

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА


на тему:

Кросплатформенна інформаційна система абітурієнта з використанням семантичного пошуку

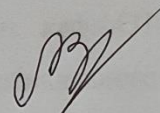
08-23.БДР.002.00.000 ПЗ

ПОЯСНЮВАЛЬНА ЗАПИСКА

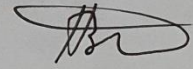
Виконав студент 4 курсу, групи 1КІ-186
спеціальності 123 - Комп'ютерна інженерія

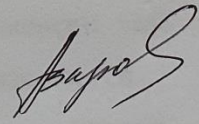
Бажан В.В. 

Керівник к.т.н., доц. каф. ОТ

Крупельницький Л.В. 

Резензент д.т.н., проф. каф. ЗІ

Лукічов В.В. 

Допущено до захисту
д.т.н., проф. Азаров О.Д. 

" 20 " червня 2022 р.

ВНТУ 2022

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

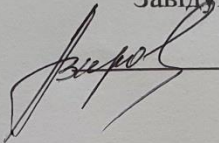
Кафедра обчислювальної техніки

Освітньо - кваліфікаційний рівень бакалавр

Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ, д.т.н., проф.


Азаров О.Д.

«10» лютого 2022 року

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Бажану Віталію Вікторовичу

- 1 Тема роботи «Кросплатформенна інформаційна система з використанням семантичного пошуку», керівник роботи к.т.н., доц. каф. ОТ Крупельницький Леонід Віталійович, затверджені наказом вищого навчального закладу від «24» березня 2022 року № 66
- 2 Строк подання студентом роботи 20.06.2022р.
- 3 Вихідні дані до роботи кросплатформенна інформаційно - пошукова система вибору ІТ - спеціальностей для абітурієнтів факультету інформаційних технологій та комп'ютерної інженерії, задачі інформаційно - пошукової системи, вимоги до функціональних можливостей.
- 4 Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Вибір платформи та інструментів розробки та техніко - організаційне обґрунтування проєкту. Проєктування та розробка складових

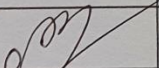
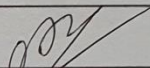
інформаційно - пошукової системи. Огляд основних функцій та можливостей системи.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6 Мапа додатку, клієнт серверна архітектура та інфраструктура додатку, JavaScript код основних компонентів та SCSS код стилів. Детальна схема алгоритму роботи застосунку.

7 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 - Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1 - 3	Крупельницький Л.В., к.т.н доцент		

8 Дата видачі завдання « 24 » березня 2022 р.

9 Календарний план виконання БДР приведені в таблиці 2.

Таблиця 2 - Календарний план

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Постановка задачі роботи	11.09.2021	Виконано
2	Огляд існуючих методів та засобів розробки інформаційно- пошукової системи	11.09.2021 - 29.09.2021	Виконано
3	Аналіз та вибір технологій розробки Web - застосунку	30.09.2021 - 19.10.2021	Виконано
4	Розробка складових інформаційно - пошукової системи вибору спеціальностей для абітурієнтів	20.10.2021 - 07.11.2021	Виконано

Продовження таблиці 2.

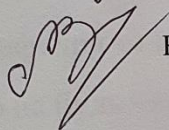
5	Проектування UI/UX дизайну.	08.11.2021 - 26.11.2021	Виконано
6	Налаштування Web - додатку.	27.11.2021 - 15.12.2021	Виконано
7	Оформлення пояснювальної записки та ілюстративного матеріалу	15.12.2022 - 02.01.2022	Виконано
8	Аналіз виконання роботи, висновки, додатки	04.01.2022 - 14.01.2022	Виконано
9	Перевірка якості виконання бакалаврської роботи та усунення недоліків	15.01.2022	Виконано

Студент



Бажан В.В.

Керівник роботи



Крупельницький Л.В.

АНОТАЦІЯ

Бажан В.В. Кроссплатформенна інформаційна система абітурієнта з використанням семантичного пошуку зі спеціальності 123 — Комп'ютерна Інженерія, Вінниця: ВНТУ, 2022. Пояснювальна записка містить 59 сторінок, 16 рисунків та 29 посилань.

Бакалаврська дипломна робота присвячена розробці інформаційній - пошуковій системі вибору ІТ — спеціальностей для абітурієнтів факультету інформаційних технологій та комп'ютерної інженерії.

Проаналізовано та обрано методи, мову програмування, фреймворки та бібліотеки необхідні для розробки застосунку. Дана система реалізована використовуючи мову програмування JavaScript та фреймворки React, Redux та Express.

Пошукова система дозволяє абітурієнтам ефективно знаходити всю необхідну інформацію про кафедри та спеціальності. Додаток є кроссплатформним, що дозволяє відкривати його на будь - якому пристрої. Розроблено пошуку систему професій по ключових словах.

ABSTRACT

Bazhan V.V. Cross-platform information system of the entrant with the use of semantic search in the specialty 123 — Computer Engineering, Vinnytsia: VNTU, 2022. The explanatory note contains 59 pages, 16 figures and 29 references.

The bachelor's thesis is devoted to the development of information - search system of choice of IT — specialties for entrants of the Faculty of Information Technology and Computer Engineering.

Methods, programming language, frameworks and libraries needed for application development are analyzed and selected. This system is implemented using the JavaScript programming language and the React, Redux and Express frameworks.

The search engine allows entrants to effectively find all the necessary information about departments and specialties. The application is cross - platform, which allows you to open it on any device. Developed a search system of professions by keywords.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД ТА АНАЛІЗ СЕМАНТИЧНОГО ПОШУКУ В ІНФОРМАЦІЙНИХ СИСТЕМАХ	9
1.1 Актуальність розробки проекту	9
1.2 Аналіз інформаційно - пошукової системи сайту.....	10
1.3 Вибір платформи та мови програмування.....	11
2 СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	15
2.1 Функціональний аналіз.....	15
2.2 Створення прототипів і дизайн додатків.....	16
2.3 Реалізація логіки програми.....	20
2.4 Структура проекту.....	20
2.5 Розміщення сайту в мережі.....	22
3 ОГЛЯД ОСНОВНИХ ФУНКЦІЙ І МОЖЛИВОСТЕЙ СИСТЕМИ	24
3.1 Схема застосування.....	25
3.2 Опис інформаційно - пошукової системи.....	26
3.3 Кросплатформена мобільна версія.....	32
3.4 Обґрунтування вибору програмного інструменту.....	34
ВИСНОВКИ	35
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	36
ДОДАТОК А. Технічне завдання	38
ДОДАТОК Б. Алгоритм роботи застосунку	41
ДОДАТОК В. Лістинг основних компонентів програми	50
ДОДАТОК Г. Лістинг загальних стилів додатку	49
ДОДАТОК Д. Лістинг загальних стилів додатку	55
ДОДАТОК Е. Лістинг загальних стилів додатку	55
ДОДАТОК Ж. Перевірка кваліфікаційної роботи на наявність текстових запозичень	58

08— 23.КБДР.002.00.000 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата				
		Бажан В.В.			Кросплатформенна інформаційна система абітурієнта з використанням семантичного пошуку.	Літ.	Аркуш	Аркушів
		Крупельницький Л.В.				59		
		Лукічов В. В.			ПОЯСНОВАЛЬНА ЗАПИСКА	ВНТУ, гр. 1КІ-18Б		
		Швець С. І.						
		Азаров О.Д						

ВСТУП

Розробка клієнт - серверних застосунків займає вагомую роль у процесі формування комп'ютерного інженера. Знання як створити правильне API, робити запити на сервер, обробляти та відсилати назад до клієнта є ключовими знаннями для створення сучасних веб- додатків.

Ефективно спроектований додаток дозволяє значно підвищити продуктивність та зменшити час пошуку інформації необхідної для вступу.

Актуальність проблеми полягає в необхідності розробки для абітурієнтів факультету інформаційних технологій та комп'ютерної інженерії єдиної системи з зручною навігацією та ефективною взаємодією, тощо.

Об'єктом дослідження є процес правильного вибору спеціальності.

Предметом дослідження є методи та алгоритми розробки веб застосунків з використанням останніх фреймворків та бібліотек.

Метою роботи є розробка методів і програмних засобів інформаційно - пошукової системи вибору спеціальностей для абітурієнтів.

Для досягнення поставленої мети необхідно виконати такі задачі:

- проаналізувати методи семантичного пошуку в інформаційних системах;
- вибрати платформу та мову програмування;
- створити пошукову систему абітурієнтів для вибор професій на основі освітніх програм факультету.

Публікація за темою роботи Л. В. Крупельницький, В.В. Бажан Кросплатформенна інформаційна абітурієнта з використанням семантичного пошуку ФІТКІ [Електронний ресурс] / Л. В. Крупельницький В.В. Бажан. - 2022. - Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12256/0> [1].

1 ОГЛЯД ТА АНАЛІЗ СЕМАНТИЧНОГО ПОШУКУ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

1.1 Актуальність розробки проекту

Суть семантичного пошуку полягає у знаходженні документів за їх змістом. Для здійснення такого типу пошуку необхідні наступні умови:

- переклад змісту документів і запитів з природної мови на інформаційно пошукову мову для створення пошукових образів документу і запиту;
- створення пошукового опису, в якому вказується додаткова умова пошуку.

Принципова різниця між адресним та семантичним пошуками полягає у тому, що при адресному пошуку документ розглядається як об'єкт з точки зору форми, а при семантичному пошуку — з точки зору змісту. При семантичному пошуку знаходиться множина документів без зазначення адрес. Це є принциповою відмінністю каталогів і картотек. Бібліотека — це є зібрання бібліографічних записів без вказування адрес.

Інформаційний пошук у Web - середовищі пов'язаний з появою та розвитком Інтернету сприяли розширенню поняття пошуку та появі більш специфічного поняття Web - пошуку. Оскільки в контексті Web фактори взаємодії людини з комп'ютером та когнітивні аспекти грають найважливішу роль, корисно деталізувати цю модель, як показано на (рис.1).

Завдання семантичного пошуку полягає в підвищенні точності пошукової видачі, завдяки розумінню намірів користувачів через контекст. За допомогою зіставлення концепту, синонімів, алгоритмів, написаних природною мовою, семантичний пошук дозволяє трансформувати структуровану і неструктуровану інформацію в інтуїтивно зрозумілу і гнучку базу даних. Такий підхід допомагає більш точно визначити наміри користувачів, підвищує здатність системи витягати правильні відповіді, надавати користувачам більш персоналізовані результати.

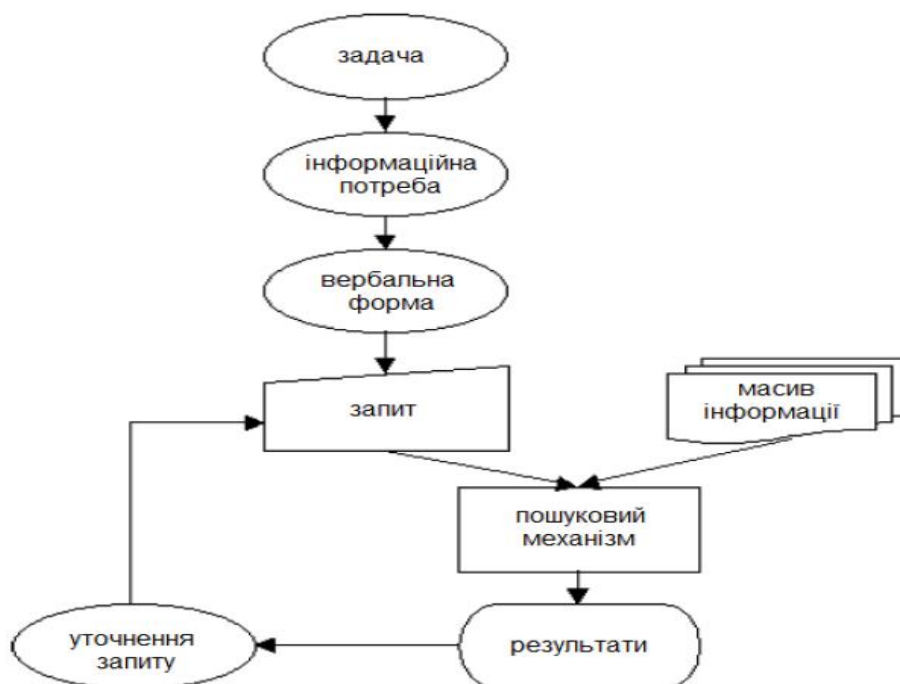


Рисунок 1— Класична модель інформаційного пошуку, поширена на Інтернет - мережу (Web)

Граф знань (технологія Google) - парадигма семантичного пошуку.

1.2 Аналіз інформаційно - пошукової системи сайту

Пошукова система являє собою веб - застосунок з такими категоріями як «Список спеціальностей», «Пошук». В проекті було реалізовано авторизацію за основі токенів. JSON Web Token (JWT) — містить три блоки, між якими ставлять крапку: заголовок (header), набір полів (payload) і сигнатуру. Також в проекті було створенно систему обробки можливії помилок [2]. Обробка винятків є критичним компонентом кожного програмного додатка. Останнє, що ви хочете, щоб ваші користувачі бачили дивні помилки, або, що ще гірше, щоб ваш додаток був зупинений. Завдяки використанню фреймворку React, проект було запроваджено в single - page

application, також відомий як веб - застосунок чи веб - сайт, який вміщується на одній сторінці з метою забезпечити користувачу досвід близький до користування настільною програмою. В моєму застосунку весь необхідний код HTML, JavaScript, та CSS — завантажується разом зі сторінкою, або динамічно довантажується за потребою, зазвичай у відповідь на дії користувача. Сторінка не оновлюється і не перенаправляє користувача до іншої сторінки у процесі роботи з нею [3].

1.3 Вибір платформи та мови програмування

Даний програмний продукт розроблено для подальшого створення на його базі Факультету інформаційних технологій та комп'ютерної інженерії.

Актуальність проекту є високою та є перспективною для реалізації інших факультетів університету та розроблення додаткових можливостей для пошуку професії [4].

При розробці даного проекту було взято мову програмування Javascript.

JavaScript — призначений для написання сценаріїв для активних HTML сторінок. Мова JavaScript не має жодного відношення до мови Java. Java розроблено фірмою SUN. JavaScript — фірмою Netscape Communication Corporation. Початкова назва — LiveScript. Після завоювання мовою Java всесвітньої популярності LiveScript з комерційних міркувань перейменували на JavaScript [5].

JavaScript не призначено для створення автономних програм. Програма JavaScript вбудовується безпосередньо у вихідний текст HTML документа і інтерпретується браузером у міру завантаження цього документа. За допомогою JavaScript можна динамічно змінювати текст завантаженого HTML документа та реагувати на події, пов'язані з діями відвідувача або змінами стану документа чи вікна.

Важлива особливість JavaScript — об'єктна орієнтованість. Програмістові доступні численні об'єкти, такі як документи, гіперпосилання,

форми, кадри і т.д. Об'єкти характеризуються описовою інформацією (властивостями) та можливими діями (методами). Мова JavaScript використовується для:

- написання сценаріїв вебсторінок для надання їм інтерактивності;
- створення односторінкових та прогресивних вебзастосунків (React, AngularJS, Vue.js);
- програмування на боці сервера (Node.js(Express.js));
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладних програмах (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF документів тощо [6].

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme [7].

Мова програмування PHP. Веб - розробка за допомогою PHP включає такі дії

- керування файлами cookie;
- збирання інформації з веб - сторінок та перетворення статичних веб- сайтів на динаміку [8].

Створення динамічних веб - сайтів PHP є одним із застосувань цієї мови сценаріїв. Іноді досить комбінувати HTML і CSS для створення статичних сайтів, які рідко змінюються. Однак, PHP може дати додатковий динамічний ефект, але динамічні елементи також можуть бути реалізовані за допомогою JavaScript Node.js. Таким чином виникає дилема вибору мови програмування

Ще один важливий аспект використання PHP — це створення динамічних веб - сайтів PHP за допомогою системи MySQL. PHP працює з низкою баз даних, але найпопулярнішою з них є MySQL [9]. У таблиці 1 наведено порівняння мов програмування JS та PHP.

Таблиця 1 — Порівняння мов програмування JS і PHP

Порівняння мов програмування JS і PHP			
PHP		Node.js	
Плюси	Мінуси	Плюси	Мінуси
Швидке серверне рішення	Мала ефективність в операціях, що інтенсивно використовують CPU	Велика кодова база	Поганий поділ відповідальності (SoC)
Одна мова на фронт - енді та бекенді	Незрілість платформи	Величезна кількість безкоштовних інструментів	Застаріла клієнт - серверна модель
Гнучкість		Кросплатформність	

Серед фреймворків на стороні клієнту було обрано React та бібліотеку контролю стану даних Redux, окрім цього було використано Redux Saga, Immer, Reselect, React Router, Axios, React Hook Form. React — це відкрита JS бібліотека для створення інтерфейсу користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб - сторінки, з якими стикаються в розробці односторінкових застосунків. Зараз цей компонент має підтримку, а також розробляється Facebook та Instagram [10].

React можна назвати:

— веб - фреймворком;

- бібліотекою функцій;
- бібліотекою JS.

Для серверної розробки було обрано платформу Node JS, база даних MongoDB та відповідні фреймворки — Express JS, Mongoose, Passport JS, JWT Handler, Axios, Multer, Nodemailer.

Node.js — платформа з відкритим кодом для виконання високопродуктивних мережеских застосунків, написаних мовою JavaScript. Засновником платформи є Раян Дал (Ryan Dahl). Якщо раніше Javascript застосовувався для обробки даних в браузері користувача, то node.js надав можливість виконувати JavaScript — скрипти на сервері та відправляти користувачеві результат їхнього виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників [11].

Node.js має наступні властивості (рисунок 1.1):

- асинхронна однопотокова модель виконання запитів;
- неблокуючий ввід/вивід;
- система модулів CommonJS;



Рисунок 1.1 — Можливості Node JS

Для керування модулями використовується пакетний менеджер npm (node package manager).

2 РОЗРОБКА КОМПОНЕНТІВ ПОШУКОВОЇ СИСТЕМИ

2.1 Функціональний аналіз

Відповідно до технічного завдання (додаток А), ця заявка повинна включати такі компоненти:

- навігаційна система;
- відомча система розподілу даних;
- системи зберігання даних;
- динамічна система візуалізації для кожного спеціальності;

Система призначена для профорієнтаційної роботи вчителів і використовується як професійний інструмент профорієнтації на вступних заходах ВНТУ.

Навігаційна система повинна мати можливість легко переходити до будь - якої частини веб - сайту. Кожна сторінка має меню, яке дозволяє користувачам отримати доступ до будь - якої сторінки.

Система повинна мати такий інформаційний зміст:

- окрема частина для магістратури (у тому числі короткострокової), магістратури;
- назву та коротку характеристику кожної професії (121, 122, 123, 125) та її освітню програму (123, 125); — кількість прийомів на програму (на сайті внту);
- середній бал бюджету та контракту за останні 2 роки;
- посилання на сайти факультету та кафедр кожної професії (штат, новини тощо);
- посилання на сторінки кафедри в соціальних мережах;
- текстовий і графічний матеріал рекламного фахівця;
- часті запитання (FAQ);
- контактні дані викладачів, відповідальних за прийом;

— перелік предметів, які викладаються у спрощеному форматі курсу у 2022 році, бажано з переходом на предметну програму на сайті кафедри.

Доступ і зручність використання системи мають забезпечувати настільні комп'ютери під керуванням Windows (xp, 7, 8, 10) Mac OS та мобільні пристрої з поточними версіями Android, iOS.

Тому для мене як розробника це означає, що веб - додаток має бути кросплатформним. Це дозволить охопити якнайширшу аудиторію. Наприклад, не всі можуть увійти з комп'ютера, користувачі повинні мати можливість входити в програму з телефону або планшета, а додаток має мати зручне розширення, яке об'єднує всі елементи системи. Додаток також має бути доступним, тобто якщо у користувача немає миші, він зможе керувати програмою лише з клавіатури.

Кросплатформенність — властивість програмного забезпечення працювати на кількох програмних системах (включаючи операційні системи) або апаратних платформах; технологія, яка дозволяє реалізувати цю властивість. Кросплатформенність може значно знизити витрати на розробку нового програмного забезпечення або адаптацію існуючого програмного забезпечення [12].

Залежно від реалізації він поділяється на кросплатформний на рівні мови програмування (інструментів для таких мов: компіляторів і комунікаційних редакторів), середовища виконання, операційної системи та апаратної платформи.

2.2 Створення прототипів і дизайн додатків

При розробці програмного забезпечення дуже важливо дотримуватися всіх принципів дизайну UI/UX . Усі прототипи та ескізи програмних продуктів створюються в додатку Figma (рис. 2.1). Figma — векторний

онлайн — сервіс для розробки інтерфейсу та прототипування з можливістю організації співпраці, розроблений однойменною компанією.

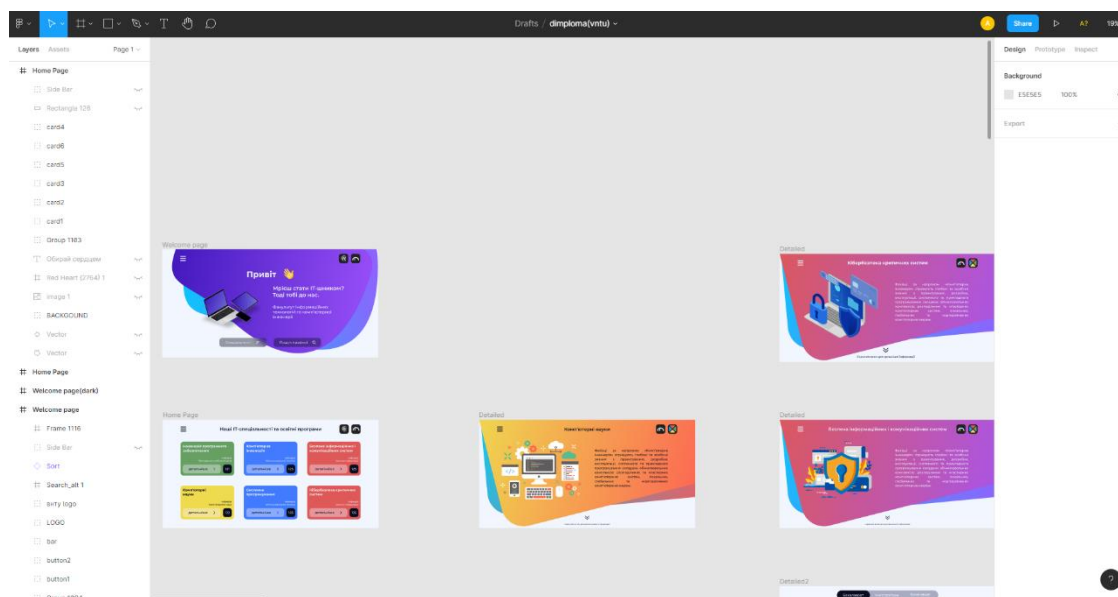


Рисунок 2.1 — Розробка шаблонів сайту в середовищі Figma

Працює в двох форматах: у браузері та як клієнтська програма на робочому столі користувача. Зберігайте онлайн — версії файлів, які використовували користувачі. Він безкоштовний для окремих користувачів і платний для професійних команд [13].

2.3 Реалізація логіки програми

Проект розроблено з використанням інфраструктури React і всіх супутніх бібліотек і фреймворків. Розробка програм на React особливо корисна в проектах, які передбачають багато взаємодії з користувачем. Таким продуктам не потрібно перезавантажувати сторінки чи отримувати й надсилати дані, ви можете завантажувати дані клієнтам для швидкої перевірки та презентації. Отже, замість того, щоб дивитися на «попередній завантажувач», «скелет» або, якщо пощастить, на порожній екран, відвідувач може отримати миттєві результати.

React — development — це опис того, що потрібно відобразити на

веб - сторінці. Це означає значне зменшення кількості коду шаблону.

Важливою особливістю React є використання JSX. Це розширення синтаксису JavaScript, яке полегшує опис інтерфейсів. JSX подібний до HTML, але все одно це JavaScript.

Для створення бази програми React ми будемо використовувати пакет Facebook «create - react - app». Це найпопулярніший спосіб налаштувати робоче середовище. Завдяки «create - react - app» програмісти отримують багато необхідних інструментів, не вибираючи їх самостійно.

Щоб встановити «create - react - app» глобально, скористайтеся такою командою: «npm i - g create - react - app».

Потім, щоб створити шаблон програми, виконайте таку команду: «create - react - app react - intro».

На цьому попередня підготовка закінчена. Для запуску програми потрібно виконати такі команди: «cd react - intro», «npm start».

Тут ми заходимо в папку проекту і запускаємо сервер розробки, який дозволяє вам відкрити нашу програму React через ваш браузер «<http://localhost:3000/>». У цьому проекті кожна частина заснована на компонентах, що є ще однією ключовою особливістю бібліотеки React.

Усі основні частини програми розбиті на компоненти. Кожен компонент повертає частину інтерфейсу користувача та його стан.

Комбінуючи ці компоненти, програмісти створюють повний інтерфейс веб - додатків. Такий підхід дозволяє повторно використовувати ці самі компоненти кілька разів.

Компоненти "Category" реалізовані за допомогою техніки зберігання, гачка "useState". Хук useState() приймає аргумент як початковий стан, який може бути представлений у вигляді значення (рядок, число, масив, об'єкт тощо) або функції. Після виклику функції useState повертає масив із двома елементами. Перший елемент — це початковий стан, який ми передаємо, а

другий — функція, яка змінює цей стан, коли ми його викликаємо та передаємо нове значення.

В даній роботі створено пошукову систему на основі семантичного пошуку. Для реалізацію був розроблений алгоритм схема якого представлена на (рис. 2.2).

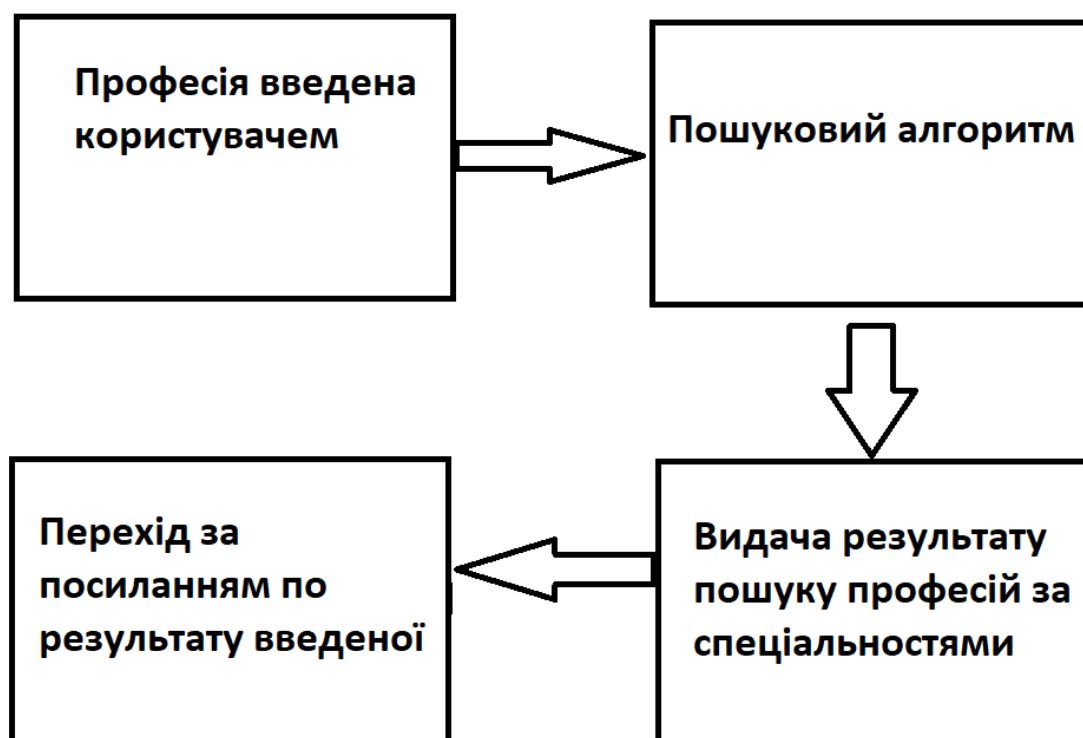


Рисунок 2.2 — Діаграма послідовності пошуку інформації

Суть алгоритму полягає в наступних діях:

- створення бази в програмному коді, яка буде відповіді за конкретні професії певної спеціальності;
- заповнення бази професіями, які відповідають спеціальностям;
- пошук та підбір спеціальності факультету на основі введеної спеціальності.

Приклад коду наведено на (рис. 2.3).

```

1005     if(String(haystack).indexOf(needle)>-1){
1006         return true;
1007     }else{
1008         return false;
1009     }
1010 }
1011 </script>
1012 <p>
1013 <span style="display:block;">
1014 <input type="text" class="welcome__finder" name="search" placeholder="Пошук..." href="/'+sUrl[i]+' " style="padding-left:28px; wi
1015 sName=[];
1016 sUrl=[];
1017 arrToSearch.forEach((element) => {
1018     //console.log( element['keywords'] );
1019     if(my_strippos(element['keywords'].toLowerCase(), this.value.toLowerCase())==true){
1020         sName[count]=element['name'];
1021         sUrl[count]=element['id'];
1022         count++;
1023     }
1024 });
1025 text='';
1026 for(i=0; i<sName.length; i++){
1027     text+ '<div><a style=\'padding:15px; margin:10px; color:#fff !important; width:100%; text-decoration:none; color:#333;\' href=\'
1028

```

Рисунок 2.3 — Програмна реалізація пошуково алгоритму

```

src > components > toImport.jsx
1
2 <div className="developing">
3 <script>
4     const arrToSearch = [
5     {
6         id: 'software-engineering',
7         name: 'Інженерія програмного забезпечення',
8         faculty: 'Програмної інженерії',
9         pictureName: 'SoftwareEngineering',
10        keywords:"інженер, компютери, інженерія, програми, ПЗ,розробник баз даних, web-програміст,seo-фахівець, верстальник, модератор, front-
11        cipher: 121,
12        colorName: 'green',
13        mainColor: '#ffffff',
14        colorSchemePrimary: {
15            color1: '#689F63',
16            color2: '#3CE02D',
17        },
18        colorSchemeSecondary: {
19            color1: '#00CFFD',
20            color2: '#79FF63',
21        },
22        description:
23        'Студенти спеціальності 121 Інженерія програмного забезпечення поглиблено вивчають процеси промислової розробки програмного забезпеч
24        undergraduate: [

```

Рисунок 2.4 — Реалізація компонентів сторінки спеціальності

Детальну реалізацію коду продемонстровано у ДОДАТОК В.

2.4 Структура проекту

Інформаційно - пошукова система має таке файлове дерево (рис. 2.5).

— package.json (файл package.json містить інформацію про вашу програму: ім'я, версію, залежності тощо. Будь - який каталог, що містить цей файл, інтерпретується як пакет Node.js);

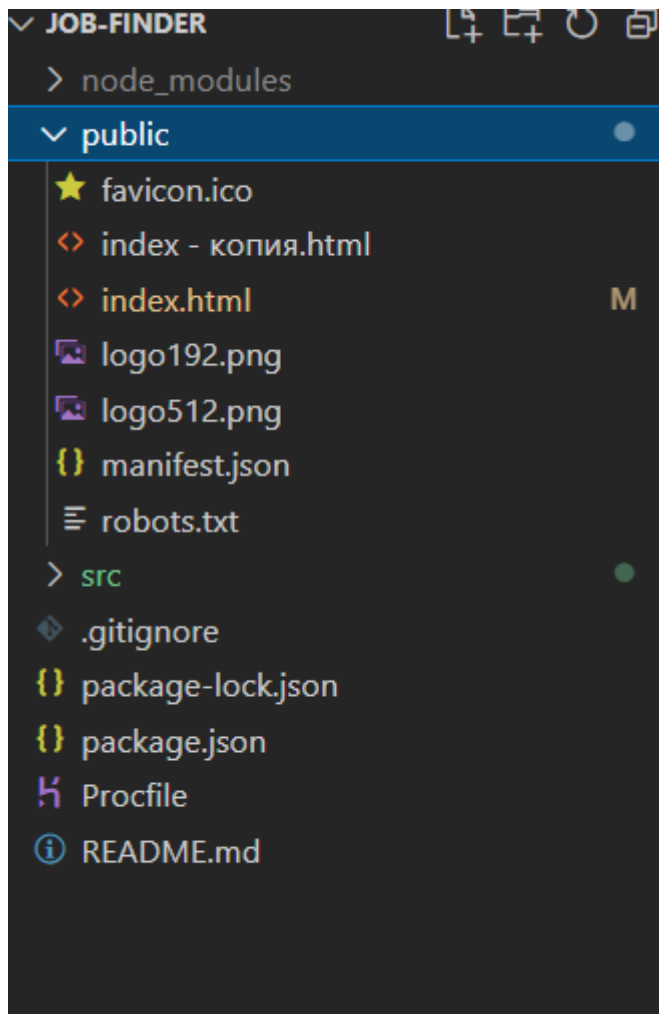


Рисунок 2.5 — Файли проекту

- `index.tsx` (точка входу в реакт проект);
- `App.tsx` (кореневий компонент запуску програми, тут знаходяться репозиторії даних підключеного відділу);
- `assets`(Активи) (папки, де зберігаються всі графічні файли, такі як значки, векторна графіка та растрові зображення);
- `component`(Компоненти) (ця папка містить усі компоненти, які були розбиті для подальшого повторного використання);
- `helpers`(Помічники) (у цій папці зберігаються всі допоміжні функції та так звані «помічники», наприклад, для редагування дат або валют);

- `pages`(сторінки) (цей каталог містить усі «компоненти сторінки», тобто компонент у цьому каталозі відображає фізичну сторінку, де ті самі компоненти зібрані для повторного використання в папці «компоненти»);
- `redux` (для створення правильної архітектури програми реалізовано окреме сховище даних, ця інформація зберігається в папці `redux`);
- `scss` (тут зберігаються всі можливі стилі та анімації, необхідні для дизайну програми);

2.5 З'єднання клієнт сервер та інтерфейси програмування прикладних програм

Якщо ви проводили час із програмним забезпеченням або онлайн — інструментами, ви, напевно, чули термін «API». API означає «Інтерфейс програмного забезпечення». Для розробників програмного забезпечення існує багато API, які дозволяють інтегрувати інформацію з інших програм у власні програми.

API — це системи інструментів і ресурсів у програмі, які дозволяють розробникам створювати програмні продукти, які взаємодіють з іншими службами. API — це те, що дозволяє двом окремим додаткам взаємодіяти один з одним. API дозволяють користувачам взаємодіяти з іншими службами або сайтами, не залишаючи вашого сайту. API важливі, оскільки без них було б важко зрозуміти синтаксис усіх програм і служб, з якими взаємодіє ваш інструмент. API пояснює, як називається кожен об'єкт і що з ним можна робити.

Є два типи запитів:

- запити зроблені клієнтською частиною до серверної частини (тобто наш API);
- запити, зроблені до клієнтських маршрутів.

Нижче наведена схема клієнтської та серверної частин, пов'язаних з базою даних і показує загальну архітектуру програми (рис. 2.6)



Рисунок 2.6 — Сценарій архітектури клієнт – сервер для програми

Клієнт (React + Redux) надсилає запит, наприклад, учасник хоче знайти сторінку зі списком усіх професій. Надсилається наступний запит — `/specialities` спеціальності (Redux Saga + Axios рис. 2.7)

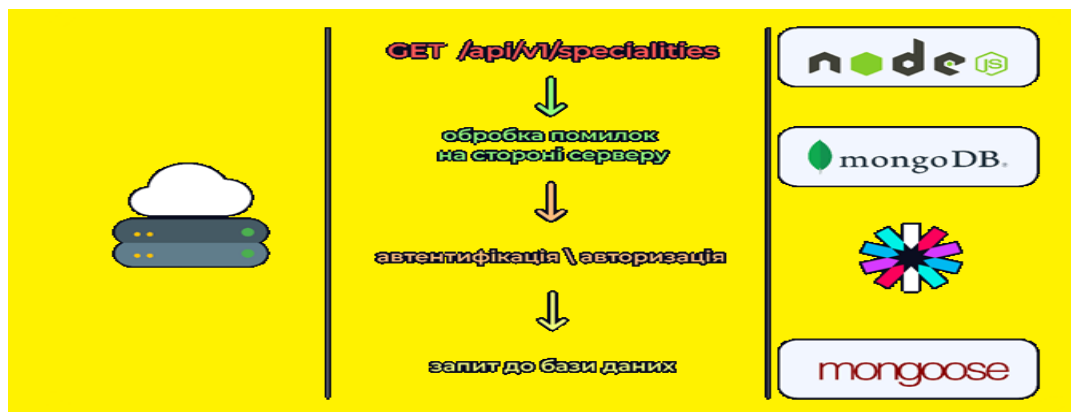


Рисунок 2.7 — Сценарій архітектури клієнт - сервер для програми, обробки подій

Клієнтський маршрутизатор обробляє цей запит, проходячи всі необхідні перевірки. Клієнт знаходить відповідний ресурс (кінцеву точку) сервера в таблиці маршрутизації — `/api/v1/specialities`, а потім клієнтська частина робить запит RESTful до цього ресурсу.

Серверна частина отримує цей запит, організовує всю необхідну аутентифікацію, авторизацію та обробку помилок, а потім робить запит до бази даних (Node JS, Express JS, Mongo DB, Mongoose, JWT) (рисунок 2.8).

Рисун
ок 2.8
—
Схем
а



клієнт - серверної архітектури додатку, серверна перевірка, автентифікація та авторизація

Після видалення всіх необхідних даних за запитом, сервер генерує відповідь у форматі, який вимагає клієнт JSON (RESTful API).

Після того, як клієнт отримує відповідь від сервера, він обробляє помилку, поміщає дані в репозиторій (Redux) і починає відтворювати сторінку, стилі та анімацію (React + Framer Motion) (рис. 2.9).

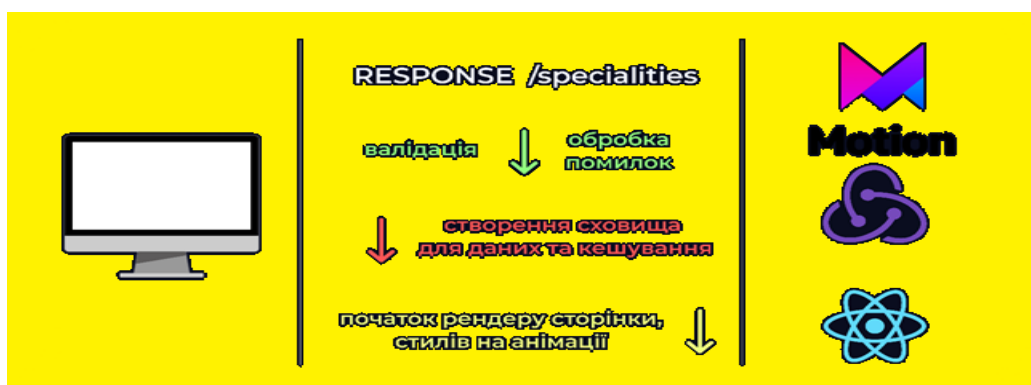


Рисунок. 2.9 — Схеми архітектури клієнт- сервер для додатків, візуалізації сторінок, стилів та анімації.

3 ОГЛЯД ОСНОВНИХ ФУНКЦІЙ І МОЖЛИВОСТЕЙ СИСТЕМИ

3.1 Схема застосування

Після розробки цієї інформаційної пошукової системи для конкурсантів була сформована карта веб - додатку (рис. 3.1).

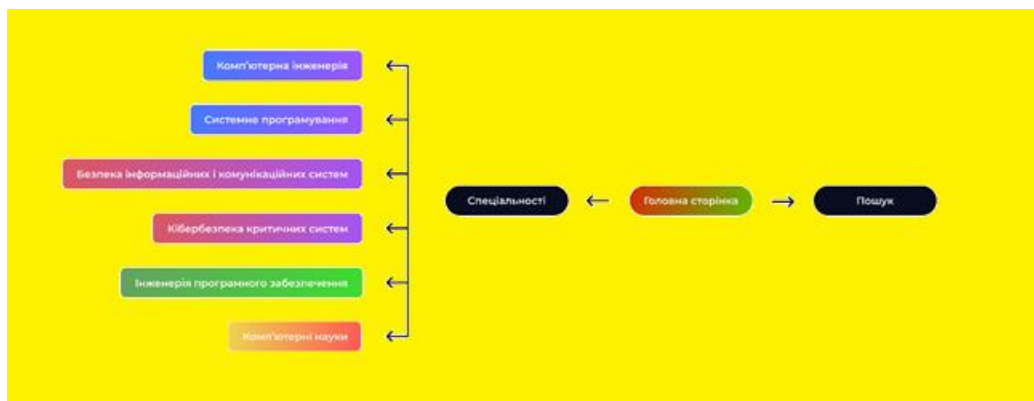


Рисунок 3.1 — Мапа додатку

Тому потенційні користувачі можуть вибрати два способи взаємодії з програмою:

- вибравши спеціальність зі списку;
- пошук за ключовим словом.

Для взаємодії між клієнтською і серверною частинами реалізовано спеціальний API за допомогою REST API.

REST — аббревіатура від Representational State Transfer, що можна перекласти як «передача стану представництва».

Це стиль проектування розподілених систем з обмеженнями. Основною абстракцією в REST є ресурс [14]. Ваш браузер (клієнт) надсилає запит GET на сервер (у цьому випадку codeguida). Сервер приймає запит і надсилає відповідь. Браузер отримує його і відображає отриману інформацію.

У моделі клієнт сервер сервер надає послугу або ресурс, який клієнт отримує, виконуючи запит. Клієнтом може бути будь - який:

- додаток для Android;

— браузер.

У нашому проекті, щоб отримати перелік спеціальностей, потрібно надіслати запит

«GET /specialities.». Ось список усіх кінцевих точок (ресурсів):

- specialities (спеціальності);
- /specialities/computer - engineering (ком'ютерна інженерія);
- /specialities/system - programming (системне програмування);
- /specialities/security - of - information - and - communication - systems (системи безпеки інформації та зв'язку);
- /specialities/cybersecurity - of - critical – systems (кібербезпека критичних систем);
- /specialities/software - engineering (програмна інженерія);
- /specialities/computer - science (комп'ютерні науки);

3.2 Опис інформаційно - пошукової системи

Домашня сторінка є передньою частиною програми, тому було докладено всіх зусиль, щоб зробити її привабливою. Використовуються принципи мінімалізму. Основне завдання цієї сторінки — розширити користувацький досвід, тобто потенційні абітурієнти можуть відразу вибрати спеціальність або відразу перейти на сторінку пошуку та здійснити пошук (рис. 3.2).

Перейшовши на сторінку з професійним списком, ми бачимо відразу 5 кольорових карток. Кожен з них має свій акцентний колір і відображає унікальну інформацію про кафедру та професію (рис. 3.3). Наприклад, спеціальність комп'ютерна інженерія має свій унікальний колір - синій, і при натисканні користувачі будуть перенаправлені на сторінку з детальною інформацією про цю спеціальність. Все це відбувається в режимі однієї сторінки, тобто коли ви переходите на цю сторінку, не відбувається зайвих перезавантажень, завантажується лише те, що запитує користувач.

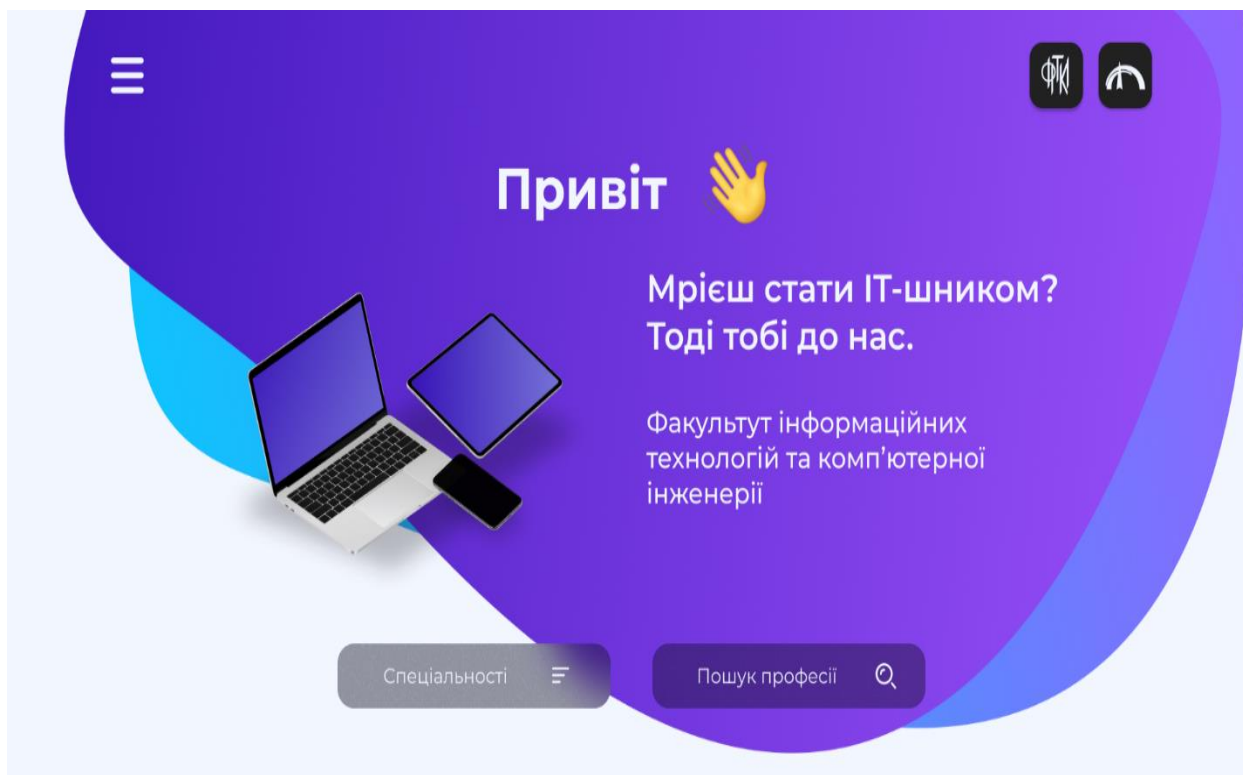


Рисунок 3.2 — Домашня сторінка додатку



Рисунок 3.3 — Сторінка спеціальностей факультета

Перейшовши на сторінку з детальною інформацією про професію, користувач відразу бачить опис та зображення, що зображують професію на (рис. 3.4).

Якщо абітурієнт хоче дізнатися більш конкретні дані, такі як плата за навчання, тривалість навчання, прохідні бали тощо, вам буде запропоновано прокрутити вниз і знайти цю інформацію (рисунок 3.5).

У верхній частині знаходиться повзунок, який дозволяє користувачеві вибрати форму навчання, яка його цікавить, знову ж таки, коли ви натискаєте повзунок, веб - сторінка не перезавантажується, а працює як рідна програма.



Рисунок 3.4 — Детальна інформація про професії, описи та зображення

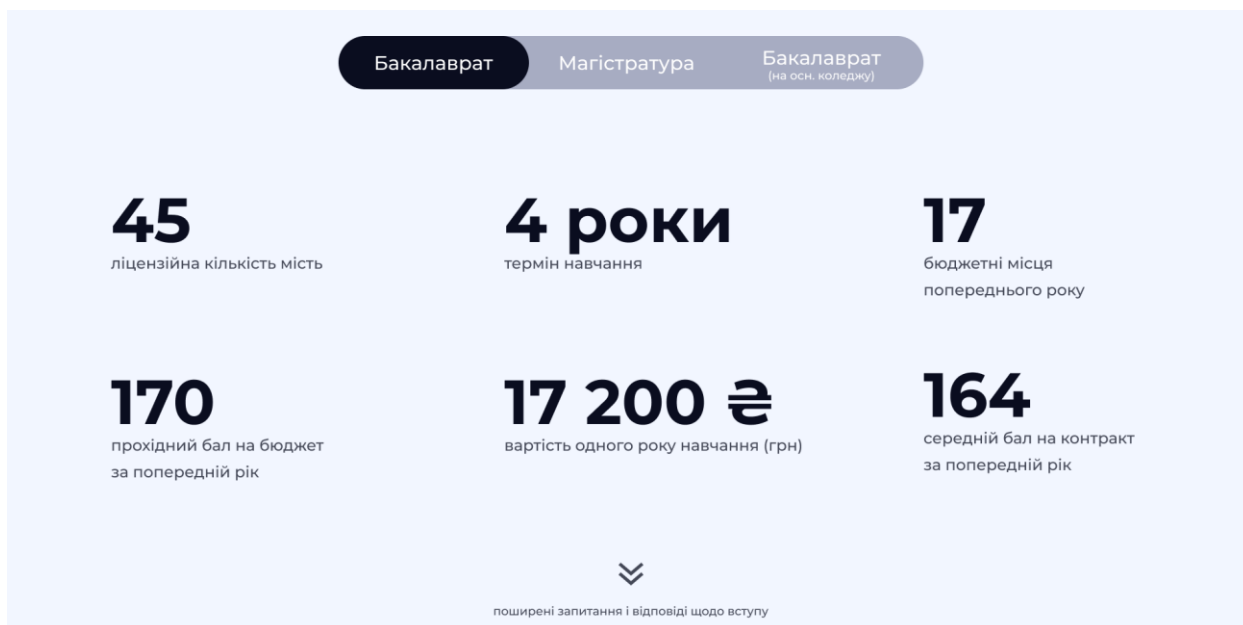


Рисунок 3.5 — Інформація для абітурієнта про навчання

Потім програма запропонує знову прокрутити вниз, щоб перейти до розділу FAQ (часті запитання та відповіді). Перейдіть на сторінку FAQ, користувач має можливість відкрити та прочитати всі запитання та відповіді про професію (рис. 3.6). Найважливішим є розділ «Пошук професії» в якому абітурієнт може за допомогою пошукових слів вибрати спеціальність на основі якої хоче отримати професію.



Рисунок 3.6 — Поширені запитання та відповіді для абітурієнта

Для зручності пошукове вікно винесено на головну сторінку, також у лівому верхньому куті є навігаційна панель (рис.3.7).

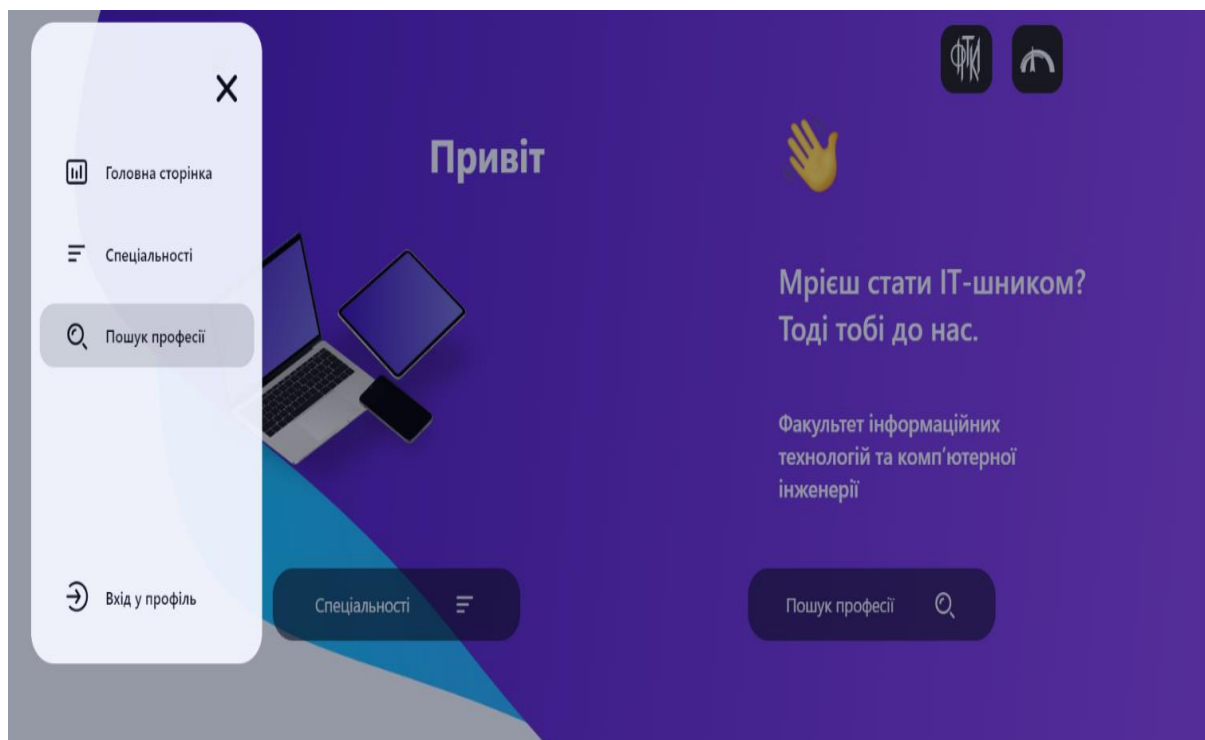


Рисунок 3.7 — Панель пошуку професії на головному екрані

Перейшовши на панель пошук професії за допомогою ключових слів можна вибрати спеціальність яка найкраще підходить для вас (рис. 3.8 - 3.9)



Рисунок 3.8 — Панель пошуку професії

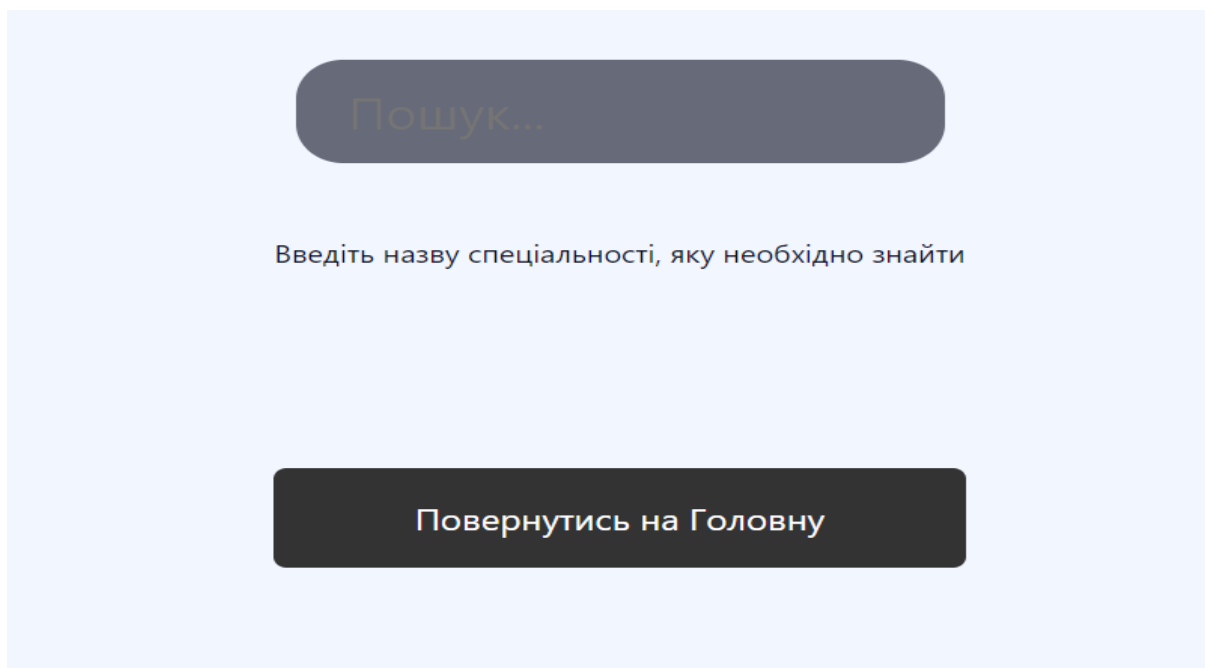


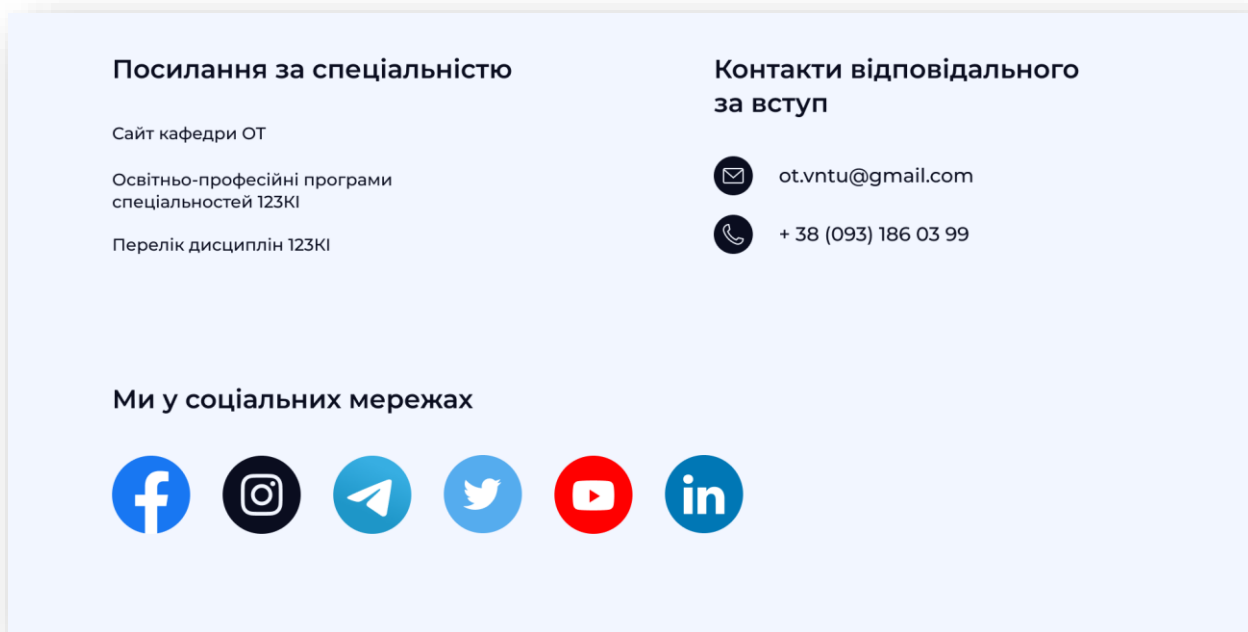
Рисунок 3.9 — Пошук професії за ключовим словом

Також слід відзначити, що клавiшi є iнтерактивними, пiсля вибору спеціальності то кліку миші можна перейти на вікно із вибраною спеціальністю та ознайомитись із інформацією для абітурієнта (рис. 3.10)



Рисунок 3.10 — Перехід на вікно вибору професії

Останній розділ — це сторінка, де можна знайти контактні дані керівника відділу прийому, силабусу, професійної освіти та професійних



програм, а також соціальної мережі кафедри (рисунок 3.11).

Рисунок 3.11 — Детальна інформація про спеціальність, контакти та соціальні мережі

3.3 Кросплатформена мобільна версія

Під час реалізації всіх функцій у рамках кросплатформного характеру програми надається та розробляється мобільна версія програми. Для реалізації мобільних версій веб - додатків використовуються так звані методики «адаптивного дизайну».

Адаптивний дизайн дозволяє налаштувати вміст сторінки відповідно до розміру вікна браузера. Тому такий сайт зручно використовувати як на великих екранах, так і на портативних пристроях. Адаптивний дизайн є «традиційним» варіантом і залишається найпопулярнішим сьогодні. З

технічної точки зору, такі сайти створюються за допомогою медіазапитів CSS (@media). Адаптивні макети мобільних додатків можна налаштувати в режимі реального часу, коли ви змінюєте ширину браузера. Цей метод використовується для розробки мобільної версії (рисунок 3.12)



Рисунок 3.12 — Мобільна версія програми.

Переваги цього методу:

- тони готових тем і шаблонів на різних cms;
- єдина url — адреса для ПК та смартфона краще з точки зору seo;
- посилання залишається незмінним, а роздача більш зручна;
- весь контент подається за посиланням;
- контент підходить для будь-якого розміру екрана;
- одна кодова база — легше підтримувати та відновлювати;
- хороша основа для рwa.

3.4 Обґрунтування вибору програмного інструменту

Вибір мови програмування, наприклад JavaScript, є дуже хорошим вибором. Ця мова програмування має величезне майбутнє та потенціал завдяки своїй популярності, великій спільноті та великій кількості фреймворків і бібліотек. Вхідний бар'єр середній, тому студенти з базовими знаннями програмування зможуть почати розвиватися. Статистика використання мови програмування розміщується на авторитетних форумах. Зростання та популярності екосистеми JavaScript продемонстрували такі цифри (рисунок 3.13) [15].



Рисунок 3.13 — Рейтинг мов програмування в 2022 році

З даних представлених на рисунку можемо зробити висновки, що мова програмування JS стає популярнішою для розробки програмних продуктів.

ВИСНОВКИ

Під час реалізації проекту бакалаврату було розроблено та впроваджено інформаційно - пошукову систему відбору ІТ — спеціальностей, з цією метою вирішено такі завдання:

аналіз сучасних інформаційно - пошукових систем, огляд аналогів і порівняння між ними, що дозволяє легко визначити аналітичні завдання та вибрати правильний напрямок дослідження інформаційно - пошукових систем;

Створено важливу клієнт - серверну інфраструктуру для програми, на якій у майбутньому програму можна поширити на всю школу і навіть на регіональний рівень;

Проаналізовано технології розробки інформаційних систем за допомогою фреймворків і бібліотек: Express, Axios, Framer React, Redux.

Тому метою даної роботи є створення інформаційно - пошукової системи для першокурсників факультету інформаційних технологій та комп'ютерної інженерії для відбору ІТ — спеціальностей, що дозволить знайти всю необхідну інформацію про вступ швидко та якісно, з більшою потужна реалізаційна взаємодія з користувачем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кроссплатформенна інформаційна система абітурента з використанням семантичного пошуку ФІТКІ [Електронний ресурс] / Л. В. Крупельницький В.В. Бажан. - 2022. - Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12256/0>
2. Что такое веб- сайт [Електронний ресурс]. Режим доступу: <https://semantica.in/blog/chto-takoe-veb-sajt.html>. Дата звернення: Квітень 29 , 2022.
3. Веб- сайт - определение и виды [Електронний ресурс]. Режим доступу: <https://www.seonews.ru/glossary/veb-sajt/> . Дата звернення: Квітень 15 , 2022.
4. Гороховський О.І. Інформаційна технологія доставки контенту у системі комп'ютеризованої підготовки спеціалістів: монографія / О.І. Гороховський, О.Д. Азаров, Т.І. Трояновська. - Вінниця: ВНТУ, 2016. - 160 с.
5. Прохоренко Н.А., Дронов В.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web- мастера. - 3- е изд., перераб. и доп. - СПб.: БХВ- Петербург, 2010. - 659 с.
6. Самоучитель и справочник по HTML, CSS, JavaScript и PHP [Електронний ресурс]. Режим доступу: <https://puzzleweb.ru/> . Дата звернення: Березень 27, 2022.
7. Скопень М.М. Комп'ютерні інформаційні технології в туризмі: навч. посіб. / М.М. Скопень. - Київ: Кондор, 2005. - 302 с.
8. FRONTEND. HTML&CSS. ОСНОВЫ [Електронний ресурс]. Режим доступу: <https://prjctr.com.ua/front-end.html>. Дата звернення: Травень 1, 2022.
9. Засоби створення сайтів [Електронний ресурс]. Режим доступу: https://pidruchniki.com/1970070547797/informatika/zasobi_stvorenniya_web-saytiv. Дата звернення: Березень 15, 2022.

10. Что такое html, css, php и js? [Электронный ресурс]. Режим доступа: https://wayup.in/blog/что-такое-html-css-php-i-js#порядок_вопросов. Дата обращения: Март 29, 2022.

11. Підходи створення інформаційних систем [Електронний ресурс]. Режим доступа: <https://studfile.net/preview/5851333/page:9/>. Дата обращения: Марта 29, 2022.

12. Какие самые важные клиентские библиотеки (HTML5/CSS/JavaScript) и почему? [Электронный ресурс]. Режим доступа: <https://www.templatemonster.com/help/ru/what-are-the-most-important-client-side-html5cssjavascript-libraries-and-why.html>. Дата обращения: Май 15, 2022.

13. What are the pros of JavaScript? [Электронный ресурс] - Режим доступа до ресурсу: https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript

14. React/Redux [Электронный ресурс] - Режим доступа до ресурсу: <https://reactjs.org/docs/getting-started.html>

15. Redux Tutorial [Электронный ресурс] - Режим доступа до ресурсу: <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>

16. Top 10 Reason: Why you should use Node.js? [Электронный ресурс] - Режим доступа до ресурсу: <https://medium.com/quick-code/top-10-reason-why-you-should-use-node-js-5926b4d1f2c1>

17. Cross-platform software [Электронный ресурс] - Режим доступа до ресурсу: https://en.wikipedia.org/wiki/Cross-platform_software

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н. О.Д. Азаров

« 10 » лютого 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської дипломної роботи

Кросплатформенна інформаційна система абітурієнта з використанням семантичного пошуку
08-23.БДР.002.00.000 ПЗ

Науковий керівник: к.т.н., доцент

Крупельницький Л. В.

студент групи 1КІ-186

Бажан В.В.

1 Підстава для виконання бакалаврської дипломної роботи (БДР)

Підставою для розробки даної бакалаврської дипломної роботи є наказ ВНТУ №66 від «26» 03 2022 року та рішення засідання кафедри обчислювальної техніки (протокол №1 від «20» 06 2022 року).

2 Вихідні дані для виконання БДР

Вихідні дані: інформація та статистика про вступ, клієнт - серверна архітектура, задачі інформаційно - пошукової системи, вимоги до функціональних можливостей.

3 Перелік задач, що повинні бути виконані

Задачі для виконання:

- аналіз, особливості та види сучасних інформаційно - пошукових систем;
- розробка пошукового алгоритму на основі семантичного пошуку
- використання мови програмування Javascript;
- функціональні можливості інформаційних систем.

4 Матеріали, що подаються до захисту БДР

До матеріалів захисту роботи додається :пояснювальна записка БДР, графічні і ілюстративні матеріали, протокол попереднього захисту БДР на кафедрі, відгук наукового керівника, відгук рецензента, анотації до БДР українською та іноземною мовами, довідка про відповідність оформлення БДР діючим вимогам.

5 Порядок контролю виконання та захисту БДР

Робота виконується в три етапи, таблиця А.1.

6 Перелік обов'язкового ілюстративного матеріалу:

- пояснювальна записка ДР;
- графічні і ілюстративні матеріали;
- протокол попереднього захисту ДР на кафедрі;
- відгук наукового керівника;
- відгук рецензента;
- анотації українською та іноземною мовами, нормоконтроль про відповідність оформлення ДР діючим вимогам.

Таблиця А.1 - Етапи виконання роботи

Етап	Зміст	Початок	Кінець	Результат
1.	Інформаційний пошук та огляд літературних джерел.	13.09.2021	14.12.2021	Розділи 1 та 2.
2.	Дослідження підходів та розробка схемної реалізації.	15.12.2021	02.01.2022	Чернетки матеріалів.
3.	Підготовка матеріалів пояснювальної записки.	03.01.2022	14.01.2022	Пояснювальна

7. Порядок контролю та прийому

До приймання бакалаврської дипломної роботи надається:

- рубіжний контроль;
- попередній захист;
- захист на ДЕК.

Технічне завдання до виконання прийняв

Бажан В.В.

ДОДАТОК Б

Загальний алгоритм роботи застосунку



Рисунок Б.1 — Сценарій архітектури клієнт - сервер для програми, обробка подій



Рисунок Б.2 — Сценарій архітектури клієнт - сервер для програм, маршрутизації.



Рисунок Б.3 — Схема архітектури клієнт - сервер для програми, аутентифікації сервера, авторизації та аутентифікації.



Рисунок Б.4 — Сценарій архітектури клієнт - сервер для програми, що запитує базу даних.



Рисунок Б.5 — Сценарій архітектури клієнт - сервер для програми, що генерує відповіді клієнтам.



Рисунок Б.6 — Схема архітектури клієнт - сервер для програми, візуалізації сторінок, стилів та анімації.

ДОДАТОК В

Лістинг основних компонентів програми

```
import React from "react";
import ReactDOM from "react- dom";
import App from "./App";
import { BrowserRouter as Router } from "react- router- dom";
ReactDOM.render(
  <Router>
    <React.StrictMode>
      <App />
    </React.StrictMode>
  </Router>,
  document.getElementById("root")
);
import React from "react";
import { Switch, Route, Link } from "react- router- dom";
import "./scss/app.scss";
import Welcome from "./pages/Welcome";
import Specialties from "./pages/Specialties";
import DetailedPage from "./pages/DetailedPage";
import DetailedTitle from "./pages/sections/DetailedTitle";
import DetailedNumbers from "./pages/sections/DetailedNumbers";
import DetailedFAG from "./pages/sections/DetailedFAG";
import DetailedLinks from "./pages/sections/DetailedLinks";
function App() {
  return (
    <div className="wrapper">
      <div className="detailed">
        <DetailedPage />
      </div>
    </div>
  );
}
```

```
</div>
<Switch>
  <Route path="/" exact>
    <Welcome />
  </Route>
  <Route path="/specialities">
    <Specialties />
  </Route>
  <Route path="/computer- engineering">
    <DetailedPage />
  </Route>
  <Route path="/software- engineering">
    <DetailedPage />
  </Route>
  <Route path="/security- of- information- and- communication">
    <DetailedPage />
  </Route>
  <Route path="/cybersecurity- of- critical- systems">
    <DetailedPage />
  </Route>
  <Route path="/system- programming">
    <DetailedPage />
  </Route>
  <Route path="/computer- science ">
    <DetailedPage />
  </Route>
</Switch>
</div>
);
```

```

}
export default App;
import React from "react";
import { Tabs, useTabState, Panel } from "../helpers/tabs.jsx";
const cn = (...args) => args.filter(Boolean).join(" ");
const Tab = ({ children, className }) => {
  const { isActive, onClick } = useTabState();
  return (
    <li className={cn(isActive && "active")} onClick={onClick}>
      {children}
    </li>
  );
};
const Categories = ({ categories, onClickCategory, activeCategory }) => {
  return (
    <ul className="categories">
      {categories &&
        categories.map((item, index) => {
          return (
            <Tab
              className={activeCategory === index ? "active" : ""}
              onClick={() => onClickCategory(index)}
              key={`_${item}_${index}`}
            >
              {item}
            </Tab>
          );
        })}
    </ul>
  );
};

```

```

    );
  };
  export default Categories;
  import React, {
    createContext,
    useState,
    useMemo,
    useContext,
    cloneElement,
    isValidElement,
  } from "react";
  import useConstant from "use-constant";
  const TabsState = createContext();
  const Elements = createContext();
  export const Tabs = ({ state: outerState, children }) => {
    const innerState = useState(1);
    const elements = useConstant(() => ({ tabs: 0, panels: 0 }));
    const state = outerState || innerState;
    return (
      <Elements.Provider value={elements}>
        <TabsState.Provider value={state}>{children}</TabsState.Provider>
      </Elements.Provider>
    );
  };
  export const useTabState = () => {
    const [activeIndex, setActive] = useContext(TabsState);
    const elements = useContext(Elements);
    const tabIndex = useConstant(() => {
      const currentIndex = elements.tabs;

```

```

    elements.tabs += 1;
    return currentIndex;
  });
  const onClick = useConstant(() => () => setActive(tabIndex));
  const state = useMemo(
    () => ({
      isActive: activeIndex === tabIndex,
      onClick,
    }),
    [activeIndex, onClick, tabIndex]
  );
  return state;
};

export const usePanelState = () => {
  const [activeIndex] = useContext(TabsState);
  const elements = useContext(Elements);
  const panelIndex = useConstant(() => {
    const currentIndex = elements.panels;
    elements.panels += 1;
    return currentIndex;
  });
  return panelIndex === activeIndex;
};

export const Tab = ({ children }) => {
  const state = useTabState();
  if (typeof children === "function") {
    return children(state);
  }
  return isValidElement(children) ? cloneElement(children, state) : children;
};

```



```
};  
export const Panel = ({ active, children }) => {  
  const isActive = usePanelState();  
  return isActive || active ? children : null;  
};
```

ДОДАТОК Г

Лістинг загальних стилів додатку

```
@import "fonts";
@import "variables";
@import "libs/normalize";
@import "components/all.scss";
.wrapper {
  color: $white;
}
.welcome {
  &__title {
    padding- top: 44px;
    position: relative;
    display: flex;
    align- items: center;
    justify- content: space- between;
    width: calc(100vw - 1500px);
    margin: 0 auto;
  }
  &__container {
    padding- top: 86px;
    display: flex;
    width: calc(100vw - 600px);
    justify- content: space- between;
    align- items: center;
    margin: 0 auto;
  }
  &__descriptions {
    display: flex;
```

```
justify- content: space- around;
flex- direction: column;
}
&__heading {
margin- bottom: 50px;
}
&__links {
display: flex;
justify- content: space- between;
width: calc(100vw - 1000px);
margin: 0 auto;
}
&__specialties,
&__finder {
margin- top: 36px;
display: flex;
justify- content: space- evenly;
align- items: center;
background- color: rgba(10, 13, 31, 0.6);
color: #f2f6ff;
width: 420px;
height: 80px;
border- radius: 25px;
border: none;
&:hover {
background- color: rgba(10, 13, 31, 0.75);
}
p {
font- size: 26px;
```

```
    font-weight: 400;
  }
}
}
.Drawer {
  transition: transform 700ms cubic-bezier(0.75, 0, 0, 1),
    visibility 600ms linear;
  position: fixed;
  top: 0;
  left: 0;
  margin-left: calc(100vw - 98%);
  margin-top: 50px;
  width: 400px;
  height: 850px;
  transform: translateX(0px);
  background: rgba(242, 246, 255, 0.95);
  z-index: 90;
  border-radius: 36px;
  ul {
    margin-top: 30px;
    display: flex;
    flex-direction: column;
    height: 86%;
    a {
      &:last-of-type {
        margin-top: auto;
      }
    }
  }
}
```

```
&__controls {
  position: relative;
  justify-content: flex-end;
  display: flex;
  width: 300px;
  margin: 40px auto 0;
  svg {
    width: 40px;
    height: 40px;
    position: relative;
  }
}

&__item {
  cursor: pointer;
  display: flex;
  align-items: center;
  margin: 16px auto 0;
  width: 380px;
  height: 82px;
  padding: 10px 40px;
  border-radius: 26px;
  font-size: 24px;
  font-weight: 500;
  &:hover {
    background: rgba(10, 13, 31, 0.15);
  }
  &:active {
    background: rgba(10, 13, 31, 0.25);
  }
}
```

```
div {
  display: flex;
}
p {
  display: block;
  margin-left: 40px;
  text-align: left;
}
}
}
.Drawer.close {
  transform: translateX(- 500px);
}
.Backdrop {
  z-index: 50;
  background: rgba(10, 13, 31, 0.4);
  position: fixed;
  top: 0;
  left: 0;
  bottom: 0;
  right: 0;
}
```

ДОДАТОК Д

Лістинг алгоритму на основі семантичного пошуку

```

unction my_stripos(haystack, needle){ //v.1.1
  //Робить пошук в строці $haystack з підстроку $needle.
  //Якщо $needle є у $haystack то вертає true
  if(String(haystack).indexOf(needle)>- 1){
    return true;
  }else{
    return false;
  }
}
</script>
<p>
  <span style="display:block;">
    <input type="text" class="welcome__finder" name="search"
placeholder="Пошук..." href="/'+sUrl[i]+' style="padding- left:28px;
width:100%; font- size:28px;" oninput="count=0;
document.getElementById('vkj').innerHTML=";
sName=[];
sUrl=[];
arrToSearch.forEach((element) => {
  //console.log( element['keywords'] );
    if(my_stripos(element['keywords'].toLowerCase(),
this.value.toLowerCase())==true){
      sName[count]=element['name'];
      sUrl[count]=element['id'];
      count++;
    }
  }
}

```

```

    });
    text="";
    for(i=0; i<sName.length; i++){
        text+='\<div><a style=\'padding:15px; margin:10px; color:#fff !important;
width:100%;          text-          decoration:none;          color:#333;\'
href=\'/'+sUrl[i]+'\'>'+sName[i]+'</a></div>';
    }
    if(this.value.length>0)
        document.getElementById('resultsJS').innerHTML='Для вас підбрано
наступні спеціальності: '+text;
    else
    document.getElementById('resultsJS').innerHTML=";" autocomplete="off">
    <br style="clear:both;">
    <div id="resultsJS" style="clear: both;">Введіть назву спеціальності, яку
необхідно знайти</div>
    </span>
    </p>

    <Link to="/">Повернутись на Головну</Link>
    </div>
);
};

```


ДОДАТОК Е

Акт впровадження результатів науково - дослідних робіт

АКТ ВПРОВАДЖЕННЯ № _____

результатів науково - дослідних робіт

Замовник Центр дистанційної освіти ВНТУ

Цим актом підтверджується, що результати роботи «Розробка кросплатформенної інформаційної абітурієнта системи з використанням семантичного пошуку», що виконана студентом гр.1КІ - 18Б Бажаном В.В. за договором про творчу співдружність без взаємних грошових розрахунків, на громадських засадах виконана з 02.02.2021 по 01.06.2022 впроваджено у Центрі дистанційної освіти ВНТУ.

1. Вид впроваджених результатів програмне забезпечення для інформаційно - пошукової системи
2. Характеристика масштабу впровадження одиничне
3. Форма впровадження програма
4. Новизна результатів науково - дослідної роботи модифікація
5. Впроваджені: в інформаційно - пошуковій системі
6. Соціальний та науково - технічний ефект спеціальне

Від виконавця:

Від ВНТУ:

Студенти групи 1КІ -18Б

Декан ФІТКИ

Бажан В.В.

Кирилищук С.А.

Науковий керівник

Крупельницький Л.В.

ДОДАТОК Ж
ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ
ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Кросплатформенна інформаційна система абітірієнта з використанням семантичного пошуку .

Тип роботи: бакалаврська дипломна робота
Підрозділ кафедра обчислювальної техніки

Показники звіту подібності
Unicheck

Оригінальність 69,2% Схожість 30,8%

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 відсутності самостійності її виконання автором. 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

Захарченко С.М.

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

Автор роботи

Бажан В.В.

Керівник роботи

Крупельницький Л.В.