

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

БАКАЛАВРСЬКА РОБОТА

на тему:

«ВЕБ-СЕРВІС КОНТРОЛЮ УСПІШНОСТІ УЧНІВ»

Виконав: студент 4 курсу

Рівень бакалавр

Галузь знань 12 «Інформаційні технології»,
спеціальність 122 «Комп'ютерні науки»

Белецький Даніїл Миколайович

Керівник: к.т.н., доцент

Задерейко Олександр Владиславович

Рецензент: к.пед.н., доцент

Логінова Наталія Іванівна

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет **кібербезпеки та інформаційних технологій**
Кафедра **інформаційних технологій**
Галузь знань: **12 Інформаційні технології**
Спеціальність: **122 Комп'ютерні науки**
Назва освітньої програми: **Комп'ютерні науки**

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних технологій

доцент Н.І. Логінова

_____ « ____ » _____ 20__ року

З А В Д А Н Н Я

на кваліфікаційну (бакалаврську) роботу
здобувачу Белецькому Даніїлу Миколайовичу

1. Тема роботи: «Веб-сервіс контролю успішності учнів»

Керівник роботи: к.т.н., доцент Задерейко Олександр Владиславович

затверджені наказом Ректора НУ «ОЮА» від «02» квітня 2024 року № 809-06.

2. Строк подання здобувачем роботи «20» травня 2024 року

3. Дата видачі завдання «15» вересня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та постановка задачі	01.12.23	
2	Опис розроблених алгоритмів та програмного забезпечення	26.01.24	
3	Результати експериментів та аналіз отриманих даних	15.03.24	
4	Оформлення пояснювальної записки	26.04.24	

Здобувач _____

(підпис)

(прізвище та ініціали)

Керівник роботи _____

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Белецький Д.М. Веб-сервіс контролю успішності учнів. – Кваліфікаційна робота бакалавра за спеціальністю 122 «Комп'ютерні науки».

Кваліфікаційна робота викладена на 54 сторінках, містить 3 розділи, 42 ілюстрації, 4 таблиці та 9 використаних джерел.

Об'єктом роботи є веб-сервіс контролю успішності та відвідувань учнів «Шкільний щоденник».

Предмет роботи – процес проектування та розробки веб-сервіс контролю успішності та відвідувань учнів.

Метою роботи є проектування та розробка веб-сервісу контролю успішності та відвідувань учнів «Шкільний щоденник».

Метод дослідження – прикладний.

У першому розділі проводиться аналіз предметної області та постановка задачі. У другому розділі описані розроблені алгоритми та програмне забезпечення. У третьому розділі приводиться результати експериментів та аналіз отриманих даних.

За результатами роботи зроблено висновки про отримані результати та потенціал її подальшого розвитку.

Ключові слова: Вебсервіс, контроль успішності учнів, HTML, CSS, JavaScript, Express, СКБД, PostgreSQL, Bootstrap.

ABSTRACT

Beletskyi D.M. Web service for monitoring students' performance. – Bachelor's qualifying work in specialty 122 "Computer Science".

The qualification work is laid out on 54 pages, contains 3 chapters, 42 illustrations, 4 tables and 9 used sources.

The object of the work is the "School Diary" web service for monitoring students' progress and attendance.

The subject of the work is the process of designing and developing web services for monitoring students' progress and attendance.

The purpose of the work is the design and development of the web service for monitoring the progress and attendance of students "School Diary".

The research method is applied.

In the first section, the analysis of the subject area and the formulation of the problem are carried out. The second section describes the developed algorithms and software. The third section presents the results of the experiments and the analysis of the obtained data.

Based on the results of the work, conclusions were made about the obtained results and the potential for its further development.

Keywords: Web Service, students' progress control, HTML, CSS, JavaScript, Express, DBMS, PostgreSQL, Bootstrap.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА	
ЗАДАЧІ	8
1.1 Постановка задачі та огляд аналогів	8
1.2 Опис предметної області	9
1.3 Опис очікуваних результатів	14
РОЗДІЛ 2. ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ ТА ПРОГРАМНОГО	
ЗАБЕЗПЕЧЕННЯ	15
2.1 Огляд використовуваних технологій та інструментів	15
2.2 Розробка структури бази даних	22
2.3 Архітектура застосунку	29
2.4 Діаграми прецедентів користувачів	33
2.5 Створення бази даних	40
2.6 Програмна реалізація системи	44
РОЗДІЛ 3. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ ТА АНАЛІЗ ОТРИМАНИХ	
ДАНИХ	46
3.1 Перевірка працездатності функціоналу вчителя	46
3.2 Перевірка працездатності функціоналу учня та батьків	54
3.3 Перевірка функціоналу адміністратора	56
ВИСНОВКИ	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТКИ	61
ДОДАТОК А. ПРОГРАМНИЙ КОД	61

ВСТУП

У даний час існує проблема комунікації між батьками та школою, у котрій навчається їх дитина. Найчастіше дитина не хоче показати погані оцінки батькам, тому він йде на різні хитрощі: виправляє оцінки у щоденнику на більше «правильні», веде різні щоденники для поганих та хороших оцінок. Говорять батькам, що вчителі «не задають» домашнє завдання, а потім не знають чому це їм поставили погану оцінку, якщо ніякого домашнього завдання не було. Не ходять на уроки, а батьки гадають, що їх дитини сидить у школі та навчається.

Саме тому метою даної роботи є проектування та реалізація веб-сервіс контролю успішності та відвідувань учнів «Шкільний щоденник» дозволяє батькам оперативно за допомогою сайту, а також повідомлень на пошту отримувати інформацію як навчаються та відвідують школу їх діти, які домашні завдання задані, розклад занять, зв'язуватися з класним керівником за допомогою чата; вчителям – виставляти чи переглядати дані з успішності, відвідувань та домашнє завдання чи завантажувати дані за допомогою файлів та спілкуватися класному керівнику з батьками; учням – переглядати свої оцінки, подивитися своє домашнє завдання розклад занять.

Для досягнення даної мети необхідно вирішити наступні задачі:

- 1) Виконати аналіз предметної області веб-сервіс контролю успішності та відвідувань учнів;
- 2) Розробити моделі предметної області;
- 3) Розробити архітектуру системи;
- 4) Розробити базу даних;
- 5) Розробити класи, які реалізують задачі системи у рамках шаблону проектування MVC для роботи з базою даних та сайтом;
- 6) Розробити користувацький інтерфейс з урахуванням прав доступу як на стороні бази даних так і на стороні користувача даної системи яким буде наданий користувацький інтерфейс;
- 7) Перевірити результати роботи системи;

Об'єктом роботи є веб-сервіс контролю успішності та відвідувань учнів «Шкільний щоденник»..

Предметом роботи є процес проєктування та розробки веб-сервіс контролю успішності та відвідувань учнів.

Метою роботи є проєктування та розробка веб-сервісу контролю успішності та відвідувань учнів «Шкільний щоденник».

Методика дослідження – прикладна.

Практична значущість результатів дослідження: Програмна система готова до роботи у реальних умовах.

Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Постановка задачі та огляд аналогів

Веб-сервіс контролю успішності та відвідувань учнів «Шкільний щоденник» повинен налагодити зручну та швидку комунікацію школи та батьків, що дозволить батькам краще контролювати і виховувати своїх дітей, а вчителям своєчасно отримувати підтримку з боку батьків для більш якісної організації учбового процесу.

Таким чином додаток повинен реалізувати такі основні можливості:

- Вхід до системи;
- Перегляд учнями та батьками домашній завдань;
- Перегляд учнями, батьками, вчителями середніх оцінок та рейтингу за предмет;
- Перегляд учнями та батьками оцінок та пропусків за проміжок дат;
- Перегляд розкладу для учнів, батьків та вчителя;
- Оперативне повідомлення батьків про відмітки, домашнє завдання через поштову розсилку;
- Виставлення та задання вчителем відміток та домашнього завдання;
- Змінити відмітки чи домашнє завдання вчителем;
- Чат повідомлення вчителя (класного керівника) з батьками;
- Завантаження через файл відміток чи домашнього завдання;
- Управління користувачами адміністратору;
- Робота з розкладом адміністратору;
- Переведення у наступний клас адміністратором;

В даний момент на ринку вже існують веб-сервіси, що надають можливість електронних щоденників. Кожна з них має свої переваги та недоліками.

Веб-сервіс «e-schools»

Веб-сервіс «e-schools» надає доступ батькам, учням, учителям, директору та адміністратору доступ, щоб переглянути чи занести інформацію для контролю успішності та відвідувань учнів.

У частині роботи з журналом інтерфейс схожий з паперовими носіями, а в разі необхідності працює підтримка з якою можна зв'язатися поштою або скайп.

За допомогою SWOT-діаграми (рисунок. 1.1) продемонстровано ряд недоліків та переваг веб-сервісу «e-schools».

Перерахуймо інструменти, що реалізують можливості

а) Зручний журнал – аналог паперового журналу, у якому виставляються оцінки та пропуски;

б) Система навігація на сайті;

в) Можливість написати коментар під час виставлення відмітки;

г) Виставлення оцінки за поведінку;

д) Новини школи;

е) Стіна – куди, можливо додати будь-яку інформацію для її перегляду усіма користувачами у системі. Які зайдуть до вас у профіль;

Компанія вважає своєю особливістю такі можливості додатку як:

Серед основних баз даних, що використовує пошукова програма:

а) Виняткова функціональність у можливості індивідуального налаштувати багато аспектів роботи;

б) Велика швидкість обробки запитів під час роботи системи;

в) Зовнішній вигляд: всі шкільні документи (журнал, щоденник та інше) виглядає схожим на паперові аналоги;

	Позитивні сторони	Негативні сторони
Зовнішні фактори	Сильні сторони: Швидкий доступ до інформації Графік та таблиця успішності	Слабкі сторони Незручна навігація на сайті Повільна обробка інформації Ціна на послуги Велика кількість непотрібної інформації
Внутрішні фактори	Додаткові можливості: Зв'язок з підтримкою системи Повідомлення на сторінці профіля Аналіз успішності учнів Можливість створювати особистий контент Перегляд гуртків	Загрози Відсутня можливість завантажувати дані у файлі Відсутні повідомлення на e-mail про отримання відміток або домашнього завдання Складне юзабіліті сайту Інформаційне навантаження

Рисунок 1.1 – SWOT-діаграма веб-сервісу «e-school»

Аналогічний веб-сервіс «nz»

Веб-сервіс «nz» дає можливість учням та батьків переглядати щоденник та розклад, вчителям роботу з журналом як виставлення оцінок чи домашнього завдання, перегляд діаграм успішності учнів, класним керівникам надання паролів учням та батькам, можливість редагування їх даних, перегляд звітів, адміністратору перегляд звітів та реєстрація класних керівників та вчителів.

Сервіс співпрацює з урядовими міністерствами та є однією зі складових всеукраїнського проекту «КУРС: Освіта». З цього можемо зробити висновок, що сервіс являється безкоштовним.

Проте в разі необхідності, неможливо щось скоректувати під свою школу індивідуально, тому що все йде стандартно.

Інструменти, які реалізують ці можливості:

- а) Можливість дивитися аналіз навчання у різних діаграм;
- б) Можливість дивитися лише свій розклад користувачу;
- в) Можливість дивитися лише власні оцінки та пропуски користувачу;

На рис. 1.2 за допомогою SWOT-діаграми демонструються результати аналізу веб-сервісу «nz».

	Позитивні сторони	Негативні сторони
Зовнішні фактори	Сильні сторони: - Перегляд діаграм успішності - Перегляд звітів з навчання - Підтримка міністерством освіти	Слабкі сторони - Застарілий інтерфейс сайту - Незручний перегляд - Непотрібна інформація на сайті
Внутрішні фактори	Додаткові можливості: - Шкільні спільноти - Додавання в друзі - Перегляд новин - Ставити теми уроків при виставленні відміток - Перегляд сайту на різних мовах	Загрози - Відсутність повідомлень на e-mail - Слабка можливість адміністрування у системі - Незручне використання - Немає можливості завантаження у файлі - Складне адаптування до сайту

Рисунок 1.2 – SWOT-діаграма веб-сервісу «nz»

1.2. Опис предметної області

Як було зазначено раніше, мета створення веб сервісу полягає у налагодженні зручної комунікації школи та батьків для покращення контролю успішності та відвідувань їх дітей, на відміну від аналогів буде надавати зручний інтерфейс користувачам для вирішення їх задач та гнучку систему спілкування між вчителями та батьками та автоматичне повідомлення батьків через email розсилку про оцінки та пропуски їх дитиною.

Додаток визначається як програмне забезпечення, мета створення (якого полягає у налагодженні зручної комунікації школи та батьків для покращення контролю успішності та відвідувань їх дітей) цього додатку полягає в тому, щоб зібрати, проаналізувати та визначити високо рівневі потреби і можливості

програмного продукту. Додаток акцентує увагу на можливостях, необхідних учителям, учням, батькам, адміністратору.

Веб-сервіс контролю успішності та відвідувань учнів «Шкільний щоденник». представляє собою програмну систему для таких користувачів. Система є закритою, доступ до якої можна отримати через реєстрацію адміністратором системи кожного користувача окремо, а вже потім можлива особиста авторизація. Незареєстрований клієнт не має можливості переглянути дані у персональному кабінеті, проте може переглядати загальну інформацію.

В результаті перегляду предметної області було виділено п'ять рівнів доступу до системи:

1) Учень – зареєстрований користувач системи. Зареєструвати учня може адміністратор, який заносить до бази дані про учня та видає йому логін та пароль. Учень має права переглядати свої оцінки, домашнє завдання, переглядати розклад занять на один з днів тижня.

2) Вчитель – зареєстрований користувач системи. Зареєструвати вчителя може адміністратор, процедура реєстрації вчителя така ж сама як і до реєстрації учня. Вчитель має право задавати домашнє завдання учням того класу, де він веде, ставити відмітки учням (оцінка, пропуск) за певний день чи змінювати оцінку або, яку він виставив. Переглядати оцінки чи домашнє завдання, а також власний розклад. Учитель, ще замість ручного внесення оцінок може завантажувати оцінки чи домашнє завдання за допомогою файлу. Також класний керівник класу, може обмінюватися повідомленнями з батьками учнів свого класу.

3) Батьки – зареєстрований користувач системи. Зареєструвати батьків може адміністратор, процедура реєстрації батьків така ж сама як і до реєстрації учня, вчителя. Батьки мають права переглядати оцінок та пропусків своїх дітей за певну дату, домашнє завдання, переглядати розклад занять на один з днів тижня та дивитися профіль, де написана інформація хто їх класний керівник та в якому класі навчається їх дитина. Також батьки можуть обмінюватися повідомленнями з класним керівником класу своєї дитини.

4) Адміністратор системи – має змогу додавати, редагувати дані у системі. Адміністратор має права завантажувати з файлу розклад занять чи видаляти домашнє завдання класу, завантажувати чи вносити у ручну нових користувачів системи, переводити у наступний клас учнів, видаляти учнів чи редагувати інформацію, перегляд рейтингу чи середньої оцінки будь-якого класу заданого в додатку.

5) Гість – незареєстрований користувач системи. Має доступ лише до переліку інформації, яка знаходиться на головній сторінці сайту. Може переглядати сайт, проте не може зайти до персонального кабінету користувача та користуватися основними функціями.

Аналіз предметної області дозволяє визначити наступні поняття (глосарій).

Учень – дитина, котра навчається у школі, яка має можливість відвідувати уроки, щоб отримувати теоретичні та практичні навички.

Батьки – людина, котра являється батьками чи опікуном учня. Контролює як навчається її дитина та повністю відповідає за неї.

Учитель – людина, яка навчає інших людей (своїх учнів), передає їм певні знання про життя, що проводить навчальну та виховну роботу з учнями в загальноосвітніх школах.

Розклад – шкільний розклад являє собою таблицю для координації цих чотирьох елементів: номеру уроку, предмету навчання, доступна класна кімната та вчитель, який навчає учнів.

Адміністратор – це особа, яка здійснює роботу з реєстрації, контролю, занесення нових даних, редагування інформації та її видалення.

Відмітка(оцінка та пропуски) – це результат оцінювання кількісне вираження навчальних досягнень учнів в цифрах, буквах.

Клас – в котрому впроваджується навчальний процес з учнями.

Предмет – навчальні курси, що вивчаються в школах протягом у рамках навчального процесу під час уроків.

Повідомлення – інформація, яка містить текст і відправляється від класного керівника до батьків чи навпаки.

Пропуск – це результат, який реалізується у буквеному досягненні як результат пропуску уроків учнем.

Email розсилка – це результат, який відправляється батькам учнів після отримання їми відміток чи домашнього завдання.

Завантаження даних – це дія, яку можуть робити вчителі чи адміністратор для швидкого занесення інформації до системи.

1.3 Опис очікуваних результатів

Для зручного користування додатком користувачем потрібно слідувати певним атрибутам якості як: зручність використання, безпека, швидкість роботи додатку та інше.

Таблиця 1.1 – Нефункціональні вимоги

Вимога	Підтримка системою
Інтерфейс програмного забезпечення	Кросплатформність – веб-додаток може працювати з усіма можливими веббраузерами: Google Chrome, Internet Explorer та тому подібне
Комунікаційний інтерфейс	Для підтримки відправлення повідомлень на пошту батькам учнів треба взаємодіяти з поштовими скриньками. Веб-додаток працює тільки з gmail поштою
Апаратні вимоги	Для взаємодії з системою потрібні лише самі мінімальні налаштування, для підтримки браузера
Програмні вимоги	Використання веб-додатку можливо на: Windows, Linux, Mac
Завантаження сайту	Швидкість завантаження сайту становить секунду часу
Дружелюбність інтерфейсу	На веб-сайті повинно бути інтуїтивно зрозумілий інтерфейс для використання
Юзабіліті сайту	Усю потрібно інформацію можливо знайти у меню сайту
Безпека на сайті	Додаток не збирає інформації, уся інформація зберігається у базі даних

Розділ 2. ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Огляд використовуваних технологій та інструментів

Архітектурним шаблоном, який буде використовуватися для впровадження системи, було обрано MVC (Model-View-Controller), далі необхідно визначити мову програмування, фреймворк та інструменти, які дозволять виконати завдання у повній мірі.

Щоб побудувати рішення з розробки додатку для учнів, батьків, учителів та адміністратора нам знадобиться веб-сервер Node.js з мовою програмування JavaScript, фреймворк Express, СКБД PostgreSQL, NodeMailer, HTML, CSS та Bootstrap, тестування jest та supertest.

Для роботи з базою даних була обрана реляційна СКБД PostgreSQL (рисунк 2.1). PostgreSQL – це потужна система з об'єктно-реляційною базою даних з відкритим кодом, яка використовує та розширює мову SQL у поєднанні з багатьма функціями, які дозволяють безпечно зберігати та масштабувати найскладніші навантаження на дані [2].

Перевагами PostgreSQL є надійність, цілісність даних, розширюваність, працює на всіх основних операційних системах, допомагає управляти даними, незалежно від того, наскільки великий чи малий набір даних, є сумісним з ACID. Наприклад, можна визначити власні типи даних, створювати власні функції, навіть писати код з різних мов програмування без перекомпіляції бази даних.



Рисунок 2.1 – Логотип PostgreSQL

Node.js – це платформа (рисунок 2.2), заснована з використанням двигуна V8 (він здійснює переклад мови JavaScript в зрозумілий комп'ютеру), що вже перетворює сам JavaScript з вузької мови програмування в мову широкого призначення. Node дає можливість взаємодіяти з таким як введення-виведення за допомогою API, підключати до свого проекту сторонні пакети), які написані на різних існуючих вже мовах програмування. Node здебільшого використовується на стороні сервер як додатку. В основі Node.js покладено асинхронне програмування з неблокуючим введенням-виводом.[3]

Для того, щоб почати роботу у Node.js треба встановити допоміжні в роботі пакети. Для роботи є багато пакетів проте виберу тільки потрібні у проекті:

а) body-parser – це пакет, який дозволяє при відправці якихось складних даних зазвичай використовуються форми. Тому для отримання даних форм із запиту необхідно використовувати спеціальний пакет, щоб передати їх до запитів, де вже їх опрацюють.

б) express-session – це пакет, що дозволяє видати проміжок часу для користувача системи, який буде знаходитися на сайті. Сесія починається в момент заходу на сайт і закінчується при закритті вкладки браузера або при переході в межах поточної вкладки на інший ресурс, і дозволяє зберігати, наприклад, дані в діях користувача, які не втрачаються при переході на іншу сторінку.

в) `cookie-parser` – це пакет, що дозволяють зберігати на стороні клієнта деякі дані протягом заданого інтервалу часу, які при зміні сторінки або повторному заході на сайт допомагають ідентифікувати користувача і, наприклад, допомагають відновити стан попередньої сесії.

г) `ejs` – це пакет, який дозволяє передавати дані в різні шаблони, причому ці шаблони будуть мати розширення `.ejs`. У цих шаблонах ми зможемо виводити наш `html` код разом з вставленим в нього `js` кодом (змінними, виводити цикли і багато іншого). Буде такий шаблон сторінки, який буде змінюватися в залежності від переданих в нього даних.

г) `multer` – це пакет, що дозволяє нам у `Node.js` завантажувати файли на сервер. У `Node.js` для цього немає вбудованих інструментів, проте ми можемо використовувати додаткові спеціальні пакет. Одним з популярних подібних пакетів є `multer`, який працює поверх `Express`.

д) `socket.io` – це пакет, який забезпечує зв'язок між браузером та сервером у режимі реального часу, двонаправленим та подієвим.

е) `pg` – це пакет, який дозволяє нам працювати з нашою базою даних, а саме `PostgreSQL`. З пакетом ми можемо підключитися до нашої бази даних та використовувати запити до неї та робити інші маніпуляції.



Рисунок 2.2 – Логотип Node.js

`NPM` – це менеджер пакетів, який використовується у `Node.js` для завантаження сторонніх пакетів до себе проект. [4].

Менеджер пакетів по замовчуванню вже вбудований в Node.js і для того, щоб його використати потрібно в консолі прописати команду для встановлення пакету у додаток. З цим ми можемо використовувати пакети, які були написані на інших мовах програмування за допомогою JavaScript. Після встановлення пакету його версія та назва одразу записується у файл `package.json` з цим програміст може легко оновлювати свої пакети та уникати якихось змін.



Рисунок 2.3 – Логотип NPM

Express – це фреймворк для Node.js, який реалізовує шар функцій, необхідних для створення ефективних програм та API. Його використання значно скорочує написання коду, тобто, зменшується час розробки [5].

Node.js Express має готові функції обробки HTTP запитів, причому для кожного HTTP методу є своя функція, що особливо зручно при створенні REST – це далеко не єдина причина використання Express.

Метод `use()` використовується для створення проміжних оброблювачів – Middleware.

Для досягнення архітектурній моделі у рамках MVC з використанням фреймворку Express, нам потрібно буде реалізувати:

а) Модель – це те, що буде обробляти дані нашого застосування. У неї повинен бути доступ до об'єкта бази даних, що повертається об'єктом `PostgreSQLClient`. Наша модель також має мати метод для її розширення у разі необхідності.

б) Представлення буде виводити інформацію на екран. Говорячи простою мовою, представлення – це клас, який відправляє браузеру відповідь. Express

надає просту можливість це робити. Response-об'єкт – це обгортка, яка допомагає нам отримувати заданий нами потрібний API.

в) Контролер – це маршрут, який направляється на middleware-функції, яка вже приймає в себе request, response чи callback-функцію next. Ці три об'єкти вже оброблюються в самій функції та завдяки ним ми вже можемо обробляти дані.



Рисунок 2.4 – Логотип Express

JavaScript або JS – це об'єктно орієнтована-динамічна мова програмування (рисунок 2.5). Зазвичай JS використовується при розробці сценарію у веб-сторінках, які будуть використовувати в браузері користувач, асинхронний обмін чи вигляду веб-сторінки. [6]

JavaScript є основною мовою для програмування на Node.js. Завдяки його гнучкості та асинхронності, можливо створювати різні дії циклічно при роботі.

JS зазвичай класифікують як скриптову мову з динамічною типізацією. Також підтримує архітектурні властивості як динамічна та слабка типізація, автоматичний збір сміття для керування пам'яттю.

Мова JS використовується для:

- а) написання коду у веб-сторінках для надання їм гнучкості
- б) розробка з фреймворками як (Express, React, AngularJS);
- в) розробка на сервері Node.js;

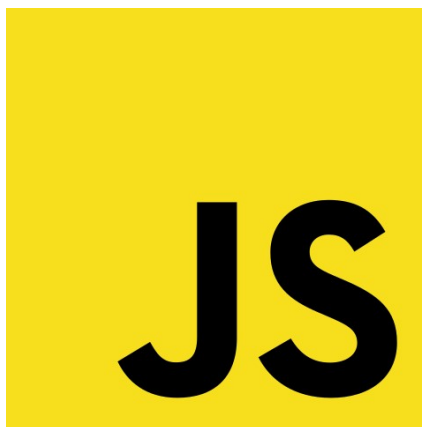


Рисунок 2.5 – Логотип мови програмування JavaScript

Nodemailer – це модуль, який дає нам можливість легко відправляти електронні листи на email при програмуванні на Node.js (рисунок 2.6).[7] Сервіс дозволяє нам не тільки здійснювати масову розсилку без ризику потрапити в чорний список SMTP, а й відстежувати доставку ваших листів. Для використання Nodemailer у своїй системі потрібно створити свій поштовий аккаунт на сервісі gmail з якого і буде відправлятися користувача системи листи. Можливості які надає нам модуль:

- а) Сильний фокус на безпеці;
- б) Підтримка Unicode для використання будь-яких символів, включаючи смайли;
- в) Підтримка Windows;
- г) Використання вмісту HTML, альтернативу простому тексту;
- г) Додати вкладення до повідомлень;
- д) Проксі для SMTP з'єднань;



Рисунок 2.6 – Логотип NodeMailer

Користувацький інтерфейс інформаційної системи подається у вигляді сайту за допомогою Bootstrap (HTML- та CSS-шаблон, включаючи JavaScript розширення) (рисунок 2.7).



Рисунок 2.7 – Логотип Bootstrap

Bootstrap є найпопулярнішою бібліотекою компонентів у світі Bootstrap є відкритим вихідним кодом для розробки з HTML, CSS і JS [8]. Швидкий прототип ідей користувачів, гнучка система з великими готовими компонентами і потужними плагінами, побудованими на jQuery.

Jest – це чудовий фреймворк тестування JavaScript з акцентом на простоті. Він працює з багатьма проектами, використовуючи фреймворк, який я використовую, а саме Express. [9]

Jest має такі особливості:

а) Тести паралельно проводяться, виконуючи їх у процесах для досягнення максимальної продуктивності;

б) Від it до expect – Jest має усі інструменти роботи в одному місці. Добре задокументований.

в) Можливість працювати з API, як GET або POST.

г) Чудова взаємодія з Express

Також з Jest будемо використовувати supertest, для виклику власне API.

2.2 Розробка структури бази даних

На основі аналізу вимог до системи необхідно створити концептуальну модель бази даних та спроектувати логічну схему цієї бази даних.

В рамках предметної області необхідно описати такі сутності:

Користувачі даної системи, основними користувачами виступають «Учні», «Батьки» та «Вчителі», які характеризуються на основі різних первинних даних. Спільними атрибутами даних трьох сутностей є: прізвище, ім'я, по-батьковій, пароль для входу та тип користувача, для отримання відповідних прав доступу. Окрім приведеного переліку учня характеризує: адреса, стать та дата народження. Вчителя, в свою чергу: дата народження адреса, стать. Привілеї видаються згідно зі таблицею «Ролі», яка містить ідентифікатор ролі та її назву ролі. Тип користувача «Гість» у таблиці не було приведений, адже інформація про незареєстрованих користувачів не буде записуватися та зберігається в самій базі даних. Екземпляр користувача з правами адміністратор системи відноситься до сутності «Вчителя», де буде зберігатися.

Журнал, в системі передбачення занесення відміток до журналу. Складається ця таблиця з відмітки, дати, учня та «навантаження вчителя».

Залежно від типу користувача та обмежень в неї вводять додавати дані може тільки той вчитель, який веде цей предмет.

Домашнє завдання, в системі передбачення занесення домашнього завдання до таблиці. Складається ця таблиця з домашнього завдання, дати за яку буде виставлене, та «навантаження вчителя». Залежно від типу користувача та обмежень в неї вводять додавати дані може тільки той вчитель, який веде цей предмет.

Розклад занять, атрибутами даної таблиці є: аудиторія, урок, день тижня та «навантаження вчителя». У розкладі представлена інформація для класів в яких навчаються учні.

Клас, сутність складається з цифри, букви класу та класного керівника цього класу.

Предмет, сутність у якій записані усі предмети, які можуть вестись та існують у школі.

Повідомлення, складається з батьків, учителя, дати-часу, повідомлення та типу.

Предмет_учитель_клас, таблиця є сполучною ланкою між таблицями (клас, учитель та предмет). Залежно від формування таблиці, ми можемо формувати так зване «навантаження вчителя» та коректування: розкладу, журналу, домашнього завдання.

У даній предметній області виділені такі сутності та їх атрибути (табл. 2.1):

Таблиця 2.1 – Опис сутностей таблиць реляційної моделі предметної області

Поле	Призначення	Обмеження
Об'єкт «sch.class»		
id_class	Ідентифікатор класу	PK
title	Цифра класу	NN, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
kurs	Буква класу	NN, А, Б, В, Г
teacher_class	Класний керівник	FK

Поле	Призначення	Обмеження
Об'єкт «sch.pupil»		
id_pupil	Ідентифікатор учня	PK
name	Ім'я учня	NN
name_otch	По-батьковій учня	NN
last_name	Серія та номер паспорта	NN
id_class	Клас у якому заходиться учень	FK
data_roz	Дата народження учня	NN
address	Адреса проживання	NN, місто, вулиця, будинок, квартира
phone	Телефон учня	U, NN
d_pol	Стать	NN, чоловік, жінка
role	Користувацька роль (учень)	FK
pass	Пароль для входу до системи	NN
id_family	Батьки учня	FK
Об'єкт «sch.family»		
id_family	Ідентифікатор батьків	PK
name	Ім'я батьків	NN
name_otch	По-батьковій	NN
last_name	Прізвище	NN
pass	Пароль для входу до системи	NN
role	Роль користувача (батьки)	FK
phone	Телефон батьків	U, NN
email	пошта	
Об'єкт «sch.teacher»		
id_teacher	Ідентифікатор вчителя	PK
name	Ім'я вчителя	NN
name_otch	По-батьковій	NN
last_name	Прізвище	NN
data_roz	Дата народження	NN
d_pol	Стать	NN, чоловік, жінка
address	Адреса проживання	NN, місто, вулиця, будинок, квартира

Поле	Призначення	Обмеження
phone	Телефон учителя	U, NN
role	Роль користувача	FK
pass	Пароль для входу в систему	NN
Об'єкт «sch.discipline»		
id_discipline	Ідентифікатор дисципліни	PK
title	Назва дисципліни	U, NN
Об'єкт «sch.ocenka»		
id_ocenka	Ідентифікатор журналу	PK
id_pupil	Учень для якого ставиться відмітка	FK
ocenka	Відмітка	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, н
id_discteacher	Яким вчителем та за який предмет поставлена відмітка	FK
date	Дата за яку виставляють відмітку	NN
Об'єкт «sch.homework»		
id_homework	Ідентифікатор домашнього завдання	PK
id_discteacher	Для якого класу, предмету та яким вчителем було задано дз	FK
hw	Домашнє завдання	
date	Дата за яку ставиться домашнє завдання	NN
Об'єкт «sch.raspisanue»		
id_raspisanue	Ідентифікатор рокзлад	PK
d_lesson	Урок на якому буде йти предмет	NN, 1, 2, 3, 4, 5, 6, 7, 8
id_discteacher	Для якого класу, предмету та яким вчителем буде вести	FK
d_day_week	День неділі за який йде урок	NN, Понеділок, Вівторок, Середа, Четверг, П'ятниця
d_classroom	Аудиторія, де проводиться урок	NN, 110, 111, 112, 113, 114, 115, 116,

Поле	Призначення	Обмеження
		117, 118, 119, 120, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419
Об'єкт «sch.discipline_teacher»		
id_discteacher	Ідентифікатор запиту	PK
id_discipline	Дисципліна	FK
id_teacher	Вчитель	FK
id_class	Клас	FK
is_deleted	Видалено чи ні	NN
Об'єкт «sch.role»		
id_role	Ідентифікатор користувача ролі	PK
role_name	Назва ролі	U, NN
Об'єкт «sch.message»		
id_message	Ідентифікатор повідомлення	PK
id_family	Батьки	FK, NN
id_teacher	Учитель	FK, NN
message	Повідомлення	NN
type11	Хто відправив повідомлення	NN

Першим етапом розробки є створення концептуальної моделі, заснованої на результатах аналізу предметної області.

Для інформаційної системи «Шкільний щоденник» концептуальна модель була побудована у вигляді діаграми сутність-зв'язок (рисунк 2.8).

У даній моделі наявні наступні типи зв'язку.

Між таблицями teacher, discipline та class встановлений зв'язок багато до багатьох. Такий тип зв'язку формалізується через створення відокремленої

таблиці `discipline_teacher`, яка містить в якості складеного первинного ключа зовнішні ключі, які посилаються на обидві сполучені таблиці.

Між іншими таблицями існує зв'язок один до багатьох, який формалізується шляхом додання атрибуту зовнішнього ключа в таблицю, відповідну N кратності зв'язку. Таким зв'язком пов'язані таблиці:

```
class та pupil (id_class);
pupil та oцenka (id_pupil);
message та family (id_family);
message та teacher (id_teacher)
role та pupil (role);
role та teacher (role);
role та family (role);
discipline_teacher та homework (id_discteacher);
discipline_teacher та oцenka (id_discteacher);
discipline_teacher та raspisanue (id_discteacher);
teacher та class (teacher_class);
pupil та family (id_family);
```


2.3 Архітектура застосунку

Для розробки додатку Шкільного щоденника було обрано триланкову архітектуру, що складається з серверу бази даних (рівень управління ресурсами), серверу додатків (рівень прикладного компоненту) та клієнтів (рівень представлення даних) (рисунок 2.9) [1]. Діаграму розгортання показана на рисунку .

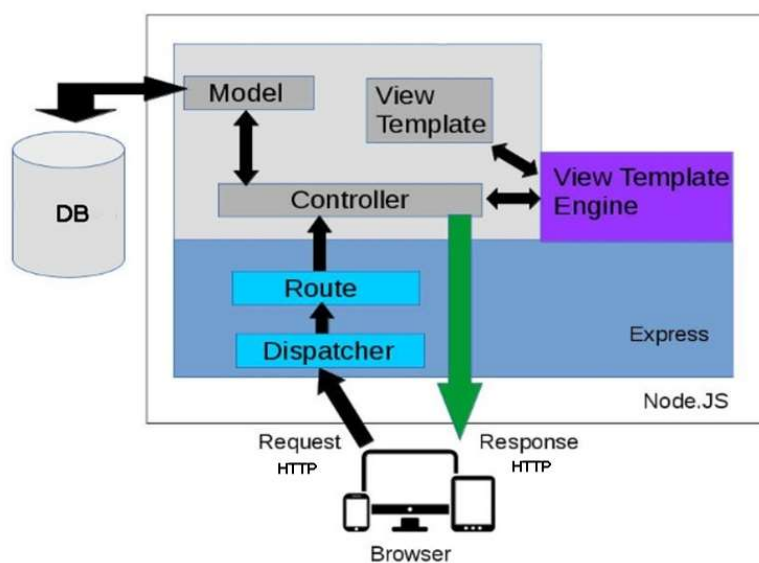


Рисунок 2.9 – Схема триланкової архітектури

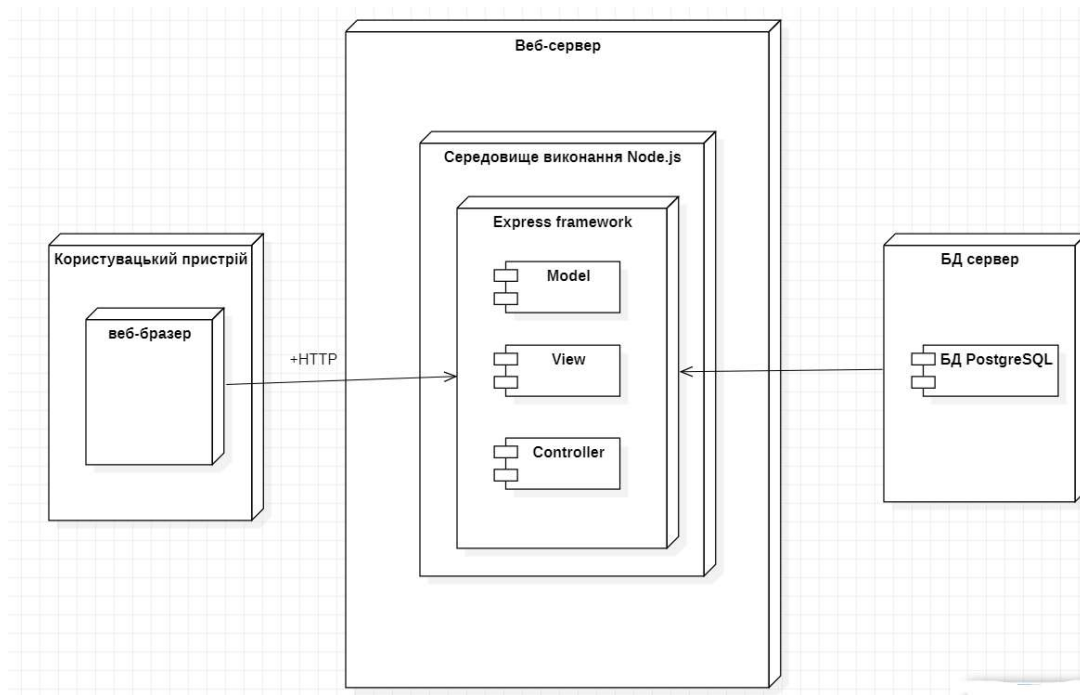


Рисунок 2.10 – Діаграма розгортання

Взаємодія між ними відбувається наступним чином: клієнт надсилає запит на перегляд сторінки який обробляється сервером додатку, після чого сервер додатків за потреби звертається до серверу бази даних та надсилає відповідь у вигляді запитуваної сторінки клієнту, а для підтримки сервісів було обрано сервісна орієнтований шаблон проектування REST.

REST – це архітектурний стиль програмного забезпечення для розподілених систем, який використовується для побудови веб-сервісів, і є дуже простим інтерфейсом управління інформацією без використання якихось додаткових внутрішніх прошарків. Кожна одиниця інформації однозначно визначається глобальним ідентифікатором, таким як URL.

Для проектування логіки додатку було обрано тришаровий архітектурний стиль у вигляді шаблону MVC, який дозволяє відокремити дані додатку, інтерфейсу користувача та управляючу логіку. MVC передбачає поділ на три окремих компонента: Model, що містить бізнес-логіку, включає методи вибірки, обробки та звертається напряму до бази даних; View (подання), що використовується для завдання зовнішнього інтерфейсу даних, отриманих з Model і Controller та містить HTML-розмітку; Controller відповідає за обробку запитів від користувача.

Для забезпечення виклику необхідного Controller виконується маршрутизація – сайт має спільну точку входу для кожного введеного URL, де проводиться його аналіз та виклик відповідного контролера. Схему взаємодії між Model, Controller та View наведено на рисунку 2.11.

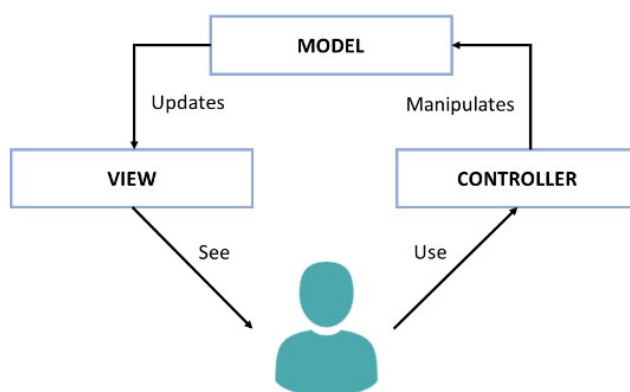


Рисунок 2.11 – Схема взаємодії між моделлю, контролером та поданням

Захист інформації реалізований як на рівні бази даних, так і на рівні додатку, підтримується декілька рівнів доступу користувачів.

Переваги моделі MVC: забезпечує хороше розділення користувацький інтерфейс репрезентації, даних і логіки, що призводить до легкої розробки та підтримки додатків, забезпечує хорошу перевірочну здатність системи.

Підсумовуючи вищенаведену інформацію, MVC є вдалим вибором через його гнучкий потенціал. Переглянути діаграму компонентів додатку можна на рисунку 2.12.

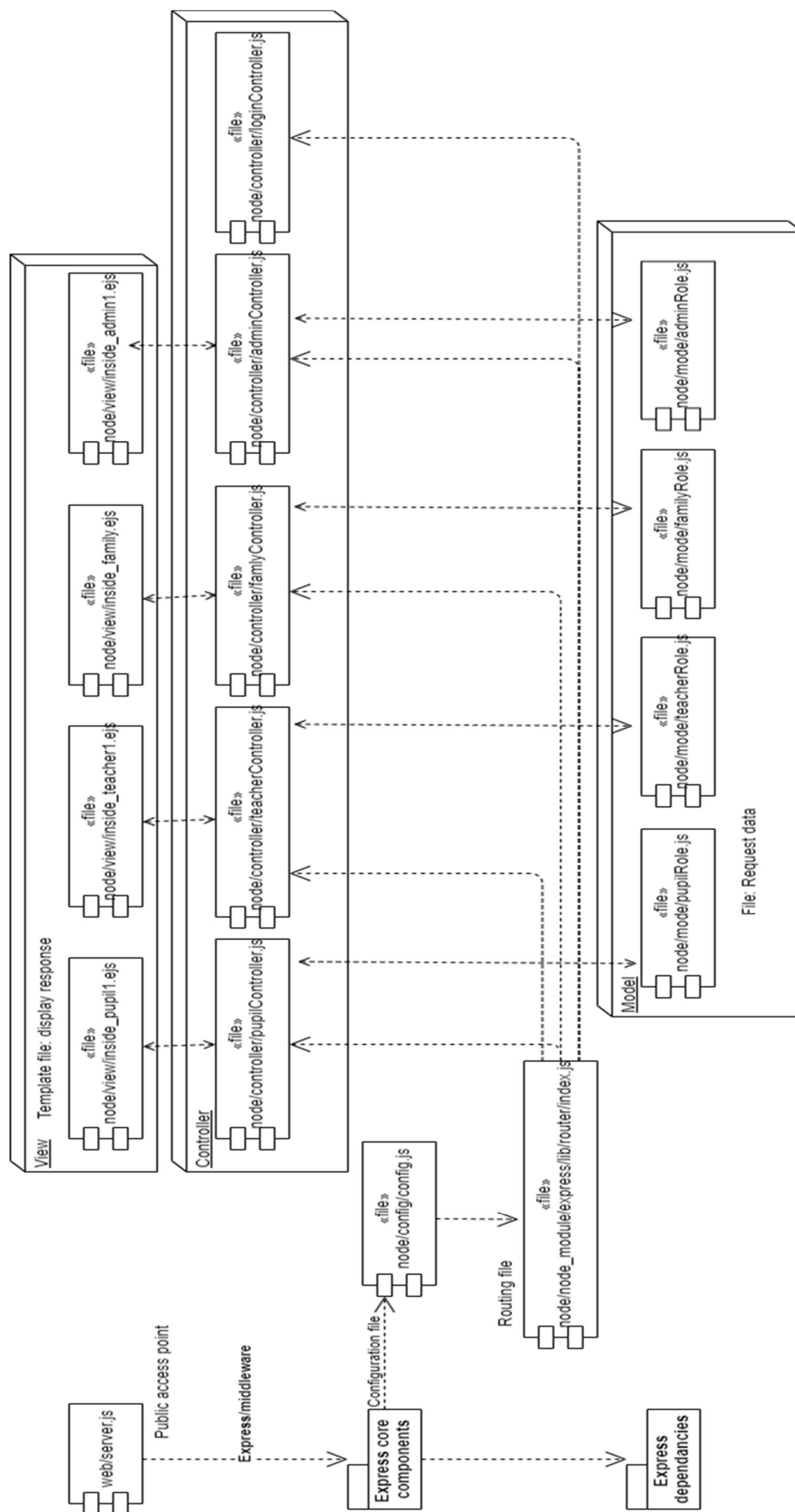


Рисунок 2.12 – Діаграма компонентів

2.4 Діаграми прецедентів користувачів

В результаті аналізу аналогів та проведеного опитування можливих користувачів було виявлено найбільш важливі функції для реалізації їх у веб-додатку (таблиця 2.2):

Таблиця 2.2 – Визначення функціональності системи

Роль	Задача	Вхідна інформація	Вихідна інформація
Учень	Перегляд свої оцінок учнем за проміжок дат	Дата з якої розпочати показ оцінок, до кінцевої дати	Перелік оцінок та предметів з датами виставлення
	Перегляд свого домашнього завдання учнем за проміжок дат	Дата з якої розпочати показ оцінок, до кінцевої дати	Перелік домашнього завдання та предметів з датами виставлення
	Перегляд розкладу уроків на кожен день тижня		Розклад уроків за: понеділок, вівторок, середу, четвергу та п'ятницю
	Вхід до персонального кабінету	Логін та пароль	Домашня сторінка учня
	Перегляд своїх середніх оцінок учнем		Середні оцінки за предмет
	Перегляд рейтингу учня по середніх оцінок за предмет	Предмет	Рейтинг учня за середньою оцінкою у класі
Вчитель	Зміна відміток учням певного класу, предмету за певну дату	Цифра класу, буква класу, предмет, дата, оцінка	Список учнів класу: прізвище ім'я, по-батьковій
	Виставлення відміток учням певного класу, предмету та дати	Цифра класу, буква класу, предмет, дата, оцінка	Список учнів класу: прізвище ім'я, по-батьковій
	Задання домашнього завдання учням певного класу, предмету за певну дату	Цифра класу, буква класу, предмет, дата, домашнє	Домашнє завдання було задано

Роль	Задача	Вхідна інформація	Вихідна інформація
		завдання	
	Зміна домашнього завдання учням певного класу, предмету за певну дату	Цифра класу, буква класу, предмет, дата, домашнє завдання	Домашнє завдання було змінено
	Перегляд середніх оцінок учнів у яких викладає	Цифра класу, буква класу, предмет	Середні оцінки за предмет
	Перегляд рейтингу учнів за середніми оцінками у класі	Цифра класу, буква класу, предмет	Рейтинг учнів по оцінкам за предмет
	Перегляд домашнього завдання або відміток класу за певний проміжок дат	Дата початку показу, дата кінця показу	Перелік домашнього завдання за заданими датами
	Перегляд власного розкладу		Розклад та тиждень
	Завантаження файлу з відмітками на сайті	CSV-файл з вже виставленими відмітками	Дані були завантаженні
	Завантаження файлу з домашнім завданням на сайті	CSV-файл з вже заданим завданням	Дані були завантаженні
	Відправка класним керівником повідомлення батькам учнів її класу	Прізвище ім'я, по-батьковій, повідомлення	Повідомлення з датою та часом
Батьки	Вхід до персонального кабінету	Логін та пароль	Домашня сторінка
	Перегляд відміток батьками їх дитини за проміжок дат	Дата початку показу, дата кінця показу	Перелік відміток та предметів з датами виставлення
	Перегляд домашнього завдання батьками їх дитини за проміжок дат	Дата початку показу, дата кінця показу	Перелік відміток та предметів з датами виставлення
Батьки	Отримання повідомлення на email, після виставлення їх дитині відмітки чи задання домашнього завдання	Дата початку показу, дата кінця показу	Повідомлення на пошті
	Перегляд розкладу уроків батьками їх дитини на день тижня	Цифра класу, буква класу, предмет,	Розклад уроків за: понеділок, вівторок,

Роль	Задача	Вхідна інформація	Вихідна інформація
		дата, домашнє завдання	середу, четвергу та п'ятницю
	Відправка повідомлення батьками учнів їх класному керівнику класу	Повідомлення	Повідомлення з датою та часом
	Вхід до персонального кабінету	Логін та пароль	Домашня сторінка батьків
	Перегляд своїх середніх оцінок дитини		Середні оцінки за предмет
	Перегляд рейтингу учня по середніх оцінок за предмет	Предмет	Рейтинг учня за середньою оцінкою у класі
Адміністратор	Реєстрація нових учнів	Ім'я, прізвище, по батьковій, адреса, стать, дата народження, клас, пароль, ПІБ батьків	Підтвердження реєстрації
	Завантаження розкладу занять	CSV-файл з класом, предметом, вчителем, днем тижня та кімнати	Дані були завантаженні
	Реєстрація нових батьків	Ім'я, прізвище, по батьковій, пароль, пошта	Підтвердження реєстрації
	Реєстрація нових вчителів	Ім'я, прізвище, по батьковій, адреса, стать, дата народження, пароль	Підтвердження реєстрації
	Завантаження нових користувачів з файлу	CSV-файл з вже заданим параметрами	Дані були завантаженні
	Завантажити файл для перевodu у наступний клас учнів	CSV-файл з вже заданим	Дані були завантаженні

Роль	Задача	Вхідна інформація	Вихідна інформація
		параметрами	
Адміністратор	Перегляд середніх оцінок усіх класів та учнів	Цифра, літера, предмет	Учні та їх середня оцінка за предмет у заданому класі
	Вхід до персонального кабінету	Логін та пароль	Домашня сторінка адміністратора
	Перегляд ранку по середніх оцінок усіх класів та учнів	Цифра, літера, предмет	Учні та їх рейтинг по середніх оцінка за предмет у заданому класі
	Видалення розкладу занять	Цифра класу, літера класу	Розклад який був попередньо завантажений до системи буде видалений
	Видалення користувачів	CSV-файл	Видалені учні-батьки чи вчителі
	Завантаження нових користувачів з файлу	CSV-файл з вже заданим параметрами	Дані були завантаженні

Для кращого ілюстрування програмних вимог була створена діаграма варіантів використання (прецедентів) . Діаграма описує роботу з додатком усіх користувачів та можливості системи та зовнішніх сервісів.

Найбільш детально проведено пошук акторів та випадки використання, і вони представлені в наступних таблицях. У таблиці 2.3 наведено аналіз акторів.

Таблиця 2.3 – Пошук та аналіз акторів

Актор	Короткий опис
Учень	Учень може переглядати свої оцінки, домашнє завдання, розклад уроків також може увійти до свого кабінету
Батьки	Батьки можуть переглядати дані своєї дитини таких як: оцінок, пропуски, розклад занять також вхід до свого кабінету та обмін повідомленнями з класним керівником дитини
Вчитель	Вчитель може: виставляти відмітки учня у яких вони ведуть, задавати домашнє завдання учням у яких веде, змінювати відмітки чи домашнє завдання, перегляд відміток чи домашнього завдання учнів, відправляти повідомленнями з батьками, завантажувати відмітки чи домашнє завдання через файл
Адміністратор	Адміністратор може: реєструвати користувачів системи, задання розкладу занять, переведення учнів у наступний клас, завантаження інформації за допомогою файлів
Система розсилки	Система розсилки може: відправлять повідомлення на поштовий e-mail батьків учнів за відмітки, домашнє завдання та повідомлення з чату

Загальний вигляд доступного функціоналу користувача вказано у вигляді UML діаграми прецедентів на рисунках 2.13, 2.14, 2.15 та 2.16.

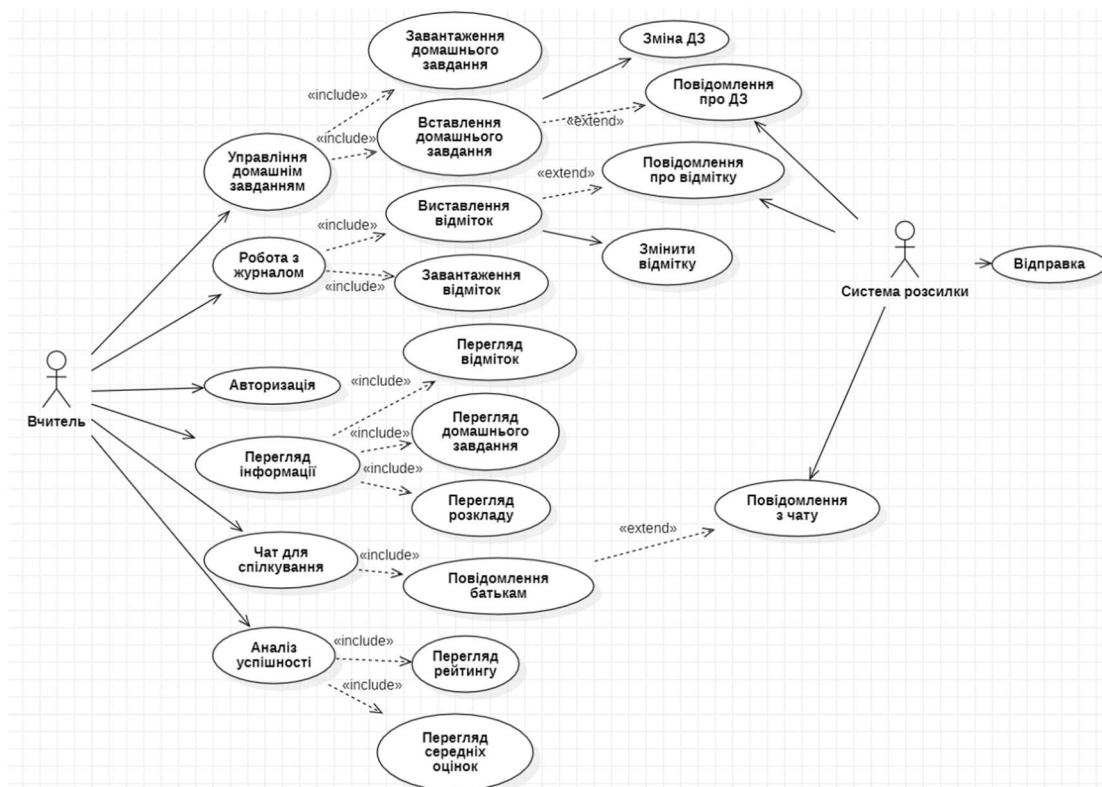


Рисунок 2.13 – Діаграма прецедентів для вчителя

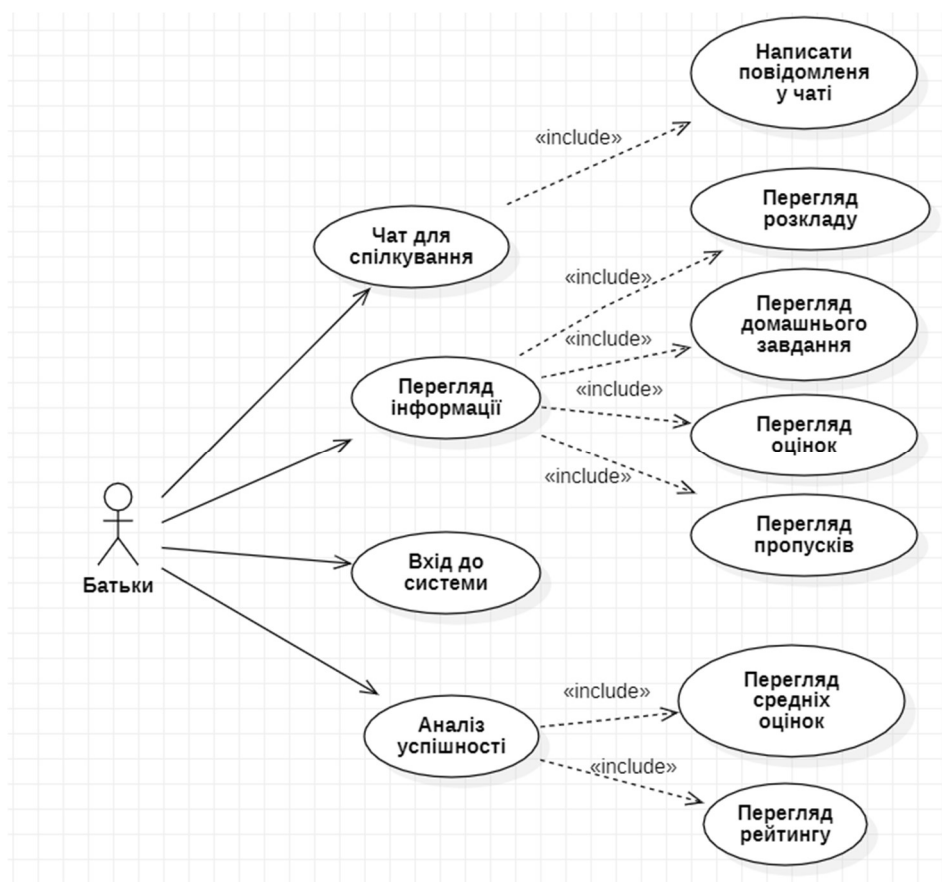


Рисунок 2.14 – Діаграма прецедентів для батьків



Рисунок 2.15 – Діаграма прецедентів для учня



Рисунок 2.16 – Діаграма прецедентів для адміністратора

2.5 Створення бази даних

В цьому підрозділі описані основні об'єкти бази даних та права доступу користувачів до неї. Повний код створення бази даних наведено у додатку А.

Усі перераховані нижче об'єкти бази даних `myschool` зберігаються у схемі `sch`. В рамках даної бази даних створена також схема `auth`, для входу у систему.

Для обмеження вводу даних у деякі з полів таблиць були створені користувацькі типи або домени, залежно від того чи іншого вимог атрибуту.

Домен – це обмежена підмножина значень даного типу. В базі даних були розроблені наступні домени:

- 1) `d_class_kurs` – містить дані про назви літер класу;;
- 2) `d_class_title` – містить дані про можливі цифри класу;
- 3) `d_classroom` – містить дані про назви класних кімнат;
- 4) `d_day_week` – містить дані про дні тижня;
- 5) `d_lesson` – містить дані про номери уроків;
- 6) `d_pol` – містить дані про статі;

Складений тип даних – це самостійно створений тип даних для використання в моєму варіанті як `address` користувачі, а саме: місто, вулиця, будинок, квартира.

Створення таблиць для реалізацію концептуальної моделі:

- 1) `pupil` – містить дані про учня. До якого класу відноситься, хто його батьки та роль користувача. Усі поля є обов'язковими до заповнення;
- 2) `teacher` – зберігає дані про вчителя та адміністратора системи, паролі та ролі користувача. Усі поля є обов'язковими до заповнення;
- 3) `family` – містить дані про батьків, учень яких навчається у школі. Усі поля є обов'язковими до заповнення;
- 4) `class` – таблиця, містить перелік класів, а саме цифри, букви та класного керівника цього класу. Усі поля є обов'язковими до заповнення;
- 5) `role` – таблиця, містить перелік ролей користувачів, які існують у системі. Усі поля є обов'язковими до заповнення;

- 6) *ocenka* – містить дані про відмітки, які були виставлені вчителем по датам, які були указані та за деякий предмет;
- 7) *discipline* – зберігає інформацію про дисципліни, яким можуть навчати у школі. Усі поля є обов'язковими до заповнення;
- 8) *homework* – містить дані про домашнє завдання, яке було виставлене вчителем по датам, які були указані за деякий предмет та деякому класу;
- 9) *raspisanue* – містить дані про урок, аудиторію та день неділі за яку будете проведено вчителем предмет у якомусь класі. Усі поля є обов'язковими до заповнення;
- 10) *discipline_teacher* – містить дані про вчителя, клас, предмет Усі поля є обов'язковими до заповнення;
- 11) *message* – містить дані про повідомлення та хто його відправив. Усі поля є обов'язковими до заповнення;

Представлення – віртуальна таблиця, що представляє собою запит який буде підставлений як підзапит при використанні .

Були створені такі Представлення в базі даних:

- 1) *idv_pupil_phone* – зберігає дані учня;
- 2) *show_rasp_vt1_phone* – зберігають дані про розклад уроків;
- 3) *idvt_family_phone* – дані батьків;
- 4) *show_rasp_family_phone* – дані про розклад уроків;
- 5) *idv_teacher_phone* – дані про вчителів;
- 6) *choice* – зберігає інформацію про учнів та в якому класі вони є;
- 7) *show_nagr_phone* – навантаження вчителя;
- 8) *mark_email* – пошти батьків для відміток;
- 9) *homework_email* – пошти батьків для домашнього завдання;
- 10) *login_phone* – містить дані усіх номерів телефонів користувачі для входу до системи. Представлення знаходиться у схемі *auth*.
- 11) *message_email* – пошти батьків для відправки повідомлень.

Збережена процедура - об'єкт бази даних, що представляє собою набір SQL-інструкцій, який компілюється один раз і зберігається на сервері. Збережені

процедури можуть також містити оголошені змінні для обробки даних і курсорів, які дозволяють організувати цикл з кількох рядках в таблиці.

Збережені процедури, які були реалізовані в додатку

- 1) `delete_rasp_class( num character varying, let character varying)` – видалення розкладу класу;
- 2) `import_csv_dz( teach integer, file_nam text)` – завантаження домашнього завдання;
- 3) `import_csv_ocenka( teach integer, file_nam text)` – завантаження відміток;
- 4) `import_csv_reg_family( file_nam text)` – завантаження батьків;
- 5) `import_csv_reg_pupil(file_nam text)` – завантаження учнів;
- 6) `import_csv_reg_teacher( file_nam text)` – завантаження вчителів;
- 7) `import_next_class( file_nam text)` – завантаження учнів для переведення у наступний клас;
- 8) `import_rasp( file_nam text)` – завантаження розкладу;
- 9) `message_family(fam integer, mes character varying)` – повідомлення батьки;
- 10) `message_family_name( teach integer)` – перегляд повідомлень;
- 11) `message_show_all_family1(teach integer)` – перегляд усіх повідомлень;
- 12) `sch.message_show_teach2( teach integer, name_ff character varying, last_nff character varying, last_off character varying)` – перегляд повідомлень з батьками;
- 13) `message_teacher( teach integer, mes character varying, name_ff character varying, last_nff character varying, last_off character varying)` – зробити повідомлення вчителем;
- 14) `teach_ocenk_phone1( oc character varying, ddate date, num character varying, let character varying, dis character varying, teach integer, l_name character varying)` – виставлення оцінок вчителем;
- 15) `teach_dz2_phone( dz character varying, ddate date, num character varying, let character varying, dis character varying, teach integer)` – виставлення домашнього завдання вчителем;

16) teach_ocenk_up_phone1(oc character varying, ddate date, num character varying, let character varying, dis character varying, teach integer, l_name character varying) – зміна оцінки;

17) message_teacher(teach integer, mes character varying, name_ff character varying, last_nff character varying, last_off character varying) – зробити повідомлення вчителем;

18) show_ocenk_pupil_ot_do(id_pupi integer, date_ot date, date_do date)– перегляд оцінок учнем;

19) show_ocenk_class_ot_do1(title_c character varying, kurs_c character varying, disc_p character varying, ot_date date, do_date date) – перегляд оцінок класу;

20) show_ocenk_family_ot_do(id_fami integer, date_ot date, date_do date) – перегляд оцінок батьками;

21) message_teacher(teach integer, mes character varying, name_ff character varying, last_nff character varying, last_off character varying) – зробити повідомлення вчителем;

22) show_dz_pupil_ot_do(pupi integer, date_ot date, date_do date) – перегляд домашнього завдання учнем;

23) show_dz_family_ot_do(fami int, date_ot date, date_do date) – перегляд домашнього завдання батьками;

24) show_dz_class_ot_do(n character varying, l character varying, d character varying, da_ot date, da_do date) – перегляд домашнього завдання класу;

25) show_avg_disc_pupil(id_pupi integer) – середні оцінки учня;

26) show_avg_disc_family(fam_phone integer) – середні оцінки батьки;

27) show_avg_disc_teach_s_dnl(teach_phone integer, disc character varying, num character varying, let character varying) – середні оцінки вчитель;

28) show_avg_disc_admin_s_dnl(disc character varying, num character varying, let character varying) (fam_phone integer) – середні оцінки адміністратор;

29) rank_pupil(phon integer, num character varying, let character varying, disc character varying) – рейтинг учня за предмет;

30) rank_family(phon integer, num character varying, let character varying, disc character varying) – рейтинг батьків за предмет;

31) rank_teacher (phon integer, num character varying, let character varying, disc character varying) – рейтинг вчитель за предмет;

32) rank_admin (num character varying, let character varying, disc character varying) – рейтинг адміністратор за предмет;

Тригер це збережена процедура особливого типу, яку користувач не викликає явно, а використання якої обумовлено настанням визначеної події (дії) у реляційній базі даних.

Тригери, які були реалізовані в додатку:

- 1) sch.ocenka_3() – для заборони виставлення оцінки на одно і ту саму дату;
- 2) sch.dz_3() – для заборони виставлення домашнього завдання на одно і ту саму дату;
- 3) sch.raspisanue_3() –заборона ставити вчителя в різні класи на один урок;
- 4) sch.no_more() – заборона ставити відмітки чи завдання на майбутнє;

2.6 Програмна реалізація системи

Бекенд сайту написаний мовою JavaScript із допомогою фреймворку Express, фронтенд – HTML/CSS із використанням фреймворку Bootstrap. Фрагменти програмного коду знаходяться у Додатку А

Файлова система проекту складається з наступних папок (рис. 2.17):

- а) view – у папці знаходяться усі ejs (html) сторінки;
- б) public – у папці знаходяться css стилі, javascript скрипти, фото, календар з датами та bootstrap стилі;
- в) controller – в папці можна знаходяться п'ять контролерів, для взаємодію у системі:
 - 1) loginController – відбувається авторизація користувача;
 - 2) pupilController – відповідає за дії учня у його кабінеті;
 - 3) familyController – відповідає за дії батьків у його кабінеті;

- 4) adminController – відповідає за дії адміністратора у його кабінеті;
- 5) teacherController – відповідає за дії вчителів у його кабінеті;
- г) model – у папці знаходяться файли для з'єднується з базою даних:
 - 1 – звертається до бази даних під роллю адміністратор;
 - г) config – знаходиться файл з параметрами для підключення до бази даних та порту серверу;
- д) __test__ – папка, де можливо знайти файл з тестуванням функцій системи;
- е) node_modules – одна з основних папок, де знаходиться усі модули та пакети які були підключені до мого проекту;
- є) package.json – файл, де знаходиться версії та імена усіх використовуючих пакеті у системі;
- ж) server.js – точна старту системи;

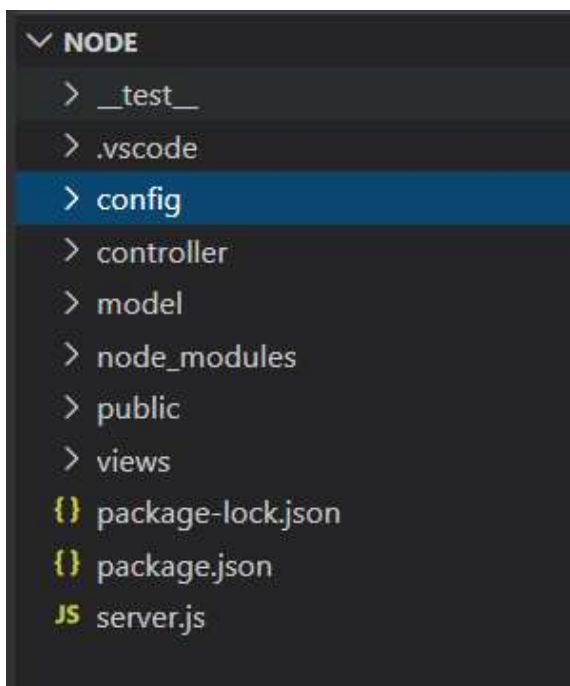


Рисунок 2.17 – Вміст папки node

Розділ 3. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ ТА АНАЛІЗ ОТРИМАНИХ ДАНИХ

3.1 Перевірка працездатності функціоналу вчителя

При першому потраплянні на сайт користувач буде бачити головну сторінку сайту з посиланнями на контакти та про сервіс. Зверху справа ми можемо побачити на (рисунок 3.1) є посилання на сторінку авторизації.

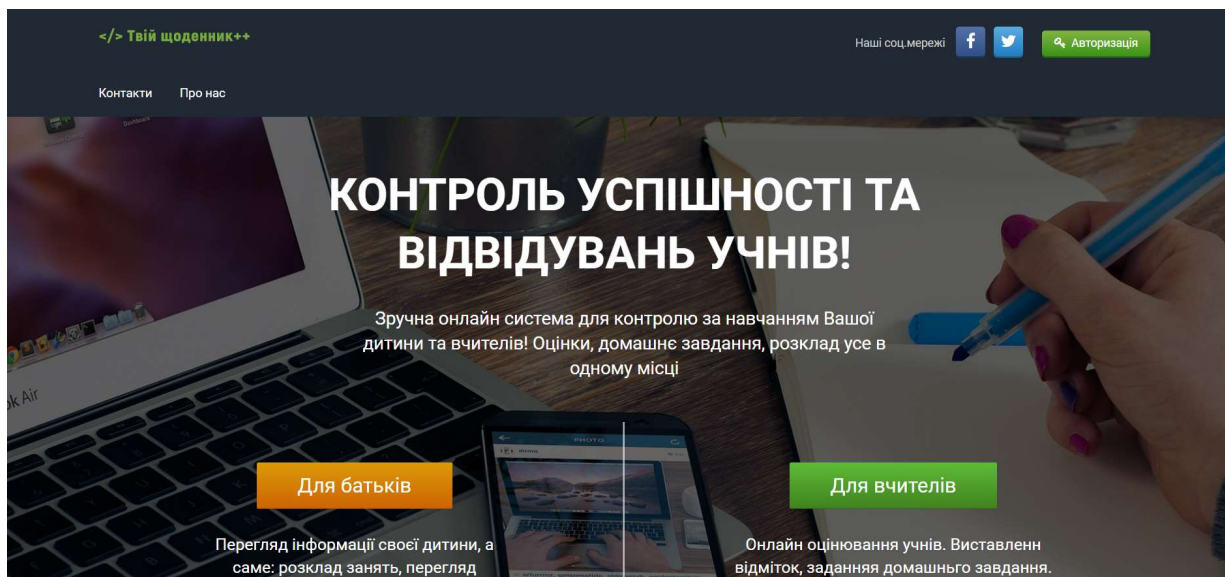


Рисунок 3.1 – Головна сторінка веб-сервісу

Після натискання на авторизацію користувач потрапляє на сторінку з вибором ролі під якою авторизуватися: учень, батьки, вчитель (рисунок 3.2).

Натиснувши на одну з кнопок і побачимо форму авторизації (рисунок 3.3), де треба ввести логін та пароль та натиснути кнопку увійти.

Після авторизації кожен користувач потрапляє до особистого кабінету. Далі опишемо дії вчителя при використанні вчителя.

Після авторизації вчитель потрапляє до свого персонального кабінету, де вже може виконувати дії які можна побачити на рисунку 3.4

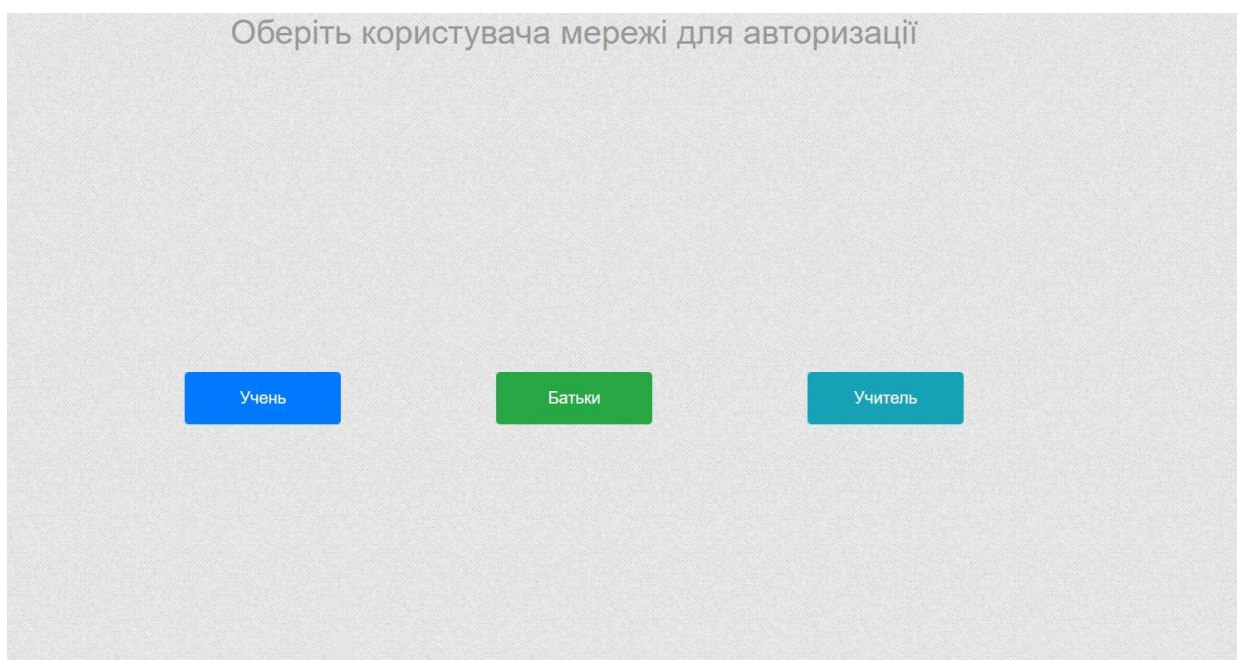


Рисунок 3.2 – Сторінка вибору для входу у систему

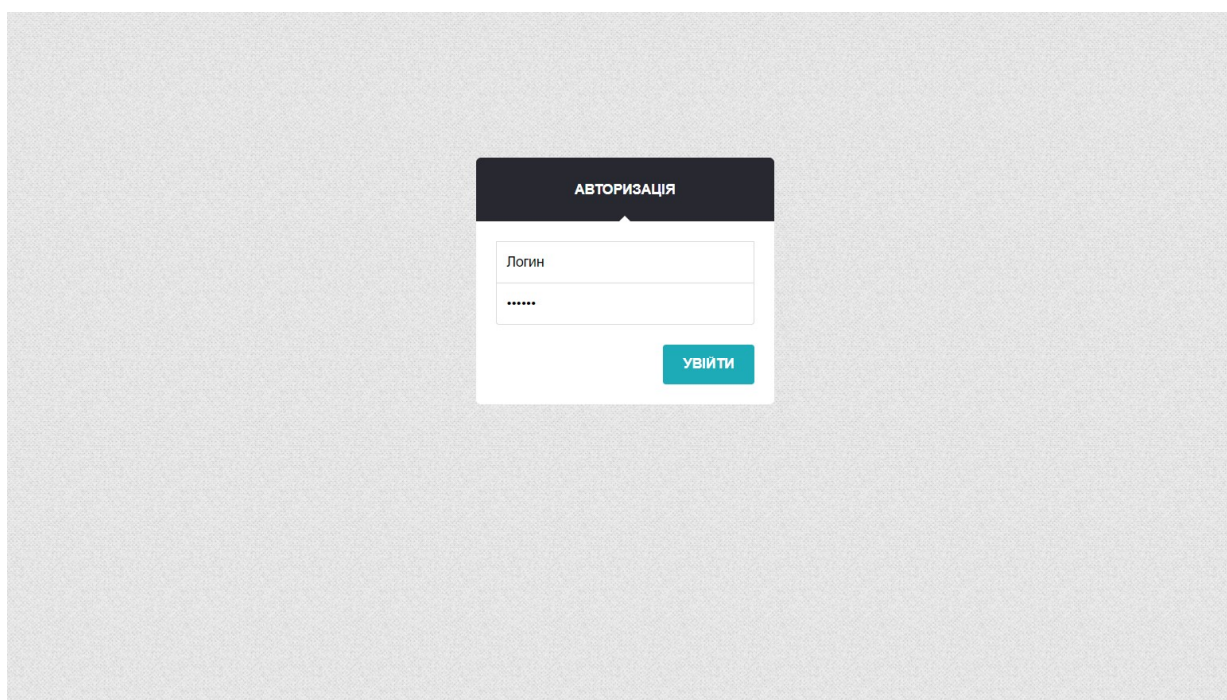


Рисунок 3.3 – Сторінка авторизації



ЛИТВИНЕНКО КСЕНІЯ
 Олеговна

Вийти з системи

Профіль учителя

Виставлення відміток або ДЗ

Розклад вчителя

Відправка повідомлення

Перегляд виставлення

Аналіз успішності

Профіль

Учитель	Профіль
ППП	Литвиненко Ксенія Олеговна
Адреса	(г.Одеса, "провулок Азмундсена", 55,17)
Стать	Жінка
Класний керівник класу	1-А

Рисунок 3.4 – Персональний кабінет учителя

Після натискання на виставлення відміток, учитель бачить, що це можна робити двома способами – це завантаження оцінок через CSV-файл або виставлення самостійно (рисунок 3.5).



ЛИТВИНЕНКО КСЕНІЯ
 Олеговна

Вийти з системи

Профіль учителя

Виставлення відміток або ДЗ

Розклад вчителя

Відправка повідомлення

Перегляд виставлення

Аналіз успішності

Виставлення відміток

1

▼

A

▼

Математика

▼

Дата виставлення:

Рисунок 3.5 – Виставлення відміток учитель

Якщо хочемо ввести самостійно, то обираємо клас, літеру, предмет та дату, а вже після натискаємо виставити та нам відкривається нова сторінка (рисунки 3.6) з учнями класу, де вже можемо ставити відмітки чи змінювати.

Виставлення оцінок:

Прізвище: Виставити оцінку: Змінити оцінку

Прізвище: Виставити оцінку: Виставити оцінку

Прізвище	Ім'я	По-батькові
Геначук	Ганна	Олеговна
Малиш	Олександра	Сергіївна
Мельник	Василь	Васильович
Шевченко	Іван	Кастимирович
Коваленко	Сергій	Іванович
Бондаренко	Олег	Сергійович

Рисунок 3.6 – Виставлення самостійно відміток

У сторінки з виставлення домашнього завдання майже однаковий дії, проте домашнє завдання можливо задати одразу для всього класу (рисунки 3.7).



ЛИТВИНЕНКО КСЕНІЯ
Олеговна

[Вийти з системи](#)

Профіль учителя
Виставлення відміток або ДЗ
Розклад вчителя
Відправка повідомлення
Перегляд виставлення
Аналіз успішності

Задання домашнього завдання

Файл не выбран

ОТПРАВИТЬ

Ци... ▾

Лт... ▾

Предмет ▾

Дата задня:

ЗАДАТИ

Рисунок 3.7 – Виставлення домашнього завдання

На сторінці розкладу занять вчителя можемо бачити його розклад з понеділка по п'ятницю (рисунок 3.8).


ЛИТВИНЕНКО КСЕНІЯ
 Олеговна
 Вийти з системи

Профіль учителя
 Виставлення відміток або ДЗ
 Розклад вчителя
Відправка повідомлення
 Перегляд виставлення
 Статистика навчання

Розклад занять

Понеділок

№	Предмет	Викладач	№каб.
2	Математика	1-А	111
3	Читання	1-А	111
4	Укр.мова	1-А	111

Вівторок

№	Предмет	Викладач	№каб.
1	Математика	1-А	111
2	Читання	1-А	111
3	Укр.мова	1-А	111

Середа

№	Предмет	Викладач	№каб.
1	Читання	1-А	111
2	Укр.мова	1-А	111
3	Математика	1-А	111
4	Навколишній світ	1-А	111

Четвер

№	Предмет	Викладач	№каб.
1	Читання	1-А	111
3	Укр.мова	1-А	111
4	Математика	1-А	111

П'ятниця

№	Предмет	Викладач	№каб.
1	Укр.мова	1-А	111
2	Математика	1-А	111
3	Навколишній світ	1-А	111

Розклад уроків

№	Час
1-й урок	8-30-9-15
2-й урок	9-25-10-10
3-й урок	10-20-11-05

Рисунок 3.8 – Виставлення домашнього завдання

Для відправлення повідомлення вчителем батькам, він повинен бути класним керівником учня. Учителю треба перейти на сторінку повідомлення та обрати одну з двох дій (рисунок 3.9). «Переглянути чи є повідомлення» це вчитель може переглянути відправили йому повідомлення хтось з батьків або він може сам обрати батьків та написати їх.



ЛИТВИНЕНКО КСЕНІЯ
 Олександрівна

Вийти з системи

Профіль учителя

Виставлення відміток або ДЗ

Розклад вчителя

Відправка повідомлення

Перегляд виставлення

Аналіз успішності

Повідомлення

Ім'я

Прізвище

По-батько...

ВІДКРИТИ ДІАЛОГ

ПОДИВИТИСЯ ЧИ Є ПОВІДОМЛЕННЯ

Рисунок 3.9 – Сторінка повідомлень

Діалог з батьками та можливість написати йому повідомлення (рисунок 3.10).

Чат повідомлень

Від Олеженко Олег Олегович
 Дякую за відповідь, будемо намагатися так продовжувати

Від Литвиненко Ксенія Олександрівна
 Ви можете пишатися нею !

Від Литвиненко Ксенія Олександрівна
 В мене ніхтл дорікань до неї немає. З однокласниками дружельюна та гарна співорозмовниа

Від Литвиненко Ксенія Олександрівна
 Добрий. Ганна дуже гарно себе поводити у школі

Від Олеженко Олег Олегович
 Добрий день, хочу дізнатися як веде себе моя донька у школі

Відслати

Рисунок 3.10 – Чат повідомлень

Учитель може також переглядати відміток та домашнє завдання, які виставив. Для цього треба перейти на сторінку перегляд відміток (рисунок 3.11).

Рисунок 3.11 – Перегляд відміток

Після введені двох дат з яких відбудеться перегляд оцінок, тоді з'являється нова сторінка з оцінками учня та датою виставлення переглянемо на рисунку 3.12.

Перегляд оцінок класу:

Прізвище	Ім'я	По-батькові	Оцінка	Дата
Геначук	Ганна	Олеговна	12	2023-04-03
Геначук	Ганна	Олеговна	9	2023-04-06
Геначук	Ганна	Олеговна	н	2023-04-07
Геначук	Ганна	Олеговна	9	2023-04-30
Геначук	Ганна	Олеговна	8	2023-04-29
Геначук	Ганна	Олеговна	7	2023-04-28
Геначук	Ганна	Олеговна	9	2023-04-09
Малиш	Олександра	Сергіївна	5	2023-04-03
Мельник	Василь	Васильович	8	2023-04-03
Коваленко	Сергій	Іванович	9	2023-04-13
Бондаренко	Олег	Сергійович	10	2023-04-13
Кисаєв	Михайло	Михайлович	11	2023-04-15

Рисунок 3.12 – Перегляд оцінок по датам

Для перегляду домашнього завдання потрібно зробити аналогічні дії. Обрати проміжок двох дат з яких відбудеться перегляд та натиснути кнопку.

Ще вчитель може переглядати середні оцінки учнів класу у якого навчає. Для цього треба перейти на «статистику навчання» та обрати середні оцінки, де вже ввести предмет та клас та отримати перелік середніх оцінок учнів (рисунок 3.13).

Середні оцінки учнів за предмет

1	▼	Літ...	▼	Предмет	▼
ПОДИВИТИСЯ					
ППП	Предмет	Середня оцінка			
Мельник В.В.	Математика	9.38			
Омно В.О.	Математика	7.40			
Геначук Г.О.	Математика	8.07			
Шевченко І.К.	Математика	7.33			
Куліш М.В.	Математика	8.29			
Ткаченко М.О.	Математика	10.50			
Кисасв М.М.	Математика	10.00			
Кравченко Н.А.	Математика	10.80			
Бондаренко О.С.	Математика	10.14			
Малиш О.С.	Математика	9.14			
Ковальчук О.А.	Математика	9.25			
Коваленко С.І.	Математика	10.33			

Рисунок 3.13 – Перегляд середніх оцінок

Також крім середніх оцінок можна переглянути рейтинг по предмету у класі учнів (рисунок 3.14).

Переглянути рангу учнів за предмет

Ци... ▼
Літ... ▼
Предмет ▼

ПОДИВИТИСЯ

ІПП	Предмет	Ранк
Кравченко Н.А.	Математика	1
Ткаченко М.О.	Математика	2
Коваленко С.І.	Математика	3
Бондаренко О.С.	Математика	4
Кисасв М.М.	Математика	5
Мельник В.В.	Математика	6
Ковальчук О.А.	Математика	7
Малиш О.С.	Математика	8
Куліш М.В.	Математика	9
Геначук Г.О.	Математика	10
Омно В.О.	Математика	11
Шевченко І.К.	Математика	12

Рисунок 3.14 – Перегляд рейтингу

3.2 Перевірка працездатності функціоналу учня та батьків

Далі, на рис. 3.15..3.19 показані результати випробування функціоналу для учня та батьків.

ГЕНАЧУК ГАННА
Олеговна

Вийти з системи

Профіль учня
Домашнє завдання
Перегляд оцінок
Розклад занять
Статистика навчання

Профіль

Учень	Профіль
Прізвище	Геначук
Ім'я	Ганна
По-батьковій	Олеговна
Адреса	(м.Одеса, вул.Варп.5,1)
Стать	Жінка
Дата народження	2012-09-25
Навчається у	1-А

Рисунок 3.15 – Персональний кабінет учня



ГЕНАЧУК ГАННА
 Оляговна

Вийти з системи

Профіль учня
 Домашнє завдання
 Перегляд оцінок
 Розклад занять
 Статистика навчання

Перегляд домашнього завдання

Дата від:
 Дата до:

Рисунок 3.16 – Перегляд домашнього завдання учня

Розклад занять					
Понеділок			Вівторок		
№	Предмет	Викладач	№	Предмет	Викладач
1	Музика	Романюк І. І.	1	Математика	Литвиненко К. О.
2	Математика	Литвиненко К. О.	2	Читання	Литвиненко К. О.
3	Читання	Литвиненко К. О.	3	Укр.мова	Литвиненко К. О.
4	Укр.мова	Литвиненко К. О.	4	Фізичне виховання	Мартинюк М. М.
			5	Малювання	Ткач С. П.
Середа			Четвер		
№	Предмет	Викладач	№	Предмет	Викладач
1	Читання	Литвиненко К. О.	1	Читання	Литвиненко К. О.
2	Укр.мова	Литвиненко К. О.	2	Фізичне виховання	Мартинюк М. М.
3	Математика	Литвиненко К. О.	3	Укр.мова	Литвиненко К. О.
4	Навколишній світ	Литвиненко К. О.	4	Математика	Литвиненко К. О.
П'ятниця			Розклад уроків		
№	Предмет	Викладач	№	Час	
1	Укр.мова	Литвиненко К. О.	1-й урок	8-30-9-15	
2	Математика	Литвиненко К. О.	2-й урок	9-25-10-10	
3	Навколишній світ	Литвиненко К. О.	3-й урок	10-20-11-05	
4	Технологія	Костенко В. В.	4-й урок	11-25-12-10	
			5-й урок	12-20-13-05	

Рисунок 3.17 – Розклад занять учня



ГЕНАЧУК ГАННА
Олеговна

Вийти з системи

Профіль учня

Домашнє завдання

Перегляд оцінок

Розклад занять

Статистика навчання

Середня оцінка учня по предметам

Предмет	Середня оцінка
Математика	8.44
Музика	10.47
Навколишній світ	9.38
Укр. мова	8.77
Читання	9.45

Рисунок 3.18 – Перегляд середніх оцінок учнем

1 ▾

A ▾

предмет ▾

ПОДИВИТИСЯ

Предмет	Ранк
Математика	10

Рисунок 3.19 – Перегляд рейтингу за предмет учнем

Батьки мають такий самий інтерфейс перегляд з додаванням відправити повідомлення батькам та перегляду відміток одразу з пропусками та оцінками.

3.3 Перевірка функціоналу адміністратора

Тепер перевіримо функції адміністратора системи (рис. 3.20..3.23).

Як можна побачити, результати роботи системи співпадають з очікуваними.


ГОНЧАР ВАЛЕНТИНА
 Іванівна
 Вийти з системи

Профіль адміна
 Реєстрація користувачів
 Робота з розкладом
 Перевести у наступний клас
 Статистика Навчання

Профіль

Адміністратор	Профіль
ПІБ	Гончар Валентина Іванівна
Адреса	(г.Одеса, "вул. Своя", 14, 221)
Стать	Жінка

Рисунок 3.20 – Профіль адміністратора


ГОНЧАР ВАЛЕНТИНА
 Іванівна
 Вийти з системи

Профіль адміна
 Реєстрація користувачів
 Робота з розкладом
 Перевести у наступний клас
 Статистика Навчання

Реєстрація

Выберите файл | Файл не выбран
 РЕЄСТРАЦІЯ ВЧИТЕЛІВ

Выберите файл | Файл не выбран
 РЕЄСТРАЦІЯ БАТЬКІВ

Выберите файл | Файл не выбран
 РЕЄСТРАЦІЯ УЧНІВ

РЕЄСТРАЦІЯ ВРУЧНУ КОРИСТУВАЧІ

Рисунок 3.21 – Реєстрація користувача адміністратором

Завантаження розкладу
 Видалення розкладу

Выберите файл | Файл не выбран
 ЗАВАНТАЖИТИ РОЗКЛАД

Ця...
 Літ...
 ВИДАЛИТИ

Рисунок 3.22 – Створення та видалення розкладу адміністратором

Перевести у наступний клас учнів

Выберите файл

Файл не выбран

ЗАВАНТАЖИТИ УЧНІВ

Рисунок 3.23 – Переведення у наступний клас адміністратором

ВИСНОВКИ

У даній роботі було проаналізовано предметну область і на основі цього, виділено основні задачі, які вирішує спроектований веб-сервіс контролю успішності та відвідувань учнів «Шкільний щоденник».

Під час аналізу предметної області виконано аналіз аналогів, визначено проблеми цієї області та необхідні задачі, які потребують користувачі системи. Таким чином, розроблюваний веб-сервіс надає багато можливостей для користувачів у своїй сфері використання. Саме тому, робота є актуальна та зрозуміла у використанні.

В ході роботи були втілені усі поставлені задачі, які були закріплені для реалізації у ході аналізу. Інтерфейс веб-сайту був зроблений максимально простим у використанні, щоб користувач будь-якого віку міг без ніяких проблем використовувати цей веб-сервіс.

Поставлені задачі виконані й система готова до роботи. У перспективах створення мобільного додатку, впровадження відеоконференцій для проведення уроків онлайн при неможливості їх проведенні у школі та впровадження у шкільні навчальні заклади.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MVC – триланкова архітектура [Електронний ресурс] – Режим доступу: <http://xiaoblog.digibrady.com/features/nodejs-mvc-example-implementation-with-xiao-blog-engine>
2. PostgreSQL 12 – Documentation by The PostgreSQL Global Development Group [Електронний ресурс] – Режим доступу: <https://www.postgresql.org/docs/12/index.html>
3. Node.js – Documentation [Електронний ресурс] – Режим доступу: <https://nodejs.org/uk/docs>
4. NPM – Documentation [Електронний ресурс] – Режим доступу: <https://docs.npmjs.com>
5. Express – Керівництво використання [Електронний ресурс] – Режим доступу: <https://expressjs.com/ru/guide/routing.html>
6. JavaScript – Documentation [Електронний ресурс] – Режим доступу: <https://devdocs.io/javascript>
7. NodeMailer – Керівництво використання [Електронний ресурс] – Режим доступу: <https://nodemailer.com/about>
8. Bootstrap – Documentation by Bootstrap 4 [Електронний ресурс] – Режим доступу: <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
9. Jest – Documentation by [Електронний ресурс] – Режим доступу: <https://jestjs.io/docs/en/getting-started.html>

ДОДАТКИ

Додаток А. ПРОГРАМНИЙ КОД

А.1 Код бази даних

```
// Створення схеми
CREATE SCHEMA sch;
CREATE SCHEMA auth;
// Створення користувацьких типів та доменів//
//Номер аудиторія
CREATE DOMAIN sch.d_classroom integer
CONSTRAINT valid_post
CHECK (VALUE IN (110, 111, 112, 113, 114, 115, 116, 117, 118, 119,
120, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 310,
311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 410, 411, 412,
413, 414, 415, 416, 417, 418, 419));
//Номер уроку
CREATE DOMAIN sch.d_lesson integer
CONSTRAINT valid_post
CHECK (VALUE IN (1, 2, 3, 4, 5, 6, 7, 8));
//День неділі
CREATE DOMAIN sch.d_day_week varchar(11)
CONSTRAINT valid_post
CHECK(VALUEIN('Понеділок','Вівторок','Середа','Четверг','Пятниця'));
//Стать
CREATE DOMAIN sch.d_pol varchar(7)
CONSTRAINT valid_post
CHECK (VALUE IN ('Чоловік', 'Жінка'));
--Цифра класу
CREATE DOMAIN sch.d_class_title varchar(2)
CONSTRAINT valid_post
```

```

CHECK (VALUE IN ('1','2', '3', '4', '5', '6', '7', '8', '9', '10',
'11', '12'));
//Буква класу
CREATE DOMAIN sch.d_class_kurs varchar(1)
CONSTRAINT valid_post
CHECK (VALUE IN ('A', 'Б', 'В', 'Г'));
//Адреса
CREATE TYPE sch.address AS (city character varying, street character
varying, house smallint, flat smallint);

// Створення таблиць //
// Учень
CREATE TABLE sch.pupil (
Id_pupil SERIAL NOT NULL PRIMARY KEY,
name varchar NOT NULL,
name_otch varchar NOT NULL,
last_name varchar NOT NULL,
birth_date DATE NOT NULL,
d_pol d_pol NOT NULL,
role INTEGER REFERENCES sch.role NOT NULL,
address address NOT NULL,
phone integer NOT NULL UNIQUE,
id_class integer REFERENCES sch.class NOT NULL,
id_family integer REFERENCES sch.family NOT NULL,
pass varchar NOT NULL
);
//Учитель
CREATE TABLE sch.teacher (
Id_teacher SERIAL NOT NULL PRIMARY KEY,
name varchar NOT NULL,
name_otch varchar NOT NULL,
last_name varchar NOT NULL,
birth_date DATE NOT NULL,
d_pol d_pol NOT NULL,
phone integer NOT NULL UNIQUE,

```

```

role INTEGER REFERENCES sch.role NOT NULL
address address NOT NULL
pass varchar NOT NULL
);
//Батьки
CREATE TABLE sch.family (
Id_family SERIAL NOT NULL PRIMARY KEY,
name varchar NOT NULL,
name_otch varchar NOT NULL,
last_name varchar NOT NULL,
role INTEGER REFERENCES sch.role NOT NULL
phone integer NOT NULL UNIQUE,
address address NOT NULL
pass varchar NOT NULL
);
//користувачів у системі та їх найменування
CREATE TABLE sch.role (
Id_role SERIAL NOT NULL PRIMARY KEY,
role_name VARCHAR NOT NULL UNIQUE
);
//Дисципліни
CREATE TABLE sch.discipline (
id_discipline SERIAL NOT NULL PRIMARY KEY,
title varchar NOT NULL UNIQUE
);
//Журнал
CREATE TABLE sch.ocenka (
Id_ocenka serial NOT NULL PRIMARY KEY,
date Date NOT NULL,
id_discteacher integer REFERENCES sch.discipline_teacher NOT NULL,
ocenk varchar NOT NULL
CHECK('1','2','3','4','5','6','7','8','9','10','11','12','н'),
id_pupil integer NOT NULL
);
//Розклад

```

```

CREATE TABLE sch.raspisanue (
  Id_raspisanue serial NOT NULL PRIMARY KEY,
  d_classroom integer NOT NULL,
  d_lesson integer NOT NULL,
  d_day_week varchar NOT NULL,
  id_discteacher integer REFERENCES sch.discipline_teacher NOT NULL
);

// Домашнє завдання
CREATE TABLE sch.homework (
  Id_homework SERIAL NOT NULL PRIMARY KEY,
  Hw NOT NULL,
  date Date NOT NULL,
  id_discteacher REFERENCES sch.discipline_teacher NOT NULL
);

//Клас
CREATE TABLE sch.class (
  Id_class serial NOT NULL PRIMARY KEY
  title varchar NOT NULL,
  kurs varchar NOT NULL,
  class_teacher REFERENCES sch.teacher NOT NULL ,
);

// предмет_вчитель_клас ('Навантаження')
CREATE TABLE sch.discipline_teacher (
  Id_disctiacher serial NOT NULL PRIMARY KEY,
  Id_teacher REFERENCES sch. teacher NOT NULL ,
  Id_class REFERENCES sch.class NOT NULL,
  Id_discipline REFERENCES sch.discipline NOT NULL,
  IS_Deleted
);

// Таблиця повідомлення
CREATE TABLE sch.message
(
  id_message serial NOT NULL PRIMARY KEY,
  id_family REFERENCES sch.family NOT NULL ,
  id_teacher REFERENCES sch.teacher NOT NULL ,

```

```

messag character varying(1000) NOT NULL,
date_time timestamp without time zone NOT NULL,
typell integer );

// Створення представлень //
// пошук пошти для відправки даних батькам про домашнє завдання //
CREATE OR REPLACE VIEW sch.homework_email
AS
SELECT ff.email,cl.title,cl.kursFROM sch.family ff
JOIN sch.pupil pu ON pu.id_family = ff.id_family
JOIN sch.class cl ON cl.id_class = pu.id_class;
// виклик даних про адміністратора //
CREATE OR REPLACE VIEW sch.idv_admin_phone
AS
SELECT
teacher.phone,teacher.name,teacher.name_otch,teacher.last_name,teach
er.data_roz,teacher.d_pol,teacher.address,teacher.pass,role.role_nam
e FROM sch.teacher
JOIN sch.role ON teacher.role = role.id_role;
// виклик даних учня //
CREATE OR REPLACE VIEW sch.idv_pupil_phone
ASSELECT
pupil.phone,pupil.name,pupil.name_otch,pupil.last_name,pupil.data_ro
z,pupil.d_pol,pupil.address,pupil.pass,role.role_name,class.title,cl
ass.kurs,family.name AS n_family,family.last_name AS l_family FROM
sch.pupil
JOIN sch.class ON pupil.id_class = class.id_class
JOIN sch.role ON pupil.role = role.id_role
JOIN sch.family ON pupil.id_family = family.id_family;
// виклик даних учителя //
CREATE OR REPLACE VIEW sch.idv_teacher_phone
AS
SELECT
teacher.phone,teacher.name,teacher.name_otch,teacher.last_name,
```

```

teacher.data_roz,teacher.d_pol,teacher.address,teacher.pass,role.rol
e_name,class.title,class.*::sch.class AS class,class.kursFROM
sch.teacher
JOIN sch.role ON teacher.role = role.id_role
JOIN sch.class ON teacher.id_teacher = class.teacher_class;
// виклик даних батьків //
CREATE OR REPLACE VIEW sch.idvt_family_phone
AS
SELECT family.phone,family.name AS f1,family.name_otch AS
f2,family.last_name AS f3,pupil.name AS n1,pupil.name_otch AS
n2,pupil.last_name AS n3,teacher.name AS t1,teacher.name_otch AS
t2,teacher.last_name AS t3,class.title,class.kursFROM sch.family
JOIN sch.pupil USING (id_family)
JOIN sch.class ON pupil.id_class = class.id_class
JOIN sch.teacher ON class.teacher_class = teacher.id_teacher;
// пошук пошти для відправки даних батькам про оцінки //
CREATE OR REPLACE VIEW sch.mark_email
AS
SELECT ff.email,pu.last_nameFROM sch.family ff
JOIN sch.pupil pu ON pu.id_family = ff.id_family;
// пошук пошти для відправки даних батькам про повідомлення //
CREATE OR REPLACE VIEW sch.message_email
AS
SELECT ff.email,ff.last_name,ff.name,ff.name_otch FROM sch.family
ff;
// розклад занять учителя //
CREATE OR REPLACE VIEW sch.show_nagr_phone
AS
SELECT
teacher.phone,discipline.title,raspisanue.d_day_week,raspisanue.d_le
sson,raspisanue.d_classroom,class.title AS numb,class.kurs AS let
FROM sch.raspisanue
JOIN sch.discipline_teacher USING (id_discteacher)
JOIN sch.discipline USING (id_discipline)
JOIN sch.class USING (id_class)

```

```

JOIN sch.teacher USING (id_teacher)
WHERE NOT discipline_teacher.is_deleted;
// розклад занять учня //
CREATE OR REPLACE VIEW sch.show_rasp_vt1_phone
AS
SELECT
pupil.phone, discipline.title, raspisanue.d_day_week, raspisanue.d_lesson,
raspisanue.d_classroom, tt.last_name, "substring"(tt.name::text, 1,
1) AS name1, "substring"(tt.name_otch::text, 1, 1) AS name_otch1 FROM
sch.raspisanue
JOIN sch.discipline_teacher USING (id_discteacher)
JOIN sch.discipline USING (id_discipline)
JOIN sch.class USING (id_class)
JOIN sch.teacher tt USING (id_teacher)
JOIN sch.pupil USING (id_class)
WHERE NOT discipline_teacher.is_deleted;
// розклад занять батьків //
CREATE OR REPLACE VIEW sch.show_rasp_family_phone
AS
SELECT
family.phone, discipline.title, raspisanue.d_lesson, raspisanue.d_class
room, raspisanue.d_day_week, tt.last_name, "substring"(tt.name::text,
1, 1) AS name1, "substring"(tt.name_otch::text, 1, 1) AS name_otch1
FROM sch.raspisanue
JOIN sch.discipline_teacher USING (id_discteacher)
JOIN sch.discipline USING (id_discipline)
JOIN sch.class ON class.id_class = discipline_teacher.id_class
JOIN sch.pupil ON pupil.id_class = class.id_class
JOIN sch.teacher tt USING (id_teacher)
JOIN sch.family USING (id_family)
WHERE NOT discipline_teacher.is_deleted;
// пошук телефону для авторизації
CREATE OR REPLACE VIEW auth.login_phone
AS
SELECT pupil.phone AS log, pupil.pass, role.role_name FROM sch.pupil

```

```

JOIN sch.role ON pupil.role = role.id_role
UNION
SELECT teacher.phone AS log,teacher.pass,role.role_name FROM
sch.teacher
JOIN sch.role ON teacher.role = role.id_role
UNION
SELECT family.phone AS log,family.pass,role.role_name FROM
sch.family
JOIN sch.role ON family.role = role.id_role;

```

// Тригер заборони вводу оцінок на одну дату двічі за один предмет.

```

CREATE FUNCTION sch.dz_3()
RETURNS trigger
LANGUAGE 'plpgsql'
AS $BODY$
begin
if(tg_op ='INSERT' or TG_OP ='UPDATE' and new.id_homework <>
old.id_homework) then
if(exists (select hw,date,id_discteacher from sch.homework where
date = new.date and id_discteacher = new.id_discteacher) =true)
then raise exception 'домашнее задание нельзя давать два раза';
return new;end if;end if;return new;end
$BODY$;
CREATE TRIGGER dz_t
BEFORE INSERT OR DELETE OR UPDATE
ON sch.homework
FOR EACH ROW EXECUTE PROCEDURE sch.dz_2();

```

// Тригер заборони вводу домашнього завдання на одну дату двічі за один предмет.

```

CREATE FUNCTION sch.ocenka_3()
RETURNS trigger
LANGUAGE 'plpgsql'
AS $BODY$
begin

```

```

if(tg_op = 'INSERT' or TG_OP = 'UPDATE' and new.id_ocenka <>
old.id_ocenka) then
if(exists (select ocenka,date,id_discteacher,id_pupil from sch.ocenka
where date = new.date and id_discteacher = new.id_discteacher and
id_pupil = new.id_pupil) =true)
then raise exception 'домашнее задание нельзя давать два
паза';return new;end if;end if;
return new;end
$BODY$;
CREATE TRIGGER ocenka_t
BEFORE INSERT OR UPDATE
ON sch.ocenka
FOR EACH ROW EXECUTE PROCEDURE sch.ocenka_3();
// Триггер заборони вводу вчителем предмету у різних класах одночасно
CREATE FUNCTION sch.raspisanue_3()
RETURNS trigger
LANGUAGE 'plpgsql'
AS $BODY$
begin
if(tg_op = 'INSERT' or TG_OP = 'UPDATE' and new.id_raspisanue <>
old.id_raspisanue) then
if(exists (select d_classroom,d_lesson,id_discteacher,d_day_week
from sch.raspisanue where d_classroom=new.d_classroom and
d_lesson=new.d_lesson and d_day_week = new.d_day_week) =true)
then raise exception 'учитель не может быть везде';return new;
end if;end if;return new;end
$BODY$;
CREATE TRIGGER rasp_t
BEFORE INSERT OR UPDATE
ON sch.raspisanue
FOR EACH ROW
EXECUTE PROCEDURE sch.raspisanue_3();

// Збережені процедури //
//Виставлення відміток

```

```

CREATE OR REPLACE FUNCTION sch.teach_ocenk_phone1(oc character
varying,ddate date,num character varying,let character varying,dis
character varying,teach integer,l_name character varying)
RETURNS void
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
insert into sch.ocenka (id_pupil,ocenk,id_discteacher,date)
select
sch.pupil.id_pupil,oc,sch.discipline_teacher.id_discteacher,ddate
from
sch.discipline_teacher,sch.discipline,sch.teacher,sch.class,sch.pupi
l where sch.discipline_teacher.id_class = sch.class.id_class and
sch.pupil.id_class = sch.class.id_class and
sch.discipline_teacher.id_discipline = sch.discipline.id_discipline
and sch.discipline_teacher.id_teacher = sch.teacher.id_teacher and
not sch.discipline_teacher.is_deleted and
EXISTS (select * from sch.discipline_teacher
where sch.discipline_teacher.id_class = sch.class.id_class and
sch.discipline_teacher.id_discipline = sch.discipline.id_discipline
and
sch.discipline_teacher.id_teacher = sch.teacher.id_teacher and
sch.discipline.title =dis and sch.teacher.phone=teach and
sch.class.title =num and sch.class.kurs =let and not
sch.discipline_teacher.is_deleted)
and sch.discipline.title =dis and sch.teacher.phone=teach and
sch.class.title =num and sch.class.kurs =let and sch.pupil.last_name
=l_name;END;
$BODY$;

```

//Задання домашнього завдання

```

CREATE OR REPLACE FUNCTION sch.teach_dz2_phone(dz character
varying,ddate date,num character varying,let character varying,dis
character varying,teach integer)

```

```

RETURNS void
LANGUAGE 'plpgsql'
AS $BODY$
BEGIN
insert into sch.homework (id_discteacher,hw,date)
select sch.discipline_teacher.id_discteacher,dz,ddate from
sch.discipline_teacher,sch.discipline,sch.teacher,sch.class
where sch.discipline_teacher.id_class = sch.class.id_class and
sch.discipline_teacher.id_discipline = sch.discipline.id_discipline
and
sch.discipline_teacher.id_teacher = sch.teacher.id_teacher and not
sch.discipline_teacher.is_deleted and
EXISTS (select * from sch.discipline_teacher
where sch.discipline_teacher.id_class = sch.class.id_class and
sch.discipline_teacher.id_discipline = sch.discipline.id_discipline
and
sch.discipline_teacher.id_teacher = sch.teacher.id_teacher and
sch.discipline.title =dis and sch.teacher.phone=teach and
sch.class.title =num and sch.class.kurs =let
and not sch.discipline_teacher.is_deleted) and sch.discipline.title
=dis and sch.teacher.phone=teach and sch.class.title =num and
sch.class.kurs =let;
END;$BODY$;

```

```

//Створення ролей//
//Адмін
CREATE ROLE a_role WITH PASSWORD '12345';
//Учень
CREATE ROLE p_role WITH PASSWORD '12' LOGIN;
//Учитель
CREATE ROLE t_role WITH PASSWORD '123' LOGIN;
//Батьки
CREATE ROLE f_role WITH PASSWORD '1234' LOGIN;
//Логін
CREATE ROLE login1 WITH PASSWORD '12' LOGIN;

```

```

//Надання привілей//
//Можливість підключатися до бази даних school
GRANT CONNECT ON DATABASE myschool TO a_role, p_role, t_role,
f_role,login1;
//Можливість працювати у схемі sch
GRANT USAGE ON SCHEMA sch TO a_role, t_role, p_role, f_role;
//Можливість працювати у схемі auth
GRANT USAGE ON SCHEMA auth TO login1;
//Надання права на представлення полі login1
GRANT SELECT ON auth.login_phone TO login1;
//Надання права на уявлення для входу та розкладу учню
GRANT SELECT ON sch.idv_pupil_phone,sch.show_rasp_vtl_phone,
sch.pupil,sch.ocenka,sch.homework TO p_role;
// права на Збережені процедури учня
GRANT execute ON function
sch.show_dz_pupil_ot_do(integer,date,date),
sch.show_dz_pupil_month(integer,integer),
sch.show_ocenk_pupil_ot_do(integer,date,date),
sch.show_avg_disc_pupil(integer),
sch.show_numb_pupil(integer),
sch.show_disc_pupil(integer),
sch.show_kurs_pupil(integer),
sch.rank_pupil(integer,character varying,character varying,
character varying) TO p_role;
//Надання права на уявлення батькам
GRANT SELECT ON sch.idvt_family_phone, sch.show_rasp_family_phone,
sch.family,sch.ocenka,sch.homework TO f_role;
// збережені процедури батькам
GRANT execute ON function
sch.show_ocenk_family_ot_do(integer,date,date),
sch.message_show_fam2(integer),
sch.show_dz_family_ot_do(integer,date,date),
sch.message_family(integer,character varying),
sch.show_avg_disc_family(integer),
sch.show_num_family(integer),

```

```

sch.show_disc_family(integer),
sch.show_kurs_family(integer),
sch.rank_family(integer,character varying,
character varying, character varying) TO f_role;
//Надання права перегляду та оновлення даних перерахованих таблиць
GRANT SELECT, INSERT,UPDATE ON sch.ocenka, sch.homework TO t_role;
//Надання права на уявлення та таблиці
GRANT SELECT ON sch.idv_teacher_phone, sch.choice,sch.teacher,
sch.raspisanue,
sch.homework_email,sch.mark_email,sch.message_email,sch.show_nagr_ph
one TO t_role;
// збережені процедури вчитель
GRANT execute ON function sch.message_family_name(integer),
sch.teach_ocenk_phonel(character varying,date,character
varying,character varying,character varying,integer,character
varying),
sch.teach_ocenk_up_phonel(character varying,date,character
varying,character varying,character varying,integer,character
varying),
sch.teach_dz2_phone(character varying,date,character
varying,character varying,character varying,integer),
sch.import_csv_ocenka(integer,text),
sch.import_csv_dz(integer,text),
sch.show_ocenk_class_ot_dol(character varying,character
varying,character varying,date,date),
sch.show_numb_teacher(integer),
sch.show_disc_teacher(integer),
sch.show_kurs_teacher(integer),
sch.show_dz_class_ot_do(character varying,character
varying,character varying,date,date),
sch.message_show_teach2(integer,character varying,character
varying,character varying),
sch.message_teacher(integer,character varying,character
varying,character varying,character varying),
sch.message_show_all_family1(integer),

```

```

sch.show_avg_disc_teach_s_dnl(integer,character varying,character
varying,character varying),
sch.rank_teacher(integer,character varying,character
varying,character varying), sch.teach_dz2_up_phone (character
varying,date,character varying,character varying, character
varying,integer)
TO t_role;
// Видалення з public
revoke execute on function [.....] from public
// Права на таблиці адміна
GRANT SELECT, INSERT,UPDATE ON sch.ocenka, sch.homework,sch.class,
sch.discipline,sch.family,sch.pupil,sch.raspisanue,sch.teacher TO
a_role;
GRANT delete ON sch.raspisanue
TO a_role;
// Права на представлення
GRANT SELECT ON sch.idv_admin_phone TO a_role;
// збережені процедури адміна
GRANT execute ON function
sch.delete_rasp_class(character varying,character varying),
sch.import_csv_reg_family(file_nam text),
sch.import_csv_reg_pupil(text),
sch.import_csv_reg_teacher(text),
sch.import_delete_pupil_f(text),
sch.import_next_class(text),
sch.import_rasp(text),
sch.register(character varying,character varying,character
varying,sch.address,date,character varying,integer,character
varying,character varying,character varying,character
varying,character varying,character varying),
sch.register_family(character varying,character varying, character
varying,integer,character varying,character varying),
sch.register_teacher(character varying,character varying,character
varying,sch.address,date,character varying,integer,character
varying),

```

```

sch.show_avg_disc_admin_s_dnl(character varying,character
varying,character varying),
sch.rank_admin(character varying,character varying,character
varying,
sch.import_delete_teacher(text)
TO a_role;
// Видалення з public
revoke execute on function [.....] from public

```

A.2 КОД ОСНОВНИХ КЛАСІВ

```

const express = require ("express")
const app = express ()
const db_pupil = require("./controller/pupilController.js")
const db_teacher = require("./controller/teacherController.js")
const db_family = require("./controller/familyController.js")
const db_admin = require("./controller/adminController.js")
const listen1 = require("./config/config_db");
app.use(express.static("public"))
app.set("views","./views")
app.set('view engine','ejs')
app.listen(listen1.listen1)
const session = require("express-session")
var cookieParser = require('cookie-parser')
app.use(session({ secret: '111', resave:false,
saveUninitialized:false, cookie: { maxAge: 600000 }}}));
app.use(cookieParser())
const bodyParser= require('body-parser')
const urlencodedParser = bodyParser.urlencoded({extended: false});
var multer = require('multer');
const upload = multer({ dest: '/tmp/' });
const path = require('path')
const nodemailer = require('nodemailer');
var transporter = nodemailer.createTransport({
service: 'gmail',

```

```

auth: {
  user: 'myschooltest12345@gmail.com',
  pass: 'Myschooltest12345-'
});

app.post('/inside_pupil1',urlencodedParser,db_pupil.inside_pupil)
app.get('/inside_pupil1',urlencodedParser,db_pupil.get_inside_pupil)
app.post('/schedule_pupil',urlencodedParser,db_pupil.schedule_pupil)
app.get('/schedule_pupil',urlencodedParser,db_pupil.get_schedule_pupil)
app.get('/homework_pupil',urlencodedParser,db_pupil.get_homework_pupil)
app.get('/mark_pupil',urlencodedParser,db_pupil.get_mark_pupil)
app.get('/avg_mark_pupil',urlencodedParser,db_pupil.get_mark_avg)
app.get('/rank_avg_pupil',urlencodedParser,db_pupil.get_rank_avg)
app.post('/rank_avg_pupil1',urlencodedParser,db_pupil.rank_avg1)
app.get('/rank_avg_pupil1',urlencodedParser,db_pupil.get_rank_avg1)
app.post('/inside_pupil_ocenk_bootstrap',urlencodedParser,db_pupil.inside_pupil_ocenk_bootstrap)
app.get('/inside_pupil_ocenk_bootstrap',urlencodedParser,db_pupil.get_inside_pupil_ocenk_bootstrap)
app.post('/inside_pupil_dz_bootstrap',urlencodedParser,db_pupil.inside_pupil_dz_bootstrap)
app.get('/inside_pupil_dz_bootstrap',urlencodedParser,db_pupil.get_inside_pupil_dz_bootstrap)
app.post('/inside_teacher1',urlencodedParser,db_teacher.inside_teacher)
app.get('/inside_teacher1',urlencodedParser,db_teacher.get_inside_teacher)
app.post('/schedule_teacher',urlencodedParser,db_teacher.schedule_teacher)
app.get('/schedule_teacher',urlencodedParser,db_teacher.get_schedule_teacher)
app.get('/mark_teacher',urlencodedParser,db_teacher.get_mark_teacher)
)

```

```

app.get('/avg_mark_teacher',urlencodedParser,db_teacher.get_mark_avg_nld)
app.post('/avg_mark_teacher1',urlencodedParser,db_teacher.mark_avg)
app.get('/avg_mark_teacher1',urlencodedParser,db_teacher.get_mark_avg)
app.get('/homework_teacher',urlencodedParser,db_teacher.get_homework_teacher)
app.get('/mark_show_teacher',urlencodedParser,db_teacher.get_mark_show_teacher)
app.get('/homework_show_teacher',urlencodedParser,db_teacher.get_homework_show_teacher)
app.get('/message_teacher1',urlencodedParser,db_teacher.get_message_teacher)
app.get('/rank_avg_teacher',urlencodedParser,db_teacher.get_rank_avg_nld)
app.post('/rank_avg_teacher1',urlencodedParser,db_teacher.rank_avg1)
app.get('/rank_avg_teacher1',urlencodedParser,db_teacher.get_rank_avg1)
app.post('/inside_teacher_journal_bootstrap',urlencodedParser,db_teacher.inside_teacher_journal_bootstrap)
app.get('/inside_teacher_journal_bootstrap',urlencodedParser,db_teacher.get_inside_teacher_journal_bootstrap)
app.post('/inside_teacher_journal_bootstrap1',urlencodedParser,db_teacher.inside_teacher_journal_bootstrap1)
app.get('/inside_teacher_journal_bootstrap1',urlencodedParser,db_teacher.get_inside_teacher_journal_bootstrap1)
app.post('/inside_teacher_journal_update_bootstrap1',urlencodedParser,db_teacher.inside_teacher_journal_update_bootstrap1)
app.post('/homework_teacher1',urlencodedParser,db_teacher.inside_teacher_dz_bootstrap)
app.get('/homework_teacher1',urlencodedParser,db_teacher.get_inside_teacher_dz_bootstrap)
app.post('/inside_teacher_import_ocenka',
upload.any(),db_teacher.inside_teacher_import_ocenka)

```

```

app.get('/inside_teacher_import_ocenka',urlencodedParser,db_teacher.
get_inside_teacher_import_ocenka)
app.post('/inside_teacher_import_dz',
upload.any(),db_teacher.inside_teacher_import_dz)
app.get('/inside_teacher_import_dz',urlencodedParser,db_teacher.get_
inside_teacher_import_dz)
app.post('/inside_teacher_show_ocenk_bootstrap',urlencodedParser,db_t
eacher.inside_teacher_show_ocenk_bootstrap)
app.get('/inside_teacher_show_ocenk_bootstrap',urlencodedParser,db_te
acher.get_inside_teacher_show_ocenk_bootstrap)
app.post('/inside_teacher_show_dz_bootstrap',urlencodedParser,db_tea
cher.inside_teacher_show_dz_bootstrap)
app.get('/inside_teacher_show_dz_bootstrap',urlencodedParser,db_tea
cher.get_inside_teacher_show_dz_bootstrap)
app.post('/inside_teacher_show_message',urlencodedParser,db_teacher.
inside_teacher_show_message)
app.get('/inside_teacher_show_message',urlencodedParser,db_teacher.g
et_inside_teacher_show_message)
app.post('/inside_teacher_show_message1',urlencodedParser,db_teacher
.inside_teacher_show_message1)
app.get('/inside_teacher_show_message1',urlencodedParser,db_teacher.
get_inside_teacher_show_message1)
app.post('/inside_teacher_all_show_message',urlencodedParser,db_tea
cher.inside_teacher_all_show_message)
app.get('/inside_teacher_all_show_message',urlencodedParser,db_tea
cher.get_inside_teacher_all_show_message)
app.post('/inside_family1',urlencodedParser,db_family.inside_family)
app.get('/inside_family1',urlencodedParser,db_family.get_inside_fami
ly)
app.post('/schedule_family',urlencodedParser,db_family.schedule_fami
ly)
app.get('/schedule_family',urlencodedParser,db_family.get_schedule_f
amily)
app.get('/rank_avg_family',urlencodedParser,db_family.get_rank_avg)
app.post('/rank_avg_family1',urlencodedParser,db_family.rank_avg1)

```

```

app.get('/rank_avg_family1',urlencodedParser,db_family.get_rank_avg1
)
app.get('/homework_family',urlencodedParser,db_family.get_homework_f
amily)
app.get('/mark_family',urlencodedParser,db_family.get_mark_family)
app.get('/message_family1',urlencodedParser,db_family.get_message_fa
mily)
app.get('/avg_mark_family',urlencodedParser,db_family.get_mark_avg)
app.post('/inside_family_ocenk_bootstrap',urlencodedParser,db_family.
inside_family_ocenk_bootstrap)
app.get('/inside_family_ocenk_bootstrap',urlencodedParser,db_family.g
et_inside_family_ocenk_bootstrap)
app.post('/inside_family_dz_bootstrap',urlencodedParser,db_family.ins
ide_family_dz_bootstrap)
app.get('/inside_family_dz_bootstrap',urlencodedParser,db_family.get_
inside_family_dz_bootstrap)
app.post('/inside_family_message_show',urlencodedParser,db_family.in
side_family_show_message)
app.get('/inside_family_message_show',urlencodedParser,db_family.get
_inside_family_show_message)
app.post('/inside_family_message_show1',urlencodedParser,db_family.i
nside_family_show_message1)
app.get('/inside_family_message_show1',urlencodedParser,db_family.ge
t_inside_family_show_message1)
app.post('/inside_admin1',urlencodedParser,db_admin.inside_admin)
app.get('/inside_admin1',urlencodedParser,db_admin.get_inside_admin)
app.get('/reg_admin',urlencodedParser,db_admin.get_reg_admin)
app.get('/schedule_admin',urlencodedParser,db_admin.get_rasp_admin)
app.get('/next_class_admin',urlencodedParser,db_admin.get_next_class
_admin)
app.get('/avg_mark_admin',urlencodedParser,db_admin.get_mark_avg_nld
)
app.post('/avg_mark_admin1',urlencodedParser,db_admin.mark_avg)
app.get('/avg_mark_admin1',urlencodedParser,db_admin.get_mark_avg)

```

```

app.get('/rank_avg_admin',urlencodedParser,db_admin.get_rank_avg_nld
)
app.post('/rank_avg_admin1',urlencodedParser,db_admin.rank_avg)
app.get('/rank_avg_admin1',urlencodedParser,db_admin.get_rank_avg)
app.post('/show_add_user',urlencodedParser,db_admin.show_add_user)
app.post('/show_add_user1',urlencodedParser,db_admin.show_add_user1)
app.post('/show_add_user2',urlencodedParser,db_admin.show_add_user2)
app.post('/show_add_user3',urlencodedParser,db_admin.show_add_user3)
app.post('/import_reg_teacher',
upload.any(),db_admin.import_reg_teacher)
app.get('/import_reg_teacher',urlencodedParser,db_admin.get_import_r
eg_teacher)
app.post('/import_reg_family',
upload.any(),db_admin.import_reg_family)
app.get('/import_reg_family',urlencodedParser,db_admin.get_import_re
g_family)
app.post('/import_reg_pupil',
upload.any(),db_admin.import_reg_pupil)
app.get('/import_reg_pupil',urlencodedParser,db_admin.get_import_reg
_pupil)
app.post('/import_rasp', upload.any(),db_admin.import_rasp)
app.get('/import_rasp',urlencodedParser,db_admin.get_import_rasp)
app.post('/import_next_class',
upload.any(),db_admin.import_next_class)
app.get('/import_next_class',urlencodedParser,db_admin.get_import_ne
xt_class)
app.get('/delete_schedule_admin',urlencodedParser,db_admin.get_del_r
asp_admin)
app.post('/delete_schedule_admin1',
urlencodedParser,db_admin.del_rasp_admin1)
app.get('/delete_schedule_admin1',urlencodedParser,db_admin.get_del_
rasp_admin1)

// файл config
const pg = require("pg")

```

```
const client_pupil1 = {
  user:'p_role',
  database:'myschool',
  password: '12',
  host:'localhost',
  port:5432,
  max:10,
  idleTimeoutMillis: 30000}
const client_teacher1 = {
  user:'t_role',
  database:'myschool',
  password: '123',
  host:'localhost',
  port:5432,
  max:10,
  idleTimeoutMillis: 30000}
const client_family1 = {
  user:'f_role',
  database:'myschool',
  password: '1234',
  host:'localhost',
  port:5432,
  max:10,
  idleTimeoutMillis: 30000}
const client_admin1 = {
  user:'a_role',
  database:'myschool',
  password: '12345',
  host:'localhost',
  port:5432,
  max:10,
  idleTimeoutMillis: 30000}
const listen1 = 4000
const client_login = {
  user:'login1',
```

```
database:'myschool',  
password: '12',  
host:'localhost',  
port:5432,  
max:10,  
idleTimeoutMillis: 30000}  
module.exports =({  
  client_pupill:client_pupill,  
  client_teacher1:client_teacher1,  
  client_family1:client_family1,  
  client_admin1:client_admin1,  
  listen1,  
  client_login:client_login})
```