

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: «Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини»

Виконала: студентка 4 курсу
групи ЗПІ-186
спеціальності

 121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

 Маланчук А.В.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Войтко В.В.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. КН Арсенюк І.Р.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедри _____

« ___ » _____ 2022 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший бакалаврський
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

УЗГОДЖЕНО
Директор Демківського ЗДО
Магальяс Л.М.
25 березня 2022 року

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О.Н.
25 березня 2022 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Маланчук Анні Василівні

1. Тема роботи – «Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини»

Керівник роботи: Войтко Вікторія Володимирівна, к.т.н., доц. кафедри ПЗ, затверджені наказом вищого навчального закладу від 24 березня 2022 року №66. Строк подання студентом роботи 13 червня 2022 року.

3. Вихідні дані до роботи: середовище розробки JetBrains IntelliJ Idea, мова розробки Java та веб-фреймворк Spring Boot, операційна система – Windows 10, сервер баз даних MySQL, базові алгоритми.

4. Зміст розрахунково-пояснювальної записки: вступ; аналіз та постановка задач; аналіз даних; розробка веб-системи; розробка методів реалізації веб-сайтів; розробка моделі веб-системи; розробка алгоритмів веб-системи; розробка програмного забезпечення веб-системи; тестування веб-системи; висновки; перелік посилань; додатки.

5. Перелік графічного матеріалу: мета, об'єкт та предмет дослідження, задачі дослідження, наукова новизна, практична цінність одержаних

результатів, порівняльний аналіз аналогів, структура веб-сайту; модель роботи веб-системи; блок-схеми алгоритмів роботи веб-системи; тестування веб-системи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Войтко В. В., к.т.н., доцент кафедри ПЗ		

7. Дата видачі завдання 25 березня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	26.03.2022-30.03.2022	Виконано
2	Аналіз методів розв'язання задачі	01.04.2022-03.04.2022	Виконано
3	Постановка задач розробки програмного продукту	05.04.2022-07.04.2022	Виконано
4	Аналіз інформаційного забезпечення	08.04.2022-11.04.2022	Виконано
5	Аналіз варіантів використання веб-системи	13.04.2022-24.04.2022	Виконано
6	Розробка моделі роботи системи	25.04.2022-29.04.2022	Виконано
7	Вибір засобів реалізації	30.04.2022-04.05.2022	Виконано
8	Розробка сайту	05.05.2022-18.05.2022	Виконано
9	Тестування роботи веб-сайту	19.05.2022-22.05.2022	Виконано
10	Оформлення матеріалів до захисту БДР	03.06.2022-10.06.2022	Виконано

Студент

Маланчук А.В.

(підпис)

(прізвище та ініціали)

Керівник бакалаврської дипломної роботи

Войтко В.В.

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 96 сторінок формату А4, на яких є 19 рисунків, 2 таблиць, список використаних джерел містить 29 найменувань.

У бакалаврській дипломній роботі спроектовано та розроблено веб-систему для комплексної оцінки та підтримки розвитку дитини. Було проведено аналіз аналогів, які існують, визначено їх переваги і недоліки.

На основі проведених досліджень було запропоновано розробити власний веб-сайт, який буде містити у собі елементи розвитку для дітей.

Розроблено веб-сайт, який несе повчальний та пізнавальний характер. Саме тут можна побачити інформацію не тільки для батьків, які зацікавлені привести свою дитину в Демківський ЗДО, а й розвиваючі ігри, інтерактиви, відео, які будуть допомагати розвиватися малюкам. Тому даний сайт буде цікавий як і дорослим, так і дітям.

Розроблено метод, модель роботи веб-системи. Для формування структури сайту використано HTML. Для додання динамічності сайту була використана мова програмування Java, для створення бази даних – SQL

Ключові слова: веб-сайт, садочок, Java .

ABSTRACT

The bachelor's thesis consists of 90 A4 pages, which have 19 figures, 2 tables, a list of sources used contains 29 titles.

In the bachelor's thesis, a web system for comprehensive assessment and support of child development was designed and developed. An analysis of existing analogues was conducted, their advantages and disadvantages were identified.

Based on the research, it was proposed to develop its own website, which will contain elements of development for children.

An educational and informative website has been developed. Here you can see information not only for parents who are interested in bringing their child to Demkivsky ZDO, but also educational games, interactives, videos that will help kids develop. Therefore, this site will be interesting for both adults and children.

A method and model of web system operation have been developed. HTML was used to form the structure of the site. Java programming language was used to add dynamism to the site, and SQL was used to create the database

Keywords: website, kindergarten, Java.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ.....	11
1.1 Аналіз предметної області	11
1.2 Порівняльний аналіз аналогів	17
1.3 Аналіз методів розв’язання задачі	19
1.4 Постановка задачі розробки програмного продукту	25
1.5 Висновок	26
2 РОЗРОБКА МЕТОДУ, МОДЕЛІ ТА СТРУКТУРИ САЙТУ	27
2.1 Аналіз інформаційного забезпечення.....	27
2.2 Розробка структури інтерфейсу сайту.....	27
2.3 Аналіз варіантів використання веб-системи.....	32
2.4 Розробка методу і моделі роботи веб-системи.....	33
2.5 Висновок	34
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СИСТЕМИ.....	35
3.1 Варіантний аналіз та обґрунтування вибору засобів реалізації	35
3.2 Розробка бази даних	45
3.3 Розробка механізмів авторизації і реєстрації.....	47
3.4 Розробка механізмів додавання і показу новин.....	49
3.5 Висновок	49
4 ТЕСТУВАННЯ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ	51
4.1 Огляд рішень для тестування сайтів.....	51
4.2 Тестування роботи програмного продукту	52

4.3 Висновок.....	55
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
ДОДАТКИ	62
ДОДАТОК А – Технічне завдання.....	63
ДОДАТОК Б –Протокол перевірки на плагіат	66
ДОДАТОК В – Акт впровадження	67
ДОДАТОК Г – Лістинг програми	68
ДОДАТОК Д – Графічна частина	88

ВСТУП

Обґрунтування вибору теми дослідження. В сучасному світі кожного дня кількість інформації, яку необхідно зберігати, росте в геометричній прогресії, набуваючи величезних розмірів. У певний момент старі варіанти зберігання інформації, такі як паперові носії або навіть усна форма, віджили своє і більше не могли задовольняти потреби сучасного соціуму.

З'явилися системи автоматизації, які до сьогоднішнього дня допомагають людям у структуруванні та зберіганні різного роду інформації, взаємодії між ролями користувачів тощо.

Сьогодні кожне підприємство та установа мають свої внутрішні клієнти, які допомагають зберігати і структурувати інформацію про роботу організації, автоматизувати деякі процеси тощо.

Веб-ресурси дозволяють зберігати й презентувати інформацію широкому колу користувачів [1].

Сьогодні кожна юридична особа прагне мати власний сайт, щоб заявити про себе і свої досягнення. Сучасні заклади освіти – не виключення. Сайти закладів шкільної та дошкільної освіти потребують підбору інформації, орієнтованої не тільки на дорослих, але й цікавої для дітей. Тому розробка сайту дитячого садка як веб-системи для комплексної оцінки та підтримки розвитку дитини є актуальною задачею.

Мета та завдання дослідження. Метою бакалаврської дипломної роботи є підвищення ефективності роботи закладу дошкільної освіти шляхом розробки спеціалізованої веб-системи з елементами розвиваючих вправ і матеріалами для дітей, що забезпечить ефективну взаємодію суб'єктів навчально-виховного процесу дитячого садка.

Основними **задачами** роботи є:

- розробити метод комплексної оцінки та підтримки розвитку дитини у закладах дошкільної освіти з використанням спеціалізованої веб-системи;

- розробити модель роботи веб-системи;
- проаналізувати інформаційне забезпечення сайту;
- розробити програмні засоби реалізації веб-системи комплексної оцінки та підтримки розвитку дитини у закладах дошкільної освіти;
- розробити зручний та зрозумілий графічний інтерфейс веб-ресурсу;
- провести тестування веб-системи відповідно до поставлених задач.

Об’єктом дослідження є процеси створення веб-системи для комплексної оцінки та підтримки розвитку дитини.

Предмет дослідження — засоби реалізації веб-системи закладу дошкільної освіти.

Методи дослідження. У процесі роботи використовувалися методи дослідження:

- методи аналізу та синтезу для побудови ключових моделей веб-системи;
- методи побудови компонентів системи та їхніх взаємозв’язків для створення загальної структури збереження та обробки інформаційних ресурсів;
- методи прикладної теорії інформації і теорії алгоритмів для розробки алгоритмів і програмного забезпечення.
- методи тестування для перевірки працездатності системи.

Наукова новизна отриманих результатів.

- Подальшого розвитку отримав метод комплексної оцінки та підтримки розвитку дитини з використанням спеціалізованої веб-системи, який, на відміну від існуючих, орієнтований на використання елементів розвиваючих вправ і матеріалів для дітей, що підвищить ефективність взаємодії суб’єктів навчально-виховного процесу дитячого садка.

- Подальшого розвитку отримала модель роботи веб-системи закладу дошкільної освіти, яка, на відміну від існуючих, орієнтована на покращення комплексної оцінки та підтримки розвитку дитини з використанням елементів

розвиваючих вправ і матеріалів для дітей, що дозволить підвищити ефективність веб-ресурсу.

Практична цінність отриманих результатів. Практична цінність полягає у кінцевій реалізації інформаційної веб-системи закладу дошкільної освіти, що дозволить налагодити ефективну взаємодію суб'єктів навчально-виховного процесу.

Особистий внесок здобувача. Усі наукові результати, викладені у бакалаврській дипломній роботі, отримані автором особисто. У науковій роботі [2], опублікованій у співавторстві, автору належить розробка інформаційної веб-системи закладу дошкільної освіти.

Апробація матеріалів бакалаврської дипломної роботи. Основні положення бакалаврської дипломної роботи доповідалися та обговорювалися на LI науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2022) ВНТУ.

Публікації. Основні результати дослідження опубліковані в науковій роботі [3] – в тезах доповіді на LI науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2022) ВНТУ.

Аналіз. У пояснювальній записці до бакалаврської дипломної роботи розглянуто 4 розділи та використано 29 літературних джерел.

1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ

1.1 Аналіз предметної області

Веб-сайт - це сукупність веб-сторінок та пов'язаного вмісту, які ідентифікуються загальним доменним ім'ям та публікуються принаймні на одному веб-сервері. Помітними прикладами є wikipedia.org, google.com та amazon.com.

Усі загальнодоступні веб-сайти в сукупності складають Всесвітню павутину. Існують також приватні веб-сайти, доступ до яких доступний лише в приватній мережі, наприклад, внутрішній веб-сайт компанії для її співробітників.

Веб-сайти, як правило, присвячені певній темі або меті, такі як новини, освіта, комерція, розваги чи соціальні мережі. Гіперпосилання між веб-сторінками спрямовує навігацію по сайту, яке часто починається з домашньої сторінки.

Користувачі можуть отримати доступ до веб-сайтів на різних пристроях, включаючи настільні, ноутбуки, планшети та смартфони. Програма, що використовується на цих пристроях, називається веб-браузером.

Всесвітня павутина (WWW) була створена в 1990 році британським фізиком ЦЕРН Тімом Бернерсом-Лі [1]. 30 квітня 1993 р. ЦЕРН оголосив, що Всесвітня павутина буде безкоштовною для використання будь-ким.

До впровадження протоколу передачі гіпертексту (НТТР) для отримання окремих файлів із сервера використовувались інші протоколи, такі як протокол передачі файлів та протокол *gopher*. Ці протоколи пропонують просту структуру каталогів, в якій користувач переходить і де він обирає файли для завантаження. Документи найчастіше подавалися у вигляді текстових файлів без форматування або кодувалися у форматах текстового процесора.

Сайти можна використовувати різними способами: особистий сайт, корпоративний сайт, державний сайт, веб-сайт організації тощо. Веб-сайти

можуть бути дітищем окремої особи, компанії чи іншої організації і, як правило, присвячені певній темі чи цілі. Кожен веб-сайт може містити гіперпосилання на будь-який інший веб-сайт, тому різниця між різними веб-сайтами, яку сприймає користувач, може бути розмитою.

Деякі сайти вимагають реєстрації користувача або підписки, щоб отримати доступ до вмісту. Приклади сайтів для підписки включають багато бізнес-сайтів, сайтів новин, сайтів академічних журналів, ігрових сайтів, файлообмінних сайтів, дошок оголошень, адрес електронної пошти, сайтів соціальних мереж, сайтів фондового ринку в реальному часі та багатьох інших сайтів.

Хоча "веб-сайт" був оригінальним написанням (іноді з великої літери "Веб-сайт", оскільки "Веб" є власним іменником, коли йдеться про Всесвітню павутину), цей варіант став рідкісним, а "веб-сайт" став стандартним написанням. Усі основні керівництва стилем, такі як Чиказький посібник стилю[3] та AP Stylebook, відображають цю зміну.

Статичний веб-сайт - це веб-сторінки, які мають веб-сторінки, що зберігаються на сервері у форматі, який надсилається клієнтському веб-браузеру. В основному він кодується мовою розмітки гіпертексту (HTML); Каскадні таблиці стилів (CSS) використовуються для контролю зовнішнього вигляду за базовим HTML. Зображення зазвичай використовуються для отримання бажаного вигляду та як частина основного змісту. Аудіо чи відео також можна вважати "статичним" вмістом, якщо він відтворюється автоматично або, як правило, є неінтерактивним. Цей тип веб-сайтів зазвичай відображає однакову інформацію для всіх відвідувачів. Подібно до роздачі друкованої брошури клієнтам або клієнтам, статичний веб-сайт, як правило, надає послідовну стандартну інформацію протягом тривалого періоду часу. Незважаючи на те, що власник веб-сайту може періодично робити оновлення, редагування тексту, фотографій та іншого вмісту здійснюється вручну, і може знадобитися базові навички дизайну веб-сайту та програмне забезпечення.

Прості форми або маркетингові приклади веб-сайтів, такі як класичний веб-сайт, веб-сайт із п'ятьма сторінками або веб-сайт брошури, часто є статичними веб-сайтами, оскільки вони представляють користувачеві заздалегідь визначену, статичну інформацію. Це може включати інформацію про компанію та її продукти та послуги через текст, фотографії, анімацію, аудіо / відео та навігаційні меню.

Статичні веб-сайти все ще можуть використовувати серверні компоненти (SSI) як зручність редагування, наприклад, спільний доступ до загального рядка меню на багатьох сторінках. Оскільки поведінка сайту до читача все ще залишається статичним, це не вважається динамічним сайтом.

Динамічний сайт – це сайт, який змінюється та адаптується часто й автоматично. Динамічні сторінки на сервері створюються "на льоту" з використанням комп'ютерного коду, який генерує зовнішній вигляд HTML (CSS). Існують різні програмні системи для створення динамічних веб-систем і динамічних веб-сайтів, таких як CGI, Java Servlet і Java Server Pages (JSP), Active Server Pages і ColdFusion (CFML).

У популярних мовах програмування, таких як Perl, PHP, Python і Ruby, є багато фреймворків веб-додатків і систем веб-шаблонів, які дозволяють легко та просто створювати складні та динамічні веб-сайти.

Сайт може відображати поточний стан розмов між користувачами, відстежувати зміну обставин або надавати інформацію випадковим чином, персоналізовану відповідно до індивідуальних вимог користувача. Наприклад, на запит домашньої сторінки сайту новин код, що виконується на веб-сервері, може поєднувати збережені фрагменти HTML з новинами, отриманими через RSS з бази даних або іншого сайту, щоб створити сторінку з актуальною інформацією. Динамічні сайти можуть взаємодіяти з HTML-формами, зберігати та читати файли cookie браузера або створювати серію сторінок, які відображають вашу попередню історію кліків. Іншим прикладом динамічного вмісту є сайт бази даних роздрібних медіа, який дозволяє користувачам вводити

пошукові запити, такі як ключові слова для The Beatles, дисків і книг. Динамічний HTML використовує код JavaScript, щоб повідомляти веб-браузеру, як взаємодіяти з вмістом сторінки.

Одним із способів моделювання конкретного типу динамічного веб-сайту, уникаючи втрати ініціалізації динамічного механізму для кожного користувача або входу, є періодичне автоматичне створення великої серії статичних сторінок.

Перші сайти мали лише текст, а незабаром і зображення. Плагіни веб-браузера потім використовувалися для додавання відео, аудіо та інтерактивності (наприклад, у випадку багатофункціональної веб-програми, яка відображає складність робочого столу як текстового редактора). Прикладами таких плагінів є Microsoft Silverlight, Adobe Flash, Adobe Shockwave і аплети Java. HTML 5 містить положення для аудіо та відео без плагінів.

JavaScript також вбудований у більшість сучасних веб-браузерів і дозволяє розробникам веб-сайтів надсилати код веб-браузера, який вказує їм, як інтерактивно змінювати вміст сторінки та спілкуватися з веб-сервером, якщо це необхідно. Внутрішнє представлення вмісту браузера відоме як модель об'єкта документа (DOM), а методика відома як динамічний HTML.

WebGL (Web Graphics Library) - це сучасний API JavaScript для візуалізації інтерактивної 3D-графіки без використання плагінів. Це дозволяє інтерактивний контент, такий як 3D-анімація, візуалізація та відео-пояснення, викласти користувачам найбільш інтуїтивно зрозумілим способом [4].

Тенденція 2010 року на веб-сайтах під назвою "адаптивний дизайн" забезпечила найкращий досвід перегляду, оскільки надає користувачам макет на основі пристрою. Ці веб-сайти змінюють свій макет відповідно до пристрою або мобільної платформи, що забезпечує багатий досвід користувачів [5].

Веб-сайти поділяють на статичні та інтерактивні. Інтерактивні сайти здебільшого є частиною спільноти Web 2.0, що дозволяє взаємодіяти між власниками сайтів і користувачами сайту. Статичні веб-сайти надають або

збирають інформацію, але не дозволяють безпосередньо взаємодіяти зі своєю аудиторією та користувачами. Деякі сайти носять інформаційний або ентузіазмовий характер, або для особистого чи розважального використання. Більшість веб-сайтів прагнуть заробляти гроші за допомогою однієї або кількох бізнес-моделей, зокрема:

- Розміщення привабливого контенту та продаж контекстної реклами або шляхом прямих продажів, або через рекламну мережу.
- Електронна комерція – тобто, товари чи послуги купуються безпосередньо через веб-сайт.
- Реклама товарів або послуг, доступних у цегельному та будівельному бізнесі
- Freemium: базовий вміст доступний безкоштовно, але преміум-вміст вимагає оплати (наприклад, веб-сайт WordPress, це платформа з відкритим кодом для створення блогу чи веб-сайту).

Поодинокі веб-сайти можуть належати до однієї або кількох із цих категорій. Наприклад, комерційний веб-сайт може рекламувати комерційні продукти, але також може публікувати інформаційні матеріали, наприклад технічні документи.

Існує також багато підкатегорій, перерахованих вище. Наприклад, є сайти електронної комерції або сайти продавця (тобто намагаються продати членство, щоб отримати доступ до свого сайту) або мають функцію соціальної мережі.

Фан-сайти можуть бути присвячені власнику певної знаменитості. Веб-сайт обмежений архітектурними обмеженнями, такими як обчислювальна потужність веб-сайту.

Масштабні веб-сайти, такі як Facebook, Yahoo!, Microsoft і Google, використовують багато серверів і обладнання для балансування навантаження, наприклад комутатори Cisco Content Services, щоб розподілити навантаження відвідувачів на кілька комп'ютерів у різних місцях.

На початок 2011 року Facebook використовував 9 центрів обробки даних із приблизно 63000 серверами.

У лютому 2009 року компанія Netcraft, компанія з моніторингу Інтернету, яка відслідковувала зростання Інтернету з 1995 р., Повідомила, що в 2009 р. Існувало 215 675 903 веб-сайтів з доменними іменами та вмістом на них, порівняно з лише 19 732 веб-сайтами в серпні 1995 р. [6]

Після досягнення 1 мільярда веб-сайтів у вересні 2014 року, етап, підтверджений NetCraft в огляді веб-серверів за жовтень 2014 року, і те, що Інтернет-статистика стала першим, хто повідомив про це, про що свідчить цей твіт від самого винахідника Всесвітньої мережі, Тіма Бернерса Лі - кількість веб-сайтів у світі згодом зменшилась, повернувшись до рівня нижче 1 мільярда. Це пояснюється з щомісячними коливаннями кількості неактивних веб-сайтів.

Кількість веб-сайтів до березня 2016 р. продовжувала зростати до понад 1 млрд. І з тих пір продовжує зростати.

Веб-розробка - це робота, пов'язана з розробкою веб-сайту для Інтернету (Всесвітня павутина) або інтрамережі (приватна мережа). Веб-розробка може починатись від розробки простої однієї статичної сторінки простого тексту до складних Інтернет-програм (Інтернет-додатків), електронного бізнесу та послуг соціальних мереж.

Більш повний перелік завдань, до яких зазвичай відноситься веб-розробка, може включати веб-інженерію, веб-дизайн, розробку веб-вмісту, зв'язок з клієнтом, сценарії на стороні клієнта або сервера, налаштування безпеки веб-сервера та мережі та розвиток електронної комерції.

Серед веб-спеціалістів "веб-розробка" зазвичай відноситься до основних недизайнерських понять побудови веб-сайтів: написання розмітки та кодування. Веб-розробка може використовувати системи управління вмістом (CMS), щоб зробити зміст простішим та доступнішим з базовими технічними навичками.

Дитячий садок – це підхід до дошкільної освіти, заснований на грі, співі, практичних заняттях, таких як малювання, та соціальній взаємодії як частина переходу від дому до школи.

Такі заклади спочатку були створені наприкінці 18 століття в Баварії та Ельзасі для обслуговування дітей, чиї батьки обоє працювали поза домом. Термін був введений німцем Фрідріхом Фрьобелем, чий підхід глобально вплинув на освіту ранніх років. Сьогодні цей термін використовується в багатьох країнах для опису різноманітних навчальних закладів та навчальних просторів для дітей віком від 2 до 6 років на основі різноманітних методів навчання.

Перевіривши можливості, які наявні при розробці сайту можна стверджувати, що це доволі не проста система, яка може забезпечити якісну роботу зв'язку адміністрації садка з батьками, дітьми, вихователями та потенційними інвесторами.

Отже, розробка сайту садка є актуальною і необхідною.

1.2 Порівняльний аналіз аналогів

Сьогодні існує дуже багато закладів дошкільної освіти, які мають власний web-сайт. В сучасних умовах, а саме в умовах інформаційних технологій – це необхідний чинник існування.

Аналіз результатів пошуку web-сайтів дошкільних закладів, а саме садочків, у мережі Internet дозволяє зробити висновок, що на даний момент існує дуже мало дійсно достойних сайтів.

Розглянемо кілька прикладів сайтів дитячих садочків.

Отже, сайт міні-садка «Смайл» [7]. На перший погляд, сайт має вдалий дизайн (рис. 1.1), який справляє приємне враження, але є суттєвий недолік – відсутність розділу призначеного саме для розвитку дітей.



Рисунок 1.1 — Дитячий садочок "Пазлик"

Також зустрічаються елементарні помилки при виборі типу кодування інформації, прикладом такого сайту може слугувати сайт «Пролісок» [8]. У даному прикладі відсутня будь-яка інформація, яка б могла зацікавити користувача, окрім адреси та кількох слів про сам садок (рис 1.2).

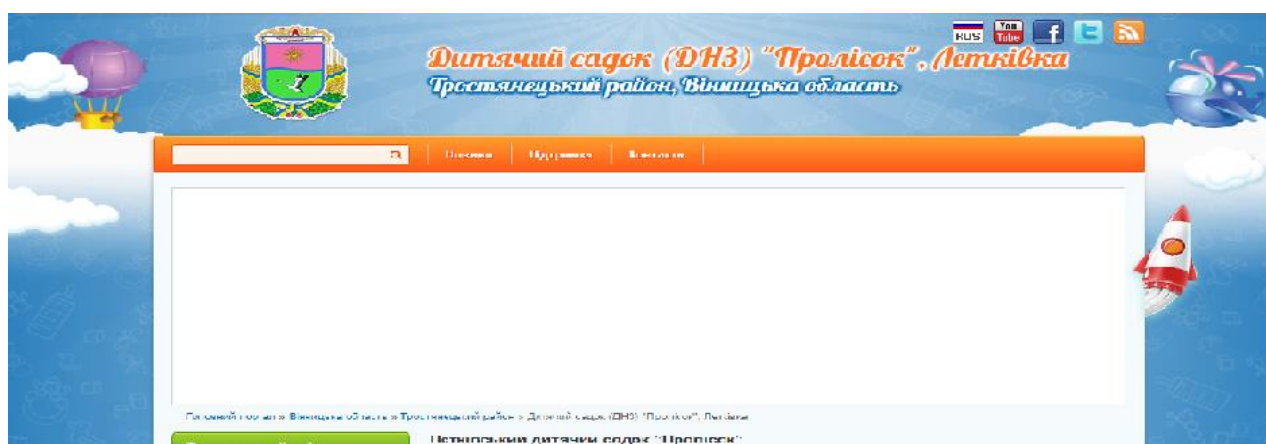


Рисунок 1.2 — Дитячий садок "Пролісок"

Для того, щоб підкреслити переваги розроблюваного додатку, було проведено порівняльний аналіз за певними функціями, які присутні в новій версії, але можливо відсутні в старій.

Результати порівняльного аналізу аналогів наведено в таблиці 1.1

Таблиця 1 – Результати порівняльного аналізу аналогів

Функції	Демківський ЗДО	Дитячий садочок "Пазлик"	Дитячий садок "Пролісок"
Новини	+	+	+
Дітям	+	+	-
Реєстрація	+	-	-
Зручна навігація	+	+	-
Швидкість	+	-	-
Контакти	+	+	-

Таблиця порівняльних характеристик показала, що розробка програмного продукту є доцільною. В результаті отримаємо продукт, що врахує наявні недоліки існуючих рішень.

1.3 Аналіз методів розв'язання задачі

Для розв'язання поставленої задачі слід створити інформаційну систему дитячого садочка.

Інформаційна система (ІС) — це формальна соціотехнічна організаційна система, призначена для зберігання, збору, обробки та поширення інформації. З соціально-технічної точки зору, інформаційні системи складаються з чотирьох

компонентів: завдання, люди, структура (або ролі) і технологія. Інформаційну систему можна визначити як інтеграцію компонентів, що використовуються для збору, обробки та зберігання даних, що використовуються в цифрових продуктах для надання інформації, полегшення знань та полегшення прийняття рішень.

Комп'ютерна інформаційна система – це система, що складається з комп'ютерів та людей, які обробляють або інтерпретують інформацію [9]. Термін також деколи використовується просто для позначення комп'ютерної системи, на якій встановлено програмне забезпечення.

«Інформаційні системи» також є академічним польовим дослідженням про системи з точним посиланням на інформацію та додаткові мережі комп'ютерного обладнання та програмного забезпечення, які люди та організації використовують для збору, обробки, фільтрації, створення та розповсюдження даних. Акцент ставиться на інформаційних системах з чітко визначеними межами, користувачами, процесорами, мережами зберігання, введення, виведення та зв'язку, як описано вище.

У багатьох організаціях відділ або підрозділ, відповідальний за інформаційні системи та обробку даних, називають «інформаційними службами»[10].

Будь-яка конкретна інформаційна система призначена для підтримки операцій, управління та прийняття рішень [11]. Інформаційні системи — це інформаційно-комунікаційні технології (ІКТ), які використовують організації, і те, як люди взаємодіють з цією технологією для підтримки бізнес-процесів [12].

Деякі автори чітко розрізняють інформаційні системи, комп'ютерні системи та бізнес-процеси. Інформаційні системи часто містять компонент ІКТ, але не повністю пов'язані з ІКТ, а зосереджені на кінцевому використанні інформаційних технологій. Інформаційні системи також відрізняються від бізнес-процесів. Інформаційні системи допомагають контролювати продуктивність бізнес-процесів [13].

Alter аргументує переваги розглядання інформаційної системи як особливого типу робочої системи [14]. Робоча система — це система, в якій люди або машини виконують процеси та дії, використовуючи ресурси для виробництва конкретних продуктів або послуг для клієнтів. Інформаційна система — це робоча система, діяльність якої спрямована на захоплення, передачу, зберігання, вилучення, маніпулювання та відображення інформації [15].

Таким чином, інформаційна система взаємопов'язана з системою даних, з одного боку, і з системою діяльності з іншого боку. Інформаційна система — це форма комунікаційної системи, в якій дані представлені та оброблені як форма соціальної пам'яті. Інформаційну систему можна також розглядати як напівформальну мову, яка підтримує прийняття рішень і дій людиною.

Інформаційні системи є основною сферою досліджень організаційної інформатики [16].

Відділи інформаційних технологій у великих організаціях мають тенденцію сильно впливати на розвиток, використання та застосування інформаційних технологій у бізнесі. Для розробки та використання інформаційної системи можна використовувати низку методологій і процесів. Багато розробників використовують підхід системної інженерії, такий як життєвий цикл розробки системи (SDLC), для систематичної розробки інформаційної системи поетапно.

Етапами життєвого циклу розробки системи є планування, системний аналіз і вимоги, проектування, розробка, інтеграція та тестування, впровадження та експлуатація, а також обслуговування. Останні дослідження спрямовані на те, щоб дозволити та виміряти поточний колективний розвиток таких систем в організації за допомогою всіх самих людей. Інформаційна система може бути розроблена власноруч (всередині організації) або передана на аутсорсинг. Це може бути досягнуто шляхом передачі на аутсорсинг певних

компонентів або всієї системи.[31] Конкретним випадком є географічний розподіл команди розробників (офшоринг, глобальна інформаційна система) [17].

Види інформаційної системи:

«Класичний» погляд на інформаційні системи, який зустрічався в підручниках у 1980-х, був пірамідою систем, що відображали ієрархію організації, як правило, системи обробки транзакцій у нижній частині піраміди, моніторинг інформаційних систем управління, системи підтримки прийняття рішень тощо. виконавчі інформаційні системи на вершині [18]. Хоча модель піраміди залишалася корисною з моменту її першої формулювання, було розроблено низку нових технологій і з'явилися нові категорії інформаційних систем, деякі з яких більше не вписуються в початкову модель піраміди. Деякі приклади таких систем:

- система підтримки прийняття рішень;
- соціальні інформаційні системи;
- система управління процесом;
- інформаційна система управління;
- інтелектуальна система;
- системи підприємства;
- сховища даних;
- планування ресурсів підприємства;
- обчислювальна платформа;
- експертні системи;
- пошукові системи;
- геоінформаційна система;
- глобальна інформаційна система;
- мультимедійна інформаційна система;

- автоматизація офісу.

(Комп'ютеризована) інформаційна система – це здебільшого інформаційна система, яка використовує комп'ютерні технології для виконання тільки деяких або всіх запланованих завдань. Основними компонентами комп'ютерної інформаційної системи є:

- Апаратне забезпечення складається з таких пристроїв, як монітори, процесори, принтери та клавіатури, які працюють разом для отримання, обробки та відображення даних та інформації.
- Програмне забезпечення – це програмне забезпечення, яке дозволяє апаратному забезпеченню обробляти дані.
- База даних – це набір пов'язаних файлів або таблиць, що містять зв'язані дані.
- Мережа – це система з'єднань, яка дозволяє різним комп'ютерам спільно використовувати ресурси.
- Процедури – це команди, які поєднують всі вищевказані компоненти для обробки інформації та досягнення бажаного результату.

Перші чотири компоненти (апаратне забезпечення, програмне забезпечення, база даних і мережа) формують обчислювальну платформу. IT-фахівці можуть використовувати ці компоненти для створення інформаційних систем, які контролюють безпеку, загрози та керування даними. Ці види діяльності відомі як IT-послуги [19].

Деякі IT-системи підтримують частини організації, інші підтримують цілі організації, а деякі – групи підтримки організацій. Кожен відділ або функціональна область в організації має свій набір додатків або IT-систем. Ці інформаційні системи функціональної області (FAIS) є опорами більш загального III, а саме систем бізнес-аналітики та панелей інструментів. Як впливає з назви, кожен FAIS підтримує певну функцію в організації, таку як

бухгалтерський облік, фінансова система, система управління виробництвом (РОМ), система маркетингу та людські ресурси.

У фінансах та бухгалтерському обліку менеджери використовують інформаційні системи для прогнозування доходів і господарських операцій, визначення найкращих джерел і використання ресурсів, а також проведення аудиту, щоб переконатися, що організація є істотно надійною, а всі фінансові звіти та документи є точними.

Інші типи організаційних інформаційних систем включають FAIS, системи обробки транзакцій, планування ресурсів підприємства, систему автоматизації офісу, систему управлінської інформації, систему підтримки прийняття рішень, експертну систему, панель керування, систему управління ланцюгом поставок та систему електронної комерції.

Інформаційні панелі – це особлива форма ІС, яка підтримує всіх керівників організації. Вони забезпечують швидкий доступ до своєчасної інформації та прямий доступ до структурованої інформації у вигляді звітів. Експертні системи намагаються дублювати роботу експертів, застосовуючи можливості міркування, знання та досвід у певній області.

Комп'ютерна інформаційна система, згідно з визначенням Лангефорса[20], є технологічно реалізованим середовищем для:

- запис, зберігання та поширення мовних виразів,
- а також для того, щоб робити висновки з таких виразів.

Географічні інформаційні системи, земельні інформаційні системи та системи про стихійні лиха є прикладами нових інформаційних систем, але їх можна широко розглядати як просторові інформаційні системи [21]. Розробка системи здійснюється поетапно, які включають:

- розпізнавання та уточнення проблеми;
- збір інформації;
- специфікація вимог до нової системи;

- проектування системи;
- побудова системи;
- реалізація системи;
- огляд і обслуговування.

1.4 Постановка задачі розробки програмного продукту

Головним завданням бакалаврської дипломної роботи є програмна реалізація веб-сайту дитячого садочку з елементами розвиваючих вправ і матеріалів для дітей.

З огляду на потрібний функціонал аналогів та з урахуванням їх недоліків сформулюємо задачі бакалаврської дипломної роботи:

- розробити метод комплексної оцінки та підтримки розвитку дитини у закладах дошкільної освіти з використанням спеціалізованої веб-системи;
- розробити модель роботи веб-системи;
- проаналізувати інформаційне забезпечення сайту;
- розробити програмні засоби реалізації веб-системи комплексної оцінки та підтримки розвитку дитини у закладах дошкільної освіти;
- розробити зручний та зрозумілий графічний інтерфейс веб-ресурсу;
- провести тестування веб-системи відповідно до поставлених задач.

Розроблюваний програмний продукт повинен мати такий функціонал:

- категоризована авторизація користувачів й адміністратора;
- показ новин садочку;
- показ посилань в розділі «Для дітей»;
- додавання новин;
- додавання посилань;
- механізм авторизації;
- механізм реєстрації;
- головна сторінка;

- навігаційне меню;
- сторінка «Про нас»;
- розвиваючі матеріали для дітей;
- інтерактиви й логічні та навчальні ігри.

1.5 Висновок

У першому розділі було проведено аналіз основних понять предметної області, розглянуто аналогічні рішення і виявлено основні методи вирішення задачі, сформульовано завдання до бакалаврської дипломної роботи.

Вся отримана інформація стане базисом для подальшої розробки власної програмної реалізації сайту дитячого садочку з елементами розвиваючих вправ і матеріалів для дітей.

2 РОЗРОБКА МЕТОДУ, МОДЕЛІ ТА СТРУКТУРИ САЙТУ

2.1 Аналіз інформаційного забезпечення

Найважливішим аспектом всієї системи є передача даних між сторінками і передача даних між клієнтом і сервером.

У даному випадку всі механізми реалізовані автоматично вбудованими засобами Spring Boot Framework за допомогою механізмів реквестів і анотацій.

Для передачі даних від сервера до клієнта достатньо додати до моделі атрибут, дати йому назву і помістити в нього відповідні дані. Після цього дані будуть доступні на стороні клієнта за допомогою відповідного синтаксису.

Для передачі даних від клієнта до сервера використовуються анотації `PostMapping` та `RequestMapping`, в залежності від ситуації.

В посиланні передаються відповідні дані, які потім, за допомогою мапінгів сервера відловлюються і форматуються.

2.2 Розробка структури інтерфейсу сайту

При розробці інтерфейсу було вирішено використовувати тенденції мінімалізму і не перевантажувати інтерфейс, ані контентно, ані з точки зору кольорів.

Головна сторінка містить декілька фотографій дитячого садочку і навігаційне меню (рис. 2.1).

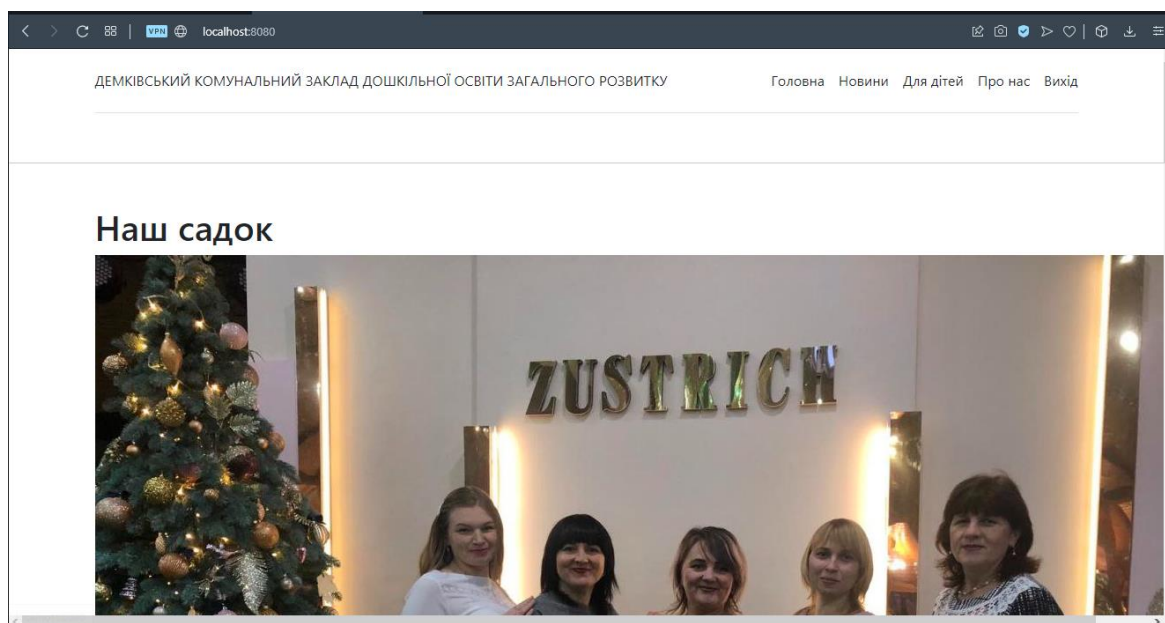


Рисунок 2.1 — Головна сторінка

Сторінка новин також виконана в стилі мінімалізму і містить окремі блоки для кожної новини. На рисунку 2.2 зображено сторінку новин з однією тестовою новиною з боку адміністратора (користувач не має можливості додавати новини).

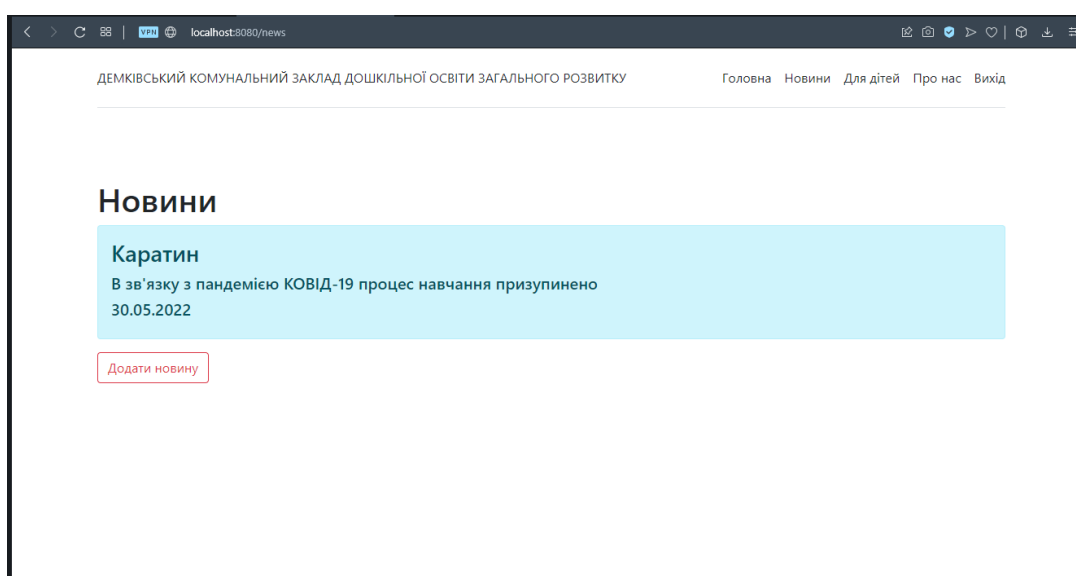


Рисунок 2.2 — Сторінка новин

Сторінка для дітей організована таким самим чином за виключенням того, що назва розділу є посиланням на контент, будь то гра чи відео на відеохостингу (рис. 2.3.).

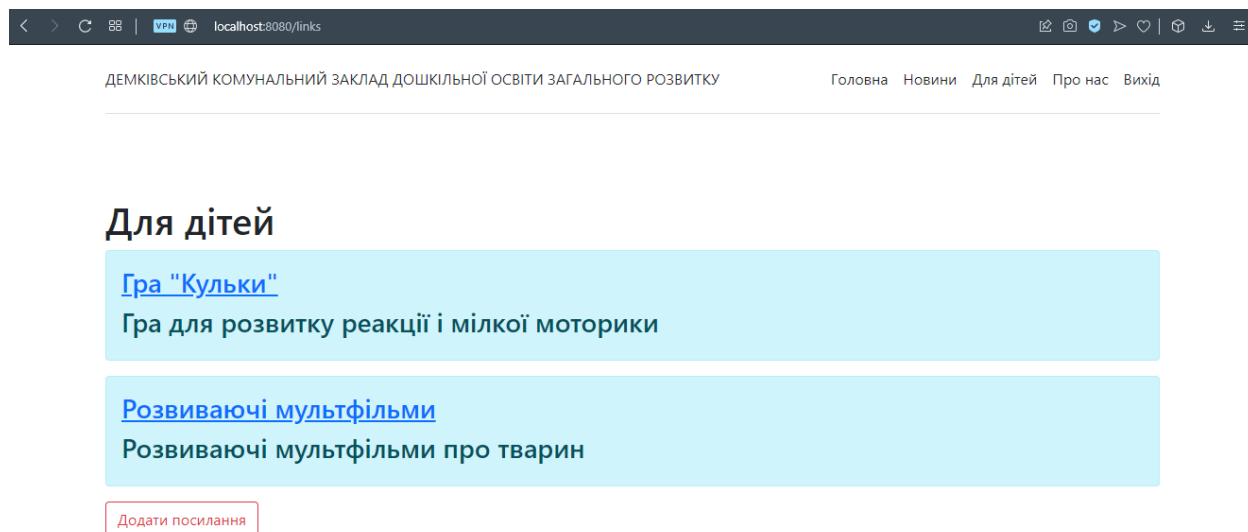


Рисунок 2.3 — Сторінка «Для дітей»

Сторінка «Про нас» організована схожим чином і містить виключно необхідну інформацію, а саме адресу, пошту, ім'я директора тощо (рис. 2.4.)

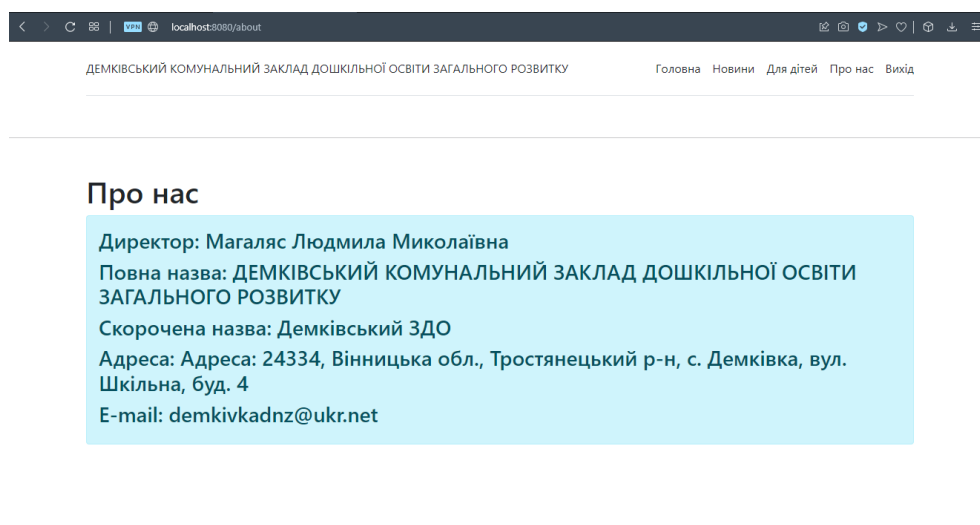
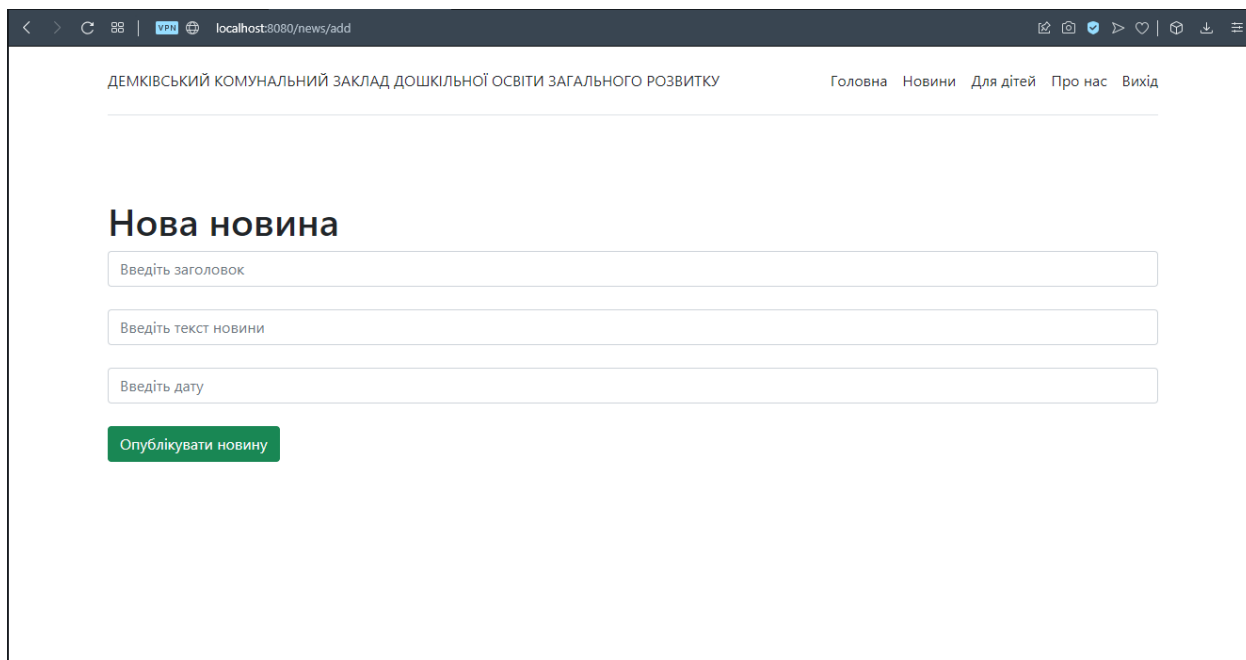


Рисунок 2.4 — Сторінка «Про нас»

Окрім цього, ресурс містить 2 сторінки, доступні лише адміністратору, а саме сторінку додавання новини (рис. 2.5) і сторінку додавання посилання (рис.2.6).

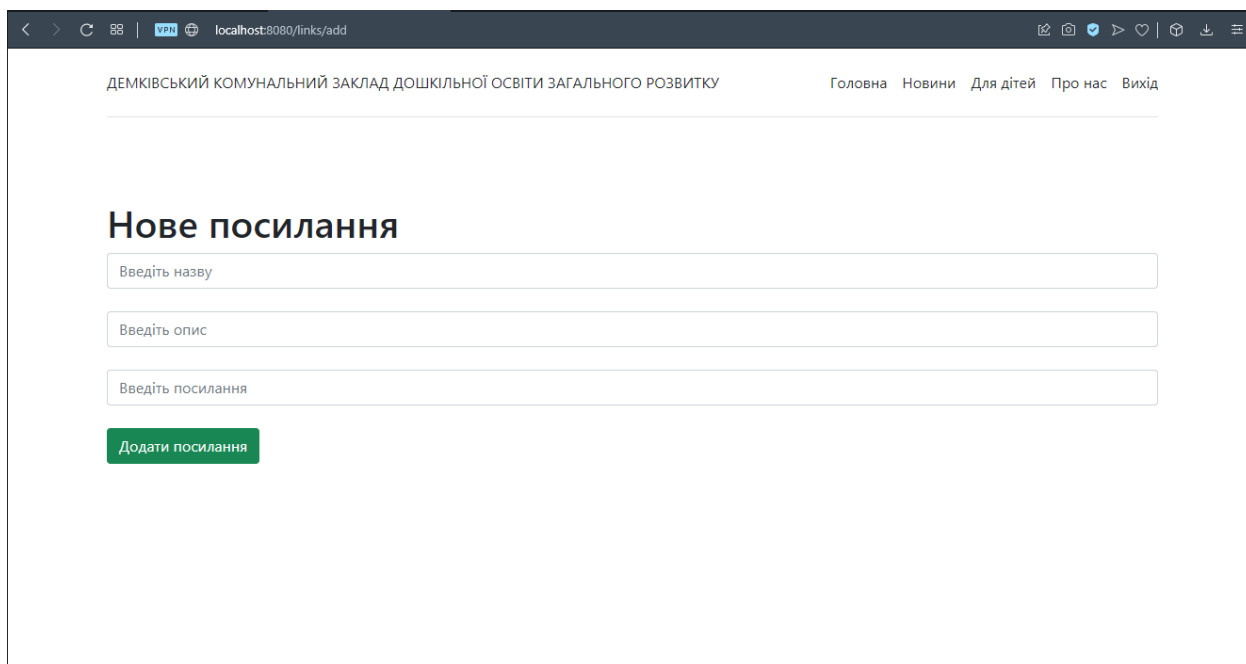


ДЕМКІВСЬКИЙ КОМУНАЛЬНИЙ ЗАКЛАД ДОШКІЛЬНОЇ ОСВІТИ ЗАГАЛЬНОГО РОЗВИТКУ

Головна Новини Для дітей Про нас Вихід

Нова новина

Рисунок 2.5 — Сторінка додавання новини



ДЕМКІВСЬКИЙ КОМУНАЛЬНИЙ ЗАКЛАД ДОШКІЛЬНОЇ ОСВІТИ ЗАГАЛЬНОГО РОЗВИТКУ

Головна Новини Для дітей Про нас Вихід

Нове посилання

Рисунок 2.6 — Сторінка створення посилання

Також сайт містить 2 загальнодоступні сторінки, сторінки авторизації (рис.2.7) і реєстрації (рис. 2.8).

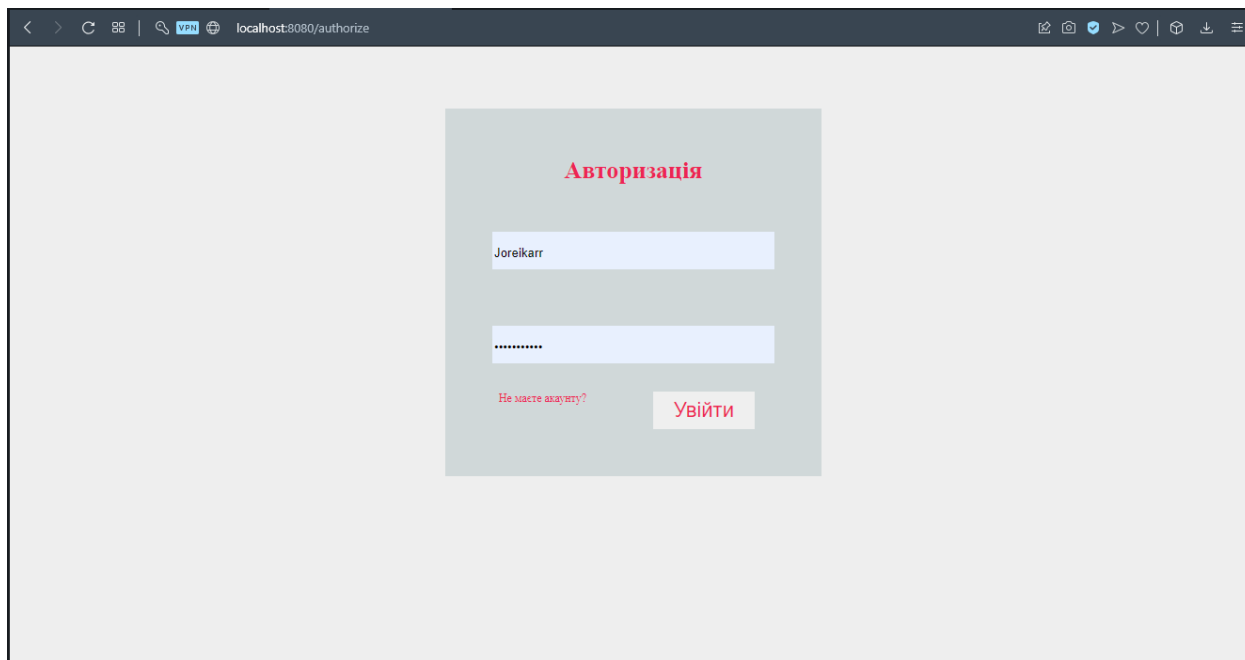


Рисунок 2.7 — Сторінка авторизації

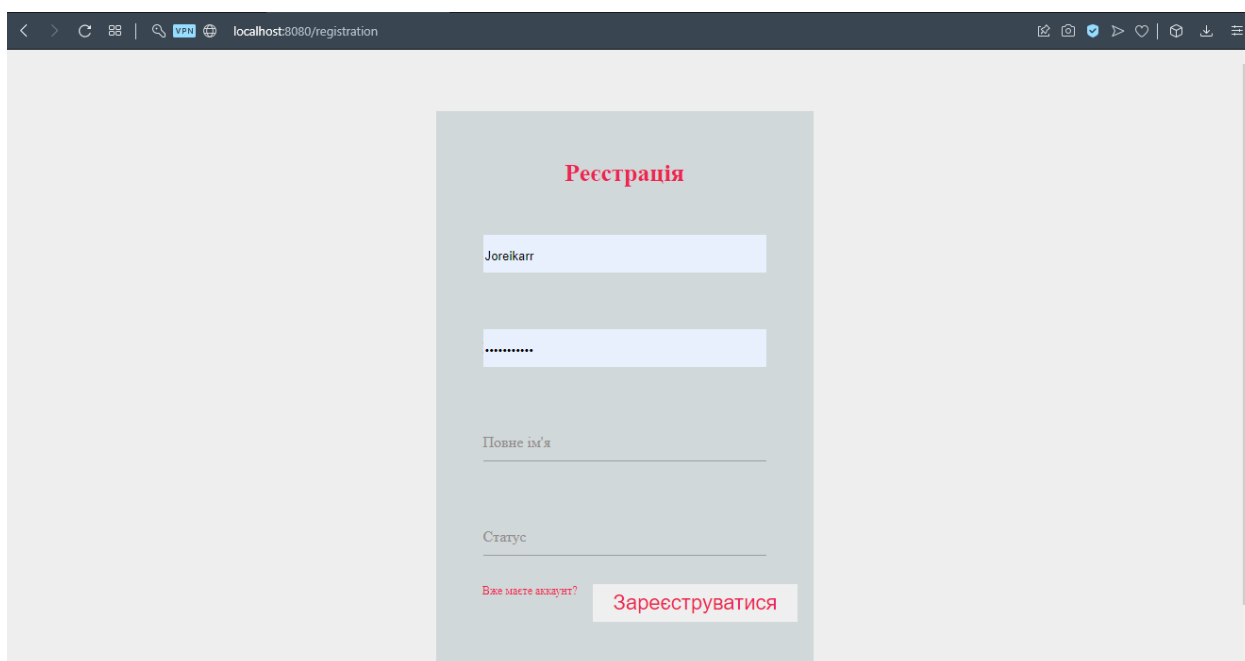


Рисунок 2.8 — Сторінка реєстрації

2.3 Аналіз варіантів використання веб-системи

Для більш детального розуміння роботи системи слід проаналізувати варіанти її активного використання.

Для цього на рисунку 2.9 зображено діаграму варіантів діяльності.

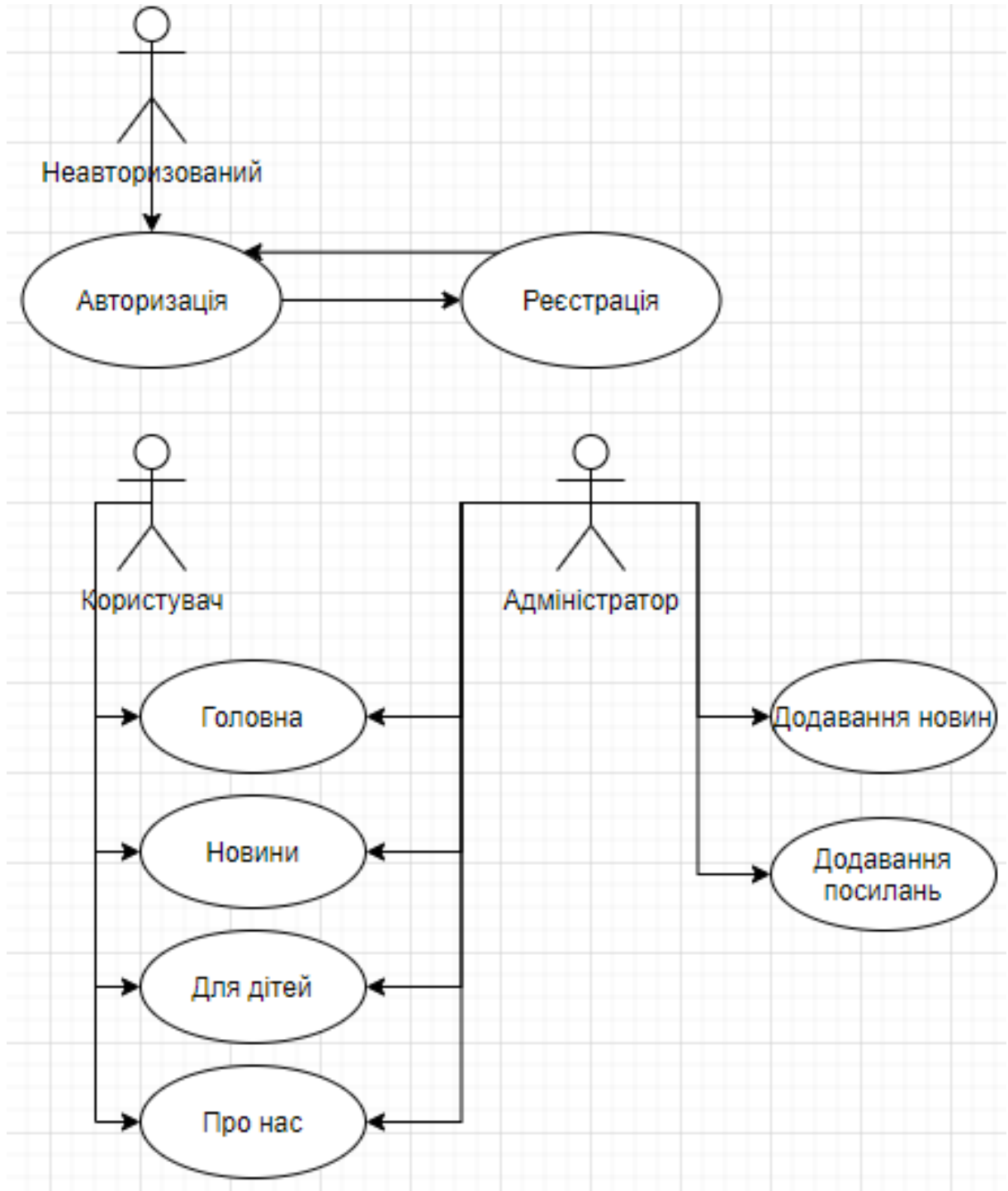


Рисунок 2.9 — Діаграма варіантів використання

2.4 Розробка методу і моделі роботи веб-системи

Розроблено модель роботи веб-системи з використанням UML діаграми діяльності (рис. 2.10), яка акумулює базові модулі системи і презентує систему взаємозв'язків між ними.

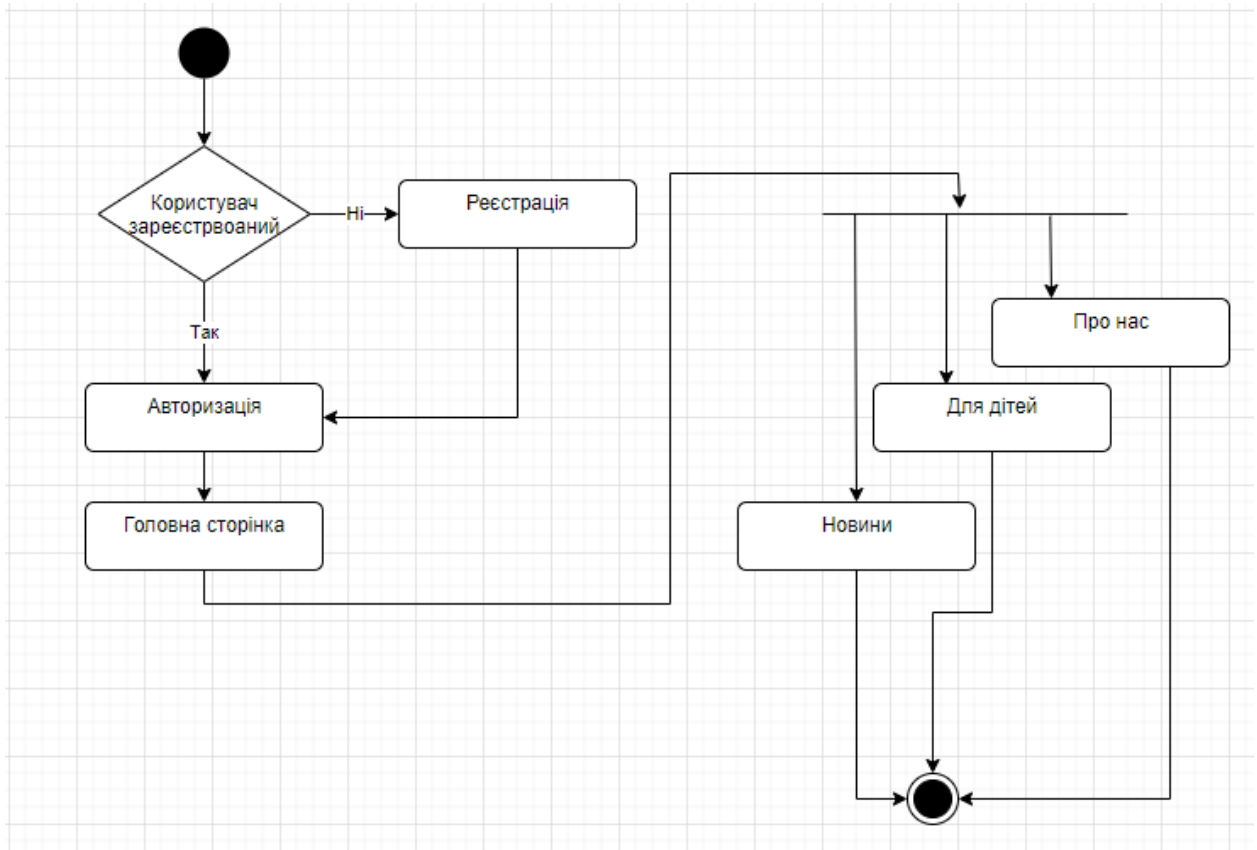


Рисунок 2.10 — Діаграма діяльності

Модель роботи веб-системи закладу дошкільної освіти спрямована на покращення комплексної оцінки та підтримки розвитку дитини з використанням елементів розвиваючих вправ і матеріалів, які зберігаються у пункті меню Для дітей.

Метод комплексної оцінки та підтримки розвитку дитини в середовищі веб-системи використовує елементи розвиваючих вправ і матеріалів для дітей. Метод акумулює послідовність дій:

1. Авторизація у веб-системі (користувачів та адміністратора).
2. Перехід в особистий кабінет.

3. Перегляд інформації на сайті.
4. У вкладці – дітям пошук матеріалів для розвитку в наочній формі з елементами інтерактиву.
5. У вкладці – дітям є доступ до розвиваючого ігрового контенту: навчальних та логічних ігор для дітей дошкільного віку.
6. Збереження результатів активності й успіху дитини при використанні інтерактивних навчально-ігрових матеріалів сайту.
7. Оновлення і корегування адміністратором інформації на сайті.

Розміщення інтерактивних навчальних матеріалів і розвиваючих ігор для дітей дошкільного віку на сайті дитячого садочка дозволить малюкам в ігровій формі отримати навчально-розвиваючий контент, наблизить дітей до використання сучасних інтернет-технологій, дозволить виконувати нескладні цікаві ігрові завдання навчально-логічного орієнтування, зберігатиме рейтинг активності дитини на сайті. Це слугуватиме, в свою чергу, ефективним рекламним засобом, привабить відвідувачів і зацікавить користувачів, а також підвищить рівень відвідуваності сайту.

2.5 Висновок

У другому розділі було проведено аналіз інформаційного забезпечення веб-системи. Розроблено графічний інтерфейс системи. Проаналізовано варіанти використання веб-ресурсу. Розроблено метод роботи веб-системи та модель системи з допомогою UML діаграми діяльності. Метод комплексної оцінки та підтримки розвитку дитини з використанням спеціалізованої веб-системи орієнтується на використання елементів розвиваючих вправ і матеріалів для дітей, що підвищить ефективність взаємодії суб'єктів навчально-виховного процесу дитячого садка.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СИСТЕМИ

3.1 Варіантний аналіз та обґрунтування вибору засобів реалізації

Java — це високорівнева, заснована на класах, об'єктно-орієнтована мова програмування, розроблена для того, щоб мати якомога менше залежностей реалізації. Це мова програмування загального призначення, розроблена для того, щоб програмісти могли писати один раз, запускати в будь-якому місці (WORA), що означає, що скомпільований код Java може працювати на будь-якій Java-сумісній платформі без перекомпіляції. Програми Java зазвичай компілюються у байт-код, який може працювати на будь-якій віртуальній машині Java (JVM), незалежно від базової архітектури комп'ютера. Синтаксис Java подібний до C і C++, але він має менше низькорівневих інструментів, ніж будь-який з них. Середовище виконання Java надає динамічні функції (наприклад, відображення та зміна коду під час виконання), які здебільшого не зустрічаються в традиційних компільованих мовах. Станом на 2019 рік Java є однією з найпопулярніших мов програмування на GitHub [22], особливо для веб-додатків клієнт-сервер дев'ятьма мільйонами розробників [23].

Java спочатку була розроблена Джеймсом Гослінгом з Sun Microsystems і випущена в травні 1995 року як основний компонент Java-платформи Sun Microsystems. Справжні та довідкові компілятори Java, віртуальні машини та бібліотеки класів спочатку були випущені Sun за власною ліцензією. У травні 2007 року Sun переліцензувала більшість своєї технології Java лише за ліцензією GPL-2.0 відповідно до специфікації Java Community Process. Oracle надає власну віртуальну машину HotSpot Java, але офіційною довідковою реалізацією є OpenJDK JVM, яка є безкоштовним програмним забезпеченням з відкритим кодом, що використовується більшістю розробників, і JVM за замовчуванням майже у всіх дистрибутивах Linux.

Станом на березень 2022 року Java 18 є останнім випуском, тоді як Java 17, 11 і 8 є поточними випусками для довгострокової підтримки (LTS). Oracle випустила останнє загальнодоступне застаріле оновлення Java 8 LTS для комерційного використання в січні 2019 року, але в іншому випадку підтримуватиме Java 8 та загальнодоступні оновлення для особистого використання на невизначений термін. Інші виробники почали пропонувати OpenJDK версії 8 і 11 безкоштовно, і вони все ще отримують оновлення безпеки та багато іншого.

Через невирішені проблеми безпеки в старих версіях Oracle (та інші) рекомендують видалити застарілі та непідтримувані версії Java [24]. Oracle рекомендує своїм користувачам негайно перейти до підтримуваного випуску, наприклад одного з випусків LTS .

Однією з цілей дизайну Java є переносимість, це означає, що програми , написані для платформи Java, повинні працювати таким чином на будь-якій комбінації апаратного забезпечення та операційних систем з відповідною підтримкою часу виконання. Це досягається шляхом компіляції коду Java у проміжне уявлення, яке називається байт-кодом Java, а не прямий машинний код, своєрідний для архітектури. Інструкції байт-коду Java подібні до машинного коду, але вони розроблені для виконання віртуальною машиною (VM), написаною спеціально для апаратного забезпечення.

Кінцеві користувачі зазвичай використовують Java Runtime Environment (JRE), встановлену на їхніх пристроях, для автономних програм Java або веб-браузерів для Java-апплетів.

Стандартна бібліотека надає загальний спосіб доступу до функціональних можливостей хоста, таких як графіка, мережа та потокове передавання..

Використання загального байт-коду може значно спростити перенесення. Однак накладні витрати на інтерпретацію байт-коду в машинних інструкціях змушують інтерпретатори майже завжди працювати повільніше, ніж рідні виконувані файли. Раніше були представлені компілятори Just-in-time (JIT), які

компілюють байт-код у машинний код під час виконання. Компілятор Java Hotspot насправді є двома компіляторами в одному; і GraalVM (наприклад, включений до Java 11, але видалений з Java 16), який дозволяє багаторівневу компіляцію [25]. Сама Java є незалежною від платформи та адаптованою до певної платформи, на якій вона працює, використовуючи віртуальну машину Java (JVM) для перетворення байт-коду Java на машинну мову платформи.

Відомо, що програми, написані на Java, повільніші та потребують більше пам'яті, ніж програми, написані на C++ [26]. Однак із запровадженням компіляції «точно вчасно» для Java-додатків у 1997/1998 рр. швидкість Java-додатків значно зросла. У Java 1.1 додані мовні функції, які підтримують кращий аналіз коду (наприклад, внутрішні класи, класи `StringBuilder`, додаткові оператори тощо), а також оптимізації у віртуальних машинах Java (таких як HotSpot), які стали JVM за замовчуванням у 2000 Sun. . У Java 1.5 продуктивність була покращена шляхом додавання пакета `java.util.concurrent` (включаючи безперешкодні реалізації `ConcurrentMaps` та інших багатоядерних колекцій), а також покращена в Java 1.6.

Java використовує автоматичний збірник сміття для керування пам'яттю протягом усього життя об'єкта. Програміст вирішує, коли створити об'єкт, а середовище виконання Java відповідає за відновлення пам'яті, коли об'єкт більше не використовується. Як тільки зв'язок об'єкта все ще існує, недоступна пам'ять стає придатною для автоматичного збирання сміття. Такі речі, як витік пам'яті, все ще можуть статися, якщо код програміста містить посилання на об'єкт, який більше не потрібен, як правило, коли об'єкт, який більше не потрібен, зберігається в контейнері, який все ще використовується. Виняток нульового покажчика створюється, якщо метод викликається для неіснуючого об'єкта [27].

Одна з ідей, що лежить в основі моделі автоматичного управління пам'яттю Java, полягає в тому, що програмісти можуть бути звільнені від ручного керування пам'яттю. У деяких мовах пам'ять, яка використовується для

створення об'єктів, або неявно виділяється в стеку, або явно виділяється та звільняється з купи. В останньому випадку відповідальність за управління пам'яттю належить програмістові. Якщо програма не звільняє об'єкт, то відбувається витік пам'яті. Якщо програма намагається отримати доступ або звільнити пам'ять, яка вже була звільнена, то результати будуть невизначеними та непередбачуваними, а програма може стати нестабільною або вийти з ладу. Це можна частково вирішити за допомогою розумних показників, але вони додають вартість і складність. Зауважте, що збір сміття не запобігає витoku логічної пам'яті, тобто пам'яті, на яку завжди посилаються, але ніколи не використовуються.

Збір сміття може відбуватися в будь-який час. В ідеалі це станеться, коли програма неактивна. Він гарантовано спрацює, якщо в купі недостатньо вільної пам'яті для виділення нового об'єкта; це може призвести до зупинки програми. Явне керування пам'яттю неможливе в Java.

Java не підтримує арифметику вказівників у стилі C/C++, де арифметичні операції можна виконувати над адресами об'єктів (наприклад, додавати або віднімати зміщення). Це дозволяє кошику переміщувати об'єкти, на які посилаються, та гарантувати безпеку.

Як і в C++ та багатьох інших об'єктно-орієнтованих мовах, змінні примітивних типів даних Java зберігаються безпосередньо в полях (для об'єктів) або стеку (для методів), а не в купі, для непримітивних даних. тип (але див. аналіз `escape`). Це свідоме рішення, прийняте дизайнерами Java з міркувань продуктивності.

Java містить в собі кілька типів збирачів сміття. Починаючи з Java 9, HotSpot за замовчуванням використовує Garbage First Garbage Collector (G1GC). Однак є декілька інших збирачів сміття, які можна використовувати для керування смітцевою купою. G1GC достатньо для більшості програм Java. Паралельний збірник сміття раніше використовувався в Java 8.

Вирішення проблем з керуванням пам'яттю не звільняє програміста від тягаря належної обробки інших типів ресурсів, таких як підключення до мережі або бази даних, дескриптори файлів тощо, особливо коли виникають винятки.

Jakarta Servlet (раніше відомий як Java Servlet) — це програмний компонент Java, який розширює функціональні можливості сервера. Хоча сервлети можуть відповідати на багато типів запитів, вони зазвичай реалізують веб-контейнер для розміщення веб-програми на веб-сервері і, таким чином, кваліфікуються як веб-API сервлета на стороні сервера. Такі веб-сервлети подібні до того, що Java використовується для інших технологій динамічного веб-контенту, таких як PHP і ASP.NET.

Сервлет Jakarta обробляє або зберігає клас Java в Jakarta EE, який відповідає API сервлетів Jakarta, стандарту для реалізації класів Java, які відповідають на запити. Сервлети в принципі можуть спілкуватися за будь-яким протоколом клієнт-сервер, але найчастіше вони використовуються з HTTP. Таким чином, «сервлет» часто використовується як скорочення «сервлет HTTP». Таким чином, розробник програмного забезпечення може використовувати сервлет для додавання динамічного вмісту на веб-сервер за допомогою платформи Java. Згенерований вміст зазвичай являє собою HTML, але можуть бути й інші дані, такі як XML і частіше JSON. Сервлети можуть підтримувати стан змінних сеансу в багатьох транзакціях сервера за допомогою файлів cookie HTTP або зіставлення URL-адрес.

API Jakarta Servlet певною мірою було замінено двома стандартними технологіями Java для веб-сервісів:

- веб-сервіси Jakarta RESTful Web Services (JAX-RS 2.0), корисні для служб AJAX, JSON і REST, а також
- Веб-служби Джакарти XML (JAX-WS), корисні для веб-служб SOAP.

Для розгортання та запуску сервлета необхідно використовувати веб-контейнер. Веб-контейнер (також відомий як контейнер сервлетів) по суті є компонентом веб-сервера, який взаємодіє із сервлетами. Веб-контейнер

відповідає за керування життєвим циклом сервлетів, зіставлення URL-адреси з певним сервлетом і забезпечення того, що запитувач URL-адрес має правильні права доступу.

API сервлетів, що міститься в ієрархії пакетів Java `javax.servlet`, визначає очікувані взаємодії веб-контейнера та сервлета.

Сервлет - це об'єкт, який отримує запит і генерує відповідь на основі цього запиту. Базовий пакет сервлетів визначає об'єкти Java для представлення запитів і відповідей сервлетів, а також об'єкти, що відображають параметри конфігурації сервлета та середовище виконання. Пакет `javax.servlet.http` визначає HTTP-специфічні підкласи загальних елементів сервлетів, включаючи об'єкти керування сеансом, які відстежують численні запити та відповіді між веб-сервером і клієнтом. Сервлети можуть бути запаковані у файл WAR як веб-додаток.

Сервлети можуть бути автоматично створені з Jakarta Server Pages (JSP) компілятором Jakarta Server Pages. Різниця між сервлетами і JSP полягає в тому, що сервлети зазвичай вбудовують HTML в код Java, тоді як JSP вбудовують код Java в HTML. Хоча пряме використання сервлетів для генерації HTML (як показано в прикладі нижче) стало рідкістю, веб-фреймворк MVC вищого рівня в Jakarta EE (JSF) все ще явно використовує технологію сервлетів для низькорівневої обробки запитів/відповідей через `FacesServlet`. . Дещо давнішим є використання сервлетів у поєднанні з JSP у шаблоні, який називається «Модель 2», який є різновидом контролера `model-view-controller`.

Поточна версія Servlet 5.0.Jakarta Server Pages (JSP; раніше `JavaServer Pages`) — це набір технологій, які допомагають розробникам програмного забезпечення створювати динамічно створювані веб-сторінки на основі HTML, XML, SOAP або інших типів документів. Запущений у 1999 році компанією Sun Microsystems, JSP схожий на PHP і ASP, але використовує мову програмування Java.

Для розгортання та запуску Jakarta Server Pages потрібен веб-сервер, сумісний із контейнером сервлетів, таким як Apache Tomcat або Jetty.

З архітектурної точки зору, JSP можна розглядати як високорівневу абстракцію сервлетів Java. JSP транслюються в сервлети під час виконання, тому JSP є сервлетом; кожен сервлет JSP кешується і використовується повторно, доки вихідний JSP не буде змінено.

Серверні сторінки Jakarta можна використовувати окремо або як компонент перегляду серверної моделі-вигляду-контролера, як правило, з JavaBeans як моделлю і сервлетами Java (або структурою, як-от Apache Struts) як контролером. Це тип архітектури Model 2.

JSP дозволяє перемежовувати код Java і певні заздалегідь визначені дії зі статичним вмістом веб-розмітки, таким як HTML. Отримана сторінка компілюється і виконується на сервері для доставки документа. Скомпільовані сторінки, а також будь-які залежні бібліотеки Java містять байт-код Java, а не машинний код. Як і будь-яка інша програма .jar або Java, код має виконуватися на віртуальній машині Java (JVM), яка взаємодіє з операційною системою хоста сервера, щоб забезпечити абстрактне, нейтральне для платформи середовище.

JSP зазвичай використовуються для доставки документів HTML і XML, але за допомогою OutputStream вони також можуть доставляти інші типи даних.

Веб-контейнер створює неявні об'єкти JSP, такі як запит, відповідь, сесія, програма, конфігурація, сторінка, pageContext, вихід і виняток. JSP Engine створює ці об'єкти на етапі перекладу.

Компілятор сторінок JavaServer — це програма, яка аналізує JSP і перетворює їх у виконувані сервлети Java. Програма такого типу зазвичай вбудовується в сервер додатків і запускається автоматично під час першого доступу до JSP, але сторінки також можуть бути попередньо скомпільовані для кращої продуктивності або скомпільовані як частина процесу збірки для перевірки на наявність помилок.

Деякі контейнери JSP підтримують налаштування того, як часто контейнер перевіряє часові позначки файлу JSP, щоб побачити, чи змінилася сторінка. Зазвичай ця позначка часу встановлюється на короткий інтервал (можливо, секунди) під час розробки програмного забезпечення та довший інтервал (можливо, хвилини або навіть ніколи) для розгорнутого веб-додатка.

IntelliJ IDEA — це інтегроване середовище розробки (IDE), яке написано на Java для розробки комп'ютерного програмного забезпечення. Був розроблений JetBrains (раніше IntelliJ) і доступний як ліцензований та приватний комерційний випуск спільноти Apache 2. Їх обидва можна використовувати для розвитку бізнесу.

Перша версія IntelliJ IDEA була випущена в січні 2001 року і була однією з перших Java IDE з вбудованою розширеною навігацією та рефакторингом коду.

У звіті InfoWorld 2010 IntelliJ отримав найвищу оцінку в центрі тестування чотирьох найкращих інструментів програмування Java: Eclipse, IntelliJ IDEA, NetBeans і JDeveloper.

У грудні 2014 року Google випустила версію 1.0 Android Studio, IDE з відкритим кодом для додатків Android, засновану на випуску IntelliJ IDEA спільноти з відкритим кодом. Інші середовища розробки, засновані на платформі IntelliJ, включають AppCode, CLion, DataGrip, GoLand, PhpStorm, PyCharm, Rider, RubyMine, WebStorm і MPS.

IDE надає певні функції, такі як додавання коду шляхом аналізу коду, навігація по коду, яка дозволяє перейти безпосередньо до класів або декларацій у коді, рефакторинг коду, налагодження коду [28], підкладка та виправлення невідповідностей з рекомендаціями. Налаштування...

Вбудовані інструменти та інтеграція. IDE забезпечує інтеграцію з такими інструментами для складання/пакування, як дрібний, bower, gradle і SBT. Він підтримує системи контролю версій, такі як Git, Mercurial, Perforce і SVN. До таких баз даних, як Microsoft SQL Server, Oracle, PostgreSQL, SQLite і MySQL,

можна отримати доступ безпосередньо з Ultimate IDE через вбудований DataGrip.

IntelliJ підтримує плагіни, які дозволяють додавати додаткові функції до IDE. Плагіни можна завантажити та встановити з веб-сайту IntelliJ Plugin Store або за допомогою вбудованого засобу пошуку та встановлення плагінів IDE. Кожна версія має окрему бібліотеку плагінів, і станом на 2019 рік у Community та Ultimate є понад 3000 плагінів.

MySQL — це система управління реляційною базою даних з відкритим вихідним кодом (RDBMS). Її ім'я є комбінацією «My» (дочка співзасновника Майкла Віденіуса) і «SQL» (скорочення від Structured Query Language). Реляційна база даних організовує дані в одній або кількох таблицях даних, де типи даних можуть бути пов'язані; ці відносини допомагають структурувати дані. SQL — це мова, яка використовується програмістами для створення, зміни та вилучення даних з реляційних баз даних, а також для контролю доступу користувачів до баз даних. На додаток до реляційних баз даних і баз даних SQL, такі бази даних, як MySQL, працюють з операційними системами, щоб інтегрувати реляційні бази даних у сховище комп'ютера, керувати користувачами, забезпечувати доступ до мережі, а також забезпечувати цілісність і резервне копіювання баз даних.

MySQL є безкоштовним програмним забезпеченням з відкритим вихідним кодом відповідно до умов GNU General Public License і доступне за різними ліцензіями власності. MySQL належить і спонсорується шведською компанією MySQL AB, що придбала Sun Microsystems (тепер Oracle Corporation). У 2010 році, коли Oracle придбала Sun, Widenius розширив проект MySQL з відкритим кодом, щоб створити MariaDB. [9]MySQL має окремий клієнт, який дозволяє користувачам безпосередньо взаємодіяти з базою даних MySQL за допомогою SQL, але частіше MySQL використовується з іншими програмами для реалізації програм, які потребують функціональності реляційної бази даних. MySQL є компонентом стеку веб-додатків LAMP (та інших), який є акронімом для Linux,

Apache, MySQL, Perl/PHP/Python. MySQL використовується багатьма веб-додатками баз даних, включаючи Drupal, Joomla, phpBB і WordPress. MySQL також використовується багатьма популярними веб-сайтами, включаючи Facebook, Flickr, MediaWiki, Twitter та YouTube.

MySQL написаний на C і C++. Його синтаксичний аналізатор SQL написаний на yacc, але він використовує саморобний lexer. [16] MySQL працює на багатьох системних платформах, включаючи AIX, BSDi, FreeBSD, HP-UX, ArcaOS, eComStation, IBM і, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp , QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos і Tru64. OpenVMS також має порт MySQL.

Серверне програмне забезпечення MySQL і клієнтські бібліотеки використовують дистрибутив з подвійною ліцензією. Вони надаються відповідно до GPL версії 2 або вашої власної ліцензії.

Підтримку можна знайти в офіційному посібнику. Крім того, різноманітні канали та форуми IRC надають безкоштовну підтримку. Oracle пропонує платну підтримку через свій продукт MySQL Enterprise. Вони відрізняються за обсягом і ціною. Крім того, існує багато сторонніх організацій, які надають підтримку та послуги.

MySQL отримав позитивні відгуки, рецензенти відзначили, що він «в середньому працює дуже добре», що є інтерфейс розробника, а документація (не кажучи вже про реальний відгук через веб-сайт тощо) дуже, дуже хороша. «Він також був протестований як «швидкий, стабільний і справжній багатокористувацький, багатопоточний сервер баз даних SQL».

MySQL можна вручну зібрати та встановити з джерела, але частіше він встановлюється з бінарних пакетів, якщо не потрібні спеціальні налаштування. У більшості дистрибутивів Linux система керування пакетами полегшує завантаження та встановлення MySQL, хоча для налаштування параметрів безпеки та оптимізації зазвичай потрібна додаткова конфігурація.

Незважаючи на те, що MySQL починався як альтернатива більш потужним власним базам даних, він поступово розвивався для підтримки потреб все більшого і більшого масштабу. Він як і раніше найчастіше використовується в малих і середніх розгортаннях з одним сервером, або як компонент у веб-додатку на основі LAMP, або як окремий сервер бази даних. Значна частина привабливості MySQL походить від її відносної простоти та зручності використання, що забезпечується екосистемою інструментів з відкритим кодом, наприклад phpMyAdmin. У середньому діапазоні MySQL можна масштабувати, розгортаючи його на більш потужному обладнанні, такому як багатопроцесорний сервер з гігабайтами пам'яті.

Однак існують обмеження щодо того, наскільки продуктивність може розширюватися на одному сервері («збільшення масштабу»), тому для більших масштабів для підвищення продуктивності та надійності потрібні багатосерверні розгортання MySQL («масштабування»). Типова конфігурація високого класу може включати потужну головну базу даних, яка обробляє операції запису даних і реплікується на кілька ведених, які обробляють усі операції читання. Головний сервер постійно передає події binlog на підключені підпорядковані пристрої, тому в разі збою підпорядкованого можна підвищити до нового головного, що мінімізує час простою. Подальшого покращення продуктивності можна досягти шляхом кешування результатів запитів до бази даних у пам'яті за допомогою memcached або розбиття бази даних на менші фрагменти, які називаються сегментами, які можуть бути розподілені між кількома розподіленими серверними кластерами [29].

3.2 Розробка бази даних

Перед розробкою програмного забезпечення необхідно розробити базу даних. При роботі з Spring Boot Framework база даних створюється засобами фреймворку через створення моделей і їх репозиторіїв.

Нижче подано код розроблених моделей і репозиторіїв (рис 3.1, 3.2, 3.3).

```

1 package com.joreijarr.studycontrol.models;
2
3 import ...
4
5 @Entity
6 public class Links {
7
8     @Id
9     @GeneratedValue(strategy = GenerationType.AUTO)
10    public Long link_id;
11    public String link_name, link_description, link_url;
12
13    public Links() {
14    }
15
16    public Links(String link_name, String link_description, String link_url) {
17        this.link_name = link_name;
18        this.link_description = link_description;
19        this.link_url = link_url;
20    }
21
22    public Long getLink_id() { return link_id; }
23
24    public void setLink_id(Long link_id) { this.link_id = link_id; }
25
26    public String getLink_name() { return link_name; }
27
28    public void setLink_name(String link_name) { this.link_name = link_name; }
29
30    public String getLink_description() { return link_description; }
31
32    public void setLink_description(String link_description) { this.link_description = link_description; }
33
34 }

```

Рисунок 3.1 — Сторінка Links.java

```

1 package com.joreijarr.studycontrol.models;
2
3 import ...
4
5 @Entity
6 public class Users {
7
8     @Id
9     @GeneratedValue(strategy = GenerationType.AUTO)
10    private Long user_id;
11
12    public Users() {
13    }
14
15    public Users(String user_login, String user_password, String user_fullname, String status) {
16        this.user_login = user_login;
17        this.user_password = user_password;
18        this.user_fullname = user_fullname;
19        this.status = status;
20    }
21
22    private String user_login;
23    private String user_password;
24    private String user_fullname;
25    private String status;
26
27    public String getStatus() { return status; }
28
29    public void setStatus(String status) { this.status = status; }
30
31    public Long getUser_id() { return user_id; }
32
33    public void setUser_id(Long user_id) { this.user_id = user_id; }
34
35    public String getUser_login() { return user_login; }
36
37    public void setUser_login(String user_login) { this.user_login = user_login; }
38
39    public String getUser_password() { return user_password; }
40
41    public void setUser_password(String user_password) { this.user_password = user_password; }
42
43    public String getUser_fullname() { return user_fullname; }
44
45    public void setUser_fullname(String user_fullname) { this.user_fullname = user_fullname; }
46
47 }

```

Рисунок 3.2 — Сторінка Users.java

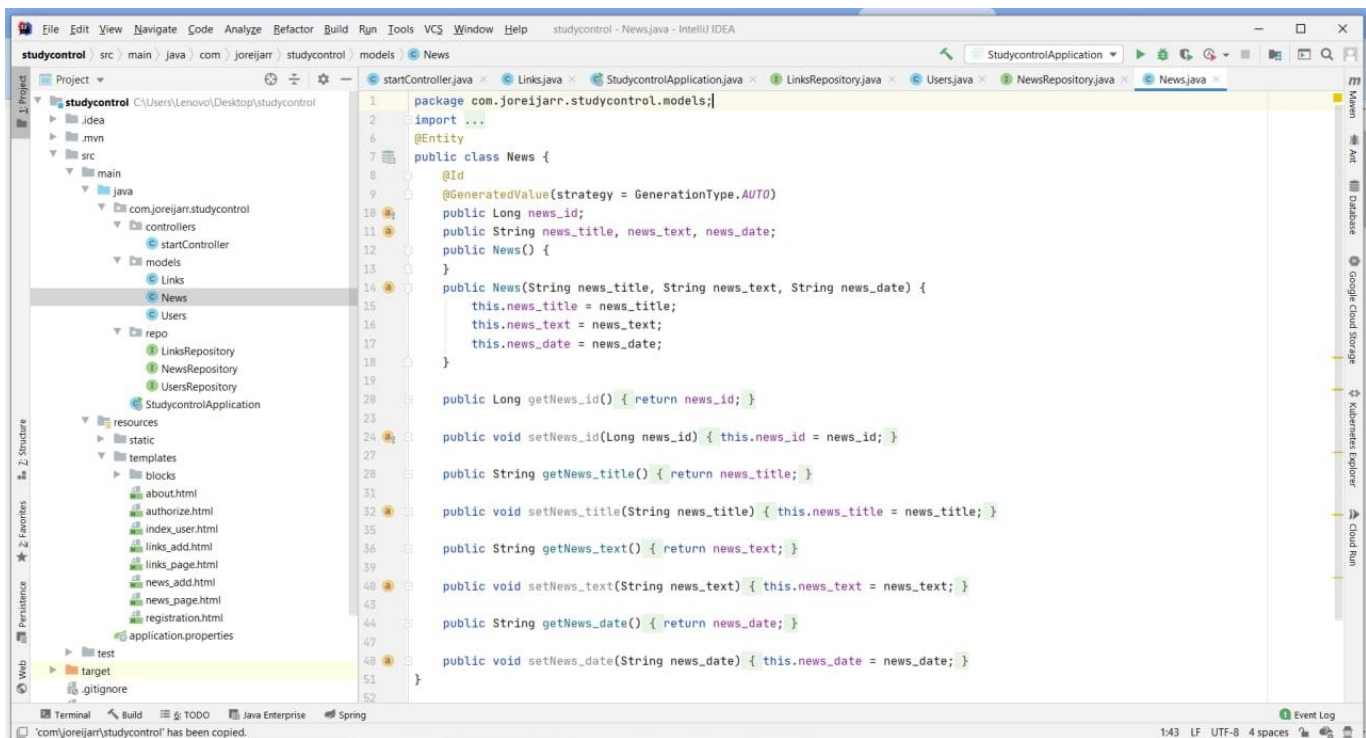


Рисунок 3.3 — Сторінка News.java

Всього створено три сутності: для зберігання користувачів, новин і посилань на ресурси для дітей.

3.3 Розробка механізмів авторизації і реєстрації

Механізм авторизації і реєстрації користувачів є одним з основних механізмів системи. Вся система реалізована механізмами Spring Boot Framework. Код подано нижче (рис 3.4,3.5)

```

C:\Users\PC\Desktop\Дипломна\app\studycontrol\src\main\java\com\jorejarr\studycontrol\controllers\startController.java - Notepad++
Файл Редагувати Пошук Вигляд Кодування Мова Налаштування Інструменти Макрос Виконати Плагіни Вікно ?
authorize.html index_user.html registration.html header_user.html StudycontrolApplication.java startController.java
36
37
38 @GetMapping("/exit")
39 public String exit(Model model)
40 {
41     current_user = null;
42     authorized = false;
43     return "redirect:/";
44 }
45
46
47
48
49 @GetMapping("/authorize")
50 public String start(Model model) {
51
52     model.addAttribute("title", "Авторизація");
53     if(!authorized)
54     {
55         return "authorize";
56     }
57     else
58     {
59         return "redirect:/";
60     }
61 }
62
63
64 @PostMapping("/authorize")
65 public String authorize(Model model, @RequestParam String user_login, @RequestParam String user_password)
66 {
67     Iterable<Users> users_it = usersRepository.findAll();
68     List<Users> users = new ArrayList<>();
69     users_it.forEach(users::add);
70     for (int i = 0; i < users.size(); i++)
71     {
72         if(users.get(i).getUser_login().equals(user_login) && users.get(i).getUser_password().equals(user_password))
73     }

```

Рисунок 3.4 — Код програми

```

C:\Users\PC\Desktop\Дипломна\app\studycontrol\src\main\java\com\jorejarr\studycontrol\controllers\startController.java - Notepad++
Файл Редагувати Пошук Вигляд Кодування Мова Налаштування Інструменти Макрос Виконати Плагіни Вікно ?
authorize.html index_user.html registration.html header_user.html StudycontrolApplication.java startController.java
72
73     if(users.get(i).getUser_login().equals(user_login) && users.get(i).getUser_password().equals(user_password))
74     {
75         authorized = true;
76         current_user = users.get(i);
77         break;
78     }
79     return "redirect:/";
80 }
81
82
83
84
85
86 @GetMapping("/registration")
87 public String registration(Model model)
88 {
89     model.addAttribute("title", "Реєстрація");
90     if(!authorized)
91     {
92         return "registration";
93     }
94     else
95     {
96         return "redirect:/";
97     }
98 }
99
100
101 @PostMapping("/registration")
102 public String registration(Model model, @RequestParam String user_login, @RequestParam String user_password,
103 @RequestParam String user_fullname, @RequestParam String status)
104 {
105     Users new_user = new Users(user_login, user_password, user_fullname, status);
106     usersRepository.save(new_user);
107     return "redirect:/authorize";
108 }

```

Рисунок 3.5 — Код програми

3.4 Розробка механізмів додавання і показу новин

Один з основних механізмів — це механізм показу і додавання новин.

Код даного механізму подано на рис 3.6 і 3.7.

```

140     @GetMapping("/news")
141     public String newsPage(Model model)
142     {
143         model.addAttribute("title", "Новини");
144         Iterable<News> news_it = newsRepository.findAll();
145         List<News> news = new ArrayList<>();
146         news_it.forEach(news::add);
147         model.addAttribute("currUser", current_user);
148         model.addAttribute("news", news);
149         return "news_page";
150     }
151

```

Рисунок 3.6 — Показ новин

```

178
179     @PostMapping("/news/add")
180     public String newsAdd_post(Model model,
181                               @RequestParam String news_title,
182                               @RequestParam String news_text,
183                               @RequestParam String news_date)
184     {
185         News news = new News(news_title, news_text, news_date);
186         newsRepository.save(news);
187         return "redirect:/news";
188     }
189
190

```

Рисунок 3.7 — Додавання новин

3.5 Висновок

У третьому розділі було обґрунтовано вибір мови програмування та технологій для розробки програмного продукту. Проаналізувавши потреби програмного продукту, було обрано мову програмування Java. Для зручності розробки графічного дизайну було обрано веб-платформу Spring Boot через можливість написання графічного інтерфейсу з використанням мови HTML. В

якості системи управління базою даних було обрано MySQL. Зі сторони програмної імплементації було розроблено модулі авторизації, реєстрації, додавання і показу новин і посилань для дітей.

4 ТЕСТУВАННЯ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

4.1 Огляд рішень для тестування сайтів

Тестування програмного забезпечення – це акт перевірки артефактів і поведінки програмного забезпечення, що тестується, шляхом перевірки та верифікації. Тестування програмного забезпечення також може забезпечити об'єктивне, незалежне уявлення про програмне забезпечення, що дає змогу підприємствам оцінити та зрозуміти ризики впровадження програмного забезпечення. Методи тестування включають, але не обмежуються:

- аналіз вимог до продукту щодо повноти та правильності в різних контекстах, таких як галузева перспектива, бізнес-перспектива, доцільність та життєздатність впровадження, зручність використання, продуктивність, безпека, міркування інфраструктури тощо.
- перегляд архітектури продукту та загального дизайну продукту
- робота з розробниками продуктів над удосконаленням техніки кодування, шаблонів проектування, тестів, які можна написати як частину коду на основі різних методів, таких як граничні умови тощо.
- виконання програми або програми з метою перевірки поведінки
- перевірка інфраструктури розгортання та пов'язаних скриптів та автоматизації
- брати участь у виробничій діяльності, використовуючи методи моніторингу та спостереження

Тестування програмного забезпечення може надати користувачам або спонсорам об'єктивну, незалежну інформацію про якість програмного забезпечення та ризик його збою.

Помилки програмного забезпечення виникають через наступний процес: Програміст робить помилку (помилку), що призводить до помилки (дефекту, помилки) у вихідному коді програмного забезпечення. Якщо ця помилка

виконана, у певних ситуаціях система дасть неправильні результати, що призведе до збою.

Не всі несправності обов'язково призведуть до збоїв. Наприклад, помилки в мертвому коді ніколи не призведуть до збоїв. Несправність, яка не виявила збоїв, може призвести до збою при зміні середовища. Приклади цих змін у середовищі включають програмне забезпечення, яке запускається на новій апаратній платформі комп'ютера, зміни вихідних даних або взаємодію з іншим програмним забезпеченням. Одна несправність може призвести до широкого спектру симптомів відмови.

Не всі помилки програмного забезпечення викликані помилками кодування. Одним із поширених джерел дорогих дефектів є прогалини у вимогах, тобто нерозпізнані вимоги, які призводять до помилок, пропущених розробником програми. Прогалини вимог часто можуть бути нефункціональними вимогами, такими як тестованість, масштабованість, ремонтпридатність, продуктивність та інші вимоги безпеки.

4.2 Тестування роботи програмного продукту

У тестуванні програмного забезпечення існує багато підходів. Огляди, покрокові інструкції або перевірки називаються статичним тестуванням, тоді як виконання запрограмованого коду з заданим набором тестових випадків називається динамічним тестуванням.

Статичне тестування часто є неявним, як-от коректура, а також, коли інструменти програмування/текстові редактори перевіряють структуру вихідного коду, а компілятори (попередні компілятори) перевіряють синтаксис і потік даних як статичний аналіз програми. Динамічне тестування відбувається під час запуску самої програми. Динамічне тестування може розпочатися до того, як програма буде на 100% завершена, щоб перевірити окремі розділи коду та застосувати до дискретних функцій або модулів. Типовими методами для них

є або використання заглушок/драйверів, або виконання з середовища налагодження.

Статичне тестування передбачає перевірку, тоді як динамічне також передбачає перевірку.

Пасивне тестування відноситься до дослідження поведінки системи без будь-якої взаємодії з програмним продуктом. На відміну від активного тестування, тестувальники не надають жодних тестових даних, а переглядають системні журнали та трасування. Вони шукають закономірності та конкретну поведінку, щоб приймати рішення. Це включає перевірку часу виконання та аналіз журналів у автономному режимі.

Дослідницьке тестування — це підхід до тестування програмного забезпечення, який коротко описується як одночасне навчання, розробка тесту та виконання тесту. Джем Канер, який ввів цей термін у 1984 році, визначає дослідницьке тестування як «стиль тестування програмного забезпечення, який підкреслює особисту свободу та відповідальність окремого тестувальника за постійну оптимізацію якості своєї роботи, розглядаючи тест-пов'язане навчання, дизайн тесту, виконання тесту та інтерпретація результатів тесту як взаємодопоміжні дії, які виконуються паралельно протягом усього проекту».

Методи тестування програмного забезпечення традиційно поділяються на тестування білого та чорного ящиків. Ці два підходи використовуються для опису точки зору, яку дотримується тестувальник під час розробки тестових випадків. До методології тестування програмного забезпечення також можна застосувати гібридний підхід, який називається тестуванням сірого ящика. Оскільки концепція тестування сірого ящика, яка розробляє тести на основі конкретних елементів дизайну, стає все більш популярною, це «довільне розходження» між тестуванням чорного та білого ящиків дещо зникло.

Результати тестування сайту наведені у таблиці 4.1

Таблиця 4.1 — Тестування додатку

№	Тест-кейс	Очікуваний результат	Отриманий результат
1	Невірні данні при авторизації	При введенні невірних даних система повідомляє користувача про те, що такого користувача не існує, або дані введені невірно	При введенні невірних даних система повідомляє користувача про те, що такого користувача не існує, або дані введені невірно
2	Пусті поля при авторизації	При спробі авторизації з пустими полями система повідомляє користувача про те, що ці поля необхідно заповнити.	При спробі авторизації з пустими полями система повідомляє користувача про те, що ці поля необхідно заповнити.
3	Неспівпадаючі паролі при реєстрації	При введенні неспівпадаючих паролів система повідомляє користувача про те, що паролі не співпадають.	При введенні неспівпадаючих паролів система повідомляє користувача про те, що паролі не співпадають.
4	Пусті поля при реєстрації	При спробі реєстрації з пустими полями система повідомляє користувача про те,	При спробі реєстрації з пустими полями система повідомляє користувача про те,

		що ці поля необхідно заповнити.	що ці поля необхідно заповнити.
5	Створення новини з пустими полями	При спробі створення новини з пустими полями система повідомляє користувача про те, що ці поля необхідно заповнити.	При спробі створення новини з пустими полями система повідомляє користувача про те, що ці поля необхідно заповнити.
6	Спроба перейти в панель адміністратора без наявності прав адміністратора	При спробі користувача без прав адміністратора перейти в панель адміністрування система повідомляє користувача, що він не є адміністратором і не може перейти на дану сторінку.	При спробі користувача без прав адміністратора перейти в панель адміністрування система повідомляє користувача, що він не є адміністратором і не може перейти на дану сторінку.

Результати тестування показують, що система повністю функціональна і готова до використання в реальних умовах.

4.3 Висновок

Проведено повноцінне тестування веб-сайту. Перевірка сайту показала, що сайт є зручним для користувача, має зручну навігаційну панель та зрозумілу логічну структуру. Під час перевірки за допомогою HTML- та CSS-валідаторів не виявлено помилок коду. В усіх браузерах сайт відображується однаково. Навігаційне меню працює без помилок.

ВИСНОВКИ

У бакалаврській дипломній роботі було розроблено веб-систему дитячого садочка з елементами розвиваючих вправ і матеріалів. Для розробки було використано середовище програмної розробки JetBrains IntelliJ Idea.

Під час виконання бакалаврської дипломної роботи було проаналізовано стан проблеми на сьогоднішній день. Було розглянуто основні аналоги програмного продукту. Було визначено їхні недоліки та порівняно з власним програмним продуктом. Базуючись на порівнянні базових характеристик аналогів, було сформульовано основні задачі бакалаврської дипломної роботи.

Під час аналізу технологій розробки було обґрунтовано вибір мови програмування Java та веб-фреймворку Spring Boot. Також було розглянуто переваги використання сервера баз даних MySQL для збереження даних користувачів.

Було розроблено інтерфейс веб-ресурсу. Розроблено метод і модель роботи веб-системи.

Тестування програми показало повну працездатність даного програмного продукту та відповідність поставленому технічному завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Binstock, Andrew (May 20, 2015). "Java's 20 Years of Innovation". Forbes. Archived from the original on March 14, 2016. Retrieved March 18, 2016. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.scribd.com/document/484532060/computer-class-notes-4>
2. Войтко В.В. Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини / В. В. Войтко, С. В. Бєвз, С. М. Бурбело, А. В. Маланчук // Матеріали LI Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2022) ВНТУ, Факультет інформаційних технологій та комп'ютерної інженерії, Секція програмного забезпечення [Електронний ресурс] – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15097/12733>
3. Barbara Liskov with John Guttag (2000). Program Development in Java - Abstraction, Specification, and Object-Oriented Design. USA, Addison Wesley. ISBN 9780201657685.
4. Java 5.0 added several new language features (the enhanced for loop, autoboxing, varargs and annotations), after they were introduced in the similar (and competing) C# language. Archived March 19, 2011, at the Wayback Machine Archived January 7, 2006, at the Wayback Machine. [Електронний ресурс] – Режим доступу до ресурсу: <https://codedocs.org/what-is/java-programming-language>
5. Gosling, James; McGilton, Henry (May 1996). "The Java Language Environment". Archived from the original on May 6, 2014. Retrieved May 6, 2014 [Електронний ресурс] – Режим доступу до ресурсу: [https://sw.wikipedia.org/wiki/Java_\(lugh_a_ya_programu\)](https://sw.wikipedia.org/wiki/Java_(lugh_a_ya_programu))
6. "The A-Z of Programming Languages: Modula-3". Computerworld.com.au. Archived from the original on January 5, 2009. Retrieved June 9, 2010.

- [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.coursehero.com/file/p4j0s962/Retrieved-February-8-2008-The-A-Z-of-Programming-Languages-Modula-3/>
7. Дитячий садочок « Смайл» [Электронний ресурс] – Режим доступу до ресурсу: <http://www.wizardsmile.com.ua/>
 8. Дитячий садок « Пролісок» [Электронний ресурс] – Режим доступу до ресурсу: <https://dytsadok.info/dytyachyj-sadok/37663376/>
 9. James Gosling, Bill Joy, Guy Steele, The Java Language Specification, Java Se 7 Edition 7th Revised ed. February 24, 2013, 672 p
 10. TechMetrix Research (1999). "History of Java" (PDF). Java Application Servers Report. Archived from the original (PDF) on December 29, 2010. The project went ahead under the name green and the language was based on an old model of UCSD Pascal, which makes it possible to generate interpretive code. [Электронний ресурс] – Режим доступу до ресурсу: <https://codedocs.org/what-is/java-programming-language>
 11. "Chapel spec (Acknowledgements)" (PDF). Cray Inc. October 1, 2015. Archived (PDF) from the original on February 5, 2016. Retrieved January 14, 2016. [Электронний ресурс] – Режим доступу до ресурсу: <https://codedocs.org/what-is/ada-programming-language>
 12. "Facebook Q&A: Hack brings static typing to PHP world". InfoWorld. March 26, 2014. Archived from the original on February 13, 2015. Retrieved January 11, 2015. [Электронний ресурс] – Режим доступу до ресурсу: <https://codedocs.org/what-is/java-programming-language>
 13. "Write once, run anywhere?". Computer Weekly. May 2, 2002. Archived from the original on August 13, 2021. Retrieved July 27, 2009. [Электронний ресурс] – Режим доступу до ресурсу: https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Write_once_run_anywhere.html

14. "1.2 Design Goals of the Java™ Programming Language". Oracle. January 1, 1999. Archived from the original on January 23, 2013. Retrieved January 14, 2013. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.oracle.com/java/technologies/java-se-support-roadmap.html>
15. Chan, Rosalie (January 22, 2019). "The 10 most popular programming languages, according to the 'Facebook for programmers'". Business Insider. Archived from the original on June 29, 2019. Retrieved June 29, 2019. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.insider.com/the-10-most-popular-programming-languages-according-to-github-2018-10>
16. "Why should I uninstall older versions of Java from my system?". Oracle. Archived from the original on February 12, 2018. Retrieved September 24, 2021. [Электронный ресурс] – Режим доступа до ресурсу: <https://boothclever.weebly.com/blog/old-versions-of-java>
17. "What is ECMA—and why Microsoft cares". ZDNet. Archived from the original on May 6, 2014. Retrieved May 6, 2014. [Электронный ресурс] – Режим доступа до ресурсу: [https://wikimili.com/en/Java_\(programming_language\)](https://wikimili.com/en/Java_(programming_language))
18. "JAVASOFT SHIPS JAVA 1.0". Archived from the original on March 10, 2007. Retrieved May 13, 2018. [Электронный ресурс] – Режим доступа до ресурсу: https://en.wikipedia.org/wiki/Java_%28programming_language%29
19. Object-oriented Programming with Java: Essentials and Applications. Tata McGraw-Hill Education. p. 34.
20. "Java Community Process website". Jcp.org. May 24, 2010. Archived from the original on August 8, 2006. Retrieved June 9, 2010. . [Электронный ресурс] – Режим доступа до ресурсу: <https://codedocs.org/what-is/java-software-platform>
21. "JAVAONE: Sun – The bulk of Java is open sourced". GrnLight.net. Archived from the original on May 27, 2014. Retrieved May 26, 2014. [Электронный

- ресурс] – Режим доступа до ресурсу:
<https://www.scribd.com/document/474683023/java>
22. McMillan, Robert (August 1, 2013). "Is Java Losing Its Mojo?". wired.com. Archived from the original on February 15, 2017. Retrieved March 8, 2017. [Электронный ресурс] – Режим доступа до ресурсу:
<https://umeshgoud611.blogspot.com/>
23. "JavaOne 2013 Review: Java Takes on the Internet of Things". [Электронный ресурс] – Режим доступа до ресурсу: www.oracle.com.
24. "Why should I uninstall older versions of Java from my system?". Oracle. Archived from the original on February 12, 2018. Retrieved September 24, 2021. [Электронный ресурс] – Режим доступа до ресурсу:
<https://github.com/ootaken/chocolatey/issues/11>
25. "Deep Dive Into the New Java JIT Compiler - Graal | Baeldung". www.baeldung.com. August 6, 2021. Retrieved October 13, 2021. [Электронный ресурс] – Режим доступа до ресурсу:
<https://zditect.com/code/java/deep-dive-into-the-new-java-jit-compiler.html>
26. "2 Changes in MySQL 5.5.62 (2018-10-22, General availability)". docs.oracle.com. Retrieved 28 April 2020. [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.mysql.com/doc/relnotes/mysql/5.6/en/>
27. Hoydalsvik, Geir (19 April 2018). "What's New in MySQL 8.0? (Generally Available)". Retrieved 10 October 2019. [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.mysql.com/blog-archive/whats-new-in-mysql-8-0-generally-available/>
28. "A Conversation with James Gosling – ACM Queue". Queue.acm.org. August 31, 2004. Archived from the original on July 16, 2015. Retrieved June 9, 2010. [Электронный ресурс] – Режим доступа до ресурсу:
<https://www.scribd.com/document/484532060/computer-class-notes-4>

29. "Database Sharding". Code Futures. Archived from the original on 16 January 2010. Retrieved 9 December 2009. [Электронный ресурс] – Режим доступа до ресурсу: <https://infogalactic.com/info/MySQL>

ДОДАТКИ

ДОДАТОК А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

УЗГОДЖЕНО
Директор Демківського ЗДО
Магальяс Л.М.
25 березня 2022 року

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
Романюк О.Н.
25 березня 2022 року

Технічне завдання
на бакалаврську дипломну роботу «Розробка веб-системи для
комплексної оцінки та підтримки розвитку дитини»
за спеціальністю
121 – Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:

_____ к.т.н., доц. В.В. Войтко

25 березня 2022 року

Виконала:

_____ студент гр. ЗПІ-186 А.В.Маланчук

25 березня 2022 року

Вінниця – 2022 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини»

Галузь застосування – веб-технології.

2. Підстава для розробки.

Завдання на роботу, яке затверджене на засіданні кафедри програмного забезпечення – протокол № 66 від 24 березня 2022 р.

3. Мета та призначення розробки.

Метою роботи є підвищення ефективності роботи закладу дошкільної освіти шляхом розробки спеціалізованої веб-системи з елементами розвиваючих вправ і матеріалами для дітей, що забезпечить ефективну взаємодію суб'єктів навчально-виховного процесу дитячого садка

Призначення роботи – комплексна оцінки та підтримка розвитку дитини.

4. Вихідні дані для проведення НДР

1. Binstock, Andrew (May 20, 2015). "Java's 20 Years of Innovation". Forbes. Archived from the original on March 14, 2016. Retrieved March 18, 2016. – p. 195.
2. "The A-Z of Programming Languages: Modula-3". Computerworld.com.au. Archived from the original on January 5, 2009. Retrieved June 9, 2010. – p. 155.
3. Chan, Rosalie (January 22, 2019). "The 10 most popular programming languages, according to the 'Facebook for programmers'". Business Insider. Archived from the original on June 29, 2019. Retrieved June 29, 2019. – p. 55.

4. Object-oriented Programming with Java: Essentials and Applications. Tata McGraw-Hill Education. – p. 34.

5. Технічні вимоги

Вхідні дані – інформація користувача, розвиваючі ігри та вправи для дітей; вихідні дані – загальні відомості про систему керування.

6. Конструктивні вимоги.

Дизайн системи має відповідати естетичним вимогам, ресурс повинен бути зручним у використанні.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до бакалаврської дипломної роботи;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

ДОДАТОК Б –Протокол перевірки на плагіат

ПРОТОКОЛ
ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини»

Тип роботи: БДР

Підрозділ : кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Войтко В.В.

Оригінальність	90,4
Схожість	9,6

Аналіз звіту подібності

■ **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 автора. Роботу направити на доопрацювання.

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 _____ Черноволик Г. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk

Автор роботи _____ Маланчук А.В.

Керівник роботи _____ Войтко В.В.

ДОДАТОК В – Акт впровадження

Акт впровадження (Демківський ДЗО)

АКТ

**про впровадження результатів бакалаврської дипломної роботи
«Розробка веб-системи для комплексної оцінки та підтримки розвитку
дитини»
студентки ВНТУ Маланчук Анни Василівни**

Результати бакалаврської роботи «Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини» студентки ВНТУ Маланчук А.В., що полягають в розробці:

- веб-системи як єдиного інформаційного освітнього простору навчально-виховного закладу;
- методу та моделі роботи веб-системи для комплексної оцінки та підтримки розвитку дитини;
- створенні умов мережевої підтримки веб-системи

впроваджені в «Демківському комунальному закладі дошкільної освіти загального розвитку».

Директор Демківського ЗДО

Магальяс Л.М.

10 червня 2022 року

ДОДАТОК Г – Лістинг програми

Links.java

```
package com.joreijarr.studycontrol.models;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Links {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Long link_id;
    public String link_name, link_description, link_url;

    public Links() {
    }

    public Links(String link_name, String link_description, String link_url) {
        this.link_name = link_name;
        this.link_description = link_description;
        this.link_url = link_url;
    }

    public Long getLink_id() {
        return link_id;
    }

    public void setLink_id(Long link_id) {
```

```
        this.link_id = link_id;
    }

    public String getLink_name() {
        return link_name;
    }

    public void setLink_name(String link_name) {
        this.link_name = link_name;
    }

    public String getLink_description() {
        return link_description;
    }

    public void setLink_description(String link_description) {
        this.link_description = link_description;
    }

    public String getLink_url() {
        return link_url;
    }

    public void setLink_url(String link_url) {
        this.link_url = link_url;
    }
}
```

Users.java

```
package com.joreijarr.studycontrol.models;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Users {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long user_id;

    public Users() {
    }

    public Users(String user_login, String user_password, String user_fullname, String status)
    {
        this.user_login = user_login;
        this.user_password = user_password;
        this.user_fullname = user_fullname;
        this.status = status;
    }

    private String user_login;
```

```
private String user_password;
private String user_fullname;
private String status;

public String getStatus() {
    return status;
}

public void setStatus(String staus) {
    this.status = staus;
}

public Long getUser_id() {
    return user_id;
}

public void setUser_id(Long user_id) {
    this.user_id = user_id;
}

public String getUser_login() {
    return user_login;
}

public void setUser_login(String user_login) {
    this.user_login = user_login;
}

public String getUser_password() {
    return user_password;
}
```

```
public void setUser_password(String user_password) {
    this.user_password = user_password;
}

public String getUser_fullname() {
    return user_fullname;
}

public void setUser_fullname(String user_fullname) {
    this.user_fullname = user_fullname;
}
}
```


News.java

```
package com.joreijarr.studycontrol.models;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class News {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public Long news_id;
    public String news_title, news_text, news_date;

    public News() {
    }

    public News(String news_title, String news_text, String news_date) {
        this.news_title = news_title;
        this.news_text = news_text;
        this.news_date = news_date;
    }

    public Long getNews_id() {
        return news_id;
    }

    public void setNews_id(Long news_id) {
        this.news_id = news_id;
    }
}
```

```
}

public String getNews_title() {
    return news_title;
}

public void setNews_title(String news_title) {
    this.news_title = news_title;
}

public String getNews_text() {
    return news_text;
}

public void setNews_text(String news_text) {
    this.news_text = news_text;
}

public String getNews_date() {
    return news_date;
}

public void setNews_date(String news_date) {
    this.news_date = news_date;
}
}
```

startController.java

```
package com.joreijarr.studycontrol.controllers;

import com.joreijarr.studycontrol.models.News;
import com.joreijarr.studycontrol.models.Links;
import com.joreijarr.studycontrol.models.Users;
import com.joreijarr.studycontrol.repo.LinksRepository;
import com.joreijarr.studycontrol.repo.NewsRepository;
import com.joreijarr.studycontrol.repo.UsersRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.util.ArrayList;
import java.util.List;

@Controller
public class startController {

    public Boolean authorized = false;
    public Users current_user;
    @Autowired
    public UsersRepository usersRepository;

    @Autowired
    public NewsRepository newsRepository;

    @Autowired
```

```
public LinksRepository linksRepository;

@GetMapping("/exit")
public String exit(Model model)
{
    current_user = null;
    authorized = false;
    return "redirect:/";
}

@GetMapping("/authorize")
public String start(Model model) {

    model.addAttribute("title", "Авторизація");
    if(!authorized)
    {
        return "authorize";
    }
    else
    {
        return "redirect:/";
    }
}

@PostMapping("/authorize")
public String authorize(Model model, @RequestParam String user_login, @RequestParam
String user_password)
{
    Iterable<Users> users_it = usersRepository.findAll();
    List<Users> users = new ArrayList<>();
    users_it.forEach(users::add);
    for (int i = 0; i < users.size(); i++)
    {
```

```

        if(users.get(i).getUser_login().equals(user_login)
users.get(i).getUser_password().equals(user_password))
        {
            authorized = true;
            current_user = users.get(i);
            break;
        }
    }
    return "redirect:";
}

```

```

@GetMapping("/registration")
public String registration(Model model)
{
    model.addAttribute("title", "Реєстрація");
    if(!authorized)
    {
        return "registration";
    }
    else
    {
        return "redirect:";
    }
}

```

```

@PostMapping("/registration")
public String registration(Model model, @RequestParam String user_login,
@RequestParam String user_password,
@RequestParam String user_fullname, @RequestParam String status)
{
    Users new_user = new Users(user_login, user_password, user_fullname, status);
    usersRepository.save(new_user);
    return "redirect:/authorize";
}

```

```
}

@GetMapping("/")
public String index(Model model)
{
    model.addAttribute("title", "Головна");
    if(authorized)
    {
        return "index_user";
    }
    else
    {
        return "redirect:/authorize";
    }
}

@GetMapping("/about")
public String about(Model model)
{
    model.addAttribute("title", "Про нас");
    return "about";
}

@GetMapping("/news")
public String newsPage(Model model)
{
    model.addAttribute("title", "Новини");
    Iterable<News> news_it = newsRepository.findAll();
    List<News> news = new ArrayList<>();
    news_it.forEach(news::add);
    model.addAttribute("currUser", current_user);
}
```



```
        @RequestParam String news_date)
    {
        News news = new News(news_title, news_text, news_date);
        newsRepository.save(news);
        return "redirect:/news";
    }

    @PostMapping("/links/add")
    public String linsAdd_post(Model model,
        @RequestParam String link_name,
        @RequestParam String link_description,
        @RequestParam String link_url)
    {
        Links links = new Links(link_name, link_description, link_url);
        linksRepository.save(links);
        return "redirect:/links";
    }
}
```


Лістинг каскадної таблиці стилів

```
{
  box-sizing: border-box;
}
body {
  background-color: #EEE;
}
.clear {
  clear: both;
}
.log {
  width: 400px;
  margin: 5% auto;
  background-color: #d0d8d9;
  padding: 30px 0;
}
.log h2 {
  text-align: center;
  color: #ed2553;
  font-weight: bold;
  font-size: 26px;
  margin-bottom: 50px;
}
.log .input-cont {
  position: relative;
  margin: 0 50px 60px;
}
.log .input-cont:last-of-type {
  margin-bottom: 30px;
}
.log .input-cont input {
  position: relative;
```

```
z-index: 1;
width: 100%;
height: 40px;
outline: none;
color: #212121;
font-size: 22px;
font-weight: 400;
background: none;
border: none;
}
.log .input-cont input:focus {
  outline: none;
}
.log .input-cont label {
  position: absolute;
  color: #948c8c;
  top: 0;
  left: 0;
  line-height: 40px;
  -webkit-transition: .3s;
  -moz-transition: .3s;
  -o-transition: .3s;
  transition: .3s;
}
.log .input-cont input:focus + label {
  margin-top: -30px;
  -webkit-transform: scale(.8);
  -moz-transform: scale(.8);
  -o-transform: scale(.8);
  transform: scale(.8);
  color: #bdbdbd;
}
.log .border1,
.log .border2 {
```

```

position: absolute;
height: 1px;
background-color: #9E9E9E;
left: 0;
bottom: 0;
width: 100%;
}
.log .border1::after,
.log .border1::before,
.log .border2::after,
.log .border2::before {
  content: "";
  position: absolute;
  bottom: 0;
  width: 0;
  height: 2px;
  -webkit-transition: .5s;
  -moz-transition: .5s;
  -o-transition: .5s;
  transition: .5s;
}
.log .border1::after,
.log .border2::after {
  right: 50%;
  background-color: #ed2553;
}
.log .border1::before,
.log .border2::before {
  left: 50%;
  background-color: #ed2553;
}
.log .input-cont input:focus ~ .border1::after,
.log .input-cont input:focus ~ .border1::before,

```

```
.log .input-cont input:focus ~ .border2::after,  
.log .input-cont input:focus ~ .border2::before {  
  width: 50%;  
}  
.log .check,  
.log a {  
  float: left;  
  width: calc(50% - 50px);  
  display: block;  
  font-size: 12px;  
  margin-bottom: 30px;  
}  
.log .check {  
  margin-left: 50px;  
}  
.log a {  
  text-align: right;  
  text-decoration: none;  
  color: #ed2553;  
}  
.log a:hover {  
  text-decoration: underline;  
  color: #F00;  
}  
.log form input[type="submit"] {  
  display: block;  
  margin: 0 auto 20px;  
  border: 2px solid transparent;  
  padding: 5px 20px;  
  font-size: 22px;  
  cursor: pointer;  
  color: #ed2553;  
  -webkit-transition: .5s;  
  -moz-transition: .5s;
```

```
-o-transition: .5s;  
  transition: .5s;  
}  
.log form input[type="submit"]:focus {  
  outline: none;  
}  
.log form input[type="submit"]:hover {  
  border: 2px solid #ed2553;  
}
```

ЛІСТИНГ сторінки

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="stylesheet" href="/css/start.css">
</head>
<body>
<div id="container">
  <div class="log">
    <h2>Реєстрація</h2>
    <form method="post">
      <div class="input-cont">
        <input type="text" name="user_login">
        <label>Логін</label>
        <div class="border1 "></div>
      </div>
      <div class="input-cont">
        <input type="password" name="user_password">
        <label>Пароль</label>
        <div class="border2"></div>
      </div>
      <div class="input-cont">
        <input type="text" name="user_fullname">
        <label>Повне ім'я</label>
        <div class="border1 "></div>
      </div>
      <div class="input-cont">
        <input type="text" name="status" text="user">
        <label>Статус</label>
        <div class="border1 "></div>
      </div>
      <a href="/authorize">Вже маєте аккаунт?</a>
      <input type="submit" value="Зареєструватися">
    </form>
```

```
</div>  
</div>  
</body>  
</html>
```

ДОДАТОК Д – Графічна частина

ГРАФІЧНА ЧАСТИНА
РОЗРОБКА ВЕБ-СИСТЕМИ ДЛЯ КОМПЛЕКСНОЇ ОЦІНКИ ТА
ПІДТРИМКИ РОЗВИТКУ ДИТИНИ

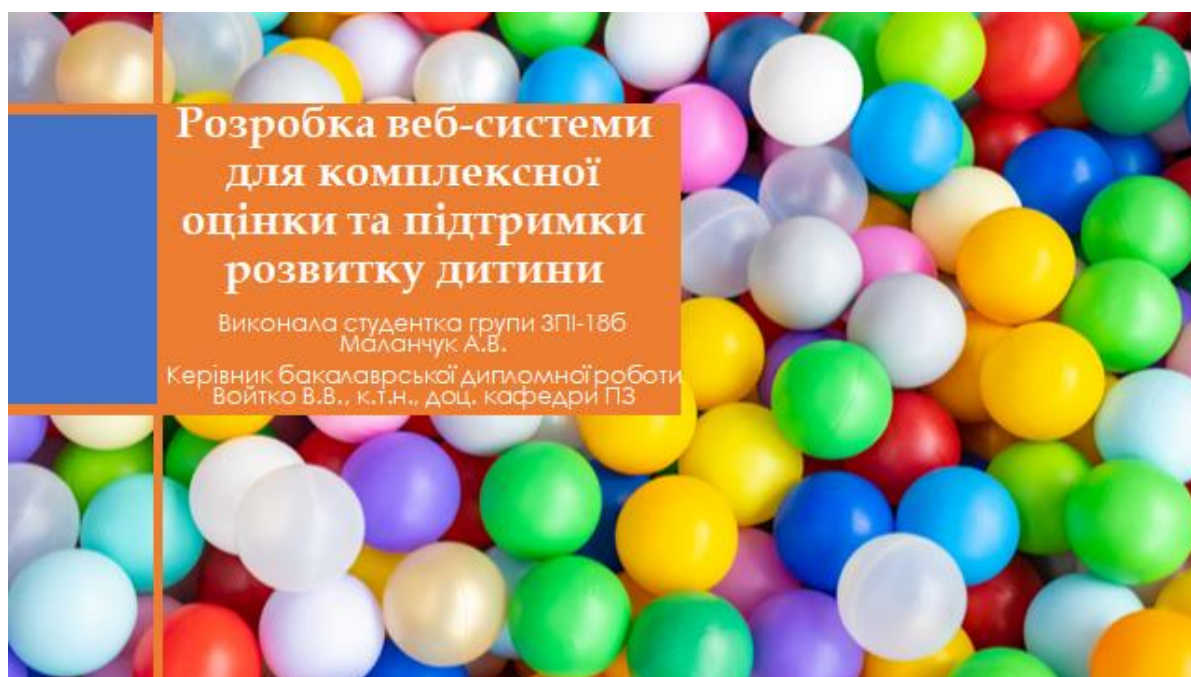


Рисунок Д.1 – Назва роботи

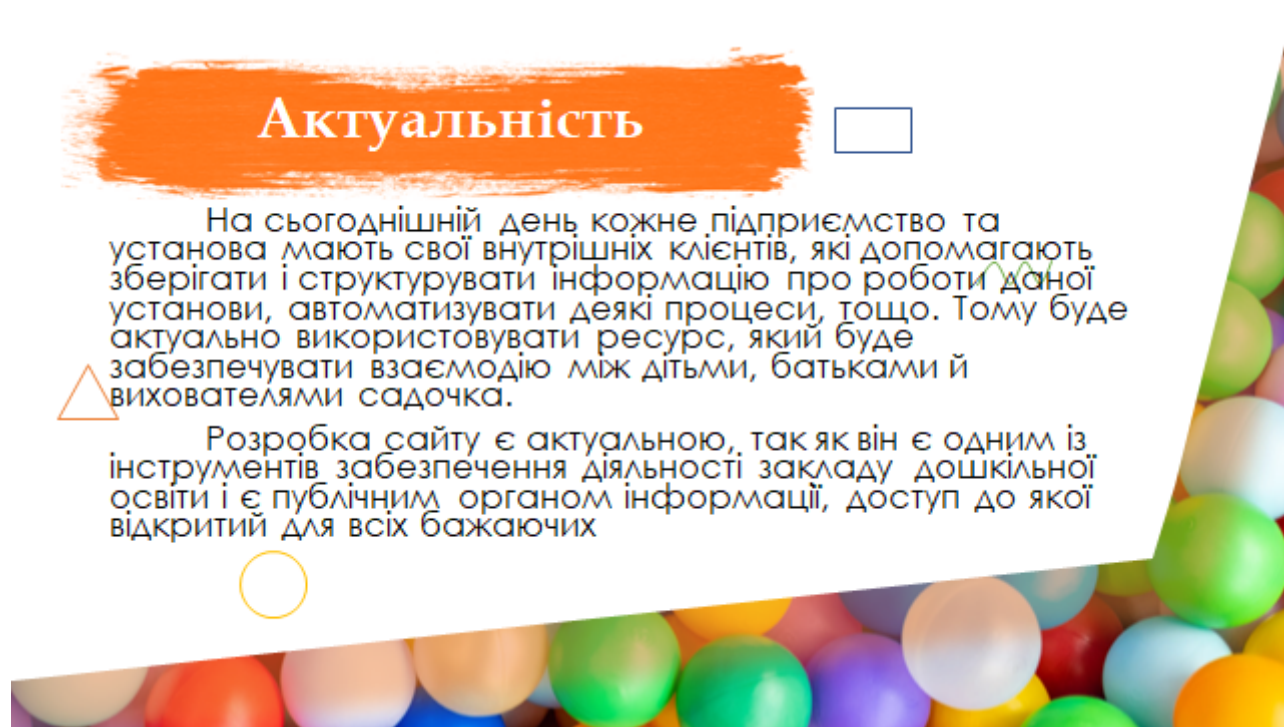


Рисунок Д.2 – Актуальність

Мета та завдання дослідження

- Метою бакалаврської дипломної роботи є підвищення ефективності роботи закладу дошкільної освіти шляхом розробки спеціалізованої веб-системи з елементами розвиваючих вправ і матеріалами для дітей, що забезпечить ефективну взаємодію суб'єктів навчально-виховного процесу дитячого садка.

Рисунок Д.3 – Мета та завдання

У процесі роботи використовувалися такі методи дослідження:

- методи аналізу та синтезу для побудови ключових моделей веб-системи;
- методи побудови компонентів системи та їхніх взаємозв'язків для створення загальної структури збереження та обробки інформаційних ресурсів;
- методи прикладної теорії інформації і теорії алгоритмів для розробки алгоритмів і програмного забезпечення.
- методи тестування для перевірки працездатності системи

Рисунок Д.4 – Методи дослідження

Наукова новизна отриманих результатів:



- Подальшого розвитку отримав метод комплексної оцінки та підтримки розвитку дитини з використанням спеціалізованої веб-системи, який, на відміну від існуючих, орієнтований на використання елементів розвиваючих вправ і матеріалів для дітей, що підвищить ефективність взаємодії суб'єктів навчально-виховного процесу дитячого садка.



- Подальшого розвитку отримала модель роботи веб-системи закладу дошкільної освіти, яка, на відміну від існуючих, орієнтована на покращення комплексної оцінки та підтримки розвитку дитини з використанням елементів розвиваючих вправ і матеріалів для дітей, що дозволить підвищити ефективність веб-ресурсу.



Рисунок Д.5 – Наукова новизна

Основними задачами роботи є:



- розробити метод комплексної оцінки та підтримки розвитку дитини у закладах дошкільної освіти з використанням спеціалізованої веб-системи;
- розробити модель роботи веб-системи;
- проаналізувати інформаційне забезпечення сайту;
- розробити програмні засоби реалізації веб-системи комплексної оцінки та підтримки розвитку дитини у закладах дошкільної освіти;
- розробити зручний та зрозумілий графічний інтерфейс веб-ресурсу;
- провести тестування веб-системи відповідно до поставлених задач



Рисунок Д.6 – Задачі роботи

Порівняльний аналіз аналогів

Функції	Демківський ЗДО	Дитячий садочок "Пазлик"	Дитячий садок "Пролісок"
Новини	+	+	+
Дітям	+	+	-
Реєстрація	+	-	-
Зручна навігація	+	+	-
Швидкість	+	-	-
Контакти	+	+	-

Рисунок Д.7 – Порівняльний аналіз аналогів

АКТ

про впровадження результатів бакалаврської дипломної роботи

«Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини» студентки ВНТУ Маланчук Анни Василівни

Результати бакалаврської роботи «Розробка веб-системи для комплексної оцінки та підтримки розвитку дитини» студентки ВНТУ Маланчук А.В., що полягають в розробці веб-системи як єдиного інформаційного освітнього простору навчально-виховного закладу;



Рисунок Д.8 – Акт впровадження

Діаграма діяльності сайту

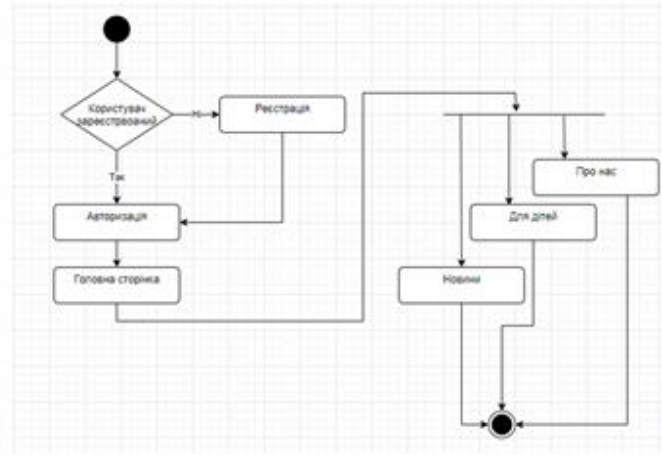


Рисунок Д.9 – Діаграма діяльності сайту

Діаграма варіантів використання

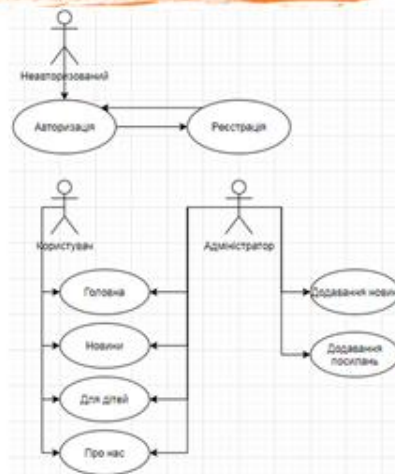


Рисунок Д.10 – Діаграма варіантів використання

Головна сторінка

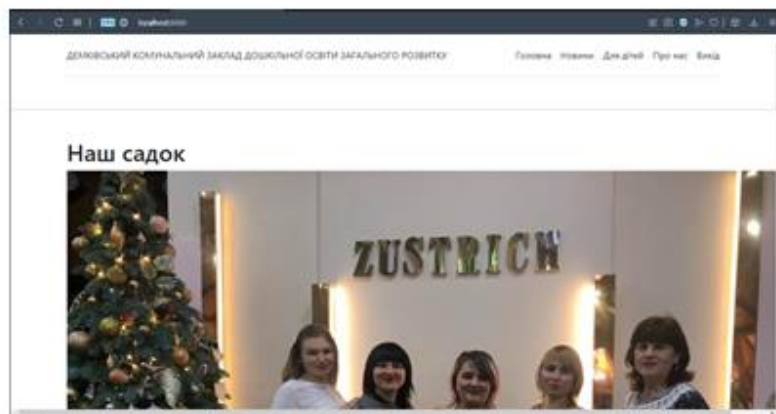


Рисунок Д.11 – Головна сторінка

Новини

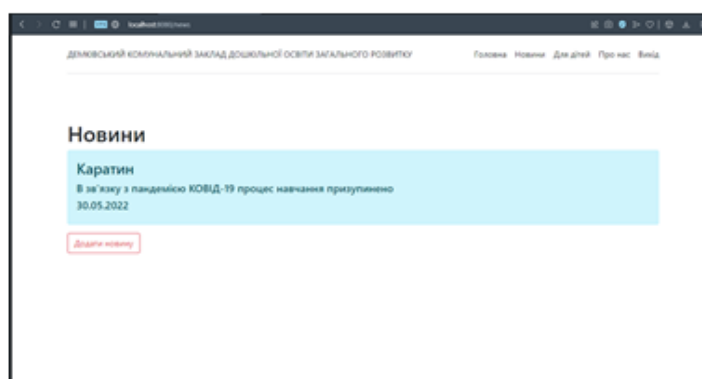


Рисунок Д.12 – Новини

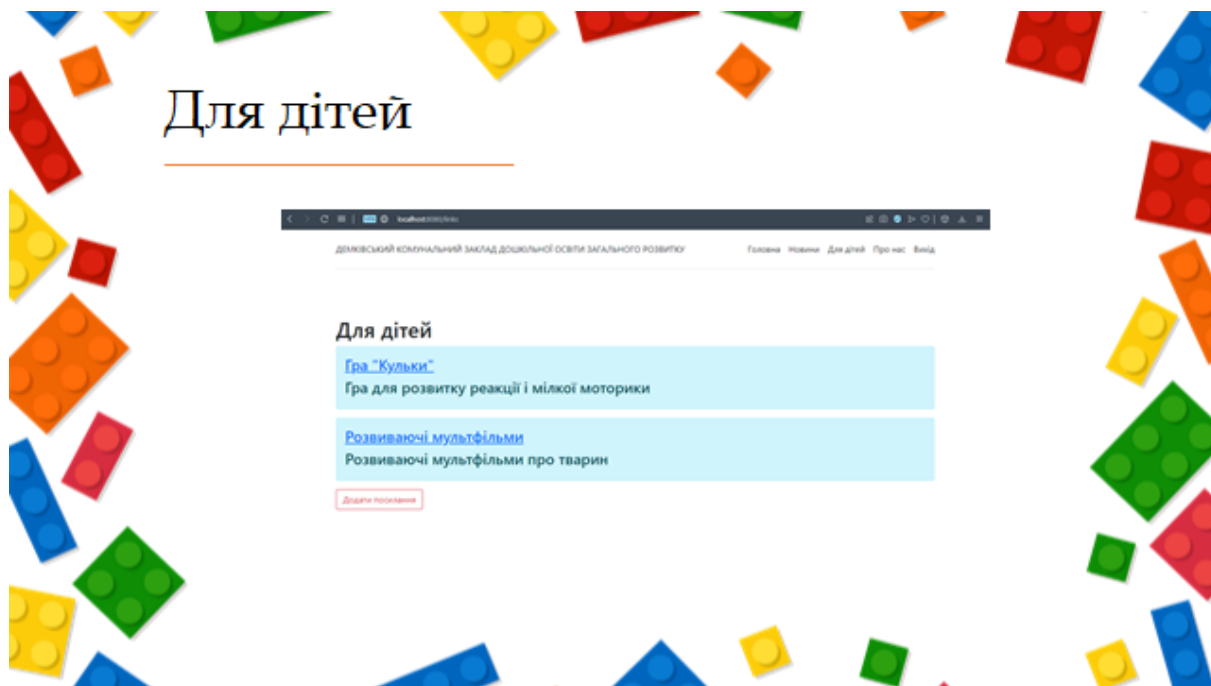


Рисунок Д.13 – Для дітей

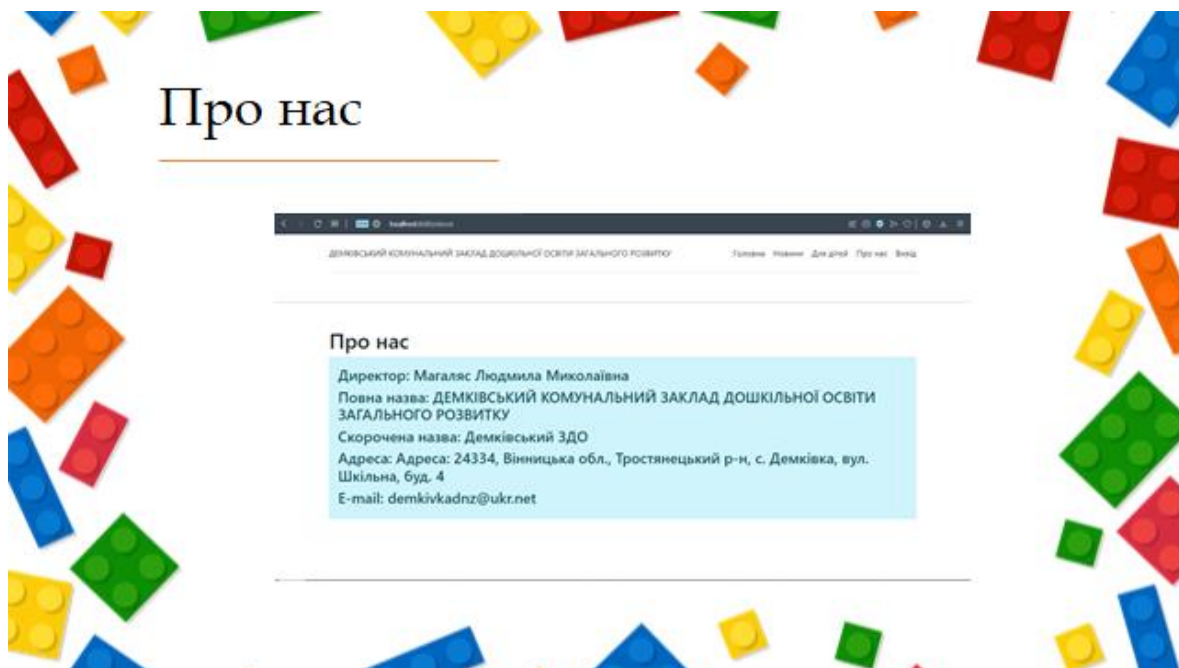


Рисунок Д.14 – Про нас

Апробація та публікації

- Основні положення бакалаврської дипломної роботи доповідалися та обговорювалися на ІІ науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2022) ВНТУ.
- Основні результати дослідження опубліковані в науковій роботі [3] – в тезах доповіді на ІІ науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2022) ВНТУ.



Рисунок Д.15 – Апробація та публікації

Висновки

- Розроблено веб-систему дитячого садочку з елементами розвиваючих вправ і матеріалів.
- Здійснено порівняльний аналіз існуючих реалізацій з урахуванням усіх недоліків і переваг.
- Розроблено схеми загального алгоритму роботи веб-системи.
- Розроблено модель роботи веб-системи.
- Розроблено мережну структуру веб-сайту. Основні розділи веб-сайту: Головна, Новини, Для дітей, Про нас, Вихід.

Рисунок Д.16 – Висновки