проникнення електромагнітних коливань ультрависокої частоти в електричну мережу змінного струму. Це необхідно для збереження якості роботи інших електронних пристроїв, підключених до мережі, а також для запобігання потенційних електромагнітних завад, які можуть вплинути на навколишнє середовище. В результаті цей пристрій забезпечує стабільність роботи апарата УВЧ-30, гарантує його ефективність і безпеку.

Технічна характеристика апарата УВЧ-30 має частоту генератора $40,68 \text{ МГц} \pm 2\%$, вихідну потужність для першого ступеня $(15 \pm 3) \text{ Вт}$, для другого - $(30 \pm 5) \text{ Вт}$, живлення від мережі змінного струму з частотою 50 Гц , напругою $127 \text{ чи } 220 \text{ В}$, споживана потужність з мережі 200 Вт , габаритні розміри

корпусу апарата 425x275x230 мм, загальна маса 10 кг.

Портативний апарат УВЧ-30 є важливим пристроєм у сфері фізіотерапії, який застосовується для місцевої лікувальної дії на тканини організму за допомогою електричних та магнітних полів ультрависокої частоти (УВЧ). УВЧ-терапія має широкий спектр застосування, включаючи лікування неврологічних, шкірних, отоларингологічних, стоматологічних та інших захворювань. Цей апарат здатний ефективно впливати на пацієнтів різного віку, як дорослих, так і дітей, завдяки своїй адаптивності до різних клінічних умов.

Малі габарити та низька маса апарата УВЧ-30 є важливими характеристиками, що роблять його надзвичайно зручним у використанні. Ці особливості дозволяють переносити апарат і застосовувати його не лише в межах фізіотерапевтичних кабінетів, але й поза ними — біля ліжка пацієнта в умовах стаціонарного лікування або в амбулаторії. Це забезпечує високу мобільність і дозволяє здійснювати терапевтичні процедури без необхідності переміщувати пацієнтів до спеціалізованих медичних установ, що є особливо важливим для ослаблених хворих або людей, які проходять реабілітацію після важких травм.

УВЧ-30 ефективно застосовується для лікування різноманітних захворювань, що вимагають місцевого впливу на тканини організму. Наприклад, у неврології апарат використовують для лікування хронічного болю, запалень, порушень кровообігу в нервових тканинах, а також для відновлення після травм нервової системи. В отоларингології апарат застосовують для лікування запальних процесів у горлі, носі, вухах, зменшення набряків і покращення кровообігу в цих ділянках. Шкірні захворювання, такі як дерматити, псоріаз, акне, можуть бути ефективно лікувані за допомогою УВЧ-терапії, оскільки ультрависокочастотні поля стимулюють регенерацію шкіри та прискорюють загоєння пошкоджених тканин. Стоматологічні проблеми, включаючи запалення ясен, зубні болі та відновлення після стоматологічних процедур, також можуть бути успішно

вирішені з допомогою апарата УВЧ-30.

Завдяки можливості застосовувати цей апарат у будь-яких умовах, він забезпечує зручність і ефективність лікування пацієнтів, навіть коли їхнє пересування обмежене. В умовах стаціонарного лікування апарат дозволяє проводити процедури без необхідності транспортування хворого в спеціалізований кабінет, що особливо важливо для пацієнтів, які знаходяться в важкому стані або мають обмежену рухливість. У амбулаторії портативний характер апарата дозволяє лікарям забезпечити своєчасне і індивідуалізоване лікування без великих витрат часу та ресурсів.

Таким чином, апарат УВЧ-30 поєднує в собі високу ефективність і універсальність у використанні, роблячи його важливим інструментом у лікувальному процесі для широкого спектра пацієнтів, включаючи тих, хто потребує лікування в умовах стаціонару або в домашніх умовах.

Механізм дії електричного поля ультрависокої частоти (УВЧ) на електроліти та діелектрики є основою для багатьох лікувальних ефектів у фізіотерапії, зокрема при використанні апарата УВЧ-30. Коли електроліт поміщається в електричне поле УВЧ, іони, що містяться в ньому, починають здійснювати вимушені коливальні рухи з частотою електричного поля. Це явище виникає через те, що в електроліті присутні заряджені частинки — катіони та аніони, які взаємодіють з електричним полем.

Ці коливання іонів є механічним процесом, при якому заряджені частинки рухаються в напрямку зміни електричного поля. Частота цих коливань відповідає частоті електричного поля УВЧ, що викликає синхронізацію руху іонів з цією частотою. Важливо зазначити, що в результаті такого руху іони починають стискатися й розширюватися, взаємодіяти з іншими частками середовища, що, в свою чергу, призводить до перетворення електричної енергії в теплову енергію, тобто в теплоту.

Цей процес нагрівання електроліту завдяки енергії, що передається електричним полем, є ключовим фактором в лікувальному ефекті УВЧ-терапії. Відповідно, при впливі на біологічні тканини, що містять електроліти, такий

механізм сприяє локальному підвищенню температури в тканинах, що активізує біохімічні процеси, покращує кровообіг і зменшує запалення. Це забезпечує терапевтичний ефект, особливо в лікуванні запальних захворювань та хронічних болей.

Для діелектриків, які не містять вільних іонів, діє механізм, схожий на поляризацію молекул під впливом електричного поля. У цьому випадку, хоча іони не здійснюють коливання, молекули діелектриків під впливом електричного поля можуть змінювати свою орієнтацію, що також веде до нагрівання, але з менш вираженим ефектом порівняно з електролітами.

Кількість теплоти, яка виділяється в одиниці об'єму електроліту за одиницю часу ($q = Q/(V\Delta t)$), можна обчислити за формулою (1.1):

$$q_c = k \frac{E^2}{p}, \quad (1.1)$$

де E – ефективне значення напруженості електричного поля;

p – питомий опір електроліту;

k – коефіцієнт пропорційності (в СІ $k = 1$).

Коли діелектрик розташовується в електричному полі ультрависокої частоти (УВЧ), електричне поле викликає орієнтаційну поляризацію молекул діелектрика. Це означає, що молекули, які мають полярність, намагаються орієнтуватися відповідно до напрямку електричного поля. Однак через те, що частота коливань електричного поля УВЧ є дуже високою, молекули діелектрика не встигають миттєво змінювати своє положення, і виникає явище фазового відставання між обертально-коливальними рухами молекул і коливанням електричного поля. Це відставання є результатом інерційних сил, що заважають молекулам слідувати змінам у полі з такою ж частотою.

Внаслідок цього відставання частина енергії електричного поля не використовується для здійснення руху молекул, а перетворюється в теплову енергію — це і є діелектричні втрати. Ці втрати характеризуються тангенсом

кута діелектричних втрат ($tg\delta$), який визначає, скільки енергії електричного поля перетворюється в тепло. Величина тангенса кута втрат залежить від кількості та типу молекул у матеріалі, їхніх взаємодій між собою, а також від частоти електричного поля.

Згодом, енергія, що втрачається внаслідок діелектричних втрат, перетворюється у внутрішню енергію діелектрика, що веде до його нагрівання. Це нагрівання виявляється як підвищення температури матеріалу, що є основним механізмом дії УВЧ-терапії на діелектричні матеріали. У фізіотерапевтичних процедурах це ефект може бути використаний для локального нагрівання тканин організму, стимулювання кровообігу та прискорення загоєння пошкоджених тканин.

Кількість теплоти, яка виділяється в одиниці об'єму за одиницю часу можна обчислити за формулою (1.2):

$$q_{\partial} = k\omega\varepsilon\varepsilon_0 E^2 tg\delta, \quad (1.2)$$

де k – коефіцієнт пропорційності;

ε – відносна діелектрична проникливість діелектрика;

ε_0 – електрична стала;

ω – кругова частота;

E – ефективна напруженість електричного поля;

δ – кут діелектричних втрат.

Описані механізми дії електричного поля ультрависокої частоти (УВЧ) на електроліти і діелектрики можуть бути застосовані для оцінки впливу цього фізичного фактора на тканини живого організму. Тканини організму, як правило, мають складну комбінацію електричних властивостей, що поєднують характеристики як електролітів, так і діелектриків. Це відображає різноманітність біологічних матеріалів, які складаються з води, електролітів, органічних і неорганічних сполук, а також з великої кількості молекул з полярними властивостями.

Тканини, що мають більшу частину води та іонів, такі як м'язова тканина або кров, більше схожі на електроліти. Вони можуть здійснювати коливальні рухи іонів у відповідь на електричне поле УВЧ, що спричиняє їх нагрівання і стимулює фізіологічні процеси, такі як покращення кровообігу або зменшення запалення. Цей ефект є важливим у фізіотерапії, де підвищення температури тканин допомагає знімати біль та покращувати метаболічні процеси.

З іншого боку, тканини з низьким вмістом вільних іонів, такі як жирова тканина або деякі ділянки сполучної тканини, проявляють більше діелектричні властивості. Це означає, що їх молекули не мають вільних заряджених частинок, здатних коливатися в електричному полі, однак поляризація молекул, що складають ці тканини, також призводить до їх нагрівання внаслідок діелектричних втрат. Цей тип впливу може бути менш вираженим, але все одно важливим для коригування теплових процесів у тілі.

Отже, тканини живого організму можна розглядати як складні електричні системи, де комбіновані механізми поляризації і коливальних рухів іонів під впливом електричного поля призводять до різного ступеня нагрівання, що може бути використано для лікувальних цілей. Розуміння цих механізмів дозволяє точніше налаштувати параметри УВЧ-терапії в залежності від характеру тканин та необхідного терапевтичного ефекту.

Тому отримуваний тепловий ефект в них при дії електричного поля УВЧ розраховується за формулою (1.3)

$$q = q_e + q_d. \quad (1.3)$$

При розташуванні електроліта в магнітному полі УВЧ останнє створює в ньому, як провіднику електричного струму, вихрові струми, енергія яких у відповідності з законом Джоуля-Ленца ($Q = kI^2Rt$) перетворюється в теплову. Кількість теплоти, яка виділяється в одиниці об'єму за одиницю часу, обчислюється за формулою (1.4):

$$q = k\gamma\omega^2 B^2, \quad (1.4)$$

де k – коефіцієнт пропорційності;

γ – питома електропровідність;

ω – кругова частота;

B – ефективне значення індукції магнітного поля.

При проходженні струму високої частоти через тканини організму, в результаті взаємодії струму з молекулами та іонами в тканинах, відбувається виділення джоулевого тепла. Це явище є результатом електричних втрат енергії, що виникають через опір тканин струму. Коли струм проходить через матеріал, частина енергії перетворюється в тепло внаслідок тертя і руху заряджених частинок, зокрема електронів та іонів. Це тепло виділяється через тепловий ефект, відомий як джоулевий ефект.

У біологічних тканинах, які складаються з води, електролітів та органічних сполук, цей процес набуває особливого значення. Тканини організму мають різний рівень електричного опору, що залежить від їх складу і структури. Високий вміст води і електролітів в тканинах (наприклад, м'язи, кров) призводить до більшого електричного провідництва, а отже, і до більш інтенсивного виділення тепла при проходженні струму високої частоти.

Це джоулеве тепло є основним механізмом нагрівання тканин під час застосування струму високої частоти в фізіотерапії, зокрема в УВЧ-терапії. Підвищення температури тканин сприяє поліпшенню кровообігу, зменшенню запальних процесів, зняттю м'язової напруги та стимуляції відновлення тканин. Тому джоулевий ефект є важливим фактором у лікувальних процедурах, де тепло використовується для терапевтичного впливу на тканини організму.

Кількість теплоти, що виділяється в одиниці об'єму за одиницю часу, обчислюється за формулою (1.5):

$$q = kj^2p, \quad (1.5)$$

де j – густина струму;

p – питомий опір тканини організму.

З усього розглянутого про механізми впливу електричного і магнітного полів ультрависокої частоти (УВЧ) та високочастотних струмів на тканини організму можна зробити висновок, що первинною терапевтичною дією є тепловий ефект. Це обумовлено тим, що при проходженні електричного струму через тканини організму відбувається виділення джоулевого тепла, яке призводить до підвищення температури в тканинах. Тепло, що виділяється, сприяє покращенню кровообігу, зменшенню запальних процесів, зняттю м'язової напруги та стимуляції метаболічних процесів. Це дозволяє використовувати УВЧ-терапію для лікування ряду захворювань, таких як неврологічні, шкіряні, отоларингологічні та інші, де тепловий ефект є основним терапевтичним фактором.

Однак поряд з тепловим ефектом, вчені висловлюють припущення про існування інших механізмів, пов'язаних з активацією біохімічних і фізіологічних процесів. Наприклад, вплив електричних і магнітних полів може спричиняти зміни в іонному складі клітин, активувати ферментні системи, покращувати проникність клітинних мембран та стимулювати відновлення тканин на молекулярному рівні. Окрім того, існують дані про можливе покращення енергетичного обміну в клітинах завдяки впливу на внутрішньоклітинні процеси, такі як активізація АТФ-синтетази та інших молекул, що відповідають за енергетичне забезпечення клітин.

Незважаючи на прогрес у розумінні цих механізмів, отримані нові факти ще не є повністю вивченими і потребують подальших детальних медико-біологічних та біофізичних досліджень. Вивчення цих процесів дозволить більш точно налаштувати параметри лікувального впливу, забезпечуючи індивідуалізований підхід до кожного пацієнта. Дослідження цих механізмів також відкриває нові перспективи для розвитку ефективних терапевтичних методик, що базуються на використанні високочастотних струмів та

електромагнітних полів.

1.2.2 Лабораторна робота №26 «Вивчення гелій-неонового лазера»

Мета роботи полягає у вивченні будови та принципів дії оптичного квантового генератора – газового лазера, а також у набутті навичок користування газовим лазером, визначенні довжини хвилі та енергії кванта лазерного випромінювання. Основним завданням є розуміння принципу роботи лазера, зокрема механізмів його генерації, які ґрунтуються на явищі вимушеного випромінювання. Газові лазери, зокрема гелій-неонові лазери, є одними з найбільш поширених типів лазерів, що використовуються для різноманітних досліджень, зокрема для вимірювання оптичних властивостей матеріалів та в спектроскопії.

Оволодіння навичками роботи з газовим лазером передбачає знання конструкції пристрою, його основних компонентів (як от активне середовище, електричні розряди, дзеркала та оптичні елементи), а також техніки налаштування лазера для отримання стабільного лазерного випромінювання. Крім того, важливо розуміти принципи роботи лазерного генератора: як в ньому здійснюється інтервал між енергетичними рівнями атомів або молекул активного середовища, та як цей процес може призвести до спонтанного або вимушеного випромінювання фотонів.

Для визначення довжини хвилі лазерного випромінювання та енергії кванта використовується дифракційна решітка. Через взаємодію лазерного випромінювання з дифракційною решіткою відбувається розділення світла на спектр, що дозволяє виміряти різні параметри випромінювання, зокрема довжину хвилі. Знаючи довжину хвилі, можна визначити енергію одного фотона лазерного випромінювання, використовуючи відому залежність між енергією фотона та довжиною хвилі [4].

Для успішного виконання лабораторної роботи потрібні такі прилади і матеріали:

1. Газовий гелій-неоновий лазер типу ЛГ-209 з блоком живлення;
2. Дифракційна решітка;
3. Оптична лава з вимірювальною лінійкою;
4. Екран з міліметровою шкалою.

У другій лабораторній роботі досліджуються квантові системи, такі як атоми, молекули та іони, що можуть перебувати в різних енергетичних станах. Кожен з цих станів характеризується певним енергетичним рівнем, який визначає енергію частинки у цьому стані. Квантові системи можуть перебувати в кількох енергетичних станах, і ці стани є дискретними, тобто можливі лише певні значення енергії, які можна описати як енергетичні рівні.

Основний стан (стаціонарний стан) частинок характеризується мінімальною енергією. Це є найнижчий рівень енергії, який може мати частинка в даній квантовій системі. Коли частинка перебуває в основному стані, вона не втрачає енергію і не випромінює випромінювання, оскільки цей стан є стабільним. Відповідно до принципу збереження енергії, частинка не може бути вищою за цей рівень без додаткового зовнішнього впливу, наприклад, поглинання фотона.

Крім основного стану, квантова система може переходити в збуджені стани, в яких енергія частинки більша за мінімальну. Такі переходи відбуваються завдяки поглинанню або випромінюванню фотонів, що спричиняє зміну енергії частинки. Переходи між енергетичними рівнями, що супроводжуються зміною енергії, є важливими для розуміння процесів випромінювання та поглинання у квантових системах, що лежить в основі роботи багатьох оптичних приладів, таких як лазери.

Згідно з квантовою механікою, енергетичні рівні атомів та молекул мають дискретну природу, що відрізняє їх від класичних систем, де енергія може змінюватися без обмежень. Кількість та розташування енергетичних

рівнів залежать від специфіки квантової системи та її внутрішньої структури, що дозволяє прогнозувати її поведінку в різних умовах.

У стаціонарному стані квантова система, така як атом, може залишатися стабільною протягом тривалого часу. Це стан з мінімальною енергією, при якому система не випромінює енергії і залишається в рівновазі. Проте при поглинанні зовнішньої енергії, наприклад, фотонів, атоми можуть переходити на вищий енергетичний рівень, що призводить до їх збудження. У збудженому стані атоми мають вищу енергію, ніж у основному стані.

Час, протягом якого атом залишається в збудженому стані, зазвичай дуже короткий — близько 10^{-8} секунд, після чого атом повертається до основного стану, випромінюючи енергію у вигляді фотонів. Це явище називається спонтанним випромінюванням. Однак є випадки, коли час перебування атома в збудженому стані значно більший. Такі стани, які мають відносно тривалий час існування (порядку 10^{-3} секунд і більше), називаються метастабільними станами.

Метастабільні стани є важливими у контексті лазерної фізики. У цих станах атоми можуть зберігати енергію протягом більш тривалого часу, що дає можливість накопичувати енергію в атомах і підвищувати ймовірність переходу до лазерного стану. Метастабільні рівні дозволяють створити умови для лазерної стимуляції, коли атоми вищого енергетичного рівня можуть бути "переведені" в лазерний стан через стимульоване випромінювання, що лежить в основі роботи лазерів.

Переходи збуджених атомів на нижчі енергетичні рівні, які супроводжуються випромінюванням квантів енергії, є основою для двох типів випромінювання: спонтанного і індукованого (вимушеного). Спонтанне випромінювання відбувається без зовнішнього впливу, коли атоми самостійно переходять з більш високих енергетичних рівнів на нижчі. Ці переходи мають випадковий характер, оскільки вони відбуваються за рахунок внутрішніх взаємодій атомів або молекул. Спонтанне випромінювання характеризується

непередбачуваністю частоти, фази та напрямку фотонів, що виникають в результаті цих переходів.

Індуковане або вимушене випромінювання виникає, коли збуджені атоми піддаються впливу зовнішніх фотонів, що мають певну частоту, або фотонів, які виникають через спонтанне випромінювання. У цьому випадку атоми переходять на нижчий енергетичний рівень під впливом зовнішнього фотонного поля. Характерною особливістю індукованого випромінювання є те, що фотон, який викликає перехід, має ту саму частоту, фазу та напрямок руху, що й фотон, який спричинив спонтанне випромінювання. Завдяки цьому індуковане випромінювання є когерентним — його хвилі зберігають взаємну синхронізацію, що є важливим фактором для генерації когерентних оптичних хвиль, наприклад, у лазерах.

Процес індукованого випромінювання має лавиноподібний характер: кожен індукований перехід збільшує кількість фотонів, що викликають подальші переходи. Таким чином, індуковані фотони можуть викликати нові вимушені переходи, що призводить до збільшення інтенсивності випромінювання, створюючи потужний потік когерентних фотонів. Цей процес можливий тільки в активних середовищах, де можна створити стан інверсної заселеності, тобто коли вищі енергетичні рівні містять більше атомів, молекул або іонів, ніж нижчі. У звичайних умовах збуджені атоми або молекули знаходяться в меншості на вищих рівнях, але в активних середовищах створюється умова для того, щоб перезаселення вищих рівнів забезпечувало активне генераційне випромінювання.

Процес інверсної заселеності є основним для лазерної генерації, тобто процесу створення когерентного, монохроматичного і високопотужного випромінювання. Для цього необхідно мати активне середовище, яке містить елементи, здатні перейти в збуджений стан і потім згенерувати індуковані переходи. Такими активними середовищами можуть бути атоми, молекули або іони в газах, рідинах, кристалах або напівпровідниках.

Активне середовище повинно мати здатність піддаватися збудженню до високих енергетичних рівнів, при цьому важливо, щоб при збудженні виникала інверсна заселеність. Після цього, завдяки зовнішньому електромагнітному полю або вже створеним фотонам, відбуваються індуковані переходи, що викликають генерацію фотонів, котрі мають ту саму частоту, фазу і напрямок, створюючи потужне когерентне випромінювання.

Лазерні пристрої, які здатні генерувати когерентне випромінювання, отримали назву оптичних квантових генераторів — лазерів. Лазери можуть бути класифіковані за типом активного середовища і методом збудження активних часток:

1. Газові лазери — використовують гази як активне середовище, наприклад, гелій-неонові лазери, де як активна речовина використовується суміш гелію і неону.
2. Рідинні лазери — активне середовище складається з рідин, таких як фарби або органічні молекули, що дозволяють створювати потужні лазери для спеціальних застосувань.
3. Кристалічні лазери — використовують кристалічні матеріали, наприклад, рубіни або інші напівпровідники, як активні середовища.
4. Напівпровідникові лазери — широко застосовуються в сучасних технологіях завдяки їх малим розмірам і здатності працювати в різних спектральних діапазонах.
5. Хімічні лазери — використовують хімічні реакції для збудження атомів, молекул або іонів і генерації лазерного випромінювання.

Кожен з цих лазерів має свої особливості в залежності від того, який матеріал використовують для створення інверсної заселеності та як цей матеріал збуджується. Це дозволяє створювати лазери, які можуть працювати в різних спектральних діапазонах, від ультрафіолету до інфрачервоного випромінювання, і застосовувати їх у різних сферах науки, техніки, медицини, а також у промисловості.

Гелій-неоновий лазер (HeNe-лазер) є одним із класичних типів газових лазерів, що працює на основі суміші гелію (He) і неону (Ne). У цьому лазері використовується ефект взаємодії атомів двох різних газів (гелію і неону), які мають близькі енергетичні рівні. Так, енергія рівня 2 гелію досить близька до енергії рівня 3 неону (рис. 1.2).

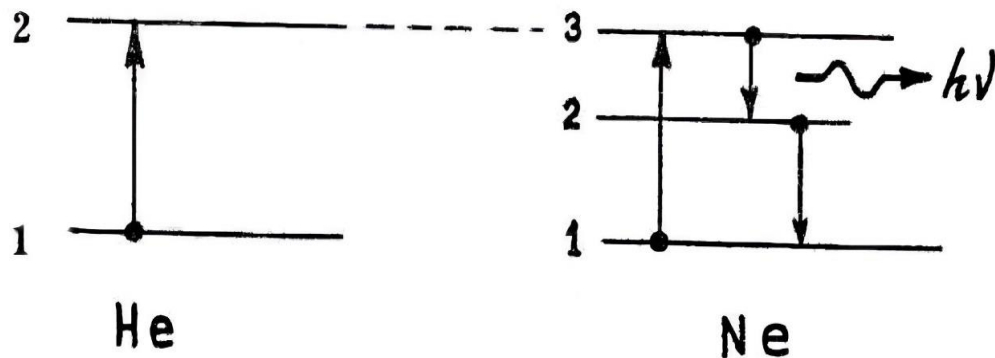


Рисунок 1.2 – Неоновий лазер безперервної дії

Збуджені за допомогою електричного розряду в трубці лазера атоми гелію переходять з основного рівня 1 на енергетичний рівень 2. В процесі непружного співударяння між атомами газової суміші збуджені атоми гелію передають свою енергію атомам неону, а самі повертаються в основний стан без випромінювання. Атоми неону, одержавши енергію від атомів гелію, переходять з основного рівня 1 на енергетичний рівень 3, який є метастабільним. Цим самим атоми гелію забезпечують перезаселеність рівня 3 неону по відношенню до рівня 2. Цьому процесу сприяє підбір оптимальних значень парціальних тисків гелію (1 мм рт.ст.) і неону (0,1 мм рт.ст.) в газорозрядній трубці. Отже, атоми гелію сприяють створенню стану інверсної заселеності атомів неону, який є активним елементом з активному середовищі гелій-неонового лазера.

Спочатку індуковані переходи атомів неону з рівня 3 на рівень 2 викликаються фотонами, які з'явилися внаслідок спонтанного випромінювання, а потім в процес включаються індуковані фотони. Перехід

втомі неону з рівня 2 на рівень 1 здійснюється без випромінювання з передачею енергії через зіткнення іншим атомам та стінкам газорозрядної трубки.

Розміри трубки підбирають такими, щоб сприяти розвантаженню рівня 2 і тим самим підтримувати стан інверсної заселеності атомів неону, тобто, щоб на рівні 3 було більше атомів, ніж на рівні 2.

Принципова структурна схема газового лазера зображена на рисунку 1.3.

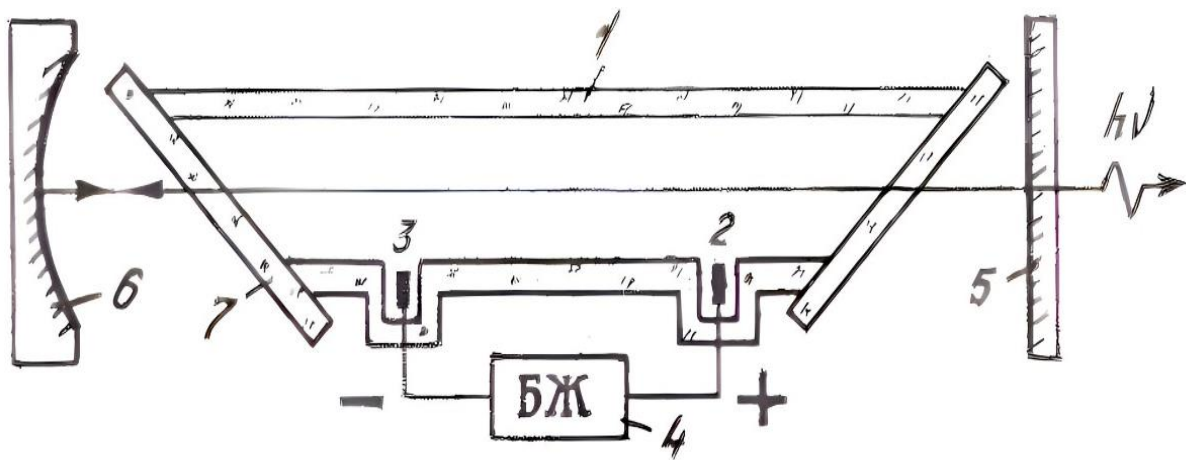


Рисунок 1.3. – Структурна схема газового лазера

Основним конструктивним елементом є наповнена газовою сумішшю газорозрядна трубка 1 з анодом 2 і катодом 3, на які подається висока стабілізована напруга від блока живлення 4. Для збільшення потужності випромінювання використовується оптичний резонатор, який складається з плоского 5 і сферичного 6 дзеркал з багатошаровим діелектричним покриттям. Відбиваючись від дзеркал і багаторазово проходячи вздовж осі трубки, потік фотонів на своєму шляху втягує в процес індукованого випромінювання все більше атомів неону, що веде до зростання інтенсивності генерованого випромінювання.

Через складну структуру енергетичних рівнів 3 і 2 (наявність підрівнів) збуджений атом неону випромінює декілька різних довжин хвиль в

інфрачервоному та видимому діапазонах. Багатошарова структура дзеркал внаслідок інтерференції забезпечує необхідний коефіцієнт відбивання для однієї довжини хвилі. Цьому сприяє також підбір відповідної довжини оптичного резонатора. Через це випромінювання лазера монохроматичне, тобто генерується хвиля певної довжини.

Газорозрядна трубка 1 закрита з торців (з двох кінців) плоскопаралельними скляними пластинками 7, які встановлені під кутом Брюстера до осі трубки, що створює умови для повної поляризації лазерного випромінювання.

Таким чином, індуковане випромінювання лазера характеризується високим ступенем когерентності, строгою монохроматичністю, достатньо великою потужністю, повною поляризованістю, вузькою спрямованістю.

Лазерний промінь широко використовується в ролі скальпеля, який має цілий ряд переваг перед звичайним, а саме: дозволяє проводити майже безкровні операції, забезпечує повну стерильність, покращує поле зору для хірурга. Лазерна мікрохірургія ока успішно лікує глаукому, катаракту, відшарування сітківки. Лазерний скальпель використовується також для проведення хірургічних операцій з урології, кардіології, дерматології та ін. Застосовується лазерний промінь і з діагностичною метою для одержання об'ємного зображення внутрішньої порожнини шлунку, для судинної ендоскопії (ангіоскопії). Для визначення основних характеристик лазерного випромінювання - довжини хвилі та енергії кванта – використовується явище дифракції лазерного променя на дифракційній решітці.

Дифракційна решітка. являє собою скляну пластину, на якій через рівні проміжки а нанесені непрозорі штрихи шириною b . Проміжки між штрихами є по суті прозорими щілинами, які пропускають світло. Сумарна ширина штриха і щілини називається сталою або періодом d дифракційної решітки, тобто $d = a + b$.

При освітленні решітки монохроматичним світлом, яким є лазерний промінь, на екрані спостерігається дифракційна картина. у вигляді темних 1

світливих областей, які чергуються. Світлі області відповідають головним дифракційним максимумам 1, 2 і т.д., що існують при умові формули (1.6):

$$dsina = K\lambda, \quad (1.6)$$

де K – порядковий номер максимуму;

λ – довжина світлової хвилі.

Головні максимуми розташовані симетрично по обидві сторони від центрального або нульового максимуму ($K=0$, $a=0$).

Знаючи d , a і K , можна визначити довжину λ хвилі світла, яке падає на дифракційку решітку, за формулою (1.7):

$$\lambda = \frac{dsina}{K}. \quad (1.7)$$

Для знаходження кута a , під яким видно головний максимум K -го порядку, скористаємося тригонометричними поняттями (рис. 1.4).

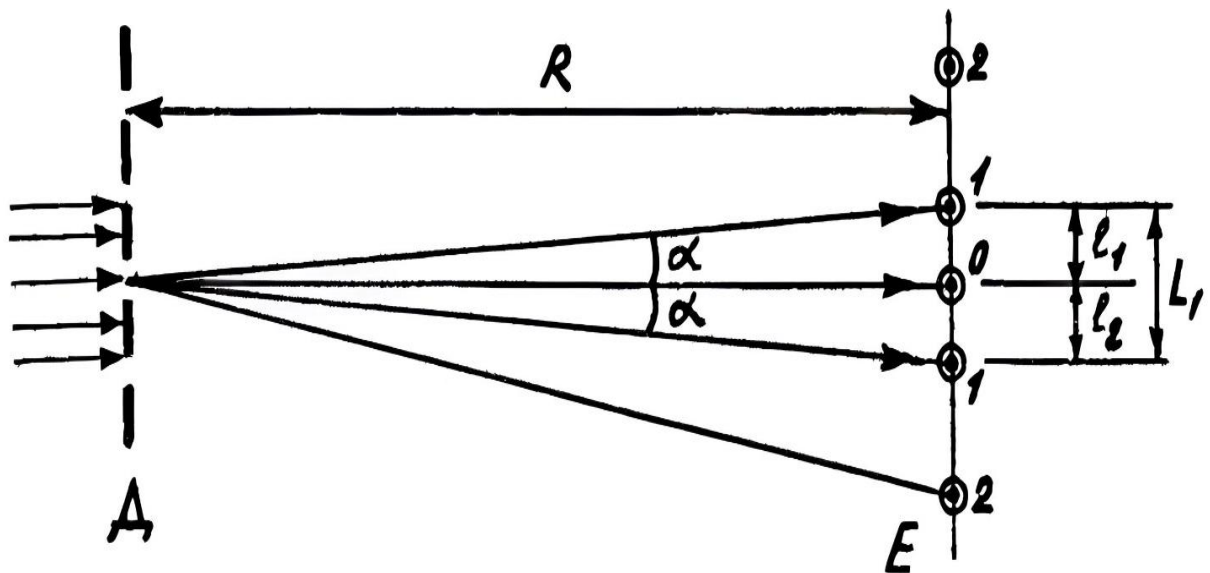


Рисунок 1.4. – Знаходження кута a

З рисунка 1.4 очевидно, що для першого максимуму кут знаходиться за формулою (1.8):

$$\operatorname{tga}_1 = \frac{l_1}{R} = \frac{L_1}{2R}, \quad (1.8)$$

де R – віддаль між дифракційною решіткою та екраном;

L_1 – віддаль між центрами максимумів першого порядку (1.9).

$$l_1 = l_2 = \frac{L_1}{2}. \quad (1.9)$$

Можна виміряти L_K між центрами максимумів інших порядків. В загальному випадку можна записати формулу (1.10):

$$\operatorname{tga}_K = \frac{L_K}{2R}. \quad (1.10)$$

За знайденим з дослідів tga за допомогою таблиці синусів (при малих a $\operatorname{tga} = \sin a$) знаходиться значення $\sin a$, яке використовується у формулі (1.6).

Енергія кванта лазерного випромінювання обчислюється за формулою (1.11):

$$\varepsilon = h\nu = \frac{hc}{\lambda}, \quad (1.11)$$

де h – постійна Планка;

c – швидкість світла у вакуумі.

Схема установки зображена на рисунку 1.5.

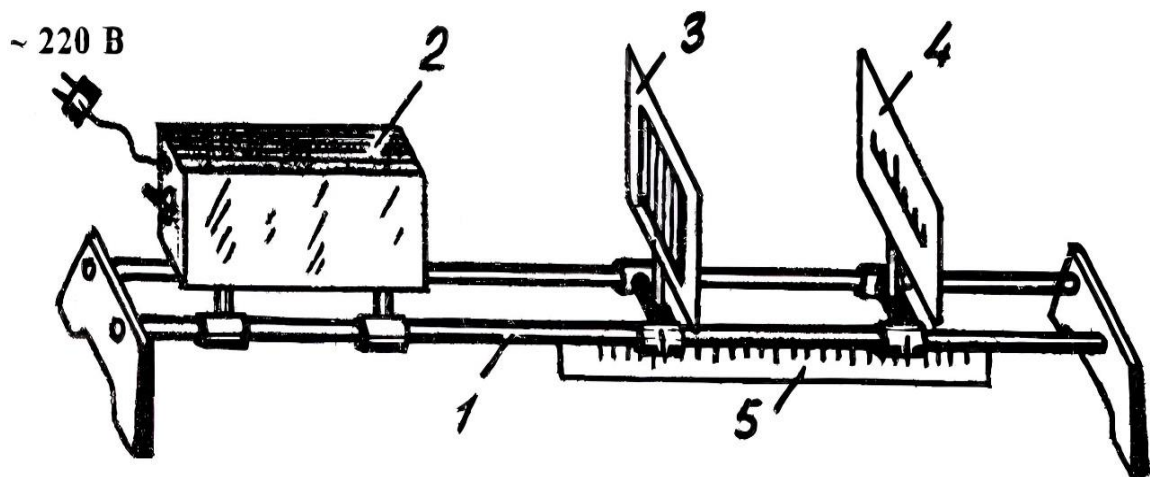


Рисунок 1.5. – Схема установки

Установка змонтована на оптичній лаві 1. Поблизу вихідного вікна 2 лазера встановлюється дифракційна решітка 3. Одержана дифракційна картина спостерігається на екрані 4, який має міліметрову шкалу. Віддаль між екраном і дифракційною решіткою вимірюється за допомогою лінійки 5.

1.3 Огляд існуючих аналогів

Комп'ютерні симуляції фізичних процесів справді стали важливою складовою сучасного освітнього процесу, і їхнє використання з кожним роком зростає. Завдяки таким симуляціям можна наочно та інтерактивно досліджувати складні явища, які важко або неможливо відтворити в лабораторних умовах через технічні обмеження або високу вартість експериментів. Вони також дозволяють зберегти безпеку студентів і викладачів під час виконання експериментів, що можуть бути небезпечними або вимагати дорогого обладнання.

Переваги використання комп'ютерних симуляцій:

- Інтерактивність та візуалізація: симуляції дозволяють користувачам взаємодіяти з фізичними моделями в реальному часі, що допомагає

зрозуміти процеси, які важко описати словами чи відтворити в реальному житті.

- Доступність та зручність: завдяки інтернету та персональним комп'ютерам, доступ до симуляцій став набагато простішим. Студенти можуть працювати з ними вдома, що дозволяє більш глибоко вивчати матеріал без обмежень навчального часу.
- Безпека: вивчення фізичних явищ, таких як експлозії, високі температури чи радіація, може бути небезпечним у реальних умовах. Комп'ютерні симуляції дозволяють досліджувати ці процеси без ризику для здоров'я.

Основні види симуляцій, які використовуються в навчанні фізики:

1. Механічні симуляції: ці симуляції дозволяють вивчати рух тіл, сили, моменти та інші механічні процеси. Наприклад, симуляції на платформі PhET, які дозволяють досліджувати закони Ньютона, рухи тіл під дією сил, роботу та енергію.
2. Електричні і магнітні поля: симуляції, які допомагають вивчати електромагнітні процеси, наприклад, взаємодію заряджених часток або магнітне поле, що створюється струмами.
3. Теплові процеси: вивчення теплових явищ, таких як теплопередача, зміна температури, робота теплових машин, може бути зручно реалізовано через симуляції, які моделюють ці процеси в різних матеріалах.
4. Квантова фізика: для студентів старших курсів важливі симуляції, що допомагають зрозуміти принципи квантової механіки, поведінку атомів, молекул і часток на мікроскопічному рівні.

PhET Interactive Simulations, Vascak і Algodoo є широко визнаними платформами для моделювання фізичних явищ та вивчення основних принципів фізики. Вони активно використовуються у навчальних і дослідницьких цілях завдяки своїй доступності та функціональності.

1.3.1 PhET Interactive Simulations

Одним із найпопулярніших і найбільш визнаних інструментів для моделювання фізичних процесів є платформа PhET Interactive Simulations [5], створена в Університеті Колорадо. Ця платформа пропонує більше 80 інтерактивних симуляцій, кожна з яких орієнтована на вивчення конкретних аспектів фізики, таких як механіка, електрика, магнетизм, термодинаміка, хвильові процеси, оптика та квантові явища. Кожна симуляція розроблена таким чином, щоб забезпечити користувачам можливість вивчати фізичні явища шляхом моделювання, маніпулювання параметрами та аналізу отриманих результатів [5].

PhET створює симуляції з урахуванням освітніх потреб різних вікових груп, зокрема учнів старшої школи, студентів бакалаврату та навіть аспірантів. Проте завдяки простому й інтуїтивному інтерфейсу платформа також підходить для учнів молодших класів, забезпечуючи базове розуміння фізичних явищ через інтерактивний підхід. Всі симуляції PhET проходять етапи ретельного тестування, щоб відповідати освітнім стандартам та бути зручними у використанні як для самостійного навчання, так і для роботи під керівництвом викладачів [6].

Інтерфейс розроблено таким чином, щоб користувачі могли легко змінювати параметри моделей, спостерігати результати й отримувати візуалізоване представлення фізичних законів. Як приклад, на рисунку 1.6 представлено типовий вигляд інтерфейсу однієї з симуляцій.

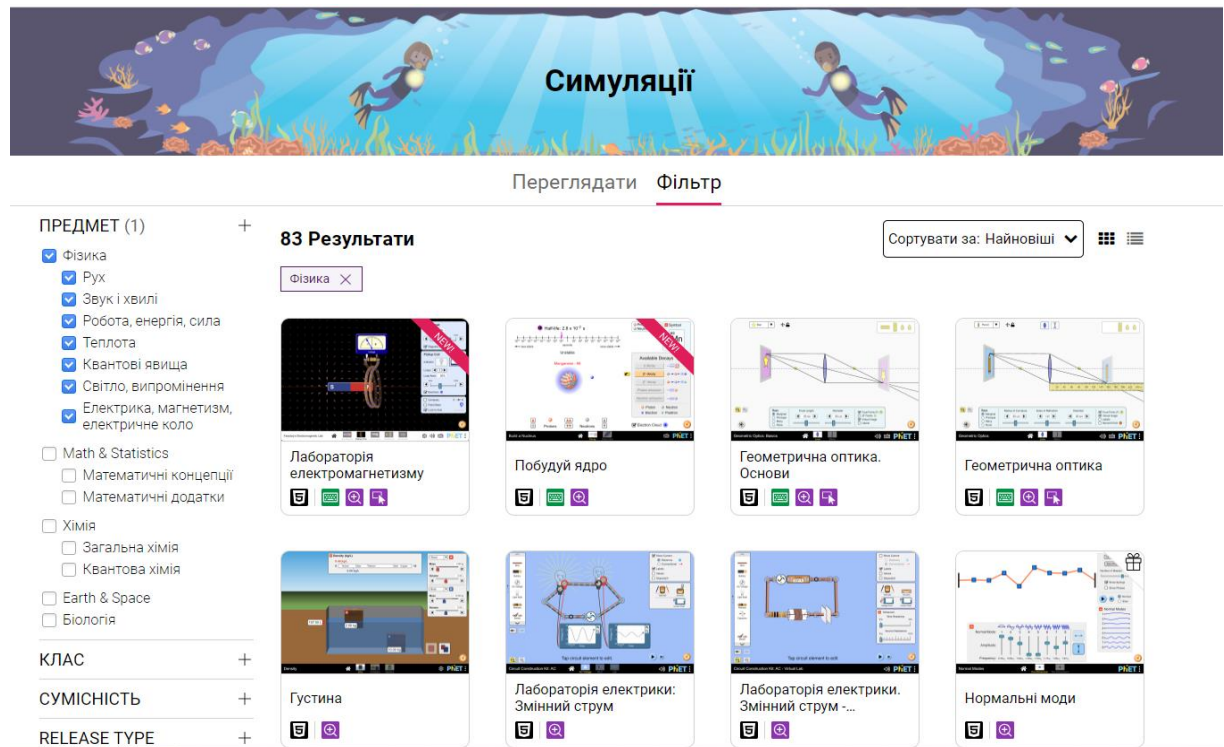


Рисунок 1.6 – Інтерфейс платформи «PhET»

Платформа PhET Interactive Simulations має низку технічних характеристик, які забезпечують її функціональність і універсальність для навчальних цілей. Основою роботи симуляцій є технологія HTML5, що гарантує їхню сумісність із сучасними веб-браузерами та широким спектром пристроїв, включаючи комп'ютери, планшети та смартфони. Це дозволяє користувачам запускати симуляції безпосередньо в браузері, без необхідності встановлення додаткового програмного забезпечення. Для тих, хто віддає перевагу офлайн-доступу, симуляції можна завантажувати на локальні пристрої [7].

Функціонал платформи включає інтерактивні елементи, які дозволяють змінювати параметри фізичних або математичних моделей у реальному часі. Наприклад, користувач може регулювати величини, такі як маса, сила, швидкість, температура або напруга, і спостерігати, як ці зміни впливають на симульовану систему. Графічний інтерфейс кожної симуляції розроблений з урахуванням інтерактивності та зручності користування, включаючи елементи візуалізації даних, графіків і анімацій, які адаптуються до введених параметрів.

PhET також підтримує мультимовний інтерфейс і локалізацію, що реалізується через інтеграцію багатомовних текстових ресурсів, зокрема назв елементів, підказок і описів. Це здійснено через модульну структуру коду, яка дозволяє розробникам та волонтерам легко додавати переклади без зміни основної функціональності.

Регулярні оновлення платформи спрямовані на вдосконалення коду, оптимізацію продуктивності та впровадження нових функцій. Наприклад, в останніх версіях були додані поліпшення графіки, адаптація для сенсорних екранів і забезпечення сумісності з новими стандартами браузерів. Всі симуляції підтримують механізм інтеграції з іншими освітніми платформами, такими як системи керування навчанням (LMS), через використання стандартів, таких як SCORM або LTI, що полегшує їх використання в освітньому процесі.

Ці технічні аспекти забезпечують платформі високу гнучкість і функціональність, що дозволяє використовувати її в різних навчальних середовищах і на різних пристроях.

1.3.2 Vascak

Іншим значущим інструментом для моделювання фізичних процесів є платформа Vascak [8], розроблена чеським фізиком і програмістом Володимиром Васцаком. Платформа містить понад 100 симуляцій і анімацій, які охоплюють широкий спектр фізичних явищ, включаючи магнітні поля, хвильову оптику, механіку, термодинаміку та аспекти ядерної фізики. Інтерфейс платформи зображений на рисунку 1.7.

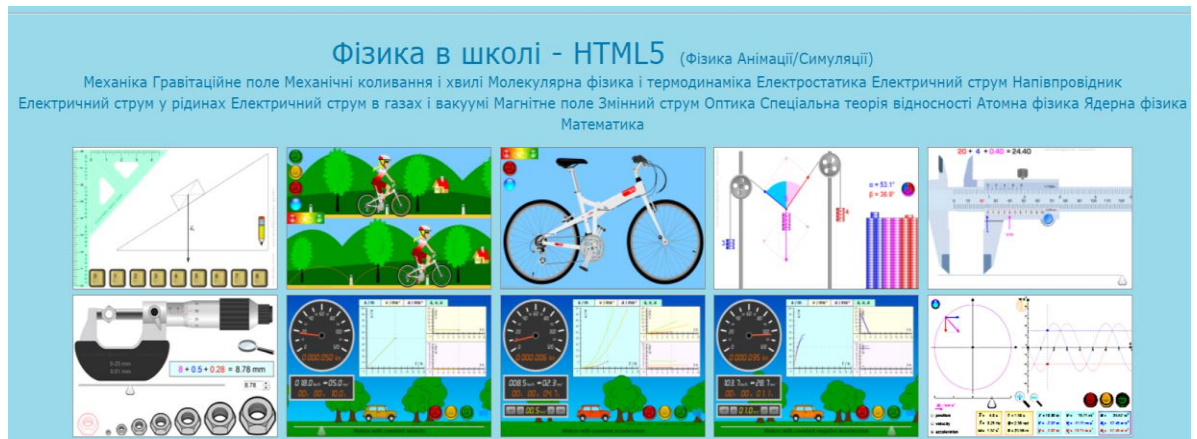


Рисунок 1.7 – Платформа «Vascak»

Симуляції платформи відзначаються деталізованим технічним підходом до відтворення фізичних процесів, що дозволяє моделювати явища з високим ступенем точності. Кожна симуляція супроводжується параметричними настройками, які дозволяють користувачам налаштовувати значення змінних, таких як напруженість магнітного поля, довжина хвилі, швидкість частинок тощо.

Технічна реалізація платформи базується на сучасних веб-технологіях, що забезпечує її сумісність із більшістю операційних систем і браузерів. Vascak підтримує як онлайн-режим, так і можливість офлайн-доступу через завантаження окремих симуляцій на пристрої користувача. Це дозволяє застосовувати платформу як в аудиторному навчанні, так і для самостійної роботи [9].

Інтерфейс платформи характеризується структурованістю та логічною організацією, де кожна симуляція має чіткий поділ на зони управління параметрами, зони візуалізації та графічного аналізу. Візуалізації мають високу якість та інтерактивність, дозволяючи користувачам змінювати параметри "на ходу" та спостерігати за відповідними змінами в анімації.

Додатково Vascak інтегрує інструменти для експорту результатів моделювання у вигляді таблиць, графіків або зображень, що робить її корисною не лише для освітніх цілей, але й для проведення простих наукових досліджень.

Платформа Vascak виконує важливу роль у забезпеченні доступу до якісних симуляцій фізичних явищ, особливо для освітніх закладів із обмеженими фінансовими ресурсами. Завдяки широкому спектру доступних симуляцій вона є корисним інструментом для викладачів і студентів. Однак поточний функціонал платформи може бути вдосконалений для поліпшення зручності користування та розширення її можливостей, зокрема через модернізацію інтерфейсу, оптимізацію роботи на різних пристроях і впровадження додаткових інтерактивних функцій.

1.3.3 Algodoo

Algodoo – це інтерактивна платформа для моделювання фізичних явищ у двовимірному середовищі, орієнтована на освітні та розважальні цілі. Вона дозволяє користувачам створювати, досліджувати та аналізувати різноманітні фізичні системи через симуляції, що охоплюють основи механіки, динаміки, енергії, зіткнень і руху тіл. Завдяки простому і візуально зрозумілому інтерфейсу платформа використовується як інструмент для інтерактивного навчання, особливо для школярів і студентів [10].

Технічно Algodoo побудована на основі фізичного рушія, який забезпечує точне відтворення взаємодії об'єктів у двовимірному просторі. Користувачі мають можливість створювати об'єкти різної форми, налаштовувати їхні фізичні параметри, такі як маса, густина, коефіцієнт тертя та пружності. Платформа дозволяє легко моделювати складні системи, включаючи маятники, катапульти, гідравлічні системи або прості машини, і спостерігати за їхньою поведінкою у реальному часі [11].

Однією з ключових особливостей Algodoo є її інтуїтивний інтерфейс, який підтримує взаємодію шляхом "перетягування" об'єктів, налаштування властивостей за допомогою панелей інструментів і миттєвого відображення результатів. Візуалізації включають використання кольорових маркерів для

позначення сили, енергії або швидкості, що сприяє кращому розумінню фізичних принципів. Інтерфейс програми зображено на рисунку 1.8.

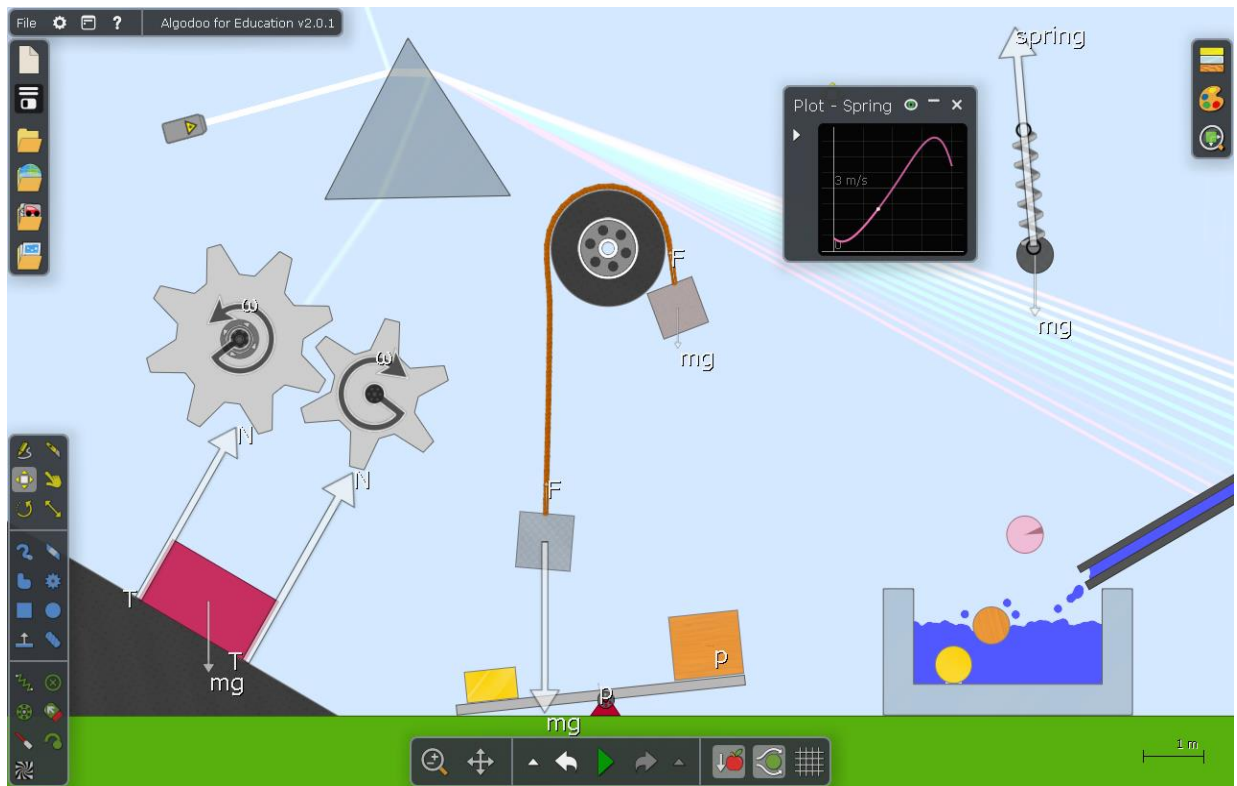


Рисунок 1.8 – Платформа «Algodoo»

Програма підтримує роботу на персональних комп'ютерах з операційними системами Windows і macOS, а також пропонує додаткові функції для інтерактивних дошок, що робить її зручною для використання в класах. Крім того, користувачі можуть зберігати створені симуляції у вигляді проєктів, ділитися ними з іншими або завантажувати симуляції, створені спільноту.

Algodoo також включає вбудовані навчальні матеріали та готові симуляції, які можуть бути використані для викладання фізики або STEM-дисциплін. Її можливості моделювання часто застосовуються для пояснення базових концепцій, таких як закони Ньютона, збереження енергії або сили тертя.

Таким чином, Algodoo є універсальним і гнучким інструментом, який дозволяє ефективно поєднувати навчання з експериментами, стимулюючи інтерес до фізики та інженерії у користувачів.

1.4 Оцінка та порівняння існуючих аналогів

Оцінка та порівняння платформ PhET Interactive Simulations, Vascak та Algodoo показують, що кожна з них має свої унікальні переваги та технічні особливості, що робить їх корисними в різних контекстах навчання фізики та суміжних дисциплін.

PhET Interactive Simulations:

- Переваги:

1. Широкий спектр тем: Платформа охоплює понад 80 симуляцій з різних фізичних дисциплін — механіка, електрика, термодинаміка, квантова фізика тощо. Це дає можливість використовувати її для вивчення різних аспектів фізики.
2. Науково обґрунтовані моделі: Симуляції розроблені на основі наукових досліджень, що забезпечує їхню точність та відповідність реальним фізичним явищам.
3. Доступність: Платформа безкоштовна та доступна для користувачів по всьому світу. Можливість використовувати симуляції онлайн або завантажувати їх для офлайн використання.
4. Інтерфейс та підтримка: Простий і інтуїтивно зрозумілий інтерфейс, а також багатомовна підтримка роблять платформу доступною для різних користувачів, включаючи тих, хто не володіє англійською мовою.
5. Регулярні оновлення: Постійно оновлюється і підтримується розробниками, що забезпечує актуальність симуляцій та нові можливості для користувачів.

Недоліки:

1. Обмеження за кількістю тем: Хоча PhET пропонує великий вибір симуляцій, деякі спеціалізовані фізичні явища або деталі можуть бути недостатньо висвітлені або не представлені.
2. Обмежена гнучкість в налаштуванні параметрів: У порівнянні з іншими платформами, PhET не дає такої гнучкості в налаштуванні численних параметрів моделей, як, наприклад, Vascak.

Vascak:

Переваги:

1. Спеціалізовані симуляції: Платформа пропонує понад 100 симуляцій для глибокого вивчення специфічних фізичних явищ, таких як магнітні поля, оптика та ядерна фізика, що робить її чудовим інструментом для вищих навчальних закладів.
2. Висока точність налаштувань: Користувачі можуть детально налаштовувати параметри моделей, що дозволяє точніше досліджувати складні фізичні процеси.
3. Багато фізичних тем: Величезний спектр фізичних тем дає можливість глибше досліджувати різні аспекти фізики, включаючи явища, що не покриваються іншими платформами.

Недоліки:

1. Складність для новачків: Через велику кількість налаштувань і складних тем, Vascak може бути важким для користувачів, які тільки починають вивчати фізику.

2. Застарілий інтерфейс: Інтерфейс платформи є менш сучасним, що може знижувати зручність користування.

3. Необхідність вдосконалення: Платформа потребує оновлення для підвищення зручності та інтерактивності, а також адаптації до нових технологій і пристроїв.

Algodoo:

Переваги:

1. Інтуїтивно зрозумілий інтерфейс: Платформа має простий і зручний інтерфейс, що дозволяє легко створювати фізичні моделі та маніпулювати параметрами.

2. Гнучкість у створенні моделей: Algodoo дозволяє користувачам створювати та експериментувати з механічними системами різної складності, що дає змогу вивчати базові закони механіки.

3. Навчальні можливості: Платформа ідеально підходить для навчання в школах та університетах, особливо для молодших класів, завдяки своїй простоті та інтерактивності.

4. Безкоштовність та доступність: Algodoo є безкоштовною і доступною для широкого кола користувачів.

Недоліки:

1. Обмеження за темами: Платформа фокусується лише на механічних моделях, що обмежує її застосування для вивчення інших аспектів фізики, таких як електрика чи квантова фізика.

2. Мало інструментів для вищого рівня: Для більш складних наукових досліджень або детального аналізу фізичних явищ, Algodoo не пропонує достатньо складних функцій і можливостей.

3. Малий вибір готових симуляцій: Хоча платформа дозволяє створювати моделі, кількість готових симуляцій на платформі є обмеженою, що може вимагати додаткових зусиль від користувачів для створення власних моделей.

Порівняльна характеристика цих платформ зображена на таблиці 1.1.

Таблиця 1.1 – Ключові характеристики порівнянних аналогів

Характеристика	PhET	Vascak	Algodoo
Кількість симуляцій	Високий рівень (80+ симуляцій)	Високий рівень (100+ симуляцій)	Середній рівень

			(обмежена кількість)
Якість графіки	Висока	Середня	Середня
Інтерактивність	Висока	Низька	Висока
Мобільний додаток	Так	Ні	Так
Україномовна підтримка	Так	Так	Ні
Безкоштовність	Так	Так	Так
Доступність тем	Високий (різні теми фізики)	Середній (фізичні явища)	Середній (механіка)
Інтуїтивність інтерфейсу	Висока	Середня	Висока
Технічні вимоги	Високі	Низькі	Середні

Критерії оцінювання визначаються за шкалою:

- Високий: Значення характеристики перевищує базовий рівень, забезпечуючи високі показники продуктивності або ефективності.
- Середній: Характеристика знаходиться на прийнятному рівні, достатньому для виконання завдань, але не є оптимальною.
- Низький: Характеристика обмежена або недостатня для повноцінного виконання завдань, що може спричинити труднощі у використанні.

З даного порівняння можна зробити висновок, що:

1. Кількість симуляцій: PhET має найбільшу кількість симуляцій, охоплюючи широкий спектр фізичних тем. Vascak також пропонує значну кількість, але спеціалізується на більш конкретних областях. Algodoo має меншу кількість симуляцій, фокусуючись на механіці.
2. Якість графіки: PhET має високу якість графіки з чудовою візуалізацією. Vascak надає менш привабливу графіку, але вона функціональна для наукових цілей. Algodoo має середню графіку, яка достатня для двовимірних механічних моделей.

3. Інтерактивність: Всі три платформи мають хороший рівень інтерактивності, але PhET і Algodoo забезпечують більш гнучке і зручне взаємодія з користувачем.
4. Мобільний додаток: Algodoo і PhET має мобільний додаток, тоді як Vascak доступний лише через веб-браузер.
5. Україномовна підтримка: PhET і Vascak має українську мову в налаштуваннях, що робить її доступною для україномовних користувачів.
6. Безкоштовність: Усі три платформи є безкоштовними для користувачів.
7. Доступність тем: PhET охоплює найбільший спектр фізичних тем, тоді як Vascak і Algodoo обмежені конкретними напрямками (фізичні явища та механіка).
8. Інтуїтивність інтерфейсу: PhET і Algodoo мають найбільш інтуїтивно зрозумілий інтерфейс, що забезпечує легкість використання. Vascak має менш зручний інтерфейс, що може вимагати додаткового часу для освоєння.
9. Технічні вимоги: Vascak має найменші технічні вимоги, що робить її доступною для більшості пристроїв, тоді як PhET і Algodoo потребують середніх ресурсів для ефективного використання.

Отже,

- PhET є універсальною платформою з великим вибором тем та простим інтерфейсом, підходить для всіх вікових груп і рівнів знань, але має обмеження у гнучкості налаштувань.
- Vascak підходить для глибокого вивчення складних фізичних процесів і забезпечує велику гнучкість налаштувань, але є менш зручним для новачків і потребує оновлення інтерфейсу.
- Algodoo є чудовим інструментом для вивчення механіки та базових фізичних принципів, але має обмеження в темах і функціональності для більш складних моделей.

Кожна з цих платформ має свої переваги і недоліки, і вибір залежить від потреб користувача та рівня його знань.

1.5 Висновки до першого розділу

У першому розділі було проведено загальну характеристику об'єкту дослідження — емуляторів фізичних процесів. Ключовими аспектами розробки таких емуляторів є точність та продуктивність. Для моделювання складних фізичних явищ, таких як, наприклад, багатокомпонентні системи або явища, що потребують високої роздільної здатності, необхідно використовувати ефективні числові методи, наприклад, методи кінцевих елементів або метод Монте-Карло, які дозволяють досягти високої точності моделювання. Крім того, з огляду на великі обсяги даних і складність обчислень, для підвищення продуктивності важливо застосовувати паралельні обчислення, що дозволяє значно скоротити час на виконання симуляцій.

Однак, при створенні таких моделей важливо знайти баланс між спрощенням моделі для прискорення обчислень та збереженням фізичної реалістичності. Інакше можна отримати модель, яка буде працювати швидше, але не відображатиме реальні фізичні процеси. Тому важливою є також задача збереження фізичної достовірності симуляцій, щоб отримані результати були придатними для наукових досліджень, інженерії, медицини, фізики та інших галузей науки.

Емулятори фізичних процесів використовуються в широкому спектрі застосувань, включаючи наукові дослідження, інженерію, медичні технології (наприклад, візуалізація фізичних процесів у біології чи медичних приладах), а також для розробки нових технологій. Зокрема, вони можуть використовуватися для моделювання складних фізичних систем, що дає можливість тестувати теорії або оцінювати ефективність нових інженерних рішень до їх впровадження в реальних умовах.

Далі розглянуто теоретичні відомості двох лабораторних робіт, на основі яких буде проводитися розробка емуляторів. Обрані лабораторії сприяють розвитку навичок у створенні моделей фізичних процесів і використуваних

для цього методів. Важливою умовою є створення інтерактивного, зрозумілого та інформативного інтерфейсу для користувачів, оскільки перевантаження інтерфейсу або незрозумілість у роботі з програмним забезпеченням можуть стати суттєвими перепонами для ефективного навчання.

Також були розглянуті існуючі аналоги емуляторів фізичних процесів, їх переваги та недоліки. Для цього було проведено оцінку та порівняння найбільш відомих платформ, таких як PhET Interactive Simulations, Vascak і Algodoo. Оцінка показала, що хоча ці інструменти мають високий потенціал для навчання і досліджень, жоден з них не відповідає повністю вимогам щодо функціональності, гнучкості та зручності для більш глибоких наукових досліджень.

Усі ці платформи забезпечують інтерактивність, доступність та безкоштовність, але кожна має свої обмеження:

1. PhET має великий набір тем і високу якість графіки, але обмежену гнучкість налаштувань та недостатньо складні симуляції для деяких спеціалізованих фізичних процесів.
2. Vascak має хорошу гнучкість і можливість налаштувань для детального моделювання, але його інтерфейс є застарілим і може бути складним для користувачів з початковим рівнем знань.
3. Algodoo забезпечує легкість у використанні, але має обмежену кількість тем і можливостей для створення складних моделей.

З огляду на це, існуючі аналоги не задовольняють повністю потреби для розробки емуляторів, що використовуються для навчальних і дослідницьких цілей у більш складних фізичних науках. Це вказує на необхідність розробки власних емуляторів, що можуть бути адаптовані до конкретних вимог і цілей, зокрема забезпечити:

- високу точність моделінгу;
- гнучкість налаштувань;
- інтуїтивно зрозумілий інтерфейс;

- ефективність в обчисленнях через використання сучасних методів числового моделювання та паралельних обчислень.

Таким чином, дослідження та порівняння існуючих платформ дозволяють зробити висновок, що для повного задоволення потреб навчальних і наукових завдань необхідно розробити нові емулятори, які поєднуюватимуть найкращі аспекти доступних платформ та надаватимуть можливість для більш точних і складних фізичних симуляцій.

2 ВИБІР ЗАСОБІВ ТА СЕРЕДОВИЩА ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір мови програмування

Серед засобів програмування для розробки програмного забезпечення емуляторів фізичних процесів варто виділити JavaScript та PHP через їхні унікальні можливості та переваги. JavaScript є ключовою мовою для розробки інтерактивних веб-додатків, завдяки своїй здатності виконувати код безпосередньо в браузері користувача, що дозволяє створювати швидкі та динамічні емулятори з високим рівнем інтерактивності. Можливість інтеграції з бібліотеками для математичних обчислень, такими як `math.js`, та підтримка асинхронних операцій дають змогу створювати реалістичні та масштабовані симуляції фізичних процесів в реальному часі. З іншого боку, PHP є потужним серверним мовою програмування, що дозволяє обробляти складні обчислення на сервері, зберігаючи високу продуктивність при обробці великих обсягів даних, і надає зручність для зберігання та маніпуляції даними в базах даних. Обидві мови поєднують зручність використання та широкі можливості для інтеграції з іншими веб-технологіями, що робить їх оптимальними для створення ефективних та інтерактивних емуляторів фізичних процесів в рамках веб-платформ [12].

JavaScript — це мова програмування, яка зазвичай використовується для додавання інтерактивних елементів до веб-сторінок, розробки веб-додатків, а також для створення серверних застосунків і навіть відеоігор. Вона є однією з основних технологій веб-розробки, поряд із HTML та CSS, і дозволяє створювати динамічний контент, що взаємодіє з користувачем без необхідності оновлення сторінки.

JavaScript є легкою у освоєнні мовою, яка може виконуватись на стороні клієнта в браузері, що забезпечує швидке і безперервне реагування веб-

сторінок на дії користувачів. Мова також використовується для створення серверних додатків через технології, як Node.js, що дозволяє використовувати JavaScript для серверного програмування.

У JavaScript є широкі можливості для роботи з асинхронними операціями, що робить його ідеальним для розробки високопродуктивних веб-додатків, ігор та інтерактивних веб-сайтів. За допомогою бібліотек та фреймворків, таких як React, Angular або Vue.js, можна створювати складні інтерфейси та веб-додатки. [13].

Мова JavaScript спочатку була створена лише як мова для браузера, але зараз її також використовують в інших середовищах. Сьогодні JavaScript вважається найбільш поширеною мовою для браузера, яка повністю інтегрована з HTML/CSS [14].

Широке застосування JavaScript у розробці веб-додатків, мобільних додатків та ігор робить його цінною мовою для вивчення.

Розробка JavaScript-фреймворків, що складаються з бібліотек JavaScript-коду, дозволяє розробникам використовувати вже написаний JavaScript-код у своїх проектах. Це економить час і зусилля, оскільки не потрібно писати код з нуля. Кожен JavaScript-фреймворк має функції, які мають на меті спростити процес розробки та налагодження.

Наприклад, інтерфейсні JavaScript-фреймворки, такі як jQuery та ReactJS, підвищують ефективність проектування. Вони дозволяють розробникам повторно використовувати та оновлювати компоненти коду, не впливаючи один на одного функціонально або за значенням.

Тим часом фреймворки для розробки мобільних додатків, такі як Cordova і Titanium, дають можливість створювати нативні або гібридні додатки. Реалізація JavaScript-коду на Node.js також відіграє важливу роль у веб-розробці. Node.js може скоротити час відповіді сервера завдяки своїй однопотоківій природі та неблокуючій архітектурі і не допускати затримок.

Node.js також достатньо легкий, щоб слугувати масштабованим інструментом для мікросервісів, дозволяючи розробляти єдиний додаток, що складається з невеликих сервісів з окремими процесами.

Однією з основних функцій JavaScript є додавання динамічності веб-сторінкам. Це включає відображення анімації, зміну видимості тексту та створення випадających меню [15].

Можливість використовувати лише HTML та CSS для створення веб-сайту матиме лише статичне відображення. За допомогою JavaScript користувач може взаємодіяти з веб-сторінками і мати кращий досвід перегляду. Крім того, JavaScript дозволяє змінювати вміст HTML і значення атрибутів без попереднього перезавантаження веб-сторінки.

Як і всі мови програмування, JavaScript має певні переваги та недоліки, які слід враховувати. Багато з них пов'язані з тим, що JavaScript часто виконується безпосередньо у браузері клієнта. Але зараз існують інші способи використання JavaScript, які дозволяють йому мати ті ж переваги, що й у серверних мовах.

Перевагами мови програмування Javascript є:

- JavaScript має тенденцію бути дуже швидким, оскільки часто виконується безпосередньо у браузері клієнта. Доки він не потребує зовнішніх ресурсів, JavaScript не сповільнюється через звернення до внутрішнього сервера. Крім того, всі основні браузери підтримують компіляцію JavaScript за технологією JIT (just in time), що означає, що немає необхідності компілювати код перед його запуском;
- Синтаксис JavaScript був натхненний мовою Java, і його відносно легко вивчити порівняно з іншими популярними мовами, такими як C++;
- JavaScript є скрізь в Інтернеті, а з появою Node.js все частіше використовується на бекенді. Існує незліченна кількість ресурсів для вивчення JavaScript. І StackOverflow, і GitHub демонструють зростання кількості проектів, які використовують JavaScript, і очікується, що популярність, яку він здобув за останні роки, тільки зростатиме;

- Існує багато способів використання JavaScript через сервери Node.js. Якщо завантажити Node.js за допомогою Express, використовувати базу даних документів, наприклад MongoDB, і використовувати JavaScript на фронтенді для клієнтів, можна розробити цілий JavaScript-додаток від початку до кінця, використовуючи лише JavaScript;
- З моменту появи ECMAScript 5 (специфікація сценаріїв, на яку спирається JavaScript), ECMA International щорічно оновлює JavaScript. Наразі спільнота отримала підтримку браузером ES6 у 2017 році і сподівається на підтримку ES7 у майбутньому.

Недоліками мови програмування JavaScript є:

- Оскільки код JavaScript виконується на стороні клієнта, помилки та недогляди іноді можуть бути використані в зловмисних цілях. Через це деякі люди вирішують повністю вимкнути JavaScript;
- Хоча серверні скрипти завжди видають однаковий результат, різні браузери іноді інтерпретують JavaScript-код по-різному. Сьогодні ці відмінності мінімальні, і вам не варто про них турбуватися, якщо ви тестуєте свій скрипт у всіх основних браузерах.

JavaScript — це потужний інструмент, який став невід'ємною частиною сучасної розробки. Завдяки його гнучкості, багатству можливостей та активній спільноті, він залишається одним із лідерів у світі програмування.

Іншою мовою програмування є PHP - це серверна мова програмування з відкритим вихідним кодом, яку можна використовувати для створення веб-сайтів, додатків, систем управління взаємовідносинами з клієнтами тощо. Це широко використовувана мова загального призначення, яку можна вбудовувати в HTML. Така функціональність з HTML означає, що мова PHP залишається популярною серед розробників, оскільки вона допомагає спростити код HTML [16].

PHP розшифровується як «PHP: Гіпертекстовий препроцесор», а первісна аббревіатура PHP означала “Персональна домашня сторінка” (англ.

Personal Home Page). Абревіатура змінювалася в міру розвитку мови з моменту її запуску в 1994 році, щоб точніше відображати її природу.

Незважаючи на те, що PHP є старою мовою програмування і їй бракує деяких функцій новіших аналогів, вона продовжує розвиватися і вдосконалюватися протягом багатьох років. Простота використання, відмінна документація та надійна підтримка роблять її улюбленою серед розробників.

Розробники PHP мають доступ до широкого спектру фреймворків, баз даних і бібліотек, а мову можна легко встановити на будь-яку ОС Linux, Windows або Unix. Крім того, вона сумісна з усіма провідними базами даних, такими як MySQL, SQLite та ODBC, а також з більшістю серверів, включаючи Apache та IIS.

Вона може додатково підтримуватися популярними PHP-фреймворками, такими як Laravel, CodeIgniter і Symfony, а також багатьма добре укомплектованими і перевіреними бібліотеками. Добре задокументовані API PHP дозволяють легко інтегруватися з усіма програмами та доповненнями до CMS, що забезпечує динамічний, інтерактивний та багатофункціональний досвід роботи.

Останні версії цієї скриптової мови, такі як PHP 7 і PHP 8, є об'єктно-орієнтованими, що допомагає розробникам створювати безпечні фреймворки і функції, які є динамічними і придатними для багаторазового використання.

Перевагами мови програмування PHP є [17]:

- PHP має значну перевагу над багатьма іншими мовами програмування в тому, що він має повністю відкритий вихідний код. Це означає, що користувачам не потрібно витратити гроші на використання мови, фреймворків, інструментів налагодження, баз даних та інших функціональних можливостей. На відміну від мов із закритим вихідним кодом, які вимагають плати за використання, PHP можна завантажити і використовувати для створення веб-сайтів і веб-додатків без будь-яких попередніх витрат;

- Однією з помітних переваг PHP є його великі бібліотеки, які пропонують заздалегідь написаний і оптимізований код для безперебійного та ефективного виконання завдань. Ці бібліотеки позбавляють розробників необхідності починати кодування з нуля, дозволяючи їм запозичувати та інтегрувати код для максимізації швидкості та ефективності виконання своїх завдань. З PHP розробники мають доступ до великої колекції бібліотек, включаючи такі популярні варіанти, як Symfony, Django, CodeIgniter та інші;
- Гнучкість і адаптивність PHP роблять його найкращим вибором для популярних систем управління контентом, таких як WordPress. Мова сценаріїв може легко інтегруватися з різними веб-технологіями та платформами, що робить її ідеальним варіантом для широкого спектру проектів з веб-розробки. Гнучкість PHP дозволяє йому ефективно інтегруватися з багатьма іншими мовами програмування, що дає змогу розробникам використовувати найефективніші технології для кожної конкретної функції. Крім того, PHP є крос-платформенною мовою, що означає, що розробники можуть використовувати будь-яку основну операційну систему, включаючи Windows, Linux і macOS, для виконання кодування. Така гнучкість значно полегшує процес розробки, роблячи його швидшим та економічно ефективнішим;
- Сумісність PHP з широким спектром баз даних є однією з його основних переваг у веб-розробці. За допомогою PHP розробники можуть підключатися практично до будь-якого типу баз даних, включаючи MySQL, mSQL, MS-SQL, SQLite, PostgreSQL, ElasticSearch, Redis, MongoDB і багато інших. Це дає розробникам свободу вибору найбільш підходящої бази даних для їхніх додатків на основі таких факторів, як масштабованість, продуктивність та економічна ефективність;
- PHP - це дуже масштабована мова, яка може ефективно працювати з великими проектами. Багато відомих веб-сайтів і веб-додатків, таких як Facebook, WordPress, Yahoo і MailChimp, були розроблені за допомогою

PHP. Однією з причин, чому програмування на PHP підходить для великомасштабних проектів, є його здатність швидко завантажуватися, що призводить до скорочення часу очікування для користувачів, які отримують доступ до веб-сайту або додатку. Тому PHP є чудовим вибором для бізнесу або приватних осіб, яким потрібно розробляти складні та масштабовані веб-додатки.

Хоча PHP, безсумнівно, корисний у сфері веб-розробки, він також має кілька недоліків, які заважають йому домінувати в цій галузі. Задля неупередженого огляду, розглянемо ці недоліки та дізнаємось, як вони можуть зашкодити майбутньому програмному забезпеченню та його бізнес-впровадженню.

Недоліками мови програмування PHP є:

- PHP - це мова програмування, яку легко вивчити та опанувати розробникам. Однак, у порівнянні з іншими мовами програмування, такими як Python, PHP більше схожа на код, що робить її менш доступною для початківців. Через це розробники-початківці вважають за краще вивчати Python як свою першу мову і рідко розглядають можливість додати PHP до свого набору навичок;
- Доступність текстового файлу ASCII PHP-скрипту через його відкритий характер викликає занепокоєння щодо безпеки продуктів на основі PHP. Це дозволяє громадськості переглядати будь-який написаний код, включаючи помилки, які можуть бути використані зловмисниками. Втім, цей недолік є поширеним серед технологій з відкритим кодом, а не тільки для PHP;
- Налагодження є невід'ємною частиною будь-якої мови програмування, і PHP не є винятком. Однак розробники часто скаржаться, що PHP має обмежені засоби налагодження, що ускладнює роботу з помилками, особливо в порівнянні з іншими мовами сценаріїв. Інструменти, необхідні для відстеження і пошуку помилок в PHP, працюють не так ефективно, що може призвести до розчарування і трудомістких виправлень помилок.

Детальний порівняльний аналіз засобів програмування зображено на таблиці 2.1.

Таблиця 2.1 – Детальний порівняльний аналіз засобів програмування для розробки програмного забезпечення емуляторів фізичних процесів.

Критерій	Javascript	PHP
Тип виконання	Виконується на стороні клієнта (браузер) або на сервері (через Node.js).	Виконується на стороні сервера.
Призначення	Динамічний контент, анімація, веб-додатки, серверні та мобільні додатки.	Серверні веб-додатки, інтеграція з базами даних, створення CMS і бекенд-систем.
Переваги	Швидке виконання, гнучкість, велика кількість фреймворків, підтримка ES6.	Відкритий код, доступність, сумісність із різними базами даних, простота інтеграції з HTML.
Недоліки	Можливі проблеми з безпекою через виконання на клієнті, залежність від браузера.	Менша кількість інструментів для налагодження, потенційні проблеми з безпекою (відкритий код).
Популярні фреймворки	React, Angular, Vue.js, Node.js	Laravel, Symfony, CodeIgniter
Сумісність	Повністю інтегрується з HTML і CSS.	Працює з HTML і підтримує більшість серверів (Apache, Nginx).

Масштабованість	Висока, особливо з використанням Node.js для серверної частини.	Висока, часто використовується для великих проєктів, таких як WordPress або Facebook.
Підтримка спільноти	Велика, активна спільнота розробників, регулярні оновлення стандартів.	Тривала історія використання, велика кількість ресурсів і бібліотек.
Складність вивчення	Відносно проста для початку, особливо для тих, хто знайомий з HTML/CSS.	Легко опанувати, але може бути складнішою для початківців через необхідність роботи з бекендом.

JavaScript є оптимальним вибором для сучасних веб-додатків завдяки своїй універсальності, високій швидкості виконання та можливості працювати як на стороні клієнта, так і на сервері за допомогою Node.js. Ця мова надає розробникам потужні інструменти для створення динамічних та інтерактивних інтерфейсів, що значно покращує досвід користувача. Ключові переваги JavaScript включають велику кількість фреймворків і бібліотек, які спрощують і пришвидшують розробку. Такі інструменти, як React або Angular, дозволяють легко створювати масштабовані й компонентні системи, а Node.js забезпечує високоефективну серверну обробку.

Таким чином, JavaScript - це багатфункціональна мова, яка забезпечує швидку, гнучку і масштабовану розробку сучасних веб-додатків, що робить її найкращим вибором для проєктів, які вимагають інтерактивності та продуктивності.

2.2 Вибір середовища розробки

Існує багато програмних інструментів, які не є надзвичайно популярними, але є надзвичайно корисними для програмування. Деякі з них є багатофункціональними та підтримують кілька мов програмування, що надає їм універсальність у світі технологій. Вивчення таких інструментів може значно спростити процес розробки коду та підвищити ефективність програміста. Одними з таких інструментів є Visual Studio Code (VSCode) та Sublime Text. Обидва є потужними середовищами для розробки програмного забезпечення, зокрема, для написання та редагування коду.

VSCode та Sublime Text є сучасними і популярними середовищами розробки (IDE), що мають багато спільних функцій, таких як підтримка різних мов програмування, плагінів для розширення функціональності, а також зручні інтерфейси для навігації та редагування коду. Однак, вибір між ними залежить від конкретних вимог та переваг розробника. VSCode вважається більш універсальним і потужним середовищем завдяки широкій підтримці плагінів, інтеграції з системами контролю версій, такими як Git, а також наявності вбудованої підтримки відладчиків та інтеграції з іншими інструментами розробки. Водночас Sublime Text є більш легким та швидким у використанні, з інтуїтивно зрозумілим інтерфейсом, який дозволяє швидко переміщатися по великому обсягу коду.

У цьому розділі буде розглянуто сильні та слабкі сторони обох середовищ, щоб визначити, який з них є кращим для різних типів задач і цілей програмування.

Код Visual Studio, який використовують розробники, - це потужний, легкий редактор вихідного коду, який можна використовувати на будь-якій настільній платформі. Він може працювати на Windows, Mac та Linux. VSCode створено з підтримкою різних мов, включаючи JavaScript, TypeScript, а також Node.js. Якщо вам цікаво, як він працює у порівнянні з Visual Studio, прочитайте про Visual Studio vs Visual Studio Code [18].

Інтерфейс середовища зображено на рисунку 2.1.

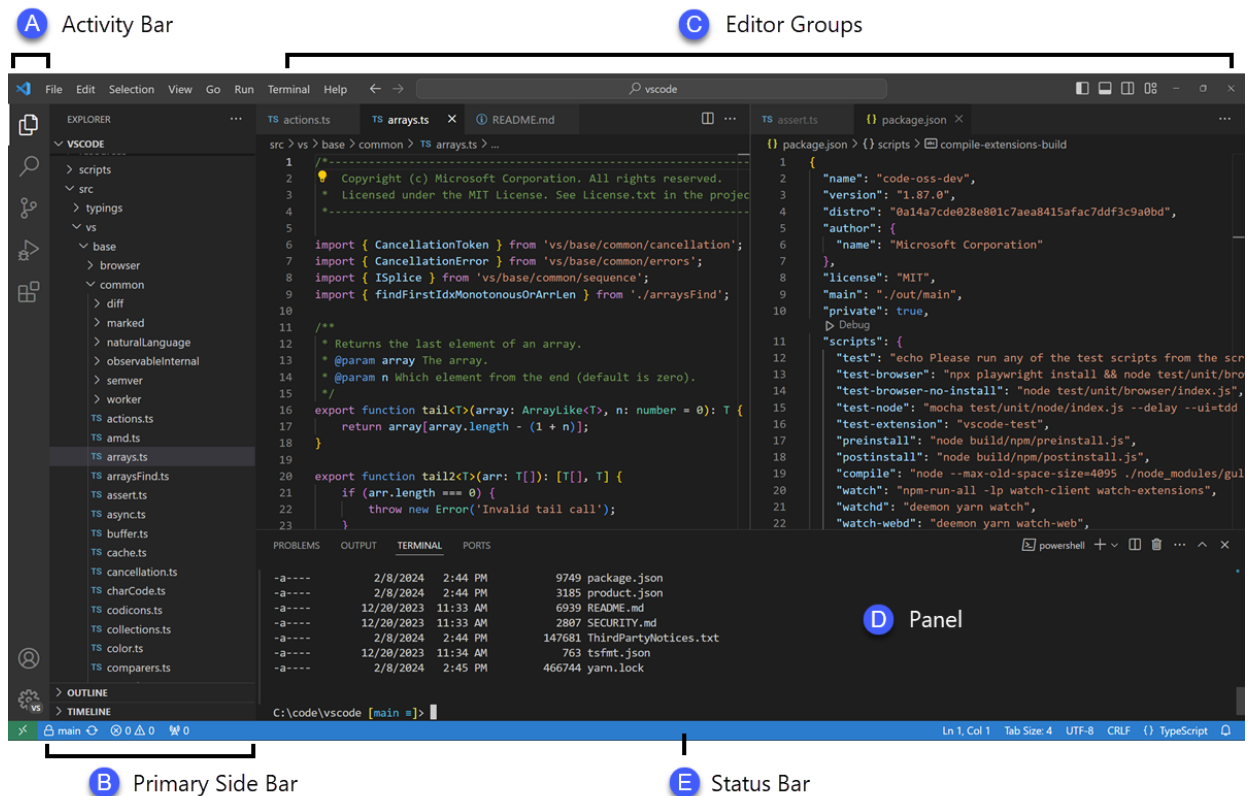


Рисунок 2.1 – Інтерфейс середовища розробки Visual Studio Code

Він також має багату екосистему розширень для інших популярних мов програмування, таких як C#, C++ та PHP. Основним компонентом VSCode є оболонка Electron, яка доповнюється такими мовами, як TypeScript, Node.js та Language Server Protocol, з часто оновлюваними розширеннями.

Підтримка, яку пропонує VSCode, залежить від мови програмування та її розширень. Наприклад, він підтримує віддалену налагодження в деяких мовах. Користувачі також можуть додати підтримку своєї улюбленої мови, використовуючи колоризатори TextMate.

На додаток до цього, завдяки інструменту завершення коду IntelliSense, VSCode надає безліч потужних інструментів для створення та редагування складного програмного забезпечення. Деякі з них включають графічну налагодження, багатокурсорне редагування, навігацію по коду, рефакторинг і підтримку Git [19]

Перевагами середовища розробки VSCode є:

- Великий маркетплейс розширень: VS Code може похвалитися величезним маркетплейсом, наповненим розширеннями практично для будь-якої функціональності, яку ви можете собі уявити, від налагодження до інтеграції з Git'ом та підтримки конкретних мов.
- Вбудована інтеграція з Git'ом: Керуйте версіями коду безпосередньо у VS Code.
- Налагодження з легкістю: VS Code надає потужний налагоджувач для проходження коду та виявлення помилок.

Недоліками середовища розробки VSCode є:

- Вимогливий до ресурсів: Порівняно з Sublime Text, VS Code може бути більш вимогливим до ресурсів, що може вплинути на продуктивність на старих машинах.
- Повільніший запуск: Незважаючи на те, що VS Code все ще швидкий, час запуску може бути трохи довшим, ніж у Sublime Text.
- Може бути непосильним: Величезна кількість функцій та можливостей налаштування може бути складною для початківців.

Завдяки своїм потужним і гнучким можливостям Sublime Text є чудовим вибором для швидкого і простого програмування коду. Він дозволяє легко перемикатися між різними вікнами для виконання різних завдань, таких як налагодження та перевірка коду. Інтерфейс середовища зображено на рисунку 2.2.

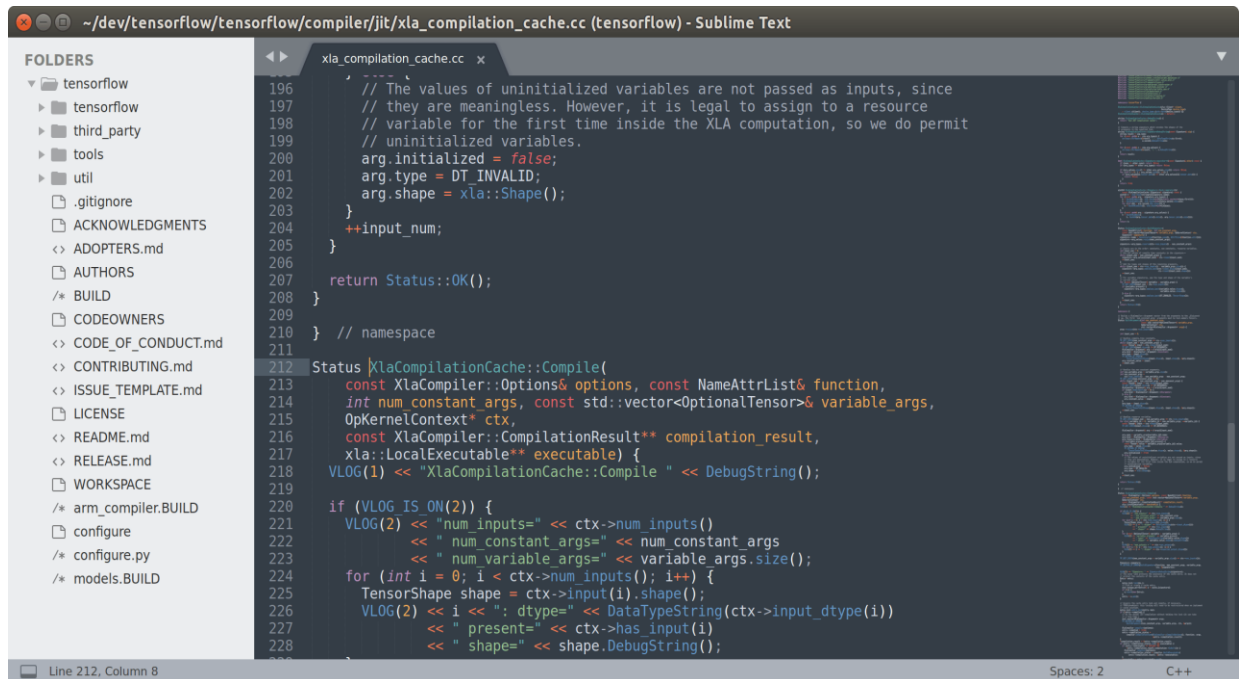


Рисунок 2.2 – Інтерфейс середовища розробки «Sublime Text»

Окрім підтримки понад 70 типів файлів, таких як HTML, JavaScript та CSS, Sublime Text також має низку інших переваг. До них відносяться кілька варіантів вибору, уніфікована палітра команд, розділені вікна, потужна підтримка Git'a і різні вікна, які можна використовувати з усіма вашими моніторами. Він також постачається з вбудованим API плагінів, який можна використовувати для створення простих JSON-файлів. Цікаво, що для програмістів, які звикли до інших редакторів, Sublime Text також підтримує пакети Vi/Vim та TextMate.

У Sublime можна налаштувати все, що стосується редактора, додавши свою колірну схему, встановивши шрифт тексту і вибравши глобальні прив'язки клавіш. Він також має безліч інших можливостей, таких як налаштування файлів JSON і файлів налаштувань XML.

- Швидкість: Sublime Text можна виділити через майже миттєвий запуск і відкриття файлів, дозволяючи розробнику зосередитися на кодуванні.
- Велика кількість налаштувань: Майже кожен аспект редактора на свій можа налаштувати на свій смак, від комбінацій клавіш до підсвічування синтаксису.

- Потужний пошук: Нечіткий пошук Sublime Text дозволяє легко знаходити файли і символи в проектах.
Однак Sublime Text має кілька недоліків:
- Обмежені вбудовані функції: Попри свою потужність, Sublime Text не має таких функцій, як налагодження або вбудована інтеграція з Git'ом, що вимагає плагінів для цих функцій.
- Менша бібліотека розширень: У порівнянні з величезним ринком VS Code, вибір плагінів Sublime Text скромніший.
- Одноразова покупка: Sublime Text вимагає одноразової покупки ліцензії, що декому може здатися менш привабливим, ніж безкоштовна модель VS Code.

Детальна порівняльна характеристика зображена на таблиці 2.2

Таблиця 2.2 – Порівняння редакторів

Критерій	VSCode	Sublime Text
Кросплатформеність	Так, підтримує Windows, macOS, Linux	Так, підтримує Windows, macOS, Linux
Інтерфейс	Інтуїтивно зрозумілий, з можливістю налаштування	Простий, але менш налаштовуваний
Підтримка мов програмування	Широка підтримка різних мов через розширення	Підтримка через плагіни, але менш універсальна
Розширення та плагіни	Має велику кількість безкоштовних розширень	Має обмежену кількість плагінів

Інтеграція з Git	Вбудована підтримка Git	Потрібно встановлювати плагіни для Git
Дебагінг	Вбудований дебагер з інтеграцією	Потрібно налаштувати плагіни для дебагінгу
Підтримка інтеграції з іншими інструментами	Висока: Docker, Kubernetes, Azure, і т.д.	Обмежена підтримка інтеграції
Швидкість роботи	Порівняно швидкий, але може споживати більше ресурсів	Дуже швидкий і легкий, менш вимогливий до ресурсів
Безкоштовність	Повністю безкоштовний	Безкоштовна версія з обмеженнями, повна версія платна
Підтримка відладчика	Вбудовані можливості відладки для багатьох мов	Потрібно налаштовувати вручну
Підтримка тем та кастомізація	Великий вибір тем, можливість глибокої кастомізації	Обмежена кастомізація інтерфейсу
Зручність налаштувань	Просте налаштування через вбудовані параметри та розширення	Не так просто налаштовується для складних сценаріїв

VSCode має значні переваги перед Sublime Text завдяки широкій підтримці плагінів, інтеграції з іншими інструментами, вбудованим можливостям дебагінгу та відладки, а також безкоштовності та регулярним

оновленням. Він є більш функціональним і універсальним середовищем для розробки, що робить його кращим вибором для більшості розробників.

2.3 Вибір бібліотеки для розробки програмного забезпечення

У постійно мінливому ландшафті веб-розробки вибір правильних інструментів і технологій має вирішальне значення для створення надійних і масштабованих додатків. Що стосується фреймворків та платформ на основі JavaScript, то Node, React, Angular є одними з найпопулярніших варіантів. Кожна з цих технологій має свої унікальні сильні сторони та варіанти використання, тому важливо розуміти їхні відмінності та можливості.

На рисунку 2.3 зображено запит цих мов у пошуковій системі Google протягом одного року [20].

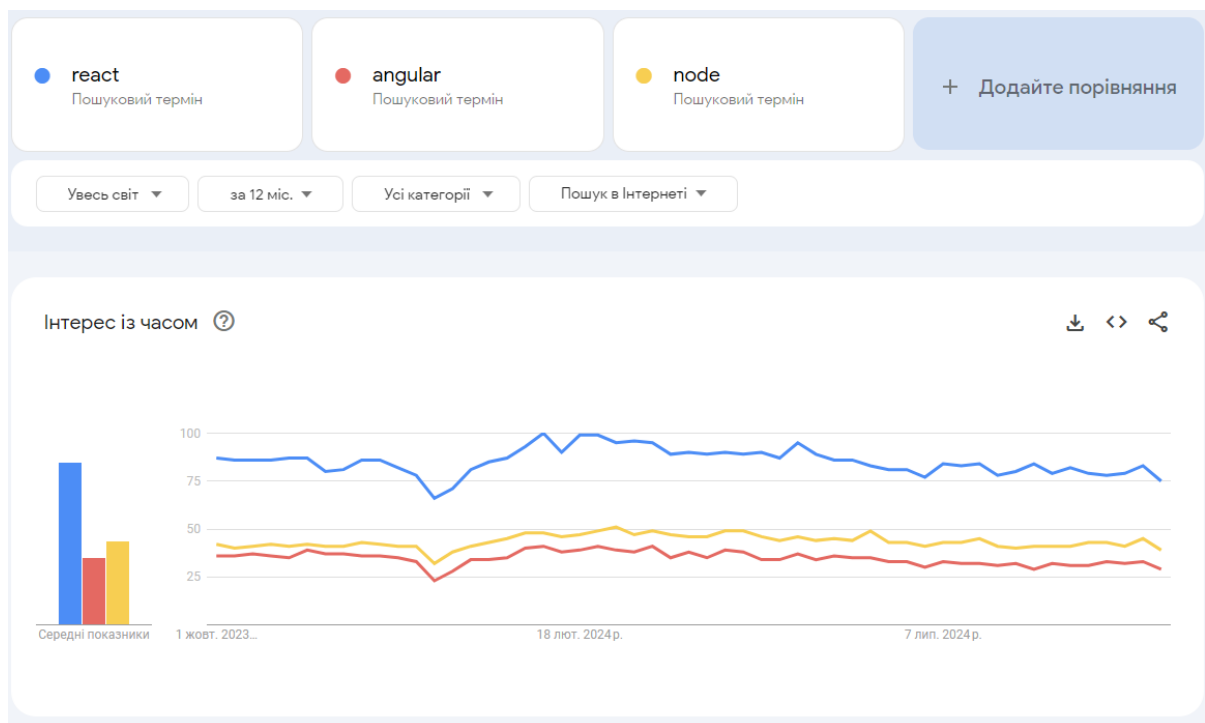


Рисунок 2.3 – Запит найпопулярніших бібліотек

Спостерігаючи за рисунком 2.3 можна зробити висновок що React має значний відрив від інших бібліотек як Angular та Node.

React.js — це популярна JavaScript-бібліотека, створена Facebook для розробки інтерфейсів користувача. Її основна мета — спрощення процесу створення складних і динамічних веб-додатків, забезпечуючи ефективну роботу з елементами інтерфейсу. React заснований на компонентному підході, що дозволяє розробникам розбивати інтерфейс на незалежні, багаторазово використовувані компоненти, кожен з яких може мати свій власний стан. Це робить код більш структурованим, легким у підтримці та тестуванні [21].

Однією з ключових особливостей React є використання віртуального DOM (Document Object Model), який забезпечує ефективну роботу з оновленнями інтерфейсу. Коли стан компонента змінюється, React не оновлює весь DOM, як це роблять традиційні підходи. Натомість він створює віртуальне представлення DOM, порівнює його з попередньою версією та застосовує тільки ті зміни, які дійсно потрібні. Це значно підвищує продуктивність, особливо в масштабних веб-додатках.

React - це один з популярних веб-фреймворків, який набув більшої популярності, ніж інші фреймворки, такі як Angular, Vue тощо. Це пов'язано з тим, що він реалізує Virtual DOM, основною метою якого є підвищення загальної продуктивності додатку. Однак є певні речі, про які слід пам'ятати перед розробкою додатків. Якщо не передбачити проблеми, які можуть виникнути в ієрархії компонентів, це призведе до погіршення продуктивності. Деякі з найпоширеніших проблем - це повторний рендеринг компонентів, відставання програми через фонові обчислення, що виконуються, відставання через обробку великих наборів даних за один проміжок часу [22]

Ще однією важливою особливістю є JSX (JavaScript XML) — синтаксичне розширення JavaScript, яке дозволяє писати HTML-подібний код безпосередньо в JavaScript. JSX робить код більш зрозумілим і зручним для роботи з інтерфейсом, одночасно надаючи всі переваги JavaScript для логіки та обробки даних.

React.js активно використовується для створення односторінкових додатків (SPA), але також підходить для серверного рендерингу завдяки фреймворкам, таким як Next.js. Це дозволяє підвищити швидкість завантаження сторінок і покращити SEO-оптимізацію.

Завдяки своїй гнучкості, високій продуктивності та активній спільноті, React.js став одним із найпопулярніших інструментів у веб-розробці, забезпечуючи ефективний і сучасний підхід до створення користувацьких інтерфейсів.

Angular – один з найкращих фреймворків, розроблених Google. Він був випущений у 2010 році, що робить його найстарішим фреймворком у порівнянні з Angular та React. Це популярний фреймворк на основі TypeScript з ліцензією MIT, а наявність ліцензії MIT означає, що ці продукти мають обмежені обмеження на повторне використання [23].

Angular пропонує широкий спектр функцій, що підтримують відмінну продуктивність додатків. Зокрема, його двостороннє зв'язування означає, що всі зміни в даних негайно відображаються у поданнях, що робить розробку більш орієнтованою на продуктивність. Крім того, рендеринг ivu в Angular забезпечує дуже ефективний рендеринг. Ця особливість також призводить до кращої продуктивності Angular-додатків [24].

Однією з найбільших переваг Angular є його зручний інтерфейс командного рядка (CLI). Він пропонує різні інструменти для створення конфігурацій для розгортання додатків у виробництві та запуску різних завдань, що дозволяє розробникам мінімізувати розмір кінцевого пакета, підвищуючи загальну продуктивність додатків. Нарешті, код Angular стає більш ефективним з кожною наступною версією, що також підвищує продуктивність програмного забезпечення, роблячи Angular одним з найбільш високопродуктивних фреймворків для розробки інтерфейсів.

Компонентна архітектура - одна з найцінніших переваг Angular. Кожна частина інтерфейсу додатку та його функції утворюють окремий компонент. Ці компоненти є самодостатніми і можуть бути налаштовані, змінені та

розроблені окремо. В результаті код стає більш структурованим, а кастомізація коду окремих компонентів не впливає на весь додаток.

Різні частини рішення можна налаштовувати та оновлювати незалежно. Звичайно, API та структури різних компонентів можуть відрізнятися і підключатися один до одного по-різному. Однак, загалом, не існує жодних проблем з виправленням цих компонентів. Крім того, ця функція полегшує повторне використання коду, оскільки один і той самий компонент можна використовувати в різних частинах програми багато разів

Деякі розробники вважають Angular досить багатослівним фреймворком. Всі компоненти є незалежними і можуть управлятися окремо, але керувати ними непросто. Для одного компонента в Angular може знадобитися до п'яти файлів. Розробник повинен створити залежності та оголосити інтерфейси життєвого циклу компонента. З усіма цими необхідними речами код Angular-компонентів іноді може бути занадто довгим і складним.

На щастя, кожен новий реліз Angular спрямований на вирішення цієї проблеми. Хоча код Angular все ще, як правило, далекий від простоти, у нових релізах впроваджуються методи, які роблять його більш стислим та зрозумілим. Крім того, ця особливість Angular може значно окупитися в довгостроковій перспективі, оскільки цей фреймворк дозволяє створити директиву, яка додасть компоненту поведінку, зменшуючи обсяг роботи і завдань з підтримки коду в майбутньому. Крім того, Angular CLI забезпечує автоматизацію, що дозволяє скоротити навантаження на шаблонний код Angular на 70%

Angular відомий як ідеальне рішення для створення динамічних односторінкових додатків. Це додатки, яким не потрібно завантажувати нові сторінки, щоб показати будь-яку інформацію. Замість цього вони представляють безліч динамічних даних і уявлень на одній сторінці. Angular пропонує широкий спектр інструментів, конфігурацій і шаблонів для розробки

односторінкових додатків, здатних обробляти значні обсяги даних з відмінною ефективністю.

NodeJS, середовище виконання, що працює на основі JavaScript-двигка Chrome V8, вивело серверний JavaScript у мейнстрім. Це революційна технологія для додатків у режимі реального часу, яка дозволяє розробникам створювати серверну логіку за допомогою JavaScript. Архітектура NodeJS, керована подіями, забезпечує високу масштабованість і швидкість реагування, що робить його ідеальним вибором для таких додатків, як чати та онлайн-ігри [25].

Node.js також підтримує єдину мову як для фронтенд, так і для бекенду, що спрощує процес розробки та зменшує кількість необхідного коду. Його модульність та легкість використання роблять його відмінним вибором для створення мікросервісів і RESTful API, а величезна екосистема пакетів npm дає розробникам доступ до понад 50 тисяч бібліотек для розширення функціональності.

Node.js використовується багатьма компаніями для створення масштабованих корпоративних застосунків, включаючи Netflix, Walmart, LinkedIn, і Trello. Завдяки своїй гнучкості та продуктивності, він став одним із найбільш популярних середовищ для веб-розробки

Архітектура Node.js базується на двох ключових принципах: неблокування операцій вводу/виводу та асинхронна парадигма [16].

Перший принцип, неблокування операцій вводу/виводу, показує, як Node.js використовує ці операції для полегшення безперебійної обробки клієнтських запитів. Ця методологія дозволяє одночасне виконання операцій вводу/виводу без перешкод для основного потоку. Перевага такого підходу подвійна: він помітно підвищує ефективність, забезпечуючи при цьому вищий ступінь реагування.

Другий принцип, асинхронна парадигма, ілюструє прихильність Node.js до асинхронної моделі програмування. Це легко узгоджується з подієво-керованою сутністю JavaScript. Завдяки розумному використанню зворотних

викликів та подієво-керованого програмування, Node.js оптимізує використання ресурсів та обходить потенційні вузькі місця, притаманні синхронним операціям. Ці два принципи в поєднанні з подієвою моделлю JavaScript підкреслюють важливість розуміння фундаментальних компонентів архітектури додатків Node.js.

Детальна порівняльна характеристика фреймворків зображена на таблиці 2.3.

Таблиця 2.3 – Порівняння фреймворків

Критерій	Node.js	React	Angular
Призначення	Серверна платформа для виконання JavaScript-коду	Бібліотека для побудови інтерфейсів користувача	Фреймворк для розробки односторінкових додатків (SPA)
Тип	Серверний технологічний стек	Бібліотека UI	Повний фронтенд-фреймворк
Продуктивність	Висока для серверної обробки	Висока завдяки віртуальному DOM та компонентному підходу	Висока, але з деякими втратами через двостороннє зв'язування
Легкість у використанні	Потрібно знання серверної розробки та асинхронного програмування	Легкий у освоєнні, підходить для початківців	Складніший, має круту криву навчання
Гнучкість	Висока, дозволяє використовувати	Дуже висока, можна інтегрувати з	Менш гнучкий через свою

	різні бібліотеки та інструменти	іншими технологіями	сувору архітектуру
Інтерактивність UI	Потрібні додаткові інструменти для роботи з UI	Висока, завдяки компонентному підходу та віртуальному DOM	Висока, але потребує більше налаштувань для складних UI
Підтримка компонентів	Не передбачена (не для побудови UI)	Підтримка компонентів на найвищому рівні	Підтримка компонентів, але менш гнучка, ніж у React
Швидкість розробки	Швидка розробка серверної частини	Дуже швидка завдяки компонуванню та повторному використанню коду	Може бути повільнішою через складність фреймворку
Масштабованість	Висока, підходить для великих і малих проектів	Висока, компонентний підхід дозволяє легко масштабувати додатки	Висока, але може потребувати додаткової конфігурації для великих проектів
Суспільна підтримка	Велика, є багато бібліотек та інструментів	Найбільша з усіх трох, велика спільнота розробників та	Велика

		багато ресурсів для навчання	
Шаблони та інструменти	Велика кількість шаблонів для серверних додатків	Відмінні бібліотеки для UI, але без шаблонів для серверної частини	Має вбудовані шаблони для всього стеку
Складність навчання	Середня, потрібні знання асинхронного програмування	Низька, швидко освоюється для розробки інтерфейсів	Висока, багато концепцій для освоєння
Наявність тестування	Є плагіни та інструменти для тестування серверної частини	Легке тестування компонентів через JSX та інструменти, такі як Jest	Має вбудовані інструменти для тестування

У висновку обрано React з кількох ключових причин, які роблять його ідеальним інструментом для розробки сучасних веб-додатків. По-перше, React.js забезпечує ефективну розробку інтерфейсів завдяки використанню віртуального DOM. Цей підхід мінімізує кількість оновлень у реальному DOM, що значно підвищує продуктивність додатків, особливо при роботі зі складними інтерфейсами.

Додатковою перевагою є компонентний підхід, який дозволяє створювати багаторазово використовувані, незалежні компоненти з власним станом. Це спрощує тестування та підтримку коду, роблячи його більш зрозумілим і структурованим. Однонаправлений потік даних, який забезпечує передбачувану поведінку додатку, також сприяє кращій керованості, особливо

у великих проєктах. React підтримує JSX, що дозволяє інтегрувати HTML-подібний синтаксис у JavaScript, спрощуючи розробку інтерфейсу. Завдяки таким фреймворкам, як Next.js, React також може використовуватися для серверного рендерингу, що покращує SEO та прискорює завантаження сторінок.

React має значні переваги перед Node.js та Angular в контексті розробки сучасних веб-додатків завдяки своїй простоті у використанні, гнучкості, потужному компонентному підходу та високій продуктивності. React дозволяє швидко створювати інтерактивні інтерфейси з мінімальними зусиллями, що робить його ідеальним вибором для фронтенд-розробки. На відміну від Angular, React має менш круту криву навчання і надає більшу свободу в інтеграції з іншими технологіями. Node.js є корисним для серверної частини, але для побудови інтерфейсів він не є достатньо потужним без додаткових інструментів.

2.4 Висновки до другого розділу

У другому розділі було проведено детальний аналіз різних мов програмування та інструментів для розробки програмного забезпечення. Зокрема, серед мов програмування розглянуто JavaScript та PHP, кожна з яких має свої особливості та переваги. JavaScript є багатофункціональною мовою, яка забезпечує швидку та гнучку розробку сучасних веб-додатків. Вона широко використовується для створення інтерактивних інтерфейсів і динамічних веб-сторінок завдяки своїй здатності працювати безпосередньо в браузері, що робить її оптимальним вибором для проєктів, які вимагають високої продуктивності та інтерактивності.

Також було проаналізовано середовища розробки, серед яких виділяються Visual Studio Code (VSCode) та Sublime Text. VSCode є найбільш популярним і універсальним середовищем для розробки, яке підтримує безліч

розширень та інтеграцій, що значно покращують процес кодування. Його кросплатформеність, підтримка багатьох мов програмування та потужні інструменти для дебагінгу роблять його оптимальним вибором для сучасних розробників. Sublime Text є більш легким та швидким у використанні, що також робить його привабливим варіантом для деяких розробників, але він має менше функціональних можливостей у порівнянні з VSCode.

Щодо фреймворків для розробки веб-додатків, було розглянуто такі варіанти, як React, Angular та Node.js. З усіх запропонованих фреймворків, React був обраний як найбільш підходящий інструмент для створення сучасних веб-додатків. Однією з основних переваг React є використання віртуального DOM, що дозволяє значно знизити кількість оновлень реального DOM і підвищити продуктивність при роботі зі складними інтерфейсами. Крім того, React підтримує компонентний підхід до розробки, що дозволяє створювати багаторазово використовувані компоненти з власним станом. Це робить код більш структурованим, легким для тестування та підтримки.

З урахуванням всіх цих факторів, у цьому розділі було вибрано мову програмування JavaScript, середовище розробки VSCode та бібліотеку ReactJS як найбільш ефективні інструменти для розробки сучасних веб-додатків.

3 ПЛАН ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для успішної розробки програмного забезпечення, а саме емуляторів фізичних процесів для лабораторних робіт з фізики, потрібен детально описаний план. Програмне забезпечення, або простими словами веб-додаток має містити чотири сторінки:

- Головна сторінка;
- Теоретичні відомості;
- Тестування;
- Практична частина.

Кожна з цих сторінок має бути максимально зрозуміла, без навантажень, без зайвих деталей, для того щоб не відволікати студента від виконання лабораторної роботи.

Сторінка «Головна» має містити в собі блоки з метою роботи, яка складається з кількох етапів та блоку обладнання і матеріали, які необхідні для успішного виконання лабораторної роботи.

Сторінка «Теоретичні відомості» має містити в собі необхідну інформацію для ознайомлення з лабораторною роботою, теорія описана чітко та стисло, описані формули та зображення за наявності.

Сторінка «Тестування» має містити в собі тестування, яке складається з вибірки питань, але на самий тест вибирається лише 10 питань. Питання та варіанти до них генеруються на кожен тест рандомно, для того щоб ускладнити проходження тестування. За кожен правильну відповідь, студент отримує 10 балів, всього 100 балів за тест. Тест має обмежений час на проходження, а саме 7 хвилин, після закінчення часу, тест закінчується. Також має бути реалізована система спроб, якщо студент складає тест більше чим за 1 спробу, за наступні спроби знімається бал, тобто якщо спроб 6, то знімається 5 балів. Мінімальний поріг складання тесту 60 балів, у випадку якщо студент набрав менше балів, рекомендується більш детально ознайомитись з

теоретичними відомостями та спробувати пройти тест ще раз. Якщо набрано мінімальний поріг – студент може перейти до практичної частини.

Сторінка «Практична частина» має містити в собі емулятор фізичного процесу, який інформативно та стисло демонструє роботу реального апарату із фізичних лабораторій. Наявність інструкції, або ходу роботи обов’язкова. В інструкції описано план дій, який має покроково виконати студент.

Використання інтернет-симуляцій студентами для виконання домашніх завдань з фізики сприяє отриманню бажаного рівня знань. Одним з найбільш перспективних напрямів використання у фізичній освіті є комп’ютерне моделювання фізичних явищ і процесів.

Для успішної розробки програмного забезпечення, скористаємось планом. Принцип розробки для лабораторних робіт схожий, змінюється лише необхідна інформація та емулятори.

3.1 Розробка головної сторінки

В першу чергу розробимо головну сторінку, яка містить в собі назву лабораторної роботи, блоки з метою роботи та обладнання і матеріали.

Сторінку реалізовано через функцію `HomePage()` та функцію `App()`, перша функція містить в собі блоки з назвою роботи, кнопкою для увімкнення англійської/української мови, а також інформаційний блок. Друга функція містить в собі `React-Router` який дозволяє створити та об’єднати між собою сторінки. Кожна сторінка має `Link`, і перенесена в кнопки для переходу.

Структура сторінки має вигляд:

```
return (
```

```
<div className="home-page">
```

```
<div className="info-container">
```

```
<div className="info-block">
```

```
<h2><b>{isEnglish ? 'Purpose of the Work' : 'Мета роботи'}</b></h2>
```

```

<div className="buttons">
  <Link to="/theory">
    <button className="home-page-button">{isEnglish ? 'Theoretical
information' : 'Теоретичні відомості'}</button>
  </Link>
  <Link to="/testing">
    <button className="home-page-button">{isEnglish ? 'Testing' :
'Тестування'}</button>
  </Link>
  <Link to="/practice">
    <button className="home-page-button">{isEnglish ? 'Practical Part' :
'Практична Частина'}</button>
  </Link>
</div>
</div>
);
}

```

Результат розробки головної сторінки для лабораторних робіт зображено на рисунку 3.1 та 3.2.

ВИВЧЕННЯ АПАРАТІВ І МЕТОДІВ ВЧ- ТА УВЧ-ТЕРАПІЇ

Українська ☐ English

Мета роботи

1. Ознайомитися з будовою, принципом дії та методикою експлуатації апарата УВЧ-терапії.
2. Дослідити дію електричного і магнітного полів УВЧ та середовища(моделі), що імітують електропровідні і діелектричні властивості біологічних тканин.

Обладнання і матеріали

1. Апарат УВЧ-30.
2. Аплікатор вихрових струмів ЕВС-1 та індикатор (люмінесцентна або неоновна лампа).
3. Пристрій для вивчення дії електричного і магнітного полів УВЧ на середовища (електроліт, касторове масло або гліцерин).
4. Секундомір.

Теоретичні відомості
 Тестування
 Практична Частина

Рисунок 3.1 – Вигляд головної сторінки для лабораторної роботи №21 «Вивчення апаратів і методів УВЧ-терапії електричним і магнітним полями та ВЧ-струмом»

ВИВЧЕННЯ ГЕЛІЙ-НЕОНОВОГО (He - Ne) ЛАЗЕРА

Українська ☒ English

Мета роботи

1. Вивчити будову і принцип дії оптичного квантового генератора - газового лазера
2. Навчитися користуватися газовим лазером, визначити довжину хвилі та енергію кванта лазерного випромінювання за допомогою дифракційної решітки

Обладнання і матеріали

1. Газовий гелій-неоновий лазер типу ЛГ-209 з блоком живлення
2. Дифракційна решітка
3. Оптична лава з вимірювальною лінійкою
4. Екран з міліметровою шкалою

Теоретичні відомості
 Тестування
 Практична Частина

Рисунок 3.2 – Вигляд головної сторінки для лабораторної роботи №26 «Вивчення гелій-неонових лазера»

3.2 Розробка сторінки «Теоретичні відомості»

Сторінку реалізовано через функцію TheoreticalPart() який містить в собі блоки з інформацією, зображеннями та формулами. Основну увагу приділено коректному відображенню зображень та формул. Також сторінка містить кнопку «Перейти до тестування», яка переходить до сторінки «Тестування». Результати розробки зображено на рисунках 3.3 та 3.4.

Структура сторінки має вигляд:

```
function TheoreticalPart() {
  const { isEnglish } = useContext(LanguageContext);
  return (
    <div className="theoretical-part">
      <p>Інформація</p>
      <div className="buttontheory-container">
        <Link to="/testing">
          <button className="theory-button">
            {isEnglish ? 'Go to testing' : 'Перейти до тестування'}
          </button>
        </Link>
      </div>
    </div>
  );
};
export default TheoreticalPart;
```

ВИВЧЕННЯ АПАРАТІВ І МЕТОДІВ ВЧ- ТА УВЧ-ТЕРАПІЇ

Українська English

Теоретичні відомості. Лабораторна робота: "Вивчення апаратів і методів УВЧ-терапії електричним і магнітним полями та вч-струмом.

В лабораторній роботі використовується апарат УВЧ-30, спрощена блок-схема якого зображена на рисунку 1.1. Апарат складається з блоку живлення (ДЖ), перешкодоушугуючого пристрою (ПП), лампового генератора (ЛГ), терапевтичного контура (ТК), і пристосувань, які називаються електродами пацієнта (ЕП) та аплікатори вихрових струмів (ЕВС-1). Два останні використовуються для здійснення локального терапевтичного впливу на тканини організму пацієнта відповідно електричним і магнітним полям ультрависокої частоти (скорочено УВЧ).

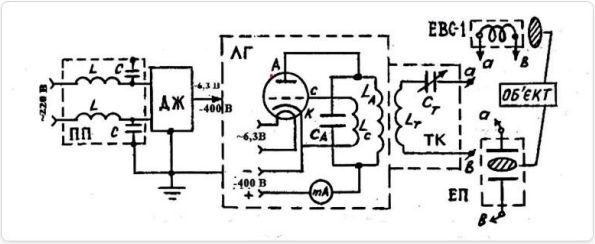


Рисунок 1.1 - Спрощена схема апарату УВЧ-30

Призначення блоків і елементів апарату УВЧ-терапії

Рисунок 3.3 –Вигляд сторінки «Теоретичні відомості» для лабораторної роботи №21 «Вивчення апаратів і методів УВЧ-терапії електричним і магнітним полями та ВЧ-струмом»

ВИВЧЕННЯ ГЕЛІЙ-НЕОНОВОГО (He - Ne) ЛАЗЕРА

Українська ☐ English

Теоретичні відомості. Лабораторна робота "Вивчення гелій-неонового лазера.

В роботі використовуються квантові системи (атоми, молекули, іони) які можуть знаходитись в різних енергетичних станах, які прийнято характеризувати енергетичними рівнями. В основному (стаціонарному) стані частинки речовини мають мінімальну енергію, якій відповідає основний енергетичний рівень. В цьому стані квантова система, наприклад атом, може знаходитись тривалий час. При поглинанні енергії ззовні атоми речовини переходять в збуджений стан, тобто на вищий енергетичний рівень. Час перебування атома в збудженому стані обмежений і складає приблизно 10^{-8} секунди. Проте існують збуджені стани з відносно більшим часом "життя" (порядку 10^3 секунди), які називаються метастабільними, а відповідні їм енергетичні рівні - метастабільними рівнями.

Переходи збуджених атомів речовини з вищих енергетичних рівнів на нижчі супроводжуються випромінюванням квантів енергії. Ці переходи можуть бути спонтанними (самодовільними) або індукованими (вимушеними). Випромінювання, яке виникає при зазначених переходах, називається відповідно спонтанним або індукованим. Спонтанне випромінювання зумовлене переходом атомів із одного енергетичного стану в інший під впливом внутрішніх факторів і має випадковий характер

Індуковане або вимушене випромінювання виникає в результаті дії на збуджені атоми фотонів зовнішнього електромагнітного поля або фотонів, які з'явилися в середовищі внаслідок спонтанного випромінювання. Характерно, що фотон індукованого випромінювання співпадає по частоті, фазі і напрямку руху з фотоном, який викликав вимушений перехід збудженого атома на більш низький енергетичний рівень. Отже, електромагнітні хвилі, які виникають при індукованому випромінюванні, являються когерентними.

Рисунок 3.4 – Вигляд сторінки «Теоретичні відомості» для лабораторної роботи №26 «Вивчення гелій-неонового лазера»

3.3 Розробка сторінки «Тестування»

Сторінка містить в собі безліч реалізованих функцій, які коротко розберемо у цьому підрозділі.

Функція `const initialQuestions` має унікальний ідентифікатор запитання – `id`, `question` яка містить назву запитання на двох мовах, `options` – містить в собі `values` - варіанти відповіді та `correctAnswer`, для визначення правильної відповіді.

Функція `const handleAnswer` робить відповідь у блоці запитання активним.

Функція `const handleNextQuestion` робить доступною кнопку «Наступне запитання» та перевіряє по унікальному ідентифікатору відповіді чи правильна вона, якщо так, додає 10 балів до загальної кількості балів.

Функція `const handleTextAgain` має в собі функції рандомізації запитань та відповідей, функцію `timer` який виставляє 420 секунд на тест, а також `const`

newAttempts який відповідає за реалізацію спроб та мінусує 1 бал за кожну спробу після першої.

Результати розробки зображено на рисунках 3.5 та 3.6. Також в перших варіантах тестування була результуюча таблиця, яка містила в собі результати тестування з назвою та номером запитання і чи надана правильна відповідь, але від цього вирішили відмовитись задля ускладнення проходження тестування студентом.

Реалізовано кнопки «Перейти до теоретичних відомостей», «Спробувати ще раз» та «На головну», яка відповідає за свої функції з назв. Кнопки та результати тестування зображені на рисунку 3.7 та 3.8.

ВИВЧЕННЯ АПАРАТІВ І МЕТОДІВ ВЧ- ТА УВЧ-ТЕРАПІЇ

Українська English

Тестування

Залишилось часу: 6:55 хв

Кількість спроб: 1

Яка методика передбачає прогрів тканин ВЧ вихровими струмами?

☐ діатермія

☐ індуктотермія

☒ електрофорез

☐ дарсонвалізація

Наступне запитання

Рисунок 3.5 –Вигляд сторінки «Тестування» для лабораторної роботи №21 «Вивчення апаратів і методів УВЧ-терапії електричним і магнітним полями та ВЧ-струмом»

ВІВЧЕННЯ ГЕЛІЙ-НЕОНОВОГО (He - Ne) ЛАЗЕРА

Українська ☒ English

Тестування

Залишилось часу: 6:36 хв

Кількість спроб: 3

Темні області дифракційної картини відповідають дифракційним...

☐ мінімумам
☐ максимумам
☐ одиницям
☐ нулям

Наступне запитання

Рисунок 3.6 – Вигляд сторінки «Тестування» для лабораторної роботи №26 «Вивчення гелій-неонового лазера»

ВІВЧЕННЯ ГЕЛІЙ-НЕОНОВОГО (He - Ne) ЛАЗЕРА

Українська ☒ English

Тестування

Залишилось часу: 6:16 хв

Кількість спроб: 3

Тест завершено!

Ваш результат: 8 %

Ви набрали малу кількість відповідей. Детальніше ознайомтесь з теоретичними відомостями та спробуйте знову.

[Перейти до теоретичних відомостей](#)
[Спробувати ще раз](#)

[На головну](#)

Рисунок 3.7 – Результат проходження тестування, випадок з незадовільним результатом

ВІВЧЕННЯ ГЕЛІЙ-НЕОНОВОГО (He - Ne) ЛАЗЕРА

Українська

☐

English

Тестування

Залишилось часу: 6:27 хв

Кількість спроб: 4

Тест завершено!

Ваш результат: 77 %

Вітаю, ви набрали достатню кількість балів. Зробіть знімок екрану результатів та переходьте до практичної частини або спробуйте підвищити бал!

Перейти до практичної частини

Спробувати ще раз

На головну

Рисунок 3.8 – Результат проходження тестування, випадок з задовільним результатом

3.4 Розробка сторінки «Практична частина»

Сторінку реалізовано за допомогою компонентів, MyJubmotron який відповідає за блок зміни мови та назву лабораторної роботи, ImageBlock, який дозволяє інтерактивно пересувати дифракційну решітку і при цьому змінюється відстань у полі вводу та на екрані, Screen який відповідає за коректне відображення максимумів і мінімумів на лінійці, та Instruction, який містить в собі поетапний хід виконання роботи з необхідною інформацією та таблицею яку потрібно заповнити по мірі виконання.

На рисунках 3.9 та 3.10 зображено емулятор та інструкцію для лабораторної роботи №21, а на рисунках 3.11 та 3.12 зображено емулятор та інструкцію для лабораторної роботи №26.

ВИВЧЕННЯ АПАРАТІВ І МЕТОДІВ ВЧ- ТА УВЧ-ТЕРАПІЇ

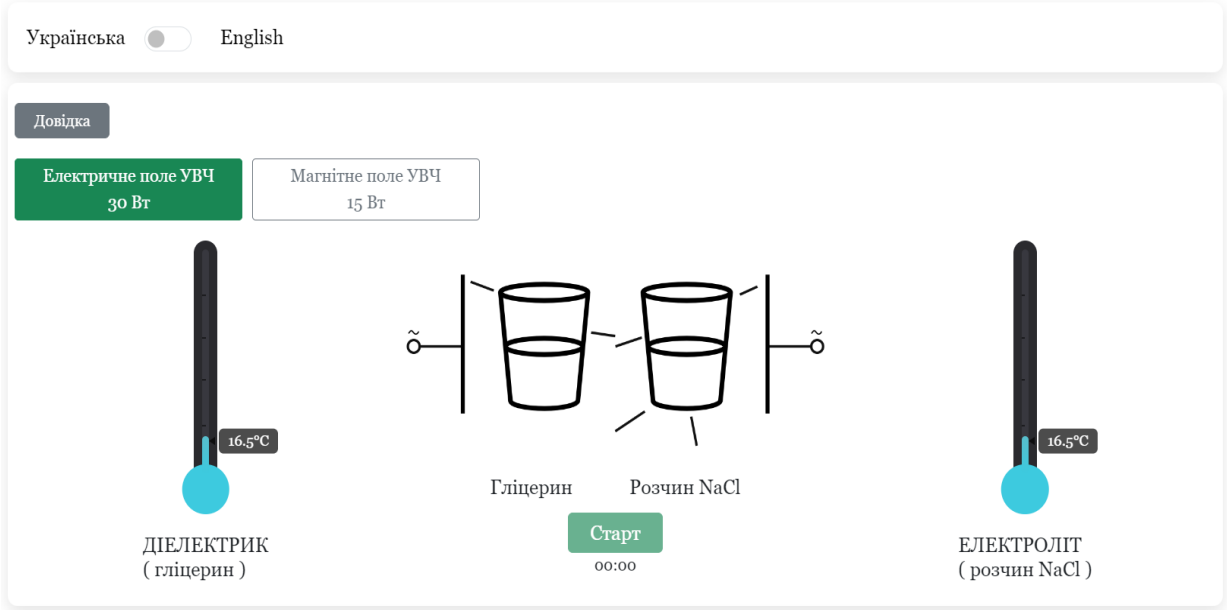


Рисунок 4.9 – Емулятор для лабораторної роботи №21

Інструкція виконання практичного завдання

Ось інструкція для виконання завдання. Дотримуйтесь кожного кроку уважно.

1. Крок 1: Виберіть потрібне поле - Електричне або Магнітне.

2. Крок 2: Натисніть кнопку "Старт".

На вашому екрані запуститься секундомір і термометр

Обробка результатів досліджень.

3. Крок 3. Побудувати графіки залежностей $t_e = f(t)$ та $t_d = f(t)$ які характеризують нагрівання електроліту і діелектрика в електричному і магнітному полях УВЧ.

4. Крок 4. Визначити середню швидкість нагрівання діелектрика і електроліту в електричному і магнітних полях увч за формулою: $v_d = \frac{\Delta t_d}{\Delta t} = \frac{t_{20} - t_0}{t_{20} - t_0}$ та $v_e = \frac{\Delta t_e}{\Delta t} = \frac{t_{20} - t_0}{t_{20} - t_0}$

5. Крок 5. Обчислити кількість теплоти, яка виділяється в одиниці об'єму досліджуваних об'єктів за одиницю часу за формулою: $q = \frac{\Delta Q}{V \Delta t} = \frac{cm \Delta t}{V \Delta t} = c_p \frac{\Delta t}{\Delta t}$

де ρ - густина речовини; c - питома теплоємність

6. Крок 6. Проаналізувати одержані результати досліджень і записати в звіт.

Густина і питома теплоємність досліджуваних речовин		
Речовина	c, Дж/(кг*град)	ρ, кг/м^3
1%, р-н NaCl(електроліт)	4200	1000
Касторове масло(діелектрик)	2100	960
Гліцерин (діелектрик)	2400	1260

Рисунок 3.10 – Інструкція для лабораторної роботи №21

ВИВЧЕННЯ ГЕЛІЙ-НЕОНОВОГО (He - Ne) ЛАЗЕРА

УкраїнськаEnglish

Довідка

Живлення

Старт

Відстань дифракційної решітки від екрану: 9 9(см)

Екран

Для отримання відстані від центрального мінімуму потрібно натиснути на нього починаючи з правої сторони відносно о.

Мінімуми зліва - дзеркально симетричні правим та мають таку ж відстань

Рисунок 3.11 – Емулятор для лабораторної роботи №26

Інструкція виконання практичного завдання

Ось інструкція для виконання завдання. Дотримуйтесь кожного кроку уважно.

1. Крок 1: Натисніть кнопку "Живлення".

2. Крок 2: Натисніть кнопку "Старт".

3. Крок 3: Встановіть відстань дифракційної решітки від екрану. Ви можете зробити це двома способами, ввести значення у полі вводу, або потягнути решітку на потрібну відстань.
На лінійці ви можете спостерігати червоні диски, симетрично розташовані відносно центрального (нульового) максимуму в горизонтальному напрямі.

4. Крок 4. Виміряти віддаль між дифракційною решіткою і екраном.

5. Крок 5: Виміряти віддаль L між центрами максимумів першого порядку та занести в таблицю.

6. Крок 6: Аналогічні виміри провести для інших максимумів та занести в таблицю.

7. Крок 7. Розрахувати довжину хвилі світла λ , яке падає на дифракційну решітку за формулою $\lambda = d \sin \alpha / K$ та вписати результат у колонку. $\lambda =$ нанометрів.

k	R	L	$l_k = L_k/2$	tg a	sin a	λ	$\delta \lambda$	ε	$\delta \varepsilon$
1									
2									
3									
						$\lambda =$	$\delta \lambda =$	$\varepsilon =$	$\delta \varepsilon =$

Рисунок 3.12 –Інструкція для лабораторної роботи №26

3.5 Висновки до третього розділу

За результатами плану, програмне забезпечення було розроблене та успішно функціонує. Програма дозволяє студентам ознайомитись з теоретичними відомостями лабораторних робіт, пройти тестування на основі цих відомостей, та виконати практичну частину як містить емулятор фізичного процесу.

У висновку розробки програмного забезпечення для симуляції фізичних процесів, було продемонстровано симуляцію фізичних процесів, а саме створення магнітного та електричного полів з застосуванням апарату УВЧ-30 [27], а також симуляції гелій-неонового лазера[28]. Створено сторінки «Головна», «Теоретичні відомості», «Тестування» та «Практична частина».

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-технічної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота за темою «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» відноситься до науково-технічних робіт, які плануються для використання безпосередньо самим замовником, тобто її результатами будуть користуватися як замовник, так і інші для замовника користувачі. У цьому випадку нам потрібно довести ефективність інвестицій, вкладених у цей проект самим замовником.

Для цього випадку необхідно виконати такі етапи робіт:

- 1) провести технологічний аудит власної науково-технічної розробки, тобто встановити її науково-технічний рівень;
- 2) розрахувати витрати на здійснення науково-технічної розробки;
- 3) провести розрахунок економічної ефективності науково-технічної розробки у випадку її впровадження замовником на власному підприємстві та обґрунтувати економічну доцільність впровадження замовником розробленого науково-технічного проекту.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» є оцінювання науково-

технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [30].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	4	3	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	38	40	39
Середньоарифметична сума балів $СБ_c$	39,0		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [30].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий

31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» становить 39,0 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розрахуємо за формулою[29]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і

при цьому має виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де I_{ni} та I_{ai} – чисельні значення конкретного i -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю- вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Кількість доступних симуляцій	од.	2	30	15	0,3
Точність відтворення симуляції	%	100	95	0,95	0,2
Швидкість обробки симуляції	с	4...5	2...3	2	0,15

Використання оперативної пам'яті	ГБ	4	2	2	0,25
Доступність інтерфейсу користувача	бал	8	9,5	1,19	0,1

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 15 \cdot 0,3 + 0,95 \cdot 0,2 + 2 \cdot 0,15 + 2 \cdot 0,25 + 1,19 \cdot 0,1 = 5,61.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 5,61 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам,

технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [30]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 20000,00 \cdot 10 / 21 = 9523,81 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Менеджер проекту (координатор)	20000,00	952,38	10	9523,81
Інженер-розробник програмного забезпечення	17500,00	833,33	42	35000,00
Консультант (к.фіз.мат.н., доцент)	15000,00	714,29	5	3571,43
Лаборант	8050,00	383,33	21	8050,00
Всього				56145,24

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_p$) за відповідними найменуваннями робіт НДР на тему «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» розраховуємо за формулою:

$$З_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=8000,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [30].

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 8000,00 \cdot 1,10 \cdot 1,15 / (21 \cdot 8) = 60,24 \text{ грн.}$$

$$З_{p1} = 60,24 \cdot 5,60 = 337,33 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця інженера-розробника програмного забезпечення	5,60	2	1,10	60,24	337,33
Інсталяція програмного забезпечення середовища розробки і моделювання	10,00	3	1,35	73,93	739,29
Компіляція програмних блоків моделювання емулятора фізичних процесів	12,00	4	1,50	82,14	985,71
Підготовка локального серверного обладнання для проведення досліджень	6,00	5	1,70	93,10	558,57
Налагодження програмних блоків збору інформації	4,50	5	1,70	93,10	418,93

Тестування емулятора фізичних процесів	16,00	2	1,10	60,24	963,81
Всього					4003,64

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.7)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{дод}} = (56145,24 + 4003,64) \cdot 10 / 100\% = 6014,89 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (4.8)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (56145,24 + 4003,64 + 6014,89) \cdot 22 / 100\% = 14556,03 \text{ грн.}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і

предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (4.9)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$C_{\text{в}j}$ – вартість відходів j -го найменування, грн/кг.

$$M_1 = 2,0 \cdot 184,00 \cdot 1,05 - 0 \cdot 0 = 386,40 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір SOFT Ultra Plus	184,00	2,0	0	0	386,40
Папір для записів Parers SOFT Light A5	114,00	3,0	0	0	359,10
Органайзер офісний OFFICE SOFT	216,00	3,0	0	0	680,40
Канцелярське приладдя (набір	210,00	2,0	0	0	441,00

офісного працівника)					
Картридж для принтера Canon LBP6500	1865,00	1,0	0	0	1958,25
Диск оптичний NewOptice CD-RW	28,00	5,0	0	0	147,00
Flesh-пам'ять Kingston 16 GB	99,00	2,0	0	0	207,90
Тека для паперів BOSS PAPERS BOX	75,00	6,0	0	0	472,50
Інші матеріали	150,00	1,0	0	0	157,50
Всього					4810,05

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування,

виготовлення, транспортування, монтаж та встановлення. Витрати за статтею відсутні.

4.3.6 Програмне забезпечення для наукових робіт

До статті «Програмне забезпечення для наукових робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{npz} = \sum_{i=1}^k C_{inprz} \cdot C_{npz.i} \cdot K_i, \quad (4.10)$$

де C_{inprz} – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{npz.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань програмних засобів.

$$B_{npz} = 6380,00 \cdot 1 \cdot 1,1 = 7018,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.8 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Visual Studio Community	1	6380,00	7018,00
Програмний пакет імітаційного моделювання	1	7400,00	8140,00
Всього			15158,00

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б}}{T_{в}} \cdot \frac{t_{вик}}{12}, \quad (4.11)$$

де $Ц_{б}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{в}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (42999,00 \cdot 2) / (3 \cdot 12) = 2388,83 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника ПЗ Комп'ютер Lenovo LOQ 17IRB8 / i5-13400F, 32, 1TB SSD, RTX 4060	42999,00	3	2	2388,83

8GB (90VH00E5UL)				
Маршрутизатор TP-LINK ER7212PC	9999,00	3	2	555,50
Робоче місце симулятор Офісний ПК + Монітор 20" Intel i5 4x3.6GHz 8 ГБ SSD Win10 + Клавіатура + Мишка(PC OFFICE)	23000,00	3	2	1277,78
Оргтехніка	9500,00	5	2	316,67
Приміщення лабораторії розробки ПЗ	400000,00	40	2	1666,67
ОС Windows	8320,00	3	2	462,22
Прикладний пакет Microsoft Office	7460,00	3	2	414,44
SSD диск Kingston Enterprise DC600M 480GB 2.5" SATAIII 3D TLC (SEDC600M/480G)	5389,00	3	2	299,39
Проектор BenQ MW560, DLP, WXGA, 4000Lm,	17719,00	5	2	590,63

20000:1, D-sub, HDMI, White (9H.JNF77.1JE)				
Всього				7972,13

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (4.12)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 10,98$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,42 \cdot 160,0 \cdot 10,98 \cdot 0,95 / 0,97 = 737,86 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника ПЗ Комп'ютер Lenovo LOQ 17IRB8 / i5-13400F, 32,	0,42	160,0	737,86

1TB SSD, RTX 4060 8GB (90VH00E5UL)			
Маршрутизатор TP-LINK ER7212PC	0,036	160,0	63,24
Робоче місце симулятор Офісний ПК + Монітор 20" Intel i5 4x3.6GHz 8 ГБ SSD Win10 + Клавіатура + Мишка(PC OFFICE)	0,12	90,0	118,58
Проектор BenQ MW560, DLP, WXGA, 4000Lm, 20000:1, D-sub, HDMI, White (9H.JNF77.1JE)	0,180	45,0	98,82
Оргтехніка	0,56	3,0	18,45
Всього			1606,15

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» відсутні.

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{с}} = (Z_{\text{о}} + Z_{\text{р}}) \cdot \frac{H_{\text{іс}}}{100\%}, \quad (4.13)$$

де $H_{\text{іс}}$ – норма нарахування за статтею «Інші витрати», приймемо $H_{\text{іс}} = 50\%$.

$$I_{\text{с}} = (56145,24 + 4003,64) \cdot 50 / 100\% = 30074,44 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків;

витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_o + З_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.14)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $H_{нзв} = 100\%$.

$$B_{нзв} = (56145,24 + 4003,64) \cdot 100 / 100\% = 60148,88 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_o + З_p + З_{доо} + З_n + M + K_e + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_e + B_{нзв}. \quad (4.15)$$

$$B_{заг} = 56145,24 + 4003,64 + 6014,89 + 14556,03 + 4810,05 + 0,00 + 0,00 + 15158,00 + 7972,13 + 1606,15 + 0,00 + 0,00 + 30074,44 + 60148,88 = 200489,46 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-технічної роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.16)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-технічної роботи, приймемо $\eta = 0,9$.

$$ЗВ = 200489,46 / 0,9 = 222766,06 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки від її впровадження безпосередньо розробником (замовником)

Визначення економічного ефекту від впровадження комп'ютерного емулятора у навчальному закладі може базуватися на ряді показників, що враховують економічні вигоди, пов'язані з його використанням:

1. Аналіз витрат на впровадження

Спочатку потрібно визначити витрати, пов'язані з придбанням, встановленням та технічною підтримкою емулятора, а також навчанням персоналу.

Загальні витрати ЗВ на завершення науково-технічної роботи складають
 $ZB = 222766,06$ грн.

2. Оцінка економії від заміщення матеріально-технічної бази

Емулятор дозволяє замінити частину матеріальних ресурсів (лабораторне обладнання, матеріали для практичних занять тощо), це зменшує витрати на їх придбання та підтримку: 1) зменшення потреби у витратних матеріалах; 2) скорочення амортизаційних витрат на обладнання; 3) зниження ризику псування дорогого обладнання через некоректне використання.

Загальна річна економія від заміщення матеріально-технічної бази складає 144860 грн. на рік. При застосуванні емулятора протягом 5 років загальна економія складе $144860 * 5 = 724300$ грн.

3. Визначення економічного ефекту від підвищення ефективності навчального процесу

Використання емулятора може підвищити якість і швидкість навчання, що, у свою чергу, впливає на продуктивність освітньої діяльності: 1) скорочення часу на вивчення матеріалу завдяки інтерактивним методам; 2) підвищення рівня засвоєння знань і практичних навичок студентів; 3) покращення підготовки випускників, що збільшує їхню конкурентоспроможність на ринку праці, підвищує рейтинг навчального закладу та сприяє зростанню кількості абітурієнтів.

4. Загальна оцінка економічного ефекту

Економічний ефект можна розрахувати як різницю між усіма економічними вигодами та витратами: $E = 724300,00 - 222766,00 = 501540,00$ грн.

Термін окупності вкладень складе : $222766,00 / 144860,00 = 1,54$ року.

4.4 Висновки до четвертого розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ» становить 39,0 бала, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 5,61 рази.

Отже можна зробити висновок про доцільність проведення науково-технічної роботи за темою «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів ВНМУ».

ВИСНОВКИ

У першому розділі було проведено загальну характеристику об'єкту досліджень, ключовими аспектами розробки емуляторів є точність і продуктивність. Для більш складних фізичних явищ необхідно застосовувати ефективні числові методи (методи кінцевих елементів, метод Монте-Карло, тощо) і паралельні обчислення. Також важливим є баланс між спрощенням моделі для прискорення обчислень і збереженням фізичної реалістичності. Емулятори можуть бути використані в наукових дослідженнях, інженерії, медицині, фізиці, а також для розробки та тестування нових технологій.

Розглянуто теоретичні відомості двох лабораторних робіт на основі яких буде проводитись розробка емуляторів. Розглянуто існуючі аналоги емуляторів фізичних процесів, їх переваги та недоліки та проведено оцінку і порівняння їх. З огляду лабораторних робіт, стало зрозуміло, що програмне забезпечення має бути інтерактивним, зрозумілим, відсутність перевантаженого інтерфейсу та інформативним. Існуючі аналоги не задовольняють цілі та потреби, тому існує необхідність у власній розробці емуляторів.

У другому розділі було проведено детальний аналіз та вибір мови програмування. Серед варіантів розглянуто JavaScript та PHP, які мають свої особливості, переваги та недоліки. JavaScript - це багатофункціональна мова, яка забезпечує швидку, гнучку і масштабовану розробку сучасних веб-додатків, що робить її найкращим вибором для проектів, які вимагають інтерактивності та продуктивності.

Розглянуто середовища розробки серед яких є найпопулярніші VSCode та Sublime Text. Visual Studio Code (VS Code) є оптимальним вибором для сучасних розробників завдяки своїй універсальності, широкій екосистемі розширень і кросплатформенності.

Серед представлених варіантів бібліотек для розробки програмного забезпечення виділено основі фреймворки React, Angular, Node. У висновку

ми обираємо React з кількох ключових причин, які роблять його ідеальним інструментом для розробки сучасних веб-додатків. По-перше, React.js забезпечує ефективну розробку інтерфейсів завдяки використанню віртуального DOM. Цей підхід мінімізує кількість оновлень у реальному DOM, що значно підвищує продуктивність додатків, особливо при роботі зі складними інтерфейсами. Додатковою перевагою є компонентний підхід, який дозволяє створювати багаторазово використовувані, незалежні компоненти з власним станом. Це спрощує тестування та підтримку коду, роблячи його більш зрозумілим і структурованим. У висновку з цього розділу, обрано мову програмування JavaScript, середовище розробки VSCode та бібліотеку ReactJS.

У третьому розділі було створено план для розробки програмного забезпечення, а також у висновку розробки програмного забезпечення для симуляції фізичних процесів, було продемонстровано симуляцію фізичних процесів, а саме створення магнітного та електричного полів з застосуванням апарату УВЧ-30, а також симуляцій гелій-неонового лазера. Створено сторінки «Головна», «Теоретичні відомості», «Тестування» та «Практична частина».

Результатом роботи є скорочення освоєння матеріалу та виконання лабораторних робіт в умовах конкретних ситуацій. Програмне забезпечення дозволяє студентам виконати швидко та якісно роботу без відвідування фізичних лабораторій, що значно покращує освітній процес.

Використання емулятора може підвищити якість і швидкість навчання, що, у свою чергу, впливає на продуктивність освітньої діяльності: 1) скорочення часу на вивчення матеріалу завдяки інтерактивним методам; 2) підвищення рівня засвоєння знань і практичних навичок студентів; 3) покращення підготовки випускників, що збільшує їхню конкурентоспроможність на ринку праці, підвищує рейтинг навчального закладу та сприяє зростанню кількості абітурієнтів.

З економічних показників можна виділити основні загальні витрати на завершення науково-технічної роботи складають $ЗВ = 222766,06$ грн.

Емулятор дозволяє замінити частину матеріальних ресурсів (лабораторне обладнання, матеріали для практичних занять тощо), це зменшує витрати на їх придбання та підтримку: 1) зменшення потреби у витратних матеріалах; 2) скорочення амортизаційних витрат на обладнання; 3) зниження ризику псування дорогого обладнання через некоректне використання.

Загальна річна економія від заміщення матеріально-технічної бази складає 144860 грн. на рік. При застосуванні емулятора протягом 5 років загальна економія складе $144860 * 5 = 724300$ грн.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Використання симуляції при викладенні нового матеріалу, розв'язанні задач та проведенні лабораторних робіт з фізики, 2022. *На Урок. Освітній проект*. URL: <https://naurok.com.ua/vikoristannya-simulyaci-pri-vikladanni-novogo-materialu-rozv-yazuvanni-zadach-ta-provedenni-laboratornih-ro-bit-z-fiziki-206501.html> (дата звернення 09.10.2024)
2. Розробка емуляторів фізичних процесів для лабораторних робіт з фізики. Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи»: тези міжнародної наук.-практ. конф. (м. Вінниця, 15 жовтня 2024 року) м. Вінниця, 2024 URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/schedConf/overview> (дата звернення: 11.11.2024)
3. В. А. Дяков, В. О. Гудзь, В. Т. Желіба, Б. Г. Іваницький, П.П. Ковальчук, Н. С. Назаренко, І. І, Хаїмзон. *Лабораторний практикум з медичної і біологічної фізики*, 1999. С. 176-184. (дата звернення 10.10.2024)
4. В. А. Дяков, В. О. Гудзь, В. Т. Желіба, Б. Г. Іваницький, П.П. Ковальчук, Н. С. Назаренко, І. І, Хаїмзон. *Лабораторний практикум з медичної і біологічної фізики*, 1999. С. 210-218. (дата звернення 10.10.2024)
5. PhET. Interactive simulations. *University of Colorado Boulder.(UOCB)*. URL: <https://phet.colorado.edu/uk/simulations/filter?subjects=physics> (дата звернення 12.10.2024)
6. What is PheT? Physics Education Technology. *Pedagogy in Action the SERC portal for Educators*. URL: <https://serc.carleton.edu/sp/library/phet/what.html> (дата звернення 12.10.2024)
7. Revolutionizing Education with PhET Simulations: A Partnership with POLYTECH. URL: <https://polytech.com.gr/news-announcements/phet-simulations-open-source-interactive-educational-material/> (дата звернення 12.10.2024)
8. Vascac. Physics in school, HTML5. *Viktor Vascac*. URL: <https://www.vascak.cz/physicsanimations.php?l=ua> (дата звернення 13.10.2024)

9. Enhancing Students' Learning Interest through the Use of Vascak Physics Animation as a Physics Learning Medium. *Jurnal Praktik Baik Pembelajaran Sekolah dan Pasantren*. URL: <https://journal.iistr.org/index.php/PBPSP/article/view/366> (дата звернення 13.10.2024)
10. Algodoo. URL: <https://www.algodoo.com> дата звернення (14.10.2024)
11. Algodoo: A Tool for Encouraging Creativity in Physics Teaching and Learning. By Bor Gregorcic and Madelin Bodin. URL: https://www.researchgate.net/publication/312021107_Algodoo_A_Tool_for_Encouraging_Creativity_in_Physics_Teaching_and_Learning (дата звернення 14.10.2024)
12. What Is JavaScript? A Basic Introduction to JS for Beginners. *Glossary by Jordana A*. URL: <https://www.hostinger.com/tutorials/what-is-javascript> (дата звернення 14.10.2024)
13. Вступ до JavaScript. *JavaScript.Info*. URL: <https://uk.javascript.info/intro> (дата звернення 14.10.2024)
14. The advantages and disadvantages of JavaScript. URL: <https://www.freecodecamp.org/news/the-advantages-and-disadvantages-of-javascript/> (дата звернення 14.10.2024)
15. A comprehensive guide to PHP programming: What you need to know. *Blog by Rory Taal*. URL: <https://codeinstitute.net/global/blog/what-is-php-programming/> (дата звернення 20.10.2024)
16. Advantages of PHP over Other Programming Languages. URL: <https://www.scaler.com/topics/php-tutorial/advantages-of-php/> Tutorial by Yash Agarwal. (дата звернення 20.10.2024)
17. Why use PHP in 2024? Advantages and disadvantages. URL: <https://softjournal.com/insights/pros-and-cons-of-php-programming-language> Tech content by content writer – Sofija Vidjikand and content marketing coordinator – Meghan Neville. (дата звернення 20.10.2024)
18. Sublime Text vs Visual Studio Code: Choosing your coding. *Linkedin*.

URL: <https://www.linkedin.com/pulse/sublime-text-vs-visual-studio-code-choosing-your-coding-tanvir-ahmed-rzmbc> (дата звернення 20.10.2024)

19. Visual Studio Code vs Sublime Text: Uses, Features, and Comparisons. By *Gospel Bassey*. URL: <https://www.turing.com/kb/visual-studio-code-vs-sublime-text> (дата звернення 20.10.2024)

20. М. О. М'ястковська, І. М. Пшембаєв. Використання Phet-симуляцій для виконання домашніх завдань з молекулярної фізики, 22.09.2016. URL: <https://phys.ippo.kubg.edu.ua/wp-content/uploads/2020/05/95141-200685-1-SM-1.pdf> (дата звернення 05.11.2023)

21. *Google Trends*. React vs Angular vs Node. URL: <https://trends.google.com/trends/explore?q=react,angular,node&hl=uk> (дата звернення 05.11.2024)

22. Що таке ReactJS? *Run-IT*. URL: <https://www.run-it.com.ua/shcho-take-react-js/> (дата звернення 07.11.2024)

23. Performance optimization techniques for Reactjs. URL: <https://ieeexplore.ieee.org/document/8869134> (дата звернення 07.11.2024)

24. Angular vs React vs Vue: Which one to choose. *TatvaSoft. Sculpting thoughts...* URL: <https://www.tatvasoft.com/blog/angular-vs-react-vs-vue/> (дата звернення 07.11.2024)

25. Why to use Angular in 2024: An overview of Angular pros and cons. *Leobit. Blog by Bogdan Radkovskyy and Roman Muzyka*. URL: <https://leobit.com/blog/why-to-use-angular-in-an-overview-of-angular-pros-and-cons/> (дата звернення 07.11.2024)

26. When and why to use Node.js: An in-depth guide for 2024. *Relevant Software. Autor by Anna Dziuba*. URL: https://relevant.software/blog/why-and-when-to-use-node-js/#The_Essentials_of_Nodejs_A_Short_Review (дата звернення 07.11.2024)

27. Вивчення апаратів і методів ВЧ- та УВЧ-терапії. URL: <https://hfhf-therapy.vercel.app> (дата звернення 27.11.2024)

28. Вивчення гелій-неонового лазера. URL: <https://he-ne-laser.vercel.app> (дата звернення 27.11.2024)
29. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с. (дата звернення 29.11.2024)
30. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с. (дата звернення 29.11.2024)

ДОДАТКИ

Додаток А
(Обов'язковий)
Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри АІТ

_____ д.т.н., проф. Олег БІСІКАЛО
(підпис) (науковий ступінь, вчене звання)
« 14 » жовтня 2024 року

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

Розробка програмного забезпечення емулятора фізичних процесів для
лабораторних робіт з фізики для студентів Вінницького національного
медичного університету

08-31.МКР.007.02.000 ТЗ

Керівник: к.т.н., доц. каф. АІТ

_____ Ярослав КУЛИК

« 11 » жовтня 2024 р.

Розробив студент гр. ІСТ-23м

_____ Сергій МИРОНЮК

« 11 » жовтня 2024 р.

Вінниця 2024

1. Назва та галузь застосування.

Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів Вінницького національного медичного університету

2. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____ 2024 р., та індивідуальне завдання на МКР, затверджене протоколом №__ засідання кафедри АІТ від «__» _____ 2024 р.

3. Мета та призначення роботи.

Метою дослідження є покращення умов дистанційного навчання, які спрощують та полегшують процес навчання з фізики, а також дозволяють набути практичні навички без перебування в фізичних лабораторіях та зменшити час навчання на вивчення фізичного процесу.

4. Джерела розробки:

В. А. Дяков, В. О. Гудзь, В. Т. Желіба, Б. Г. Іваницький, П.П. Ковальчук, Н. С. Назаренко, І. І, Хаїмзон. Лабораторний практикум з медичної і біологічної фізики, 1999. С. 176-184.

В. А. Дяков, В. О. Гудзь, В. Т. Желіба, Б. Г. Іваницький, П.П. Ковальчук, Н. С. Назаренко, І. І, Хаїмзон. Лабораторний практикум з медичної і біологічної фізики, 1999. С. 210-218.

5. Показники призначення

Вихідні дані до роботи: Апарат УВЧ-30 з технічними характеристиками: частота генератора 40,68 МГц, вихідна потужність для першого ступеня (15+-3), для другого (30+-5), напруга 220В, маса 10кг, габарити корпусу апарата 425x275x230мм; Гелій-неоновий лазер Екран, Дифракційна Ришівка, Штатив з лінійкою, Операційна система не нижче 7, Оперативна пам'ять не нижче 4, Час заняття 45 хвилин, Максимальний час на виконання лабораторної роботи до 2 годин, Доступ до інтернету, Мова програмування JavaScript, Середовище розробки VSCode, Бібліотека для розробки ReactJS.

Результати роботи програми: створення програмного забезпечення, яке дозволяє симулювати фізичні процеси з фізики, веб-додаток містить також теоретичні відомості для успішного виконання лабораторної роботи, тестування задля визначення засвоєння результату студентом, та практична частина із інструкцією виконання та емулятором.

6. Економічні показники

До економічних показників входять:

- витрати на оплату праці – 56 тис. грн;
- витрати на матеріали та вартість програмного забезпечення – до 20 тис. грн;
- витрати на амортизацію обладнання, програмних засобів та приміщень – до 8 тис. грн;
- витрати на паливо та енергію – до 2 тис. грн;
- загальні витрати на розробку – 200 тис. грн;
- науково-технічна розробка переважає існуючі аналоги приблизно в 5,61 рази;

7. Стадії розробки:

1. Аналіз предметної області
2. Вибір оптимальних інформаційних технологій
3. Вибір мови програмування та середовища розробки
4. Програмна реалізація
5. Експериментальна перевірка роботи програми
6. Оформлення пояснювальної записки, графічного матеріалу і презентації

8. Порядок контролю та приймання

Рубіжний контроль провести до « 15 » листопада 2024 р.

Попередній захист МКР провести « 05 » грудня 2024 р.

Захист МКР провести до « 20 » грудня 2024 р.

Розробив студент групи ІСТ-23м _____ Сергій МИРОНЮК

Додаток Б
(Обов'язковий)

Ілюстративна частина

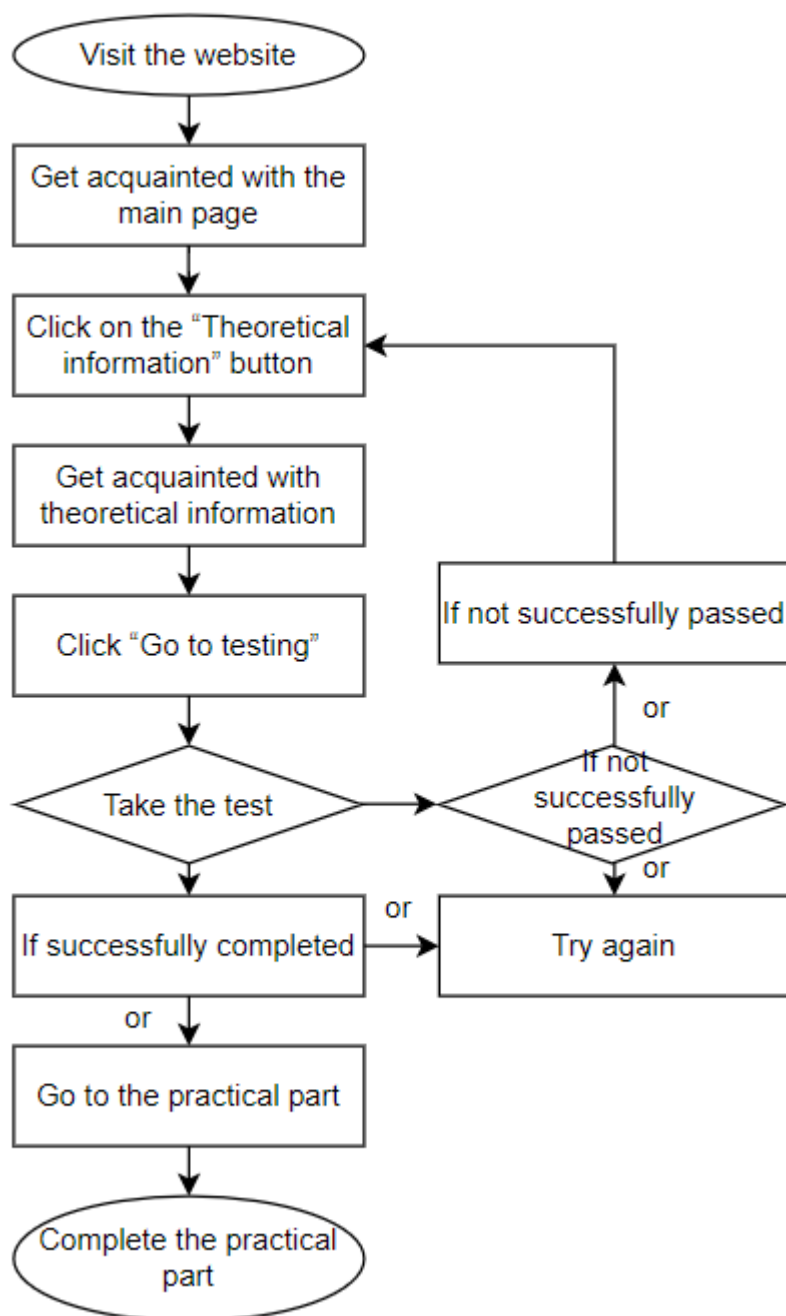


Рисунок Б.1 –Алгоритм виконання лабораторної роботи

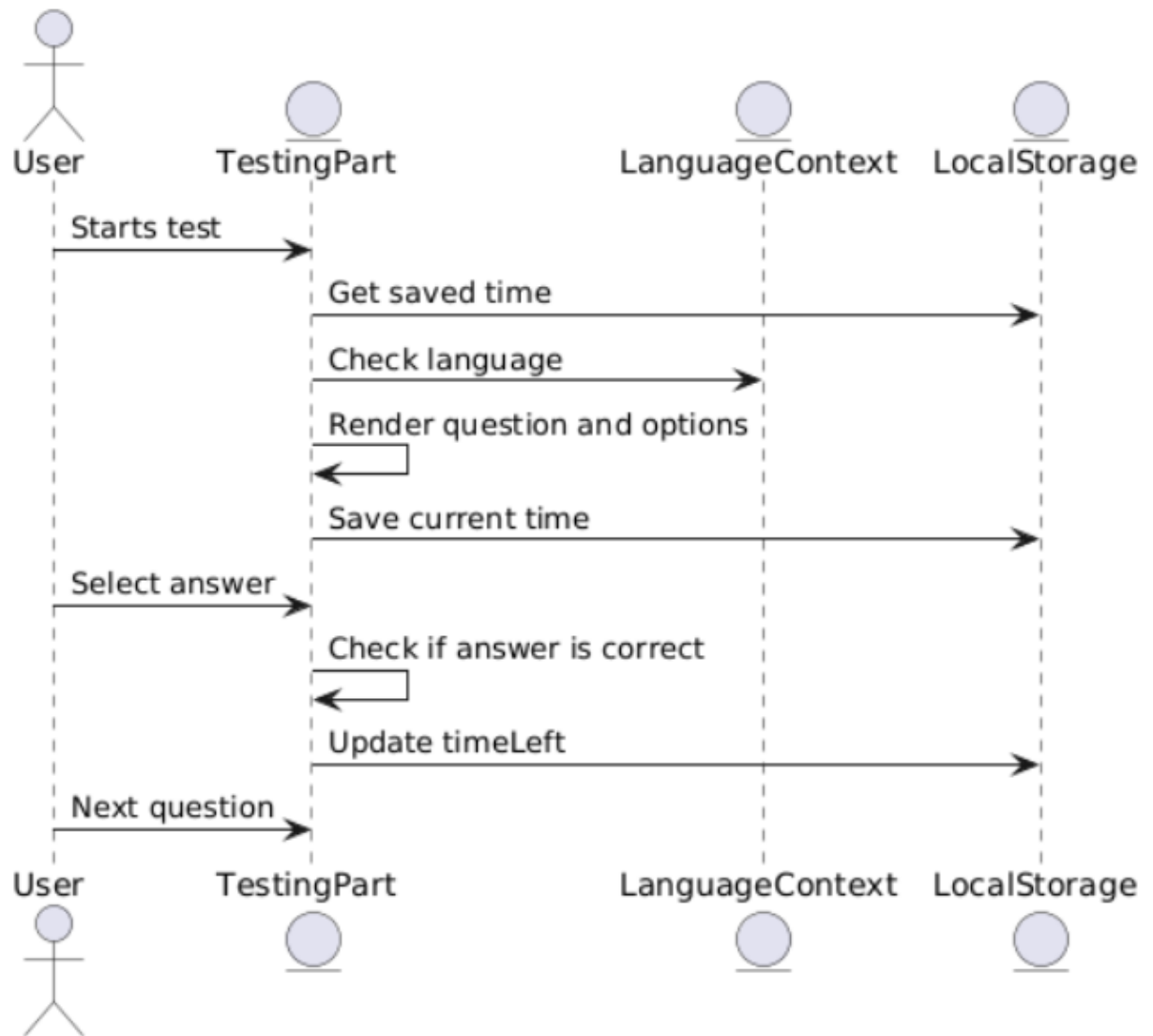


Рисунок Б.2 – UML діаграма послідовностей системи

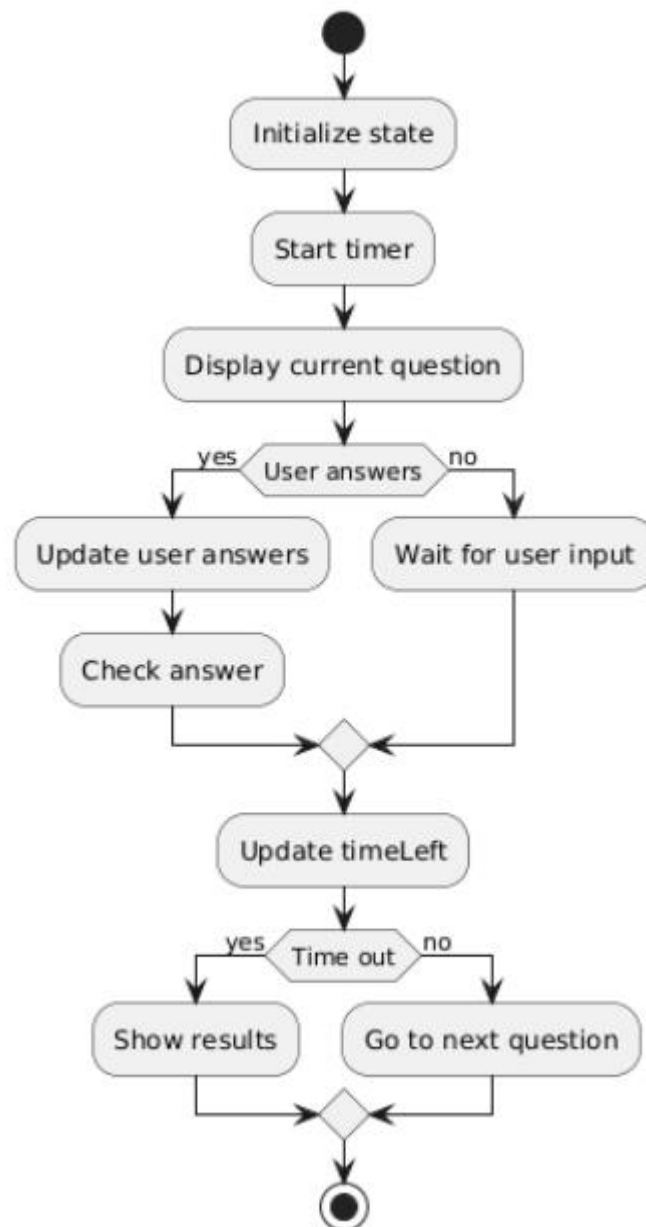


Рисунок Б.3 – UML діаграма активностей системи

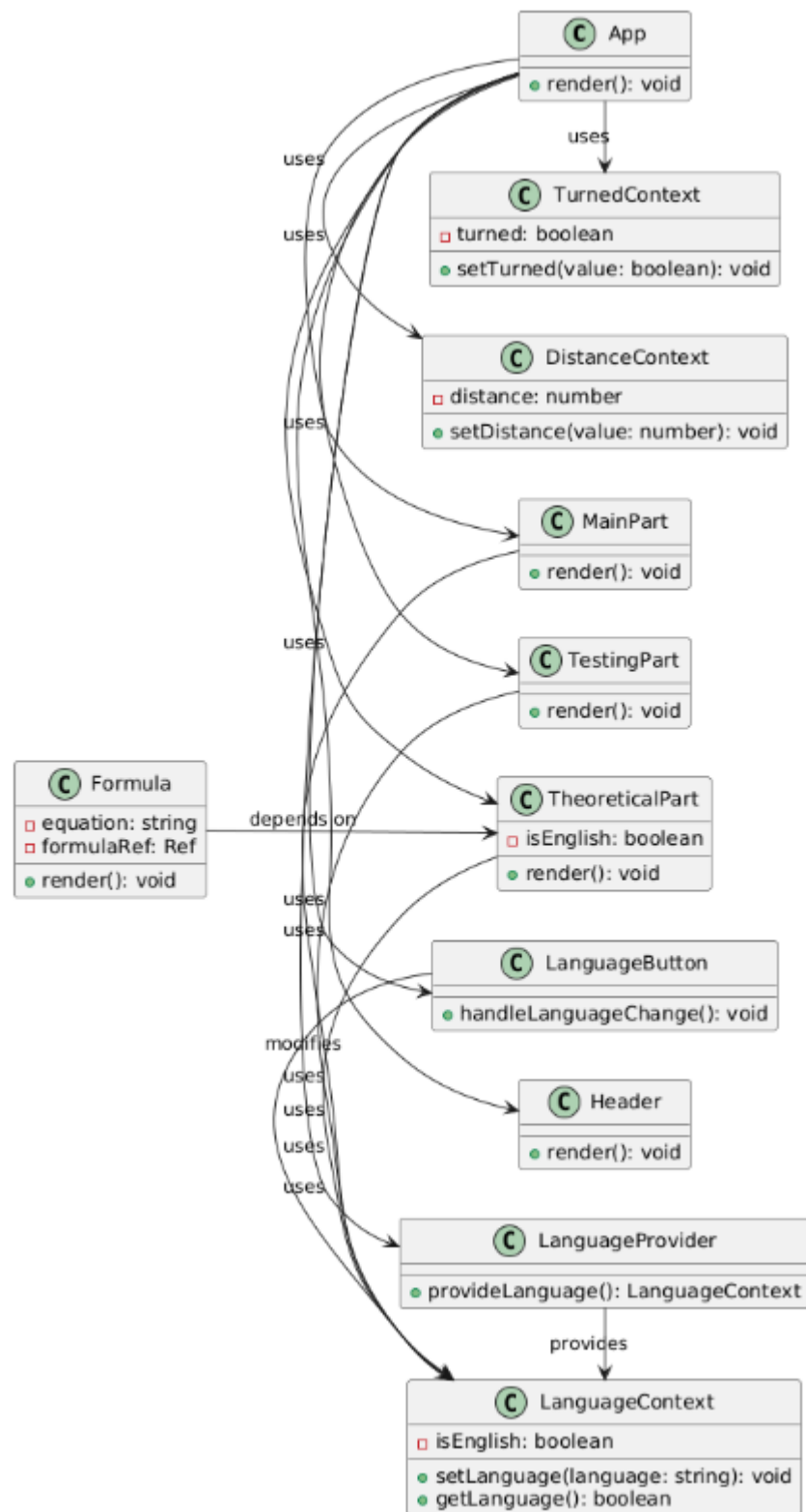


Рисунок Б.4 – UML діаграма класів програми

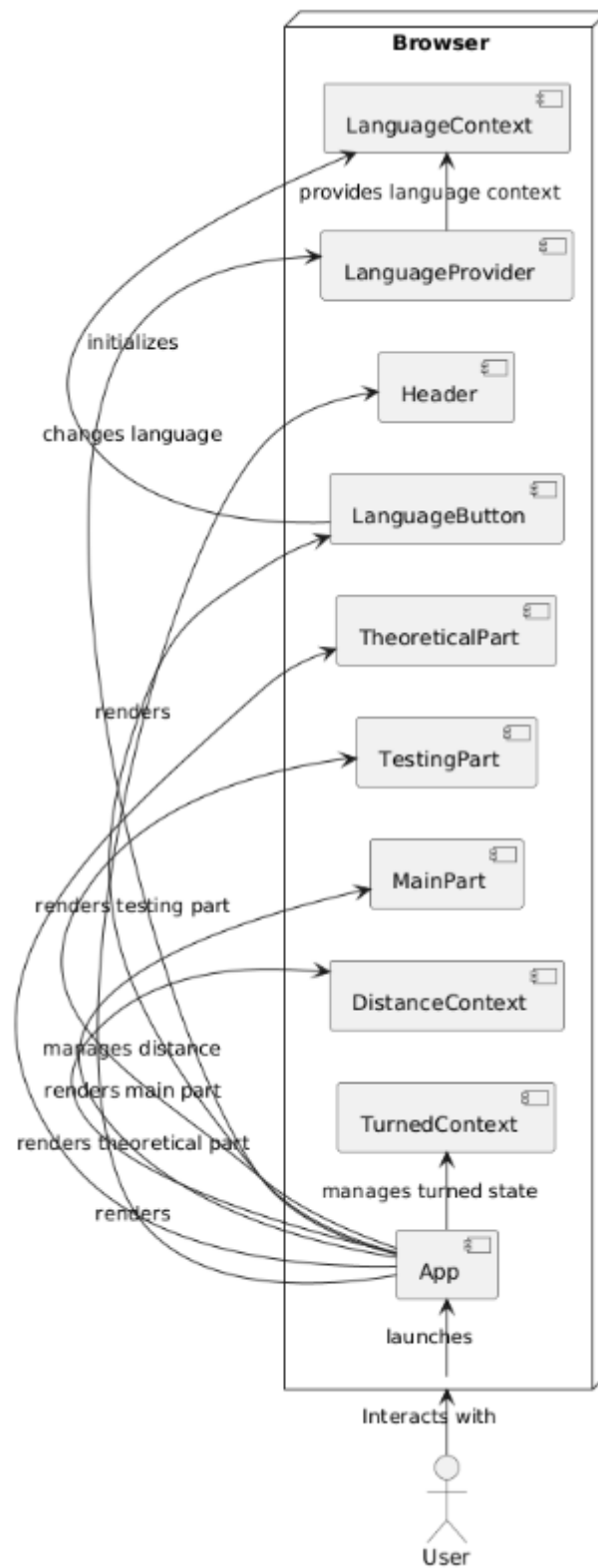


Рисунок Б.5 – UML діаграма розгортання системи

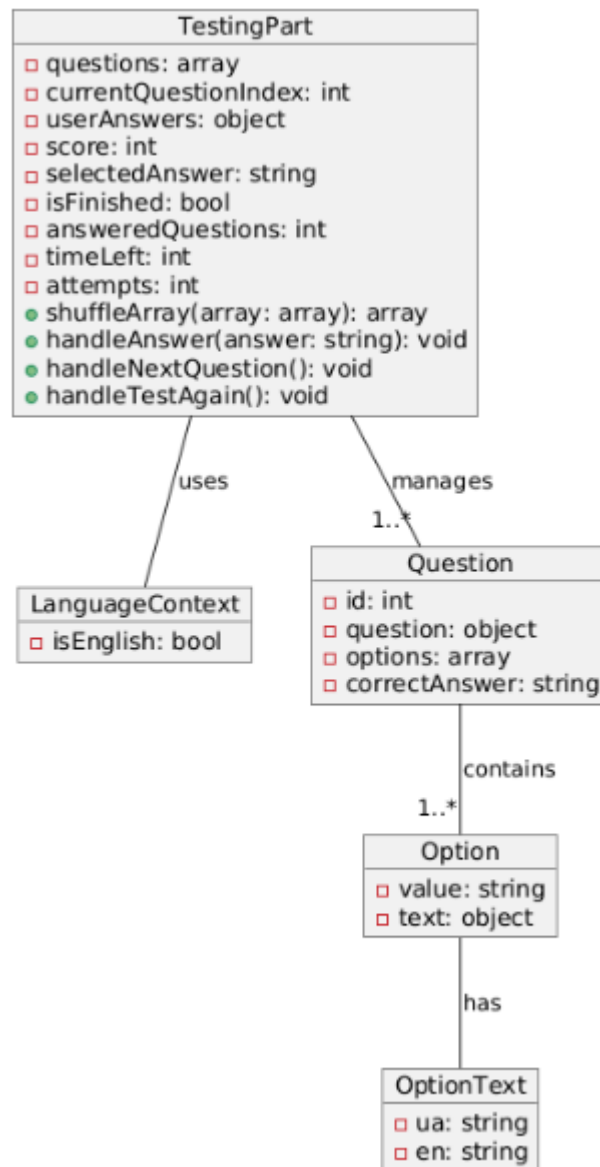


Рисунок Б.6 – UML діаграма об'єктів системи

Додаток В (Обов'язковий) Лістинг коду

Header.jsx

```
import { useContext } from 'react';
import LanguageContext from '../providers/LanguageContext';

import './Header.css';

export default function Header() {
  const { isEnglish } = useContext(LanguageContext);
  return (
    <h1 className="display-4 header">
      {isEnglish
        ? 'STUDY OF THE HE-OF NE LASER'
        : 'ВИВЧЕННЯ ГЕЛІЙ-НЕОНОВОГО (He - Ne) ЛАЗЕРА'}
    </h1>
  );
}
```

ImageBlock.jsx

```
import { useContext } from 'react';
import LanguageContext from '../providers/LanguageContext';
import DistanceContext from '../providers/DistanceContext';
import TurnedContext from '../providers/TurnedContext';

import laserOff from '../assets/laserOff.jpg';
import laserOn from '../assets/laserOn.jpg';
import './ImageBlock.css';

const ImageBlock = () => {
  const { isEnglish } = useContext(LanguageContext);
  const { distance, setDistance } = useContext(DistanceContext);
  const { isTurned } = useContext(TurnedContext);

  const maxDistance = 15;
  const minDistance = 3;

  const handleSliderChange = (e) => {
```

```

const value = parseFloat(e.target.value);
setDistance(value);
};

return (
  <div className="container">
    <div className="image-block">
      <div>
        <img src={isTurned ? lazerOn : lazerOff} className="image" />
      </div>

      <div className="under-image-block">
        <div className="under-image-text">
          <p>
            {isEnglish
              ? 'Distance of the diffraction grating from the screen: '
              : 'Відстань дифракційної решітки від екрану:'}
          </p>
          <p>
            {distance}
            {isEnglish ? '(cm)' : '(см)}
          </p>
        </div>
        <input
          type="number"
          value={distance}
          className="input-range"
          min={minDistance}
          max={maxDistance}
          step="0.1"
          onChange={(e) => setDistance(e.target.value)}
        />
      </div>
      <input
        type="range"
        value={distance}
        className="input-slider custom-slider"
        min={minDistance}
        max={maxDistance}
        step="0.1"
        onChange={handleSliderChange}
        style={{ direction: 'rtl' }}
      />
    </div>
  </div>
);

```

```

    />
  </div>
</div>
);
};

export default ImageBlock;

```

Instruction.jsx

```

import React, { useState, useContext } from 'react';
import './Instruction.css';
import LanguageContext from '../providers/LanguageContext';

const Instruction = () => {
  const { isEnglish } = useContext(LanguageContext);

  const initialData = [
    ['k', 'R', 'L', 'l<sub>k</sub> = L<sub>k</sub>/2', 'tg a', 'sin a', 'λ', 'δ<sub>λ</sub>', 'ε', 'δ<sub>ε</sub>'], //
    Заголовки стовпців
    ['1', '', '', '', '', '', '', ''],
    ['2', '', '', '', '', '', '', ''],
    ['3', '', '', '', '', '', '', ''],
    ['', '', '', '', '', 'λ- =', 'δλ- =', 'ε- =', 'δε- ='],
  ];

  const [tableData, setTableData] = useState(initialData);

  const handleChange = (e, rowIndex, colIndex) => {
    const newData = [...tableData];
    newData[rowIndex][colIndex] = e.target.value;
    setTableData(newData);
  };

  return (
    <div className="instruction-content">
      <h2>{isEnglish ? 'Instructions' : 'Інструкція виконання практичного завдання'}</h2>
      <p>
        {isEnglish
          ? 'Here are the instructions for performing the task. Follow each step carefully.'
          : 'Ось інструкція для виконання завдання. Дотримуйтеся кожного кроку уважно.'}
      </p>
    </div>
  );
}

```

```

<ol>
  <li>{isEnglish ? 'Step 1: Click the “Power” button.' : 'Крок 1: Натисніть кнопку "Живлення".'}</li>
  <li>{isEnglish ? 'Step 2: Click the “Start” button.' : 'Крок 2: Натисніть кнопку "Старт".'}</li>
  <li>{isEnglish ? 'Step 3: Set the distance of the diffraction grating from the screen. There are two ways to do
this: you can either enter the distance in the input field or move the diffraction grating to the desired distance.' :
'Крок 3: Встановіть відстань дифракційної ґратки від екрану. Ви можете зробити це двома способами,
ввести значення у полі вводу, або потягнути ґратку на потрібну відстань.'}</li>
    <p>{isEnglish ? 'On the ruler, you can observe red disks symmetrically located relative to the central (zero)
maximum in the horizontal direction.' : 'На лінійці ви можете спостерігати червоні диски, симетрично
розташовані відносно центрального (нульового) максимуму в горизонтальному напрямі.'}</p>
    <li>{isEnglish ? 'Step 4. Measure the distance between the diffraction grating and the screen.' : 'Крок 4.
Виміряти віддаль між дифракційною ґраткою і екраном.'}</li>
    <li>{isEnglish ? 'Step 5: Measure the distance L between the centers of the first-order maxima and enter it in
the table.' : 'Крок 5: Виміряти віддаль L між центрами максимумів першого порядку та занести в
таблицю.'}</li>
    <li>{isEnglish ? 'Step 6: Perform similar measurements for the other maxima and enter it in the table.' : 'Крок
6: Аналогічні виміри провести для інших максимумів та занести в таблицю.'}</li>
    <li>{isEnglish
      ? (<
        Step 7: Calculate the wavelength of light  $\lambda$  incident on the diffraction grating using the formula  $\lambda = d \sin \alpha / K$ 
and write the result in the column.
         $\lambda =$  <input type="text" className="inline-input" /> nanometers.</>)
      : (< Крок 7. Розрахувати довжину хвилі світла  $\lambda$ , яке падає на дифракційну ґратку за формулою  $\lambda =$ 
 $d \sin \alpha / K$  та вписати результат у колонку.
         $\lambda =$  <input type="text" className="inline-input"/> нанометрів.</> )}</li>
</ol>

```

```

<table className="data-table">
  <thead>
    <tr>
      {tableData[0].map((header, index) => (
        <th key={index}>
          <span dangerouslySetInnerHTML={{ __html: header }} />
        </th>
      ))}
    </tr>
  </thead>
  <tbody>
    {tableData.slice(1).map((row, rowIndex) => (
      <tr key={rowIndex + 1}>
        {row.map((value, colIndex) => (
          <td key={colIndex}>

```

```

    { /* Якщо це перший стовпець, показуємо лише текст */ }
    { colIndex === 0 ? (
      <span>{value}</span>
    ) : (
      <input
        type="text"
        value={value}
        onChange={(e) => handleChange(e, rowIndex + 1, colIndex)}
      />
    ) }
  </td>
)}
</tr>
)}
</tbody>
</table>
</div>
);
};

```

export default Instruction;

LanguageButton.jsx

```

import { useContext } from 'react';
import LanguageContext from '../providers/LanguageContext';

import './LanguageButton.css';

export default function LanguageButton() {
  const { isEnglish, setEnglish } = useContext(LanguageContext);
  return (
    <div className="header__language">
      <div className="lg">
        <p>Українська</p>
      </div>
      <div className="form-check form-switch turn-switch">
        <input
          id="switch-lang"
          className="form-check-input"
          type="checkbox"
          role="switch"

```



```

        onClick={() => setEnglish(!isEnglish)}
      />
    </div>
    <div className="lg">
      <p>English</p>
    </div>
  </div>
);
}

```

MainPart.jsx

```

import ImageBlock from '../ImageBlock/ImageBlock';
import MyJubmotron from '../MyJubmotron/MyJubmotron';
import Screen from '../Screen/Screen';
import Instruction from '../Instruction/Instruction';

```

```

import './MainPart.css';

```

```

export default function MainPart() {
  return (
    <>
      <div className="main-part">
        <MyJubmotron />
        <ImageBlock />
        <Screen />
      </div>
      <Instruction />
    </>
  );
}

```

Modal.jsx

```

import { Modal } from 'react-bootstrap';
import image1 from '../../assets/modal/1.png';
import image2 from '../../assets/modal/2.jpg';
import image3 from '../../assets/modal/3.png';
import image4 from '../../assets/modal/4.jpg';
import image5 from '../../assets/modal/5.jpg';

const MyModal = ({ show, handleClose }) => {
  return (

```

```

<Modal show={show} onHide={handleClose} size="lg">
  <Modal.Header closeButton></Modal.Header>
  <Modal.Body className="d-grid gap-md-5">
    <div className="row justify-content-center">
      <div className="col d-flex justify-content-center">
        <img src={image1} className="modal-image" />
      </div>
      <div className="col d-flex justify-content-center">
        <img src={image2} className="modal-image" />
      </div>
    </div>
    <div className="row justify-content-evenly">
      <div className="col d-flex justify-content-center">
        <img src={image3} className="modal-image" />
      </div>
      <div className="col d-flex justify-content-center">
        <img src={image4} className="modal-image" />
      </div>
    </div>
    <div className="row">
      <div className="col d-flex justify-content-center">
        <img src={image5} className="modal-image" />
      </div>
    </div>
  </Modal.Body>
</Modal>
);
};

```

```
export default MyModal;
```

MyJubmotron.jsx

```

import { useContext, useState, useEffect } from 'react';
import { Button, Container } from 'react-bootstrap';
import MyModal from '../Modal/Modal';

import LanguageContext from '../providers/LanguageContext';
import TurnedContext from '../providers/TurnedContext';
import DistanceContext from '../providers/DistanceContext';

import './MyJubmotron.css';

```

```

const MyJumbotron = () => {
  const { isEnglish } = useContext(LanguageContext);
  const { setTurned, setStarted } = useContext(TurnedContext);
  const { distance } = useContext(DistanceContext);

  const [isOpen, setOpen] = useState(false);
  const [isAbled, setAbled] = useState(false);

  useEffect(() => {
    distance >= 3 ? setAbled(true) : setAbled(false);
  }, [distance]);

  return (
    <div className="my-jumbotron">
      <Container className='my-jumbotron-container'>
        <MyModal show={isOpen} handleClose={setOpen} />
        <Button
          className="info-button"
          variant="secondary"
          size="lg"
          data-bs-toggle="modal"
          data-bs-target="#myModal"
          onClick={() => setOpen(true)}
        >
          {isEnglish ? 'Info' : 'Довідка'}
        </Button>

        <div className="turn-block">
          <p>{isEnglish ? 'Power' : 'Живлення'}</p>
          <div className="form-check form-switch turn-switch">
            <input
              className="form-check-input"
              type="checkbox"
              role="switch"
              onClick={(e) => {
                setTurned(e.target.checked);
                if (distance >= 3) {
                  setAbled(e.target.checked);
                }
                if (!setTurned) {
                  setStarted(false);

```

```

    }
  }}
/>
</div>
</div>

<button
  className="btn btn-success btn-lg start-btn"
  disabled={!isAbled}
  onClick={() => {
    setStarted(true);
    setAbled(false);
  }}
  variant="success"
  id="start"
>
  {isEnglish ? 'Start' : 'Crapt'}
</button>
</Container>
</div>
);
};

export default MyJumbotron;
```

Ruler.jsx

```

import { useContext, useState } from 'react';
import TurnedContext from '../providers/TurnedContext';

const Ruler = () => {
  const { isTurned } = useContext(TurnedContext);

  const [minZeroPos, setMinZeroPos] = useState(
    `translate3d(589px, -220px, 0px)`
  );

  const rulerSpacing = [
    15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7,
    8, 9, 10, 11, 12, 13, 14, 15,
  ];
};
```

```

const getDotPosition = (dot, index) => {
  let finalDotPosition = dot.distance;
  setSelectedMin(index + 1);
  for (let i = 0; i < index; i++) {
    finalDotPosition += dot.distance;
  }
  setDotPosition(finalDotPosition);
};

return (
  <div className="ruler">
    {rulerSpacing.map((space) => (
      <a key={space}>{space}</a>
    ))}
    {isTurned && (
      <ul className="laser">
        <li
          className="red-dot"
          style={{ transform: minZeroPos }}
          onClick={() => {
            setDotPosition(0.0000000000);
            setSelectedMin(0);
          }}
        ></li>
        {minimumsPosRight.map((right, i) => (
          <li
            key={right}
            className="red-dot"
            style={{ transform: right.pos, opacity: right.opacity }}
            onClick={() => getDotPosition({ dot: right, index: i })}
          ></li>
        ))}
      </ul>
    )}
  </div>
);
};

export default Ruler;

```

Screen.jsx

```
import React, { useState, useEffect, useContext } from 'react';
```

```

import LanguageContext from '../providers/LanguageContext';
import TurnedContext from '../providers/TurnedContext';
import DistanceContext from '../providers/DistanceContext';

import './Screen.css';

const Screen = () => {
  const { isEnglish } = useContext(LanguageContext);
  const { isStarted } = useContext(TurnedContext);
  const { distance } = useContext(DistanceContext);

  const rulerSpacing = [
    15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7,
    8, 9, 10, 11, 12, 13, 14, 15,
  ];

  const minZeroPosVal = -475;
  const oneCmStep = 3;
  const amountOfMinimums = 10;
  const dotsOpacity = 0.6;

  const [minZeroPos, setMinZeroPos] = useState(
    `translate3d(-475px, 20px, 0px)`
  );

  const [minimumsPosRight, setMinimumsPosRight] = useState([]);
  const [minimumsPosLeft, setMinimumsPosLeft] = useState([]);
  const [dotPosition, setDotPosition] = useState(0);
  const [selectedMin, setSelectedMin] = useState(0);

  const getDistanceBetweenDots = distance;

  useEffect(() => {
    calculateDistance();
  }, [getDistanceBetweenDots]);

  const getDotPosition = (dot, index) => {
    let finalDotPosition = dot.distance;
    setSelectedMin(index + 1);
    for (let i = 0; i < index; i++) {
      finalDotPosition += dot.distance;
    }
    setDotPosition(finalDotPosition);
  };

```

```

const calculateDistance = (distanceCm) => {
  let yPos = 20;
  let opacityRight = dotsOpacity;
  let distanceBetweenDots = getDistanceBetweenDots * oneCmStep;

  const minimumsPosRightArr = [];
  const minimumsPosLeftArr = [];

  for (let i = 0; i < amountOfMinimums; i++) {
    const distance = minZeroPosVal + distanceBetweenDots * (i + 1);
    if (distanceCm) {
      opacityRight -= distanceCm / 25;
    }
    minimumsPosRightArr.push({
      pos: `translate3d(${distance}px, -${-20 + yPos}px, 0px)`,
      opacity: opacityRight,
      distance: distanceCm,
    });
    yPos += 20;
  }

  let yPosLeft = 10;
  let opacityLeft = dotsOpacity;

  for (let i = 0; i < amountOfMinimums; i++) {
    const distance = minZeroPosVal - distanceBetweenDots * (i + 1);
    if (distanceCm) {
      opacityLeft -= distanceCm / 25;
    }
    minimumsPosLeftArr.push({
      pos: `translate3d(${distance}px, ${yPosLeft - 210}px, 0px)`,
      opacity: opacityLeft,
      distance: distanceCm,
    });
    yPosLeft -= 20;
  }

  setMinimumsPosRight(minimumsPosRightArr);
  setMinimumsPosLeft(minimumsPosLeftArr);
};

```

```

return (
  <div className="screen">
    <div className="screen-container">
      <div className="screen-text">
        <h1>{isEnglish ? 'Screen' : 'Екран'}</h1>
        <p>
          {isEnglish
            ? 'To get the distance from the center minimum you need to tap on it starting from the right side towards 0.'
            : 'Для отримання відстані від центрального мінімуму потрібно натиснути на нього починаючи з
правої сторони відносно 0.'}
        </p>
        <p>
          {isEnglish
            ? 'The minima on the left are mirror-symmetric rules with the same distance'
            : 'Мінімуми зліва - дзеркально симетричні правим та мають таку ж відстань'}
        </p>
      </div>

      <div className="ruler-container">
        <div className="ruler">
          {rulerSpacing.map((space) => (
            <span key={space + Math.random() * 100}>{space}</span>
          ))}
          {isStarted && (
            <ul className="laser">
              <li
                className="red-dot"
                style={{ transform: minZeroPos }}
                onClick={() => {
                  setDotPosition(0.0000000000);
                  setSelectedMin(0);
                }}
              ></li>
              {minimumsPosRight.map((right, i) => (
                <li
                  key={right + Math.random() * 100}
                  className="red-dot"
                  style={{ transform: right.pos, opacity: right.opacity }}
                  onClick={() => getDotPosition({ dot: right, index: i })}
                ></li>
              ))}
              {minimumsPosLeft.map((left, i) => (

```



```

    <li
      key={left + Math.random() * 100}
      className="red-dot"
      style={{ transform: left.pos, opacity: left.opacity }}
      onClick={() => getDotPosition({ dot: left, index: i })}
    ></li>
  )))
</ul>
})
</div>
</div>
</div>
);
};

```

```
export default Screen;
```

TestingPart.jsx

```

import React, { useState, useEffect, useContext } from "react";
import { Link } from "react-router-dom";
import LanguageContext from '../providers/LanguageContext';
import './TestingPart.css';

const initialQuestions = [
  {
    id: 1,
    question: {
      ua: "Переходи атомів речовини з вищих енергетичних рівнів на нижчі супроводжуються...",
      en: "The transitions of atoms from higher energy levels to lower ones are accompanied by..."
    },
    options: [
      { value: "1", text: { ua: "збудженням атомів", en: "excitation of atoms" } },
      { value: "2", text: { ua: "випромінюванням квантів", en: "emission of quanta" } },
      { value: "3", text: { ua: "поглинанням енергії", en: "absorption of energy" } },
      { value: "4", text: { ua: "незмінним станом атомів", en: "unchanged state of atoms" } }
    ],
    correctAnswer: "2",
  },
];

```

```
const shuffleArray = (array) => array.sort(() => Math.random() - 0.5);
```

```
function TestingPart() {
```

```
  const { isEnglish } = useContext(LanguageContext);
```

```
  const [questions, setQuestions] = useState([]);
```

```
  const [currentQuestionIndex, setCurrentQuestionIndex] = useState(0);
```

```
  const [userAnswers, setUserAnswers] = useState({});
```

```
  const [score, setScore] = useState(0);
```

```
  const [selectedAnswer, setSelectedAnswer] = useState(null);
```

```
  const [isFinished, setIsFinished] = useState(false);
```

```
  const [answeredQuestions, setAnsweredQuestions] = useState(0);
```

```
  const [timeLeft, setTimeLeft] = useState(() => {
    const savedTime = localStorage.getItem("timeLeft");
    return savedTime ? parseInt(savedTime, 10) : 420;
  });
```

```
  const [attempts, setAttempts] = useState(() => {
    const savedAttempts = localStorage.getItem("attempts");
    return savedAttempts ? parseInt(savedAttempts, 10) : 1;
  });
```

```
  useEffect(() => {
    const randomQuestions = shuffleArray(initialQuestions)
      .slice(0, 10)
      .map((question) => ({
        ...question,
        options: shuffleArray(question.options),
      }));
    setQuestions(randomQuestions);
  }, []);
```

```
  useEffect(() => {
    if (timeLeft > 0 && !isFinished) {
      const timer = setInterval(() => {
        setTimeLeft((prevTime) => {
          const newTime = prevTime - 1;
          localStorage.setItem("timeLeft", newTime);
          if (newTime <= 0) {
            clearInterval(timer);
          }
        });
      }, 1000);
    }
  });
```

```

        setIsFinished(true);
    }
    return newTime;
});
}, 1000);

return () => clearInterval(timer);
}
}, [timeLeft, isFinished]);

const handleAnswer = (answer) => {
    setSelectedAnswer(answer);
};

const handleNextQuestion = () => {
    const currentQuestion = questions[currentQuestionIndex];

    if (selectedAnswer === currentQuestion.correctAnswer) {
        setScore((prevScore) => prevScore + 10);
    }

    setUserAnswers({
        ...userAnswers,
        [currentQuestion.id]: {
            selectedAnswer,
            isCorrect: selectedAnswer === currentQuestion.correctAnswer,
        },
    });

    setAnsweredQuestions((prevAnswered) => prevAnswered + 1);

    if (currentQuestionIndex === questions.length - 1) {
        const totalScore = score + (selectedAnswer === currentQuestion.correctAnswer ? 10 : 0);

        const penalty = Math.max(0, attempts - 1);
        const finalScore = totalScore - penalty;

        const correctedFinalScore = Math.max(finalScore, 0);
        setScore(correctedFinalScore);
        setIsFinished(true);
    } else {
        setCurrentQuestionIndex(currentQuestionIndex + 1);
    }
}

```

```

    setSelectedAnswer(null);
  }
};

const handleTestAgain = () => {
  const randomQuestions = shuffleArray(initialQuestions)
    .slice(0, 10)
    .map((question) => ({
      ...question,
      options: shuffleArray(question.options),
    }));
  setQuestions(randomQuestions);
  setCurrentQuestionIndex(0);
  setUserAnswers({});
  setAnsweredQuestions(0);
  setIsFinished(false);
  setTimeLeft(420);
  localStorage.setItem("timeLeft", 420);

  const newAttempts = attempts >= 10 ? 1 : attempts + 1;
  setAttempts(newAttempts);
  localStorage.setItem("attempts", newAttempts);

  setScore(0);
};

if (questions.length === 0) {
  return <div>{isEnglish ? 'Load...' : 'Завантаження...'}</div>;
}

const currentQuestion = questions[currentQuestionIndex];

return (
  <div className="testing-part">
    <h2 className="title">{isEnglish ? 'Testing' : 'Тестування'}</h2>
    <p className="timer">{isEnglish ? 'Time is running out.' : 'Залишилось часу:'} {Math.floor(timeLeft /
60)}:{String(timeLeft % 60).padStart(2, "0")} {isEnglish ? 'min' : 'хв'}</p>
    <p className="attempts">{isEnglish ? 'Number of attempts:' : 'Кількість спроб:'} {attempts}</p>
    {!isFinished ? (
      <div className="question-container">
        <p className="question-text">{currentQuestion.question[isEnglish ? 'en' : 'ua']}</p>
        <div className="options-container">

```

```

{currentQuestion.options.map((option) => (
  <label key={option.value} className="option">
    <input
      type="radio"
      name={`question-${currentQuestion.id}`}
      value={option.value}
      onChange={(e) => handleAnswer(e.target.value)}
      checked={selectedAnswer === option.value}
      className="radio-input"
    />
    {option.text[isEnglish ? 'en' : 'ua']}
  </label>
)}}
</div>

<div className="button-container">
  <button
    onClick={handleNextQuestion}
    disabled={selectedAnswer === null}
    className="next-button"
  >
    {isEnglish ? 'Next question' : 'Наступне запитання'}
  </button>
</div>
</div>
): (
  <div className="result-container">
    <h3>{isEnglish ? 'Test completed!' : 'Тест завершено!'}</h3>
    <p>{isEnglish ? 'Your result is:' : 'Ваш результат:'} {score} {isEnglish ? 'percent' : '%'}</p>

    {score >= 60 ? (
      <div>
        <p>{isEnglish ? 'Congratulations, you have scored enough points. Take a screenshot of your results and proceed to the practice section or try to improve your score!' : 'Вітаю, ви набрали достатню кількість балів. Зробіть знімок екрану результатів та переходьте до практичної частини або спробуйте підвищити бал!'}</p>
        <Link to="/practice">
          <button className="next-button">{isEnglish ? 'Go to the practical part' : 'Перейти до практичної частини'}</button>
        </Link>
        <button onClick={handleTestAgain} className="next-button"> {isEnglish ? 'Try again' : 'Спробувати ще раз'}</button>
      </div>
    )
  )
}

```

```

    ) : (
      <div>
        <p>{isEnglish ? 'You scored too low. Read more about the theory and try again.' : 'Ви набрали малу
        кількість відповідей. Детальніше ознайомтесь з теоретичними відомостями та спробуйте знову.'}</p>
        <Link to="/theory">
          <button onClick={handleTestAgain} className="next-button">{isEnglish ? 'Go to theoretical
          information' : 'Перейти до теоретичних відомостей'}</button>
        </Link>
        <button onClick={handleTestAgain} className="next-button"> {isEnglish ? 'Try again' : 'Спробувати ще
        раз'}</button>
      </div>
    )}
    <div className="button-container">
      <Link to="/">
        <button onClick={handleTestAgain} className="next-button">{isEnglish ? 'Go to the main page' : 'На
        головну'}</button>
      </Link>
    </div>
  </div>
)}
</div>
);
}

```

```
export default TestingPart;
```

TheoreticalPart.jsx

```

import "../TheoreticalPart.css";
import "katex/dist/katex.min.css";
import heneImage from "../../assets/theoreticallaser/hene.jpg";
import schemeImage from "../../assets/theoreticallaser/scheme.jpg";
import findAngle from "../../assets/theoreticallaser/findangle.jpg";
import schemelaserImage from "../../assets/theoreticallaser/schemelaser.jpg";
import LanguageContext from "../../providers/LanguageContext";
import React, { useEffect, useRef, useContext } from 'react';
import katex from 'katex';
import { Link } from 'react-router-dom';

const Formula = ({ equation }) => {
  const formulaRef = useRef(null);

```

```

useEffect(() => {
  if (formulaRef.current) {
    katex.render(equation, formulaRef.current, {
      throwOnError: false,
    });
  }
}, [equation]);

return <span ref={formulaRef}></span>;
};

function TheoreticalPart() {

  const { isEnglish } = useContext(LanguageContext);

  return (
    <div className="theoretical-part">
      <h1>{isEnglish
        ? 'Theoretical information. Laboratory work "Study of a helium-neon laser.'"
        : 'Теоретичні відомості. Лабораторна робота "Вивчення гелій-неонового лазера."}</h1>
      <p>
        {isEnglish
          ? 'In this work, a continuous-phase helium-neon laser is used, in which the active medium is a mixture of Ne
            and Ne. Its operation is based on the interaction of atoms of two gases that have close energy levels. Thus, the
            energy of level 2 of helium is quite close to the energy of level 3 of neon (Fig. 1.1).'
          : 'В даній роботі використовується гелій-неоновий лазер безперервної дії, в якому активним
            середовищем є суміш He і Ne. В основу його роботи покладено взаємодію атомів двох газів, які мають
            близькі енергетичні рівні. Так, енергія рівня 2 гелію досить близька до енергії рівня 3 неону (рис. 1.1).'}
        </p>

        <div className="image-container">
          <img src={heneImage} alt="Гелій-неоновий лазер" />
          <p className="image-caption">{isEnglish ? 'Figure 1.1 - Helium-neon laser' : 'Рисунок 1.1 - Гелій-
            неоновий лазер'}</p>
        </div>

        <div className="image-container">
          <img src={schemeImage} alt="Структурна схема газового лазера" />
          <p className="image-caption">{isEnglish ? 'Figure 1.2 - Block diagram of a gas laser' : 'Рисунок 1.2 -
            Структурна схема газового лазера'}</p>
        </div>

```

<p>

{isEnglish

? 'Diffraction grating. is a glass plate on which opaque strokes of width b are applied at regular intervals a . The spaces between the strokes are essentially transparent slits that transmit light. The total width of the stroke and slit is called the constant or period d of the diffraction grating, i.e. $d = a + b$. When the grating is illuminated with monochromatic light, which is a laser beam, a diffraction pattern is observed on the screen in the form of alternating dark and light areas. The light regions correspond to the main diffraction maxima 1, 2, etc., which exist under the condition of formula (1.1):'

: 'Дифракційна решітка. являє собою скляну пластину, на якій через рівні проміжки a нанесені непрозорі штрихи шириною b . Проміжки між штрихами є по суті прозорими щілинами, які пропускають світло. Сумарна ширина штриха і щілини називається сталою або періодом d дифракційної решітки, тобто $d = a + b$. При освітленні решітки монохроматичним світлом, яким є лазерний промінь, на екрані спостерігається дифракційна картина. у вигляді темних і світлих областей, які чергуються. Світлі області відповідають головним дифракційним максимумам 1, 2 і т.д., що існують при умові формули (1.1):'

</p>

<div className="formula-container">

<Formula equation=" $d \sin \alpha = K \lambda$ " />

(1.1)

</div>

<p>

{isEnglish

? 'where K is the ordinal number of the maximum; λ is the light wavelength.'

: 'де K – порядковий номер максимуму; λ – довжина світлової хвилі.'

</p>

<p>

{isEnglish

? 'The main maxima are located symmetrically on either side of the central or zero maximum ($K=0$, $a=0$). Knowing d , a , and K , we can determine the wavelength λ of the light incident on the diffraction grating by the formula (1.2):'

: 'Головні максимуми розташовані симетрично по обидві сторони від центрального або нульового максимуму ($K=0$, $a=0$). Знаючи d , a і K , можна визначити довжину λ хвилі світла, яке падає на дифракційку решітку, за формулою (1.2):'

</p>

<div className="formula-container">

<Formula equation=" $\lambda = \frac{d \sin \alpha}{K}$ " />

(1.2)

</div>


```

<p>
  {isEnglish
    ? 'To find the angle a at which the main maximum of the K-th order is visible, we use trigonometric concepts
      (Fig. 1.3).'
    : 'Для знаходження кута а, під яким видко головний максимум К - го порядку, скористаємося
      тригонометричними поняттями (рис. 1.3).'}
  </p>

  <div className="image-container">
    <img src={findAngle} alt="Знаходження кута а" />
    <p className="image-caption">{isEnglish ? 'Figure 1.3 - Finding the angle a' : 'Рисунок 1.3 - Знаходження
      кута а'}</p>
  </div>

  <p>
    {isEnglish
      ? 'It is obvious from Figure 1.3 that for the first maximum, the angle is determined by the formula (1.3):'
      : 'З рисунка 1.3 очевидно, що для першого максимуму кут знаходиться за формулою (1.3):'}
    </p>

    <div className="formula-container">
      <Formula equation="(t_{ga_1} = \frac{l_1}{R} = \frac{L_1}{2R})"/>
      <span className="formula-number">(1.3)</span>
    </div>

    <p>
      {isEnglish
        ? 'where R is the distance between the diffraction grating and the screen;'
        : 'де R – віддаль між дифракційною решіткою та екраном;'}
      </p>

      <p>
        {isEnglish
          ? 'L1 is the distance between the centers of the first-order maxima (1.4).'
          : 'L1 – віддаль між центрами максимумів першого порядку (1.4).'}
        </p>

        <div className="formula-container">
          <Formula equation="l_1 = l_2 = \frac{L_1}{2}" />
          <span className="formula-number">(1.4)</span>
        </div>

```

<p>

{isEnglish

? 'It is possible to measure L_k between the centers of the maxima of other orders In general, we can write the formula (1.5):'

: 'Можна виміряти L_k між центрами максимумів інших порядків В загальному випадку можна записати формулу (1.5):'

</p>

<div className="formula-container">

<Formula equation=" $t_{ga_K} = \frac{L_K}{2R}$ " />

(1.5)

</div>

<p>

{isEnglish

? 'Using the sine table (for small a , $tga = \sin a$), the value of $\sin a$ is found from the experiments and used in formula (1.6). The energy of the laser radiation quantum is calculated by formula (1.6):'

: 'За знайденим з дослідів tga за допомогою таблиці синусів (при малих a $tga = \sin a$) знаходиться значення $\sin a$, яке використовується у формулі (1.6). Енергія кванта лазерного випромінювання обчислюється за формулою (1.6):'

</p>

<div className="formula-container">

<Formula equation=" $\varepsilon = h\nu = \frac{hc}{\lambda}$ " />

(1.6)

</div>

<p>

{isEnglish

? 'where h is Planck`s constant; c is the speed of light in a vacuum.'

: 'де h – постійна Планка; c – швидкість світла у вакуумі.'

</p>

<p>

{isEnglish

? 'The installation scheme is shown in Figure 1.4.'

: 'Схема установки зображена на рисунку 1.4.'

</p>

<div className="image-container">


```

    <p className="image-caption">{isEnglish ? 'Figure 1.4 - Installation scheme' : 'Рисунок 1.4 - Схема
установки'}</p>
  </div>

```

```

    <p>
      {isEnglish
        ? 'The setup is mounted on an optical bench 1. A diffraction grating 3 is installed near the laser exit window 2.
The resulting diffraction pattern is observed on a screen 4, which has a millimeter scale. The distance between the
screen and the diffraction grating is measured using a ruler 5.'
        : 'Установка змонтована на оптичній лаві 1. Поблизу вихідного вікна 2 лазера встановлюється
дифракційна решітка 3. Одержана дифракційна картина спостерігається на екрані 4, який має міліметрову
шкалу. Вістань між екраном і дифракційною решіткою вимірюється за допомогою лінійки 5.'}
    </p>

```

```

    <h3>
      {isEnglish
        ? 'Read the theoretical information carefully. When you are ready to proceed to the next part of the test, click
the "Go to the test" button. Good luck!'
        : 'Ретельно ознайомтесь з теоретичними відомостями. Коли будете готові перейти до наступної
частини роботи, натисніть кнопку "Перейти до тестування. Успіхів!""}
    </h3>

```

```

    <div className="buttontheory-container">
      <Link to="/testing">
        <button className="theory-button">
          {isEnglish ? 'Go to testing' : 'Перейти до тестування'}
        </button>
      </Link>
    </div>
  </div>
);
};

```

```
export default TheoreticalPart;
```

App.jsx

```

import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
import { LanguageProvider } from './providers/LanguageContext';
import { DistanceProvider } from './providers/DistanceContext';
import { TurnedProvider } from './providers/TurnedContext';
import React, { useContext } from 'react';

```

```

import Header from './components/Header/Header';
import LanguageButton from './components/LanguageButton/LanguageButton';
import MainPart from './components/MainPart/MainPart';
import TheoreticalPart from './components/TheoreticalPart/TheoreticalPart';
import TestingPart from './components/TestingPart/TestingPart';
import LanguageContext from './providers/LanguageContext';

import './App.css';

function HomePage() {
  const { isEnglish } = useContext(LanguageContext);

  return (
    <div className="home-page">
      <div className="info-container">
        <div className="info-block">
          <h2><b>{isEnglish ? 'Purpose of the Work' : 'Мета роботи'}</b></h2>
          <p>
            {isEnglish ? '1. Study the structure and principle of operation of an optical quantum generator - a gas laser' :
'1. Вивчити будову і принцип дії оптичного квантового генератора - газового лазера'}
          </p>
          <p>
            {isEnglish ? '2. Learn how to use a gas laser, determine the wavelength and energy of a laser quantum using
a diffraction grating' : '2. Навчитися користуватися газовим лазером, визначити довжину хвилі та енергію
кванта лазерного випромінювання за допомогою дифракційної решітки'}
          </p>
          <h2><b>{isEnglish ? 'Equipment and Materials' : 'Обладнання і матеріали'}</b></h2>
          <p>
            {isEnglish ? '1. Gas helium-neon laser type LG-209 with power supply unit' : '1. Газовий гелій-неоновий
лазер типу ЛГ-209 з блоком живлення'}
          </p>
          <p>
            {isEnglish ? '2. Diffraction grating' : '2. Дифракційна решітка'}
          </p>
          <p>
            {isEnglish ? '3. Optical bench with measuring ruler' : '3. Оптична лава з вимірювальною лінійкою'}
          </p>
          <p>
            {isEnglish ? '4. Screen with a millimeter scale' : '4. Екран з міліметровою шкалою'}
          </p>
        </div>
      </div>
    </div>
  );
}

```

```

</div>

<div className="buttons">
  <Link to="/theory">
    <button className="home-page-button">{isEnglish ? 'Theoretical information' : 'Теоретичні
відомості'}</button>
  </Link>
  <Link to="/testing">
    <button className="home-page-button">{isEnglish ? 'Testing' : 'Тестування'}</button>
  </Link>
  <Link to="/practice">
    <button className="home-page-button">{isEnglish ? 'Practical Part' : 'Практична Частина'}</button>
  </Link>
</div>
</div>
);
}

function App() {
  return (
    <LanguageProvider>
      <TurnedProvider>
        <DistanceProvider>
          <Router>
            <div className="App">
              <Header />
              <LanguageButton />
              <Routes>
                <Route path="/" element={<HomePage />} />
                <Route path="/theory" element={<TheoreticalPart />} />
                <Route path="/testing" element={<TestingPart />} />
                <Route path="/practice" element={<MainPart />} />
              </Routes>
            </div>
          </Router>
        </DistanceProvider>
      </TurnedProvider>
    </LanguageProvider>
  );
}

export default App;

```

Додаток Д
(Обов'язковий)

Протокол перевірки магістерської кваліфікаційної роботи на наявність
текстових запозичень

Назва роботи: «Розробка програмного забезпечення емулятора фізичних процесів для лабораторних робіт з фізики для студентів Вінницького національного медичного університету»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра автоматизації та інтелектуальних інформаційних технологій

Показники звіту подібності Turnitin

Оригінальність 90 %

Схожість 10 %

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____
(підпис)

Роман МАСЛІЙ

Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи

(підпис)

Сергій МИРОНЮК

Керівник роботи

(підпис)

Ярослав КУЛИК