

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДІВ

ПЛАНУВАННЯ НАВЧАЛЬНОГО ПРОЦЕСУ

RESEARCH AND IMPLEMENTATION OF METHODS

OF PLANNING THE EDUCATIONAL PROCESS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-123м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ГРЕВЦОВА О.А.

(ПРИЗВИЩЕ та ініціали)

Керівник ЗАЙКО Т.А.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГРЕВІЦОВОЇ Олександрі Антонівни
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та реалізації методів планування навчального процесу.
Research and Implementation of Methods of Planning Educational Process

керівник проєкту (роботи) к.т.н., доцент, ЗАЙКО Тетяна Анатоліївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 07 ” жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.
2. Матеріали і методи. 3. Опис програми. 4. Керівництво програміста. 5. Експлуатація,
тестування та експериментальне дослідження програми.

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ЗАЙКО Т.А., доцент		
Нормоконтролер	ДЕЙНЕГА Л.Ю., ст.викладач		

7. Дата видачі завдання “ 09 ” вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2-3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4-5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї.	10-11 тижні	Додатки
8	Нормоконтроль та рецензування.	12 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка) _____ Олександра ГРЕВЦОВА _____
(підпис) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи) _____ Тетяна ЗАЙКО _____
(підпис) (Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
96 с., 8 табл., 41 рис., 3 дод., 16 джерел.

НАВЧАЛЬНИЙ ПРОЦЕС, МЕТОД ПЛАНУВАННЯ НАВЧАЛЬНОГО ПРОЦЕСУ, СИСТЕМА УПРАВЛІННЯ НАВЧАННЯМ, ВЕБСАЙТ, РОЗРОБКА, ДІАГРАМА, АВТОРИЗАЦІЯ, ВЗАЄМОДІЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для управління навчальним процесом.

Предмет дослідження – методи та програмні засоби для планування навчального процесу.

Мета роботи – дослідження методів та розробка програмного забезпечення для планування навчального процесу.

Матеріали, методи та технічні засоби: метод планування навчального процесу, фреймворк Vue.js, мова програмування Node.js, мова розмітки HTML, таблиця каскадних стилів CSS, база даних MongoDB, середовище розробки Visual Studio Code.

Результати. Розроблено програмне забезпечення для планування навчального процесу за допомогою мови програмування Vue.js та середовища Microsoft Visual Studio.

Практична цінність роботи полягає у розробці програмного забезпечення, яке визначає необхідний алгоритм дій щодо досягнення позитивного результату у користувача у сфері досягнення освіти.

Виконано проектування програмного забезпечення для спрощення моніторингу навчального процесу. Розроблено програмне забезпечення для користувачів, та представників сфери освіти. Здійснено тестування розробленого програмного забезпечення для планування навчального процесу.

Галузь використання – загальноосвітні школи, вищі навчальні заклад, корпоративне навчання, онлайн-освіта.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 96 pages, 8 tables, 41 figures, 3 appendixes, 16 sources.

EDUCATIONAL PROCESS, EDUCATIONAL PROCESS PLANNING METHOD, LEARNING MANAGEMENT SYSTEM, WEBSITE, DEVELOPMENT, DIAGRAM, AUTHORIZATION, INTERACTION.

The object of research is the process of developing software for managing the educational process.

The subject of research is methods and software tools for planning the educational process.

The purpose of the work is to study methods and develop software for planning the educational process.

Materials, methods and technical means:; initial process planning method, Vue.js framework, Node.js programming language, HTML markup language, CSS cascading style sheet, MongoDB database, Visual Studio Code development environment.

Results: Software for planning the educational process was developed using the Vue.js programming language and the Microsoft Visual Studio environment.

The practical value of the work lies in the development of software that determines the necessary algorithm of actions to achieve a positive result for the user in the field of educational achievement.

Software design has been completed to simplify monitoring of the educational process. Software has been developed for users and representatives of the education sector. The developed software has been tested for planning the educational process.

The field of use is secondary schools, higher education institutions, corporate training, online education.

ЗМІСТ

	С.
Перелік скорочень та умовних познач	9
Вступ	10
1 Аналіз проблеми та постановка завдань дослідження	12
1.1 Основні поняття і актуальність планування навчального процесу	12
1.2 Програмні аналоги додатків планування навчального процесу	13
1.3 Висновки за розділом 1	18
2 Матеріали та методи	20
2.1 Вибір мови програмування та інших технологій розробки	20
2.2 Вибір середовища розробки для створення програмного забезпечення системи управління планування навчанням	23
2.3 Метод планування навчальним процесом	25
2.4 Висновки за розділом 2	27
3 Опис програми	29
3.1 Структура додатку системи планування навчального процесу	29
3.2 Проектування бази даних додатку системи планування навчанням	30
3.3 Функціонування системи управління навчальним процесом	35
3.4 Авторизація	35
3.5 Метод планування системи планування навчального процесу	38
3.6 Додавання предметів	43
3.7 Висновки за розділом 3	49
4 Керівництво програміста	51
4.1 Призначення й умови застосування програми	51
4.2 Характеристики розробленого продукту	29
4.3 Структурна схема компонентів вебсервісу	53
4.4 Висновки за розділом 4	53
5 Експлуатація, тестування та експериментальне дослідження програми	54
5.1 Опис роботи з програмою	29
5.1.1 Звернення до програми	54

5.1.2 Вхідні й вихідні дані.....	54
5.1.3 Повідомлення.....	54
5.2 Інструкція з використання	55
5.3 Експериментальне дослідження (тестування)	62
5.4 Висновки за розділом 5	63
Висновки.....	51
Перелік джерел посилання	66
Додаток А Техічне завдання.....	68
Додаток Б Текст програми.....	72
Додаток В Слайди презентації	89

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

CSS	– Cascading Style Sheets;
DB	– Data Base;
HTML	– HyperText Markup Language;
HTTPS	– HyperText Transfer Protocol Secure;
JS	– JavaScript;
OS	–Operating System;
VS	– Visual Studio;
БД	– база даних;
СКБД	– система керування базами даних.

ВСТУП

Планування навчального процесу є дуже важливим аспектом якісної освіти та надає ефективне, послідовне та гнучке навчання. Має велике значення, тому що забезпечує організацію освітньої діяльності та сприяє досягненню конкретних освітніх цілей.

Планування навчального процесу має систематичність та послідовність, тому що допомагає поступово засвоювати знання та розвиток навичок. Орієнтоване на досягнення конкретних поставлених цілей, що дозволяє визначити ефективні методи навчання і оцінки успіху

Раціональне використання власного часу дуже важливий аспект. Оскільки викладачі можуть більш краще організовувати темп викладання, розподілення матеріалу на конкретні етапи та уникати перенавантаження студентів та втрати часу на неефективні завдання.

Навчальні плани надають чітку структуру та очікування для студентів та учнів, тому це допомагає підтримувати мотивацію та дисципліну серед користувачів. Також планування залишає простір для гнучкості та змін у залежності від потреб користувачів або непередбачених обставин.

Різноманітні програмні забезпечення допомагають більш простіше планувати свій час в досягненні освіти. За допомогою різноманітних програм можна автоматично створювати розклади занять, враховуючи завантежність вчителів, доступність студентів та використаних ресурсів.

Найновітніші додатки надають можливість перегляду змін в навчальних планах та розкладах. Дозволяють автоматично відстежувати успішність студентів, зберігаючи їх оцінки та формуючи аналітику на основі навчальних досягнень та інтегрувати різні освітні ресурси (відео, презентації, посилання, тести), що також робить навчання більш різноманітним та цікавим.

Таким чином, планування навчального процесу є фундаментом гарної, якісної освіти. А програмні додатки допомагають викладачам та учням

ефективно планувати, організовувати і контролювати навчальний процес.

Тому актуальною є розробка програмного забезпечення для планування навчального процесу.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів;
- розробити модель взаємодії користувача з програмним забезпеченням;
- здійснити проектування програмного забезпечення ;
- створити програмне забезпечення;
- виконати тестування розробленого програмного забезпечення ;

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Основні поняття і актуальність планування навчального процесу

Планування навчального процесу є однією з ключових складових управління освітньою діяльністю, яка забезпечують ефективну організацію безпосередньо навчання, досягнення освітніх цілей та підвищення якості освіти. Це системний процес, що включає підготовку, контроль та структурування усіх етапів навчання.

Навчальний процес – це цілісна система організації навчальної діяльності, яка забезпечує засвоєння знань, формування вмінь та навичок, розвиток компетенцій, необхідних для успішного виконання професійних або інших життєвих завдань [1].

Планування навчального процесу – це процес визначення змісту, структурита послідовності навчальних дій і заходів для досягнення конкретних освітніх результатів. Планування охоплює розробку навчальних планів та програм, складання розвитку занять, вибір форм та методів навчання, оцінку якості знань учнів та інше [2].

Актуальність планування навчального процесу дуже зростає в сучасних умовах та реаліях. Суспільство вимагає високоякісної освіти, а ринок праці постійно змінюється, вимагаючи нових знань та компетенцій. Планування дозволяє забезпечити системність та послідовність у навчанні, а це є необхідною умовою для досягнення якісних результатів. Дає змогу вибудувати освітній процес таким чином, щоб учні отримували не тільки теоретичні знання, а й практичні навички, які знадобляться їм у майбутньому.

Ефективне планування навчального процесу допомагає уникнути хаотичності в організації занять і забезпечує оптимальне використання навчального часу. Воно дозволяє враховувати індивідуальні особливості студентів та підлаштовувати навчання під їхні потреби, що сприяє більш

ефективному засвоєнню матеріалу. Важливим аспектом є також можливість адаптації освітніх програм до новітніх технологій, які активно впроваджуються в сучасну систему освіти.

Актуальність планування навчального процесу зумовлена необхідністю забезпечення належного рівня контролю знань. Воно дозволяє врахувати всі етапи оцінювання, що дозволяє ефективно перевірити контроль знань та вчасно виявити прогалини в заняттях.

Загалом, без планування навчального процесу неможливо організувати якісний освітній процес, що відповідає вимогам сучасного світу, де гнучкість і здатність до швидкого навчання стають ключовими як для особистого так і професійного розвитку.

1.2 Програмні аналоги додатків планування навчального процесу

В сучасному світі дуже багато різних додатків, які допомагають структурувати освіту, як для студентів так і для вчителів.

Наприклад системи управління навчанням Moodle, Google Classroom та Canvas Student.

Всі платформи призначені для об'єднання педагогів та учнів в одну надійну, безпечну та інтегровану систему для створення персоналізованого навчального середовища.

Moodle – це безкоштовна, відкрита система управління навчанням. Вона реалізує філософію «педагогіки соціального конструктивізму» та орієнтована насамперед на організацію взаємодії між викладачем та учнями, хоча підходить і для організації традиційних дистанційних курсів, а також підтримки очного навчання (рис. 1.1) [3].



Рисунок 1.1 – Лого системи управління навчанням Moodle

На рисунку 1.2 продемонстровано інтерфейс системи Moodle



Рисунок 1.2 – Інтерфейс системи управління навчання Moodle

Google Classroom – це вебсервіс, створений Google для навчальних закладів з метою спрощення створення, поширення і класифікації завдань безпаперовим шляхом. Основна мета сервісу — прискорити процес поширення файлів між педагогами та здобувачами освіти. Може використовуватися вчителями та учнями у школах, або у закладах вищої освіти викладачами та студентами (рис. 1.3) [4].



Рисунок 1.3 – Лого системи управління навчанням Google Classroom

На рисунку 1.4 продемонстровано інтерфейс системи Google Classroom

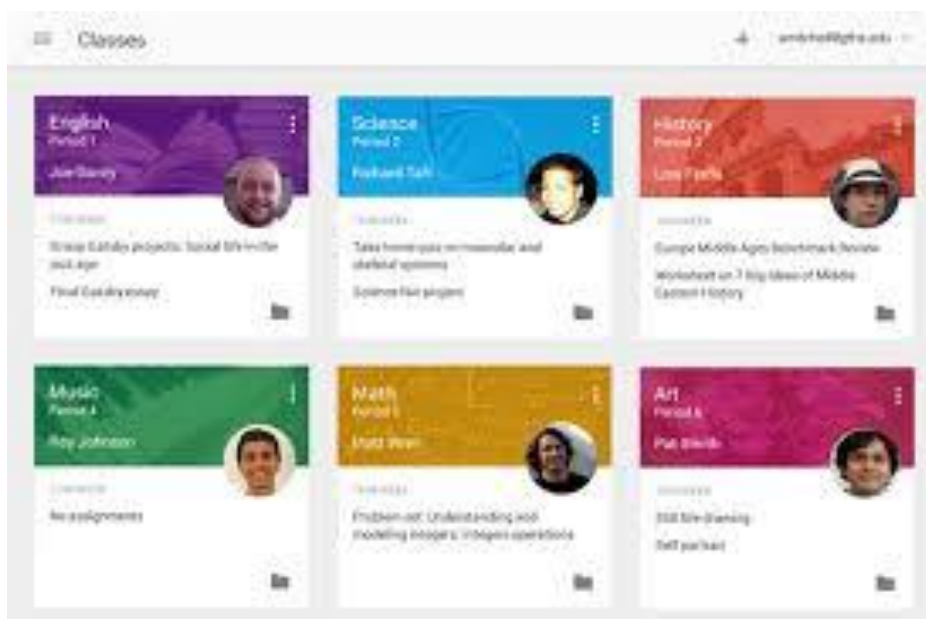


Рисунок 1.4 – Інтерфейс системи управління навчання Google Classroom

Canvas Student – це додаток, який дозволяє студентам отримувати доступ до своїх курсів і груп за допомогою мобільного пристрою. Студенти можуть надсилати завдання, брати участь в обговореннях, переглядати оцінки та отримувати доступ до матеріалів курсу. Програма також надає

доступ до календарних курсів, завдань, сповіщень та повідомлень (рис. 1.5) [5].



Рисунок 1.5 – Лого системи управління навчанням Canvas Student

На рисунку 1.6 продемонстровано інтерфейс системи Canvas Student.

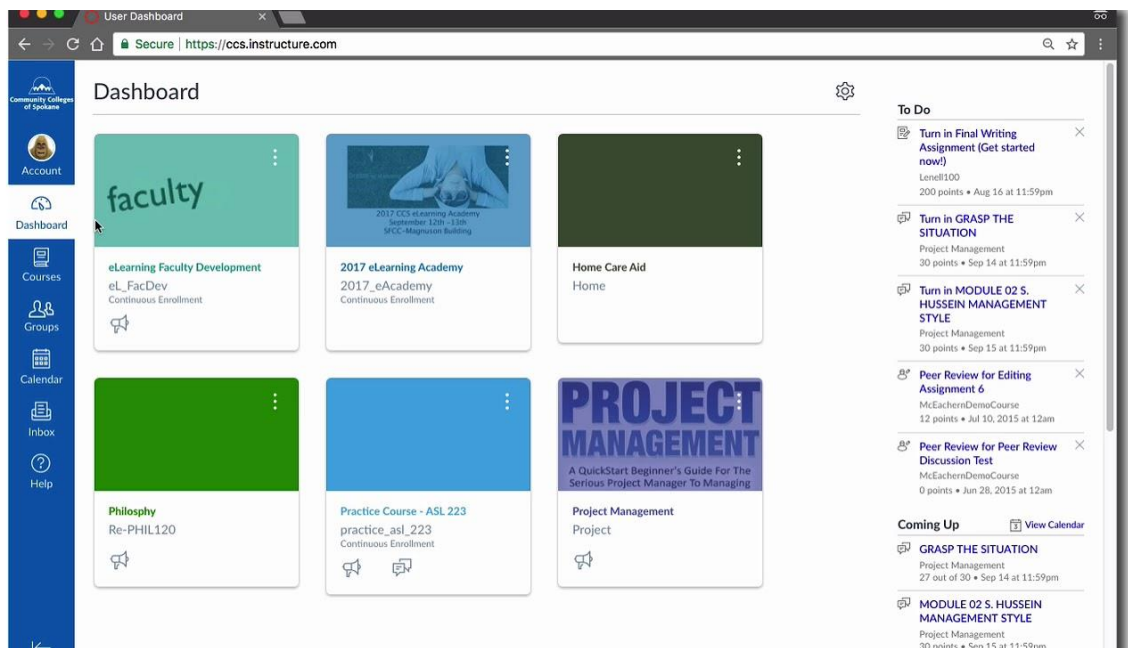


Рисунок 1.6 – Інтерфейс системи управління навчання Canvas Student

Порівняння систем управління навчанням представлено у таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця систем управління навчанням

Характеристика	Moodle	Google Classroom	Canvas Student
Тип платформи	Відкрите ПЗ з відкритим кодом	Хмарна платформа від Google	Хмарна платформа з опціями локальних інсталяцій
Гнучкість	Висока, можливість налаштування через плагіни	Обмежена кількість налаштувань	Висока гнучкість, багато інтеграцій
Підтримка інтеграцій	Безліч плагінів, та сторонніх інтеграцій	Інтеграція з Google-сервісами	Підтримка LTI, інтеграції з безліччю сервісами
Кастомізація	Дуже висока, можна створювати власні теми та модулі	Мінімальна кастомізація	Висока кастомізація через модулі та інтеграції

Як висновок, можна зробити, що Moodle підходить для великих закладів. Він ідеальний для університетів з великом обсягом курсів. Google Classroom простий у використанні, зручний для шкіл і менших освітніх установ. Canvas Student поєднує гнучкість та сучасний інтерфейс, з акцентом на університети та корпоративне навчання.

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для управління навчальним процесом. Проте деякі сервіси у вигляді вебсайтів для роботи з навчанням можуть бути схильні до технічних неполадок, такі як відключення серверів

або збої в інтерфейсі, можуть негативно вплинути на процес торгівлі. Це може призвести до втрати даних або проблем з доступ до важливої інформації в реальному часі.

Тому актуальною є розробка програмного забезпечення для планування навчальним процесом, яка буде захищена від шахрайських цілей.

При розробці системи управління навчальним процесом необхідно забезпечити такі функціональні вимоги:

- управління розкладом;
- управління курсами та програмами;
- реєстрація студентів у додатку;
- сповіщення та нагадування;
- доступ через веб-інтерфейс та мобільні додатки.

1.3 Висновки за розділом 1

За результатами проведеного аналізу можна зробити висновок, що сьогодні існує досить багато програмних засобів для підтримки управління навчальним процесом. Проте деякі з цих програм, реалізованих у вигляді вебсайтів, можуть бути вразливими до технічних неполадок, таких як збої серверів або проблеми з інтерфейсом, що може негативно вплинути на ефективність навчання. На інших платформах для управління навчальним процесом виникають проблеми з достовірністю інформації, внаслідок чого студенти можуть отримати некоректні навчальні матеріали або завдання, які не відповідають заявленому змісту, оскільки вони не завжди мають можливість перевірити їх перед початком курсу. Крім того, попри заходи безпеки, що реалізовані в програмних засобах для управління навчальним процесом, існує ризик витоку особистих даних студентів та викладачів, що може призвести до їх несанкціонованого використання.

Тому важливою є розробка програмного забезпечення для підтримки діяльності систем управління навчальним процесом, яке б забезпечило

надійність і безпеку користувачів.

Сформульовано функціональні вимоги до системи управління навчальним процесом.

2 МАТЕРІАЛИ ТА МЕТОДИ

2.1 Вибір мови програмування та інших технологій розробки

Вибір мови програмування та інших технологій є важливим етапом у процесі розробки програмного продукту. У цьому проекті для створення було використано мови програмування Vue.js, JavaScript та Node.js.

Однією з основних мов програмування для серверної частини проекту було обрано Node.js. Це рішення прийнято через те, що Node.js є популярним інструментом для веброзробки, який дозволяє ефективно працювати з великою кількістю запитів завдяки неблокуючій моделі вводу/виводу. Node.js побудований на JavaScript, що дозволяє використовувати одну мову для написання як серверної, так і клієнтської частин, забезпечуючи зручність розробки. Він також має велику кількість модулів і бібліотек, які спрощують створення вебдодатків і інтеграцію з базами даних або іншими сервісами.

Vue.js – це JavaScript-фреймворк, який був використаний для розробки клієнтської частини вебзастосунку. Ми обрали Vue.js для нашого проекту через його простоту та гнучкість, що дозволяє створювати інтерактивний і динамічний інтерфейс користувача. За допомогою Vue.js ми змогли легко управляти станом компонентів і забезпечити плавну взаємодію з користувачем без необхідності перезавантаження сторінки. Крім того, Vue.js спрощує реалізацію таких функцій, як перевірка даних у формах, інтерактивні елементи та асинхронні запити до сервера [6].

Для розробки були застосовані технології, зокрема HTML5 та CSS3. HTML5 є новою версією мови розмітки HTML, яка впровадила численні інновації, включаючи підтримку аудіо та відео, можливість малювання за допомогою canvas, а також доступ до апаратних можливостей пристроїв та багато інших функцій [7].

CSS3 — це оновлена версія каскадних таблиць стилів, яка запровадила безліч нових можливостей, таких як анімація, плавні переходи,

адаптивні сітки (Flexbox і Grid), а також можливість роботи з тінями та фоновими зображеннями [8].

Одним і аналогів Vue.js є фреймворк React.js. Обидва фреймворки дозволяють створювати інтерактивні веб-додатки та мають свої особливості. Vue.js відзначається простотою використання та гнучкістю, тоді як React забезпечує потужну екосистему і широкий вибір бібліотек для управління станом.

Обидва варіанти можуть бути ефективними для різних проектів в залежності від вимог і цілей розробки.

Vue.js має такі переваги:

- завдяки зрозумілій документації та простому синтаксису, Vue.js легко освоюється навіть новачками;
- Vue дозволяє інтегруватися в існуючі проекти або використовувати його для створення нових з нуля. Це робить його дуже універсальним;
- система реактивності Vue забезпечує автоматичне оновлення даних в інтерфейсі при зміні стану, що робить роботу з даними зручною та ефективно;
- Vue.js дозволяє створювати повторно використовувані компоненти, що сприяє кращій організації коду та спрощує підтримку проектів;
- Vue.js оптимізує оновлення DOM, що забезпечує швидке завантаження і роботу додатків.

Vue має багатий набір бібліотек і інструментів, таких як Vue Router для маршрутизації та Vuex для управління станом, що розширює його можливості [9].

Ці особливості роблять Vue.js популярним вибором для розробки сучасних веб-додатків.

Переваги використання Node.js code.js має безліч переваг, які роблять його популярним вибором для розробки веб-додатків:

- завдяки неблокуючій архітектурі та використанню JavaScript на стороні сервера, Node.js забезпечує швидку обробку запитів і високу продуктивність;
- використання JavaScript як на клієнтській, так і на серверній стороні спрощує розробку та дозволяє командам працювати в одному середовищі;
- Node.js має величезну бібліотеку пакетів через npm (Node Package Manager), що дозволяє легко знаходити і використовувати готові рішення для різних задач;
- Node.js легко масштабувати горизонтально (додавання нових серверів) і вертикально (додавання ресурсів на існуючі сервери);
- Node.js ідеально підходить для створення додатків, які вимагають взаємодії в реальному часі, таких як чати, ігри чи колабораційні платформи;
- Node.js чудово підходить для створення API, що дозволяє легко інтегруватися з різними сервісами та додатками;
- Node.js підходить для створення мікросервісів, що дозволяє розподіляти навантаження та покращувати управління проектами [10].

Ці переваги роблять Node.js відмінним вибором для створення швидких, масштабованих та ефективних веб-додатків.

Обґрунтування вибору мови програмування для розробки програмного забезпечення системи планування навчального процесу наведено в таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування

Параметри порівняння мов програмування	Мови програмування	
	Vue.js	React.js
Простота навчання	+	-
Архітектура	+	+
Гнучкість	+	+
Стан управління	+	+
Шаблони	+	-

Продовження таблиці 2.1

Параметри порівняння мов програмування	Мови програмування	
	Vue.js	React.js
Продуктивність	+	+
Екосистема	-	+
Середовище	+	+
Підтримка мобільних додатків	+	+

Отже для створення програмного додатку системи управління навчального процесу обрано мови Vue.js та Node.js. Для клієнтської частини був обран фреймворк Vue.js, для серверної – Node.js. Кожна мова програмування має свої власні сильні сторони та використовуються для різноманітних цілей.

Комбінація цих двох технологій забезпечила створення повнофункціонального та інтерактивного вебсайту, що відповідає вимогам користувачів. При виборі мови програмування або технології для веброзробки важливо враховувати специфіку проєкту і досвід розробника. У багатьох випадках Vue.js та Node.js є чудовим вибором для вирішення різноманітних завдань у процесі розробки.

2.2 Вибір середовища розробки для створення програмного забезпечення системи управління планування навчанням

Visual Studio Code — це легкий, але потужний текстовий редактор, розроблений компанією Microsoft. Він призначений для роботи з різними мовами програмування і підтримує багато функцій, які роблять його популярним інструментом серед розробників.

Однією з ключових переваг Visual Studio Code є його кросплатформеність. Він працює на всіх основних операційних системах,

включаючи Windows, macOS і Linux, що робить його доступним для більшості користувачів незалежно від їхнього обладнання. Це дозволяє розробникам використовувати один інструмент незалежно від середовища, в якому вони працюють [11].

Редактор підтримує роботу з безліччю мов програмування. Це робить його універсальним інструментом як для початківців, так і для досвідчених програмістів. Visual Studio Code підтримує різні популярні мови, але завдяки системі розширень список мов, з якими він може працювати, можна розширювати майже до безкінечності [12].

Інтелектуальні можливості редактора включають функцію IntelliSense, яка надає розробникам автодоповнення коду та перевірку синтаксичних помилок. Це суттєво прискорює процес написання коду та допомагає уникнути типових помилок. До того ж, система підсвічування синтаксису робить код більш читабельним, що є важливим для розробників, які працюють з великими проектами [13].

Ще однією важливою функцією є інтеграція з Git — найпопулярнішою системою контролю версій. Прямо в редакторі можна переглядати зміни в коді, створювати коміти, перемикатися між гілками та вирішувати конфлікти. Це значно спрощує роботу з командними проектами та допомагає контролювати процес розробки [14].

Visual Studio Code є універсальним інструментом, який підійде для різних типів розробки: від невеликих особистих проектів до масштабних корпоративних систем. Початківці зможуть швидко освоїти цей редактор завдяки його простому інтерфейсу, а професіонали оцінять його розширені можливості та інструменти для оптимізації робочого процесу.

Обґрунтування середовища розробки для створення програмного забезпечення системи управління навчального процесу наведено в таблиці 2.2.

Таблиця 2.2 - Обґрунтування середовища розробки для створення програмного забезпечення системи управління навчального процесу

Характеристика	VS Code	Sublime Text
Кросплатформеність	Windows, macOS, Linux	Windows, macOS, Linux
Швидкодія	Дуже висока	Висока
Підтримка плагінів/розширень	Багато, через Marketplace	Обмежена кількість плагінів
Інтелектуальне автодоповнення	IntelliSense (сильне)	Базове
Вбудований термінал	Так	Ні
Підтримка налагодження	Так, через плагіни	Ні

Таким чином, VS Code поєднує в собі легкість, гнучкість і потужність, перевершуючи інші текстові редактори та середовища розробки за багатьма критеріями. Він забезпечує винятковий баланс між функціональністю та швидкістю, пропонуючи розробникам інструменти, необхідні для продуктивної роботи, незалежно від мови програмування чи типу проекту.

2.3 Метод планування навчальним процесом

Наукова робота, пов'язана з плануванням навчального процесу, базується на аналізі даних про студентів, їх успішності та виявлення ключових показників для оптимізації процесу навчання. Основним завданням є створення методичної системи, яка дозволяла б викладачам і студентам приймати більш поінформовані рішення на основі об'єктивних даних. Такий підхід покращує навчальний процес, допомагає ідентифікувати слабкі сторони та підбирати відповідні коригувальні дії.

Для досягнення цих цілей необхідно впровадити систему, яка агрегуватиме дані щодо успішності студентів, класифікувати предмети за категоріями та на основі певних метрик (наприклад, середній бал по групі предметів) рекомендувати, які з них потребують додаткової уваги.

Розроблений метод планування навчального процесу ґрунтується на кількох ключових компонентах:

- категоризація предметів.

Кожен навчальний предмет належить до певної категорії. Це потрібно для того, щоб групувати дисципліни за певними напрямками. Наприклад: «Основи програмування», «Алгоритми та структури даних», «Веб-дизайн» належать до категорії «Комп'ютерні науки». «Математика», «Теорія ймовірностей» відносяться до категорії «Математичні науки».

Це необхідно для агрегування даних за категоріями та подальшого аналізу успішності студентів не лише з окремих предметів, а й за цілими напрямками знань.

- Збір та обробка даних.

Для кожного студента збираються дані щодо оцінок за різні предмети. Ці дані заносяться до системи та обробляються для подальшого аналізу.

- Аналіз успішності.

На основі даних про успішність по кожному предмету всередині категорії обчислюється середня оцінка з усіх предметів, що належать до цієї категорії. Алгоритм роботи системи виглядає так: збираються оцінки з усіх предметів у категорії, обчислюється середнє арифметичне оцінювання для кожного студента за кожною категорією предметів. Якщо середня оцінка категорії нижче порога, наприклад, 3,5, це вважається індикатором того, що студент потребує додаткової уваги з цієї групи предметів. Цей кордон може бути гнучким залежно від вимог освітнього закладу.

Результати аналізу візуалізовані та представлені користувачеві у зручному форматі. Даний метод зображує результат у вигляді графіків, який

наочно показує поточний рівень успішності та області, які вимагають поліпшення.

Для того щоб перевірити дієвість цього методу, можна використати емпіричний аналіз результатів студентів.

До впровадження даної системи необхідно зібрати дані про успішність студентів до застосування нового методу. Це можуть бути їхні оцінки, середній бал по категоріях предметів, а також відсоток студентів, які потребують додаткової допомоги.

Після впровадження системи (наприклад, один семестр або рік) знову зібрати дані про успішність студентів. Порівняти середній бал до і після впровадження системи, оцінити зміни у відсотку студентів, які поліпшили свої результати.

Таким чином, запропонований метод планування навчального процесу з використанням метрик на основі успішності студентів дозволяє ефективніше відстежувати прогрес у навчанні. За рахунок категоризації предметів та обчислення середнього балу за кожною категорією викладачі та студенти можуть детальніше аналізувати навчальний процес та вживати заходів для його покращення. Впровадження такої системи допоможе в індивідуалізації навчальних планів та оптимізації роботи з студентами, що відстають.

2.4. Висновки за розділом 2

Для розробки програмного забезпечення додатку системи планування навчальним процесом було обрано мови програмування Vue.js і Node.js. Ці мови є ключовими інструментами у веброзробці, дозволяючи створювати вебдодатки та сайти, які задовольняють потреби користувачів і забезпечують функціональність, необхідну для успішної роботи сервісу.

Для розробки додатку було обрано середовище Visual Studio Code. Воно забезпечує розширену підтримку таких ключових технологій, як Vue.js

і Node.js, що використовуються в проєкті, а також є безкоштовним, відкритим та потужним інструментом для розробників.

Запропоновано метод планування навчанням, який на основі метрик успішності студентів дозволяє більш ефективно відслідковувати їхній навчальний прогрес. Завдяки поділу предметів на категорії та розрахунку середнього балу для кожної з них, викладачі та студенти отримують можливість детально аналізувати хід навчання і вживати заходів для його покращення. Впровадження такої системи сприятиме індивідуалізації навчальних планів і покращенню роботи з відстаючими студентами.

Дієвість методу можна перевірити за допомогою емпіричного аналізу результатів студентів, але для цього потрібен деякий час.

3 ОПИС ПРОГРАМИ

3.1 Структура додатку системи планування навчального процесу

Структура програмного забезпечення системи планування навчального процесу наведено на рис. 3.1.

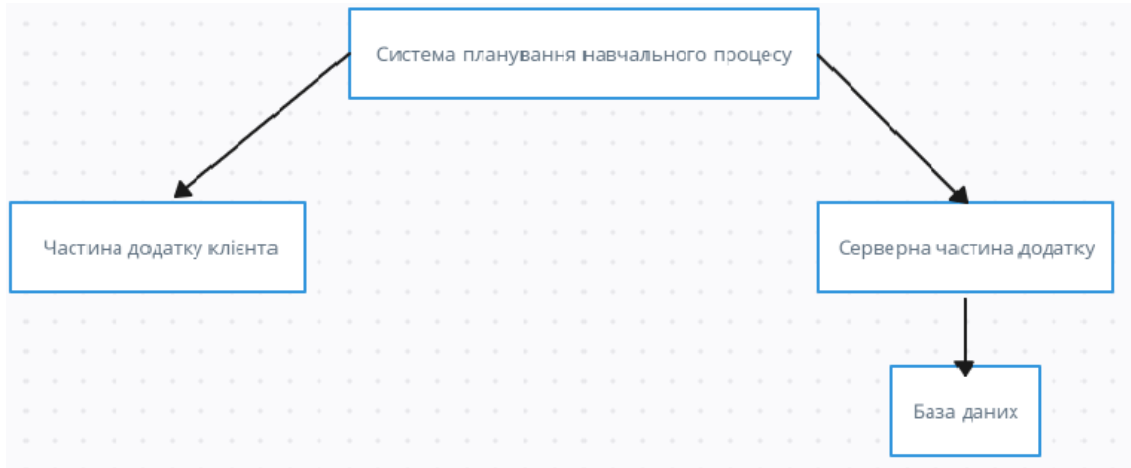


Рисунок 3.1 – Структура додатку системи планування навчанням

На рисунку зображено основні складові програмного забезпечення:

- частину додатку клієнта;
- серверна частина додатку;
- база даних.

Проведемо аналіз призначення основних компонентів програмного забезпечення додатку системи планування навчального процесу.

Частиною для клієнта об'єкту розробки виступають вебсторінки, створені за допомогою фреймворку Vue.js, гіпертекстової розмітки HTML5 та таблиці каскадних стилів CSS3. Серверна частина додатка була реалізована за допомогою програмної платформи Node.js.

Вебсервіс системи планування навчанням являє собою кросплатформений вебзастосунок, доступ до якого виконується через мережу інтернет за допомогою вебпротоколу HTTPS.

Для зберігання всіх даних вебсервісу була обрана база даних, яка гарантує безперебійний і безпечний доступ до наявної інформації. У процесі

розробки було вирішено використовувати MongoDB, яка відзначається високою швидкістю, надійністю та простотою у використанні.

3.2 Проєктування бази даних додатку системи планування навчанням

У процесі використання додатку системи планування навчанням виникає необхідність зберігати великі обсяги даних. Це зумовлює потребу у впровадженні бази даних для забезпечення коректного функціонування всіх модулів сервісу.

У цій роботі було використано базу даних MongoDB (табл. 3.1). MongoDB — це сучасна, документно-орієнтована система керування базами даних (СКБД), яка належить до категорії NoSQL. Вона була розроблена для зберігання і обробки великих обсягів неструктурованих даних, таких як текст, мультимедійні файли або дані з вебдодатків.

Серед основних можливостей MongoDB можна виділити:

- гнучкість даних. MongoDB дозволяє зберігати документи різних структур у межах однієї колекції, що забезпечує гнучкість і можливість легкого оновлення схем даних;
- масштабованість. MongoDB підтримує горизонтальне масштабування, що дозволяє розподіляти дані між кількома серверами для забезпечення високої продуктивності і надійності;
- висока продуктивність. Завдяки індексації та агрегаційним операціям MongoDB забезпечує швидкий доступ до даних і підтримує складні запити;
- документальна модель. Замість традиційних таблиць, як у реляційних базах даних, MongoDB використовує документи у форматі BSON (розширений формат JSON), які дозволяють зберігати складні, вкладені структури даних;

– реплікація та відновлення. MongoDB дозволяє налаштовувати реплікацію даних через кластер серверів, що забезпечує високу доступність і швидке відновлення після збоїв [15].

MongoDB підходить для застосування, де потрібна гнучкість у роботі з даними, масштабованість і швидка реакція на зміну вимог. Наприклад, коли ми створюємо:

- веб-додатки та мобільні додатки;
- системи аналізу великих даних (Big Data);
- інтернет-магазини, соціальні мережі та платформи для зберігання мультимедійних файлів.

MongoDB стала популярною через свою здатність обробляти великі та динамічні набори даних, забезпечуючи при цьому високу продуктивність і простоту в управлінні (табл. 3.1) [16].

Таблиця 3.1 – Таблиця порівняння популярних СКБД

Характеристика	MongoDB (NoSQL)	SQL(Реляційні бази даних)
Модель даних	Документна	Таблична
Схема	Безсхемна	Фіксована схема
Запити	Використовує MongoDB Query Language (MQL), схожий на JSON	SQL (Structured Query Language)
Інтеграція з великими даними	Добре підходить для великих і неструктурованих даних	Краще для структурованих даних
Типи даних	Гнучка підтримка різних типів даних	Обмежена типовими значеннями (рядки, числа, дати)

Продовження таблиці 3.1

Характеристика	MongoDB (NoSQL)	SQL(Реляційні бази даних)
Зберігання зв'язків між даними	Складніше через відсутність JOIN-запитів, потрібно використовувати вбудовані документи або посилання	Добре підходить для складних зв'язків між таблицями (JOIN-запити)
Відновлення після збоїв	Підтримує реплікацію та автоматичне відновлення	Підтримка реплікації, але налаштування складніше

Тож, MongoDB краще підходить для додатків, де потрібна гнучкість у структурі даних, та можливість масштабування, особливо для веб-додатків, аналітичних платформ та обробки великих обсягів неструктурованих даних.

Структура даних компонентів системи зображена у таблиці 3.2

Таблиця 3.2 – Структура даних таблиці «Розклад»

Назва ключа	Тип значення	Опис
_id	ObjectId	Забезпечує унікальність запису
classNumber	Number	Номер предмета
day	String	День
startTime	String	Початок пари
endTime	String	Кінець пари
subject	String	Назва предмету
teacher	String	Вчитель, який викладає предмет
category	String	Категорія предметів

Структура даних таблиці «Користувач» представлена у таблиці 3.3.

Таблиця 3.3 – Структура даних таблиці «Користувач»

Назва ключа	Тип значення	Опис
_id	String	Забезпечує унікальність запису
username	String	Ім'я користувача
password	String	Пароль
role	String	Роль користувача (студент/вчитель)

Структура даних таблиці «Завдання» представлена у таблиці 3.4.

Таблиця 3.4 – Структура даних таблиці «Завдання»

Назва ключа	Тип значення	Опис
_id	ObjectId	Забезпечує унікальність запису
subject	String	Назва предмету
taskDescription	String	Описа завдання
deadline	Date	Дедлайн
studentId	Number	Код студента
username	String	Ім'я користувача
completed	Boolean	Виконані завдання
grades	Map	Оцінки
files	Array	Завантажені файли

Структура даних таблиці «Студент» представлена у таблиці 3.5.

Таблиця 3.5 – Структура даних таблиці «Студент»

Назва ключа	Тип значення	Опис
_id	String	Забезпечує унікальність запису
username	String	Ім'я користувача

Продовження таблиці 3.5

Назва ключа	Тип значення	Опис
role	String	Роль користувача

На рисунку 3.2 зображено взаємозв'язок між таблицями системи планування навчального процесу.

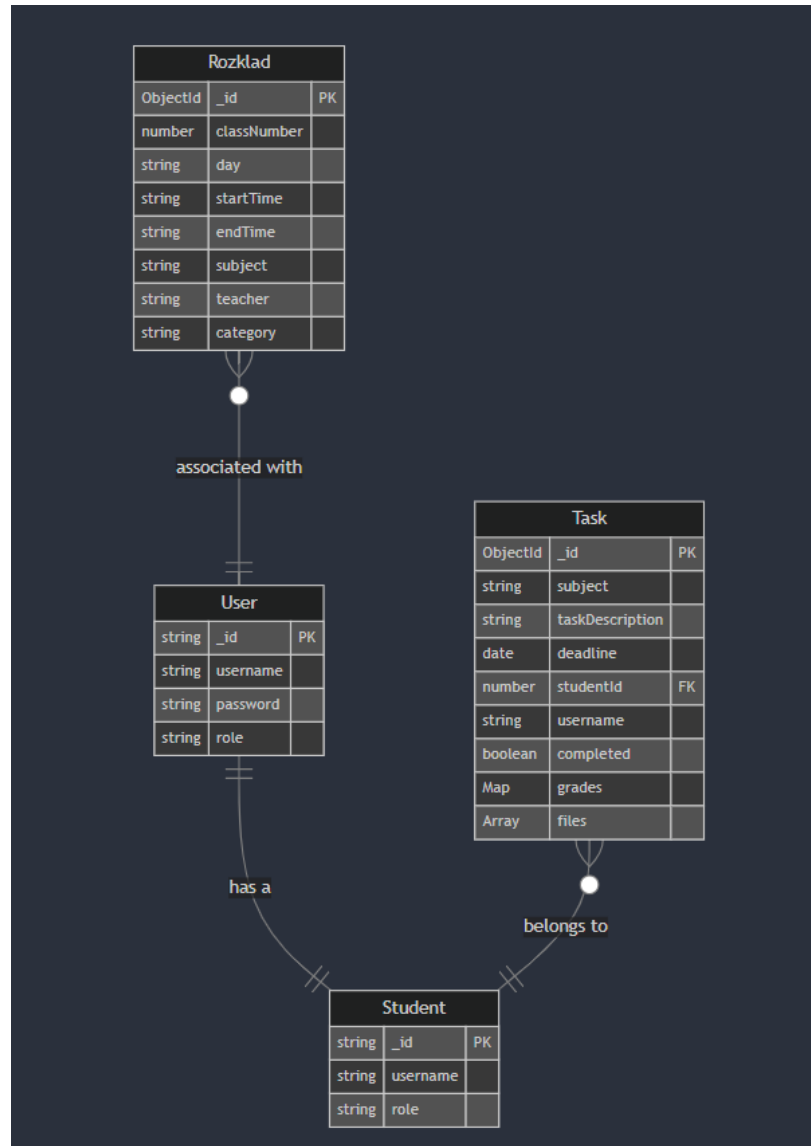


Рисунок 3.2 – Взаємозв'язок між таблицями

За допомогою рисунку можна побачити, що у таблиці «User» до таблиці «Rozklad» взаємозв'язок «один к багатьом», таблиця «Student» до

таблиці «User» взаємозв'язок «один до одного», у таблиці «Student» до таблиці «Task» взаємозв'язок «один до багатьох».

3.3 Функціонування системи управління навчальним процесом

Схема функціонування додатку системи планування навчальним процесом зображено на рис. 3.3.

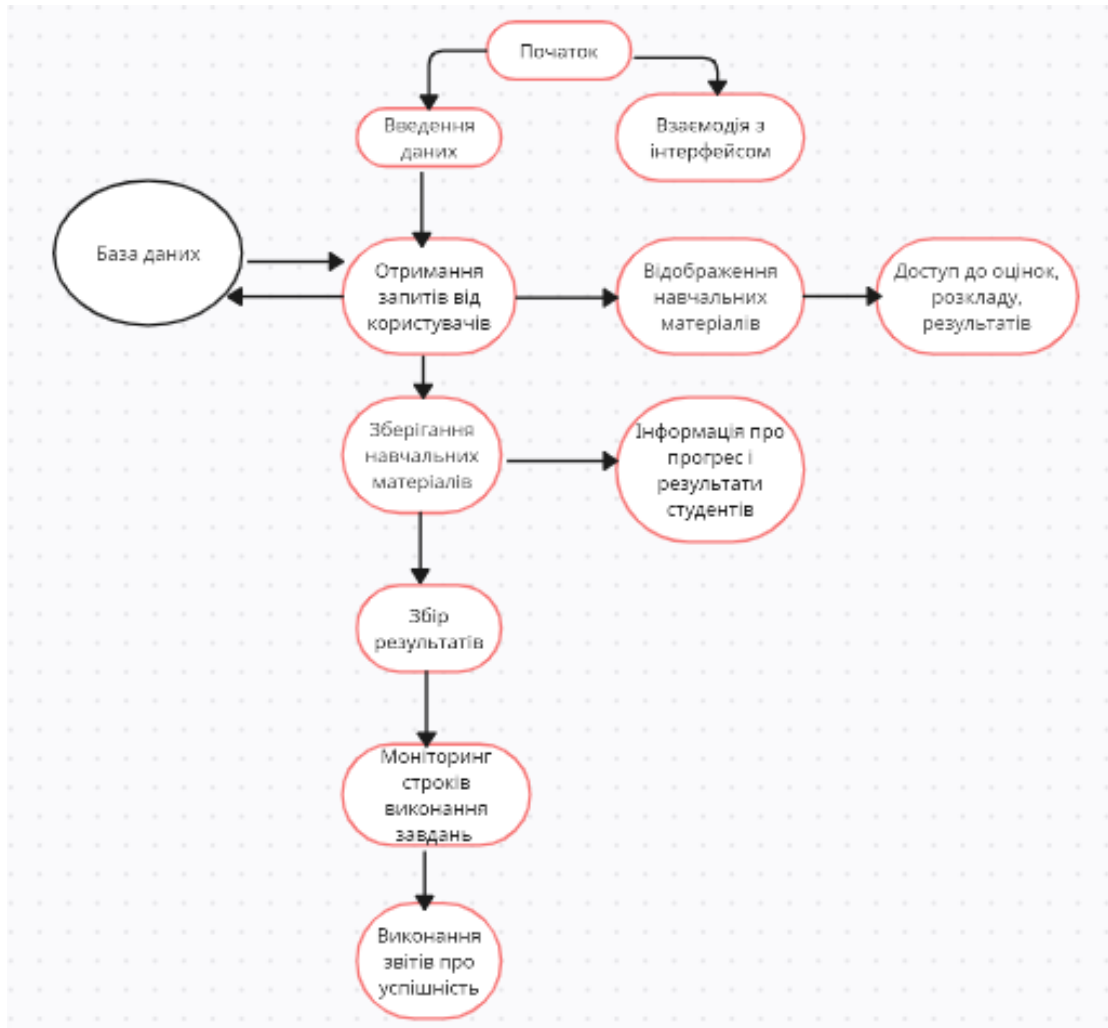


Рисунок 3.3 – Схема функціонування додатку

3.4 Авторизація

Для того, щоб почати користуватись додатком треба авторизуватись. Користувач вводить логін та пароль, щоб потрапити у систему. Якщо користувач є студентом, то він вводить логін – student1, пароль – 123. Якщо в

систему ми заходимо як вчитель, то логін – teacher, пароль – 123. При реєстрації, кожний користувач придумує свої дані. На рисунку 3.4 зображено вікно авторизації.



Рисунок 3.4- Вікно авторизації

```
const mongoose = require('mongoose');
const UserSchema = new mongoose.Schema({
  _id: { type: String },
  username: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  role: { type: String, enum: ['student', 'teacher'], required: true }
});
module.exports = mongoose.model("User", UserSchema);
const UserController = require('../controllers/userController');
module.exports = function(app) {
  app.route('/login')
    .post(UserController.login_user);
  app.route('/register')
    .post(UserController.register_user);
};
const User = require('../models/userModel');
const Student = require('../models/studentModel');
exports.login_user = async function(req, res) {
  const { username, password } = req.body;
```

```

    try {
      const user = await User.findOne({ username, password })
      if (!user) {
        return res.status(401).json({ message: "Неправильний логін або
пароль" });
      }
      res.status(200).json({
        message: "Успішний вхід",
        user: {
          id: user._id,
          username: user.username,
          role: user.role}); } catch (error) {
        console.error('Помилка при логіні користувача:', error);
        res.status(500).json({ message: "Внутрішня помилка сервера" });
      });
    }
    exports.register_user = async function(req, res) {
      const { id, username, password, role } = req.body;
      try {
        const existingUserById = await User.findById(id);
        if (existingUserById) {
          return res.status(400).json({ message: "Користувач з таким ID
вже існує" });
        }
        const existingStudent = await Student.findOne({ username });
        if (existingStudent) {
          return res.status(400).json({ message: "Студент з таким іменем
вже існує" });
        }
        const newUser = new User({ _id: id, username, password, role });
        await newUser.save();
        if (role === 'student') {
          const newStudent = new Student({
            _id: id,

```

```

    username,
    role: 'student' });
await newStudent.save(); }

```

3.5 Метод планування системи управління навчанням

Метод планування такий: спочатку збираються оцінки з усіх предметів у певній категорії, потім для кожного студента обчислюється середня оцінка по цій категорії. Якщо середнє значення оцінки в категорії нижче встановленого порога, наприклад, 3,5, це означає, що студент потребує додаткової підтримки чи уваги з цих предметів.

```

<template>
  <div class="performance-chart">
    <h2>Статистика по предметам та завданням</h2>
    <div class="charts-container">
      <div class="chart-wrapper">
        <h3>Розподіл предметів по категоріях</h3>
        <div class="chart-container">
          <canvas ref="subjectsChartCanvas"></canvas>
        </div>
      </div>
      <div class="chart-wrapper">
        <h3>Статус виконання завдань</h3>
        <div class="chart-container">
          <canvas ref="tasksChartCanvas"></canvas>
        </div>
      </div>
      <div class="chart-wrapper full-width">
        <h3>Успішність студента</h3>
        <div class="select-container">

```

```

<select v model="selectedStudentId"
@change="updateStudentPerformance">
<option v-for="student in students" :key="student.id" :value="student.id">
    {{ student.username }}
</option>
</select>
</div>

<div class="chart-container">
    <canvas ref="studentPerformanceCanvas"></canvas>
</div>
</div>

<div>
    <h3 id="totalAverage">Загальна середня оцінка: 0.00</h3>
</div>

<div class="recommendation-section">
    <h3>Рекомендації та відомість оцінок</h3>
    <div class="improvement-suggestions">
        <ul ref="improvementList"></ul>
    </div>
    <div class="grades-report">
        <table>
            <thead>
                <tr>
                    <th>Предмет</th>
                    <th>Оцінка</th>
                </tr>
            </thead>
            <tbody>
                <tr v-for="(grade, index) in studentGrades" :key="index">
                    <td>{{ grade.subject }}</td>

```

```

        <td>{{ grade.averageGrade.toFixed(2) }}</td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</template>
<script>
import { ref, onMounted } from 'vue';
import axios from 'axios';
import Chart from 'chart.js/auto';
export default {
  name: 'PerformanceChart',
  setup() {
    const subjectsChartCanvas = ref(null);
    const tasksChartCanvas = ref(null);
    const studentPerformanceCanvas = ref(null);
    const improvementList = ref(null);
    const students = ref([]);
    const selectedStudentId = ref(null);
    const studentGrades = ref([]);
    let studentPerformanceChart = null;
    const fetchSubjectsData = async () => {
      try {
        const response = await axios.get('http://10.0.0.102:3000/rozklad');
        return response.data;
      } catch (error) {

```

```

        console.error('Помилка при отриманні даних про предмети:',
error);

        return []; } };

const fetchTasksData = async () => {
  try {
    const response = await axios.get('http://10.0.0.102:3000/tasks');
    return response.data;
  } catch (error) {
    console.error('Помилка при отриманні даних про завдання:', error);
    return [];});});

const processSubjectsData = (data) => {
  const categories = {};
  data.forEach(item => {
    if (!categories[item.category]) {
      categories[item.category] = 0; }
    categories[item.category]++; });
const percentage = ((value / total) * 100).toFixed(1);
    return `${label}: ${value} (${percentage}%)`;}}}}}}});});

const createStudentPerformanceChart = (performanceData) => {
  const ctx = studentPerformanceCanvas.value.getContext('2d');
  if (studentPerformanceChart) {
    studentPerformanceChart.destroy(); }
  studentPerformanceChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: performanceData.map(item => item.subject),
      datasets: [{
        label: 'Середня оцінка',
        data: performanceData.map(item => item.averageGrade),
        fill: false,

```

```

        borderColor: 'rgb(75, 192, 192)',
        tension: 0.1]],
options: {
  responsive: true,
  maintainAspectRatio: false,
  scales: {y: {
    beginAtZero: true,
    max: 5,
    title: {
      display: true,
      text: 'Середня оцінка'}},
    x: {
      title: {
        display: true,
        text: 'Предмети'}} } } }));
const displayImprovementSuggestions = (performanceData) => {
  const suggestions = performanceData.filter(item => item.averageGrade
< 3.5);

  const improvementListElement = improvementList.value;
  improvementListElement.innerHTML = "";
  if (suggestions.length > 0) {
    suggestions.forEach(item => {
      const li = document.createElement('li');
      li.textContent = `Підтягніть: ${item.subject} (Середня оцінка:
${item.averageGrade.toFixed(2)})`;
      improvementListElement.appendChild(li);});
  } else {
    improvementListElement.textContent = 'Всі предмети на рівні.';});
const displayTotalAverage = (totalAverage) => {
  const averageDisplay = document.getElementById('totalAverage');

```

```

        averageDisplay.textContent = `Загальна середня оцінка:
        ${totalAverage.toFixed(2)}`;};

        const updateStudentPerformance = async () => {
            const tasksData = await fetchTasksData();

            const { averageGrades, totalAverage } =
processStudentPerformance(tasksData, selectedStudentId.value);

            createStudentPerformanceChart(averageGrades);
            displayImprovementSuggestions(averageGrades);
            displayTotalAverage(totalAverage); };

        onMounted(async () => {
            const subjectsData = await fetchSubjectsData();
            const tasksData = await fetchTasksData();
            const processedSubjectsData = processSubjectsData(subjectsData);
            const processedTasksData = processTasksData(tasksData);
            const uniqueStudentsMap = new Map();
            tasksData.forEach(task => {
                if (!uniqueStudentsMap.has(task.username)) {
                    uniqueStudentsMap.set(task.username, task.studentId); });
            students.value = Array.from(uniqueStudentsMap, ([username, id]) => ({
                id,
                username }));
            if (students.value.length > 0) {
                selectedStudentId.value = students.value[0].id;}

```

3.6 Додавання предметів

У додатку системи планування навчального процесу можна додати предмети (якщо користувач у ролі викладача), для того щоб студенти бачили, які завдання їм треба робити. Також у додатку, як вкості користувача студента, можна загрузити файли для того щоб здати завдання.

```

const mongoose = require('mongoose');
const taskSchema = new mongoose.Schema({
  subject: { type: String, required: true },
  taskDescription: { type: String, required: true },
  deadline: { type: Date, required: true },
  studentId: { type: Number },
  username: { type: String },
  completed: { type: Boolean, default: false },
  grades: { type: Map, of: Number },
  files: [{
    filename: String,
    path: String,
    uploadDate: Date,
    uploadedBy: String,
  }],
});
const Task = mongoose.model('Task', taskSchema);
module.exports = Task;
const Task = require('../controllers/taskController');
module.exports = function(app) {
  app.route('/tasks')
    .get(Task.list_of_tasks)
    .post(Task.create_a_task);
  app.route('/tasks/:id')
    .put(Task.update_a_task)
    .delete(Task.delete_a_task);
  app.route('/tasks/:id/upload')
    .put(Task.upload_files, Task.update_task_with_files);
  app.route('/tasks/:taskId/files/:fileId')
    .delete(Task.delete_file);
};

```

```

const Task = require('../models/taskModel');
const multer = require('multer');
const path = require('path');
const Student = require('../models/studentModel');
const fs = require('fs').promises;
exports.get_user_by_id = async function(req, res) {
  const { id } = req.params;
  try {
    const user = await User.findById(id);
    if (!user) {
      return res.status(404).json({ message: "Користувача не знайдено"
}); }

    res.status(200).json({
      id: user._id,
      username: user.username,
      role: user.role
    }); } catch (error) {
    console.error('Помилка при отриманні користувача:', error);
    res.status(500).json({ message: "Внутрішня помилка сервера" });});
const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, 'uploads/');},
  filename: function (req, file, cb) {
    const timestamp = new Date().toISOString().replace(/[:.-]/g, '_');
    const fileType = path.extname(file.originalname);
    const newFileName = `file_${timestamp}${fileType}`;
    cb(null, newFileName); } });
const upload = multer({ storage: storage });
exports.upload_files = upload.array('files', 5);
exports.update_task_with_files = async function(req, res) {

```

```

try {
  const taskId = req.params.id;
  const username = req.body.username;
  const files = await Promise.all(req.files.map(async (file) => {
    const filePath = file.path;
    const fileType = path.extname(file.originalname).toLowerCase()
    if (fileType === '.txt' || fileType === '.csv') {
      const data = await fs.readFile(filePath, 'utf8');
      await fs.writeFile(filePath, data, { encoding: 'utf8' }); }
    return {
      filename: file.originalname,
      path: file.path,
      uploadDate: new Date(),
      uploadedBy: username    }; }));
  const updatedTask = await Task.findByIdAndUpdate(
    taskId, {
      $push: { files: { $each: files } },
      completed: true },
    { new: true, runValidators: true } );
  res.status(200).json(updatedTask);
} catch (err) {
  res.status(400).json({ message: 'Error updating task', error: err }); }
}

exports.list_of_tasks = async function(req, res) {
  try {
    const studentId = req.query.userId;
    const filter = {};
    if (studentId) {
      filter.studentId = studentId;    }
    const tasks = await Task.find(filter);
    res.json(tasks);
  }

```

```

    } catch (error) {
      console.error("Error fetching tasks:", error);
      res.status(500).send("Internal Server Error"); });
exports.create_a_task = async function(req, res) {
  try {
    const newTask = new Task({
      subject: req.body.subject,
      taskDescription: req.body.taskDescription,
      deadline: req.body.deadline,
      studentId: req.body.studentId,
      username: req.body.username,
      completed: req.body.completed || false, });
    const task = await newTask.save();
    const populatedTask = await Task.findById(task._id);
    res.status(201).json(populatedTask);
  } catch (err) {
    console.error('Error saving to MongoDB:', err);
    res.status(400).json({ message: 'Failed to create task', error: err.message });
  });
exports.update_a_task = async function(req, res) {
  const { id } = req.params;
  const { studentId, grade } = req.body;
  try {const updatedTask = await Task.findByIdAndUpdate(
    id, {
      $set: {
        [`grades.${studentId}`]: grade,
        completed: true  }  },
    { new: true, runValidators: true } );
    if (!updatedTask) {
      return res.status(404).send("Task not found"); }

```

```

    res.status(200).send(updatedTask);
  } catch (error) {
    console.error('Error updating task:', error);
    res.status(400).send(error.message || "Bad request"); }};
exports.delete_a_task = async function(req, res) {
  try {
    const deleted_task = await Task.findByIdAndDelete(req.params.id);
    if (!deleted_task) {
      return res.status(404).send("Task not found"); }
    res.status(200).send(deleted_task);
  } catch (error) {
    console.error('Error deleting task:', error);
    res.status(500).send("Internal Server Error"); }};
exports.delete_file = async function(req, res) {
  try {
    const { taskId, fileId } = req.params;
    console.log(`Спроба видалення файлу. TaskID: ${taskId}, FileID:
    ${fileId}`);
    const task = await Task.findById(taskId);
    if (!task) {
      console.log(`Завдання не знайдено. TaskID: ${taskId}`);
      return res.status(404).send("Завдання не знайдено"); }
    const fileIndex = task.files.findIndex(file => file._id.toString() ===
    fileId);
    if (fileIndex === -1) {
      console.log(`Файл не знайдено в завданні. TaskID: ${taskId},
    FileID: ${fileId}`);
      return res.status(404).send("Файл не знайдено");}
    const fileToDelete = task.files[fileIndex];
    const filePath = path.join(__dirname, '..', fileToDelete.path);

```

```

        console.log(`Спроба видалення файлу з файлової системи. Шлях:
${filePath}`);
        try {
            await fs.unlink(filePath);
            console.log(`Файл успішно видалено з файлової системи. Шлях:
${filePath}`);
        } catch (error) {
            console.error(`Помилка при видаленні файлу з файлової системи:
${error.message}`); }
        task.files.splice(fileIndex, 1);
        if (task.files.length === 0) {
            task.completed = false;    }
        await task.save();
        console.log(`Інформацію про файл видалено з бази даних. TaskID:
${taskId}, FileID: ${fileId}`);
        console.log(`Статус завдання: ${task.completed ? 'виконане' :
'невиконане'}`);
        res.status(200).send(task);
    } catch (error) {
        console.error('Помилка видалення файлу:', error);
        res.status(500).send(`Внутрішня помилка сервера:
${error.message}`); } };
```

3.7 Висновки за розділом 3

Було розроблено програмне забезпечення для планування та моніторингу навчального процесу. Відображено структуру системи. У ході розробки вебсервісу були визначені його основні складові: частина додатку клієнта, що являє собою інтерфейс, який забезпечує передачу інформації між користувачем та програмними компонентами, серверна частина додатку, яка

забезпечую обробку запитів користувачів, авторизацію, метод планування навчальним процесом та різноманітні взаємодії з компонентами.

Описано схему функціонування системи планування навчальним процесом.

Здійснено проєктування бази даних системи планування навчального процесу.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення й умови застосування програми

Програмне забезпечення призначене для покращення якості освіти, організації навчального процесу та взаємодії між усіма учасникам освітнього процесу. Додаток розроблений на таких мовах програмування як, Vue.js, Node.js, Html, CSS. Для розробки вебсервісу було обрано Microsoft Visual Studio Code. Цей кросплатформений інструмент є безкоштовним і підтримує багато мов програмування. Він ідеально підходить для створення вебзастосунків та хмарних рішень. Для створення бази даних було обрано MongoDB.

Програмне забезпечення системи планування навчального процесу забезпечує виконання таких функцій:

- розклад пар (можна створити, редагувати та дивитися);
- збереження домашніх завдань, лабораторних робіт;
- встановлення дедлайнів;
- додавання важливих подій (контрольні роботи або тести);
- список найприоритетніших справ на основі домашніх робіт та зазначених дат;
- можливість відзначити виконані завдання;
- можливість додавання оцінок.

Для комфортного використання програмного забезпечення системи планування навчального процесу необхідні такі апаратні та програмні компоненти:

- ОЗУ - не менш 4 Гб;
- операційна система з можливістю взаємодії з програмними інтерфейсами через мережу Інтернет;
- відео адаптер: DirectX 10;
- процесор – не менш 1 ГГц
- мережний адаптер.

- програмне забезпечення для перегляду вебсторінок.

Функціональні характеристики розробленого додатку наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення оптимізації режимів обробки технічних виробів наведено у додатку Б.

4.2 Характеристики розробленого продукту

Програмне забезпечення системи планування навчанням функціонує постійно, адаптуючись до потреб користувачів. Швидкість взаємодії з вебсервісом визначається пропускнуою здатністю вебшлюзу сервера, на якому розміщена система, а також обсягом вебтрафіку користувача і продуктивністю його комп'ютерного обладнання та програмного забезпечення.

Під час роботи програмне забезпечення системи управління навчальним процесом надає інформацію користувачу у вигляді сформованих вебсторінок на яких відображається наступна інформація:

- авторизація;
- відображення для користувача всього розкладу занять;
- можливість додавати предмети самостійно;
- можливість завантажувати файли;
- можливість проставляти оцінки за виконані завдання;
- отримування повідомлень;
- створення графіків успішності;
- створення графіків розподілу по категоріям предметів та завдань та створення діаграм статусу виконання завдань;
- підрахування загальної середньої оцінки.

Програмне забезпечення системи планування навчанням під час своєї роботи активно використовує можливості сучасних систем керування базами даних (СКБД) для обробки великих обсягів інформації, що генерується в процесі функціонування вебдодатку.

4.3 Структурна схема компонентів вебсервісу

Створений додаток умовно можна поділити на дві основні логічні частини. Першою частиною є клієнтський модуль, де користувачі програми можуть ознайомитися з розкладом занять, додавати чи видаляти предмети зі списку і т.д.. Користувачі можуть сдавати свої домашні завдання, практичні роботи та отримувати за них оцінки, отримуючи повідомлення які предмети треба підтягнути.

Друга логічна частина вебсервісу — адміністративний модуль, призначений для розробників або менеджерів, які після автентифікації можуть редагувати базу даних. У цій частині вони мають можливість змінювати наявні записи та додавати нові елементи. У той час клієнтський модуль не має доступу до редагування даних — всі функції, пов'язані з модифікацією, заблоковані для загального користування.

4.4 Висновки за розділом 4

У цьому розділі розглянуто ключове призначення та умови використання додатку, призначеного для планування навчального процесу. Представлено основні характеристики вебсервісу та описано зміст основних інформаційних сторінок. Також проаналізовано структурну схему компонентів програми, яка відображає логічну взаємодію елементів, розроблених у додатку.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Опис роботи з програмою

5.1.1 Звернення до програми

Клієнти можуть звертатися до вебсервісу системи управління навчальним процесом через будь-який сучасний веббраузер, який підтримує виконання вебскриптів.

5.1.2 Вхідні й вихідні дані

Вхідними даними під час авторизації є логін користувача та пароль. Вихідними є підтвердження входу та розклад. Після авторизації на вході є роль користувача для визначення прав доступу, на виході є повний розклад для відображення, доступ до функцій додавання, редагування, видалення для «викладача».

Завдання для студентів: вхідними даними є ідентифікатор студента, щоб показати його завдання `studentId`, вихідними список завдань з оцінками. Управління завданнями: вхідні дані – це нове завдання або оцінка студента, вихідними – оновлена інформація про завдання.

Перелік завдань, призначених студентам та візуалізація даних (статистика): на вході є оцінки студентів по предметам на виході – візуальні графіки з успішністю за предметами.

5.1.3 Повідомлення

Користувач додатку системи управління навчальним процесом в процесі взаємодії з вебсервісом може отримати вебповідомлення після оновлення під час проставлення оцінок. Також коли користувач оновлює

завдання та коли користувач здає завдання.

5.2 Інструкція з використання

Після переходу за веб-адресою <http://10.0.0.106:5173/>, перед користувачем з’являється сторінка авторизації (рис. 5.1).

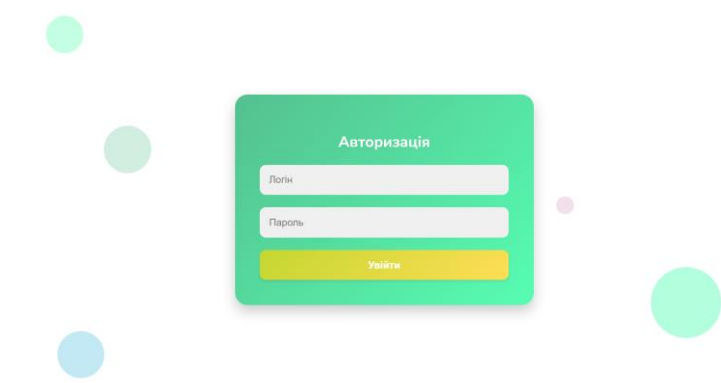


Рисунок 5.1 – Вікно авторизації

На цій сторінці користувач вводить свої дані (логін та пароль). Для прикладу, якщо користувач входить систему як вчитель, було обрано базові дані: login – teacher, password – 123. Якщо користувач входить до системи як студент, то login – student1, password – 123. Після авторизації, користувач потрапляє на головну сторінку додатку (рис. 5.2).

☰

Система планування навчального процесу

Розклад занять

Додати заняття

№	ЧАС	ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР	П'ЯТНИЦЯ
1	11:14 - 12:14	ПБП Іванов В.І. ✔ ✖	Геометрія Кірінова В.А. ✔ ✖	Алгебра Моїсєнко Т.А. ✔ ✖	Штучний інтелект Ковальчук Д.М. ✔ ✖	
2	12:17 - 13:17	Розробка Петров М.Р. ✔ ✖	ПБП Іванов В.І. ✔ ✖		Менеджмент Носенко О.Г. ✔ ✖	Психологія Сидоренко С.С. ✔ ✖
3	13:31 - 14:31		Література Шевченко К.М. ✔ ✖	Українська мова Колісник Д.О. ✔ ✖		Фізика Михайличук Ю.Р. ✔ ✖
4	15:01 - 16:01				Операційні системи Сидоров В.Л. ✔ ✖	

Рисунок 5.2 – Головна сторінка системи управління навчанням

На головній сторінці як для студента так і для вчителя зображено розклад занять, розписаний по тижням. У лівому верхньому куту є кнопка, яка відкриває панель з різними критеріями. Для студента – «Мої завдання», «Розклад», «Графік успішності» та кнопка «Вийти» (рис. 5.3).

×

Мої завдання

Розклад

Графіки успішності

Вийти

Система планування навчального процесу

Розклад занять

	ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР	П'ЯТНИЦЯ
1 11:14 - 12:14	ПБП Іванов В.І.	Геометрія Кірюнова В.А.	Алгебра Моїсєєнко Т.А.	Штучний інтелект Ковальчук Д.М.	
2 12:17 - 13:17	Розробка Петров М.Р.	ПБП Іванов В.І.		Менеджмент Носенко О.Г.	Психологія Сидоренко С.С.
3 13:31 - 14:31		Література Шевченко К.М.	Українська мова Колісник Д.О.		Фізика Михальчук Ю.Р.
4 15:01 - 16:01				Операційні системи Сидоров В.Л.	

Рисунок 5.3 – Панель критеріїв для студента

Для вчителя – «Завдання», «Розклад», «Графіки успішності» (рис 5.4).

×

Завдання

Розклад

Графіки успішності

Вийти

Система планування навчального процесу

Розклад занять

№	ЧАС	ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР	П'ЯТНИЦЯ
1	11:14 - 12:14	ПБП Іванов В.І. ✓	Геометрія Кірюнова В.А. ✓	Алгебра Моїсєєнко Т.А. ✓	Штучний інтелект Ковальчук Д.М. ✓	
2	12:17 - 13:17	Розробка Петров М.Р. ✓	ПБП Іванов В.І. ✓		Менеджмент Носенко О.Г. ✓	Психологія Сидоренко С.С. ✓
3	13:31 - 14:31		Література Шевченко К.М. ✓	Українська мова Колісник Д.О. ✓		Фізика Михальчук Ю.Р. ✓
4	15:01 - 16:01				Операційні системи Сидоров В.Л. ✓	

Рисунок 5.4 – Панель критеріїв для вчителя

Зверху для вчителя є кнопка «Додати заняття». Ця кнопка дозволяє користувачу викладати нові завдання для студентів (рис. 5.5).

Додати нове заняття

Номер пари:

День:

Час початку:

Час закінчення:

Предмет:

Викладач:

Додати Скасувати

Рисунок 5.5 – Кнопка «Додати заняття»

На цьому рисунку можна побачити, що користувач може додати такі пункти: номер пари, день, час початку та час закінчення, сам предмет, та викладача, який веде цей предмет. Після заповнення вікна, користувач натискає кнопку або додати або скасувати.

Якщо (для вчителя) користувач натискає кнопку «Завдання», відкривається вікно «Завдання та Дедлайни», де користувач бачить які завдання та хто зі студентів зробив їх. На рисунку 5.6 можна побачити, що вчитель може обрати які саме предмети він веде, зробити фільтрацію по предметам, дату, може обрати студента. Нижче зображено приклади, коли студенти здають свої завдання, прикріплюючи свої файли. Користувач, як вчитель, може релагувати, видалити або поставити оцінку за завдання.

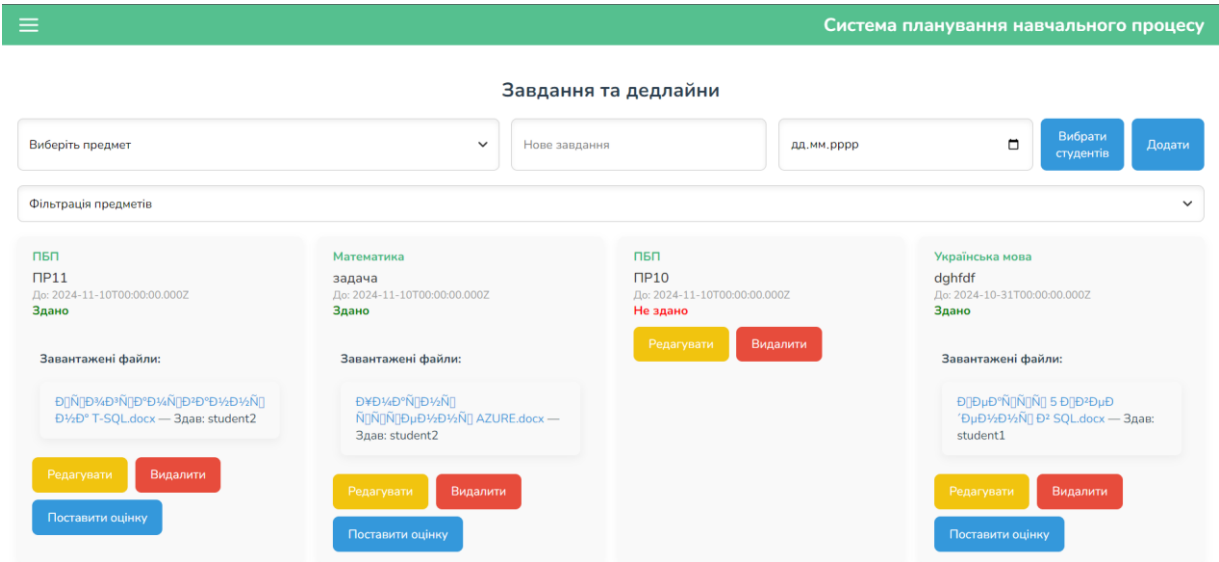


Рисунок 5.6 – Вікно «Завдання та дедлайни»

Вчитель може редагувати власні завдання, якщо десь помилився (рис. 5.7).

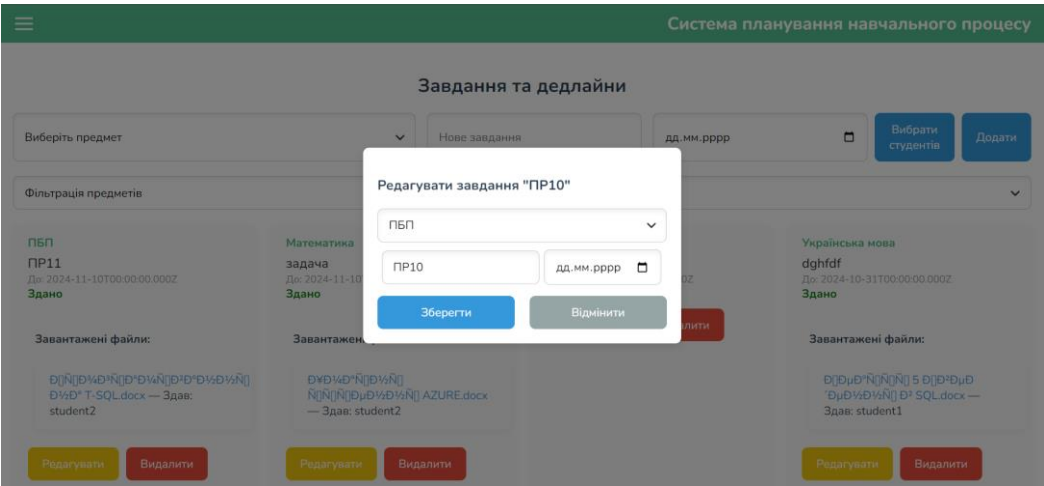


Рисунок 5.7 – Вікно редагування

На рисунку 5.8 зображено вікно, коли вчитель ставить оцінки за виконані завдання.

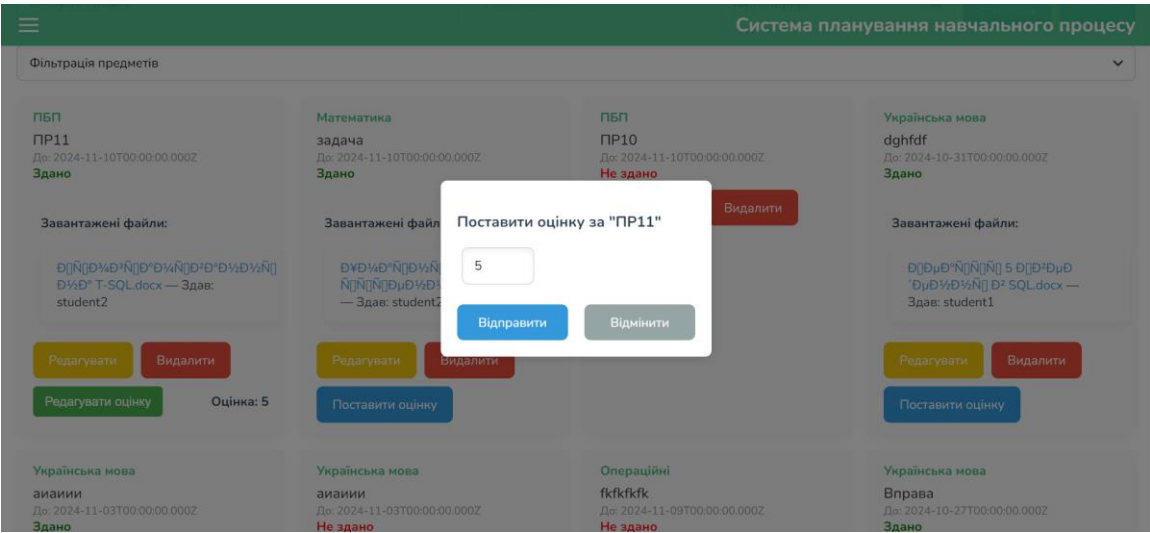


Рисунок 5.8 – Вікно «Поставити оцінку»

На рисунку 5.9 зображено вікно «Завдання для студента». Тут ми бачимо верхню панель, яка містить кнопки «Усі завдання», «Здані», «Не здані», «Пропущені терміни». Нижче зображені предмети, які треба сдати, або які вже здані та за них виставлена оцінка.

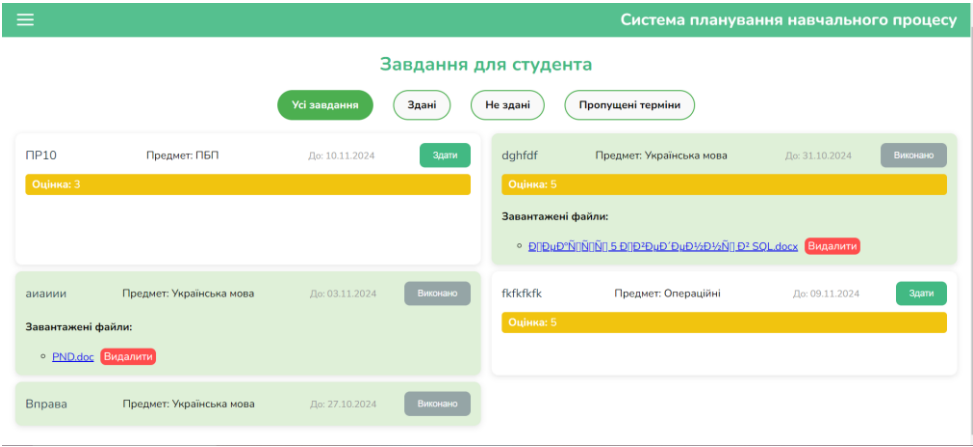


Рисунок 5.9 – Вікно «Завдання для студента»

Студент який хоче здати якесь завдання, він натискає на кнопку «Здати» та бачить вікно «Завантажити файли» (рис. 5.10).

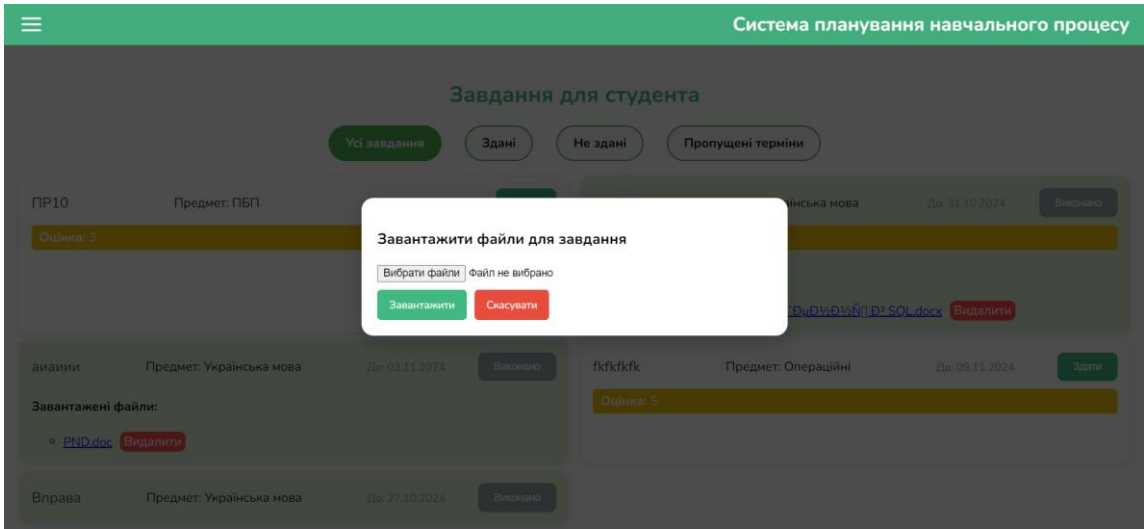


Рисунок 5.10 – Вікно «Завантажити файли»

В системі управління навчальним процесом існують повідомлення. Для вчителя, коли він редагує або виставляє завдання (рис.5.11).

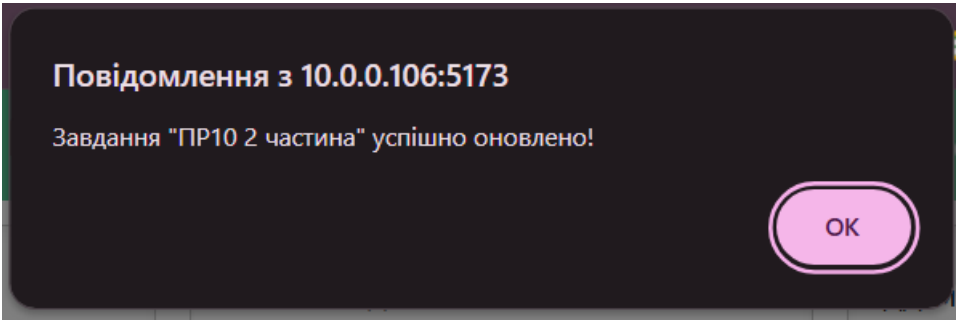


Рисунок 5.11 – Вікно повідомлення для вчителя

Для студента повідомлення приходить, коли він додає своє завдання (рис 5.12).

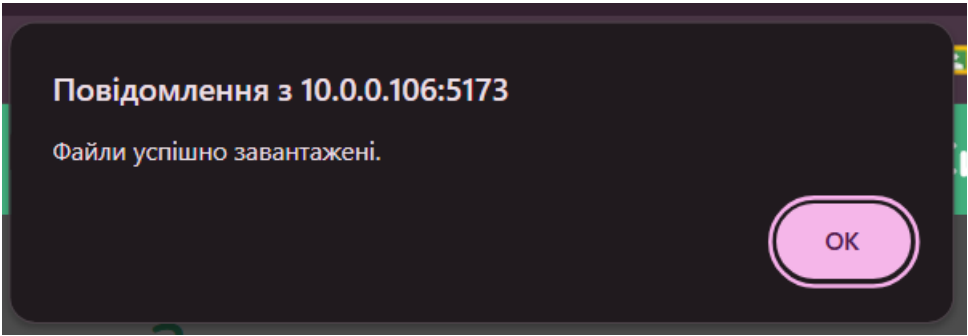


Рисунок 5.12 – Вікно повідомлення для студентів

Після натискання на кнопку «Графік успішності» користувач може побачити власну статистику по предметам та завданням. Наприклад розподіл предметів по категоріям та завдання, які здав/ не здав користувач (статус виконання завдань) (рис. 5.13).

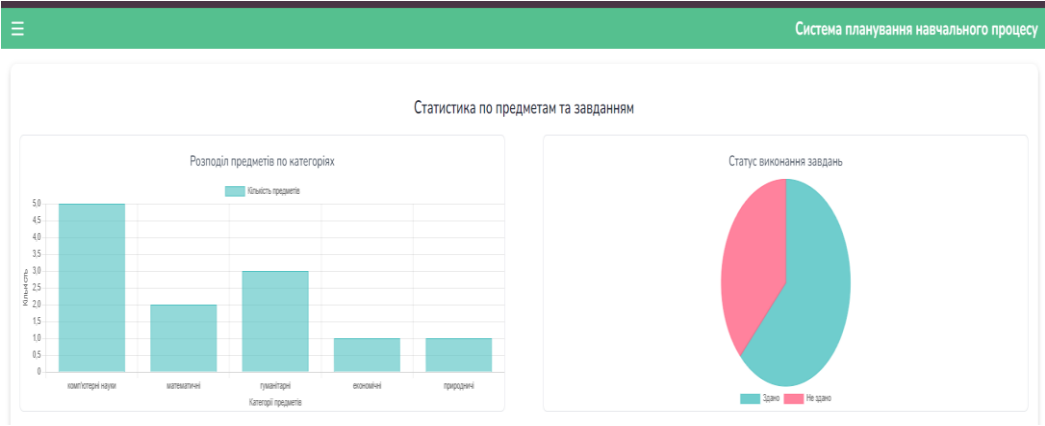


Рисунок 5.13 - Статистика по предметам та завданням

Також користувач може побачити графік успішності студента по предметам (рис. 5.14). У правому верхньому куту користувач бачить свою загальну середню оцінку.

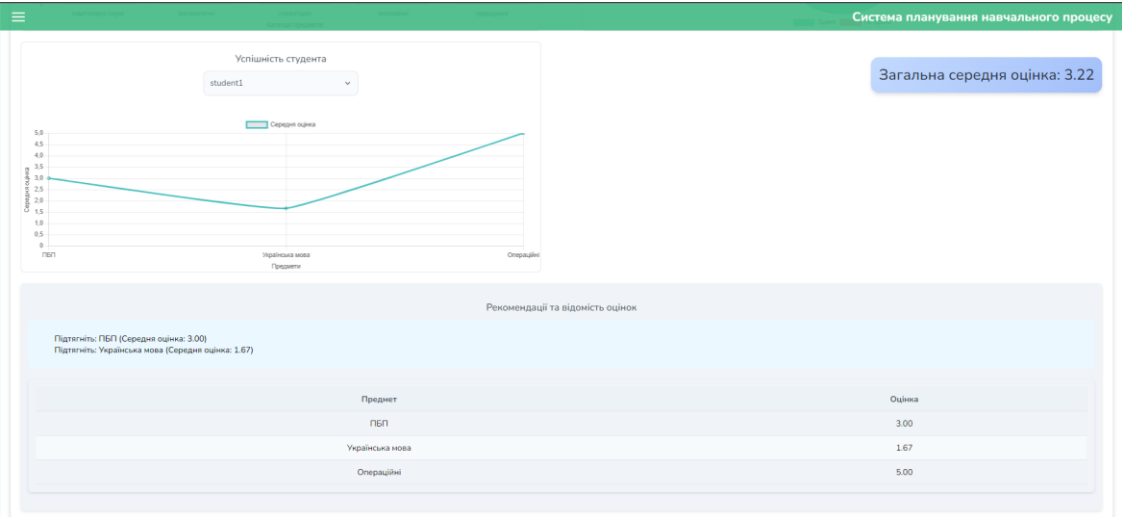


Рисунок 5.14 – Графік успішності

5.3 Експериментальне дослідження (тестування)

Основним критерієм оцінки тестування вебсайту системи управління навчальним процесом є його відповідність заданим технічним вимогам.

Відповідно до технічного завдання були визначені такі функціональні вимоги до розроблювального додатку:

- відображення для користувача всього розкладу занять;
- можливість додавати предмети самостійно;
- можливість завантажувати файли;
- можливість проставляти оцінки за виконані завдання;
- отримування повідомлень;
- створення графіків успішності;
- створення графіків розподілу по категоріям предметів та завдань та створення діаграм статусу виконання завдань;
- підрахування загальної середньої оцінки.

Під час виконання дипломної роботи було зроблено порівняння різних систем планування навчального процесу (Moodle, Google Claassroom, Canvas). Під час розробки власної системи, були враховані всі попередні характеристики конкурентів та створено власні, такі яких не має в жодної з перерахованих (графіки успішності, підрахування загальної середньої оцінки).

Результати оцінки відповідності вебдодатку технічному завданню показали, що вебсайт повністю задовольняє встановлені вимоги та реалізує всі необхідні функції.

Тестування показало, що розроблене програмне забезпечення системи управління навчальним процесом повністю відповідає вимогам технічного завдання, демонструючи високі показники ефективності.

5.4 Висновки за розділом 5

Було описано програмне забезпечення для управління навчальним процесом та розроблено інструкцію з його використання. Проведено тестування системи управління навчальним процесом, що включало перевірку на відповідність усім вимогам технічного завдання. Результати показали, що система із впровадженням методом планування навчальним процесом повністю відповідає технічним вимогам і демонструє високі показники ефективності

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено методи планування навчальним процесом та створено систему управління навчальним процесом.

Були розглянуті основні концепції, тенденції розвитку та питання популярності систем планування навчальним процесом. Також виконано огляд і аналіз програмних аналогів, серед яких для детальнішого вивчення виділено такі системи, як Moodle, Google Classroom та Canvas.

Для розробки програмного забезпечення системи управління навчальним процесом обрано мови програмування Vue.js та Node.js, які є ключовими інструментами у веброботі. Вони дозволяють створювати різноманітні вебдодатки та сайти, що відповідають потребам користувачів.

Для створення системи планування навчальним процесом було обрано середовище розробки Visual Studio Code, який надає розширену підтримку багатьом фреймворкам, таким як Vue.js та Node.js, основних технологій, що використовуються у програмі, та є відкритим, потужним та безкоштовним інструментом для розробки.

Запропоновано метод планування навчальним процесом, який базується на аналізі даних про студентів, їх успішності та виявлення ключових показників для оптимізації процесу навчання. На основі даних про успішність по кожному предмету всередині категорії обчислюється середня оцінка з усіх предметів, що належать до цієї категорії. Алгоритм роботи системи виглядає так: збираються оцінки з усіх предметів у категорії, обчислюється середнє арифметичне оцінювання для кожного студента за кожною категорією предметів. Якщо середня оцінка категорії нижче порога, наприклад, 3,5, це вважається індикатором того, що студент потребує додаткової уваги з цієї групи предметів. Цей кордон може бути гнучким залежно від вимог освітнього закладу.

Результати аналізу візуалізовані та представлені користувачеві у зручному форматі. Даний метод зображує результат у вигляді графіків, який наочно показує поточний рівень успішності та області, які вимагають поліпшення.

Дієвість методу можна перевірити за допомогою емпіричного аналізу результатів студентів, але для цього потрібен деякий час.

Було розроблено структуру та описані основні компоненти програмного забезпечення системи планування навчальним процесом. Основні модулі системи включають: користувацький інтерфейс вебсервісу, який забезпечує взаємодію користувача з усіма функціями сервісу; серверну частину, що обробляє запити від інтерфейсу і базу даних, яка зберігає інформацію про всі процеси та взаємодії сервісу.

Описано схему функціонування додатку управління навчальним процесом.

Визначено основне призначення й умови використання системи планування навчального процесу.

Були наведені основні характеристики розроблюваного додатку та був описаний вміст основних інформаційних вебсторінок.

Описано структурну схему розробленого програмного забезпечення.

Було описано програмне забезпечення системи планування навчального процесу. Також була розроблена інструкція з використання додатку. Виконано тестування розробленого програмного забезпечення. Після проведення тестування встановлено, що системи планування навчальним процесом з впровадженим методом планування навчального процесу повністю відповідає вимогам технічного завдання, та за рахунок впровадженого методу має лідуючі показники метрик ефективності серед представлених аналогів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Поняття процесу навчання [Електрон. ресурс]. – Режим доступу: https://eduknigi.com/ped_view.php?id=123#google_vignette
2. Процес навчання та його особливості [Електрон. ресурс]. – Режим доступу: <https://vseosvita.ua/blogs/protses-navchannia-ioho-osoblyvosti-i-funktsii-85954.html>
3. Moodle [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Moodle>
4. Google Classroom [Електрон. ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Google_Classroom
5. Canvas [Електрон. ресурс]. – Режим доступу: <https://www.instructure.com/canvas>
6. Ribeiro, H. Vue.js 3 CookBook: Discover actionable solutions for bulding modern web apps with the latest Vue features and TypeScript / H. Ribeiro. – Washington: Spring, 2020. – 12 p.
7. Daket, J. HTML & CSS: Design and Build Web Sites / J. Daket. – New York: Spring, 2011. – 55 p.
8. Powell, T. HTML & CSS: The Complete Reference, Fifth Edition / T. Powell. – New York: Spring, 2010. – 785 p.
9. Lerner, A. Fullstack Vue: The Complete Guide to Vue.js / A. Lerner, N. Murray, H. Djirdeh. –San Francisco: Spring, 2018. – 332 p.
10. Casciaro, M. Node.js Design Patterns / M. Casciaro. – Hamilton: Spring, 2016. – 221 p.
11. Johnson, B. Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers/ B. Johnson. – New York: Wiley, 2022. – 15 p.
12. Johnson, B. Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers/ B. Johnson. – New York: Wiley, 2022. – 35 p.
13. Johnson, B. Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers/ B. Johnson. – New York: Wiley, 2022. – 213 p.

14. Johnson, B. Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers/ B. Johnson. – New York: Wiley, 2022. – 291 p.
15. Cbodorow, K. MongoDB: The Definitive Guide/ K. Cbodorow. – New York: Springer, 2013. – 386 p.
16. Koplend, R. MongoDB Applied Design Patterns/ R.Koplend. – Washington: Springer, 2013. – 209 p.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для планування навчального процесу.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація методів планування навчального процесу», затверджене наказом Національного університету «Запорізька політехніка» від 07 жовтня 2024 р. за № 398.

A.2 Призначення розробки

Програмний продукт призначений для забезпечення та підтримки процесу планування навчального процесу.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення планування навчального процесу забезпечує виконання таких функцій:

- авторизація;
- відображення для користувача всього розкладу занять;
- можливість додавати предмети самостійно;
- можливість завантажувати файли;
- можливість проставляти оцінки за виконані завдання;
- отримування повідомлень;
- створення графіків успішності;
- створення графіків розподілу по категоріям предметів та завдань та створення діаграм статусу виконання завдань;

– підрахування загальної середньої оцінки.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс додатку системи планування навчального процесу повинен бути зручним та зрозумілим для користувачів.

A.3.3 Вимоги до надійності

Програмне забезпечення системи планування навчального процесу повинно забезпечити надійне та стабільне функціонування.

A.3.4 Умови експлуатації

Для експлуатації програмного забезпечення системи планування навчального процесу у користувача повинен бути наявний пристрій з встановленою операційною системою, з можливістю взаємодії з програмними інтерфейсами через мережу інтернет.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для комфортного використання програмного забезпечення системи планування навчального процесу необхідні такі апаратні та програмні компоненти:

- ОЗУ - не менш 4 Гб;
- операційна система з можливістю взаємодії з програмними інтерфейсами через мережу Інтернет;
- відео адаптер: DirectX 10;
- процесор – не менш 1 ГГц
- мережний адаптер.

- програмне забезпечення для перегляду вебсторінок.

A.3.6 Вимоги до маркування і пакування

Програму для планування навчального процесу можна записати на будь-який носій інформації. На пакування обов'язково повинна бути вказана назва програми: «Програмне забезпечення для планування навчального процесу».

A.4 Вхідні дані до роботи

Вхідними даними під час авторизації є логін користувача та пароль. Вихідними є підтвердження входу та розклад. Після авторизації на вході є роль користувача для визначення прав доступу, на виході є повний розклад для відображення, доступ до функцій додавання, редагування, видалення для «викладача».

Завдання для студентів: вхідними даними є ідентифікатор студента, щоб показати його завдання `studentId`, вихідними список завдань з оцінками. Управління завданнями: вхідні дані – це нове завдання або оцінка студента, вихідними – оновлена інформація про завдання.

Перелік завдань, призначених студентам та візуалізація даних (статистика): на вході є оцінки студентів по предметам на виході – візуальні графіки з успішністю за предметами.

Додаток Б
Текст програми

Б.1 Додавання та видалення файлів з файлової системи

```

const studentController = require('../controllers/studentController');
module.exports = (app) => {
  app.route('/students')
    .get(studentController.getStudents);};
const Task = require('../models/taskModel');
const multer = require('multer');
const path = require('path');
const Student = require('../models/studentModel');
const fs = require('fs').promises;
exports.get_user_by_id = async function(req, res) {
  const { id } = req.params;
  try {
    const user = await User.findById(id);
    if (!user) {
      return res.status(404).json({ message: "Користувача не знайдено" });}
    res.status(200).json({
      id: user._id,
      username: user.username,
      role: user.role});
  } catch (error) {
    console.error('Помилка при отриманні користувача:', error);
    res.status(500).json({ message: "Внутрішня помилка сервера" });});
const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, 'uploads/'); },
  filename: function (req, file, cb) {
    const timestamp = new Date().toISOString().replace(/[:.-]/g, '_');
    const fileType = path.extname(file.originalname);

```

```

const newFileName = `file_${timestamp}${fileType}`;
cb(null, newFileName); });
const upload = multer({ storage: storage });
exports.upload_files = upload.array('files', 5);
exports.update_task_with_files = async function(req, res) {
  try {
    const taskId = req.params.id;
    const username = req.body.username;
    const files = await Promise.all(req.files.map(async (file) => {
      const filePath = file.path;
      const fileType = path.extname(file.originalname).toLowerCase();\
      if (fileType === '.txt' || fileType === '.csv') {
        const data = await fs.readFile(filePath, 'utf8');
        await fs.writeFile(filePath, data, { encoding: 'utf8' }); }
      return {
        filename: file.originalname,
        path: file.path,
        uploadDate: new Date(),
        uploadedBy: username}; }));
    const updatedTask = await Task.findByIdAndUpdate(
      taskId, {
        $push: { files: { $each: files } },
        completed: true },
        { new: true, runValidators: true } );
    exports.delete_file = async function(req, res) {
      try {
        const { taskId, fileId } = req.params;
        console.log(`Спроба видалення файлу. TaskID: ${taskId}, FileID:
        ${fileId}`);
        const task = await Task.findById(taskId);

```

```

    if (!task) {
      console.log(`Завдання не знайдено. TaskID: ${taskId}`);
      return res.status(404).send("Завдання не знайдено"); }

      const fileIndex = task.files.findIndex(file => file._id.toString() ===
fileId);

      if (fileIndex === -1) {
        console.log(`Файл не знайдено в завданні. TaskID: ${taskId},
FileID: ${fileId}`);

        return res.status(404).send("Файл не знайдено");}

      const fileToDelete = task.files[fileIndex];
      const filePath = path.join(__dirname, '..', fileToDelete.path);

      console.log(`Спроба видалення файлу з файлової системи. Шлях:
${filePath}`)

      try {
        await fs.unlink(filePath);

        console.log(`Файл успішно видалено з файлової системи. Шлях:
${filePath}`);

      } catch (error) {
        console.error(`Помилка при видаленні файлу з файлової системи:
${error.message}`)

        task.files.splice(fileIndex, 1);
        if (task.files.length === 0) {
          task.completed = false; }
        await task.save();

        console.log(`Інформацію про файл видалено з бази даних. TaskID:
${taskId}, FileID: ${fileId}`);

        console.log(`Статус завдання: ${task.completed ? 'виконане' :
'невиконане'}`);

        res.status(200).send(task);
      } catch (error) {

```

```

        console.error('Помилка видалення файлу:', error);
        res.status(500).send(`Внутрішня помилка сервера:
        ${error.message}`); } };
```

Б.2 Авторизація

```

const User = require('../models/userModel');
const Student = require('../models/studentModel')
exports.login_user = async function(req, res) {
    const { username, password } = req.body;
    try {
        const user = await User.findOne({ username, password });
        if (!user) {
            return res.status(401).json({ message: "Неправильний логін або
пароль" });}
        res.status(200).json({
            message: "Успішний вхід",
            user: {
                id: user._id,
                username: user.username,
                role: user.role } });
    } catch (error) {
        console.error('Помилка при логіні користувача:', error);
        res.status(500).json({ message: "Внутрішня помилка сервера" }); } }
exports.register_user = async function(req, res) {
    const { id, username, password, role } = req.body;
    try {
        const existingUserById = await User.findById(id);
        if (existingUserById) {
```

```

        return res.status(400).json({ message: "Користувач з таким ID
вже існує" }); }

        const existingStudent = await Student.findOne({ username });
        if (existingStudent) {
            return res.status(400).json({ message: "Студент з таким іменем
вже існує" });}

        const newUser = new User({ _id: id, username, password, role });
        await newUser.save();
        if (role === 'student') {
            const newStudent = new Student({
                _id: id,
                username,
                role: 'student' });
            await newStudent.save(); }
        res.status(201).json({
            message: "Користувача успішно зареєстровано",
            user: {
                id: newUser._id,
                username: newUser.username,
                role: newUser.role} });
    } catch (error) {
        console.error('Помилка при реєстрації користувача:', error);
        res.status(500).json({ message: "Внутрішня помилка сервера" }); } }

```

Б.3 Додавання предметів

```

const mongoose = require('mongoose');
const taskSchema = new mongoose.Schema({
    subject: { type: String, required: true },
    taskDescription: { type: String, required: true },

```

```

deadline: { type: Date, required: true },
studentId: { type: Number },
username: { type: String },
completed: { type: Boolean, default: false },
grades: { type: Map, of: Number },
files: [{
  filename: String,
  path: String,
  uploadDate: Date,
  uploadedBy: String, }],});
const Task = mongoose.model('Task', taskSchema);
module.exports = Task;
const Task = require('../controllers/taskController');
module.exports = function(app) {
  app.route('/tasks')
    .get(Task.list_of_tasks)
    .post(Task.create_a_task);
  app.route('/tasks/:id')
    .put(Task.update_a_task)
    .delete(Task.delete_a_task);
  app.route('/tasks/:id/upload')
    .put(Task.upload_files, Task.update_task_with_files);
  app.route('/tasks/:taskId/files/:fileId')
    .delete(Task.delete_file);});
<div v-if="task.files && task.files.length > 0" class="uploaded-files">
  <h4>Завантажені файли:</h4> <ul>
    <li v-for="(file, fileIndex) in task.files" :key="fileIndex">
      <a :href="getFileUrl(file)" target="_blank" download>{{ file.filename
}}</a>
    <span> — Здав: {{ file.uploadedBy }}</span></li> </ul>

```

```

</div>
<div class="actions">
    <button @click="openEditModal(index)" class="edit-
button">Редагувати</button>
    <button @click="deleteTask(index)" class="delete-
button">Видалити</button>
</div>
<div class="grade-section">
    <button
        v-if="task.completed && task.grade === null"
        @click="openGradeModal(index)"
        class="set-grade-button" >
        Поставити оцінку
    </button>
    <button v-if="task.completed && task.grade !== null"
@click="openEditGradeModal(index)" class="edit-grade-button">Редагувати
оцінку</button>
    <span v-if="task.grade !== null" class="grade-display">Оцінка: {{
task.grade }}</span> </div> </li> </ul>
    <div v-if="showGradeModal !== null" class="modal-overlay"
@click.self="closeGradeModal">
    <div class="modal">
        <h3>Поставити оцінку за "{{
tasks[showGradeModal].taskDescription }}"</h3>
        <input
            type="number"
            v-model.number="tempGrade"
            min="0"
            max="100"
            placeholder="Введіть оцінку" />

```

```

<div class="modal-actions">
    <button @click="submitGrade(showGradeModal)"
:disabled="!isValidGrade" class="submit-grade-button">
        Відправити
    </button>
    <button @click="closeGradeModal" class="cancel-button">
        Відмінити
    </button>
</div> </div></div>

```

Б.4 Метод планування навчання

```

template>
<div class="performance-chart">
    <h2>Статистика по предметам та завданням</h2>
    <div class="charts-container">
        <div class="chart-wrapper">
            <h3>Розподіл предметів по категоріях</h3>
            <div class="chart-container">
                <canvas ref="subjectsChartCanvas"></canvas></div> </div>
        <div class="chart-wrapper">
            <h3>Статус виконання завдань</h3>
            <div class="chart-container">
                <canvas ref="tasksChartCanvas"></canvas> </div> </div>
        <div class="chart-wrapper full-width">
            <h3>Успішність студента</h3>
            <div class="select-container">
                <select v-model="selectedStudentId"
@change="updateStudentPerformance">

```

```

      <option v-for="student in students" :key="student.id"
:value="student.id">
        {{ student.username }}
      </option> </select></div>
<div class="chart-container">
  <canvas ref="studentPerformanceCanvas"></canvas>
</div></div><div>
<h3 id="totalAverage">Загальна середня оцінка: 0.00</h3></div>
<div class="recommendation-section">
  <h3>Рекомендації та відомість оцінок</h3>
  <div class="improvement-suggestions">
    <ul ref="improvementList"></ul></div>
  <div class="grades-report"> <table>
    <thead><tr>
      <th>Предмет</th>
      <th>Оцінка</th> </tr>
    </thead>
    <tbody>
      <tr v-for="(grade, index) in studentGrades" :key="index">
        <td>{{ grade.subject }}</td>
        <td>{{ grade.averageGrade.toFixed(2) }}</td> </tr>
      </tbody> </table></div></div></div> </div>
</template>
<script>
import { ref, onMounted } from 'vue';
import axios from 'axios';
import Chart from 'chart.js/auto';
export default {
  name: 'PerformanceChart',
  setup() {

```

```

const subjectsChartCanvas = ref(null);
const tasksChartCanvas = ref(null);
const studentPerformanceCanvas = ref(null);
const improvementList = ref(null);
const students = ref([]);
const selectedStudentId = ref(null);
const studentGrades = ref([]);
let studentPerformanceChart = null;
const fetchSubjectsData = async () => {
  try {
    const response = await axios.get('http://10.0.0.102:3000/rozklad');
    return response.data;
  } catch (error) {
    console.error('Помилка при отриманні даних про предмети:',
error);return [];});
const fetchTasksData = async () => {
  try {
    const response = await axios.get('http://10.0.0.102:3000/tasks');
    return response.data;
  } catch (error) {
    console.error('Помилка при отриманні даних про завдання:', error);
    return [];});
const processSubjectsData = (data) => {
  const categories = {};
  data.forEach(item => {
    if (!categories[item.category]) {
      categories[item.category] = 0;}
    categories[item.category]++;});
  return categories;});
const processTasksData = (data) => {

```

```

const status = { completed: 0, incomplete: 0 };
data.forEach(task => {
  if (task.completed) {
    status.completed++;} else {
    status.incomplete++;});
return status;});

const processStudentPerformance = (data, studentId) => {
  const studentTasks = data.filter(task => task.studentId === studentId);
  const subjectGrades = {};
  studentTasks.forEach(task => {
    if (!subjectGrades[task.subject]) {
      subjectGrades[task.subject] = [];}
    const grade = task.grades && typeof task.grades[task.studentId] ===
'number' ? task.grades[task.studentId] : 0;
    subjectGrades[task.subject].push(grade);});
  const averageGrades = Object.entries(subjectGrades).map(([subject,
grades]) => ({
    subject,
    averageGrade: grades.length > 0 ? grades.reduce((sum, grade) => sum
+ grade, 0) / grades.length : 0 }));
  studentGrades.value = averageGrades;
  const totalAverage = averageGrades.reduce((sum, item) => sum +
item.averageGrade, 0) / averageGrades.length;
  return { averageGrades, totalAverage }; });

const createSubjectsChart = (categories) => {
  const ctx = subjectsChartCanvas.value.getContext('2d');
  new Chart(ctx, {
    type: 'bar',
    data: {
      labels: Object.keys(categories),

```

```

    datasets: [{
      label: 'Кількість предметів',
      data: Object.values(categories),
      backgroundColor: 'rgba(75, 192, 192, 0.6)',
      borderColor: 'rgba(75, 192, 192, 1)',
      borderWidth: 1 }]],
    options: {
      responsive: true,
      maintainAspectRatio: false,
      scales: {
        y: {
          beginAtZero: true,
          title: {
            display: true,
            text: 'Кількість' }},
        x: {title: {
          display: true,
          text: 'Категорії предметів' } } } } }));
const createTasksChart = (status) => {
  const ctx = tasksChartCanvas.value.getContext('2d');
  new Chart(ctx, {
    type: 'pie',
    data: {
      labels: ['Здано', 'Не здано'],
      datasets: [{
        data: [status.completed, status.incomplete],
        backgroundColor: ['rgba(75, 192, 192, 0.8)', 'rgba(255, 99, 132,
0.8)'],
        borderColor: ['rgba(75, 192, 192, 1)', 'rgba(255, 99, 132, 1)'], }]],
    options: {

```

```

    responsive: true,
    maintainAspectRatio: false,
    plugins: {
      legend: {position: 'bottom', },
      tooltip: {
        callbacks: {
          label: function(context) {
            const label = context.label || "";
            const value = context.raw;
            const total = context.dataset.data.reduce((acc, data) => acc +
data, 0);

            const percentage = ((value / total) * 100).toFixed(1);
            return `${label}: ${value} (${percentage}%)`;}} } } } }));};
const createStudentPerformanceChart = (performanceData) => {
  const ctx = studentPerformanceCanvas.value.getContext('2d');
  if (studentPerformanceChart) {
    studentPerformanceChart.destroy();}
  studentPerformanceChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: performanceData.map(item => item.subject),
      datasets: [{
        label: 'Середня оцінка',
        data: performanceData.map(item => item.averageGrade),
        fill: false,
        borderColor: 'rgb(75, 192, 192)',
        tension: 0.1} ]},
    options: {
      responsive: true,
      maintainAspectRatio: false,

```

```

scales: { y: {
  beginAtZero: true,
  max: 5,
  title: {
    display: true,
    text: 'Середня оцінка' } },
x: {title: {
  display: true,
  text: 'Предмети' } } } }));});

const displayImprovementSuggestions = (performanceData) =
const suggestions = performanceData.filter(item => item.averageGrade <
3.5);

const improvementListElement = improvementList.value;
improvementListElement.innerHTML = "";
if (suggestions.length > 0) {
  suggestions.forEach(item => {
    const li = document.createElement('li');
    li.textContent = `Підтягніть: ${item.subject} (Середня оцінка:
${item.averageGrade.toFixed(2)})`;
    improvementListElement.appendChild(li);});} else {
  improvementListElement.textContent = 'Всі предмети на рівні.'; } };
const displayTotalAverage = (totalAverage) => {
  const averageDisplay = document.getElementById('totalAverage');
  averageDisplay.textContent = `Загальна середня оцінка:
${totalAverage.toFixed(2)}`; };
const updateStudentPerformance = async () => {
  const tasksData = await fetchTasksData();
  const { averageGrades, totalAverage } =
processStudentPerformance(tasksData, selectedStudentId.value);
createStudentPerformanceChart(averageGrades);

```

```

        displayImprovementSuggestions(averageGrades);
        displayTotalAverage(totalAverage);});
    onMounted(async () => {
const subjectsData = await fetchSubjectsData();
const tasksData = await fetchTasksData();
const processedSubjectsData = processSubjectsData(subjectsData);
const processedTasksData = processTasksData(tasksData);
const uniqueStudentsMap = new Map();
tasksData.forEach(task => {
    if (!uniqueStudentsMap.has(task.username)) {
        uniqueStudentsMap.set(task.username, task.studentId);});
students.value = Array.from(uniqueStudentsMap, ([username, id]) => ({
    id,
    username}));
if (students.value.length > 0) {
    selectedStudentId.value = students.value[0].id;}
createSubjectsChart(processedSubjectsData);
createTasksChart(processedTasksData);
await updateStudentPerformance();});
    return {
        subjectsChartCanvas,
        tasksChartCanvas,
        studentPerformanceCanvas,
        improvementList,
        students,
        selectedStudentId,
        studentGrades,
        updateStudentPerformance }));
</script>
<style scoped>

```

```
@import url('https://fonts.googleapis.com/css2?family=Fredoka:wght@300..700&family=Nunito:ital,wght@0,200..1000;1,200..1000&display=swap');  
.grades-report {
```

Додаток В
Слайди презентації

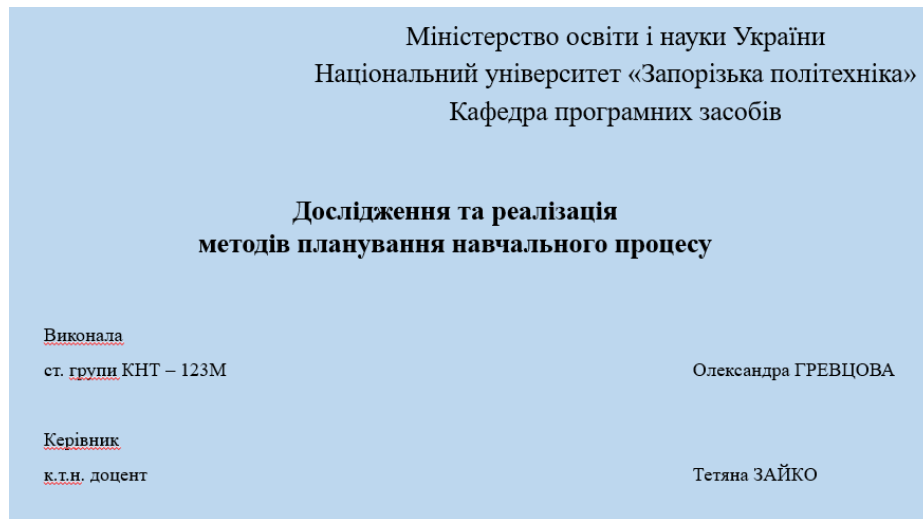


Рисунок В.1 – Титульний слайд

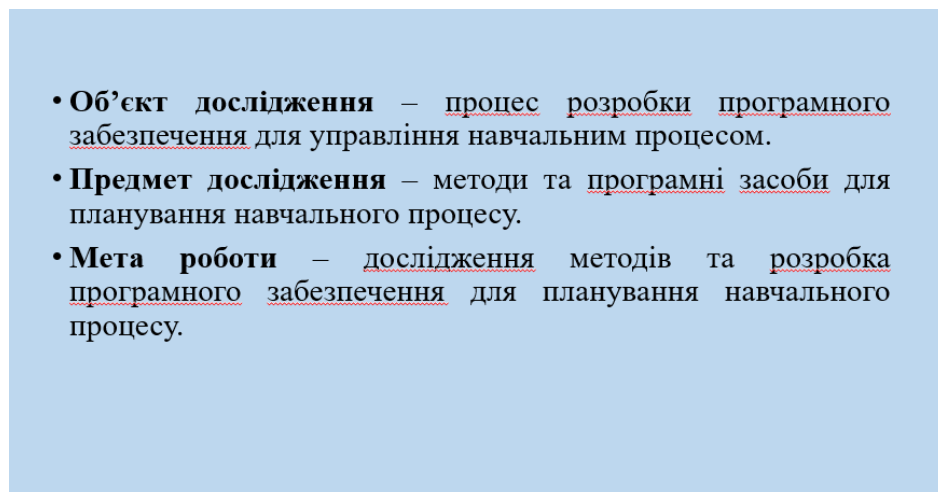


Рисунок В.2 – Об'єкт роботи, предмет та мета

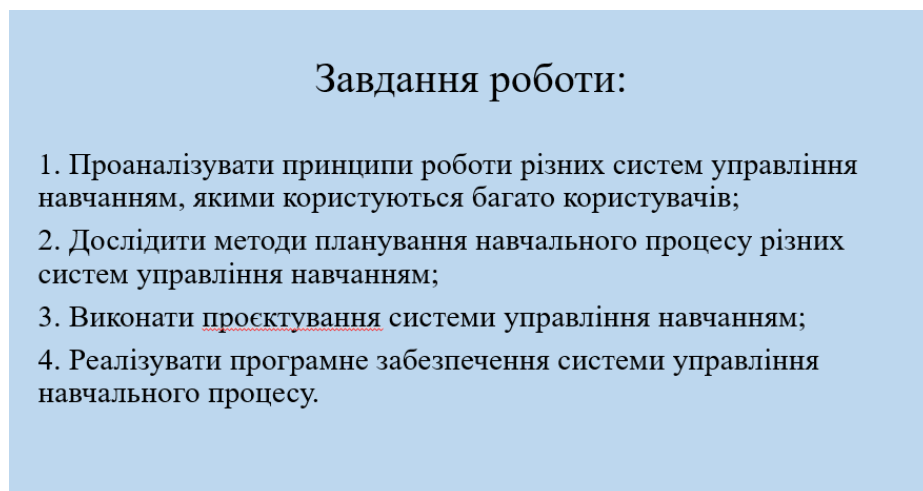


Рисунок В.3 – Завдання роботи

Вибір мови програмування

Параметри порівняння мов програмування	Мови програмування	
	Vue.js	React.js
Простота навчання	+	-
Архітектура	+	+
Гнучкість	+	+
Стан управління	+	+
Шаблони	+	-
Продуктивність	+	+
Екосистема	-	+
Середовище	+	+
Підтримка мобільних додатків	+	+

Рисунок В.4 – Вибір мови програмування

Вибір середовища програмування

Характеристика	VS Code	Sublime Text
Кросплатформеність	Windows, macOS, Linux	Windows, macOS, Linux
Швидкодія	Дуже висока	Висока
Підтримка плагінів/розширень	Багато, через Marketplace	Обмежена кількість плагінів
Інтелектуальне автодоповнення	IntelliSense (сильне)	Базове
Вбудований термінал	Так	Ні
Підтримка налагодження	Так, через плагіни	Ні

Рисунок В.5 – Вибір середовища програмування

Структура додатку

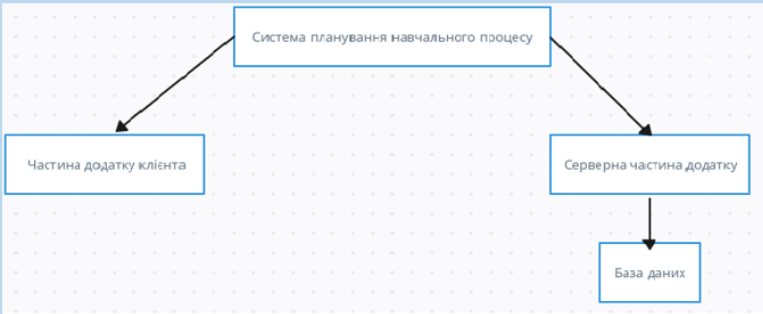


Рисунок В.6 – Структура додатку

Головна сторінка системи планування навчального процесу

Система планування навчального процесу

Розклад занять

Додати заняття

№	ЧАС	ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР	П'ЯТНИЦЯ
1	12:14 - 12:14	ПБП Іванова В.І.	Геометрія Кравцова В.А.	Алгебра Майорова Т.А.	Штучний інтелект Ковальчук Д.М.	
2	12:17 - 13:17	Розробка Петров М.Р.	ПБП Іванова В.І.		Менеджмент Носова О.Г.	Психологія Сидорова С.С.
3	13:31 - 14:31		Література Шевченко К.М.	Українська мова Колосин Д.О.		Фізика Михайлюк Ю.Р.
4	15:01 - 16:01				Операційні системи Сидоров В.Л.	

Рисунок В.7 – Головна сторінка додатку

Панель критеріїв програми для користувачів

Система планування навчального процесу

Розклад занять

Мій завдання
Розклад
Графіки успішності
Вийти

	ПОНЕДІЛОК	ВІВТОРОК	СЕРЕДА	ЧЕТВЕР	П'ЯТНИЦЯ
1	ПБП Іванова В.І.	Геометрія Кравцова В.А.	Алгебра Майорова Т.А.	Штучний інтелект Ковальчук Д.М.	
2	Розробка Петров М.Р.	ПБП Іванова В.І.		Менеджмент Носова О.Г.	Психологія Сидорова С.С.
3		Література Шевченко К.М.	Українська мова Колосин Д.О.		Фізика Михайлюк Ю.Р.
4				Операційні системи Сидоров В.Л.	

Рисунок В.8 – Панель критеріїв програми

Вікно додавання завдання

Додати нове заняття

Номер пари:

День:

Час початку:

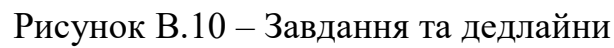
Час закінчення:

Предмет:

Викладач:

Додати Скасувати

Рисунок В.9 – Вікно додавання завдання



Завдання для студента



Завантаження файлу

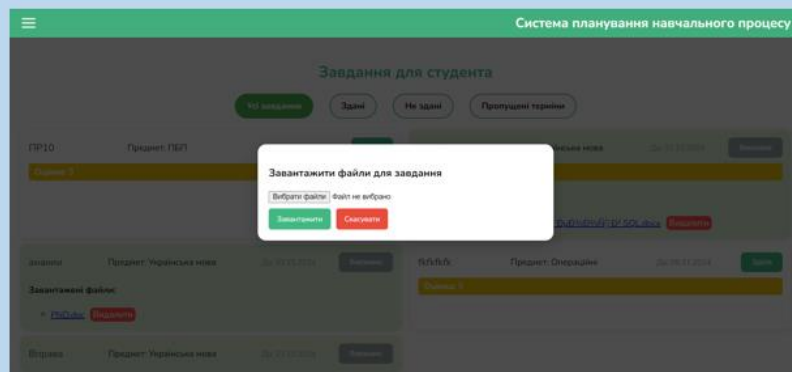


Рисунок В.12 – Слайд завантаження файлу

Повідомлення для вчителя/студента

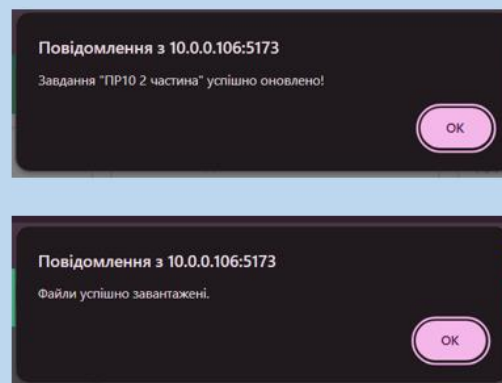


Рисунок В.13 – Слайд повідомлення

Статистика по предметам та завданням

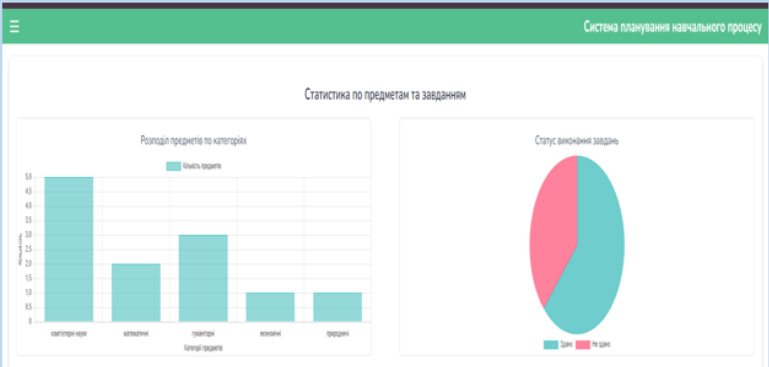


Рисунок В.14 – Статистика по предметам та завданням

Графік успішності

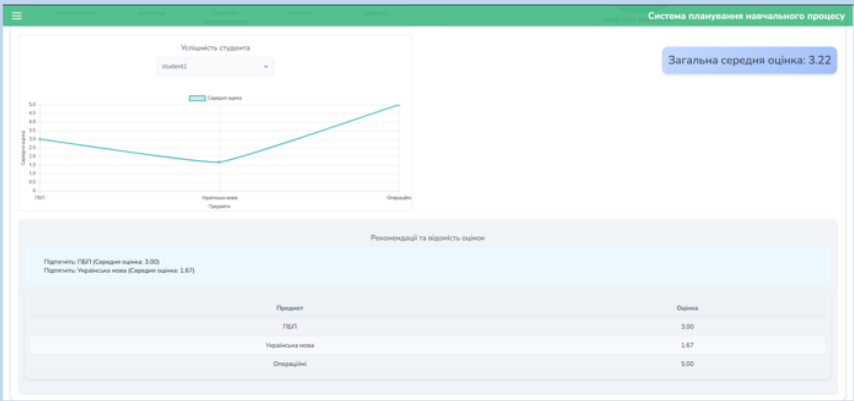


Рисунок В.15 – Слайд графіків успішності

Висновки

1. В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено методи планування навчальним процесом та створено систему управління навчальним процесом.
2. Запропоновано метод планування навчальним процесом, який базується на аналізі даних про студентів, їх успішності та виявлення ключових показників для оптимізації процесу навчання. На основі даних про успішність по кожному предмету всередині категорії обчислюється середня оцінка з усіх предметів, що належать до цієї категорії. Алгоритм роботи системи виглядає так: збираються оцінки з усіх предметів у категорії, обчислюється середнє арифметичне оцінювання для кожного студента за кожною категорією предметів. Якщо середня оцінка категорії нижче порога, наприклад, 3,5, це вважається індикатором того, що студент потребує додаткової уваги з цієї групи предметів. Цей кордон може бути гнучким залежно від вимог освітнього закладу.

Рисунок В.16 – Висновки

Висновки

4. Результати аналізу візуалізовані та представлені користувачеві у зручному форматі. Даний метод зображує результат у вигляді графіків, який наочно показує поточний рівень успішності та області, які вимагають поліпшення. Дієвість методу можна перевірити за допомогою емпіричного аналізу результатів студентів, але для цього потрібен деякий час.
5. Виконано тестування розробленого програмного забезпечення. Після проведення тестування встановлено, що системи планування навчальним процесом з впровадженням методом планування навчального процесу повністю відповідає вимогам технічного завдання, та за рахунок впровадженого методу має лідуючі показники метрик ефективності серед представлених аналогів.

Рисунок В.17 - Висновки