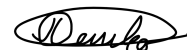


**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

**Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:
«РОЗРОБКА ШКІЛЬНОГО ЕЛЕКТРОННОГО ЖУРНАЛУ
ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ASP.NET WEBAPI, ANGULAR»**

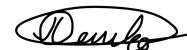
Виконав студент 4-го курсу
Сергій ПЕШКО



Науковий керівник:
Олексій ТКАЧЕНКО

Засвідчую, що в цій дипломній роботі немає запозичень
з праць інших авторів відповідних посилань.

Студент



Роботу розглянуто й допущено до захисту
на засіданні кафедри теорії та технології програмування

«__» _____ 2023 р.,

протокол №

Завідувач кафедри

М. С. НІКІТЧЕНКО

РЕФЕРАТ

Обсяг роботи 56 сторінок, 28 ілюстрацій, 15 таблиць, 24 джерела посилань.

ANGULAR, C#, DOCKER, HIGHCHARTS, HTTP-ЗАПИТ, JWT, REST, ВЕБ-ДОДАТОК, ВЧИТЕЛЬ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, ОСВІТА ТРИРІВНЕВА АРХІТЕКТУРА, УЧЕНЬ, ШКІЛЬНИЙ ЕЛЕКТРОННИЙ ЖУРНАЛ.

Об'єктом дослідження є облік успішності учнів та адміністрування навчального процесу. Предметом дослідження є інформаційні системи автоматизації обліку успішності учнів та адміністрування навчального процесу.

Метою даної роботи є дослідження шкільних електронних журналів наявних на ринку та розробка власного електронного журналу.

Для досягнення поставленої мети було визначено наступні задачі: вивчення існуючих на ринку систем, що містять електронний журнал, визначення основних переваг та недоліків цих систем, дослідження потенційних вдосконалень, розробка власного електронного журналу.

Методи дослідження включають поєднання якісних і кількісних підходів, таких як метод експертних оцінок, моделювання, аналіз, метод ООП та тестування інтерфейсу і зручності використання. Інструменти розробки: Microsoft Visual Studio – інтегроване середовище розробки (IDE) мовою програмування C#, яке є безкоштовним.

В результаті дослідження було виконано загальний огляд електронних засобів навчання, проаналізовано переваги та недоліки шкільних журналів присутніх на ринку та розроблену свій продукт, який являє собою шкільний електронний журнал. Актуальність даної роботи впливає з потреби у підвищенні ефективності роботи вчителів та учнів використовуючи цифрові інструменти, зокрема електронний журнал.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	4
ВСТУП	5
РОЗДІЛ 1 ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ	7
РОЗДІЛ 2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	11
2.1 Призначення системи	11
2.2 Цілі створення системи	12
2.3 Вимоги до функцій, які виконуються системою	13
РОЗДІЛ 3 ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ТА АРХІТЕКТУРИ	20
3.1 Опис використаних технологій	20
3.2 Огляд архітектури системи	22
РОЗДІЛ 4 РЕАЛІЗАЦІЯ СИСТЕМИ	24
4.1 Організація бази даних	24
4.2 Програмна реалізація системи	29
РОЗДІЛ 5 РОБОТА З СИСТЕМОЮ “ЕЛЕКТРОННИЙ ЖУРНАЛ”	33
ВИСНОВКИ	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	49
ДОДАТОК А Структура файлів системи	52
ДОДАТОК Б Діаграми класів серверної частини	54

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API – application programming interface, прикладний програмний інтерфейс;

CRUD – create, read, update, delete, 4 основні функції управління даними;

CSS – cascading style sheets, мова стилю сторінок;

DI – dependency injection, впровадження залежностей;

HTML – hypertext markup language, мова гіпертекстової розмітки;

HTTP – hypertext transfer protocol, протокол передачі даних;

IDE – integrated development environment, інтегроване середовище розробки;

JWT – json web token, стандарт токена доступу на основі JSON;

ORM – object-relational mapping, об'єктно-реляційне відображення;

REST – representational state transfer, підхід до архітектури мережевих протоколів;

SASS – syntactically awesome style sheets, мова сценаріїв CSS препроцесора;

SOLID – аббревіатура складена з перших літер п'яти базових принципів об'єктно-орієнтованого програмування та дизайну;

SPA – single-page application, односторінковий додаток;

SQL – structured query language, структурована мова запитів;

ОС – операційна система;

ПЗ – програмне забезпечення;

СКБД – система керування базами даних;

ООП – об'єктно-орієнтоване програмування.

ВСТУП

Оцінка сучасного стану об'єкта розробки. Розвиток технологій у 21 столітті радикально змінює повсякденне життя людей, включаючи сферу освіти. Сьогодні в школах використовується все більше і більше електронних інструментів і шкільний електронний журнал є одним з головних таких інструментів. Розроблений додаток полегшує співпрацю між вчителями та учнями, надаючи централізований інструмент для відслідковування учнів, вчителів, оцінок, пропусків та надавання різної аналітики по учнях.

Актуальність роботи та підстави для її виконання. Актуальність даної роботи впливає з підвищення ефективності роботи вчителів та учнів шляхом автоматизації обліку оцінок використовуючи цифрові інструменти, зокрема електронний журнал. Потреба в зручних та універсальних інструментах стає все більш виразною, оскільки система освіти розвивається в епоху цифровізації. Крім того, що вдосконалення дизайну та функціональності електронного журналу сприяє підвищенню якості освіти, воно також зменшує розрив у комунікації між вчителями та учнями забезпечуючи плідну співпрацю між ними.

Мета й завдання роботи. Метою даної роботи є дослідження шкільних електронних журналів наявних на ринку та розробка власного електронного журналу. Для досягнення цієї мети поставлено такі завдання:

- вивчення існуючих на ринку систем, що містять електронний журнал;
- визначення основних переваг та недоліків цих систем;
- дослідження потенційних удосконалень;
- розробка власного електронного журналу;

Об'єкт, методи й засоби розроблення. Об'єктом дослідження є облік успішності учнів та адміністрування навчального процесу. Предметом

дослідження є інформаційні системи автоматизації обліку успішності учнів та адміністрування навчального процесу. Методи розроблення включають поєднання якісних і кількісних підходів, таких як метод експертних оцінок, моделювання, аналіз, методи ООП та тестування інтерфейсу і зручності використання.

В якості інструменту створення програмного засобу було обрано Microsoft Visual Studio [1] – інтегроване середовище розробки (IDE) мовою програмування C# [2], яке є безкоштовним.

C# - сучасна, об'єктно-орієнтована мова програмування. C# дозволяє розробникам створювати багато типів безпечних і надійних програм, які працюють на платформі .NET [3].

Для розробки клієнтської частини було використано фреймворк Angular [4]. Angular – це безкоштовна платформа для розробки веб-додатків з відкритим вихідним кодом на основі TypeScript. Фреймворк розроблений компанією Google.

Можливі сфери застосування. Потенційні сфери застосування виходять за межі випадку електронного журналу. Відомості, отримані в результаті даної роботи, можуть стати основою для розробки інших веб-додатків у сфері освіти і зробити внесок у ширшу сферу EdTech [5]. Крім того, запропонований додаток для школи можна підлаштувати під університети.

Взаємозв'язок з іншими роботами. За методами розробки, дана робота виконувалася сумісно з іншими дослідженнями в галузі освіти та комп'ютерних наук.

1 ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ

На даний час на ринку існує багато систем які містять в собі шкільний електронний журнал для автоматизації обліку успішності учнів.

1. Edmodo [6]. Edmodo – це комплексний інструмент, який пропонує широкий набір функцій як для вчителів і учнів так і для батьків. Вчителі мають можливість виставляти оцінки учням, призначати їм домашнє завдання. Всі користувачі можуть обмінюватися повідомленнями між собою. Учні мають можливість прикріпляти виконані завдання та брати участь в обговореннях під завданнями. Батькам доступний функціонал перегляду успішності своїх дітей.

Переваги: безкоштовне користування, додаток надає широкий функціонал комунікації сприяючи співпраці учнів та їх спільному навчанню, наявна можливість створення та проведення тестових робіт.

Недоліки: Edmodo не має функції адміністративного керування, що робить його менш ефективним у завданнях загального управління шкільним процесом.

2. Google Classroom [7]. Google Classroom є частиною пакету Google for Education та дуже поширеною платформою, яка дозволяє вчителям створювати, розповсюджувати та оцінювати завдання, а також комунікувати з учнями.

Переваги: сильна сторона Google Classroom полягає в його інтеграціям з іншими системами Google, такими як Google Docs, Google Sheets, Google Slides чи Google Drive, що робить його універсальним інструментом для управління освітнім процесом, дана система є безкоштовною. Ще однією з переваг є можливість додавання інших вчителів до курсу.

Недоліки: у Google Classroom відсутні деякі функції, такі як детальна аналітика даних чи автоматичне створення звітів. Його залежність від екосистеми Google може бути недоліком для користувачів, які надають перевагу іншим платформам або для тих, кому є важливою конфіденційність даних. Ще одним з недоліків є те, що при великій кількості студентів та завдань, на гугл диску може закінчитись пам'ять, а його розширення є платним.

3. Blackboard [8]. Blackboard – це комплексна система управління освітнім процесом, яка успішно показала себе у багатьох навчальних закладах. Ця система надає інструменти для керування предметами, оцінками, дошкою обговорень та віртуальними класами.

Переваги: однією з сильних сторін даної системи є її широкий функціонал і здатність швидко опрацьовувати велику кількість даних. Додаток є дуже адаптованим до дистанційного навчання, що в наші дні є величезною перевагою.

Недоліки: одним з мінусів є вартість системи Blackboard, яка є набагато дорожчим рішенням ніж всі вищезгадані, що робить його менш доступним для деяких державних шкіл з обмеженим бюджетом.

4. Schoology [9]. Schoology – це ще одна комплексна система управління навчальним процесом, яка пропонує широкий спектр функцій, таких як управління предметами, управління оцінками, аналітика даних та комунікація між користувачами.

Переваги: Schoology має широкий спектр інтеграцій з іншими сервісами. Крім того ця система володіє потужними аналітичними функціями, які обробляють велику кількість даних та формують різного роду звіти по них.

Недоліки: мобільний додаток Schoology має набагато меншу кількість функцій ніж веб версія, тобто є дуже урізаним. Крім того синхронізація між веб-версією та мобільною версією додатка відбувається рідко, що ускладнює його використання.

5. School Today [24]. School Today – ще одна система управління навчальним процесом, яка була розроблена в Україні. Вона набула широкого використання завдяки своїй комплексній функціональності.

Переваги: однією з переваг даної системи є її глибина та різноманітність налаштувань. Вчителі мають можливість адаптувати шкалу оцінювання відповідно до своїх вподобань. Крім того система має адаптивний дизайн, що дозволяє її використовувати з будь-якого пристрою. Також School Today підключається про безпеку даних користувачів. Для цього система використовує вдосконалені методи шифрування.

Недоліки: система налаштована під українську систему освіти, що ускладнює її використання в іноземних навчальних закладах. Повільна швидкість оновлення інформації в мобільному додатку. Ще одним з недоліків неможливість для батьків з одного облікового запису слідкувати за декількома дітьми, які навчаються в різних навчальних закладах.

Таблиця 1. Переваги та недоліки деяких існуючих рішень

Назва	Переваги	Недоліки
Edmodo	Безкоштовна система. Широкий функціонал комунікації між учнями та вчителями.	Відсутні функції адміністративного керування.
Google Classroom	Інтеграція з багатьма системами Google. Можливість додавання інших вчителів до курсу.	Відсутність аналітики даних. Залежність від екосистеми Google. Після закінчення місця на гугл диску, його розширення є платним.
Blackboard	Широкий функціонал. Висока швидкість опрацювання даних. Адаптованість до дистанційного навчання.	Платне рішення, яке може собі дозволити не кожна державна школа.
Schoology	Широкий спектр інтеграції з іншими сервісами. Потужні аналітичні функції.	Неадаптований дизайн під мобільні платформи. Синхронізація між веб-версією та мобільним додатком повільна.
School Today	Різноманітність налаштувань. Адаптованість під різні пристрої. Вдосконалені методи шифрування.	Налаштована лише під українську систему освіти. Батьки можуть слідкувати з одного облікового запису лише за дітьми однієї школи.

Підсумовуючи, хоча розглянуті системи володіють широким спектром функцій та зробили значний внесок в цифрову освіту, все ще є місця для вдосконалень. Шкільний журнал розроблений у цій роботі намагається взяти найкраще від конкурентів та усунути їх недоліки.

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Шкільний електронний журнал має декілька призначень, які націлені на покращення якості навчання. Основним призначенням є створення централізованої, ефективної та інтерактивної платформи для обробки та збереження інформації про учнів та вчителів.

Однією з головних цілей є полегшення моніторингу успішності учнів. Електронний журнал дозволяє вчителям ставити оцінки учням, відмічати відвідуваність та виконувати багато інших операцій. Такий моніторинг має важливе значення, оскільки дозволяє виявляти учнів, які мають труднощі в навчанні та потребують підтримки, а також виявляти тих, хто має високу успішність та можуть вирішувати складніші задачі.

Система також виступає мостом для комунікації між школою та батьками надаючи останнім доступ до записів успішності їхніх дітей. Ця прозора та відкрита лінія комунікації сприяє створенню більш сприятливого для співпраці освітнього середовища.

Крім того, електронний шкільний журнал оптимізує адміністративні процеси в школі. Завдяки автоматизації рутинних завдань, система значно

зменшує навантаження вчителів, дозволяючи їм приділяти більше уваги своєму основному обов'язку – навчанню учнів.

На додаток, шкільний електронний журнал сприяє розвитку цифрової грамотності у вчителів, учнів та батьків, що є важливим фактором у 21 столітті, коли технології все більше заповнюють наше життя.

2.2 Цілі створення системи

Кінцева мета розробленої системи – сприяти створенню ефективного, привабливого та дієвого освітнього середовища за допомогою використання технологій. Для досягнення цієї мети система переслідує декілька головних цілей.

Спрощення адміністративних процесів у школі. Система спрямована на спрощення та оптимізацію адміністративних процесів у школі. Система значно зменшує навантаження на працівників освітніх закладів надаючи адміністратору можливість керувати учнями, вчителями, класами та предметами в рамках однієї платформи.

Моніторинг успішності учнів. Іншою важливою ціллю системи є надання вчителям інструменту для ефективного управління своїми учнями. Додаток надає можливість легко ставити та відстежувати оцінки учнів. Автоматична генерація семестрових та річних оцінок економить вчителям неймовірну кількість часу, дозволяючи їм більше концентруватися на навчальному процесі. Крім того, система переслідує ціль надання вчителям інструментів аналітики, які мають на меті допомогти їм краще проаналізувати успішність учнів і приймати ефективні рішення.

Персоналізований досвід навчання. Надаючи учням доступ до їх академічних даних, система виховує в них почуття відповідальності за своє навчання та сприяє самовдосконаленню. Електронний журнал має на меті

надати учням зручні інструменти аналітики, які можуть допомогти їм відстежувати свою успішність та прогрес.

Прийняття рішень на основі аналізу даних. Ще однією ключовою ціллю є полегшення прийняття рішень на основі аналізу даних у школах. Надаючи комплексну аналітику навчального процесу, система має на меті надати цінну інформацію, яка може стати базою для планування навчального процесу. Система робить ці дані легко доступними та зрозумілими за допомогою візуальних діаграм та графіків.

Постійне вдосконалення та адаптація. Нарешті, система має на меті постійне вдосконалення та адаптацію, щоб задовольнити потреби користувачів та йти в ногу з прогресом. Постійні дослідження технологій в сфері освіти будуть вдосконалювати та розвивати систему.

2.3 Вимоги до функцій, які виконуються системою

Таблиця 2 відображає перелік функцій та задач, що підлягають автоматизації для підсистеми адміністратора.

Таблиця 2. Перелік функцій, задач підсистеми адміністратора

Функція	Задачі
Управління вчителями	Перегляд списку всіх вчителів
	Додавання вчителя
	Редагування вчителя
	Видалення вчителя
Управління учнями	Перегляд списку всіх учнів
	Додавання учня
	Редагування учня
	Видалення учня

Управління класами	Перегляд списку всіх класів
	Додавання класу
	Додавання учня до класу
	Видалення учня з класу
	Редагування класу
	Видалення класу
Управління предметами	Додавання предмету
	Видалення предмету
	Додавання вчителя до предмету
	Додавання предмету до класу
	Перегляд списку всіх предметів

Таблиця 3 відображає перелік функцій та задач, що підлягають автоматизації для підсистеми вчителя.

Таблиця 3. Перелік функцій, задач підсистеми вчителя

Функція	Задачі
Управління класами	Перегляд списку всіх класів, які навчає вчитель
	Перегляд списку предметів, які навчає вчитель для даного класу
Управління уроками	Перегляд списку розділів даного предмету в даному класі
	Перегляд списку тем в даному розділі

Управління оцінками	Перегляд оцінок учнів даного класу з даного предмету
	Виставлення оцінки учню
	Видалення оцінки учню
	Збереження змін
	Перегляд оцінок для конкретного розділу даного предмету
	Автоматична генерація оцінок за 1 семестр
	Автоматична генерація оцінок за 2 семестр
	Автоматична генерація річних оцінок
Експорт оцінок в Excel	
Перегляд аналітики	Перегляд кількості проведених уроків по даному предмету для даного класу
	Перегляд загальної кількості виставлених оцінок по заданому предмету для заданого класу
	Перегляд кількості контрольних робіт по заданому предмету для заданого класу
	Перегляд загального середнього балу по всіх оцінках по заданому предмету для заданого класу

	Перегляд графіку середнього балу по учнях по заданому предмету
	Перегляд графіку прогулів по учнях по заданому предмету
	Перегляд графіку зміни кількості прогулів за певний проміжок часу по заданому предмету
	Перегляд графіку розподілу оцінок по заданому предмету для заданого класу

Таблиця 4 відображає перелік функцій та задач, що підлягають автоматизації для підсистеми учня.

Таблиця 4. Перелік функцій, задач підсистеми учня

Функція	Задачі
Управління оцінками	Перегляд предметів, які викладаються учню
	Перегляд оцінок по вибраному предмету
	Експорт оцінок в Excel
Перегляд аналітики	Перегляд середнього балу по заданому предмету
	Перегляд кількості всіх балів по заданому предмету
	Перегляд кількості пропущених уроків по заданому предмету
	Перегляд кількості відвіданих уроків по заданому предмету
	Перегляд графіку зміни оцінок протягом часу для заданого предмету
	Перегляд графіку розподілу оцінок по заданому предмету
	Перегляд графіку розподілу відвідуваності по заданому предмету
	Перегляд графіку розподілу кількості оцінок по заданому предмету

На рисунку 1 показані Use Case діаграми [10] системи шкільного електронного журналу для адміністратора, вчителя та учня.

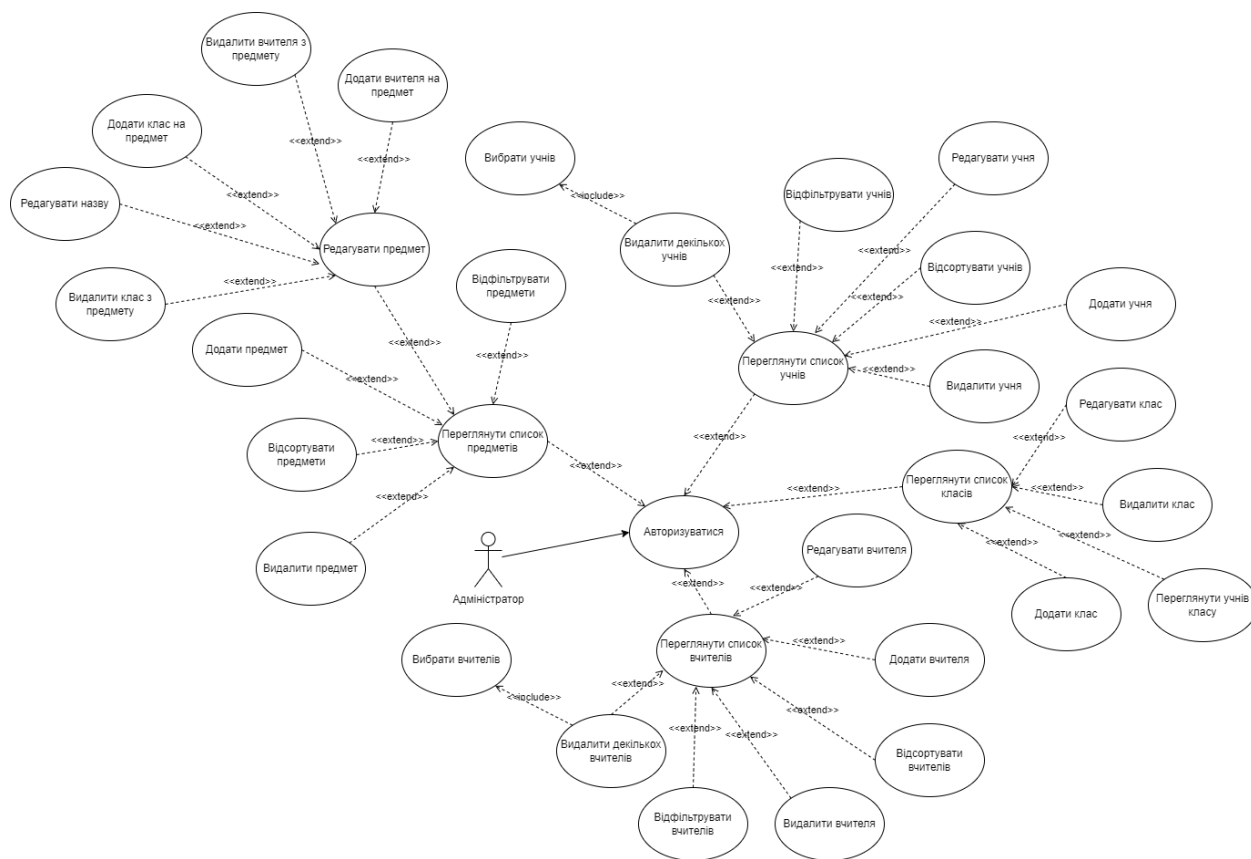


Рисунок 1 – Use Case діаграма: а – система адміністратора

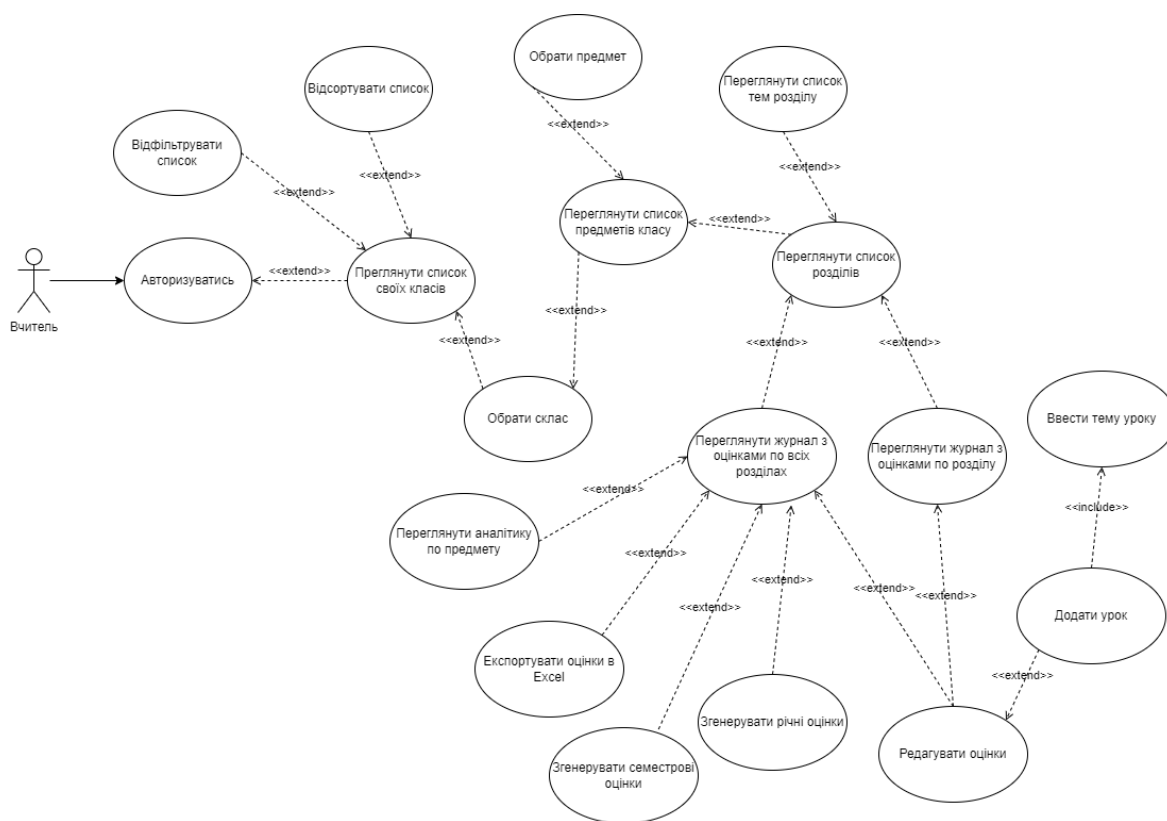


Рисунок 1 (продовження): б – система вчителя

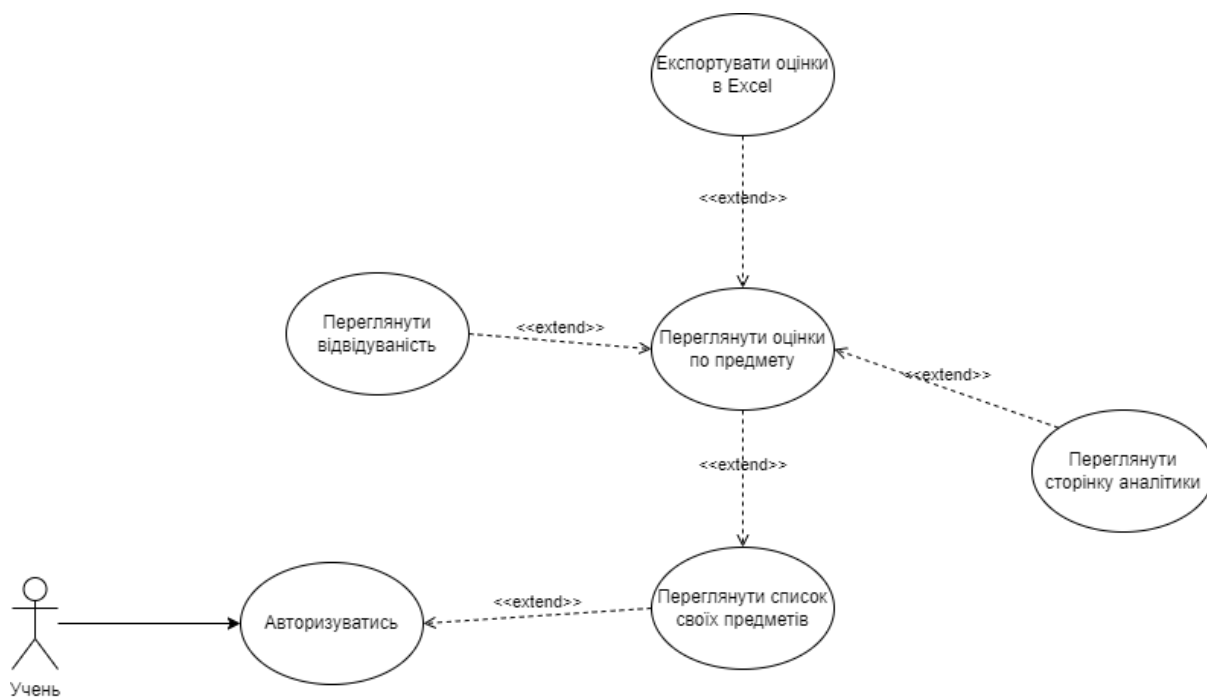


Рисунок 1 (продовження): в – система учня

3 ОПИС ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ТА АРХІТЕКТУРИ

3.1 Опис використаних технологій

При розробці даного електронного журналу використовувалась низка складних, ефективних та перевірених часом технологій, щоб гарантувати, що додаток буде надійним, безпечним, гнучким та масштабованим. В цьому підрозділі розглянуто кожен технологію, яка була використана при розробці системи.

C#. C# - це універсальна об'єктно-орієнтована мова програмування розроблена компанією Microsoft. Цілями розробки цієї мови були простота, сучасність та гнучкість у поєднанні з безпекою типів, високою ефективністю та продуктивністю. C# є строго типізованою мовою, що дозволяє запобігти багатьом проблемам пов'язаним з динамічними типами. Крім того, C# працює на платформі .NET, яка в свою чергу має величезну кількість бібліотек готових до використання. У створеній системі C# є мовою на якій написана вся серверна частина.

MS SQL Server [11]. MS SQL Server – це високоефективна та надійна система бази даних. Як і C#, вона розроблена компанією Microsoft та пропонує гнучке комплексне рішення для зберігання даних. У контексті розроблюваної системи MS SQL Server відіграє роль основної бази даних, яка зберігає всю важливу інформацію, від облікових записів користувачів до оцінок учнів.

Angular. Angular – це веб-фреймворк для створення веб-додатків, який розроблений компанією Google. Цей фреймворк відомий своєю високою швидкістю та продуктивністю, а також здатністю створювати односторінкові веб-додатки. Він є другим найпопулярнішим веб-фреймворком [12] після React. Angular використовує компонентно-орієнтовану архітектуру, що сприяє багаторазовому використанню коду та підвищує зручність обслуговування

кової бази. У розроблюваній системі Angular використовується для створення клієнтської частини.

Docker [13]. Docker – це платформа, яка використовує віртуалізацію на рівні ОС для запуску ПЗ в контейнерах. Контейнери є ізольованими один від одного та містять в собі бібліотеки, файли та залежності необхідні для запуску системи для того щоб гарантувати, що вона запуститься не залежно від середовища.

Highcharts [14]. Highcharts – сучасна бібліотека діаграм та графіків на основі SVG, яка відома своєю простотою, гнучкістю та легкістю використання. Вона написана на мові JavaScript і надає простий спосіб додавання інтерактивних графіків до веб-додатків. У розробленій системі Highcharts відіграє ключову роль на сторінках аналітики вчителів та учнів, надаючи візуально привабливі, інтерактивні діаграми та графіки.

Entity Framework [15]. Entity Framework – це ORM інструмент з відкритим кодом, який є обгорткою до ADO.NET. Він забезпечує більш простий спосіб роботи з даними автоматизуючи більшість процесів, таких як відслідковування сутностей, валідація даних чи написання SQL-запитів. В розробленому електронному журналі ця технологія відіграє важливу роль, спрощуючи роботу з даними та взаємодію з MS SQL Server.

Insomnia [16]. Insomnia – це багатofункціональний REST API-клієнт. Він спрощує процес розробки та тестування API, дозволяючи розробнику надсилати HTTP-запити, переглядати результати запитів, керувати файлами cookie, тощо. Під час розробки шкільного електронного журналу Insomnia широко використовувалась для тестування та налагоджування кінцевих точок API сервера.

3.2 Огляд архітектури системи

Для розробки системи шкільного електронного журналу було обрано клієнт-серверну архітектуру [17] (рисунок 2).

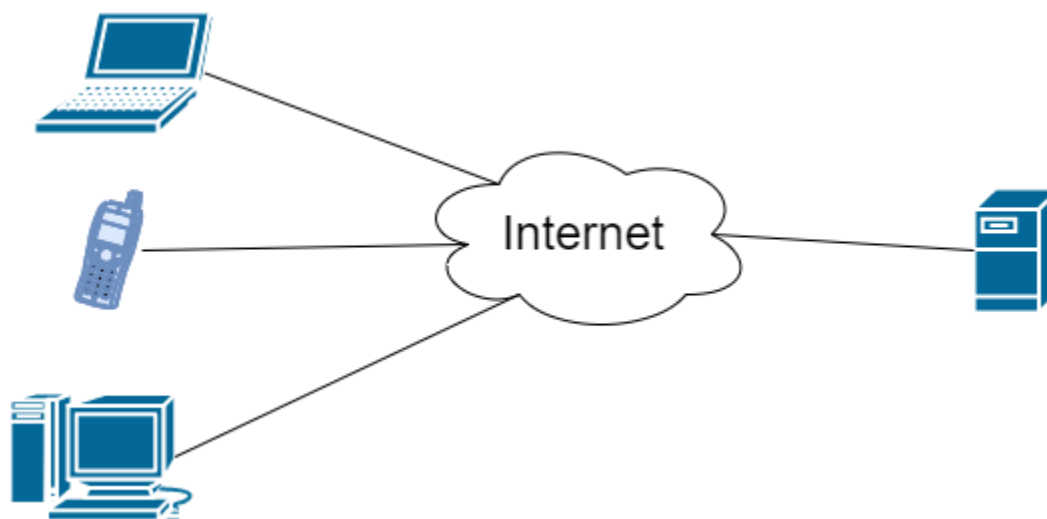


Рисунок 2 – Клієнт-серверна архітектура

В клієнт-серверній архітектурі система поділяється на дві частини: клієнт (мобільний телефон, комп'ютер, ноутбук, тощо), відповідальний за взаємодію з користувачем, і сервер, відповідальний за обробку запитів і управління даними та ресурсами.

В свою чергу для реалізації серверної частини була використана 3-рівнева архітектура [18] (рисунок 4). Ця архітектура розділяє систему на три взаємопов'язані рівні: презентаційний рівень, рівень бізнес-логіки та рівень доступу до даних.



Рисунок 4 – трирівнева архітектура

Презентаційний рівень це найвищий рівень, який керує циклом HTTP-запитів, їх обробкою та поверненням результату. Він є мостом між клієнтом та рівнем бізнес-логіки.

Рівень бізнес-логіки містить основну логіку програми. Він взаємодіє як з презентаційним рівнем, так і з рівнем доступу до даних. Даний рівень обробляє дані отримані від презентаційного рівня застосовуючи до них необхідну логіку та передає їх до рівня доступу до даних або ж повертає назад до рівня презентації.

Рівень доступу до даних взаємодіє з базою даних, в нашому випадку за допомогою Entity Framework. Він виконує всі необхідні операції пов'язані з базою даних, наприклад CRUD [19] операції. Оброблені дані він відправляє у базу даних або назад до рівня бізнес-логіки.

Модель клієнт-сервер у поєднанні з трирівневою архітектурою на стороні сервера сприяє ефективній розробці, підтримці кодової бази, масштабованості системи та безпеці. Ця комплексна архітектура разом із ретельним вибором технологій робить систему електронного журналу

надійною, стійкою та готовою задовольняти потреба сучасного освітнього процесу.

4 РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 Організація бази даних

Для зберігання даних в системі було використано СКБД MS SQL Server. Реляційна база даних містить 11 таблиць. Схема бази даних наведена на рисунку 5.

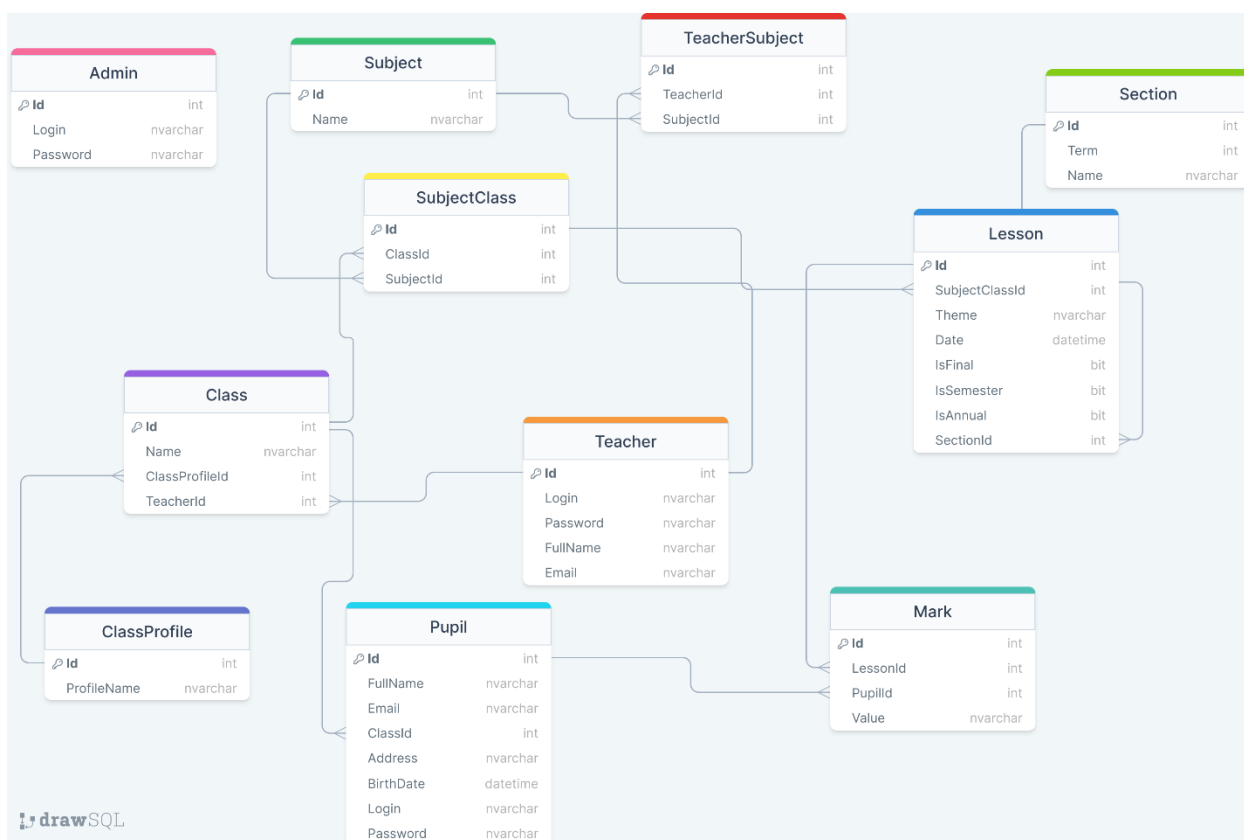


Рисунок 5 – Схема бази даних

Детальна інформація про поля таблиць наведена у таблицях 5-15.

Таблиця 5. Структура таблиці “Admin”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
Login	nvarchar	Логін
Password	nvarchar	Пароль

Таблиця 6. Структура таблиці “Class”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
Name	nvarchar	Назва класу
ClassProfileId	int	Ідентифікатор профілю до якого належить клас
TeacherId	int	Ідентифікатор вчителя, який є класним керівником класу

Таблиця 7. Структура таблиці “ClassProfile”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
ProfileName	nvarchar	Назва профілю

Таблиця 8. Структура таблиці “Subject”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
Name	nvarchar	Назва предмету

Таблиця 9. Структура таблиці “SubjectClass”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
ClassId	int	Ідентифікатор класу
SubjectId	int	Ідентифікатор предмету

Таблиця 10. Структура таблиці “Pupil”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
FullName	nvarchar	П.І.Б учня
Email	nvarchar	Електронна адреса
ClassId	int	Ідентифікатор класу
Address	nvarchar	Адреса проживання
BirthDate	datetime	Дата народження
Login	nvarchar	Логін
Password	nvarchar	Пароль

Таблиця 11. Структура таблиці “Teacher”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
Login	nvarchar	Логін
Password	nvarchar	Пароль
FullName	nvarchar	П.І.Б вчителя
Email	nvarchar	Електронна адреса

Таблиця 12. Структура таблиці “TeacherSubject”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
TeacherId	int	Ідентифікатор вчителя
SubjectId	int	Ідентифікатор предмету

Таблиця 13. Структура таблиці “Section”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
Term	int	Номер семестру в якому знаходиться цей розділ
Name	nvarchar	Назва розділу

Таблиця 14. Структура таблиці “Lesson”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
SubjectClassId	int	Ідентифікатор запису, що містить інформацію про клас та предмет де проводився цей урок
Theme	nvarchar	Назва теми уроку
Date	datetime	Дата проведення уроку
IsFinal	bit	Чи є урок контрольною
IsSemester	bit	Чи є запис семестровою
IsAnnual	bit	Чи є запис річною
SectionId	int	Ідентифікатор розділу в якому знаходиться даний урок

Таблиця 15. Структура таблиці “Mark”

Атрибут	Тип	Опис
Id	int	Унікальний ідентифікатор
LessonId	int	Ідентифікатор уроку
PupilId	int	Ідентифікатор учня
Value	nvarchar	Значення оцінки

4.2 Програмна реалізація системи

Під час процесу розробки електронного журналу було використано сучасні практики розробки програмного забезпечення. У цьому підрозділі наведено огляд основних кроків, здійснених під час розробки системи.

Аналіз і проектування. Розробка почалася з комплексного аналізу для розуміння цілей системи та очікувань від неї. Цей етап включав в себе визначення ключових функціональних можливостей, таких як керування класами, предметами, користувачами, запис і отримання оцінок, створення аналітичних графіків тощо. Далі було проектування архітектури системи. Як розказувалося раніше, система розроблялася з використанням клієнт-серверної архітектури із трирівневою архітектурою на стороні сервера.

Налаштування середовища розробки. В якості IDE було обрано Visual Studio 2022. Це IDE повністю підтримує створення веб-додатків за допомогою ASP.NET Core, що робить її ідеальним вибором для створюваної системи. Середовище Visual Studio включає в себе надійний редактор коду, вбудовані інструменти налагодження та повну інтеграцію з системою контролю версій Git, що значно полегшує процес розробки. Крім того, до середовища було додано Docker, що забезпечило правильну роботу системи незалежно від операційної системи чи апаратної складової. Також було використано такий інструмент як Insomnia для подальшого тестування API.

Розробка серверної частини. Сервер розроблено з використанням C# як мови програмування та ASP.NET Core як веб-фреймворку. Додаток Б містить діаграми класів серверної частини системи. Рівень доступу до даних містить в собі опис всіх сутностей і їх конфігурацій та слідує патерну проектування під назвою “Repository” [20]. Базуючись на цьому патерні, для кожної сутності було створено свій клас репозиторій, який відповідає за

реалізацію CRUD операцій (рисунок А.1). Крім того на даному рівні зберігаються всі міграції бази даних. Рівень бізнес-логіки містить в собі всі сервіси та валідатори. Сервіси відповідають за всю логіку системи, таку як створення сутностей та їх передача до рівня доступу до даних, валідація даних, формування звітів та багато іншого. Валідатори використовуються сервісами для того, щоб впевнитися в тому, що дані які передаються на рівень доступу до даних є коректними та не виникне ніяких виключень при обробці їх базою даних. Рисунок А.2 відображає структуру файлів рівня бізнес-логіки. Презентаційний рівень містить в собі контролери, які відповідають за обробку HTTP-запитів та конфігурацію серверу. Цей рівень використовує сервіси рівня бізнес-логіки, куди передає дані отримані від запиту. Даний рівень розроблювався з дотриманням принципів REST [21]. Ці принципи включають в себе наступні обмеження:

- клієнт-серверна архітектура. Це обмеження робить клієнт та сервер слабо зв'язаними, не надаючи кожному з них деталей реалізації іншого;
- відсутність стану. Це обмеження говорить про те, що сервер не повинен зберігати стан програми, а повинен отримувати всю необхідну інформацію від клієнтам разом із запитом;
- кешування. Це обмеження говорить про те, що сервер повинен підтримувати систему кешування. Це допомагає уникнути повторюваних запитів від клієнта;
- єдиний інтерфейс. Це обмеження вказує на те, що кожен ресурс на сервері повинен мати унікальну адресу та бути доступним через загальний інтерфейс. Для керування ресурсами клієнт має використовувати загальний набір основних методів (GET, POST, PUT, DELETE).

Більше того, на сервері широко використовувався принцип SOLID під назвою Dependency Inversion [22]. Для цього були використані вбудовані можливості фреймворку ASP.NET Core, який надає DI контейнери з цією метою. Даний принцип говорить про те, що модулі повинні залежати від абстракцій, а не від конкретних реалізацій. Це робить код більш масштабованим, модулі менш зв'язаними та дозволяє простіше вносити зміни в систему та підтримувати її. Використання абстракцій можна прослідкувати як на рисунку 6, так і на рисунку 7, де для кожного репозиторію та сервісу використовуються інтерфейси.

Для авторизації було використано технологію JWT [23]. Дана технологія працює таким чином, що при авторизації користувача, сервер перевіряє чи дійсно даний користувач існує в системі. Якщо користувач існує, то сервер генерує два токени: короткоживучий access токен та довгоживучий refresh токен і віддає їх до клієнта. Клієнт зберігає ці токени і включає їх до заголовків кожного HTTP-запиту. Якщо, access токен стає не валідним, то клієнт використовує refresh токен для того, щоб отримати новий токен доступу від сервера. На додаток токен доступу містить інформацію про роль користувача, що дозволяє надавати доступ лише до ресурсів дозволених даному користувачу. Структура JWT складається з трьох частин. Заголовок містить інформацію про тип токена та алгоритм, який використовується для підпису. Корисне навантаження містить зашифровану інформацію про користувача, наприклад його логін, роль, електронну пошту та інше. Підпис створюється за допомогою шифрування заголовку та корисного навантаження та їх об'єднання. Перевагою авторизації з використанням JWT є те, що в неї відсутній стан. Тобто серверу не потрібно пам'ятати інформацію про токени які він видав, оскільки кожен токен містить всю необхідну інформацію. Це

добре узгоджується з принципами REST, які були використані на серверній частині та описані вище.

Реалізація клієнтської частини. Клієнтську частину було розроблено з допомогою фреймворку Angular. Дана частина містить в собі 5 основних модулів: admin, authorization, pupil, teacher, shared (рисунок 8), кожен з яких містить набір компонентів, що відповідають за якусь логіку.

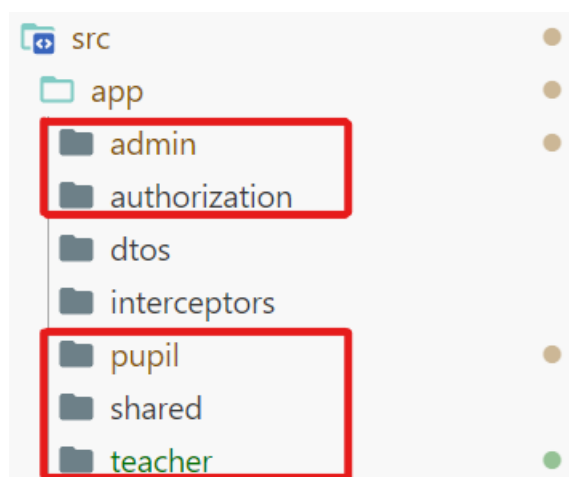


Рисунок 8 – Модулі клієнтської частини

Програма була розроблена як SPA, щоб забезпечити плавну роботу користувача з плавним переходом між різними маршрутами.

Кожна веб-сторінка програми була структурована за допомогою HTML, який створив скелет інтерфейсу користувача використовуючи різні HTML-елементи. HTML є стандартною мовою розмітки для документів призначених для відображення у веб-браузері.

SASS, сценарна мова препроцесора, яка компілюється в CSS, була використана для стилізації елементів HTML. Порівняно зі звичайним CSS, SASS надає більше контролю над стилями за допомогою функцій, змінних, вкладених класів та міксинів. Цей препроцесор надає більш зручну та

організовану структуру таблиці стилів та полегшує створення дизайну інтерфейсу користувача.

TypeScript – це строго типізована мова, розроблена компанією Microsoft. TypeScript є надбудовою над мовою JavaScript, та використовувався в розроблюваній системі як основна мова програмування клієнтської частини. Ця мова робить код більш надійним, легшим для змін, тестування та підтримки порівняно зі звичайним JavaScript.

Для реалізації функціоналу аналітичних звітів була використана бібліотека Highcharts. Вона надає функції аналітики та візуалізації даних у вигляді інтерактивних графіків та діаграм.

Клієнтська частина теж використовує Docker для відкидання залежностей від середовища виконання.

5 РОБОТА З СИСТЕМОЮ “ЕЛЕКТРОННИЙ ЖУРНАЛ”

Для входу в систему користувачу необхідно виконати авторизацію. Для даного кроку користувач повинен ввести свій логін та пароль у вікні зображеному на рисунку 9. Роль користувача система знайде автоматично на основі логіна та пароля.

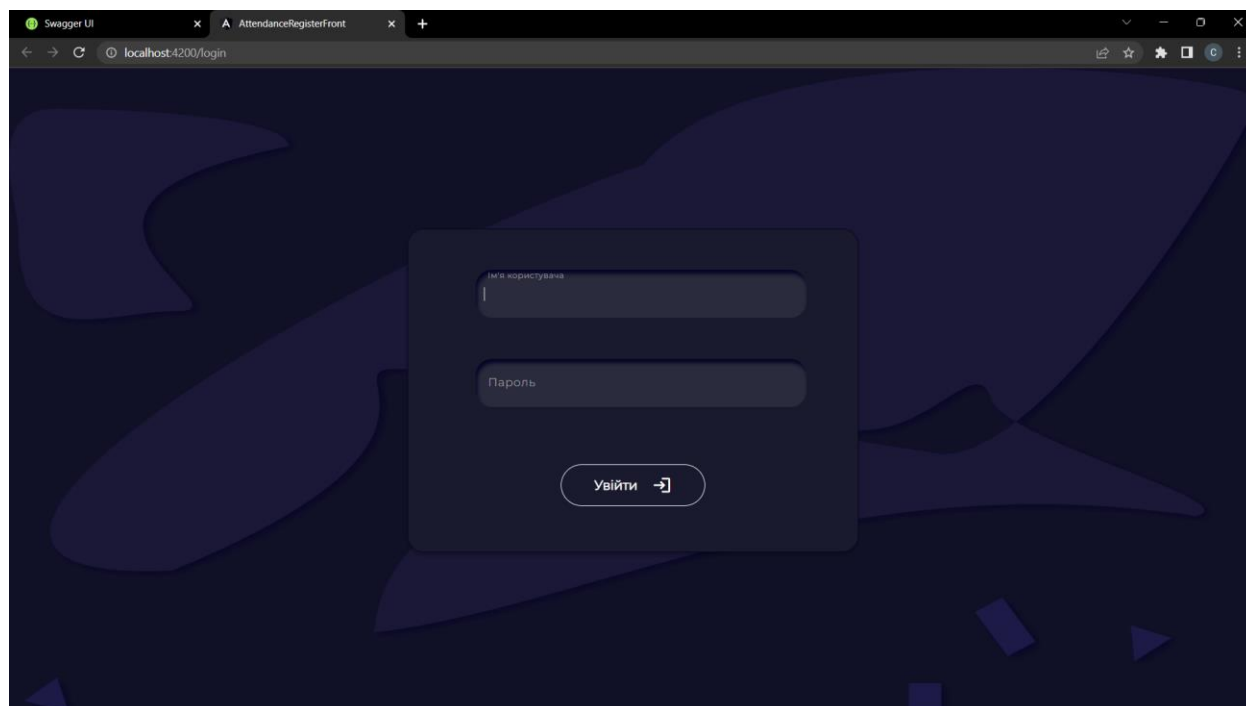


Рисунок 9 – Сторінка авторизації

Розглянемо спочатку сторінку адміністратора. При вході адміністратор одразу бачить список всіх учнів, елементи для роботи з списком (фільтрація та сортування), кнопки додавання, редагування та видалення учнів, а також чотири вкладки (учні, вчителі, класи, предмети) як показано на рисунку 10.

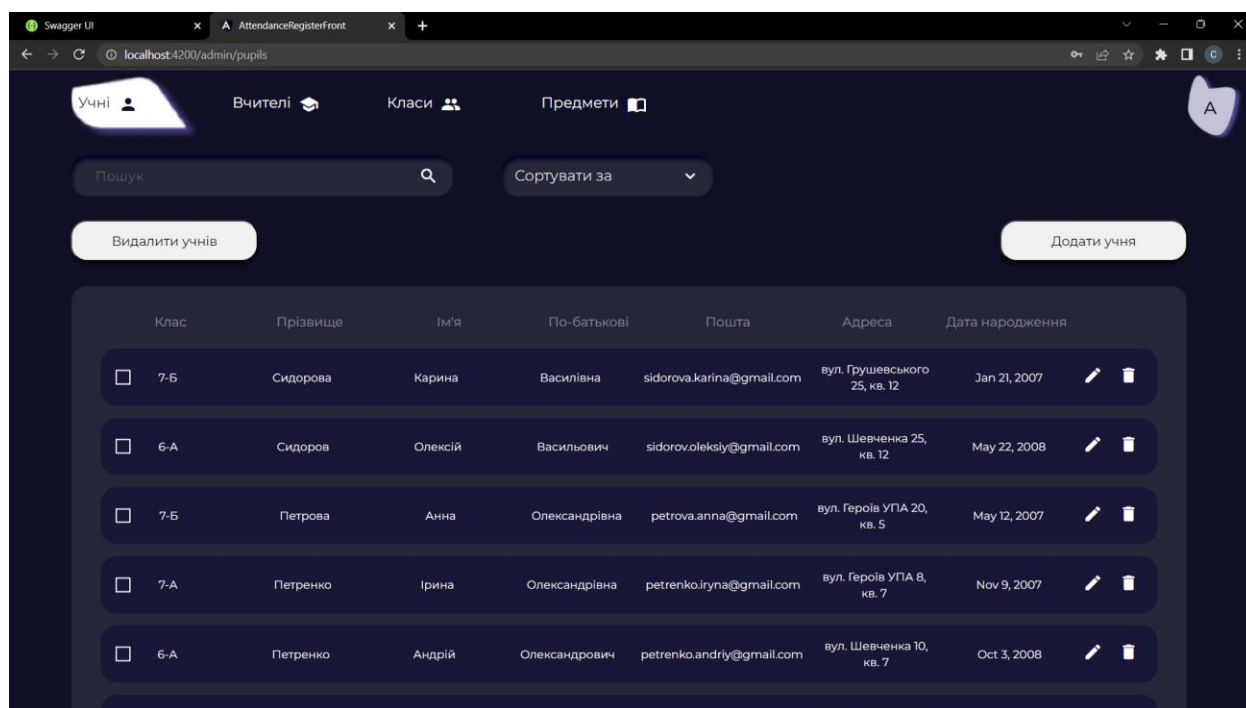


Рисунок 10 – Сторінка управління учнями

При натисканні на кнопку “Додати учня” або на кнопку редагування учня, відкривається вікно куди потрібно внести дані (рисунок 11). Різниця між цими кнопками полягає в тому, що при редагуванні поля з даними будуть заповнені.

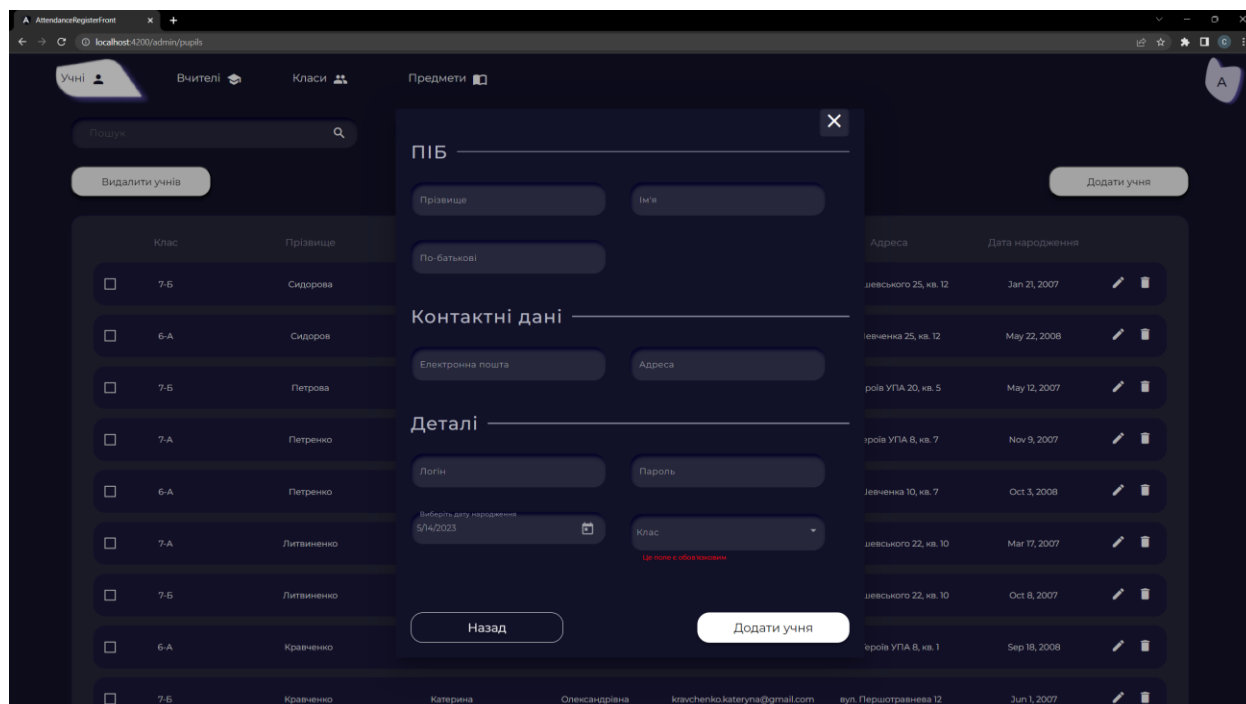


Рисунок 11 – Вікно додавання учня

При натисканні на кнопку видалення учня, як і видалення будь-чого ще відкривається вікно з підтвердженням операції як показано на рисунку 12.

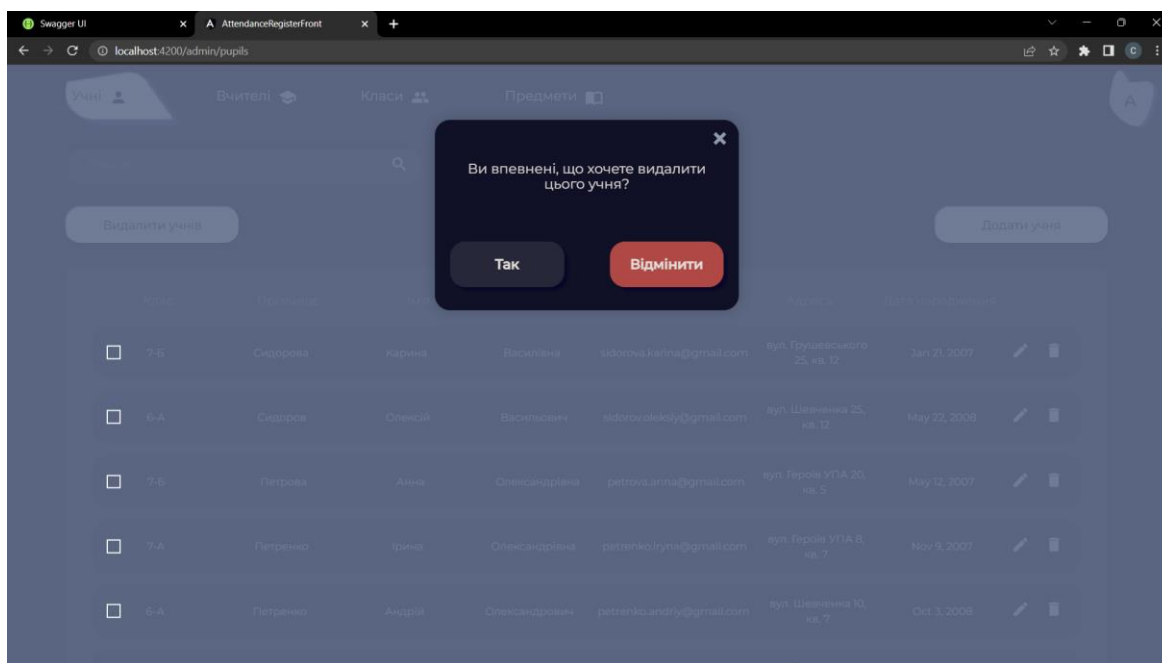


Рисунок 12 – Вікно підтвердження видалення

При переході на сторінку вчителів, адміністратор бачить ідентичну сторінку але уже зі списком вчителів (рисунок 13).

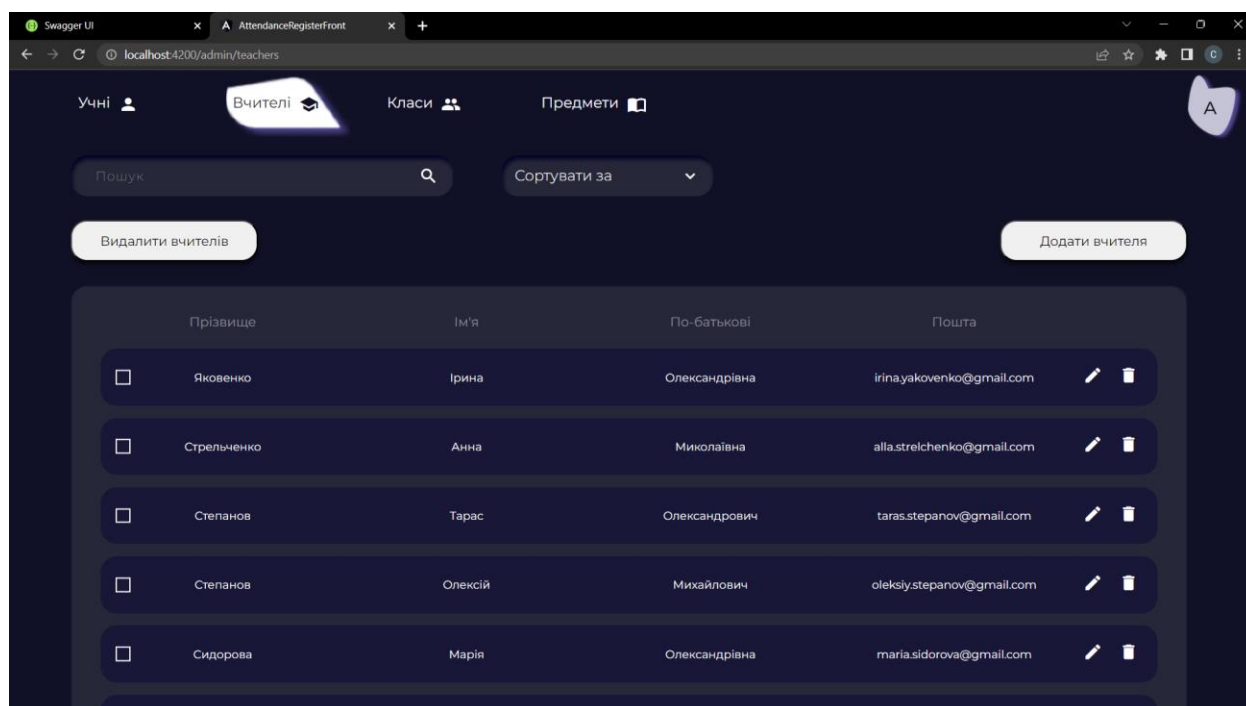


Рисунок 13 – Сторінка управління вчителями

При переході на вкладку “Класи”, адміністратор бачить список класів з різною інформацією про них, такою як назва, профіль, класний керівник та кількість учнів (рисунок 14).

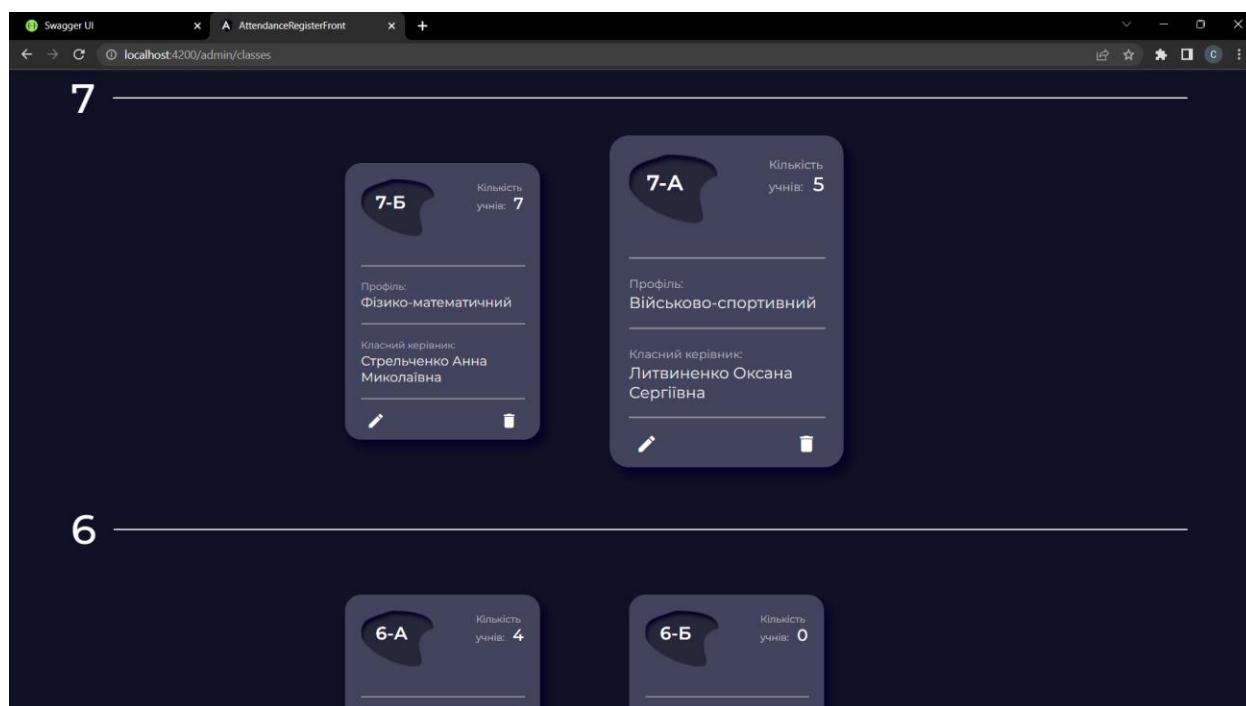


Рисунок 14 – Сторінка управління класами

Крім того адміністратор може додати, редагувати чи видалити клас. При додаванні та редагуванні відкривається відповідне вікно для внесення даних. Натиснувши на клас, система відкриє список учнів як на рисунку 10, але лише з учнями, що навчаються в вибраному класі.

Остання вкладка на сторінці адміністратора називається “Предмети”. На даній вкладці адміністратор бачить список предметів, вчителів які ведуть предмети та класи в яких предмети викладаються (рисунок 15). Список можна відфільтрувати та відсортувати. Крім того можна додати новий предмет, видалити чи редагувати існуючий.

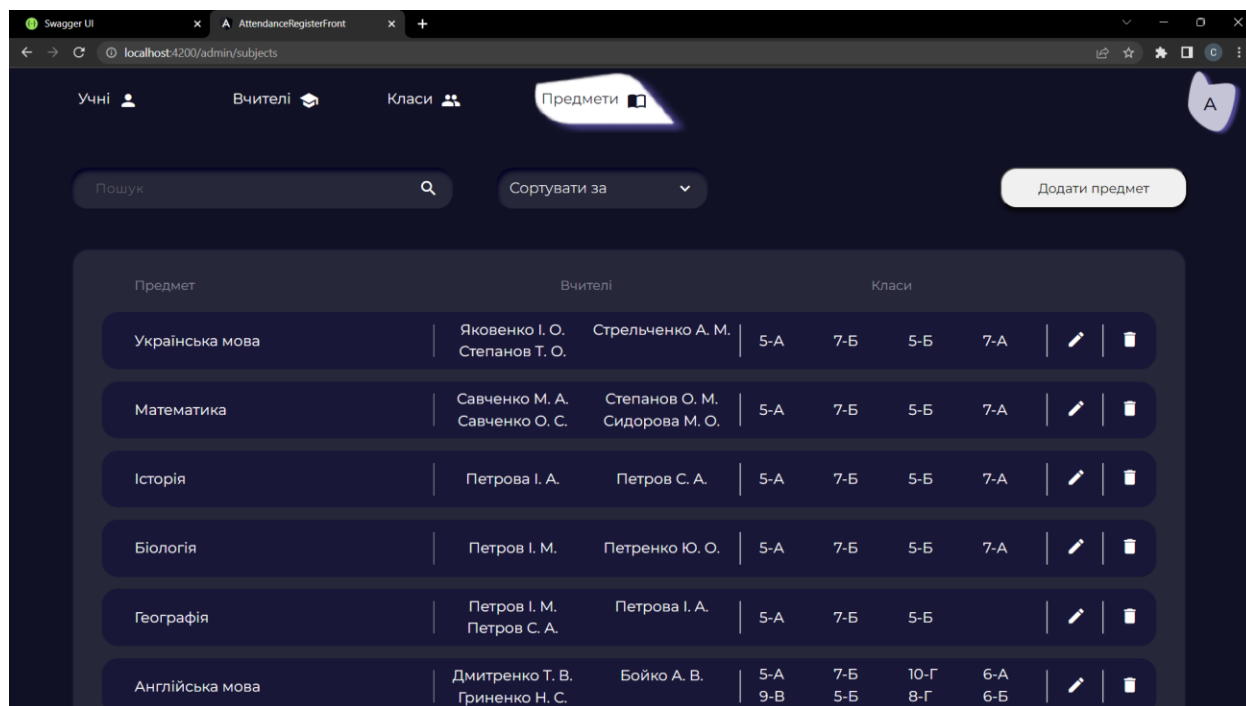


Рисунок 15 – Сторінка управління предметами

Сторінка редагування відрізняється від сторінок редагування інших сутностей. На даній сторінці можна додати чи видалити вчителя, який веде предмет (рисунок 16) або додати чи видалити клас в якому цей предмет ведеться (рисунок 17). При вводі даних присутнє автозаповнення.

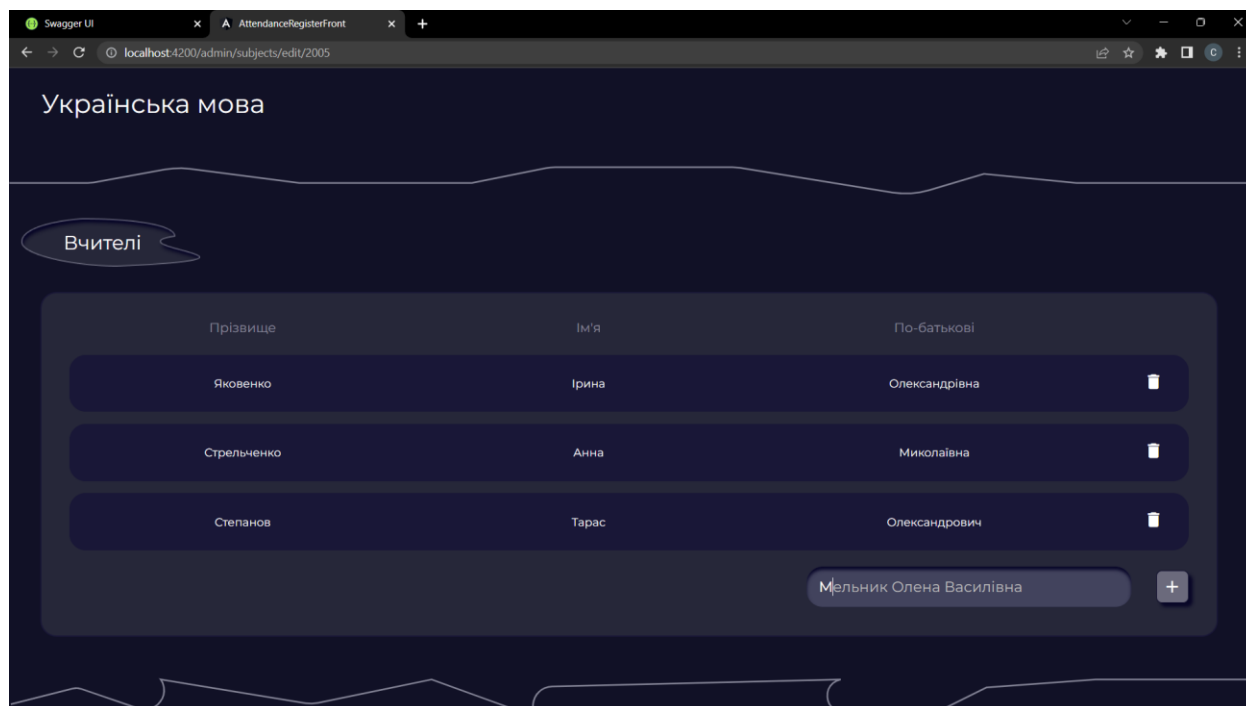


Рисунок 16 – Управління вчителями предмету

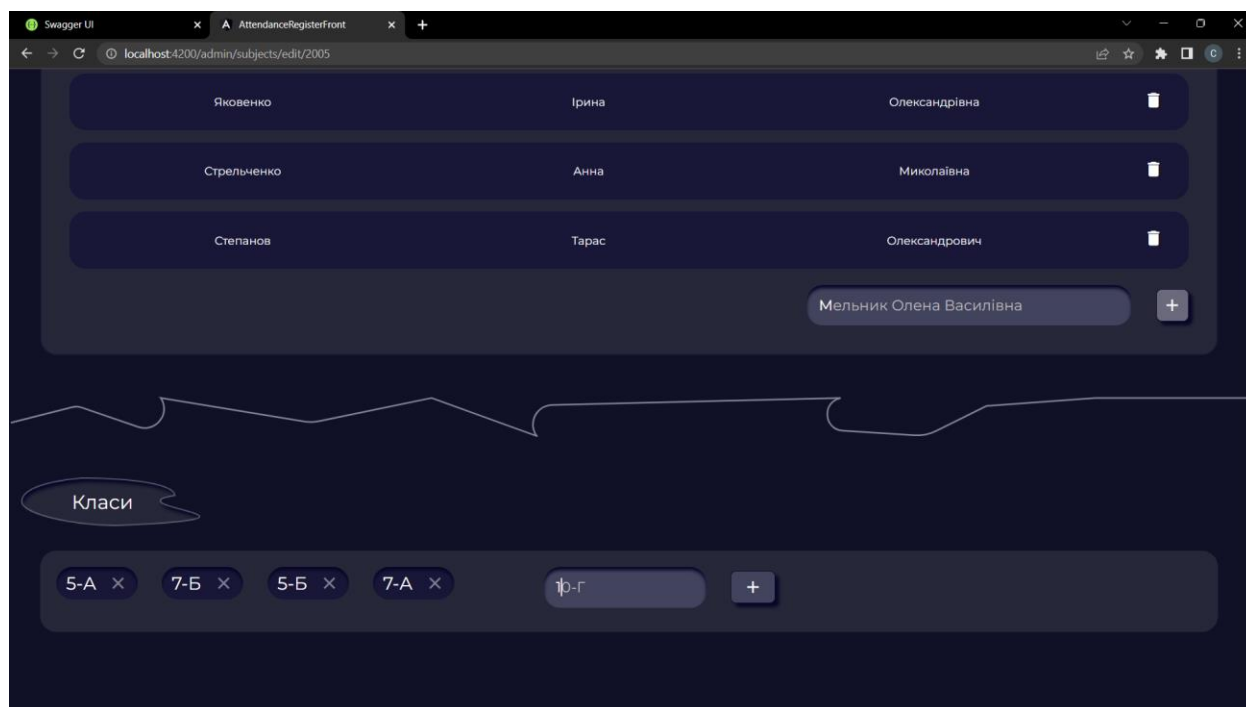


Рисунок 17 – Управління класами предмету

Якщо користувач є вчителем, то він після авторизації переходить на сторінку з класами які він учить та предметами (рисунок 18, 19).

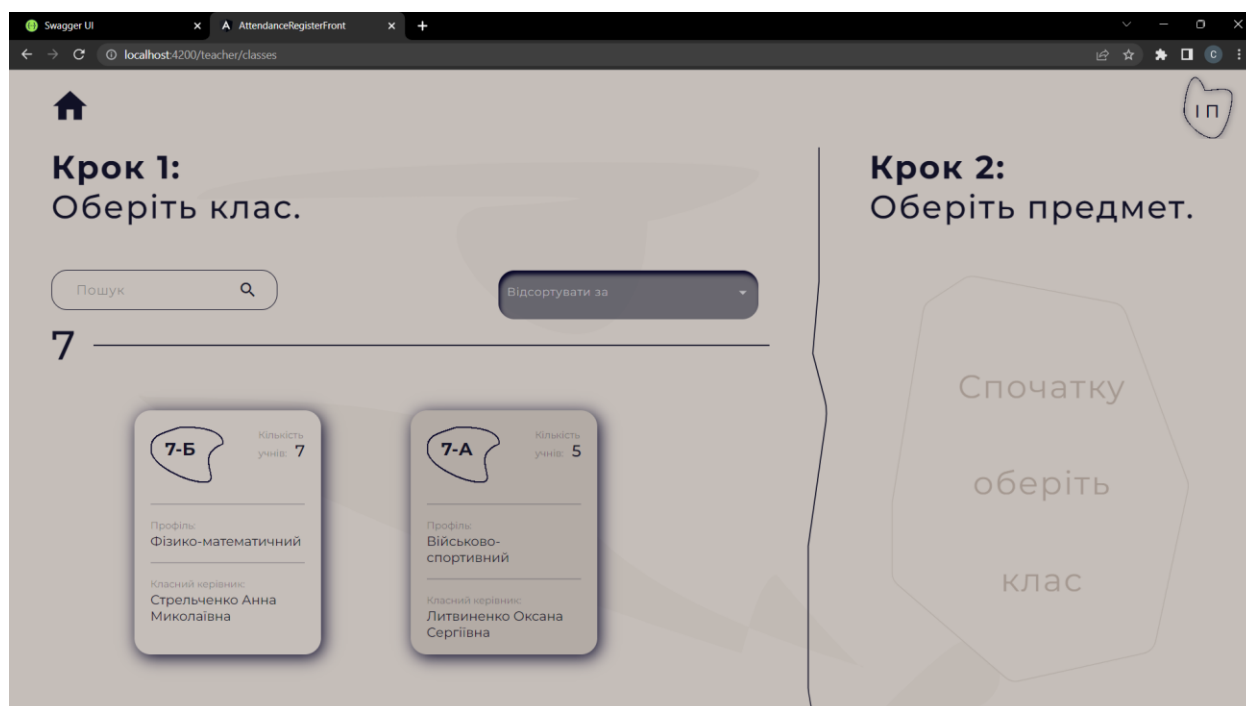


Рисунок 18 – Сторінка з класами вчителя (клас не вибрано)

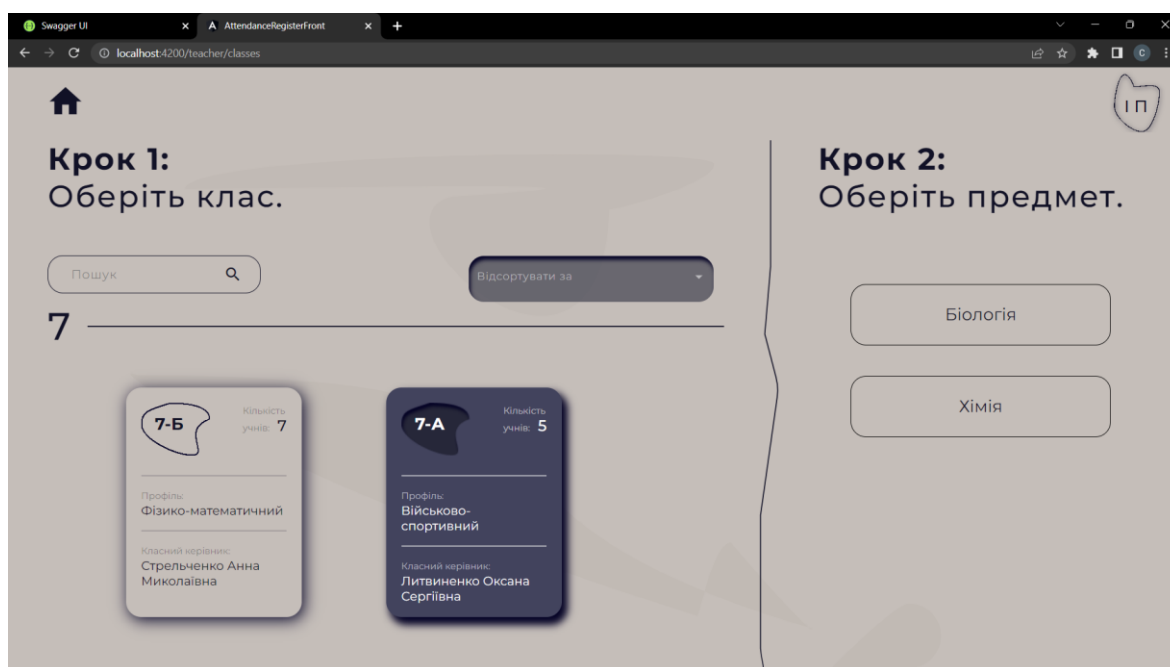


Рисунок 19 – Сторінка з класами вчителя (клас вибрано)

При натисканні на предмет, система відкриває сторінку з розділами цього предмету (рисунок 20).

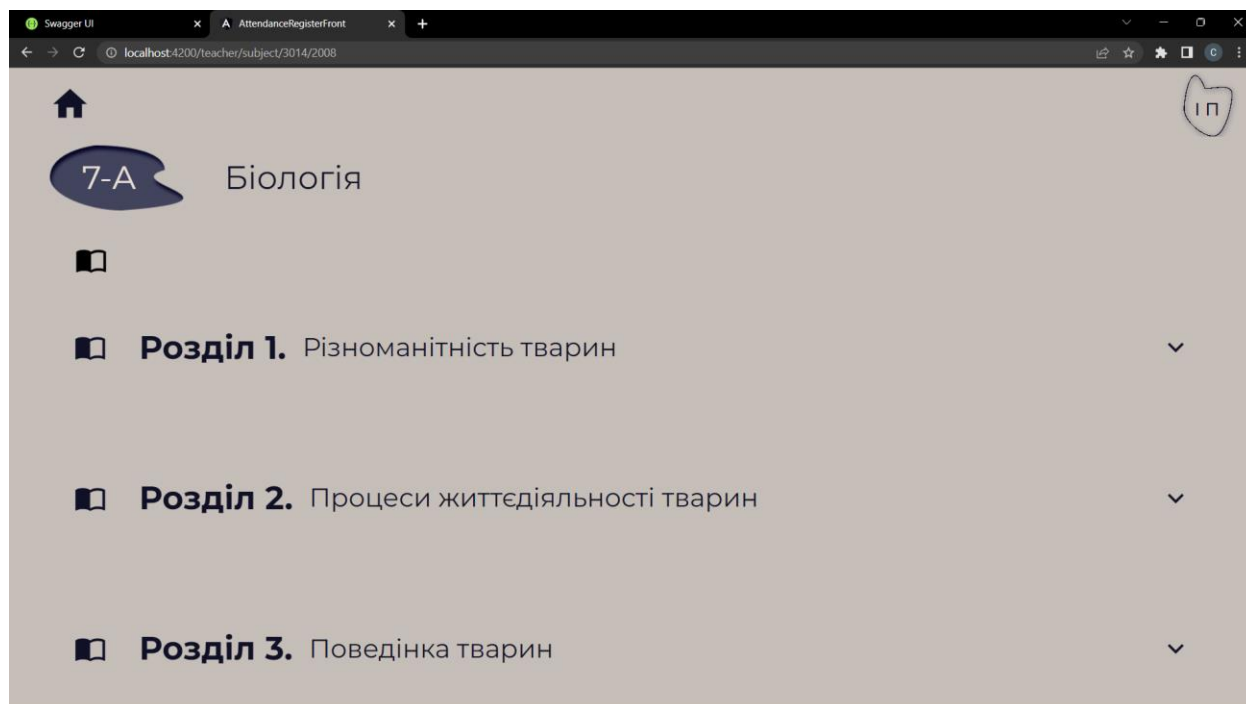


Рисунок 20 – Сторінка з розділами предмету

При натисканні на розділ, відкриваються теми розділу як показано на рисунку 21.

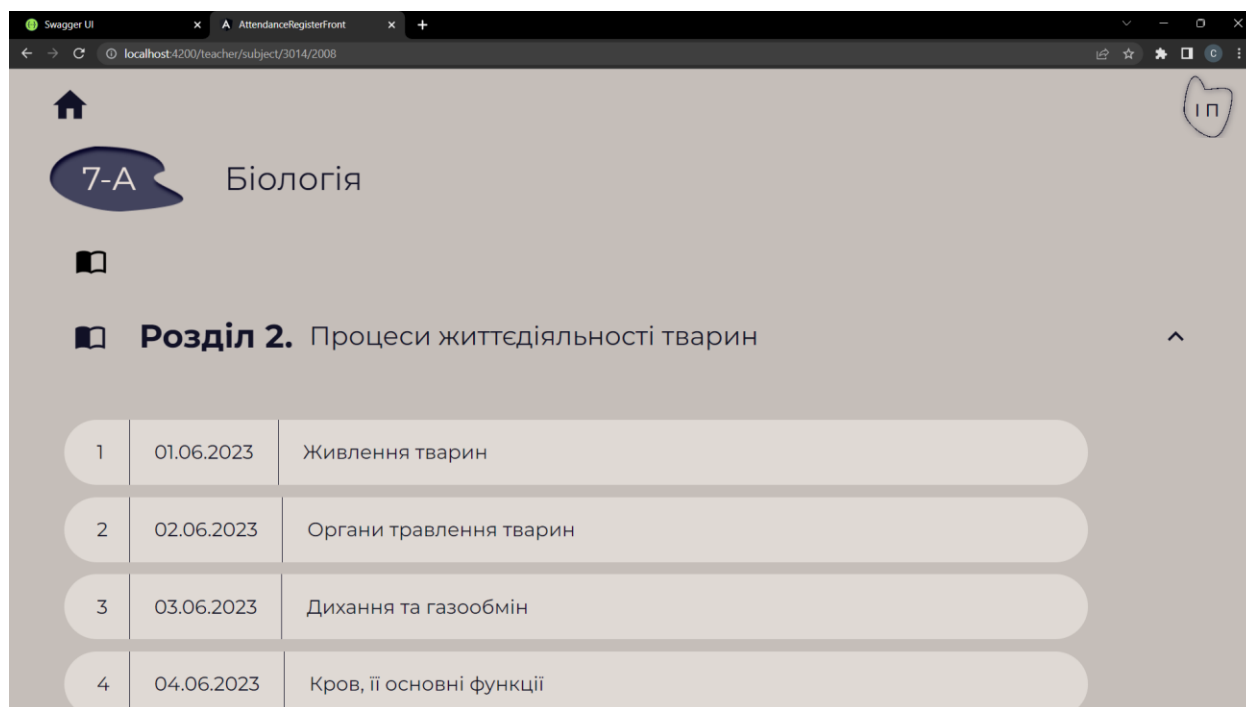


Рисунок 21 – Теми розділу

При натисканні на іконку журналу під назвою класу відкривається журнал з оцінками по всіх розділах та темах (рисунок 22). Контрольні роботи, семестрові та річні оцінки підсвічуються іншим кольором. Також можна натиснути на іконку журналу біля конкретного розділу для того щоб відкрити оцінки лише по вибраному розділу.

Учень	Контрольна робота 1												Контрольна робота 2												Контрольна робота 3											
	08.05	09.05	10.05	11.05	12.05	13.05	14.05	15.05	16.05	17.05	18.05	19.05	20.05	21.05	22.05	23.05	24.05	01.06	02.06	03.06	04.06	05.06	06.06	07.06	08.06	09.06	10.06	11.06								
Коваленко Михайло Сергійович	н		9					6	5								7	н		11																
Петренко Ірина Олександрівна			н		11				12		10							11		12	н			11		10		7								
Григоренко Андрій Олегович		11		10		н			12									12			11				6	12		5								
Литвиненко Юлія Сергіївна			6						8		9		н				9		8			5		н	9	9										
Жуков Олександр Віталійович		н			11				10	н			9				11	2						2		11		н								

Рисунок 22 – Журнал з оцінками

На цій сторінці вчителі можуть виставляти оцінки або відмітки про відсутність учня на уроці ввівши літеру “н”. Мітки з відсутністю підсвічуються червоним кольором, що можна побачити на рисунку 22. Крім того вчитель може додавати нові теми. Додатково можна побачити кнопки за допомогою яких можна автогенерувати оцінки за семестри або за рік, що економить значну кількість часу вчителям не зобов’язуючи їх рахувати ці оцінки вручну. Також присутня кнопка “Експортувати”, яка експортує дану таблицю в Excel файл як на рисунку 23. Ця функція допоможе коли потрібно, наприклад, роздрукувати список оцінок.

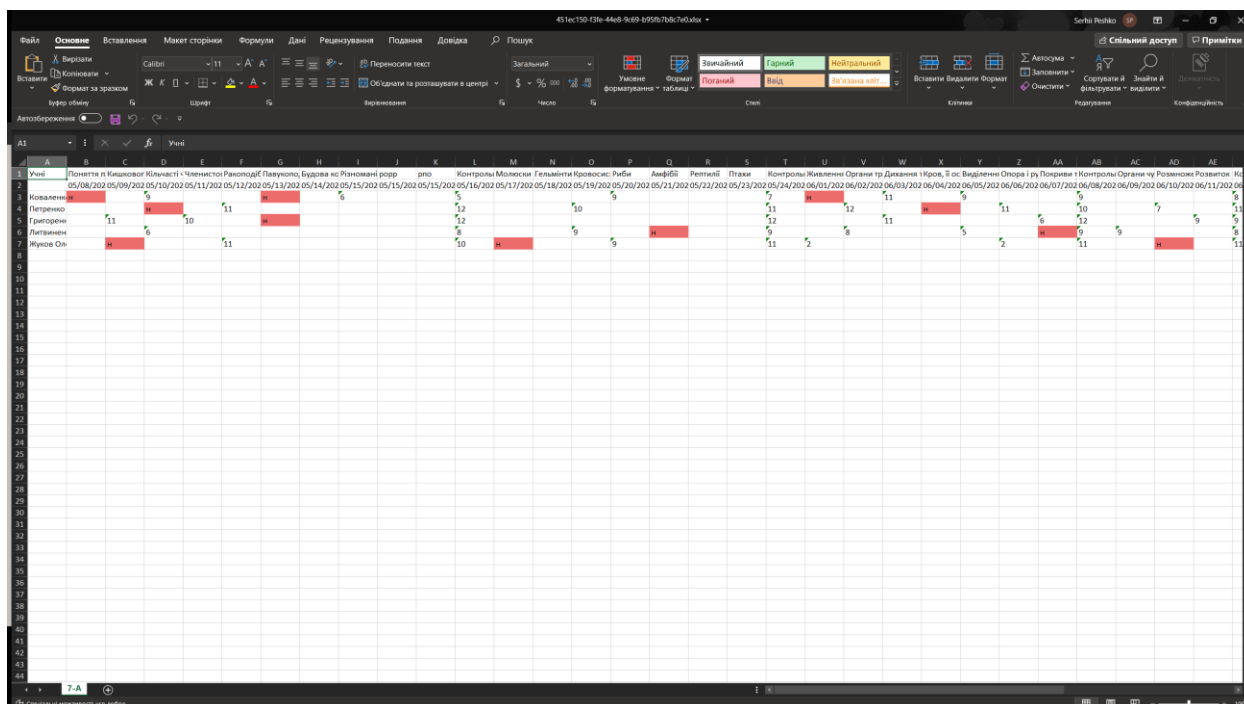


Рисунок 23 – Оцінки експортовані в Excel файл

Також присутня кнопка “Аналітика”, яка відкриває сторінку аналітики роботи учнів даного класу з даного предмету (рисунок 24).

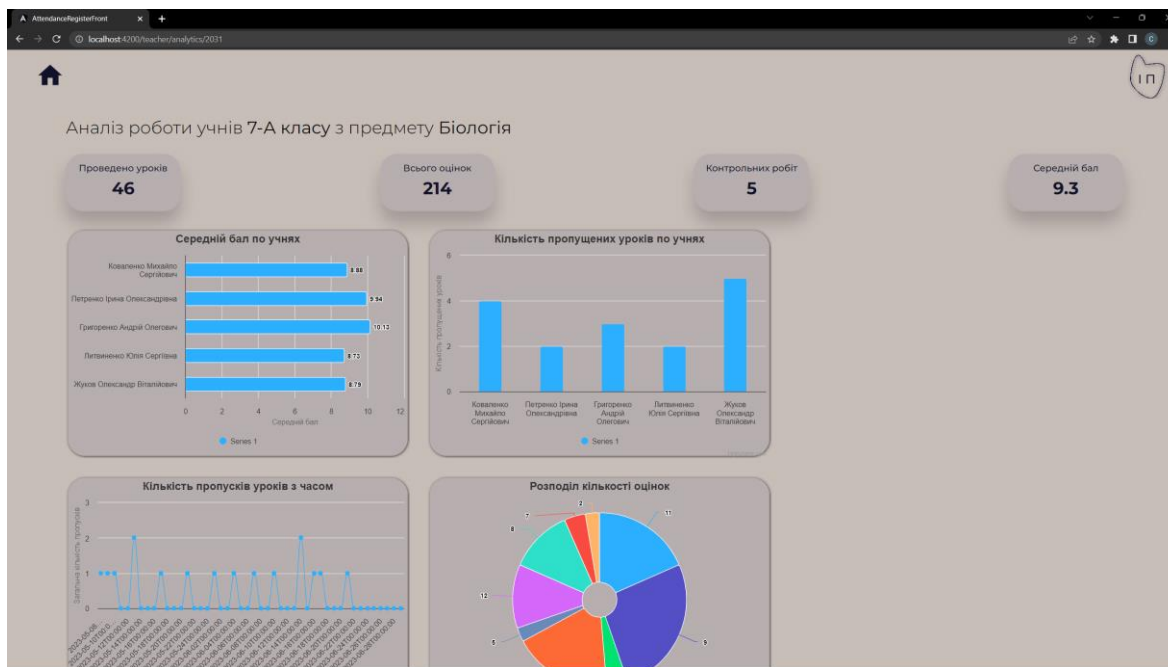


Рисунок 24 – Сторінка аналітики роботи учнів

На даній сторінці вчитель може переглянути такі показники як кількість проведених уроків, виставлених оцінок, контрольних робіт та середній бал всіх учнів. Крім того, вчитель має змогу переглянути деякі графіки роботи учнів, такі як середній бал кожного учня, кількість пропущених уроків по учнях, як змінювалась кількість пропусків з часом та розподіл оцінок.

Якщо авторизований користувач є учнем, то йому відкривається список предметів, які навчають в його класі (рисунок 25).



Рисунок 25 – Список предметів учня

При натисканні на предмет учню відкривається журнал з оцінками, але він їх не може редагувати (рисунок 26).

Учень	08.05	09.05	10.05	11.05	12.05	13.05	14.05	15.05	16.05	17.05	18.05	19.05	20.05	21.05	22.05	23.05	24.05	01.06	02.06	03.06	04.06	05.06	06.06	07.06	08.06	09.06	10.06	11.06
	Поняття про класифікацію тварин	Кишковогортаючі	Кільчасті черви	Членистоногі	Ракподібні	Павукоподібні	Будова комах	Різноманітність комах	Контрольна робота 1	Молоски	Гельмінти	Кровосисні членистоногі	Риби	Амфібії	Рептилії	Птахи	Контрольна робота 2	Живлення тварин	Органи травлення тварин	Дихання та газообмін	Кров, її основні функції	Виділення та його значення	Опора і рух	Покриття тіла тварин	Контрольна робота 3	Органи чуття тварин	Розмноження та його значення	
Петренко Ірина Олександрівна			н		11				12			10					11	12		н		11		10		7		
Литвиненко Юлія Сергіївна			6						8			9	н				9	8			5		н	9	9			
Коваленко Михайло Сергійович	н		9		н			6	5			9					7	н	11		9				9			
Жуков Олександр Віталійович		н			11				10	н		9					11	2					2	11		н		
Григоренко Андрій Олегович		11		10		н			12								12			11				6	12			

Рисунок 26 – Сторінка з оцінками від лица учня

При натисканні на кнопку “Аналітика” учень може переглянути згенеровану аналітику його роботи з даного предмету (рисунок 27).



Рисунок 27 – Сторінка аналітики учня

На сторінці аналітики учень може переглянути свій середній бал, кількість своїх оцінок з даного предмету, кількість пропущених уроків та кількість відвіданих уроків. Крім того він може переглянути такі графіки як зміну оцінок протягом часу, розподіл своїх оцінок, розподіл відвідуваності та розподіл кількості оцінок. Всі графіки є інтерактивними, тобто при наведенні на якусь частину будь-якого графіку можна переглянути детальнішу інформацію по цій частині (рисунок 28), а інші графіки підлаштуються під фільтр.

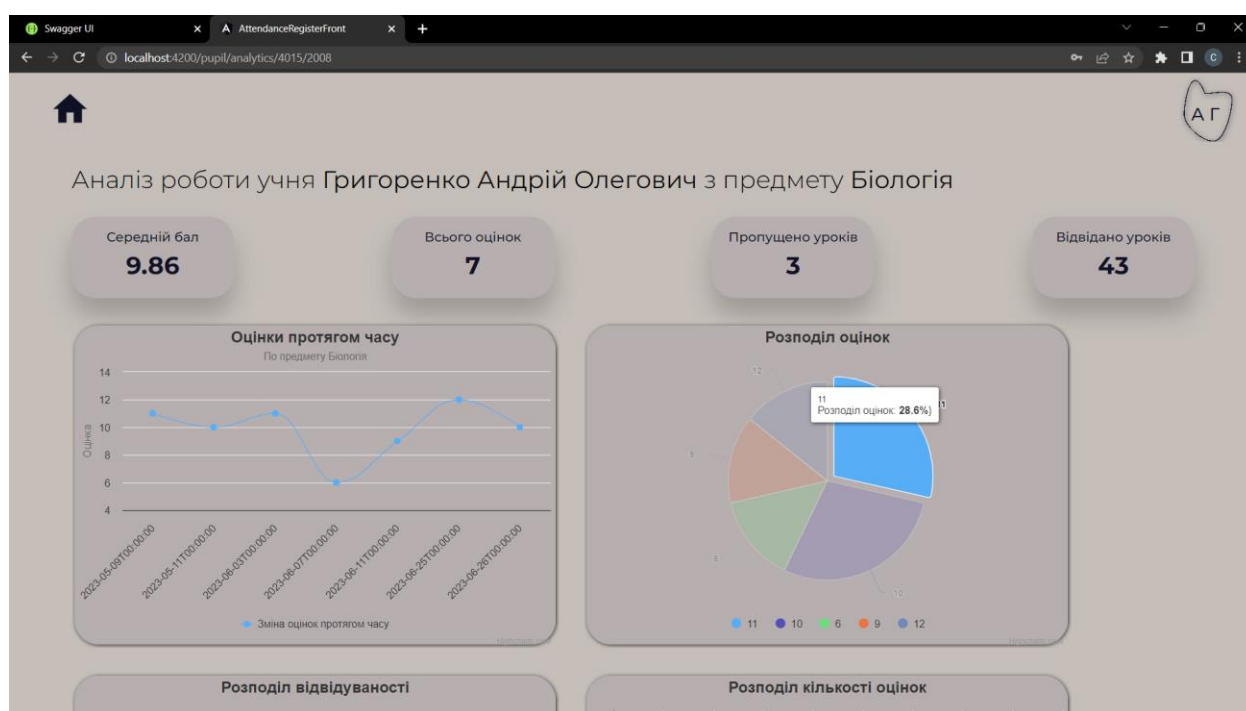


Рисунок 28 – Детальна інформація по частині графіку

ВИСНОВКИ

Проведено порівняльний аналіз ряду систем які містять шкільний електронний журнал, що дозволило зрозуміти виклики та можливості пов'язані зі сферою EdTech. Визначено основні переваги та недоліки систем на ринку, що містять електронний журнал та досліджено можливі вдосконалення цих систем. Розроблено систему “Електронний журнал”, яке дає змогу здійснювати моніторинг та проводити аналіз успішності учнів та адмініструвати навчальний процес.

Розроблена система може полегшити адміністрування школи, ведення обліку вчителів, учнів, предметів, виставлення оцінок, формування аналітичних звітів тощо. Даний проект підкреслює важливість дизайну орієнтованого на користувача, надійної безпеки, функціональності та гнучкості для розробки освітніх інструментів.

Перспективи подальших досліджень: вдосконалення системи шляхом додавання штучного інтелекту чи елементів машинного навчання для формування прогнозів, додавання інтеграцій з іншими сервісами для більшої гнучкості та функціональності системи, додавання ролі “Батьки” зі своїм функціоналом.

Дана робота показує, що майбутнє освіти полягає у використанні потужності технологій для створення середовища, яке є сприятливим та ефективним для навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Visual Studio [Електронний ресурс] – режим доступу до ресурсу: <https://visualstudio.microsoft.com/>.
2. Troelsen A. Pro C# 10 with .NET 6 / A. Troelsen, P. Japikse. – Apress Berkeley, CA, 2022. – 1640 с.
3. Build. Test. Deploy. [Електронний ресурс] – режим доступу до ресурсу: <https://dotnet.microsoft.com/en-us/>.
4. Angular (web framework) [Електронний ресурс] – режим доступу до ресурсу: <https://angular.io/docs>.
5. The Open Encyclopedia of Educational Technology / [J. Backman, A. Bozkurt, M. Brown та ін.], 2022. – 179 с.
6. Edmodo [Електронний ресурс] – режим доступу до ресурсу: <https://www.edmodo.com/>.
7. Google Classroom [Електронний ресурс] – режим доступу до ресурсу: <https://support.google.com/edu/classroom/answer/6020279?hl=en>.
8. Blackboard [Електронний ресурс] – режим доступу до ресурсу: <https://www.blackboard.com/>.
9. Schoology Learning [Електронний ресурс] – режим доступу до ресурсу: <https://www.powerschool.com/classroom/schoology-learning/>.
10. UML Use Case Diagrams [Електронний ресурс] – режим доступу до ресурсу: <https://www.uml-diagrams.org/use-case-diagrams.html>.
11. Microsoft SQL Server [Електронний ресурс] – режим доступу до ресурсу: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>.

12. The Most Popular Frontend Frameworks in 2023 [Электронный ресурс] – режим доступа до ресурсу: <https://stackdiary.com/front-end-frameworks/>.

13. Docker [Электронный ресурс] – режим доступа до ресурсу: <https://www.docker.com/>.

14. Highcharts [Электронный ресурс] – режим доступа до ресурсу: <https://www.highcharts.com/>.

15. Entity Framework Core [Электронный ресурс] – режим доступа до ресурсу: <https://learn.microsoft.com/en-us/ef/core/>.

16. The Collaborative API Development Platform – Insomnia [Электронный ресурс] – режим доступа до ресурсу: <https://insomnia.rest/>.

17. What is Client Server Architecture [Электронный ресурс] – режим доступа до ресурсу: <https://intellipaat.com/blog/what-is-client-server-architecture/?US>.

18. 3 Layered Architecture [Электронный ресурс] – режим доступа до ресурсу: <https://www.ecanarys.com/Blogs/ArticleID/76/3-Layered-Architecture>.

19. CRUD – definition & overview [Электронный ресурс] – режим доступа до ресурсу: <https://www.sumologic.com/glossary/crud/>.

20. Repository Design Pattern [Электронный ресурс] – режим доступа до ресурсу: <https://medium.com/@pererikbergman/repository-design-pattern-e28c0f3e4a30>.

21. What is a REST API? [Электронный ресурс] – режим доступа до ресурсу: <https://www.ibm.com/topics/rest-apis>.

22. Deursen S. Dependency Injection Principles, Practices, and Patterns / S. Deursen, M. Seemann. – Shelter Island, NY, USA: Manning Publications, 2019. – 552 с.

23. Introduction to JSON Web Tokens [Электронный ресурс] – режим доступа до ресурсу: <https://jwt.io/introduction>.

24. School Today [Электронный ресурс] – режим доступа до ресурсу: <https://school-today.com/>.

ДОДАТОК А

Структура файлів системи

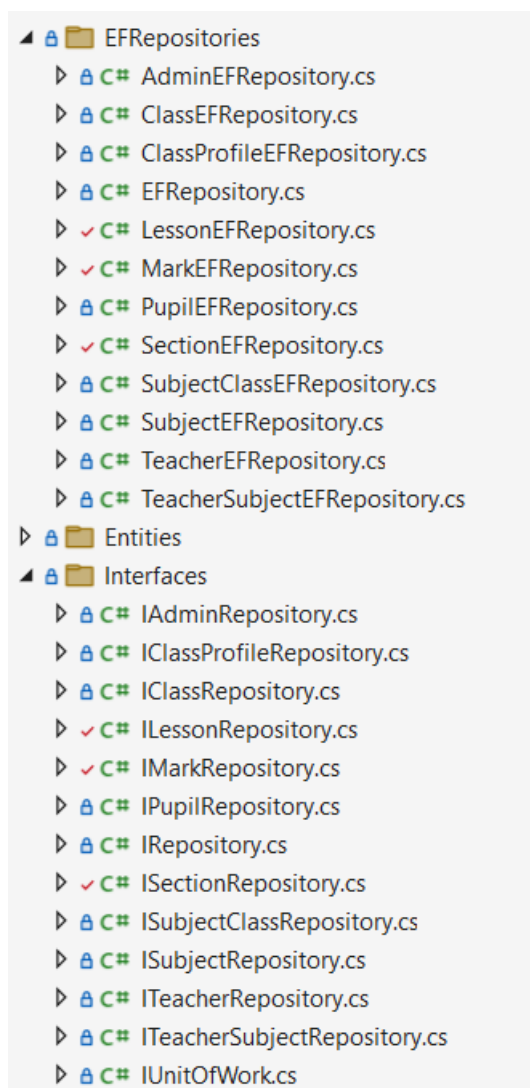


Рисунок А.1 – Структура класів репозиторіїв

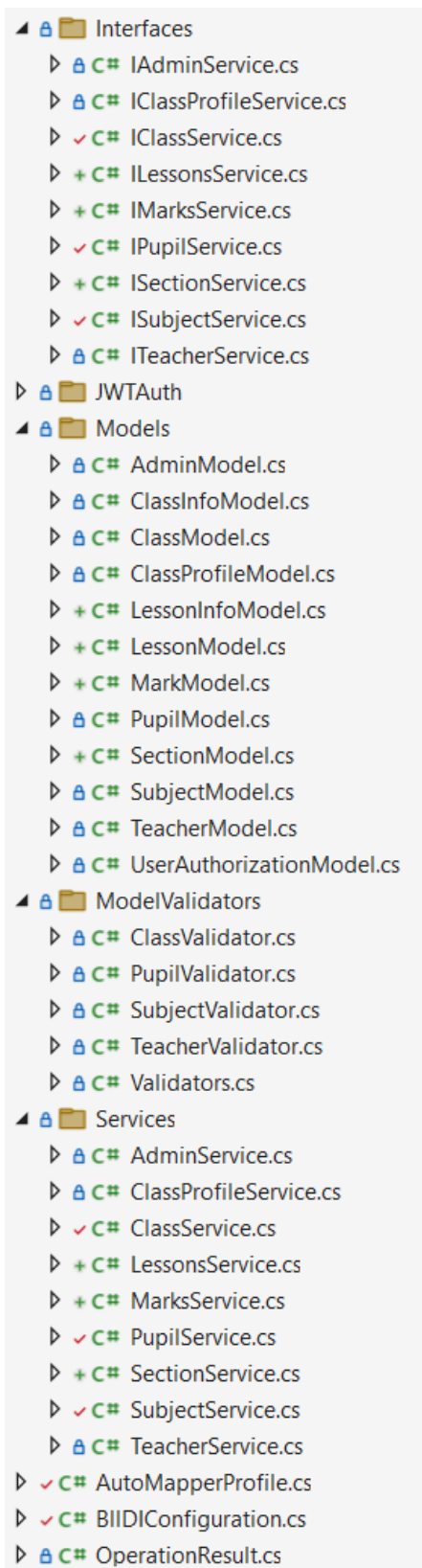


Рисунок А.2 – Структура файлів рівня бізнес-логіки

ДОДАТОК Б

Діаграми класів серверної частини

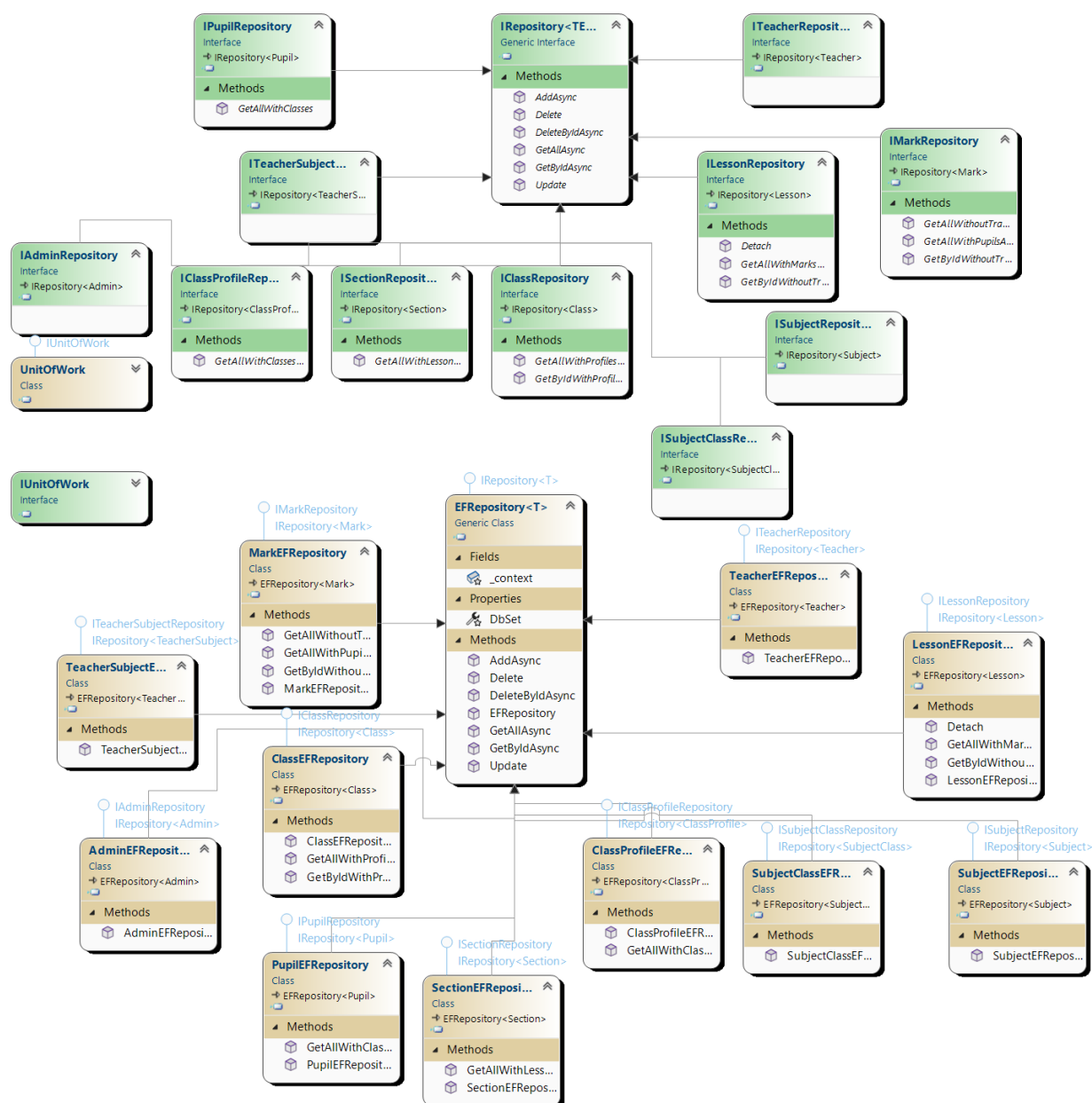


Рисунок Б.1 – Діаграма класів рівня доступу до даних

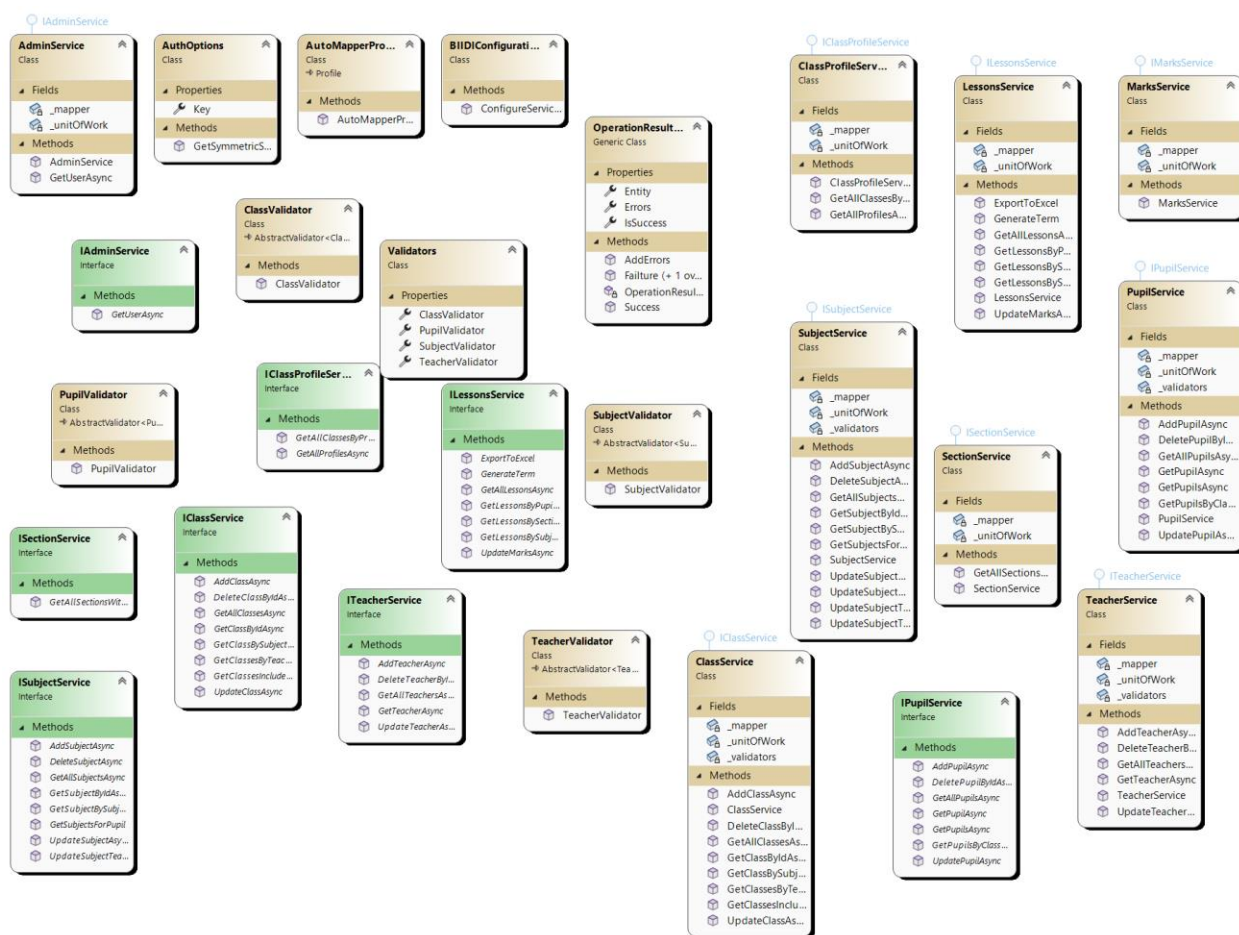


Рисунок Б.2 – Діаграма класів рівня бізнес-логіки