

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря
Сікорського”**

Навчально-науковий інститут атомної та теплової енергетики

**Кафедра інженерії програмного забезпечення
в енергетиці**

КУРСОВА РОБОТА

з курсу «Бази даних»

Тема «Цифровий журнал для шкіл»

| | |
|------------------------|------------------------|
| Керівник Дацюк О.А. | Виконала Карпенко Я.Ю. |
| Допущено до захисту | Студентка 2 курсу |
| « ____ » _____ 20__ р. | Групи ТВ-11 |
| Захищено з оцінкою | Залікова книжка |
| _____ | № ТВ-1124 |

2022 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»

Кафедра інженерії програмного забезпечення в енергетиці

КУРСОВА РОБОТА

з дисципліни «Бази даних»

(назва дисципліни)

на тему: «Цифровий журнал для шкіл»

Студентки групи ТВ-11
спеціальність 121 «Інженерія програмного забезпечення»
освітня програма Інженерія
програмного забезпечення
інтелектуальних кібер-фізичних систем в енергетиці

Ліла В.О.

(прізвище та ініціали)

Керівник старший викладач Дацюк О.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

Київ - 2022 рік

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет НН ІАТЕ

(повна назва)

Кафедра інженерії програмного забезпечення в енергетиці

(повна назва)

Освітньо–кваліфікаційний рівень: бакалавр

Напрямок підготовки 121- “Інженерія програмного забезпечення”

(шифр і назва)

ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТКИ

Карпенко Ярослави Юріївни

(прізвище, ім'я, по-батькові)

1.Тема роботи: «Цифровий журнал для шкіл»

Керівник проекту(роботи): Дацюк Оксана Антонівна

(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

2.Срок подання студентом роботи: «20» грудня 2022 р.

3.Вихідні дані до проекту(роботи): мова програмування Python з використанням фреймворку Django, середовище розробки PyCharm, система керування базами даних – Mysql.

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) розробити інформаційну систему «Цифровий журнал для шкіл», спроектувати базу даних.

5.Дата видачі завдання «07» вересня 2019 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назви етапів виконання дипломного проекту(роботи) | Строк виконання етапів проекту(роботи) | Примітка |
|-------|---|--|----------|
| 1 | Затвердження теми роботи | 19.09.2022 | |
| 2 | Вивчення та аналіз задачі | 22.09.2022 | |
| 3 | Проектування структури БД | 23.09.2022 | |
| 4 | Розробка алгоритму реалізації задачі | 03.10.2022 | |
| 5 | Програмна реалізація задачі | 14.11.2022 | |
| 6 | Тестування програми | 29.11.2022 | |
| 7 | Оформлення пояснювальної записки | 19.12.2022 | |

Студентка

(підпис)

Керівник курсової роботи

(підпис)

Карпенко Я.Ю.

(прізвище та ініціали)

Дацюк О.А.

(прізвище та ініціали)

АНОТАЦІЯ

У курсовій роботі було створено систему, яка надає можливість бачити розклад уроків для вчителів та учнів. Функціональність також дає змогу учням бачити свої оцінки, а вчителям виставляти та редагувати поточний результат у разі потреби. Було реалізовано логіку розподілу користувачів відносно їхньої ролі.

У даній роботі описано створення бази даних та розроблення програмного продукту під неї. Також було обґрунтовано вибір зв'язків між таблицями, типів даних та обмежень на значення полів.

ANNOTATION

In coursework, a system was created to allow teachers and students to see the schedule of lessons. The functionality also allows students to see their grades and teachers to set and edit the current score if it is necessary. Logic was implemented to allocate users relative to their role.

This paper describes the creation of the database and software development for it. The choice of links between tables, data types and restrictions on the value of fields is also justified.

ЗМІСТ

| | |
|---|----|
| Вступ..... | 5 |
| 1 Опис інфомаційної системи | 6 |
| 2 Опис функціональної моделі інформаційної системи..... | 7 |
| 3 Опис бази даних | 10 |
| 3.1 Конептуальна модель бази даних..... | 10 |
| 3.2 Структура таблиць бази даних | 11 |
| 3.3 Приклади заповнення таблиць..... | 13 |
| 4 Опис програмної реалізації | 16 |
| 4.1 Використані засоби розробки | 16 |
| 4.2 Індивідуальний внесок в розробку системи..... | 18 |
| 5 Опис роботи системи | 21 |
| 5.1 Функціонал учнів | 21 |
| 5.2 Функціонал вчителів..... | 22 |
| 5.3 Функціонал адміністратора..... | 24 |
| Висновки | 26 |
| Список використаних джерел | 27 |
| Додаток..... | 28 |

ВСТУП

В наш час величезна кількість компаній використовують персональні комп'ютери для збереження та обробки будь-якого типу інформації. Ця інформація міститься у базах даних. Бази даних відіграють важливу роль у світі технологій, що активно розвивається. Все, з чим ми взаємодіємо кожен день у житті, мабуть, зберігається у якійсь базі даних. Робота з базами даних є необхідною навичкою для роботи з комп'ютером, а фахівці цієї галузі мають все більш попит.

Сьогодні організації неспроможні обійтись без баз даних. Це банки, навчальні заклади, заводи, магазини, державні установи тощо. Вони використовують їх для зацифрування та об'єднання даних, а також для швидкого доступу до них. Це економить час та витрати.

Метою даної курсової роботи постає розробка програмної системи “Електронний щоденник” в результаті чого ми отримаємо навички формування та створення бази даних та створення програмного забезпечення для роботи з ними.

Завданням є безпосередня розробка вебзастосунку зі зручним інтерфейсом. Можливістю виставлення та редагування оцінок вчителями, та переглядом своїх результатів учнями. Також вивід розкладу усім користувачам.

Об'єктом дослідження є сфера навчання та процес шкільної документації для подальшого покращення процесів обробки даних. Аналіз цих сфер допоміг розробити зручний інтерфейс для максимального спрощення збереження та перегляду оцінок.

Предметом дослідження є алгоритми розробки програмного забезпечення, використовуючи мову програмування Python та фреймворк Django

1. ОПИС ІНФОРМАЦІЙНОЇ СИСТЕМИ

Наша система становить собою вебзастосунок, що дозволяє відмовитись від паперових шкільних журналів. У користувача є можливість переглянути розклад та поточну успішність.

Перед роботою з застосунком треба авторизуватись, бо без цього доступу до сайту не буде.

У системі передбачаються такі ролі: учень, вчитель та адміністратор.

В учнів тільки режим перегляду. На сторінці журнал будуть автоматично виводитись всі його оцінки по всіх предметах. В журналі є можливість подивитись коли була поставлена оцінка. На сторінці розкладу виводиться тільки розклад класу, в якому навчається учень. В блоці уроку є інформація про назву предмета, час початку, ім'я вчителя та кабінет. Інтерфейс розроблений так аби було все інтуїтивно зрозуміло, не виникало жодних проблем і всі могли розібратись із застосунком.

У вчителів вже є можливість зміни даних. На сторінці журналу наявна опція вибору класу для виставлення/перегляду оцінок. Оцінки можна виставляти та редагувати в разі помилки або перездачі роботи. В розкладі виводиться розклад вчителя, де вказано назву предмета, час початку, клас та кабінет.

Адміністратор має змогу додавати користувачів та інформацію про них в систему, адже абихто не може отримати доступ до щоденника. А також додавати або редагувати розклад.

В нашому вебзастосунку майже вся інформація підтягується автоматично, базуючись та обліковому записі користувача, тому від нього вимагається мінімум дій.

2. ОПИС ФУНКЦІОНАЛЬНОЇ МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Оскільки робота була командною, треба було використати інструменти для організації робочого процесу та чіткого розуміння завдання. Було обрано дві діаграми: Data Flow Diagram та Use Case Diagram.

Діаграма потоків даних (DFD) — це тип діаграми-схеми, яка показує рух інформації з одного місця в інше в рамках певного процесора в цілому[1]. В інших випадках — DFD може показати, як співпрацюють різні відділи організації — це робить речі зрозумілими та послідовними.

Діаграми потоків даних побудовані на основі спрощеної системи позначень, яка включає набір прямокутників і кіл у поєднанні зі стрілками та лаконічними аббревіатурами, що позначають вхід, вихід, пункти зберігання та маршрути між кожним пунктом призначення.

Все це влучно зображує «Потік».

Існує чотири фундаментальні елементи нотації DFD — так звані елементи даних:

- Процес — представляє будь-який перетворювальний процес вхідного потоку інформації у вихідний робочий потік. Процес отримує вхідні дані та генерує вихідні дані;
- Потік даних — являє собою рух інформації всередині системи між зовнішніми сутностями, сховищами даних і процесами. Відображає характер даних, що використовуються в системі;
- Сховище даних — являє собою сховища для даних, які не рухаються в цей момент. Це може бути або буфер про всяк випадок, або черга для подальшого використання.
- Зовнішня сутність — являє собою джерела або пункти призначення інформації поза межами описуваної системи. Сутності або надають дані системі, або отримують їх від процесів.

На рисунку 2.1 зображена діаграма потоків даних для системи «Електронний щоденник».

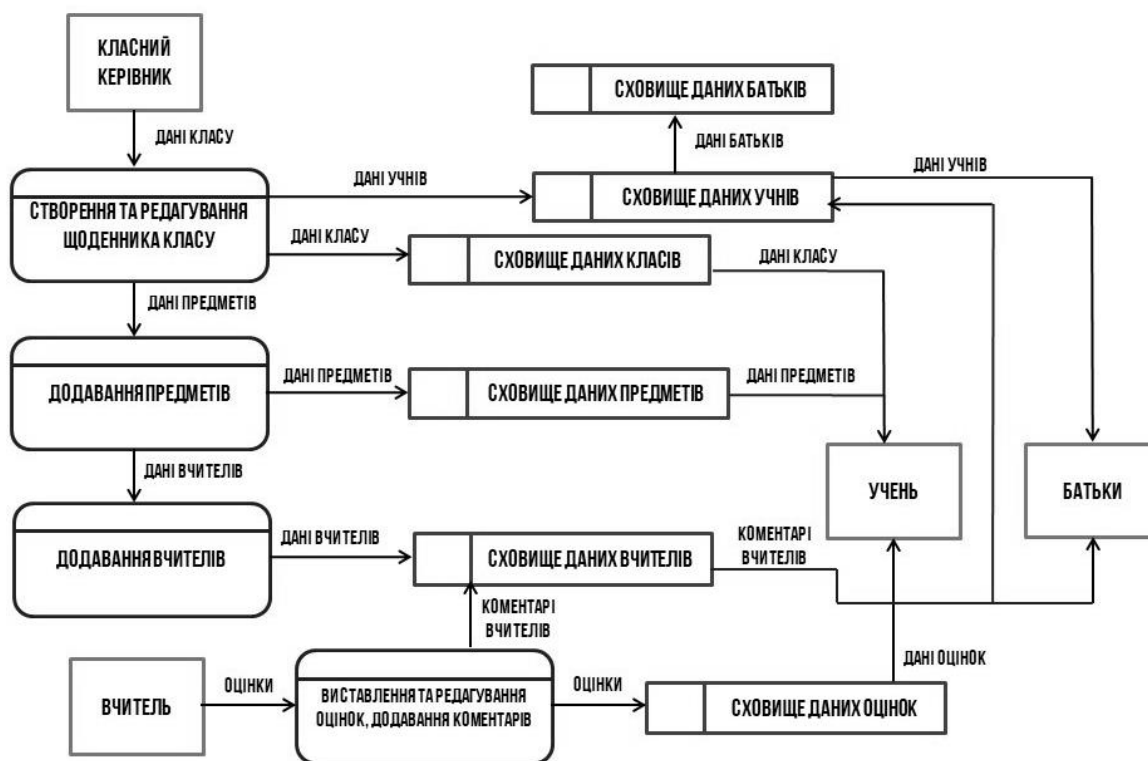


Рисунок 2.1 – DFD для системи «Електронний щоденник»

Також був використаний один із видів UML діаграм — Use Case Diagram.

Unified Modeling Language (UML) — уніфікована мова моделювання[2]. Розшифруємо: modeling передбачає створення моделі, що описує об'єкт. Unified (універсальний, єдиний) — підходить для широкого класу проєктованих програмних систем, різних сфер застосувань, типів організацій, рівнів компетентності, розмірів проєктів.

Use-case diagram або діаграма прецедентів використовує 2 основні елементи:

- Actor (учасник) — безліч логічно пов'язаних ролей, які виконуються під час взаємодії з прецедентами або сутностями (система, підсистема або клас). Учасником може бути людина, роль людини в системі або інша система, підсистема чи клас, які представляють щось поза сутністю.

- Use case (прецедент) — опис окремого аспекту поведінки системи з точки зору користувача. Прецедент не показує, «як» досягається певний результат, а тільки «що» саме виконується.

На рисунку 2.2 зображена діаграма прецедентів для системи «Електронний щоденник».

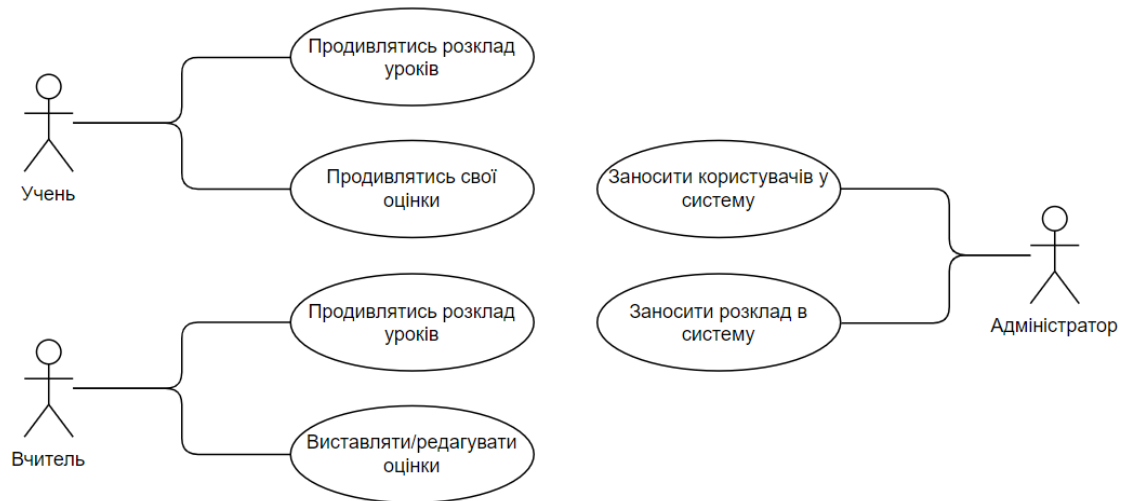


Рисунок 2.2 – Use-case diagram для системи «Електронний щоденник»

3. ОПИС БАЗИ ДАНИХ

З метою розробки бази даних для нашої інформаційної системи було розроблено покроковий план, який передбачає декілька етапів проєктування бази даних за допомогою моделей та діаграм. Вони допомагають структурувати та дані та визначити сутність системи.

3.1. Концептуальна модель бази даних

Перший пункт розробки це побудова концептуальної моделі бази даних.

Концептуальна модель даних — схема з мінімальною кількістю подробиць[3]. Перевага цього підходу полягає в можливості відобразити загальну структуру моделі та всю архітектуру системи. Менш масштабні системи можуть обійтися і без цієї моделі. Етап концептуального моделювання полягає в побудові опису предметної області в термінах формальної мови, наприклад, в термінах моделі сутностей і зв'язків. Концептуальна модель є першим рівнем деталізації ER-діаграми.

Схема «сутність-зв'язок» (також ERD або ER-діаграма) — це різновид блок-схеми, де показано, як різні «сутності» (люди, об'єкти, концепції тощо) пов'язані між собою всередині системи. ER-діаграми найчастіше застосовують для проєктування та налагодження реляційних баз даних у сфері освіти, дослідження та розроблення програмного забезпечення та інформаційних систем для бізнесу.

ER-діаграми часто використовують у поєднанні з діаграмами DFD, які схематично показують рух потоків інформації в межах процесу або системи.

На рисунку 3.1 зображена концептуальна модель бази даних системи «Електронний щоденник».

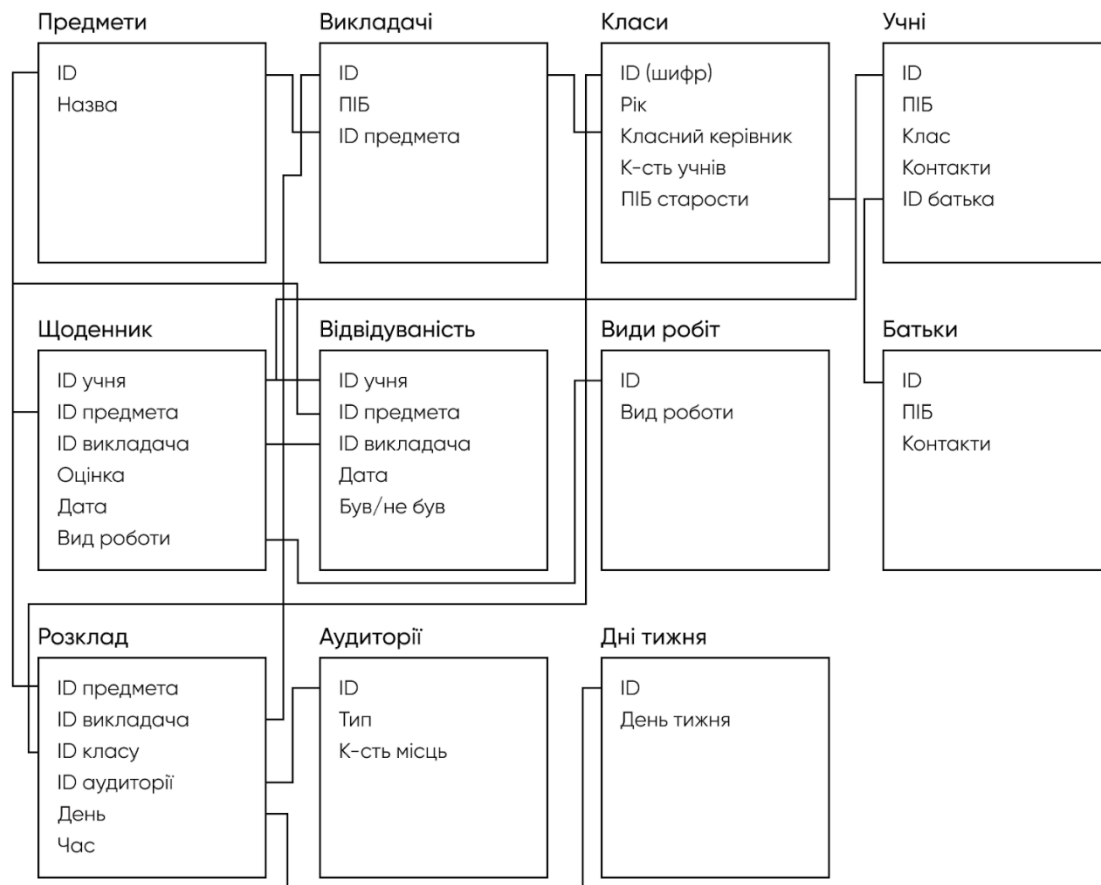


Рисунок 3.1 – Концептуальна модель бази даних для системи
«Електронний щоденник»

Створена модель буде корисна в подальшій розробці бази даних та самого продукту. На ній компактно та зрозуміло зображені атрибути кожної сутності та зв'язки між ними.

3.2 Структура таблиць бази даних

В нашій базі даних є такі сутності: предмет, викладач, учень, клас, вид роботи, батьки, дні тижня, аудиторія, розклад та щоденник. Після створення таблиць були реалізовані зв'язки між ними. На рисунку 3.2 зображена фізична модель бази даних нашої системи.

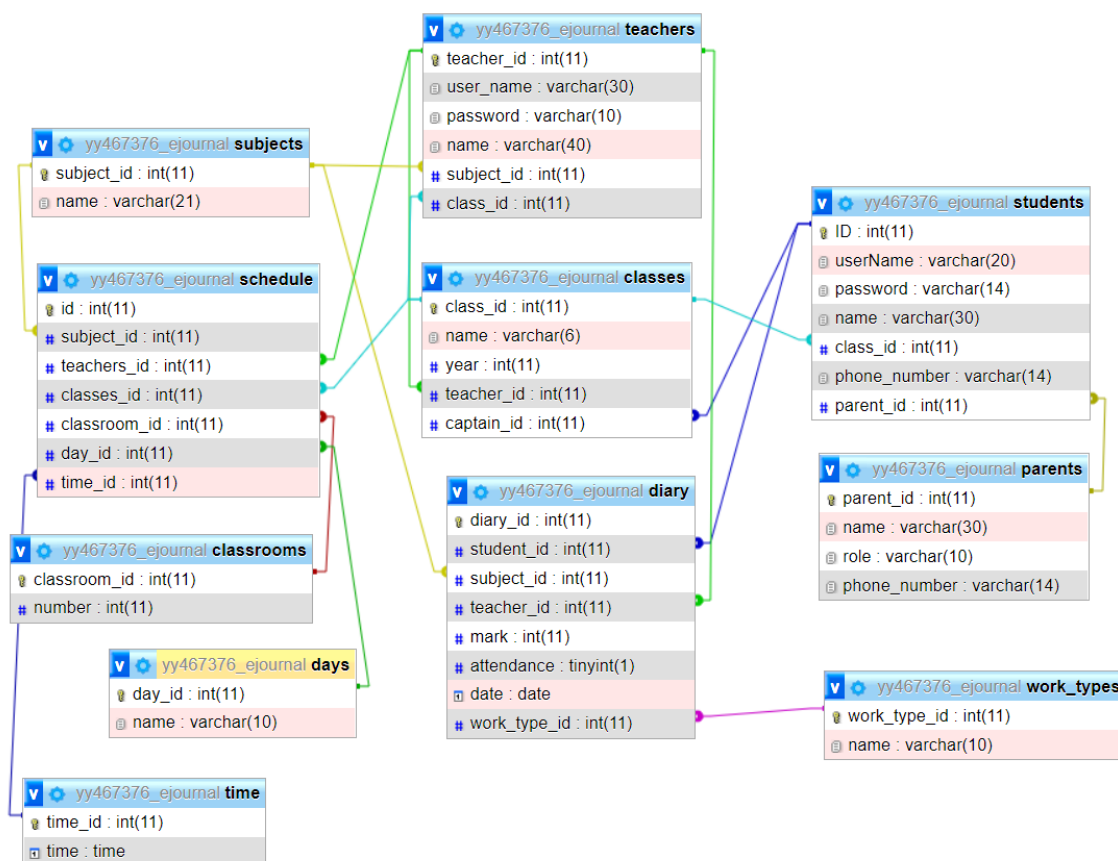


Рисунок 3.2 – Фізична модель бази даних для системи «Електронний щоденник»

Основними зв'язками між даними сутностями є учень та батьки; учень та клас; клас та класний керівник; вчитель та предмет; розклад та предмет, клас, вчитель; щоденник та учень, вчитель, предмет.

Створення бази даних відбувалось на основі середовища phpMyAdmin. Вона налічує в собі 11 таблиць. Розглянемо основні з них.

Для зберігання даних про всіх учнів було реалізовано таблицю students, яка містить такі поля: id – первинний ключ, user_name – ім'я користувача в системі, password – пароль для входу в систему, name – повне ім'я, class_id – клас, в якому навчається, phone_number – номер телефону, parent_id – айдішник одного з батьків.

Схожа таблицка для зберігання інформації про вчителів – teacher з полями: id – первинний ключ, user_name – ім'я користувача в системі, password – пароль для входу в систему, name – повне ім'я, subject_id – предмет, який викладає, class_id – клас, у якого цей вчитель є класним керівником.

Також була створена таблиця classes(класи): id – первинний ключ, name – назва класу (шифр), year – рік в якому почали навчатись, teacher_id – класний керівник, captain_id – староста класу.

Таблиця subjects має id та name, де зберігаються назви всіх предметів.

Однією з двох головних таблиць є schedule(розклад) з такими полями: id – первинний ключ, subject_id – предмет, class_id – клас, teacher_id – вчитель, classroom_id – класна кімната, day_id – день, time_id – час.

Другою є таблиця для самого щоденника з оцінками – diary з відповідними полями: id – первинний ключ, subject_id – предмет, student_id – учень, teacher_id – вчитель, mark – оцінка, attendance – присутність, date – дата, work_type – вид роботи.

Також були створені допоміжні таблиці:

- Parents. Таблиця з інформацією про батьків, їхню роль з сім'ї (тобто батько чи матір) та контактний номер телефону.
- Classrooms. Таблиця з всіма кабінетами певної школи.
- Days. Таблиця з робочими днями тижня
- Time. Таблиця з розкладом дзвінків уроків.

Як індивідуальний внесок було розроблено такі таблиці як schedule, classrooms, days та time. Також було встановлено зв'язки між цими таблицями та заповнено їх даними.

3.3 Приклади заповнення таблиць

Для подальшого проєктування застосунку та роботи з базою даних її потрібно було заповнити даним. Заповнення таблиці відбувається за допомогою команди INSERT, в деяких випадках UPDATE, якщо заповнення

значення містить в собі помилку або вже не актуальне на час використання.

На рисунку 3.4 зображена команда заповнення таблиці students.

```
INSERT INTO `students`(`ID`, `userName`, `password`, `name`,  
                        `class_id`, `phone_number`, `parent_id`)  
VALUES (3,'bulavkomv','1003','Булавко Максим Вікторович',  
        1,'0984567901',6);
```

Рисунок 3.4 – Приклад команди заповнення таблиці students

Після повторного занесення всіх даних, з урахуванням всіх обмежень цілісності даних, маємо сформовану таблицю учнів.

Вона містить в собі первинний ключ, ім'я, логін та пароль, номер телефону, айдішник класу, в якому вчиться учень та айдішник одного із батьків. Кінцевий результат прикладу заповнення цієї таблиці можемо побачити на рисунку 3.5.

| ID | userName | password | name | class_id | phone_number | parent_id |
|----|------------------|----------|--------------------------------|----------|--------------|-----------|
| 3 | bulavkomv | 1003 | Булавко Максим Вікторович | 1 | 0984567901 | 6 |
| 4 | vordakmm | 1004 | Вордак Мстислав Максимович | 1 | 0984432321 | 8 |
| 5 | gruschenkoot | 1005 | Грищенко Олександр Тарасович | 1 | 0986789021 | 10 |
| 6 | danulenkood | 1006 | Даниленко Олександр Денисович | 1 | 0989067321 | 12 |
| 7 | derkachpg | 1007 | Деркач Павло Глібович | 1 | 0984599921 | 14 |
| 8 | kalibabao | 1008 | Калібаба Олег Олександрович | 1 | 0564567321 | 16 |
| 9 | kovalchukim | 1009 | Ковальчук Іван Маркови | 1 | 0984907321 | 18 |
| 10 | kohanovskiydr | 1010 | Кохановський Данііл Русланович | 1 | 0984565321 | 20 |
| 11 | lipady@gmail.com | 2001 | Липа Денис Янович | 2 | 0984567000 | 22 |
| 12 | muronenkoav | 2002 | Мироненко Антон Владиславович | 2 | 0984567600 | 24 |
| 13 | potiykoigr | 2003 | Потійко Євгенія Романівна | 2 | 0964567901 | 26 |
| 14 | pohukuysm | 2004 | Похилий Стас Макарович | 2 | 0984442321 | 28 |
| 15 | ryabchenkoav | 2005 | Рябенко Анастасія Вікторівна | 2 | 0986785021 | 30 |
| 16 | sunyavskaav | 2006 | Синявська Анастасія Вадимівна | 2 | 0989467321 | 32 |
| 17 | syrmaima | 2007 | Сурмай Марія Адріївна | 2 | 0994599921 | 34 |
| 18 | tumoschenkoip | 2008 | Тимошенко Ілля Пилипович | 2 | 0564563321 | 36 |
| 19 | yzkaay | 2009 | Узька Анна Юріївна | 2 | 0984937321 | 38 |
| 20 | fedorchukif | 2010 | Федорчук Ігор Федорович | 2 | 0987565321 | 40 |

Рисунок 3.5 – Приклад заповненої таблиці students.

Також ми маємо таблиці, де значення більшості полів це посилання на дані з інших таблиць. Розглянемо таке заповнення на прикладі. Воно потребує неабиякої уважності, бо існує велика ймовірність внести помилкові дані. Маємо таблицю розкладу з такими полями: первинний ключ, вчитель, клас, предмет, класна кімната, день та час. На рисунку 3.6 зображена команда заповнення таблиці schedule.

```
INSERT INTO `schedule`(`id`, `subject_id`, `teachers_id`, `classes_id`,
                        `classroom_id`, `day_id`, `time`)
VALUES (6,5,4,1,21,1,6)
```

Рисунок 3.6 – Приклад команди заповнення таблиці schedule

Після занесення всіх даних, маємо кінцевий результат. Приклад заповнення цієї таблиці можемо побачити на рисунку 3.7.

| id | subject_id | teachers_id | classes_id | classroom_id | day_id | time_id |
|----|------------|-------------|------------|--------------|--------|---------|
| 47 | 3 | 13 | 5 | 25 | 1 | 2 |
| 45 | 1 | 13 | 5 | 8 | 1 | 0 |
| 46 | 2 | 13 | 5 | 20 | 2 | 1 |
| 44 | 7 | 13 | 5 | 1 | 2 | 3 |
| 50 | 1 | 13 | 5 | 7 | 3 | 3 |
| 48 | 1 | 13 | 5 | 22 | 4 | 2 |
| 49 | 4 | 13 | 5 | 30 | 5 | 0 |

Рисунок 3.7 – Приклад заповненої таблиці schedule

Оскільки заповнення розкладу безпосередньо командами тягне за собою велику кількість помилок, то для задач такого типу зазвичай розробляють інтерфейс.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Використані засоби розробки

Для подальшої розробки програмного забезпечення треба було обрати мову написання, фреймворк та середовище для спільного написання коду.

Щодо мови програмування, то вибір пав на Python. Python - це високорівнева інтерпретована мова динамічного програмування загального призначення, в якій основна увага приділяється читабельності коду[4]. Вона зазвичай має невеликі програми порівняно з Java і C. Python входить до числа найбільш популярних і швидкозростаючих мов у світі. Python - потужна, гнучка і проста у використанні мова. Крім того, спільнота Python дуже активна. Вона використовується в багатьох організаціях, оскільки підтримує безліч парадигм програмування. Вона також здійснює автоматичне керування пам'яттю.

Для проєктування самого вебзастосунку було обрано фреймворк Django. Django - це високорівневий вебфреймворк на мові Python, який сприяє швидкій розробці та чистому, прагматичному дизайну[5]. Створений досвідченими розробниками, він бере на себе більшу частину клопоту, пов'язаного з веброботкою, тож розробник може зосередитися на написанні свого додатка без необхідності винаходити колесо.

Переваги Django:

- Неймовірно швидкий. Django був розроблений для того, щоб допомогти розробникам якнайшвидше довести додаток від концепції до завершення.
- Надійна безпека. Django серйозно ставиться до безпеки та допомагає розробникам уникнути багатьох поширених помило.
- Чудова масштабованість. Деякі з найжвавіших сайтів в Інтернеті використовують здатність Django швидко і гнучко масштабуватися.

Для сумісного написання коду було обрано вебзастосунок github. GitHub – це вебінтерфейс, який використовує Git, програмне забезпечення з відкритим вихідним кодом для контролю версій, що дозволяє декільком людям одночасно вносити окремі зміни до вебсторінки. Оскільки GitHub дозволяє співпрацювати в режимі реального часу, він заохочує команди до спільної роботи над створенням та редагуванням контенту своїх сайтів. На рисунку 4.1 зображено схематичний принцип роботи GitHub та Git[6].

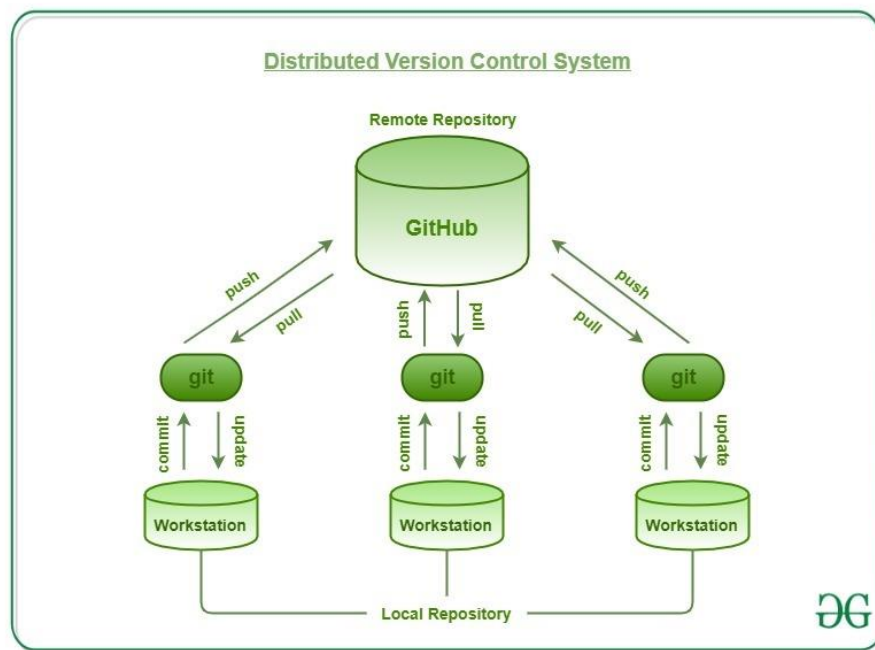


Рисунок 4.1 – Розподілена система контролю версій

Розподілені системи – це системи, які дозволяють користувачам виконувати роботу над проєкт з усього світу. Розподілена система містить центральний репозиторій, до якого можуть мати доступ багато віддалених співробітників за допомогою системи контролю версій. Git дзеркально відображає весь репозиторій на кожному знімку версії, яку витягує користувач. В такому випадку, якщо центральний сервер вийде з ладу, то копії репозиторіїв можна буде отримати назад у користувачів, які завантажили останній знімок проєкт.

Для написання клієнтської частини вебзастосунку, було використано HTML, CSS та JS.

4.2 Індивідуальний внесок у розробку системи

Під час роботи було реалізовано різні функціональності для виводу відповідного розкладу.

Після того, як користувач заходить у свій особистий кабінет, відбувається його ідентифікація. Далі підтягується вся необхідна його особиста інформація. Якщо це вчитель, то для вибору розкладу нам потрібне повне ім'я, якщо учень, то необхідно дізнатись клас, в якому він навчається. На рисунку 4.2 можна побачити програмний код, де відбувається ідентифікація учня та отримання інформації, потрібної для формування відповідного розкладу.

```
current_user = request.user
if current_user.role == "STUDENT":
    ''' Displaying schedule '''
    group = current_user.classes
    schedules = Schedule.objects.filter(classes=group)
    groups = Classes.objects.filter(class_id=group)
    for i in groups:
        name = i.name
        days = Days.objects.all()
        times = Time.objects.all()
```

Рисунок 4.2 – формування необхідної для розкладу інформації

Після того, як вся інформація була отримана, переходимо до безпосереднього формування розкладу. Для цього було розроблено алгоритм, для заповнення списку з уроками в розкладі. Цикл проходиться по днях, годинах та самим урокам. Таким чином формується список спочатку залежно від послідовності уроку, потім від дня тижня. Розглянути сам алгоритм можна на рисунку 4.3.

```

lessons = []

for time in times:
    for day in days:
        counter = 0
        for lesson in schedules:
            if lesson.day == day and lesson.time == time and not counter:
                lessons.append(lesson)
                counter += 1
        if not counter:
            lessons.append(None)

```

Рисунок 4.3 – алгоритм заповнення розкладу

Далі всі необхідні дані передаються в template та також за допомогою циклу виводяться на екран. На рисунку 4.4 зображено вивід розкладу в template.

```

<div class="row d-flex justify-content-center" style="...">
    {% for day in days %}
        <div class="col-lg-2 col-md-6 day text-center"><div class="day-text">{{ day.name }}</div></div>
    {% endfor %}
</div>
<div class="row d-flex justify-content-center">
    {% for item in lessons %}
        {% if item != None %}
            <div class="col-lg-2 col-md-6 lesson text-center">
                <div class="time">{{ item.time }}</div>
                <div class="title">{{ item.subject }}</div>
                <div class="teacher">Викладач: {{ item.teachers }}</div>
                <div class="room">Кабінет: {{ item.classroom }}</div>
            </div>
        {% elif item == None %}
            <div class="col-lg-2 col-md-6 lesson-empty">
                </div>
        {% endif %}
    {% endfor %}
</div>

```

Рисунок 4.4 – вивід розкладу в template

Додавши стилі, маємо приємний для ока інтерфейс з розкладом уроків для учня. Побачити його можна на рисунку 4.5.

| | | | | |
|--|--|--------|--|---|
| Журнал | Розклад | | | |
| Клас: 7-В | | | | |
| Понеділок | Вівторок | Середа | Четвер | П'ятниця |
| 08:00:00 Англійська мова Викладач: Олена Хаймарс Петрівна Кабінет: 107 | | | | 08:00:00 Зарубіжна література Викладач: Олена Хаймарс Петрівна Кабінет: 309 |
| | 08:55:00 Українська мова Викладач: Олена Хаймарс Петрівна Кабінет: 209 | | | |
| 09:50:00 Українська література Викладач: Олена Хаймарс Петрівна Кабінет: 304 | | | 09:50:00 Англійська мова Викладач: Олена Хаймарс Петрівна Кабінет: 301 | |
| | 10:55:00 Алгебра Викладач: Олена Хаймарс Петрівна Кабінет: 100 | | | |

Рисунок 4.5 – розклад учня

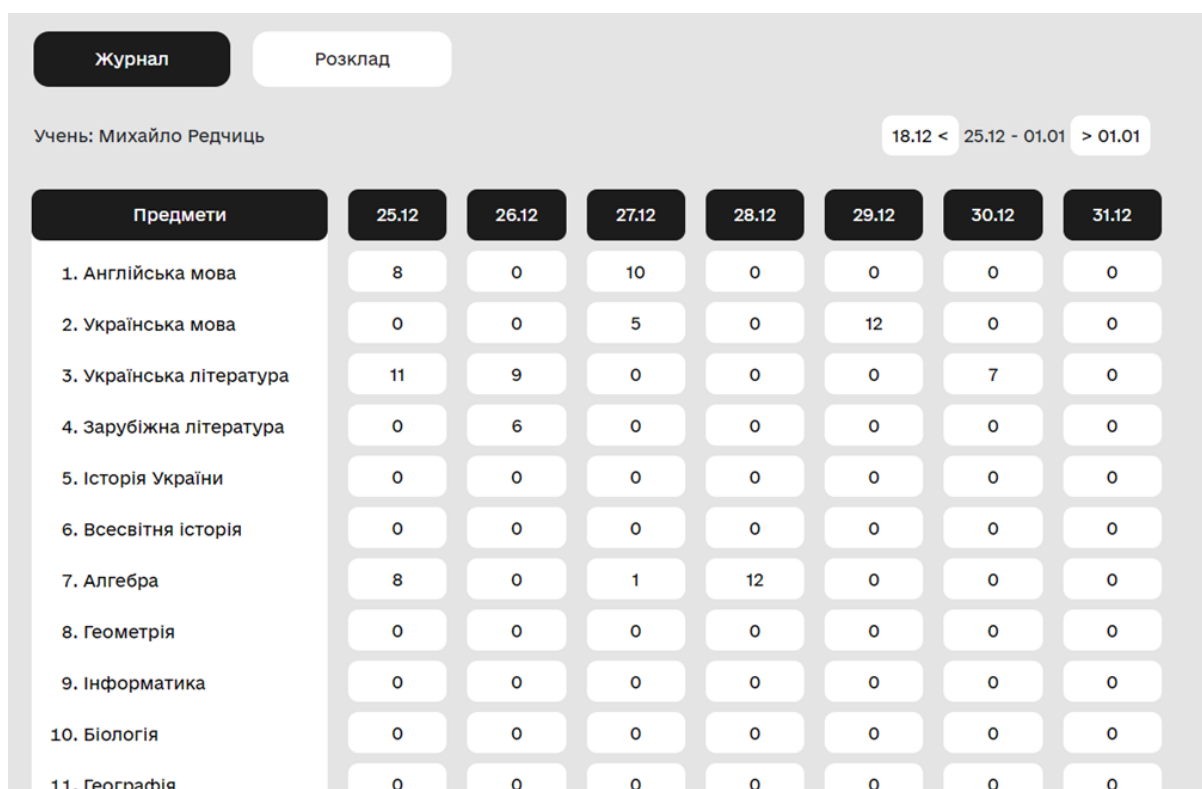
Для вчителя алгоритм формування розкладу аналогічний, тому розглядати його не будемо.

5. ОПИС РОБОТИ СИСТЕМИ

В результаті роботи було одержано вебзастосунок зі зручним та корисним функціоналом, який покращить та полегшить процес ведення документації для шкіл. Було реалізовано декілька ролей з різними можливостями.

5.1 Функціонал учнів

Учні не наділені великою кількістю можливостей, в їхньому розпорядженні тільки перегляд своїх оцінок та розкладу. Після авторизації, вони автоматично попадають на сторінку свого щоденника з оцінками. На рисунку 5.1 можна побачити інтерфейс оцінок учнів.



| Журнал | | Розклад | | | | | |
|--------------------------|-------|-------------------------------|-------|-------|-------|-------|-------|
| Учень: Михайло Редчиць | | 18.12 < 25.12 - 01.01 > 01.01 | | | | | |
| Предмети | 25.12 | 26.12 | 27.12 | 28.12 | 29.12 | 30.12 | 31.12 |
| 1. Англійська мова | 8 | 0 | 10 | 0 | 0 | 0 | 0 |
| 2. Українська мова | 0 | 0 | 5 | 0 | 12 | 0 | 0 |
| 3. Українська література | 11 | 9 | 0 | 0 | 0 | 7 | 0 |
| 4. Зарубіжна література | 0 | 6 | 0 | 0 | 0 | 0 | 0 |
| 5. Історія України | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6. Всесвітня історія | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7. Алгебра | 8 | 0 | 1 | 12 | 0 | 0 | 0 |
| 8. Геометрія | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9. Інформатика | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10. Біологія | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11. Географія | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рисунок 5.1 – щоденник оцінок учня

На сторінці виводиться список всіх предметів, які викладаються в певному класі. Дати відображаються, починаючи від поточної, тобто дня

перегляду оцінок. Зверху маємо перемикач дат на 7 днів. Зверху є кнопки перемикання між розкладом та щоденником. Перейшовши до розкладу маємо екран, який можна побачити на рисунку 5.2.

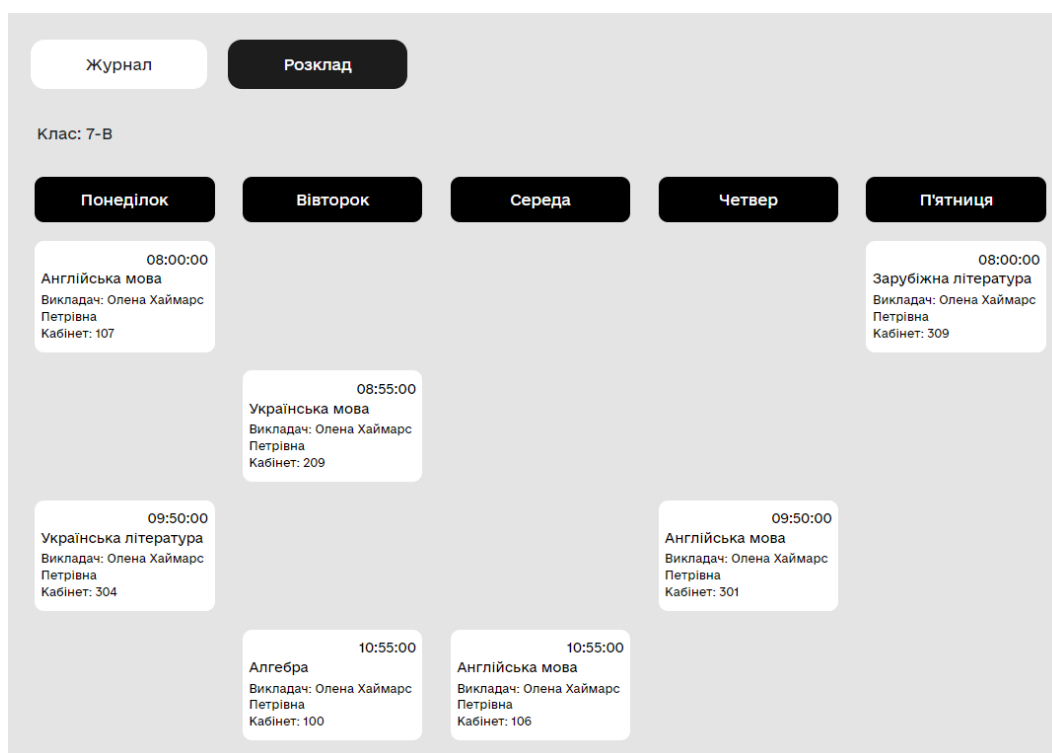


Рисунок 5.2 – розклад учня

В блоці уроку відображається назва предмета, час початку, викладач та кабінет проведення уроку.

5.2 Функціонал вчителя

У вчителів більше можливостей, адже вони мають виставляти оцінки. Авторизувавшись, попадаємо на сторінку виставлення оцінок. Її вигляд можна побачити на рисунку 5.3.

Журнал

Розклад

Клас

Предмет

Дата

7-В

Українська література

25.12.2022

Підтвердити

Вчитель: Олена Хаймарс Петрівна

| Учні | 25.12 | 26.12 | 27.12 | 28.12 | 29.12 | 30.12 | 31.12 |
|----------------------|-------|-------|-------|-------|-------|-------|-------|
| 1. Михайло Редчиць | 11 | 9 | 0 | 0 | 0 | 7 | 0 |
| 2. Пчелінцев Роман | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3. Битько Антон | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4. Кохась Дмитро | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5. Євгеній Проценко | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6. Перепелиця Максим | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Виставити оцінку

Студент

Оцінка

Дата

Михайло Редчиць

1

дд.мм.рррр

Виставити оцінку

Рисунок 5.3 – сторінка виставлення оцінок у вчителя

Інтерфейс дещо складніший. Маємо можливість обирати клас, предмет та дату для перегляду поточного результату. Під списком учнів маємо форму для виставлення або виправлення оцінки. Обираємо учня, оцінку та дату та натискаємо “Виставити оцінку”. Після оновлення сторінки, оцінка з’являється у відповідній комірці. Перемкнувшись на розклад, маємо сторінку, яку можна побачити на рисунку 5.4.

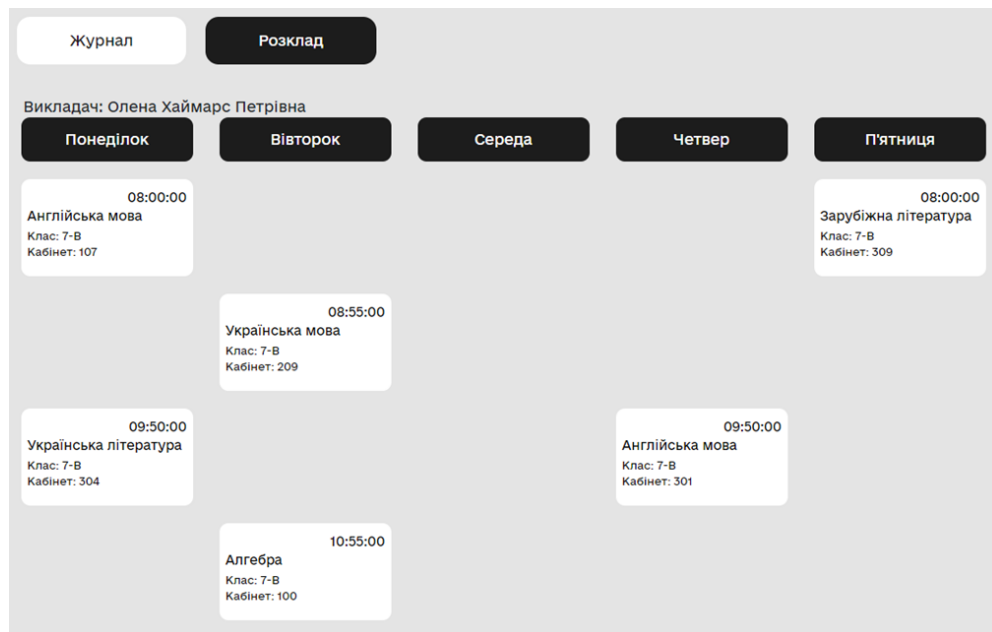


Рисунок 5.4 – розклад вчителя

Розклад маємо відповідний до розкладу учнів, але замість імені викладача в блоці відображається шифр класу

5.3 Функціонал адміністратора

Після входу в акаунт адміністратора ми маємо можливість створювати акаунти вчителів та учнів, формувати класи, а також розміщувати розклад уроків. Маємо початковий інтерфейс показаний на рисунку 5.5.

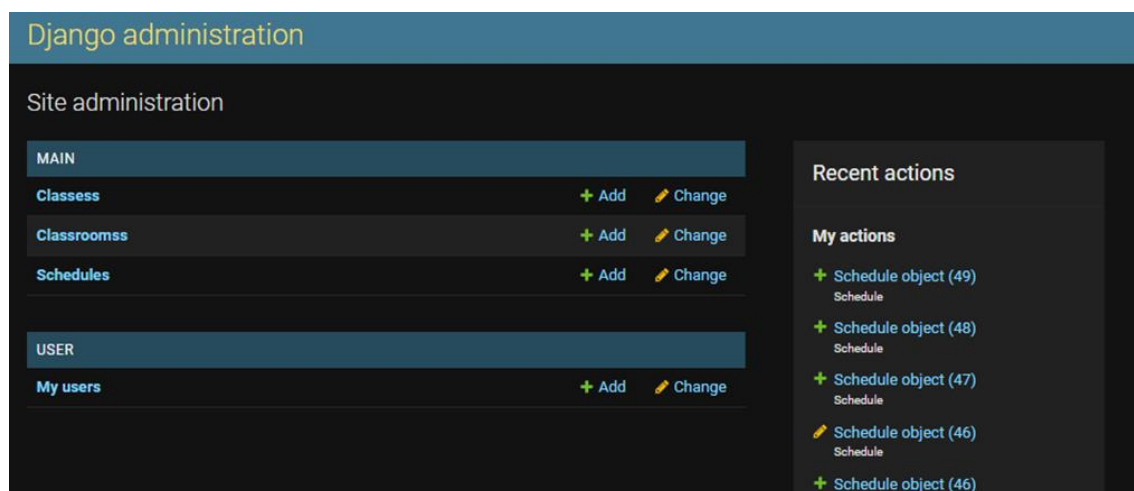


Рисунок 5.5 – Основний інтерфейс вчителя

Натиснувши на «My user» можемо створити нового користувача за допомогою зручної форми, що відображається на рисунку 5.6.

Рисунок 5.6 – Форма додавання користувачів.

Після чого ми можемо переглянути список всіх вже наявних користувачів на рисунку 5.7.

| EMAIL ADDRESS | NAME | ROLE | CLASSES | IS ACTIVE | IS ADMIN |
|----------------------|--------------------------|---------|---------|-----------|----------|
| admin2@gmail.com | - | - | - | ● | ● |
| andrii@gmail.com | Andrii Derevianko | Student | 11-A | ● | ● |
| anton@gmail.com | Битюко Антон | Student | 7-B | ● | ● |
| blydshyy@gmail.com | Блудший Ярослав Олегович | Student | 11-A | ● | ● |
| class_test@gmail.com | class test | Student | 11-A | ● | ● |
| dmytro@gmail.com | Кохась Дмитро | Student | 7-B | ● | ● |
| himara@gmail.com | Олена Хаймарс Петрівна | Teacher | - | ● | ● |
| jeka@gmail.com | Євгеній Проценко | Student | 7-B | ● | ● |
| lipady@gmail.com | Липа Денис Якович | Student | 11-B | ● | ● |
| max@gmail.com | Перепелица Максим | Student | 7-B | ● | ● |
| michael@gmail.com | Михайло Редько | Student | 7-B | ● | ● |
| pinchukm@gmail.com | Пінчук Леся Михайлівна | Teacher | - | ● | ● |
| roman@gmail.com | Роман Роман | Student | 7-B | ● | ● |
| teacher@gmail.com | teacher teacher | Teacher | 11-A | ● | ● |
| test2@gmail.com | test 2 | Student | 11-B | ● | ● |

Рисунок 5.7 – Список всіх користувачів.

Для додавання уроків в розклад маємо таку ж форму. Також існує можливість перегляду, редагування та видалення вже наявних уроків.

ВИСНОВКИ

У курсовій роботі було створено систему, яка надає можливість вчителям, учням та завучам оптимізувати навчальний процес. Був створений функціонал для кожної зі сторін.

По заданій предметній області було створено таблиці класів, користувачів, предметів, днів, часу, розкладу, оцінок, кабінетів в базі даних на основі класів-сутностей з відповідними взаємозв'язками та реалізовано логіку керування даними, виставлення оцінок, проглядання розкладу та формування його. Програмне забезпечення включає збереження даних при усіх операціях в базу даних і перевірку виключних ситуацій.

Для виконання курсової роботи було використано середовище розробки PyCharm. Для написання коду була використана мова програмування Python з фреймворком Django.

Як індивідуальний внесок було розроблено таблиці розкладу, днів, класних кімнат та часу уроків. Встановлено зв'язок між ними та заповнено відповідними даними. Надалі створено алгоритм формування розкладу відповідно користувача, що авторизувався, та вивід розкладу на відповідну сторінку. Уроки були розділені на блоки з усією потрібною інформацією: предметом, часом початку, кабінетом та вчителем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Data Flow Diagrams (DFD) Explained : Art of business analysis : вебсайт. URL: <https://www.artofba.com/uk/post/data-flow-diagrams-dfd-explained-1> (дата звернення: 25.12.2022)
2. UML для бизнес-моделирования: зачем нужны диаграммы процессов : Evergreen : вебсайт. URL: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> (дата звернення: 25.12.2022)
3. Что такое ER-диаграмма и как ее создать? : Lucidchart : вебсайт. URL: <https://www.lucidchart.com/pages/ru/erd-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0> (дата звернення: 25.12.2022)
4. Python Language advantages and applications : Geeksforgeeks URL: <https://www.geeksforgeeks.org/python-language-advantages-applications/> (дата звернення: 25.12.2022)
5. Meet Django : Django : вебсайт. URL: <https://www.djangoproject.com/> (дата звернення: 25.12.2022)
6. Git features : Geeksforgeeks : вебсайт. URL: <https://www.geeksforgeeks.org/git-features/> (дата звернення: 25.12.2022)

ДОДАТОК

Лістинг програмного модуля

НТУУ «КП» ІАТЕ ІПЗЕ ТВ-11

Листів 3

Київ – 2022

Функція формування розкладу

```
def index(request):
    current_user = request.user
    if current_user.role == "STUDENT":

        ''' Displaying schedule '''
        group = current_user.classes
        schedules = Schedule.objects.filter(classes=group)
        groups = Classes.objects.filter(class_id=group)
        for i in groups:
            name = i.name
        days = Days.objects.all()
        times = Time.objects.all()

        lessons = []

        for time in times:
            for day in days:
                counter = 0
                for lesson in schedules:
                    if lesson.day == day and lesson.time == time
and not counter:
                        lessons.append(lesson)
                        counter += 1
                if not counter:
                    lessons.append(None)

        content = {'today': starting_date, 'diary': diary,
'week_ago': week_ago, 'week_forward': week_forward,
                    'week': week, 'subjects': subjects,
'marksView': marksView, 'current_user': current_user,
                    'days': days, 'lessons': lessons, 'time':
times, 'name': name}
        return render(request, 'main/student-view.html',
content)

    elif current_user.role == "TEACHER":

        teacher_id = current_user.id
        schedules = Schedule.objects.filter(teachers=teacher_id)
        lessons = []

        for time in times:
            for day in days:
                counter = 0
```

```

        for lesson in schedules:
            if lesson.day == day and lesson.time == time
and not counter:
                lessons.append(lesson)
                counter += 1
            if not counter:
                lessons.append(None)
        content = {'days': days, 'time': times, 'lessons':
lessons, 'name': name, 'week_ago': week_ago,
                'week_forward': week_forward, 'week': week,
'today': starting_date, 'classes': classes,
                'subjects': subjects, 'students': students,
'marksView': marksView,
                'default_class': class_id, 'default_subject':
subject_id, 'default_date': starting_date,
                'form': form}

        return render(request, 'main/teacher-view.html',
content)
    else:
        return redirect('admin/')

```

Код шаблону для відображення розкладу учня

```

<div class="row d-flex justify-content-center" style="margin-
top: 3%;">
    {% for day in days %}
        <div class="col-lg-2 col-md-6 day text-center"><div
class="day-text">{{ day.name }}</div></div>
        {% endfor %}
</div>
<div class="row d-flex justify-content-center">
    {% for item in lessons %}
        {% if item != None %}
            <div class="col-lg-2 col-md-6 lesson text-center">
                <div class="time">{{ item.time }}</div>
                <div class="title">{{ item.subject }}</div>
                <div class="teacher">Викладач: {{ item.teachers
}}</div>
                <div class="room">Кабінет: {{ item.classroom }}</div>
            </div>
        {% elif item == None %}
            <div class="col-lg-2 col-md-6 lesson-empty">
                </div>
        {% endif %}
    {% endfor %}

```

```

    {% endfor %}
</div>

```

Код шаблону для відображення розкладу вчителя

```

<div class="row d-flex justify-content-center">
    {% for day in days %}
        <div class="col-lg-2 col-md-6 day text-center"><div
class="day-text">{{ day.name }}</div></div>
    {% endfor %}
</div>
<div class="row d-flex justify-content-center">
    {% for item in lessons %}
        {% if item != None %}
            <div class="col-lg-2 col-md-6 lesson text-center">
                <div class="time">{{ item.time }}</div>
                <div class="title">{{ item.subject }}</div>
                <div class="teacher">Клас: {{ item.classes }}</div>
                <div class="room">Кабінет: {{ item.classroom }}</div>
            </div>
            {% elif itme == None %}
                <div class="col-lg-2 col-md-6 lesson-empty">
                </div>
            {% endif %}
        {% endfor %}
    </div>

```