

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

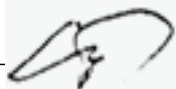
Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

—  — проф. Приходько С.Б.

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

**на тему: “Розробка вебзастосунку для автоматизації складання розкладу
занять для Березнегуватського ліцею”**

Виконав: студент групи 4151

 Швец С.О.

(підпис, ПІБ)

Керівник роботи:

Доцент, к.т.н., доцент

(посада, науковий ступень вчене звання)

—  — Пухалевич А.В.

(підпис, ПІБ)

Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»



Гарант освітньої програми
_____ доц. Макарова Л.М.

(підпис)

« 08 » ____ 04 ____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту Швецю Сергію Олександровичу

(Прізвище, ім'я, по батькові)

1. Тема роботи: “Розробка вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею”

Керівник роботи Пухалевич А.В.

Затверджені наказом ректора №153 від 08.04.2025 р.

2. Термін подання роботи: 02.06.2025 р.

3. Вихідні дані по роботі:

4. Перелік питань, що належать до розробки (найменування розділів)

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проєкт програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1. Титульний слайд; 2. Мета та завдання кваліфікаційної роботи; 3. Аналіз вимог до програмного забезпечення; 4. Архітектура програмного забезпечення; 5. Контекстна діаграма; 6. Діаграма варіантів використання; 7. Діаграма “сутність - зв'язок”; 8. Логічна модель даних; 9. Діаграма класів; 10. Фізична модель даних; 11. Результат розробки; 12. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 08.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	31.03.2025	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	07.04.2025	ПП*
3.	Аналіз вимог до ПЗ	14.04.2025	ПП*
4.	Розробка архітектури ПЗ	21.04.2025	
5.	Розробка проєкту ПЗ	05.05.2025	
6.	Реалізація та тестування ПЗ	12.05.2025	
7.	Підготовка розділу Результати розробки ПЗ	16.05.2025	
8.	Підготовка розділу з охорони праці	19.05.2025	ПП*
9.	Підготовка розділу ВИСНОВКИ	23.05.2025	
10.	Оформлення списку використаних джерел та додатків	26.05.2025	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	02.06.2025	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку
12.	Підготовка презентації та доповіді	06.06.2025	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	09.06.2025	
14.	Подання на кафедру ПЗАС електронних версій наступних 16.06.2025 документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	16.06.2025	

- за результатами переддипломної практики (ПП), яка була з 03.02.2025 до 23.03.2025 р.

Студент

Швець
(підпис)

Швець

(підпис)

Керівник роботи

Пухалевич
(підпис)

Пухалевич

(підпис)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці вебзастосунку для автоматизації складання розкладу в диспетчерській Березнегуватського ліцею. Метою розробки вебзастосунку є оптимізація обробки даних, зменшення часових витрат робочого персоналу та полегшення доступу до інформації для учнів даного навчального закладу.

Був проведений аналіз практичної задачі інженерії програмного забезпечення, а саме задачі розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею, розробку вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею та результат розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.

Робота виконана на 96 сторінках друкованого тексту та містить 45 рисунки, 17 таблиць, список використаних джерел з 15 найменувань та 5 додатків.

Кваліфікаційна робота виконана українською мовою.

ABSTRACT

The qualification work is devoted to the development of a web application to automate the scheduling in the dispatch room of the Berezhneuvatsky Lyceum. The purpose of developing the web application is to optimize data processing, reduce the time spent by the staff, and facilitate access to information for students of this educational institution.

An analysis of a practical software engineering problem was conducted, namely the problem of developing a web application to automate the compilation of class schedules for the Berezhneuvatsky Lyceum, the development of a web application to automate the compilation of class schedules for the Berezhneuvatsky Lyceum, and the result of developing a web application to automate the compilation of class schedules for the Berezhneuvatsky Lyceum.

The work is completed on 96 pages of printed text and contains 45 figures, 17 tables, a list of sources used with 15 names and 5 appendices.

The qualification work is done in Ukrainian.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ	11
1.1 Аналіз практичної задачі інженерії програмного забезпечення розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	11
1.2 Аналіз існуючого програмного забезпечення для автоматизації складання розкладу занять	12
1.3 Постановка задачі на розробку вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	15
2 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ.....	18
2.1 Аналіз вимог до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	18
2.1.1 Побудова контекстної діаграми до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	18
2.1.2 Розробка словника даних контекстної діаграми до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	19
2.1.3 Декомпозиція процесів до мініспецифікацій для вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	

2.1.4 Побудова таблиці відповідності потоків даних на діаграмах всіх рівнів до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	21
2.1.5 Розробка мініспецифікацій до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	22
2.1.6 Розробка варіантів використання до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	24
2.2 Розробка архітектури до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	26
2.2.1 Обґрунтування вибору архітектурного стилю до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	26
2.2.2 Архітектура до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	27
2.3 Розробка проєкту вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	29
2.3.1 Розробка концептуальної моделі бази даних вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	29
2.3.2 Побудова діаграми "сутність-зв'язок" вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	32
2.3.3 Розробка графічного інтерфейсу вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	33
2.3.4 Розробка логічної моделі даних вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	35
2.3.5 Розробка діаграми класів вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	36
2.3.6 Специфікація основних класів вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	36

2.3.7 Розробка фізичної моделі даних вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	38
2.4 Реалізація проєкту вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	40
2.4.1 Використані СКБД та інші програмні засоби до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	40
2.4.2 Фрагменти коду до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	42
2.5 Тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	46
2.5.1 Обґрунтування вибору тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	46
2.5.2 Тестування чорного ящика до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	47
3 РЕЗУЛЬТАТ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ	51
3.1 Загальний опис результатів розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	51
3.2 Відповідність результатів розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею вимогам технічного завдання	52
3.3 Кількісна та якісна характеристика до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.....	52
3.4 Результати тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею	53
4 ОХОРОНА ПРАЦІ	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	57

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ	59
ДОДАТОК Б – ТЕКСТ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ	63
ДОДАТОК В – ОПИС ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ	84
ДОДАТОК Г – ІНСТРУКЦІЯ ПРОГРАМІСТУ, ОПЕРАТОРУ Й КОРИСТУВАЧЕВІ ДО ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ	86
ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ	95

ВСТУП

В даній кваліфікаційній роботі розглядається практична задача інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов: розробка вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.

У сучасних навчальних закладах важливо мати доступний електронний розклад занять для учнів та працівників закладу, а також вести облік викладачів і предметів, які вони викладають. Відсутність відповідного застосунку ускладнює процес отримання інформації щодо розкладу та контактних даних викладачів. Це може призвести до неефективної комунікації між учнями та викладачами, а також до труднощів в організації навчального процесу. Подібні проблеми характерні для багатьох навчальних закладів і потребують розробки програмного забезпечення для автоматизації процесу управління розкладом занять.

Розробка відповідного вебзастосунку є доцільною, оскільки наявні аналоги, не завжди враховують специфіку навчального процесу, зокрема автоматизоване формування розкладу, зміну занять, інші функції. Вебзастосунок з базою даних забезпечить ефективне управління навчальним процесом, дозволяючи зберігати, додавати та видаляти інформацію про розклад, предмети та викладачів у зручному форматі.

Мета кваліфікаційної роботи – це розробка вебзастосуноку для автоматизації складання розкладу занять для Березнегуватського ліцею.

Для досягнення мети потрібно виконати ряд завдань, а саме:

- Проаналізувати практичну задачу інженерії програмного забезпечення вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею;
- Проаналізувати існуюче програмне забезпечення для автоматизації складання розкладу занять;

- Виконати постановку задачі на розробку вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею;
- Провести аналіз вимог до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею;
- Вибрати архітектуру до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею;
- Розробити проєкт вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею;
- Реалізувати вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею;
- Виконати тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.

Об'єкт, який розглядається у кваліфікаційній роботі - це процес розробки програмного забезпечення для автоматизації ведення розкладу занять у навчальному закладі.

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

1.1 Аналіз практичної задачі інженерії програмного забезпечення розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

В даній кваліфікаційній роботі в якості практичної задачі інженерії програмного забезпечення розглядається розробка вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.

Складання розкладу занять охоплює процес планування та організації навчального процесу для здобувачів освіти. Основна мета складання розкладу занять полягає у забезпеченні оптимального використання часу та ресурсів навчального закладу, а також в забезпеченні якісного навчання для учнів. Крім того, складання розкладу занять є важливим елементом управління навчальним процесом та може бути використане для покращення якості навчання.

Наразі, Березнегуватський ліцей не має даного програмного забезпечення, тому їм доводиться вручну вести весь розклад навчання, що є дуже повільним процесом. Тому автоматизація цих дій дозволить значно зменшити ці всі недоліки, а інформація про викладачів та предмети буде завжди доступною в електронному вигляді, що допоможе швидше і точніше виконувати цю роботу.

Наразі для вирішення цієї задачі розробляється спеціалізоване програмне забезпечення. Для цього можуть використовуватись такі мови програмування як Java та HTML. Також використовується база даних така як MariaDB.

Для складання розкладу необхідно мати дані про викладачів:

- ПІБ викладача;

- електрона пошта (щоб можна було написати викладачеві, у разі потреби);
- номер телефону.

Розклад занять містить наступні відомості:

- клас;
- день тижня;
- номер уроку;
- дисципліна;
- викладач.

Для складання розкладу занять потрібно врахувати кількість годин для кожного з предметів.

1.2 Аналіз існуючого програмного забезпечення для автоматизації складання розкладу занять

При виконанні цієї роботи важливо розглянути уже існуючі програмні рішення для автоматизації складання розкладу занять. Нижче наведено аналіз декількох таких програм.

FET — це програма з відкритим вихідним кодом для автоматичного складання розкладів занять у навчальних закладах. Вона написана на C++ з використанням бібліотек Qt і розповсюджується безкоштовно під ліцензією GNU AGPL v3. FET враховує багато параметрів навчального процесу: студентів, викладачів, дисципліни, аудиторії, будівлі, заняття, поділ на потоки, групи та підгрупи, а також різноманітні обмеження. Після введення необхідних даних програма аналізує їх і пропонує одне або декілька можливих розкладів. Інтерфейс FET перекладено на багато мов, і вона підтримує імпорт/експорт у форматах CSV, HTML та XML. Проте, через універсальність програми, її налаштування та адаптація можуть вимагати значних зусиль [1].

Скріншот додатку FET представлена на рисунку 1.1.

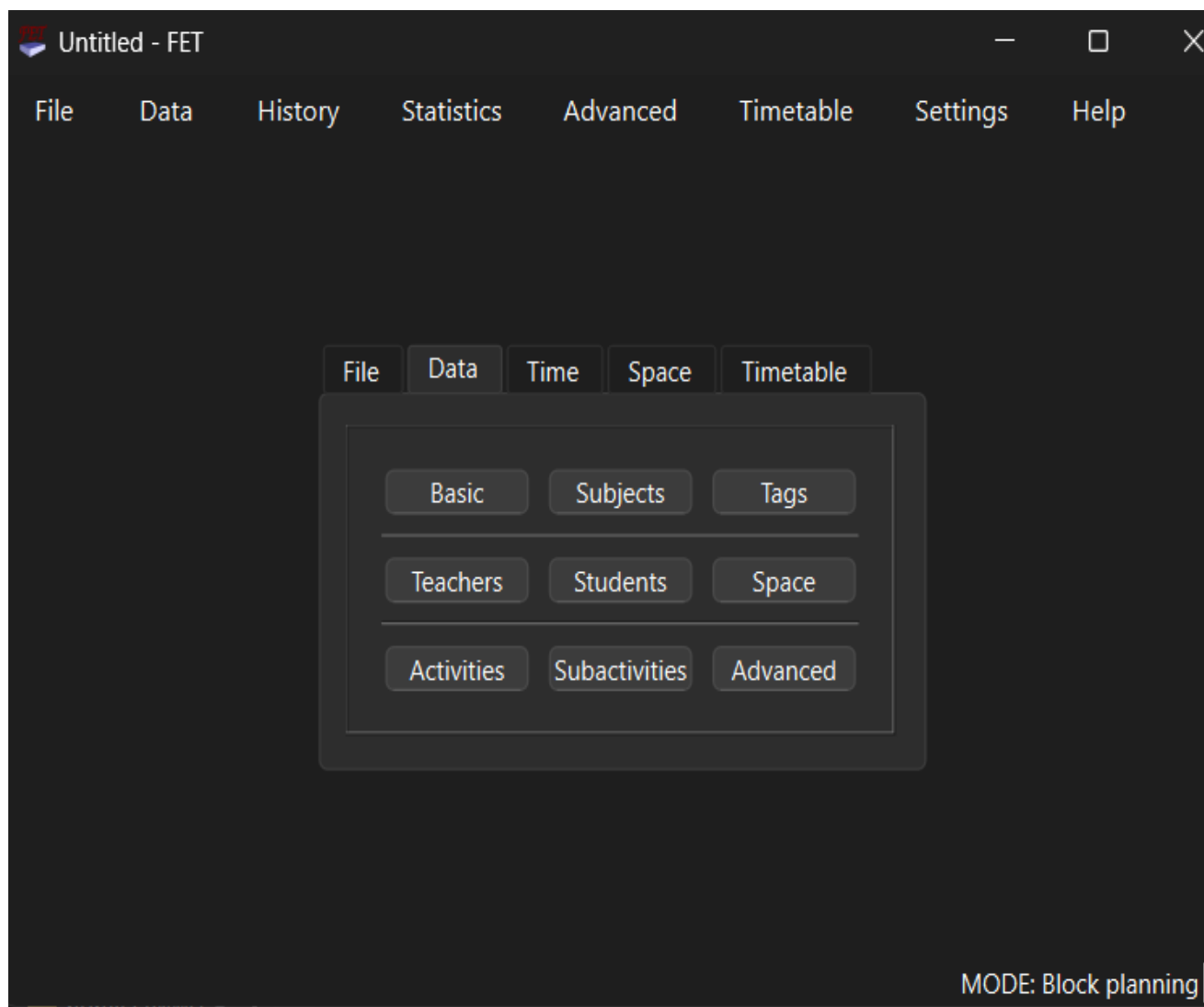


Рисунок 1.1 – Додаток FET

Smart School — це система автоматизації освітнього процесу, призначена для закладів загальної середньої освіти, професійно-технічних навчальних закладів та вищих навчальних закладів I–II рівнів акредитації. Вона поєднує в собі модулі для складання розкладу, електронного журналу, автоматичної звітності, особистого кабінету учня (студента), тестування та інші функції. Перевагами Smart School є багатофункціональність та простота використання. Однак, для впровадження цієї системи може знадобитися значний час на навчання персоналу та адаптацію до існуючих процесів закладу [2].

Скріншот вебзастосунку Smart School представлена на рисунку 1.2.

Основні можливості Smart School

Для адміністрації	Для викладачів	Для учнів (студентів) та батьків
Формування розкладу навчальних занять	Доступ до роботи з журналами	Актуальна інформація про оцінки
Доступ до журналів успішності всіх класів (академічних груп)	Зв'язок з батьками	Інформація про відвідування занять
Доступ до всіх типів звітності	Розміщення навчальних матеріалів	Розклад занять та зміни в ньому
Керування списками учнів, батьків та викладачів	Створення власних навчальних планів	Домашні завдання
Тарифікація викладачів	Класне керівництво	Графіки успішності
Розподіл робочого часу викладачів	Створення онлайн-тестувань	Зв'язок з вчителями
Дізнатися більше	Дізнатися більше	Дізнатися більше

Рисунок 1.2 – Вебзастосунок Smart School

Незважаючи на наявні аналоги, все ж потрібно розробити власний вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею і на це є ряд причин:

Адаптація: більшість існуючих програм орієнтовані на універсальне використання і не враховують індивідуальні вимоги. Власний вебзастосунок дозволить точно налаштувати функціональність під потреби закладу.

Простіший інтерфейс: більшість програм мають дуже багато функцій, тому звичайному користувачу буде важко відразу працювати в таких програмах. Власний вебзастосунок дозволить зробити інтерфейс простішим для розуміння та необхідним мінімумом функцій.

1.3 Постановка задачі на розробку вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Виходячи з проведеного аналізу сформулюємо постановку задачі на розробку вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею.

Потрібно створити вебзастосунок для перегляду розкладу занять та іншої інформації для учнів або викладачів. Вебзастосунок повинен мати зрозумілий та гарний інтерфейс. У базі даних вебзастосунку повинна зберігатися наступна інформація:

Про розклад:

- номер розкладу;
- клас;
- день тижня;
- номер уроку;
- назва дисципліни;
- викладач.

Про викладачів:

- номер викладача;
- прізвище, ім'я, по батькові викладача;
- електрона пошта;
- номер телефону;

Про дисципліни:

- номер дисципліни;
- назва дисципліни.

З даним застосунком повинні мати можливість працювати такі групи користувачів:

- диспетчер;
- викладачі;
- учні.

Вебзастосунок повинен забезпечити можливість виконання наступних функцій:

1) Редагування розкладу

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження редагування.

2) Додавання розкладу

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження додавання.

3) Видалення розкладу

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження видалення.

4) Формування розкладу

Вхідні дані: Натиснути на кнопку «Розклад».

Вихідні дані: Розклад занять (День тижня, номер уроку, клас, назва дисципліни, викладач).

5) Генерування розкладу занять

Вхідні дані: Натиснути на кнопку «Створити розклад занять».

Вихідні дані: Розклад занять (День тижня, номер уроку, клас, назва дисципліни, викладач).

6) Формування розкладу занять для викладачів

Вхідні дані: Натиснути на кнопку «Розклад для викладачів».

Вихідні дані: Розклад занять для викладача (День тижня, номер уроку, клас, назва дисципліни).

При роботі з вебзастосунком **диспетчер** повинен мати можливість вирішувати такі завдання:

- 1) Створити розклад;
- 2) Переглянути інформацію про розклад;
- 3) Редагування, видалення або додавання інформації про розклад;
- 4) Переглянути інформацію про викладачів;
- 5) Редагування, видалення або додавання інформації про викладачів;

- 6) Переглянути інформацію про дисципліни;
- 7) Редагування, видалення або додавання інформації про дисципліни;

При роботі з вебзастосунком **викладач** повинен мати можливість вирішувати такі завдання:

- 1) Переглянути інформацію про розклад викладача;

Учень повинен мати можливість вирішувати такі завдання:

- 1) Переглянути інформацію про розклад класу;
- 2) Переглянути інформацію про викладачів.

2 РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

2.1 Аналіз вимог до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

2.1.1 Побудова контекстної діаграми до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Контекстна діаграма - це вид діаграми в сфері системного аналізу та проектування, яка використовується для відображення зв'язків між системою та її зовнішнім середовищем. Ця діаграма допомагає зрозуміти, як система взаємодіє з іншими сутностями, без деталей її внутрішньої структури.

Контекстна діаграма вебзастосунку представлена на рисунку 2.1.



Рисунок 2.1 – Контекстна діаграма вебзастосунку

2.1.2 Розробка словника даних контекстної діаграми до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Словник даних контекстної діаграми містить в собі назви потоків та їх складові. Словник даних контекстної діаграми представлений на таблиці 2.1.

Таблиця 2.1 - Словник даних контекстної діаграми

Найменування потоку	Складові потоку
1	2
Інформація про розклад	-Запит на редагування розкладу(день тижня, номер уроку, клас, назва дисципліни, викладач) -Запит на додавання розкладу(день тижня, номер уроку, клас, назва дисципліни, викладач) -Запит на видалення розкладу (день тижня, номер уроку, клас, назва дисципліни, викладач)
Підтвердження обробки розкладу	- Підтвердження редагування (Результат редагування) -Підтвердження додавання (Результат додавання) -Підтвердження видалення (Результат видалення)
Розклад занять	Розклад занять (день тижня, номер уроку, клас, назва дисципліни, викладач)
Запит на перегляд розкладу	Натиснути кнопку «Розклад»
Запит на перегляд розкладу для викладачів	Натиснути кнопку «Розклад для викладачів»

Кінець таблиці 2.1

1	2
Запит на генерацію розкладу занять	Натиснути на кнопку «Створити розклад занять»
Згенерований розклад занять	Новий згенерований розклад занять (день тижня, номер уроку, клас, назва дисципліни, викладач)

2.1.3 Декомпозиція процесів до мініспецифікацій для вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Декомпозиція процесів до мініспецифікацій представлена на рисунках 2.2-2.3.



Рисунок 2.2 – Діаграма потоків даних 1-го рівня



Рисунок 2.3 – Декомпозиція процесу 1 “Обробити інформацію про розклад”

2.1.4 Побудова таблиці відповідності потоків даних на діаграмах всіх рівнів до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Відповідності потоків на діаграмах всіх рівнів представлені на таблиці 2.2.

Таблиця 2.2 - Таблиця відповідності потоків на діаграмах всіх рівнів

Потоки 0-го рівня	Потоки 1-го рівня	Потоки 2-го рівня	Атрибути потоку
1	2	3	4
Інформація про розклад	Інформація про розклад	Запит на редагування розкладу	день тижня, номер уроку, клас, назва дисципліни, викладач
		Запит на додавання розкладу	день тижня, номер уроку, клас, назва дисципліни, викладач
		Запит на видалення розкладу	день тижня, номер уроку, клас, назва дисципліни, викладач

Кінець таблиці 2.2

1	2	3	4
Підтвердження обробки розкладу	Підтвердження обробки розкладу	Підтвердження редагування	Результат редагування
		Підтвердження додавання	Результат додавання
		Підтвердження видалення	Результат видалення
Запит на перегляд розкладу	Запит на перегляд розкладу		Натискання на кнопку «Розклад»
Запит на перегляд розкладу для викладачів	Запит на перегляд розкладу для викладачів		Натиснути кнопку «Розклад для викладачів»
Запит на генерацію розкладу занять	Запит на генерацію розкладу занять		Натиснути на кнопку «Створити розклад занять»

2.1.5 Розробка мініспецифікацій до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Редагування розкладу

Призначення: редагування інформації про розклад в базі даних.

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження редагування.

Умови виконання: Занесення даних правильного формату у форму і натискання кнопки керування “Зберегти”.

Процедура обробки: При введенні даних правильного формату відбувається заміна старих даних на нові в таблиці «Розклад» бази даних.

Додавання розкладу

Призначення: додавання інформації про розклад в базу даних.

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження додавання.

Умови виконання: Занесення даних правильного формату у форму і натискання кнопки керування “Зберегти”.

Процедура обробки: При введенні даних правильного формату відбувається присвоєння унікального номеру запису й занесення кортежу в таблицю «Розклад» бази даних.

Видалення розкладу

Призначення: видалення інформації про розклад з бази даних.

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження видалення.

Умови виконання: Занесення даних правильного формату у форму і натискання кнопки керування “Видалити”.

Процедура обробки: При введенні даних правильного формату відбувається видалення розкладу в таблиці «Розклад» бази даних.

Формування розкладу

Призначення: сформувати розклад для перегляду.

Вхідні дані: Натиснути на кнопку «Розклад».

Вихідні дані: Розклад занять (День тижня, номер уроку, клас, назва дисципліни, викладач).

Умови виконання: Натискання кнопки керування “Розклад”.

Процедура обробки: При натисканні на кнопку відкривається розклад занять .

Генерування розкладу занять

Призначення: генерувати новий розклад занять.

Вхідні дані: Натиснути на кнопку «Створити розклад занять».

Вихідні дані: Розклад занять (День тижня, номер уроку, клас, назва дисципліни, викладач).

Умови виконання: Натискання кнопки керування “ Створити розклад занять ”.

Процедура обробки: При натисканні на кнопку генерується новий розклад занять .

Формування розкладу занять для викладачів

Призначення: сформувати розклад для викладача.

Вхідні дані: Натиснути на кнопку «Розклад для викладачів».

Вихідні дані: Розклад занять для викладача (День тижня, номер уроку, клас, назва дисципліни).

Умови виконання: Натискання кнопки керування “ Розклад для викладачів”.

Процедура обробки: При натисканні на кнопку відкривається розклад занять для викладача .

2.1.6 Розробка варіантів використання до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Короткий опис акторів представлено в таблиці 2.3.

Таблиця 2.3 – Виявлення акторів

Актор	Короткий опис
1	2
Диспетчер	Займається контролем розкладу, має доступ до бази даних та всіх її таблиць.

Кінець таблиці 2.3

1	2
Учень	Моє доступ до розкладу занять.
Викладач	Має доступ до розкладу занять для викладачів.

Виявлені варіанти використання представлені у таблиці 2.4.

Таблиця 2.4 – Виявлення варіантів використання

Код	Основний актор	Назва	Формулювання
1	2	3	4
D1	Диспетчер	Перегляд розкладу занять	Цей варіант використання дозволяє диспетчеру переглянути розклад занять.
D2	Диспетчер	Додавання нового розкладу занять	Цей варіант використання дозволяє диспетчеру додавати новий розклад занять до бази.
D3	Диспетчер	Редагування розкладу занять	Цей варіант використання дозволяє диспетчеру редагувати розклад занять у базі.
D4	Диспетчер	Видалення розкладу занять	Цей варіант використання дозволяє диспетчеру видаляти розклад занять із бази.
D5	Диспетчер	Генерування нового розкладу занять	Цей варіант використання дозволяє диспетчеру генерувати новий розклад занять.
U1	Учень	Перегляд розкладу занять	Цей варіант використання дозволяє учню переглянути розклад занять.

Кінець таблиці 2.4

1	2	3	4
V1	Викладач	Перегляд розкладу занять для викладачів	Цей варіант використання дозволяє викладачу переглянути розклад занять для викладачів.

Усі варіанти використання представлені на рисунку 2.4.

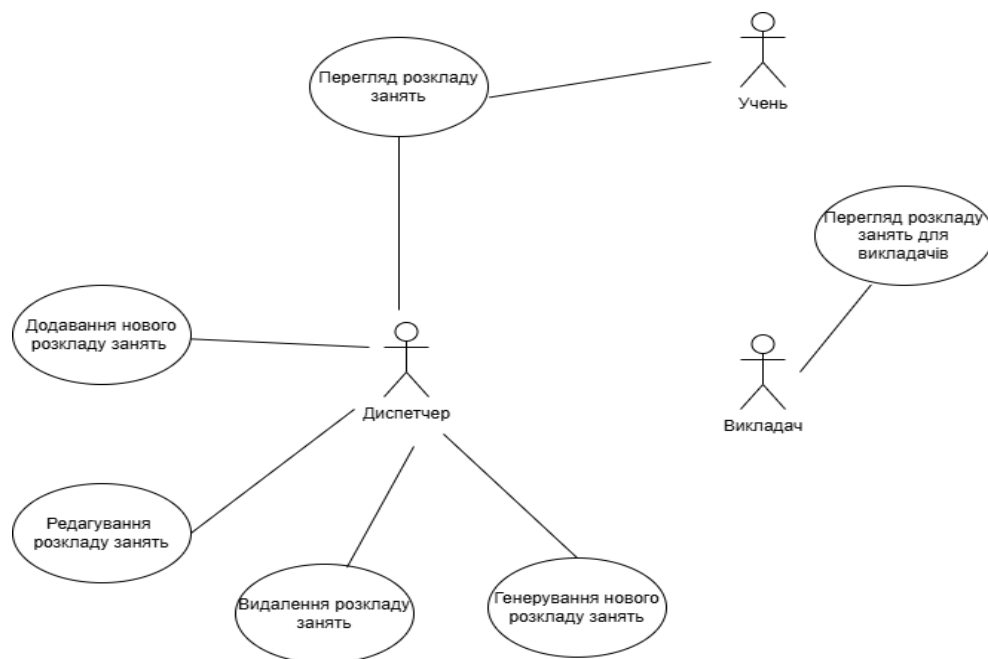


Рисунок 2.4 – Діаграма варіантів використання вебзастосунку

2.2 Розробка архітектури до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

2.2.1 Обґрунтування вибору архітектурного стилю до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Архітектурний стиль, для вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею, залежить від її функціональних і нефункціональних вимог:

- Компонентна архітектура підходить через чітке розділення відповідальності між компонентами.
- Клієнт-серверна архітектура корисна для роботи в багатокористувацькому середовищі, де є центральний сервер для обробки даних і клієнти для взаємодії.
- Архітектура з використанням MVC (Model-View-Controller) дозволяє відокремити бізнес-логіку (модель) від користувацького інтерфейсу (представлення), що забезпечує зручність у супроводі та масштабуванні.

Для розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею було обрана архітектура з використанням MVC:

- Модель (Model): містить логіку обробки даних (інформація про викладачів, предмети та розклад).
- Контролер (Controller): обробляє запити користувачів і координує роботу моделі та уявлення.
- Представлення (View): забезпечує інтерфейс для користувачів.

2.2.2 Архітектура до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Основні компоненти:

Schedule Controller (Управління розкладом):

- Додавання розкладу
- Редагування розкладу
- Видалення розкладу
- Генерація нового розкладу

User Interface (Інтерфейс користувача):

- Вебінтерфейс для диспетчера, викладачів та учнів
- API для зовнішніх інтеграцій

Специфікація компонентів:

- **ScheduleController:** відповідає за запити на додавання/редагування/видалення розкладу та генерацію нового розкладу.
- **DataRepository:** зберігає дані про викладачів, предмети та розклад до бази даних.
- **UI Module:** забезпечує зручну взаємодію з користувачем.

Діаграма компонентів покаже основні модулі вебзастосунку та їх взаємодію представлено на рисунку 2.5.

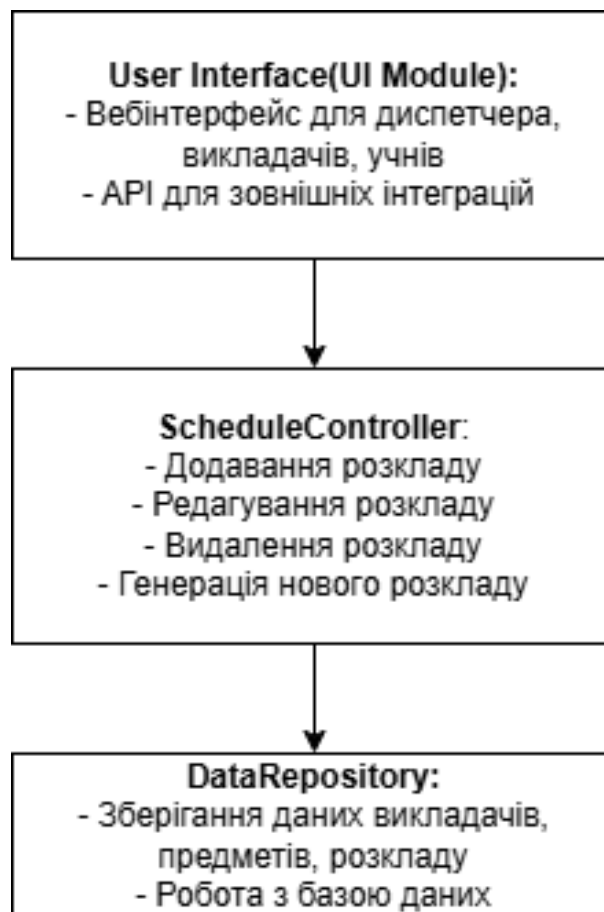


Рисунок 2.5 – Діаграма компонентів

Діаграма розміщення представлено на рисунку 2.6.

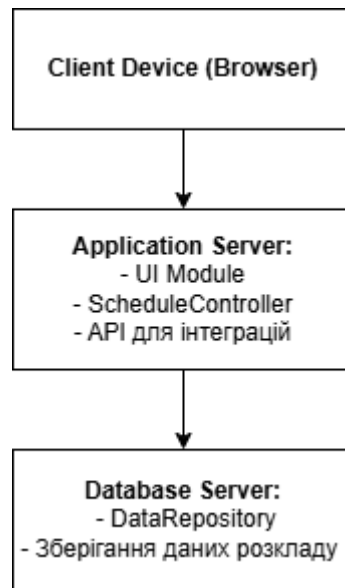


Рисунок 2.6 – Діаграма розміщення

2.3 Розробка проєкту вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

2.3.1 Розробка концептуальної моделі бази даних вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Концептуальна модель даних є абстрактною представленням даних, яке описує концепції та взаємозв'язки між різними сутностями у вебзастосунку. Це важливий етап в розробці бази даних, оскільки вона дозволяє визначити структуру та логіку зберігання інформації без деталей реалізації.

Ключові поняття концептуальної моделі даних включають в себе:

Сутності: Представляють основні об'єкти, про які збирається інформація (наприклад, “Клієнт”, “Продукт”, “Замовлення”).

Атрибути: Це характеристики сутностей, які описують їхні властивості (наприклад, для сутності “Клієнт” атрибутами можуть бути ім'я, адреса, телефон).

Відносини: Визначають зв'язки між сутностями і вказують на те, як вони взаємодіють (наприклад, “Клієнт зробив Замовлення”).

Ключі: Унікальні ідентифікатори для сутностей, які дозволяють однозначно ідентифікувати записи в базі даних.

Атрибути унікальності та цілісності: Визначають обмеження щодо допустимих значень та відносин між ними.

Сутності:

- Розклад;
- Предмети;
- Класи;
- Викладач_предмет_клас;
- Викладачі.

Атрибути сутності “Розклад” представлені у таблиці 2.5.

Таблиця 2.5 – Атрибути сутності “Розклад”

Атрибут	Опис
Код розкладу	Унікальний код розкладу
день тижня	Назва дня тижня
номер уроку	Номер кожного уроку
клас	Номер класу
предмет	Назва предмету
викладач	ПІБ викладача

Атрибути сутності “Класи” представлені у таблиці 2.6.

Таблиця 2.6 – Атрибути сутності “Класи”

Атрибут	Опис
Код класу	Унікальний код класу
Номер класу	Номер кожного класу

Атрибути сутності “ Викладач_предмет_клас ” представлені у таблиці 2.7.

Таблиця 2.7 – Атрибути сутності “ Викладач_предмет_клас ”

Атрибут	Опис
Код викладач_предмет_класу	Унікальний код викладач_предмет_класу
Код класу	Парний або не парний тиждень
Код викладача	Унікальний код викладача
Код предмету	Унікальний код предмету
Години	Кількість годин на тиждень

Атрибути сутності “ Предмети ” представлені у таблиці 2.8.

Таблиця 2.8 – Атрибути сутності “ Предмети ”

Атрибут	Опис
Код предмету	Унікальний код предмету
Назва предмету	Назва предмету

Атрибути сутності “ Викладачі ” представлені у таблиці 2.9.

Таблиця 2.9 – Атрибути сутності “ Викладачі ”

Атрибут	Опис
Код викладача	Унікальний код викладача
ПІБ	Прізвище, ім'я, по батькові викладача

Відносини між сутностями представлені у таблиці 2.10.

Таблиця 2.10 – Зв'язки між сутностями

Назва сутності	Назва сутності	Тип зв'язку	Пояснення
Класи	Розклад	1:M	Для одного класу є багато занять в розкладі, а кожному заняттю в розкладі відповідає один клас
Викладач_предмет_клас	Розклад	1:M	Для викладач_предмет_класу є багато занять в розкладі, а кожному заняттю відповідає один викладач_предмет_клас
Предмети	Розклад	1:M	Для одного предмету є багато занять в розкладі, а кожному заняттю в розкладі відповідає один предмет
Розклад	Викладачі	M:1	Для кожного викладача є багато занять в розкладі, а кожному заняттю в розкладі відповідає один викладач

2.3.2 Побудова діаграми "сутність-зв'язок" вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Концептуальна модель представлена на рисунку 2.7.

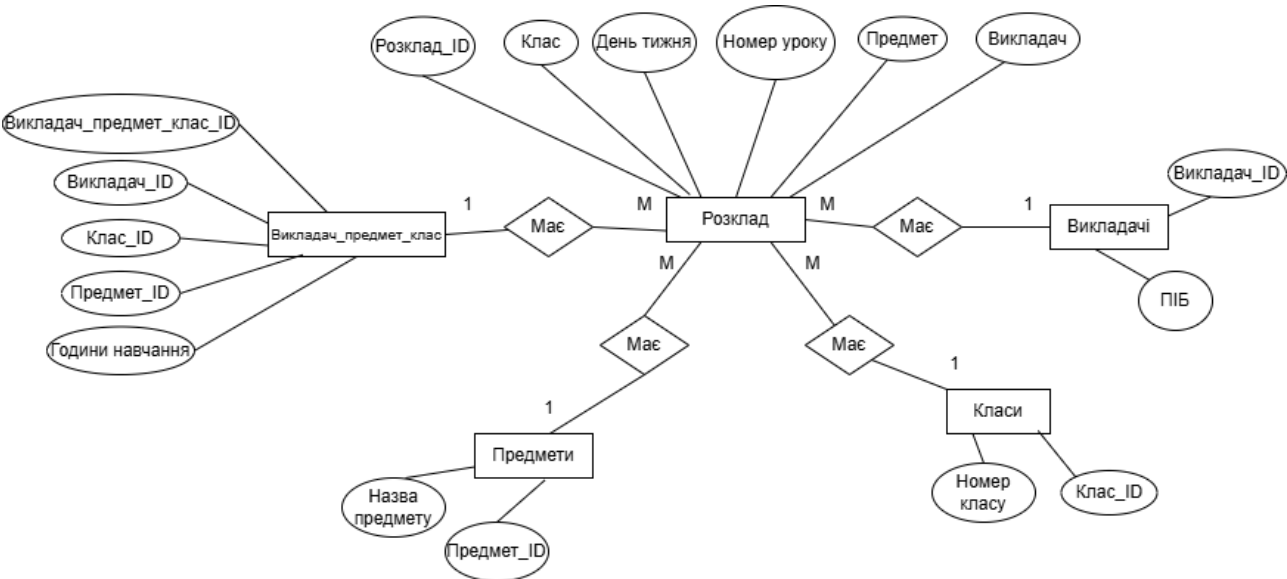


Рисунок 2.7 – Діаграма “сутність - зв’язок”

2.3.3 Розробка графічного інтерфейсу вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Прототип користувацького інтерфейсу вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею представлений на рисунках 2.8-2.10.

Вийти

Розклад для класу 5А

Понеділок

Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
2	Історія	Скуратовська Н.Г.	Редагувати	Видалити
3	Англійська мова	Ткачук О.М.	Редагувати	Видалити

Вівторок

Урок №	Предмет	Викладач	Редагування	Видалення
1	Англійська мова	Ткачук О.М.	Редагувати	Видалити
2	Українська мова	Авраменко Л.А.	Редагувати	Видалити
3	Історія	Скуратовська Н.Г.	Редагувати	Видалити

Рисунок 2.8 – Прототип сторінки “Розклад занять”

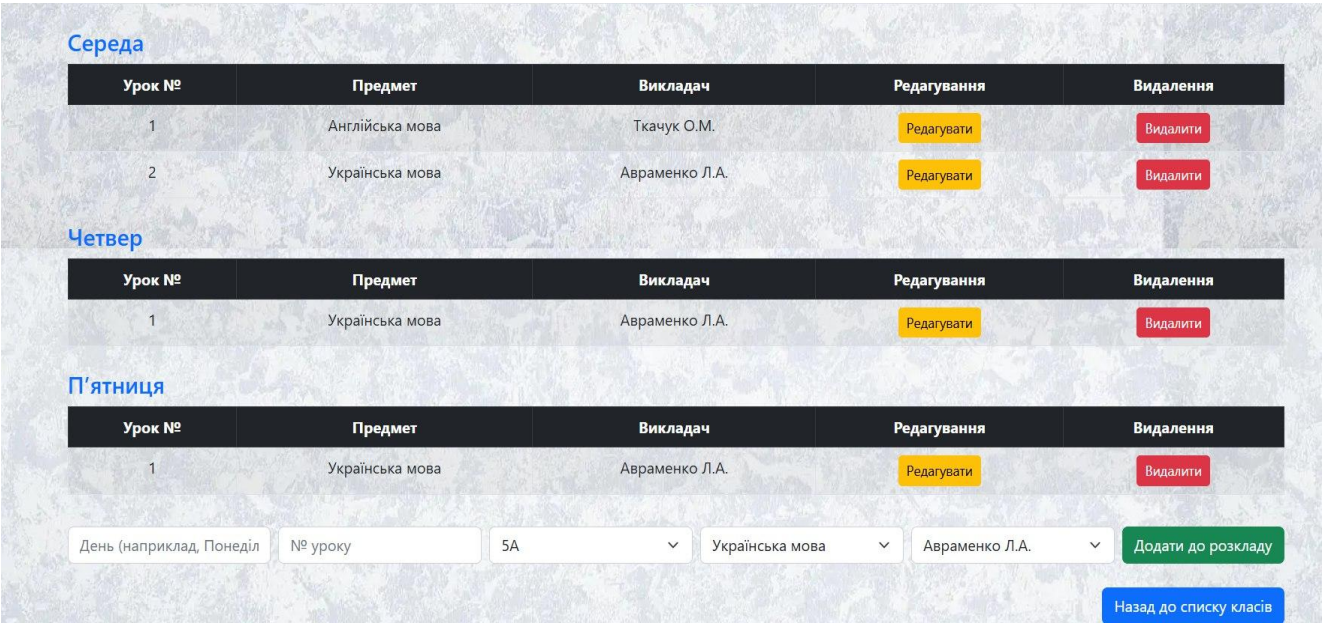


Рисунок 2.9 – Прототип сторінки “Розклад занять”

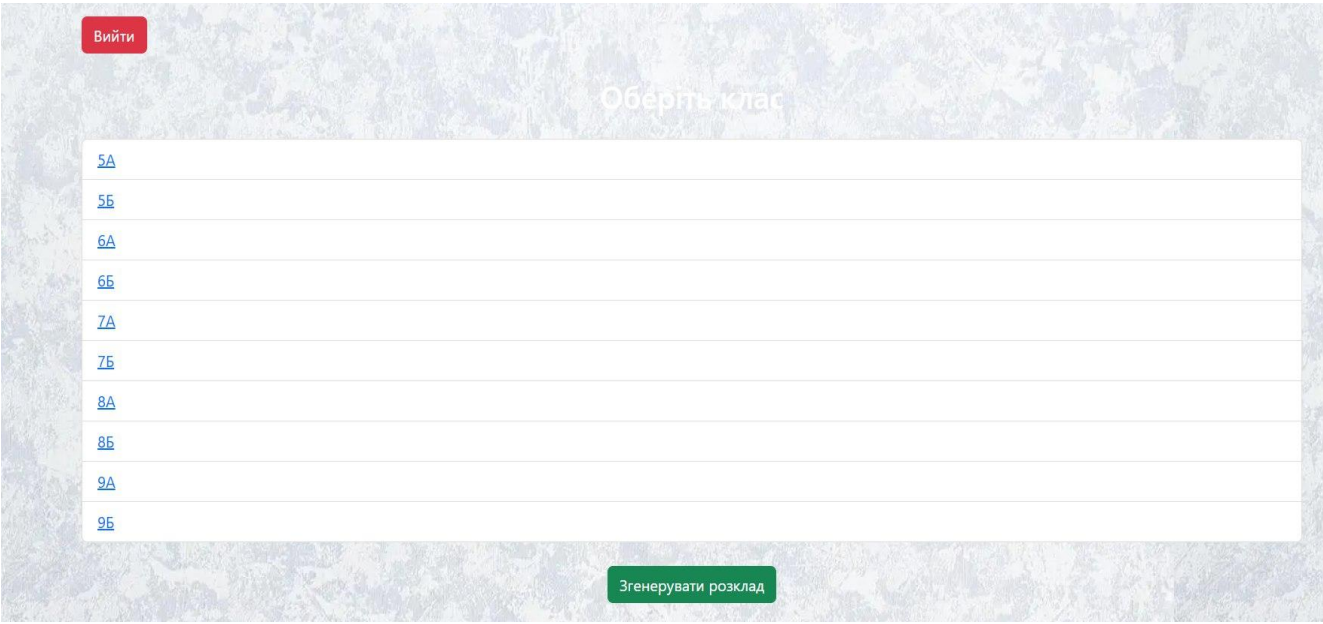


Рисунок 2.10 – Прототип сторінки “Список класів”

Інтерфейс вебзастосунку грає ключову роль у покращені роботи з розкладом занять, роблячи цю роботу більш зручною та ефективною. Інтуїтивно зрозумілі елементи керування вебзастосунку допомагають користувачеві швидко орієнтуватися у вебзастосунку без додаткового навчання по роботі з ним.

2.3.4 Розробка логічної моделі даних вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Логічна модель даних - це деталізоване відображення концептуальної моделі, що враховує внутрішні аспекти бази даних і її компонентів. Це вже більш конкретний погляд на те, як дані будуть зберігатися та організовані в базі даних.

Основні елементи логічної моделі даних включають:

Таблиці: Представляють сутності концептуальної моделі і стають реальними таблицями в базі даних. Кожна таблиця має колонки, які відображають атрибути сутностей.

Колонки: Відповідають атрибутам сутностей і визначають тип даних, який буде зберігатися в кожній конкретній колонці.

Ключі: Включають первинні ключі, що ідентифікують унікальні записи в таблицях, і зовнішні ключі, що визначають зв'язки між таблицями.

Зв'язки: Визначають взаємозв'язки між таблицями та відображаються за допомогою зовнішніх ключів.

Нормалізація: Процес організації даних для уникнення аномалій та забезпечення ефективності.

Логічна модель даних представлена на рисунку 2.11.

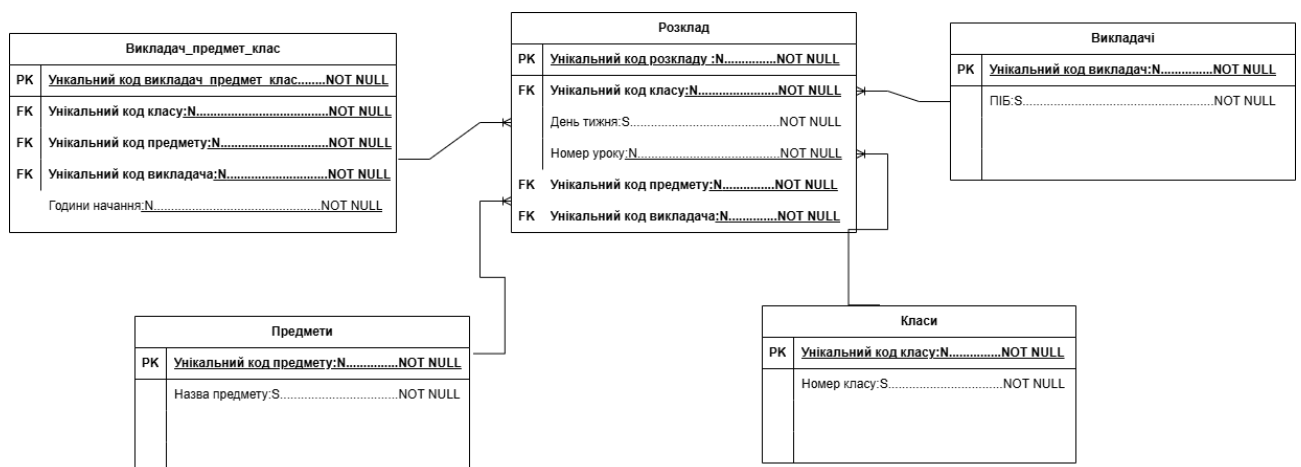


Рисунок 2.11 – Логічна модель даних

2.3.5 Розробка діаграми класів вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Діаграма класів – це набір статичних, декларативних елементів моделі, інформація з діаграми класів відображається у вихідний код програми. Таким чином, діаграма класів кінцевий результат проектування та відправна точка процесу розробки.

Діаграма класів представлена на рисунку 2.12.

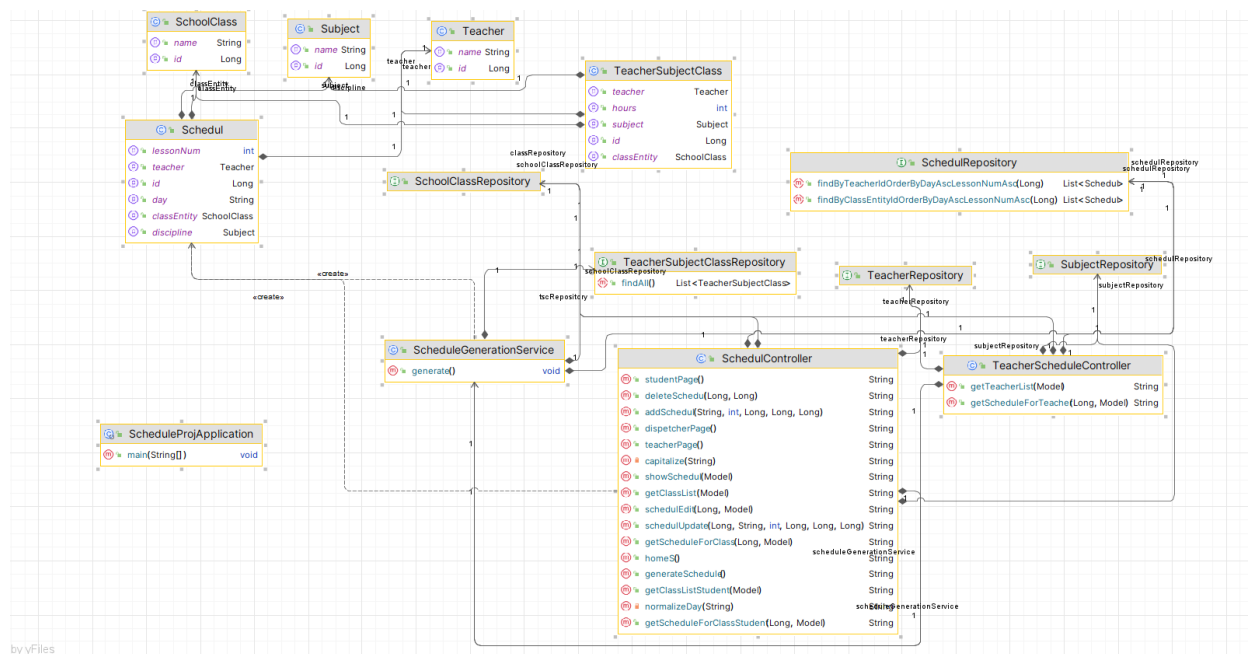


Рисунок 2.12 – Діаграма класів

2.3.6 Специфікація основних класів вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Клас: «SchedulerController»

Відповідальність: Клас який відповідає за більшість основних функцій розкладу

Нижче представлені функції класу

- studentPage() – відповідає за перенаправлення користувача на сторінку для студентів;

- deleteSchedul() – відповідає за видалення розкладу;

- addSchedul()

- dispatcherPage() – відповідає за перенаправлення користувача на сторінку для диспетчера;

- teacherPage() – відповідає за перенаправлення користувача на сторінку для викладача;

- capitalize() – відповідає за форматування назв днів;

- showSchedul() – відповідає за показ розкладу занять;

- schedulEdit() – відповідає за редагування розкладу;

- schedulUpdate() – відповідає за оновлення редагованого розкладу;

- homeS() – відповідає за повернення на початкову сторінку;

- generateSchedule() – відповідає за виклик сервісу scheduleGenerationService;

- normalizeDay() – відповідає за нормалізацію назв дня тижня.

Клас: «TeacherScheduleController»

Відповідальність: Клас який відповідає за розклад для викладачів

Нижче представлені функції класу

- getTeacherList() – відповідає за створення списку викладачів;

- getSchedulForTeacher() – відповідає за створення розкладу для викладача.

Клас: «ScheduleGenerationService»

Відповідальність: Клас який відповідає за логіку генерації розкладу занять

Нижче представлені функції класу

- generate() – відповідає за логіку генерації розкладу занять з урахуванням всіх потрібних умов.

2.3.7 Розробка фізичної моделі даних вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Фізична модель даних є однією з трьох ключових моделей бази даних, разом з концептуальною і логічною моделями. Фізична модель зосереджена на тому, як дані будуть зберігатися, організовані та взаємодіяти у фізичному середовищі бази даних, включаючи аспекти, такі як типи даних, індексація, партіціонування та інші деталі реалізації.

Фізична модель даних представлена на рисунку 2.13.

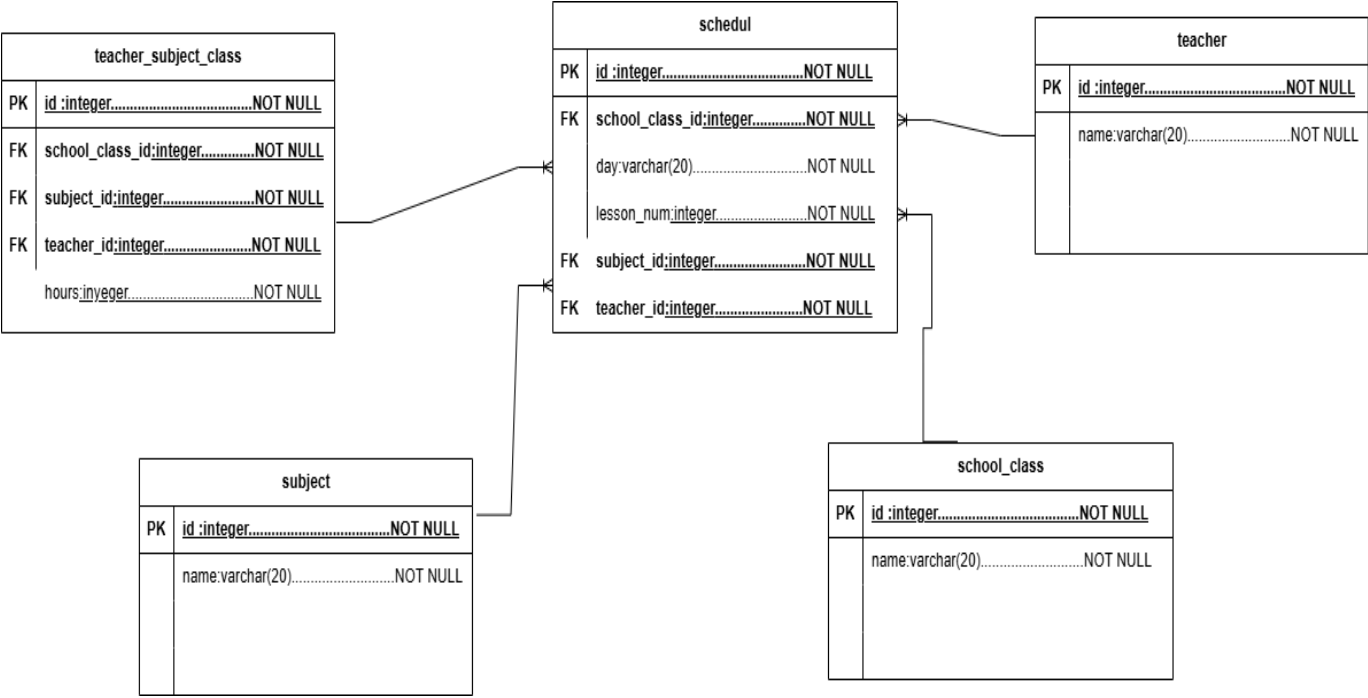


Рисунок 2.13 – Фізична модель даних

Відповідність назв таблиць та атрибутів логічної та фізичної моделей представлені на таблиці 2.11.

Таблиця 2.11 – Відповідність назв таблиць та атрибутів логічної та фізичної моделей

Назва таблиці в логічній моделі	Назва таблиці в фізичній моделі	Атрибути логічної моделі	Атрибути фізичної моделі
1	2	3	4
Розклад	schedul	Унікальний код розкладу	id
		День тижня	day
		Номер уроку	lesson_num
		Унікальний код класу	school_class_id
		Унікальний код предмету	subject_id
		Унікальний код викладача	teacher_id
Викладачі	teacher	Унікальний код викладача	id
		ПІБ	name
Предмети	subject	Унікальний код предмету	id
		Назва предмету	name
Класи	school_class	Унікальний код класу	id
		Номер класу	name

Кінець таблиці 2.11

1	2	3	4
Викладач_предмет_клас	teacher_subject _class	Унікальний код	id
		Унікальний код класу	school_class_id
		Унікальний код предмету	subject_id
		Унікальний код викладача	teacher_id
		Години	hours

2.4 Реалізація проєкту вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

2.4.1 Використані СКБД та інші програмні засоби до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Для кодування програмного забезпечення було обране програмне забезпечення JetBrains IntelliJ IDEA. JetBrains IntelliJ IDEA – це провідне середовище швидкої розробки на мові Java, включаючи Java 8. IntelliJ IDEA являє собою високотехнологічний комплекс тісно інтегрованих інструментів програмування, що включає інтелектуальний редактор вихідних текстів з розвиненими засобами автоматизації, потужні інструменти рефакторинга коду, вбудовану підтримку технологій J2EE, механізми інтеграції з середовищем тестування Ant / JUnit і системами управління версіями, унікальний інструмент оптимізації та перевірки коду Code Inspection, а також інноваційний візуальний конструктор графічних інтерфейсів [3].

Мова програмування була обрана Java 17-ої версії. Java є мовою програмування, за допомогою якої розробники програмного забезпечення створюють різні прикладні додатки для комп'ютерів, смартфонів, планшетів та інших інтелектуальних пристроїв. Особливістю програм на Java є те, що вони можуть запускатись на будь-яких комп'ютеризованих пристроях, які працюють під різними операційними системами, причому без повторної компіляції коду [4].

Також в програмі були використані наступні програмні засоби:

Maven – це інструмент побудови та управління проєктами, який зазвичай використовується в фреймворках, побудованих на Java. Він розроблений Apache Software Foundation. Maven, слово з мови ідиш, означає «збирач знань». Він був введений, щоб зробити процес запуску побудови в Джакартському турбінному проєкті [5].

Maven контролюється файлом Project Object Model (pom). Під час роботи з вбудованими фреймворками Java нам часто доводиться мати справу з низкою залежностей [5].

До того, як Maven з'явився в картині, усі залежності, які є не чим іншим, як файлами JAR, повинні були бути додані в наш фреймворк вручну. Крім того, нам потрібно було подбати про оновлення програмного забезпечення в нашому проєкті [5].

CSS – це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи вебсторінки), написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі розмічені мовою HTML, XHTML та XML [6].

HTML – скорочення від “HyperText Mark-up Language” – перекладається як “Мова розмітка гіпертексту” (Гіпертекст – це текст, що не послідовно зв'язаний з іншими документами, тобто у вас є змога з першої сторінки документу перейти на останню). Іншими словами HTML – це мова розмітки, або ще один спосіб зберігання інформації. За допомогою HTML ти позначаєш текст, вказуючи своєму вебпереглядачу, як він має розуміти позначений текст, так само як і на жорсткому диску інформація зберігається в блоках, кластерах, секторах,

доріжках і тільки за допомогою, такої, визначеної структури твій комп'ютер розуміє, що треба, а що не треба зчитувати [7].

Spring Boot – це Java фреймворк, що є частиною Java–EE фреймворку Spring Framework. Фактично, Spring Boot являє собою плагін для систем автоматичного збирання проєктів Maven та Gradle. Він надає можливості для комфортної розробки та тестування Spring додатків. Окрім того, Spring Boot дозволяє упаковувати додаток в окремий jar-файл із вбудованим повноцінним контейнером Tomcat. Основною перевагою використання Spring Boot є зведення до мінімуму конфігураційних xml файлів, які зазвичай використовуються при розробці Spring додатку [8].

MariaDB – це зворотно сумісна гілка MySQL® Database Server, що замінює її. Він включає всі основні механізми зберігання з відкритим кодом, включаючи механізм зберігання Maria [9].

2.4.2 Фрагменти коду до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Найбільш показові фрагменти кодів:

1) Додавання розкладу:

```
@PostMapping("/addschedul")
public String addSchedul(@RequestParam String day,
                        @RequestParam int lessonNum,
                        @RequestParam Long classId,
                        @RequestParam Long subjectId,
                        @RequestParam Long teacherId) {

    String normalizedDay = normalizeDay(day);

    Schedul schedul = new Schedul();
    schedul.setDay(normalizedDay);
    schedul.setLessonNum(lessonNum);

    schedul.setClassEntity(schoolClassRepository.findById(classId).orElseThrow());
    ;

    schedul.setDiscipline(subjectRepository.findById(subjectId).orElseThrow());
    schedul.setTeacher(teacherRepository.findById(teacherId).orElseThrow());

    schedulRepository.save(schedul);
    return "redirect:/schedule/class/" + classId;
}
```

Цей фрагмент коду відповідає за додавання нового розкладу до бази даних. Він отримує дані з форми про розклад, створює об'єкт "Schedul", до цього об'єкту додається нова інформація і після цього зберігає створений запис до бази даних.

2) Видалення розкладу:

```
@GetMapping("/delete_schedul")
public String deleteSchedul(@RequestParam Long id, @RequestParam Long
classId) {
    schedulRepository.deleteById(Math.toIntExact(id));
    return "redirect:/schedule/class/" + classId;
}
```

Цей фрагмент коду видаляє розклад з бази даних за вказаним id.

3) Редагування розкладу:

```
@GetMapping("/schedul_edit")
public String schedulEdit(@RequestParam Long id, Model model) {
    Optional<Schedul> optionalSchedul =
    schedulRepository.findById(Math.toIntExact(id));
    if (optionalSchedul.isEmpty()) {
        return "redirect:/schedul";
    }
    model.addAttribute("schedul", optionalSchedul.get());
    model.addAttribute("classes", schoolClassRepository.findAll());
    model.addAttribute("subjects", subjectRepository.findAll());
    model.addAttribute("teachers", teacherRepository.findAll());
    return "schedul_edit";
}
```

Цей фрагмент коду шукає розклад за вказаним id, якщо такого запису не знайдено, то перенаправляє користувача. Якщо запис знайдено, то у шаблон сторінки передаються параметри розкладу, та перенаправляє користувача на сторінку з редагуванням.

```
@PostMapping("/update_schedul")
public String schedulUpdate(@RequestParam Long id,
    @RequestParam String day,
    @RequestParam int lessonNum,
    @RequestParam Long classId,
    @RequestParam Long subjectId,
    @RequestParam Long teacherId) {
    Optional<Schedul> optionalSchedul =
    schedulRepository.findById(Math.toIntExact(id));
    optionalSchedul.ifPresent(s -> {
        s.setDay(normalizeDay(day));
        s.setLessonNum(lessonNum);

        s.setClassEntity(schoolClassRepository.findById(classId).orElseThrow());

        s.setDiscipline(subjectRepository.findById(subjectId).orElseThrow());
        s.setTeacher(teacherRepository.findById(teacherId).orElseThrow());
        schedulRepository.save(s);
    });
    return "redirect:/schedule/class/" +
    schoolClassRepository.findById(classId).get().getId();
}
```

Цей фрагмент коду шукає в базі розклад з даним id , якщо запис знайдено, то виконується оновлення даних цього розкладу. Після оновлення запис зберігається у базі даних. Після оновлення користувача перенаправляє до сторінки розкладу, з уже оновленим розкладом відповідного класу.

4) Створення розкладу:

```
private final SchedulerRepository schedulerRepository;
private final TeacherSubjectClassRepository tscRepository;
private final SchoolClassRepository classRepository;

private static final List<String> DAYS = List.of("Понеділок", "Вівторок",
"Середа", "Четвер", "П'ятниця");
private static final int MAX_LESSONS_PER_DAY = 7;

public void generate() {
    schedulerRepository.deleteAll();

    List<TeacherSubjectClass> assignments = tscRepository.findAll();
    List<SchoolClass> classes = classRepository.findAll();

    Map<Long, Map<String, Set<Integer>>> teacherSchedule = new
    HashMap<>();

    for (SchoolClass schoolClass : classes) {
        List<TeacherSubjectClass> classAssignments = assignments.stream()
            .filter(a ->
a.getClassEntity().getId().equals(schoolClass.getId()))
            .toList();

        Map<Long, Integer> assignmentUsage = new HashMap<>();

        Map<String, List<TeacherSubjectClass>> dailySubjects = new
        HashMap<>();
        for (String day : DAYS) {
            List<TeacherSubjectClass> shuffled = new
            ArrayList<>(classAssignments);
            Collections.shuffle(shuffled);
            dailySubjects.put(day, shuffled);
        }

        for (String day : DAYS) {
            Set<String> usedSubjectsToday = new HashSet<>();

            int lessonCount = 0;

            List<TeacherSubjectClass> subjectsToday =
            dailySubjects.get(day);

            for (int lessonNum = 1; lessonNum <= MAX_LESSONS_PER_DAY &&
            lessonCount < MAX_LESSONS_PER_DAY; lessonNum++) {
                for (TeacherSubjectClass assign : subjectsToday) {

                    Long assignId = assign.getId();
                    Long teacherId = assign.getTeacher().getId();
                    String subjectName = assign.getSubject().getName();

                    int used = assignmentUsage.getOrDefault(assignId, 0);
                    if (used >= assign.getHours()) continue;

                    if (usedSubjectsToday.contains(subjectName)) continue;
```

```

        teacherSchedule.putIfAbsent(teacherId, new
HashMap<>());
        Map<String, Set<Integer>> daily =
teacherSchedule.get(teacherId);
        daily.putIfAbsent(day, new HashSet<>());
        if (daily.get(day).contains(lessonNum)) continue;

        Schedul schedul = new Schedul();
        schedul.setDay(day);
        schedul.setLessonNum(lessonNum);
        schedul.setClassEntity(schoolClass);
        schedul.setDiscipline(assign.getSubject());
        schedul.setTeacher(assign.getTeacher());

        schedulRepository.save(schedul);

        assignmentUsage.put(assignId, used + 1);
        usedSubjectsToday.add(subjectName);
        daily.get(day).add(lessonNum);

        lessonCount++;
        break;
    }
}
}
}
}

```

Цей фрагмент коду відповідає за автоматичне створення розкладу на тиждень. Перше, що він робить це видаляє попередній розклад перед створенням нового.

```
schedulRepository.deleteAll();
```

Далі він отримує дані, призначення предметів та вчителів для класів і всі доступні класи.

```

List<TeacherSubjectClass> assignments = tscRepository.findAll();
List<SchoolClass> classes = classRepository.findAll();

```

Створюється порожній розклад для вчителів, це для того щоб уникнути конфліктів (один учитель не може вести більше одного уроку одночасно)

```
Map<Long, Map<String, Set<Integer>>> teacherSchedule = new HashMap<>();
```

Далі фільтрує предмети які стосуються визначеного класу.

```

List<TeacherSubjectClass> classAssignments = assignments.stream()
    .filter(a -> a.getClassEntity().getId().equals(schoolClass.getId()))
    .toList();

```

Для кожного дня створюється окремий список предметів, перемішаний випадковим чином, щоб розклад був варіативним.

```

for (String day : DAYS) {
    List<TeacherSubjectClass> shuffled = new ArrayList<>(classAssignments);
    Collections.shuffle(shuffled);
    dailySubjects.put(day, shuffled);
}

```

Цей цикл обирає перший доступний предмет, який ще має години. Перевіряє, чи не повторюється предмет у цей день. Потім перевіряє, чи вчитель

вільний на цей урок і після цього створює “Schedul” та зберігає розклад до бази даних.

```

    for (int lessonNum = 1; lessonNum <= MAX_LESSONS_PER_DAY && lessonCount <
MAX_LESSONS_PER_DAY; lessonNum++) {
        for (TeacherSubjectClass assign : subjectsToday) {

            Long assignId = assign.getId();
            Long teacherId = assign.getTeacher().getId();
            String subjectName = assign.getSubject().getName();

            int used = assignmentUsage.getOrDefault(assignId, 0);
            if (used >= assign.getHours()) continue;

            if (usedSubjectsToday.contains(subjectName)) continue;

            teacherSchedule.putIfAbsent(teacherId, new HashMap<>());
            Map<String, Set<Integer>> daily = teacherSchedule.get(teacherId);
            daily.putIfAbsent(day, new HashSet<>());
            if (daily.get(day).contains(lessonNum)) continue;

            Schedul schedul = new Schedul();
            schedul.setDay(day);
            schedul.setLessonNum(lessonNum);
            schedul.setClassEntity(schoolClass);
            schedul.setDiscipline(assign.getSubject());
            schedul.setTeacher(assign.getTeacher());

            schedulRepository.save(schedul);

            assignmentUsage.put(assignId, used + 1);
            usedSubjectsToday.add(subjectName);
            daily.get(day).add(lessonNum);

            lessonCount++;
            break;
        }
    }

```

2.5 Тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

2.5.1 Обґрунтування вибору тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

До вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею існує багато різних видів тестування, але було обране саме тестування чорного ящика.

Тестування чорного ящика — це тестування, при якому тестувальник немає доступу до внутрішньої структури коду. Перевіряються тільки вхідні дані, очікувана поведінка вебзастосунку та реальний результат роботи.

Саме цей вид тестування дозволить перевірити правильну функціональність вебзастосунку на користувацькому рівні, що дозволить найкраще зрозуміти чи буде звичайному користувачеві працювати з цим вебзастосунком.

2.5.2 Тестування чорного ящика до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Тестування чорного ящика до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею представлено на таблиці 2.12.

Таблиця 2.12 – Тестування чорного ящика

Назва тесту	Вхідні дані	Очікуваний результат	Реальний результат
1	2	3	4
Додавання розкладу	День(Пт), урок(2), клас(5А), предмет(Українська мова), викладач(Авраменко Л.А.)	Успішне додавання розкладу	Успішно
Редагування розкладу	Id(438), урок(3)	Успішне редагування розкладу	Успішно
Видалення розкладу	Id(438)	Успішне видалення розкладу	Успішно

Кінець таблиці 2.12

1	2	3	4
Створення розкладу	Натиснути на кнопку “Згенерувати розклад”	Успішне створення розкладу	Успішно
Перегляд розкладу	Натиснути на кнопку “Список класів”, та обираємо клас “5А”	Успішний перегляд розкладу для класу “5А”	Успішно

Всі успішні тестування представлені на рисунках 2.14 — 2.23.

Рисунок 2.14 – Вхідні дані для додавання розкладу

П'ятниця

Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
2	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок 2.15 – Результат додавання розкладу

Рисунок 2.16 – Редагування доданого розкладу

П'ятниця

Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
3	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок 2.17 – Результат редагування розкладу

П'ятниця

Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок 2.18 – Видалення редагованого розкладу

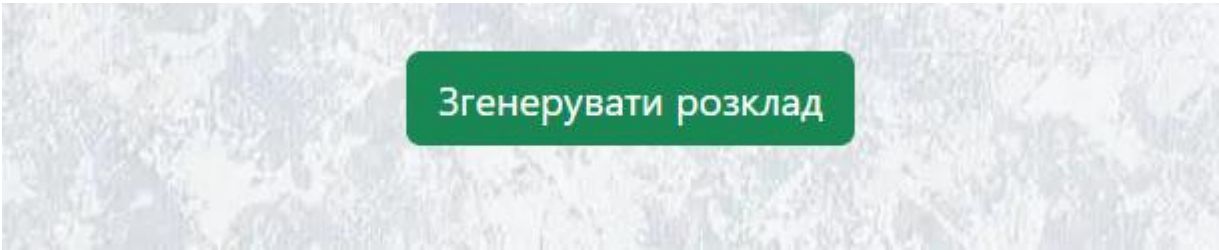


Рисунок 2.19 – Кнопка для створення розкладу

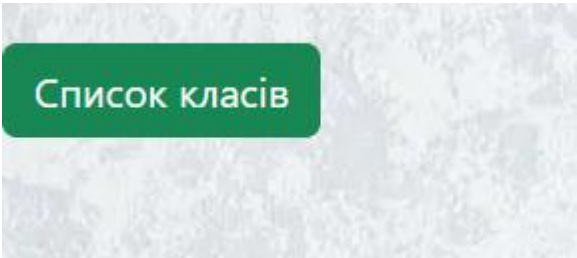


Рисунок 2.20 – Кнопка для перегляду списку класів

Оберть клас

5А
5Б
6А
6Б
7А
7Б
8А
8Б
9А
9Б

Рисунок 2.21 – Список класів

Розклад для класу 5А				
Понеділок				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
2	Англійська мова	Ткачук О.М.	Редагувати	Видалити
3	Історія	Скуратовська Н.Г.	Редагувати	Видалити
Вівторок				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Англійська мова	Ткачук О.М.	Редагувати	Видалити
2	Українська мова	Авраменко Л.А.	Редагувати	Видалити
3	Історія	Скуратовська Н.Г.	Редагувати	Видалити

Рисунок 2.22 – Створений розклад занять для 5А класу

Середа				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
2	Англійська мова	Ткачук О.М.	Редагувати	Видалити
Четвер				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
П'ятниця				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок 2.23 – Створений розклад занять для 5А класу

3 РЕЗУЛЬТАТ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

3.1 Загальний опис результатів розробки вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею

У рамках кваліфікаційної роботи був розроблений вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею з функціоналом додавання, редагування, видалення, створення та перегляду розкладу занять. Було реалізовано 3 основних модулі, які відповідають за основний функціонал вебзастосунку.

Модулі та їх функціонал показано на таблиці 3.1.

Таблиця 3.1 – Модулі та функціонал

Модулі	Функціонал
SchedulController	Цей модуль відповідає за всі функції, які стосуються розкладу занять, а саме: додавання, редагування, видалення та перегляд розкладу занять.
TeacherScheduleController	Цей модуль відповідає за створення та перегляд розкладу саме для вчителів.
ScheduleGenerationService	Цей модуль відповідає саме за створення нового розкладу занять.

3.2 Відповідність результатів розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею вимогам технічного завдання

Порівняння, що вимагалось в технічному завданні та, що було реалізовано наведено у таблиці 3.2.

Таблиця 3.2 – Відповідність технічного завдання та реалізації

Вимога з ТЗ	Реалізовано
Авторизація користувача	Частково (поки тільки через кнопки)
Автоматичне створення розкладу	Так
Додавання розкладу	Так
Редагування розкладу	Так
Видалення розкладу	Так
Перегляд розкладу для учнів	Так
Перегляд розкладу для викладачів	Так

3.3 Кількісна та якісна характеристика до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

Загальний обсяг вихідного коду становить 1054 рядка з яких 482 відносяться до frontend, 572 до backend, а розмір файлу дорівнює 45 МБ. Робота вебзастосунку перевірялась на ПК з операційною системою Windows 10 та характеристиками: 4 ГБ ОЗУ, CPU 2 ядра. Запуск вебзастосунку відбувається до 5 секунд, та створення розкладу займає до 3 секунд, всі інші функції спрацьовують за 0.1 секунду. На даний час було проведено випробування тільки на двох пристроях одночасно, у всі дані, що я описав вище були однаковими для

одного користувача і двох одночасно. На виконання цієї роботи пішло близько 30 годин.

В даний час вебзастосунок працює без збоїв та вона не допускає збереження некоректних даних, для цього було додано для таких функцій як редагування та додавання розкладу, списки з шаблонами усіх можливих варіантів додавання чи редагування. Також передбачена масштабованість, тобто якщо буде потрібно, то можливо буде, досить не важко, додавати нові і важливі функції. Інтерфейс було створено максимально простим і зрозумілим. Інтерфейс було протестовано на 3 людях без підготовки та всі змогли виконати завдання.

3.4 Результати тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

В рамках кваліфікаційної роботи було проведено тестування чорного ящика 5 сценаріїв використання по основним функціям вебзастосунку. Всі сценарії пройшли успішно.

4 ОХОРОНА ПРАЦІ

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності [14].

У сфері інформаційних технологій, зокрема при розробці програмного забезпечення, охорона праці має специфічні аспекти, пов'язані з особливостями робочого процесу. Робота розробників програмного забезпечення зазвичай не є фізично важкою, але має високе інтелектуальне та психоемоційне навантаження, що потребує належної організації робочого середовища, режиму роботи та відпочинку.

Роботодавець - власник підприємства, установи, організації або уповноважений ним орган, незалежно від форм власності, виду діяльності, господарювання, і фізична особа, яка використовує найману працю [14].

У сфері розробки програмного забезпечення роботодавець зобов'язаний створити безпечні та нешкідливі умови праці, забезпечити працівників відповідним обладнанням, організувати інструктажі з охорони праці, а також дотримуватися вимог трудового законодавства щодо тривалості робочого дня, перерв, відпусток тощо.

Працівник - особа, яка працює на підприємстві, в організації, установі та виконує обов'язки або функції згідно з трудовим договором (контрактом) [14].

Розробники програмного забезпечення, як і інші працівники, мають право на безпечні умови праці, а також зобов'язані дотримуватися правил внутрішнього трудового розпорядку, інструкцій з охорони праці, техніки безпеки, поводження з технічними засобами та програмним забезпеченням, а також повідомляти про виявлені порушення або небезпеки на робочому місці.

В таблиці 4.1 перераховані шкідливі та небезпечні виробничі фактори, з якими можуть заткнутися, під час роботи, працівники [15].

Таблиця 4.1 – Шкідливі та небезпечні виробничі фактори

Найменування фактору	Можливі джерела його виникнення	Характер дії
Теплові небезпеки	Наявність речей які можуть загорітися (Електрична апаратура)	Небезпечний та шкідливий
Небезпека електромагнітного опромінення	Наявність дисплеїв комп'ютерів, які є джерелом рентгенівського, радіочастотного, ультрафіолетового й інфрачервоного випромінювання та випромінювання звукового діапазону	Шкідливий
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Психофізіологічні небезпеки	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий

Довготривала і інтенсивна робота за комп'ютером може привести до важких професійних та звичайних захворювань.

ВИСНОВКИ

Під час виконання роботи було досягнуто мети кваліфікаційної роботи, а саме розроблено вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею

Був проведений аналіз практичної задачі інженерії програмного забезпечення, а саме задачі розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею. Також було проаналізовані існуючі аналоги та обгрунтовано, чому саме потрібно розробляти власний вебзастосунок. Після цієї роботи була сформульована постановка задачі.

При розробці вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею було виконано ряд завдань. Було проведено аналіз вимог до вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею. Також був обгрунтований вибір архітектурного стилю, розроблено проєкт та реалізовано вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею. Наступним етапом було тестування вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею, де всі тести були успішними.

В кінці роботи було сформовано результат розробки вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. FET. URL:
[https://ru.wikipedia.org/wiki/FET_\(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5\)](https://ru.wikipedia.org/wiki/FET_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5))
2. Smart School. URL: <https://smart-school.com.ua/>
3. Інформація про JetBrains IntelliJ IDEA: вебсайт. URL:
<https://sva.ua/ru/jetbrains-intellij-idea/>
4. Інформація про мову Java: вебсайт. URL: https://dut.edu.ua/ua/news-1-626-6031-mova-programuvannya-java-ta-platforma-javafx-prikladi-zastosuvannya_kafedra-kompyuternih-nauk-ta-informaciyних-tehnologiy
5. Інформація про Maven: вебсайт. URL: <https://uk.myservername.com/what-is-maven-maven-tutorial>
6. Інформація про CSS: вебсайт. URL: https://css.in.ua/article/shcho-take-html_10
7. Інформація про HTML: вебсайт. URL: https://css.in.ua/article/shcho-take-css_3
8. Інформація про Spring: вебсайт. URL: <http://surl.li/gyeie>
9. Інформація про MariaDB: вебсайт. URL:
<https://web.archive.org/web/20100308052059/http://askmonty.org/wiki/index.php/MariaDB>
10. Застосування Spring: вебсайт. URL: <https://spring.io/guides>
11. Використання Thymeleaf: вебсайт. URL:
<https://www.thymeleaf.org/documentation.html>
12. Використання Lombok: вебсайт. URL: <https://projectlombok.org/features/>
13. Застосування Maven: вебсайт. URL: <https://maven.apache.org/>

14. Про охорону праці: Закон України від 21.11.2002 р. No 229-IV. Дата оновлення: 24.08.2024. URL: <https://zakon.rada.gov.ua/laws/main/2694-12#Text>.

15. Методичні вказівки до виконання розділу «ОХОРОНА ПРАЦІ». URL: <https://opcb.kpi.ua/wp-content/uploads/2015/01/DR-%D0%A4%D0%98%D0%9E%D0%A2.pdf>

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

1 Вступ

1.1 Назва програмного забезпечення

Вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею

1.2 Сфера застосування

Вебзастосунок застосовується в освітній сфері для автоматизації створення та управління розкладом навчальних занять.

2 Підстава для розробки програмного забезпечення

Розробка програмного забезпечення виконується на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова.

3 Призначення розробки

3.1 Функціональне призначення

Функціональним призначенням вебзастосунку є створення, редагування, видалення та перегляд розкладу занять для учнів, викладачів і диспетчера.

3.2 Експлуатаційне призначення

Вебзастосунок призначене для використання в локальній або мережевій інфраструктурі навчального закладу.

4 Технічні вимоги до програми

4.1 Вимоги до функціональних характеристик

4.1.1 Вимоги до переліку виконуваних функцій

- Авторизація користувача
- Автоматичне створення розкладу

- Додавання, редагування та видалення розкладу
- Перегляд розкладу для учнів та викладачів

4.1.2 Вимоги до організації вхідних та вихідних даних

1) Редагування розкладу

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження редагування.

2) Додавання розкладу

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження додавання.

3) Видалення розкладу

Вхідні дані: День тижня, номер уроку, клас, назва дисципліни, викладач.

Вихідні дані: Підтвердження видалення.

4) Формування розкладу

Вхідні дані: Натиснути на кнопку «Розклад».

Вихідні дані: Розклад занять (День тижня, номер уроку, клас, назва дисципліни, викладач).

5) Генерування розкладу занять

Вхідні дані: Натиснути на кнопку «Створити розклад занять».

Вихідні дані: Розклад занять (День тижня, номер уроку, клас, назва дисципліни, викладач).

6) Формування розкладу занять для викладачів

Вхідні дані: Натиснути на кнопку «Розклад для викладачів».

Вихідні дані: Розклад занять для викладача (День тижня, номер уроку, клас, назва дисципліни).

4.2 Вимоги до надійності

Час відповіді сервера не повинен перевищувати 3 секунд. Вебзастосунок повинен стабільно працювати при середньому навантаженні.

4.3 Умови експлуатації

4.3.1 Кліматичні умови експлуатації

Умови стандартного офісного середовища.

4.3.2 Вимоги до чисельності та кваліфікації користувачів

Користувач повинен мати базові навички роботи з браузером.

4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Вимоги до користувацького обладнання

ПК з Windows 10 або вище, мінімум 4 ГБ ОЗУ, процесор з 2 ядрами.

4.4.2 Вимоги до серверного обладнання

Підтримка бази даних MariaDB, сумісність з Java сервлетами.

4.5 Вимоги до інформаційної та програмної сумісності

Клієнт повинен мати сучасний браузер (Chrome, Firefox, Edge). Сервер має підтримувати Java, MariaDB.

4.6 Вимоги до маркування та упаковки

Вимоги до маркування та упаковки відсутні.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання відсутні.

4.8 Спеціальні вимоги

Спеціальні вимоги відсутні.

5 Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- текст програми;
- керівництво користувача;
- опис програми;
- програму і методику випробувань.

6 Техніко-економічні показники

У кваліфікаційній роботі техніко-економічні показники не розраховуються.

7 Стадії та етапи розробки

Етапи розробки програмного забезпечення наведено в таблиці А.1.

Таблиця А.1 - Етапи розробки програмного забезпечення

Номер	Етап розробки	Термін виконання
1.	Аналіз вимог	14.04.2025
2.	Розробка архітектури	21.04.2025
3.	Розробка проєкту	05.05.2025
4.	Реалізація проєкту	12.05.2025
5.	Тестування	12.05.2025

8 Порядок контролю і прийомки

Контроль здійснюється керівником кваліфікаційної роботи на кожному етапі розробки програмного забезпечення. Приймання здійснюється за програмою та методикою випробувань. Випробування проводиться в зазначений термін.

ДОДАТОК Б – ТЕКСТ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

Текст програми представлені на лістингах 1-26 . Java-класи на лістингах 1-14. Сторінки HTML на 15-25. Таблиця стилю CSS представлена на лістингу 26.

Лістинг 1

ScheduleProjApplication.java

Цей код є головним класом запуску Spring Boot застосунку.

```
package com.example.scheduleproj;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class ScheduleProjApplication {
    public static void main(String[] args) {
        SpringApplication.run(ScheduleProjApplication.class, args);
    }
}
```

Лістинг 2

SchedulController.java

Цей код відповідає за основні функції такі як: додавання, редагування, видалення, перегляд розкладу занять

```
package com.example.scheduleproj.controllers;

import com.example.scheduleproj.dao.SchedulRepository;
import com.example.scheduleproj.dao.SubjectRepository;
import com.example.scheduleproj.dao.TeacherRepository;
import com.example.scheduleproj.dao.SchoolClassRepository;
import com.example.scheduleproj.entities.Schedul;
import com.example.scheduleproj.entities.SchoolClass;
import lombok.AllArgsConstructor;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import com.example.scheduleproj.ScheduleGenerationService;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.util.*;

@Controller
```

```

@AllArgsConstructor
public class SchedulController {
    private static final Map<String, String> DAY_NORMALIZATION = Map.ofEntries(
        Map.entry("пн", "Понеділок"),
        Map.entry("пон", "Понеділок"),
        Map.entry("понеділок", "Понеділок"),

        Map.entry("вт", "Вівторок"),
        Map.entry("вів", "Вівторок"),
        Map.entry("вівторок", "Вівторок"),

        Map.entry("ср", "Середа"),
        Map.entry("середа", "Середа"),

        Map.entry("чт", "Четвер"),
        Map.entry("чет", "Четвер"),
        Map.entry("четвер", "Четвер"),

        Map.entry("пт", "П'ятниця"),
        Map.entry("пят", "П'ятниця"),
        Map.entry("п'ятниця", "П'ятниця"),

        Map.entry("сб", "Субота"),
        Map.entry("субота", "Субота")
    );
    private String normalizeDay(String input) {
        if (input == null) return "Невідомо";

        String key = input.trim().toLowerCase().replace("'", "");

        return DAY_NORMALIZATION.getOrDefault(key, capitalize(input.trim()));
    }

    private String capitalize(String str) {
        if (str.isEmpty()) return str;
        return str.substring(0, 1).toUpperCase() +
str.substring(1).toLowerCase();
    }

    private SchedulRepository schedulRepository;
    private ScheduleGenerationService scheduleGenerationService;
    private SchoolClassRepository schoolClassRepository;
    private SubjectRepository subjectRepository;
    private TeacherRepository teacherRepository;

    @GetMapping("/returns")
    public String homeS() {
        return "index";
    }

    @GetMapping("/schedul")
    public String showSchedul(Model model) {
        List<Schedul> scheduls = schedulRepository.findAll();
        model.addAttribute("schedul", scheduls);

        model.addAttribute("classes", schoolClassRepository.findAll());
        model.addAttribute("subjects", subjectRepository.findAll());
        model.addAttribute("teachers", teacherRepository.findAll());

        return "schedul";
    }
}

```

```

@PostMapping("/addschedul")
public String addSchedul(@RequestParam String day,
                        @RequestParam int lessonNum,
                        @RequestParam Long classId,
                        @RequestParam Long subjectId,
                        @RequestParam Long teacherId) {

    String normalizedDay = normalizeDay(day);

    Schedul schedul = new Schedul();
    schedul.setDay(normalizedDay);
    schedul.setLessonNum(lessonNum);

    schedul.setClassEntity(schoolClassRepository.findById(classId).orElseThrow());

    schedul.setDiscipline(subjectRepository.findById(subjectId).orElseThrow());
    schedul.setTeacher(teacherRepository.findById(teacherId).orElseThrow());

    schedulRepository.save(schedul);
    return "redirect:/schedule/class/" + classId;
}

@GetMapping("/delete_schedul")
public String deleteSchedul(@RequestParam Long id, @RequestParam Long
classId) {
    schedulRepository.deleteById(Math.toIntExact(id));
    return "redirect:/schedule/class/" + classId;
}

@GetMapping("/schedul_edit")
public String schedulEdit(@RequestParam Long id, Model model) {
    Optional<Schedul> optionalSchedul =
    schedulRepository.findById(Math.toIntExact(id));
    if (optionalSchedul.isEmpty()) {
        return "redirect:/schedul";
    }
    model.addAttribute("schedul", optionalSchedul.get());
    model.addAttribute("classes", schoolClassRepository.findAll());
    model.addAttribute("subjects", subjectRepository.findAll());
    model.addAttribute("teachers", teacherRepository.findAll());
    return "schedul_edit";
}

@PostMapping("/update_schedul")
public String schedulUpdate(@RequestParam Long id,
                            @RequestParam String day,
                            @RequestParam int lessonNum,
                            @RequestParam Long classId,
                            @RequestParam Long subjectId,
                            @RequestParam Long teacherId) {
    Optional<Schedul> optionalSchedul =
    schedulRepository.findById(Math.toIntExact(id));
    optionalSchedul.ifPresent(s -> {
        s.setDay(normalizeDay(day));
        s.setLessonNum(lessonNum);

    s.setClassEntity(schoolClassRepository.findById(classId).orElseThrow());

    s.setDiscipline(subjectRepository.findById(subjectId).orElseThrow());
        s.setTeacher(teacherRepository.findById(teacherId).orElseThrow());
    });
}

```

```

        schedulRepository.save(s);
    });
    return "redirect:/schedule/class/" +
schoolClassRepository.findById(classId).get().getId();
}

@PostMapping("/generate_schedule")
public String generateSchedule() {
    scheduleGenerationService.generate();
    return "redirect:/schedul";
}

@GetMapping("/schedule/classes")
public String getClassList(Model model) {
    List<SchoolClass> classes = schoolClassRepository.findAll();
    model.addAttribute("classes", classes);
    return "schedul";
}

@GetMapping("/schedule/classes/student")
public String getClassListStudent(Model model) {
    List<SchoolClass> classes = schoolClassRepository.findAll();
    model.addAttribute("classes", classes);
    return "student";
}

@GetMapping("/schedule/class/{id}")
public String getSchedulForClass(@PathVariable Long id, Model model) {
    List<Schedul> schedulList =
schoolRepository.findByIdEntityIdOrderByDayAscLessonNumAsc(id);
    SchoolClass schoolClass =
schoolClassRepository.findById(id).orElse(null);

    List<String> daysOrder = List.of("Понеділок", "Вівторок", "Середа",
"Четвер", "П'ятниця");
    Map<String, List<Schedul>> scheduleByDay = new LinkedHashMap<>();

    for (String day : daysOrder) {
        scheduleByDay.put(day, new ArrayList<>());
    }

    for (Schedul s : schedulList) {
        scheduleByDay.get(s.getDay()).add(s);
    }

    for (List<Schedul> daySchedule : scheduleByDay.values()) {
        daySchedule.sort(Comparator.comparingInt(Schedul::getLessonNum));
    }

    model.addAttribute("scheduleByDay", scheduleByDay);
    model.addAttribute("className", schoolClass != null ?
schoolClass.getName() : "Невідомий клас");
    model.addAttribute("classes", schoolClassRepository.findAll());
    model.addAttribute("subjects", subjectRepository.findAll());
    model.addAttribute("teachers", teacherRepository.findAll());

    return "class_schedule";
}

@GetMapping("/schedule/class/student/{id}")
public String getSchedulForClassStudent(@PathVariable Long id, Model model)
{

```

```

        List<Schedul> schedulList =
        schedulRepository.findByClassEntityIdOrderByDayAscLessonNumAsc(id);
        SchoolClass schoolClass =
        schoolClassRepository.findById(id).orElse(null);

        List<String> daysOrder = List.of("Понеділок", "Вівторок", "Середа",
        "Четвер", "П'ятниця");
        Map<String, List<Schedul>> scheduleByDay = new LinkedHashMap<>();

        for (String day : daysOrder) {
            scheduleByDay.put(day, new ArrayList<>());
        }

        for (Schedul s : schedulList) {
            scheduleByDay.get(s.getDay()).add(s);
        }

        for (List<Schedul> daySchedule : scheduleByDay.values()) {
            daySchedule.sort(Comparator.comparingInt(Schedul::getLessonNum));
        }

        model.addAttribute("scheduleByDay", scheduleByDay);
        model.addAttribute("className", schoolClass != null ?
        schoolClass.getName() : "Невідомий клас");
        model.addAttribute("classes", schoolClassRepository.findAll());
        model.addAttribute("subjects", subjectRepository.findAll());
        model.addAttribute("teachers", teacherRepository.findAll());

        return "student_class_schedule";
    }

    @GetMapping("/user_student")
    public String studentPage() {
        return "user_student";
    }

    @GetMapping("/user_teacher")
    public String teacherPage() {
        return "user_teacher";
    }

    @GetMapping("/user_dispatcher")
    public String dispatcherPage() {
        return "dispatcher";
    }
}

```

Лістинг 3

TeacherScheduleController.java

Цей код відповідає за створення та показ розкладу для викладачів

```

package com.example.scheduleproj.controllers;

import com.example.scheduleproj.ScheduleGenerationService;
import com.example.scheduleproj.dao.SchedulRepository;
import com.example.scheduleproj.dao.SubjectRepository;
import com.example.scheduleproj.dao.TeacherRepository;
import com.example.scheduleproj.dao.SchoolClassRepository;
import com.example.scheduleproj.entities.Schedul;

```

```

import com.example.scheduleproj.entities.Teacher;
import lombok.AllArgsConstructor;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import java.util.*;

@Controller
@AllArgsConstructor
public class TeacherScheduleController {

    private SchedulRepository schedulRepository;
    private ScheduleGenerationService scheduleGenerationService;
    private SchoolClassRepository schoolClassRepository;
    private SubjectRepository subjectRepository;
    private TeacherRepository teacherRepository;

    @GetMapping("/schedule/teachers")
    public String getTeacherList(Model model) {
        List<Teacher> teachers = teacherRepository.findAll();
        model.addAttribute("teachers", teachers);
        return "teachers";
    }

    @GetMapping("/schedule/teacher/{id}")
    public String getScheduleForTeacher(@PathVariable Long id, Model model) {
        List<Schedul> schedulList =
        schedulRepository.findByTeacherIdOrderByDayAscLessonNumAsc(id);
        Teacher teacher = teacherRepository.findById(id).orElse(null);

        List<String> daysOrder = List.of("Понеділок", "Вівторок", "Середа",
        "Четвер", "П'ятниця");
        Map<String, List<Schedul>> scheduleByDay = new LinkedHashMap<>();

        for (String day : daysOrder) {
            scheduleByDay.put(day, new ArrayList<>());
        }

        for (Schedul s : schedulList) {
            scheduleByDay.get(s.getDay()).add(s);
        }

        for (List<Schedul> daySchedule : scheduleByDay.values()) {
            daySchedule.sort(Comparator.comparingInt(Schedul::getLessonNum));
        }

        model.addAttribute("scheduleByDay", scheduleByDay);
        model.addAttribute("teacherName", teacher != null ? teacher.getName() :
        "Невідомий викладач");
        model.addAttribute("teachers", teacherRepository.findAll());
        model.addAttribute("subjects", subjectRepository.findAll());
        model.addAttribute("classes", schoolClassRepository.findAll());

        return "teacher_schedule";
    }
}

```

Лістинг 4

SchedulRepository.java

```

package com.example.scheduleproj.dao;

import com.example.scheduleproj.entities.Schedul;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface SchedulRepository extends JpaRepository<Schedul, Integer> {
    List<Schedul> findByClassEntityIdOrderByDayAscLessonNumAsc(Long id);

    List<Schedul> findByTeacherIdOrderByDayAscLessonNumAsc(Long id);
}

```

Лістинг 5

SchoolClassRepository.java

```

package com.example.scheduleproj.dao;
import com.example.scheduleproj.entities.SchoolClass;
import org.springframework.data.jpa.repository.JpaRepository;
public interface SchoolClassRepository extends JpaRepository<SchoolClass, Long>
{
}

```

Лістинг 6

SubjectRepository.java

```

package com.example.scheduleproj.dao;
import com.example.scheduleproj.entities.Subject;
import org.springframework.data.jpa.repository.JpaRepository;
public interface SubjectRepository extends JpaRepository<Subject, Long> {
}

```

Лістинг 7

TeacherRepository.java

```

package com.example.scheduleproj.dao;
import com.example.scheduleproj.entities.Teacher;
import org.springframework.data.jpa.repository.JpaRepository;
public interface TeacherRepository extends JpaRepository<Teacher, Long> {
}

```

Лістинг 8

TeacherSubjectClassRepository.java

```

package com.example.scheduleproj.dao;

import com.example.scheduleproj.entities.TeacherSubjectClass;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface TeacherSubjectClassRepository extends
JpaRepository<TeacherSubjectClass, Long> {
}

```

```
List<TeacherSubjectClass> findAll();
}
```

Лістинг 9

Schedul.java

Цей код представляє таблицю schedule з бази даних

```
package com.example.scheduleproj.entities;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;
@Getter
@Setter
@Entity
@Table(name = "schedul")
public class Schedul {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String day;

    private int lessonNum;

    @ManyToOne
    @JoinColumn(name = "class_id", nullable = false)
    private SchoolClass classEntity;

    @ManyToOne
    @JoinColumn(name = "subject_id", nullable = false)
    private Subject discipline;

    @ManyToOne
    @JoinColumn(name = "teacher_id", nullable = false)
    private Teacher teacher;
}
```

Лістинг 10

SchoolClass.java

Цей код представляє таблицю school_class з бази даних

```
package com.example.scheduleproj.entities;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;

import java.util.LinkedHashSet;
import java.util.Set;

@Getter
@Setter
```

```

@Entity
public class SchoolClass {

    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

}

```

Лістинг 11

Subject.java

Цей код представляє таблицю subject з бази даних

```

package com.example.scheduleproj.entities;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;

import java.util.LinkedHashSet;
import java.util.Set;

@Getter
@Setter
@Entity
public class Subject {

    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;

}

```

Лістинг 12

Teacher.java

Цей код представляє таблицю teacher з бази даних

```

package com.example.scheduleproj.entities;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;

import java.util.LinkedHashSet;
import java.util.Set;

@Getter
@Setter
@Entity
public class Teacher {

    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

}

```

```
private String name;
}
```

Лістинг 13

TeacherSubjectClass.java

Цей код представляє таблицю teacher_subject_class з бази даних

```
package com.example.scheduleproj.entities;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;

import java.util.LinkedHashSet;
import java.util.Set;

@Getter
@Setter
@Entity
public class TeacherSubjectClass {

    @Id @GeneratedValue
    private Long id;

    @ManyToOne
    private Teacher teacher;

    @ManyToOne
    private Subject subject;

    @ManyToOne
    private SchoolClass classEntity;

    private int hours;
}
```

Лістинг 14

ScheduleGenerationService.java

Цей код відповідає за генерацію розкладу занять

```
package com.example.scheduleproj;

import com.example.scheduleproj.dao.SchedulRepository;
import com.example.scheduleproj.entities.Schedul;
import com.example.scheduleproj.entities.SchoolClass;
import com.example.scheduleproj.entities.TeacherSubjectClass;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import com.example.scheduleproj.dao.TeacherSubjectClassRepository;
import com.example.scheduleproj.dao.SchoolClassRepository;

import java.util.*;

@Service
```



```

        <td th:text="${lesson.lessonNum}"></td>
        <td th:text="${lesson.discipline.name}"></td>
        <td th:text="${lesson.teacher.name}"></td>
        <td>
            <form th:action="@{/schedule_edit}" method="get">
                <input type="hidden" name="id" th:value="${lesson.id}">
                <button type="submit" class="btn btn-warning btn-
sm">Редагувати</button>
            </form>
        </td>
        <td>
            <form th:action="@{/delete_schedule}" method="get">
                <input type="hidden" name="id" th:value="${lesson.id}">
                <input type="hidden" name="classId"
th:value="${lesson.classEntity.id}">
                <button type="submit" class="btn btn-danger btn-
sm">Видалити</button>
            </form>
        </td>
    </tr>
</tbody>
</table>
</div>

<form action="/addschedule" method="post" class="row g-2 mt-4">
    <div class="col-md">
        <input type="text" name="day" placeholder="День (наприклад,
Понеділок)" required class="form-control">
    </div>
    <div class="col-md">
        <input type="number" name="lessonNum" placeholder="№ уроку" min="1"
max="8" required class="form-control">
    </div>
    <div class="col-md">
        <select name="classId" required class="form-select">
            <option th:each="c : ${classes}" th:value="${c.id}"
th:text="${c.name}"></option>
        </select>
    </div>
    <div class="col-md">
        <select name="subjectId" required class="form-select">
            <option th:each="s : ${subjects}" th:value="${s.id}"
th:text="${s.name}"></option>
        </select>
    </div>
    <div class="col-md">
        <select name="teacherId" required class="form-select">
            <option th:each="t : ${teachers}" th:value="${t.id}"
th:text="${t.name}"></option>
        </select>
    </div>
    <div class="col-md-auto">
        <input type="submit" value="Додати до розкладу" class="btn btn-
success">
    </div>
</form>

<div class="text-end mt-4">
    <a th:href="@{/schedule/classes}" class="btn btn-primary">Назад до
списку класів</a>
</div>
</div>

<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>

```

```
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>
```

Лістинг 16

dispatcher.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Розклад</title>
  <link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
  <link th:rel="stylesheet" type="text/css" th:href="styles.css">
</head>
<style>
  body {
    background-image: url(https://images.prom.ua/3955300103_w640_h640_oboi-
vinilovye-na.jpg);
  }
</style>
<body>
<nav class="mb-4">
  <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>
</nav>

<a th:href="@{/schedule/classes}" class="btn btn-success" >Список класів</a>

<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>
```

Лістинг 17

index.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Головна</title>
  <link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
  <link th:rel="stylesheet" type="text/css" th:href="styles.css">

</head>
<style>
  body {
    background-image: url(https://images.prom.ua/3955300103_w640_h640_oboi-
vinilovye-na.jpg);
  }
</style>
```

```

<body>
<h1>Оберіть обліковий запис</h1>
<a th:href="@{/user_student}" class="btn btn-success">Студент</a>
<a th:href="@{/user_teacher}" class="btn btn-success">Викладач</a>
<a th:href="@{/user_dispatcher}" class="btn btn-success" >Диспетчер</a>

<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>

```

Лістинг 18

schedul.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Розклад</title>
<link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
<link th:rel="stylesheet" type="text/css" th:href="styles.css">
<style>
    body {
        background-image:
url(https://images.prom.ua/3955300103_w640_h640_oboi-vinilovye-na.jpg);
    }
</style>
</head>
<body>
<div class="container mt-5">

    <nav class="mb-4">
        <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>
    </nav>

    <h2 class="text-center text-white">Оберіть клас</h2>
    <ul class="list-group mt-4">
        <li th:each="cl : ${classes}" class="list-group-item">
            <a th:href="@{/schedule/class/{id} (id=${cl.id})}"
th:text="${cl.name}">Class Name</a>
        </li>
    </ul>

    <form th:action="@{/generate_schedule}" method="post" class="mt-4 text-
center">
        <button type="submit" class="btn btn-success">Згенерувати
розклад</button>
    </form>
</div>

<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>

```

Лістинг 19

schedul_edit.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Розклад</title>
  <link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
  <link th:rel="stylesheet" type="text/css" th:href="styles.css">
</head>
<body style="background-image:
url('https://images.prom.ua/3955300103_w640_h640_oboi-vinilovye-na.jpg');">
<div class="container mt-4">

  <nav class="mb-4">
    <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>
  </nav>

  <h2 class="text-white text-center bg-dark p-2 rounded">Редагувати запис
розкладу</h2>
  <form th:action="@{/update_schedul}" method="post" class="bg-light p-4
rounded shadow">
    <input type="hidden" name="id" th:value="${schedul.id}">

    <div class="mb-3">
      <label for="day" class="form-label">День</label>
      <input type="text" id="day" name="day" th:value="${schedul.day}"
class="form-control" required>
    </div>

    <div class="mb-3">
      <label for="lessonNum" class="form-label">Номер уроку</label>
      <input type="number" id="lessonNum" name="lessonNum" min="1" max="8"
th:value="${schedul.lessonNum}" class="form-control" required>
    </div>

    <div class="mb-3">
      <label for="classId" class="form-label">Клас</label>
      <select id="classId" name="classId" class="form-select" required>
        <option th:each="c : ${classes}" th:value="${c.id}"
th:selected="${c.id} == ${schedul.classEntity.id}" th:text="${c.name}"></option>
      </select>
    </div>

    <div class="mb-3">
      <label for="subjectId" class="form-label">Предмет</label>
      <select id="subjectId" name="subjectId" class="form-select"
required>
        <option th:each="s : ${subjects}" th:value="${s.id}"
th:selected="${s.id} == ${schedul.discipline.id}" th:text="${s.name}"></option>
      </select>
    </div>

    <div class="mb-3">
      <label for="teacherId" class="form-label">Викладач</label>
      <select id="teacherId" name="teacherId" class="form-select"
required>
        <option th:each="t : ${teachers}" th:value="${t.id}"
th:selected="${t.id} == ${schedul.teacher.id}" th:text="${t.name}"></option>
      </select>

```

```

</div>

<button type="submit" class="btn btn-primary">Зберіти зміни</button>
<a th:href="@{'/schedule/class/' + ${schedule.classEntity.id}}"
class="btn btn-secondary ms-2">Скасувати</a>
</form>
</div>

<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>

```

Лістинг 20

student_class_schedule.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title th:text="'Розклад для ' + ${className}">Розклад</title>
<link th:rel="stylesheet"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
</head>
<style>
body {
background-image: url(https://images.prom.ua/3955300103_w640_h640_oboi-
vinilovye-na.jpg);
}
</style>
<body>
<div class="container mt-4">

<nav class="mb-4">
<a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>
</nav>

<h2 th:text="'Розклад для класу ' + ${className}" class="text-center text-
white bg-dark p-2 rounded"></h2>

<div th:each="entry : ${scheduleByDay}">
<h4 th:text="${entry.key}" class="mt-4 text-primary"></h4>
<table class="table table-bordered table-striped text-center">
<thead class="table-dark">
<tr>
<th>Урок №</th>
<th>Предмет</th>
<th>Викладач</th>
</tr>
</thead>
<tbody>
<tr th:each="lesson : ${entry.value}">
<td th:text="${lesson.lessonNum}"></td>
<td th:text="${lesson.discipline.name}"></td>
<td th:text="${lesson.teacher.name}"></td>
</tr>
</tbody>

</table>

```

```

</div>

<a th:href="@{/schedule/classes/student}" class="btn btn-primary mt-4">Назад
до списку класів</a>
</div>
<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>

```

Лістинг 21

teacher_schedule.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<title>Розклад</title>
<link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
<link th:rel="stylesheet" type="text/css" th:href="styles.css">
<style>
    body {
        background-image:
url(https://images.prom.ua/3955300103_w640_h640_oboi-vinilovye-na.jpg);
    }
</style>
</head>
<body>
<div class="container mt-4">

    <nav class="mb-4">
        <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>
    </nav>

    <h2 th:text="'Розклад викладача: ' + ${teacherName}" class="text-center
text-primary"></h2>

    <div th:each="entry : ${scheduleByDay}">
        <h4 th:text="${entry.key}" class="mt-4 text-secondary"></h4>
        <table class="table table-bordered table-striped text-center">
            <thead class="table-dark">
                <tr>
                    <th>№ Уроку</th>
                    <th>Предмет</th>
                    <th>Клас</th>
                </tr>
            </thead>
            <tbody>
                <tr th:each="lesson : ${entry.value}">
                    <td th:text="${lesson.lessonNum}"></td>
                    <td th:text="${lesson.discipline.name}"></td>
                    <td th:text="${lesson.classEntity.name}"></td>
                </tr>
            </tbody>
        </table>
    </div>

    <a th:href="@{/schedule/teachers}" class="btn btn-secondary mt-4">Назад до
списку викладачів</a>
</div>
<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>

```

```
<script th:src="@{/webjars/popperjs_core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>
```

Лістинг 22

teachers.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Розклад</title>
  <link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
  <link th:rel="stylesheet" type="text/css" th:href="styles.css">
  <style>
    body {
      background-image:
url(https://images.prom.ua/3955300103_w640_h640_oboi-vinilovye-na.jpg);
    }
  </style>
</head>
<body>
<div class="container mt-4">

  <nav class="mb-4">
    <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>
  </nav>
  <h2 class="text-center text-primary">Список викладачів</h2>

  <ul class="list-group">
    <li th:each="teacher : ${teachers}" class="list-group-item d-flex
justify-content-between align-items-center">
      <span th:text="${teacher.name}"></span>
      <a th:href="@{/schedule/teacher/{id}(id=${teacher.id})}" class="btn
btn-primary btn-sm">Переглянути розклад</a>
    </li>
  </ul>
</div>
<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs_core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>
```

Лістинг 23

user_student.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Розклад</title>
  <link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
  <link th:rel="stylesheet" type="text/css" th:href="styles.css">
</head>
```

```

<style>
  body {
    background-image: url(https://images.prom.ua/3955300103_w640_h640_oboi-
vinilovye-na.jpg);
  }
</style>
<body>
<nav class="mb-4">
  <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>

<a th:href="@{/schedule/classes/student}" class="btn btn-success" >Список
класів</a>

<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-
alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>

```

Лістинг 24

user_teacher.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Розклад</title>
  <link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
  <link th:rel="stylesheet" type="text/css" th:href="styles.css">
</head>
<style>
  body {
    background-image: url(https://images.prom.ua/3955300103_w640_h640_oboi-
vinilovye-na.jpg);
  }
</style>
<body>
<nav class="mb-4">
  <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>

<a th:href="@{/schedule/teachers}" class="btn btn-success" >Список
викладачів</a>

<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-
alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>

```

Лістинг 25

student.html

```

<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Розклад</title>

```

```

    <link th:rel="stylesheet" type="text/css"
th:href="@{/webjars/bootstrap/5.2.3/css/bootstrap.min.css}">
    <link th:rel="stylesheet" type="text/css" th:href="styles.css">
</head>
<style>
    body {
        background-image: url(https://images.prom.ua/3955300103_w640_h640_oboi-
vinilovye-na.jpg);
    }
</style>
<body>
<nav class="mb-4">
    <a class="btn btn-danger" th:href="@{/returnS}">Вийти</a>
</nav>

<div class="container mt-5">
    <h2 class="text-center">Оберіть клас</h2>
    <ul class="list-group mt-4">
        <li th:each="cl : ${classes}" class="list-group-item">
            <a th:href="@{/schedule/class/student/{id}(id=${cl.id})}"
th:text="${cl.name}">Class Name</a>
        </li>
    </ul>
</div>
<script th:src="@{/webjars/jquery/3.6.4/dist/jquery.js}"></script>
<script th:src="@{/webjars/popperjs__core/2.11.7/dist/umd/popper.js}"></script>
<script th:src="@{/webjars/bootstrap/5.3.0-alpha2/dist/js/bootstrap.bundle.js}"></script>
</body>
</html>

```

Лістинг 26

styles.css

```

.clr{
    color: darkorange;
}

.headW {
    background-color: darkred;
    overflow: hidden;
}

.headW a {
    float: left;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    font-size: 17px;
}

.headW a:hover {
    background-color: greenyellow;
    color: darkorange;
}

.headW a.active {
    background-color: darkviolet;
    color: yellow;
}

```

ДОДАТОК В – ОПИС ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

1 Загальні відомості

- Позначення: ScheduleApp-1.0
- Назва: Вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею
- Необхідне ПЗ: ОС Windows 10+, Java
- Мова програмування: Java, HTML/CSS

2 Функціональне призначення

- Додаток призначений для автоматизованого додавання, редагування, видалення, генерації та перегляду розкладу занять для Березнегуватського ліцею.
- Функціональні обмеження:
 - Відсутність підтримки мобільного інтерфейсу
 - Обмежена кількість одночасних користувачів

3 Опис логічної структури

- Алгоритм:

Користувач вводить дані через вебінтерфейс → контролер обробляє запит → дані зберігаються/зчитуються з бази → оновлений розклад відображається
- Методи:
 - Операції додавання, перегляду, редагування, видалення розкладу
 - Алгоритм генерації розкладу з урахуванням вільного часу викладачів
- Структура програми:
 - ScheduleProjApplication.java — основний сервер
 - controllers/ — логіка розкладу
 - entities/ — сутності БД (викладачі, предмети, розклад, класи)
 - dao/ — репозиторії (викладачі, предмети, розклад, класи)

- templates/ — HTML-шаблони
- static/ — CSS

4 Необхідні технічні засоби

- ОС: Windows 10+
- Процесор: від 2 ядер
- RAM: мінімум 4 ГБ
- ПЗ: браузер Chrome/Firefox, Java середовище

5 Виклик і завантаження

- Запуск через файл ScheduleProjApplication.java
- Доступ до вебінтерфейсу здійснюється через браузер:

<http://localhost:8080>

6 Вхідні дані

- Дані про викладачів, класи, предмети
- Формат: форма HTML (текстові поля, списки)
- Кодування: UTF-8

7 Вихідні дані

- Збережений у базі даних розклад
- Відображення у вигляді таблиці на вебсторінці

ДОДАТОК Г – ІНСТРУКЦІЯ ПРОГРАМІСТУ, ОПЕРАТОРУ Й КОРИСТУВАЧЕВІ ДО ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

1 Призначення програми

Функціональне призначення: Програма призначена для автоматизації процесів створення, додавання, редагування, видалення та перегляду розкладу занять у навчальному закладі. Забезпечує зручний інтерфейс для диспетчера, викладача та учня.

Експлуатаційне призначення: Використовується у навчальних закладах для централізованого керування розкладом. Доступ до функцій залежить від ролі користувача.

Склад функцій:

- Авторизація користувачів
- Додавання/редагування/видалення розкладу
- Генерація нового розкладу
- Перегляд розкладу

2 Умови виконання програми

Мінімальний склад апаратних засобів:

- ПК з ОС Windows 10
- CPU: від 2 ядер
- RAM: 4 ГБ

Мінімальний склад програмних засобів:

- Java
- Google Chrome
- XAMPP

Вимоги до персоналу:

- Базове вміння працювати з комп'ютером
- Розуміння структури розкладу навчального процесу

- Знання основ адміністрування (для диспетчера)

3 Виконання програми

Завантаження та запуск програми:

- 1) Запустити XAMPP і в ньому запустити Apache та MySQL
- 2) Відкрити директорію з програмою.
- 3) Запустити cmd та ввести команду `mvn spring-boot:run`.
- 4) Запустити браузер та ввести посилання `http://localhost:8080/`
- 5) З'являється головне вікно авторизації.

Завантаження та запуск програми представлений на рисунках Г.1 – Г.3.

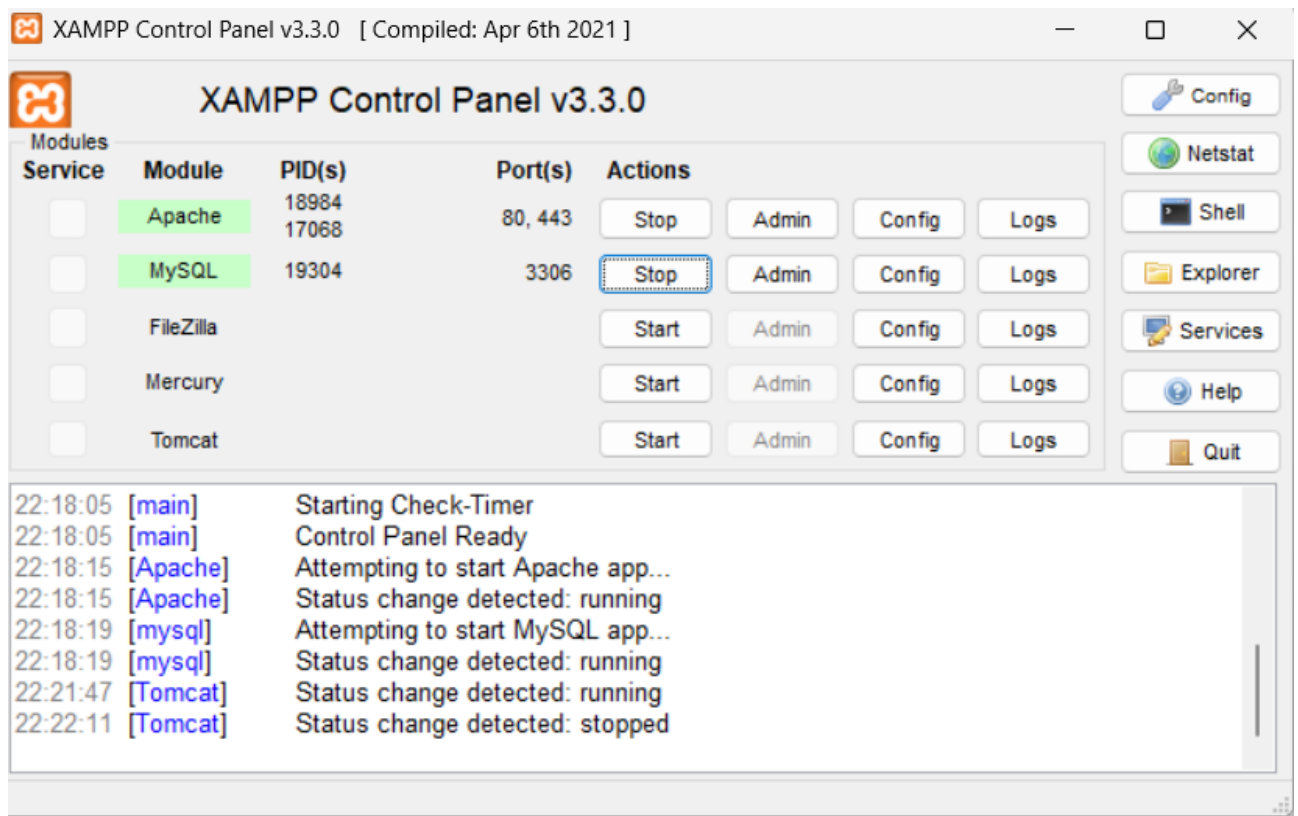
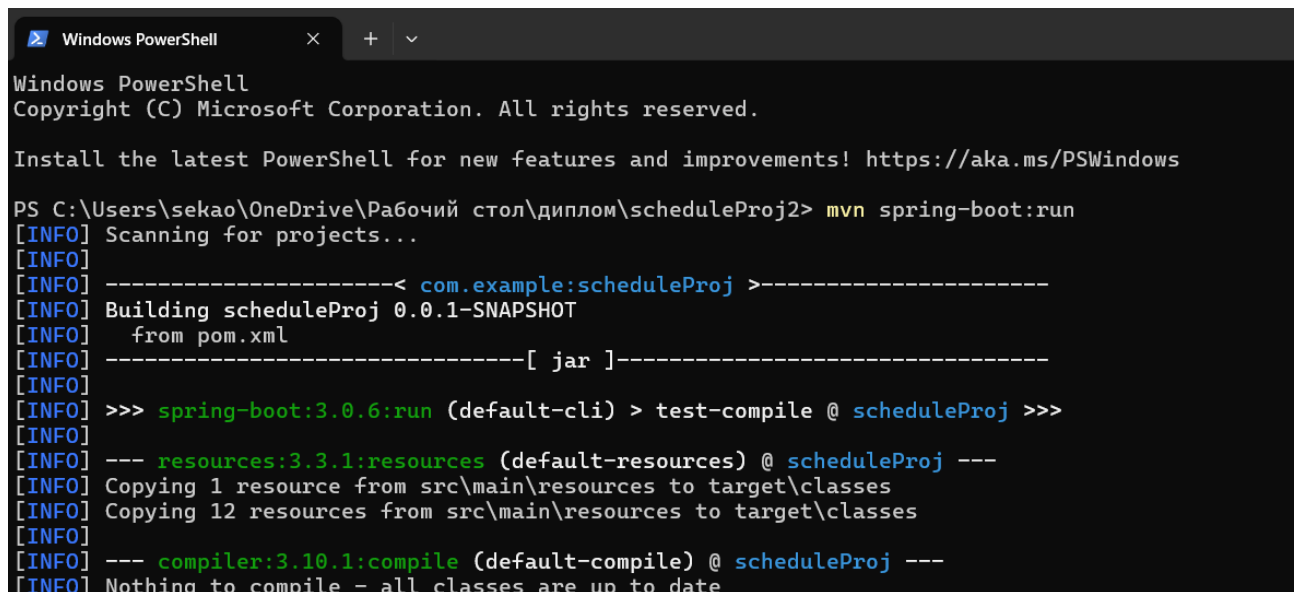


Рисунок Г.1 – Запуск XAMPP



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sekao\OneDrive\Рабочий стол\диплом\scheduleProj2> mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:scheduleProj >-----
[INFO] Building scheduleProj 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot:3.0.6:run (default-cli) > test-compile @ scheduleProj >>>
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ scheduleProj ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 12 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.10.1:compile (default-compile) @ scheduleProj ---
[INFO] Nothing to compile - all classes are up to date

```

Рисунок Г.2 – Запуск вебзастосунку через cmd

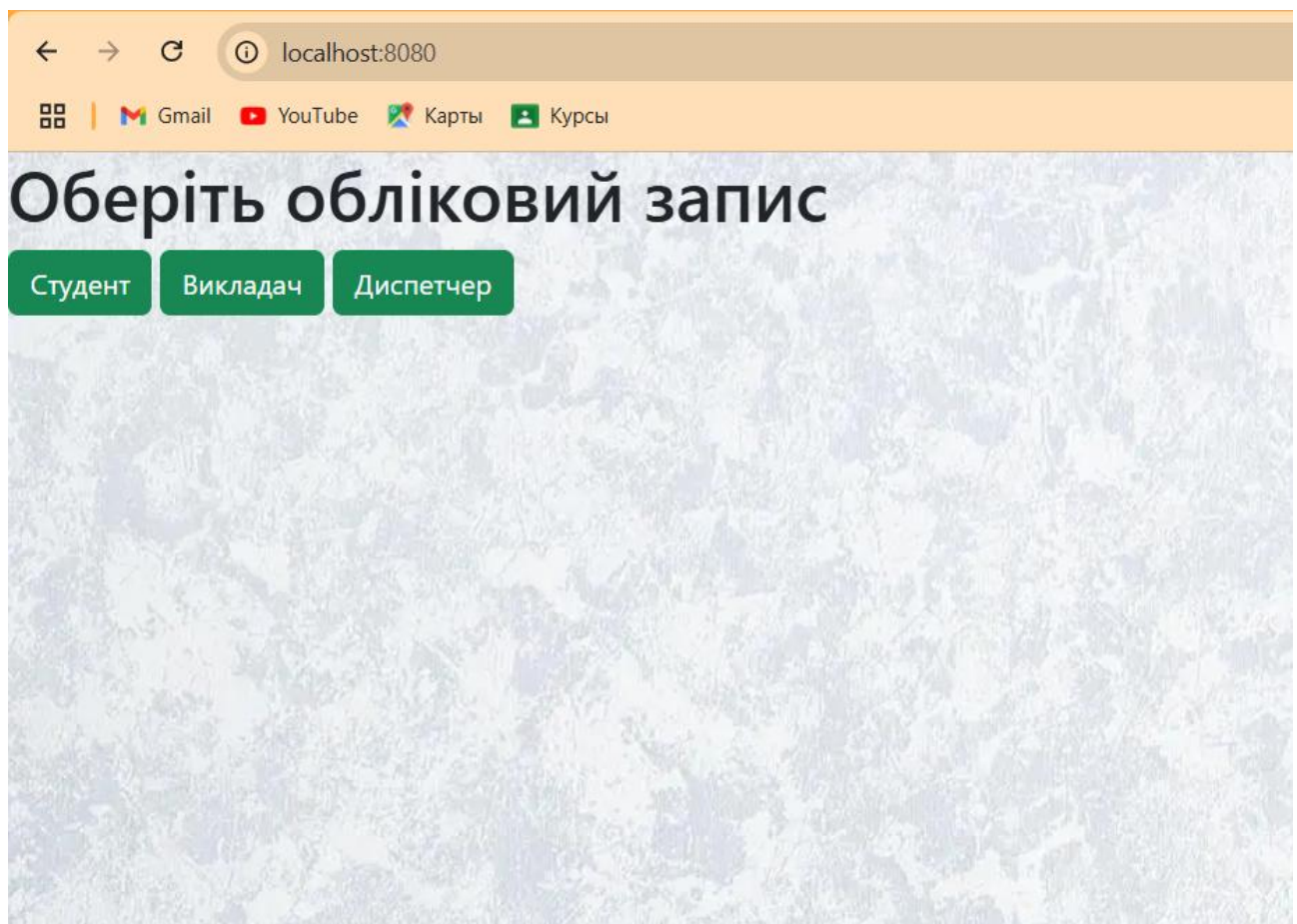


Рисунок Г.3 – Сторінка з обліковими записами в браузері

Виконання програми:

- Після входу відкривається головне меню.

- Диспетчер може:
 - Генерувати новий розклад
 - Додавати розклад
 - Редагувати розклад
 - Видаляти розклад
 - Переглядати розклад
- Викладач може переглядати свій розклад
- Учень переглядає розклад занять

Виконання програми представлене на рисунках Г.4 – Г.20.

Form fields and buttons:

- Day: пт
- Lesson Number: 2
- Class: 5A
- Subject: Українська мова
- Teacher: Авраменко Л.А.
- Buttons: Додати до розкладу, Назад до списку класів

Рисунок Г.4 – Вхідні дані для додавання розкладу для диспетчера

П'ятниця

Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
2	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок Г.5 – Результат додавання розкладу для диспетчера

Form fields and buttons:

- Day: П'ятниця
- Lesson Number: 3
- Class: 5A
- Subject: Українська мова
- Teacher: Авраменко Л.А.
- Buttons: Зберегти зміни, Скасувати

Рисунок Г.6 – Редагування доданого розкладу для диспетчера

П'ятниця

Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
3	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок Г.7 – Результат редагування розкладу для диспетчера

П'ятниця

Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок Г.8 – Видалення редагованого розкладу для диспетчера

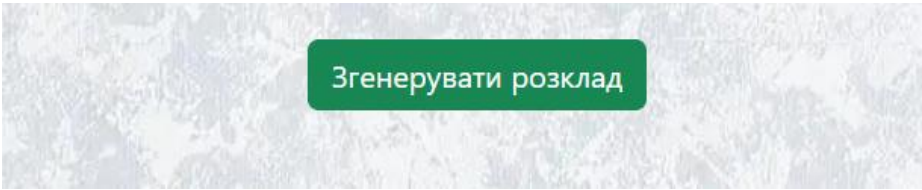


Рисунок Г.9 – Кнопка для створення розкладу для диспетчера

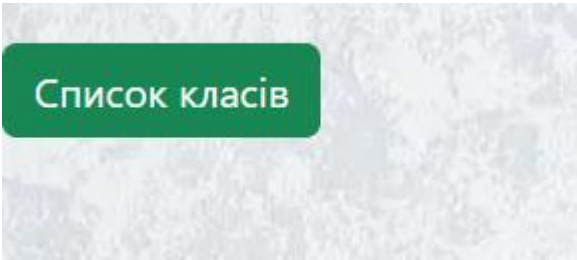


Рисунок Г.10 – Кнопка для перегляду списку класів для диспетчера та учня

Оберть клас

5А
5Б
6А
6Б
7А
7Б
8А
8Б
9А
9Б

Рисунок Г.11 – Список класів для диспетчера та учня

Розклад для класу 5А				
Понеділок				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
2	Англійська мова	Ткачук О.М.	Редагувати	Видалити
3	Історія	Скуратовська Н.Г.	Редагувати	Видалити
Вівторок				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Англійська мова	Ткачук О.М.	Редагувати	Видалити
2	Українська мова	Авраменко Л.А.	Редагувати	Видалити
3	Історія	Скуратовська Н.Г.	Редагувати	Видалити

Рисунок Г.12 – Створений розклад занять для 5А класу для диспетчера

Середа				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
2	Англійська мова	Ткачук О.М.	Редагувати	Видалити
Четвер				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити
П'ятниця				
Урок №	Предмет	Викладач	Редагування	Видалення
1	Українська мова	Авраменко Л.А.	Редагувати	Видалити

Рисунок Г.13 – Створений розклад занять для 5А класу для диспетчера

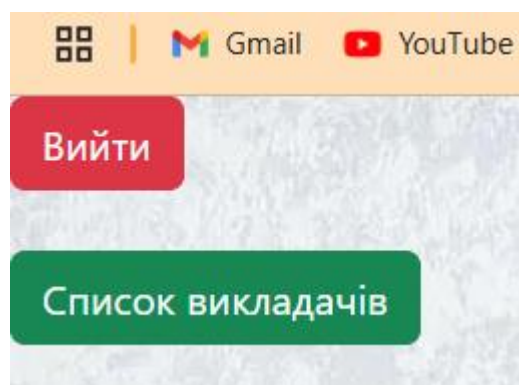


Рисунок Г.14 – Кнопка список викладачів для викладачів

Вийти

Список викладачів

Авраменко Л.А.	Переглянути розклад
Стащенко Т.С.	Переглянути розклад
Троян О.П.	Переглянути розклад
Ткачук О.М.	Переглянути розклад
Зайкіна І.М.	Переглянути розклад
Шандренко О.В.	Переглянути розклад
Скуратовська Н.Г.	Переглянути розклад
Коваленко Н.В.	Переглянути розклад
Коваль Л.В.	Переглянути розклад
Коваль Ю.О.	Переглянути розклад

Рисунок Г.15 – Список викладачів для викладачів

Вийти

Розклад викладача: Авраменко Л.А.

Понеділок

№ Уроку	Предмет	Клас
1	Українська мова	5А
2	Українська мова	8А
3	Українська мова	8Б
4	Українська мова	9А
5	Українська мова	9Б

Вівторок

№ Уроку	Предмет	Клас
1	Українська мова	8А
2	Українська мова	5А
3	Українська мова	8Б
4	Українська мова	9А

Рисунок Г.16 – Розклад обраного викладача для викладачів

5	Українська мова	9Б
Середа		
№ Уроку	Предмет	Клас
1	Українська мова	5А
2	Українська мова	8А
3	Українська мова	8Б
4	Українська мова	9А
5	Українська мова	9Б
Четвер		
№ Уроку	Предмет	Клас
1	Українська мова	5А
2	Українська мова	8А
3	Українська мова	8Б
4	Українська мова	9А
5	Українська мова	9Б

Рисунок Г.17 – Розклад обраного викладача для викладачів

П'ятниця		
№ Уроку	Предмет	Клас
1	Українська мова	5А
Назад до списку викладачів		

Рисунок Г.18 – Розклад обраного викладача для викладачів

Вийти		
Розклад для класу 6А		
Понеділок		
Урок №	Предмет	Викладач
1	Українська мова	Стащенко Т.С.
3	Англійська мова	Ткачук О.М.
Вівторок		
Урок №	Предмет	Викладач
1	Українська мова	Стащенко Т.С.
3	Англійська мова	Ткачук О.М.
Середа		
Урок №	Предмет	Викладач
1	Українська мова	Стащенко Т.С.
3	Англійська мова	Ткачук О.М.

Рисунок Г.19 – Розклад обраного класу для учнів

Четвер		
Урок №	Предмет	Викладач
2	Українська мова	Стащенко Т.С.
П'ятниця		
Урок №	Предмет	Викладач
2	Українська мова	Стащенко Т.С.
Назад до списку класів		

Рисунок Г.20 – Розклад обраного класу для учнів

Завершення роботи програми:

- Натискання кнопки "Вийти" та закриття вкладки у браузері (для користувачів)
- Для повного завершення роботи потрібно просто закрити cmd і програма автоматично завершить роботу, при тому всі зміни будуть збережені у базі даних

ДОДАТОК Д – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ ДЛЯ БЕРЕЗНЕГУВАТСЬКОГО ЛІЦЕЮ

1 Об'єкт випробувань

- Найменування програми: Вебзастосунок для автоматизації складання розкладу занять для Березнегуватського ліцею
- Область застосування: Автоматизація додавання, редагування, видалення, генерації та перегляду розкладу занять для Березнегуватського ліцею.
- Позначення: ScheduleApp-1.0

2 Мета випробувань

Метою випробувань є перевірка відповідності функціональних можливостей вебзастосунку для автоматизації складання розкладу занять для Березнегуватського ліцею вимогам технічного завдання.

3 Вимоги до програми

Під час випробувань перевіряються наступні вимоги:

- Додавання, редагування, видалення розкладу
- Генерація нового розкладу
- Відображення актуального розкладу у вебінтерфейсі користувача
- Захист від введення некоректних даних

4 Вимоги до програмної документації

- Склад документації:
 - 1) Технічне завдання
 - 2) Вихідні тексти програмних модулів
 - 3) Опис програмного забезпечення
 - 4) Інструкція користувача
 - 5) Програма та методика випробувань

5 Засоби і порядок випробувань

- Технічні засоби:
 - 1) ПК з ОС Windows 10+
 - 2) Оперативна пам'ять: від 4 ГБ
 - 3) Процесор: від 2 ядер
- Програмні засоби:
 - 1) Java
 - 2) Браузер (Google Chrome/Firefox)
- Порядок випробувань:
 - 1) Запуск програми на тестовому ПК
 - 2) Перевірка функцій додавання, редагування, видалення та генерації розкладу
 - 3) Порівняння результатів із очікуваними згідно ТЗ

6 Методи випробувань

Методом випробувань розробленого вебзастосунку є поетапна перевірка кожної функції вебзастосунку під час роботи з ним.

Перелік випробувань наведений у таблиці Д.1.

Таблиця Д.1 – Перелік випробувань

Назва випробування	Вхідні дані	Очікуваний результат
Додавання розкладу	День(Пт), урок(2), клас(5А), предмет(Українська мова), викладач(Авраменко Л.А.)	Успішне додавання розкладу
Редагування розкладу	Id(438), урок(3)	Успішне редагування розкладу
Видалення розкладу	Id(438)	Успішне видалення розкладу
Створення розкладу	Натиснути на кнопку “Згенерувати розклад”	Успішне створення розкладу
Перегляд розкладу	Натиснути на кнопку “Список класів”, та обираємо клас “5А”	Успішний перегляд розкладу для класу “5А”