

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ  
«РОЗКЛАД АУДИТОРНИХ ЗАНЯТЬ СТУДЕНТІВ»

Виконала: студентка 4 курсу, групи 6.1260  
спеціальності 126 інформаційні системи та технології  
(шифр і назва спеціальності)  
освітньої програми інформаційні системи та штучний інтелект  
(назва освітньої програми)

С.С. Матюха

(ініціали та прізвище)

Керівник завідувач кафедри фундаментальної та прикладної  
математики, д.т.н., професор Гребенюк С.М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри комп'ютерних наук,  
доцент, к.т.н. Решевська К.С.

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

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 126 інформаційні системи та технології

(шифр і назва)

Освітня програма інформаційні системи та штучний інтелект

**ЗАТВЕРДЖУЮ**

Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

\_\_\_\_\_ Лісняк А.О.  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**

Матюхи Софії Сергіївни

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка інформаційної системи  
«Розклад аудиторних занять студентів»

керівник роботи Гребенюк Сергій Миколайович, д.т.н., професор  
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.  
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)  
1. Постановка задачі.  
2. Основні теоретичні відомості.  
3. Система управління інформаційною системою.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_  
презентація за темою доповіді

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.12.2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	09.01.2024	
2.	Збір вихідних даних.	23.01.2024	
3.	Обробка методичних та теоретичних джерел.	21.02.2024	
4.	Розробка першого та другого розділу.	09.04.2024	
5.	Розробка третього розділу.	20.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	22.06.2024	

Студент \_\_\_\_\_  
(підпис)

С.С. Матюха \_\_\_\_\_  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

С.М. Гребенюк \_\_\_\_\_  
(ініціали та прізвище)

## Нормоконтроль пройдено

Нормоконтролер \_\_\_\_\_  
(підпис)

А.В. Столярова \_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка інформаційної системи «Розклад аудиторних занять студентів»: 53 с., 26 рис., 26 джерел, 2 додатки.

БАЗА ДАНИХ, ФРЕЙМБОРК, CSS, DOCKER, HTML, JAVA SCRIPT, JQUERY, LARAVEL, MYSQL, NGINX, PHP, SQL.

Об'єкт дослідження – розклад аудиторних занять студентів.

Мета роботи: розробка та реалізація інформаційної системи для формування розкладу аудиторних занять студентів.

Методи дослідження – методи збору та аналізу вимог до програмного забезпечення, методи моделювання, проектування, розробки та тестування програмних рішень.

У кваліфікаційній роботі розроблено інформаційну систему «Розклад аудиторних занять студентів». При розробці інформаційної системи був проведений аналіз предметної області, обрана архітектура і платформа реалізації, розроблена функціональна модель, спроектована архітектура системи, реалізована інформаційна система з використанням мови програмування PHP, здійснене тестування.

Результати роботи можуть бути використані для подальшого застосування в освітній сфері при створенні розкладу аудиторних занять студентів, а також автоматизованого розподілу навчального навантаження викладачів.

## SUMMARY

Bachelor's qualifying paper «Development of the Information System “Schedule of Classroom Classes of Students”»: 53 pages, 26 figures, 26 references, 2 supplements.

DATABASE, FRAMEWORK, CSS, DOCKER, HTML, JAVA SCRIPT, JQUERY, LARAVEL, MYSQL, NGINX, PHP, SQL.

The object of the study is the schedule of students' lessons.

The aim of the study is to develop and implement an information system for forming a schedule of student's lessons.

The methods of research are methods of collecting and analysing software requirements, methods of modelling, designing, developing and testing software solutions.

In the qualification work, the information system ‘Schedule of Classroom Classes of Students’ has been developed. During the development of the information system, the subject area has been analysed, the architecture and implementation platform have been selected, a functional model has been developed, the system architecture has been designed, the information system has been implemented using the PHP programming language, and testing has been carried out.

The results of the work can be used for further application in the educational sphere to create schedules for students, as well as automated distribution of the teaching load of teachers.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Вступ.....	7
1 Технічне завдання .....	9
1.1 Характеристика та призначення фреймворку Laravel.....	9
1.2 Огляд готових кейсів .....	11
1.3 Створення сайту на Laravel.....	12
1.4 Призначення та мета створення інформаційної системи.....	13
1.5 Вимоги до системи.....	14
1.6 Огляд технологій.....	16
2 Моделювання інформаційної системи.....	17
2.1 Дослідження предметної області.....	17
2.2 Проектування інформаційної моделі .....	21
2.3 Схема варіантів використання.....	24
2.4 Схема взаємодії та послідовності.....	25
2.5 Діаграма IDEF4.....	28
2.6 Діаграма розгортання та компонентів .....	30
3 Розробка інформаційної системи .....	33
3.1 Розробка дизайну. Графічний інтерфейс.....	33
3.2 Створення інформаційної системи.....	38
3.3 Створення бази даних та робота з даними .....	39
Висновки .....	40
Перелік посилань.....	42
Додаток А Програмна реалізація бази даних .....	45
Додаток Б Програмний код.....	47

## ВСТУП

У сучасному світі, де швидкість і доступність інформації є ключовими чинниками ефективного функціонування будь-якої системи, інформаційні технології стають невід'ємною складовою будь-якої сфери життя. Особливо це стосується освіти, де потрібно забезпечити оптимальне використання часу та ресурсів для ефективного навчання.

Метою кваліфікаційної роботи бакалавра є розробка та реалізація сайту для формування розкладу аудиторних занять студентів. Вона полягає в створенні зручного та функціонального інструменту, який допоможе автоматизувати процес планування та розподілу навчальних занять, забезпечуючи оптимальне використання ресурсів та задоволення потреб користувачів.

Задачі, які необхідно розв'язати для досягнення поставленої мети:

- проаналізувати існуючі підходи до управління розкладом у навчальних закладах, їхні переваги та недоліки;
- проаналізувати переваги та недоліки;
- сформулювати вимоги до інформаційної системи;
- спроектувати схему даних;
- реалізувати базу даних;
- протестувати інформаційну систему.

Об'єктом дослідження є розклад навчального процесу в університеті.

Предмет дослідження – інформаційна система «Розклад аудиторних занять студентів».

Методи дослідження: методи програмної інженерії, системний аналіз, об'єктно-орієнтоване програмування.

Розробка програмного забезпечення, представленого в роботі, дозволить автоматизувати процес планування розкладу, забезпечуючи зручний та сучасний інтерфейс для користувачів, а також швидкий доступ до

актуальної інформації про розклад. Вона базується на використанні сучасних технологій програмування та методів управління даними для створення ефективної та надійної інформаційної системи.

Процес розробки даної інформаційної системи включає в себе: аналіз потреб користувачів, проектування структури даних, розробку програмного забезпечення та тестування його ефективності. З моменту впровадження ця система буде сприяти оптимізації процесу планування розкладу, підвищенню доступності інформації та забезпеченню більшої ефективності навчального процесу.

В кваліфікаційній роботі буде представлено детальний огляд вимог до системи, проектування та реалізації програмного забезпечення, а також результати тестування його функціональності. Результати цієї роботи сприятимуть подальшому розвитку та вдосконаленню інформаційних систем у сфері освіти, забезпечуючи більш ефективне та продуктивне використання ресурсів для навчання студентів та оптимізації роботи викладачів.



# 1 ТЕХНІЧНЕ ЗАВДАННЯ

## 1.1 Характеристика та призначення фреймворку Laravel

Фреймворк (англ. framework – каркас) – це програмне середовище, яке спрощує та прискорює створення програмного забезпечення.

Laravel – це PHP-фреймворк з відкритим вихідним кодом, розроблений Тейлором Отуеллом (англ. Taylor Otwell) у 2011 році і призначений для розробки вебдодатків відповідно до шаблону model-view-controller (MVC).

Кожна нова версія Laravel удосконалюється новими вебрішеннями та можливостями. Фреймворк по праву заслужив звання найнадійнішого та найвитривалішого PHP-фреймворку за свою систему безпеки та безліч модулів, які допомагають створювати високопродуктивні та масштабовані проекти [19].

Наразі, Laravel – це високорівневий вебфреймворк для мови програмування PHP, який став дуже популярним серед веброзробників завдяки своїй простоті використання, елегантному синтаксису та багатофункціональності. Розглянемо деякі характеристики та призначення фреймворку Laravel [26]:

- елегантний синтаксис Laravel, що полегшує розробку вебдодатків (він спрощує та прискорює процес розробки, дозволяючи розробникам фокусуватися на бізнес-логіці додатку, а не на надмірному шаблонному коді);
- маршрутизація: надає простий та потужний механізм для визначення URL-адрес (маршрутів) та їх обробки відповідними контролерами або функціями;
- ORM (Object-Relational Mapping): вбудований ORM в Laravel, який називається Eloquent, дозволяє розробникам взаємодіяти з базою

даних за допомогою об'єктно-орієнтованих моделей, що значно спрощує роботу з базою даних;

- шаблонізатор Blade: Blade – це потужний шаблонізатор, що використовується в Laravel для відображення HTML-шаблонів (він має простий та зрозумілий синтаксис і підтримує такі функції, як наслідування шаблонів та включення підшаблонів);
- міграції баз даних: фреймворк надає механізм міграцій, який дозволяє розробникам легко керувати структурою бази даних в процесі розробки та міграції додатків;
- аутентифікація та авторизація Laravel – дозволяє швидко реалізувати механізми входу користувачів та контролю доступу до різних частин додатку;
- система маршрутизації: простий та ефективний механізм для керування доступом до маршрутів на основі HTTP-запитів та авторизаційних правил;
- інтеграція з сторонніми сервісами: Laravel має широкий набір розширень та пакетів, що дозволяють легко інтегрувати додаток з різними сторонніми сервісами, такими як платіжні системи, соціальні мережі тощо;
- підтримка тестування: Laravel надає зручні інструменти для написання та виконання автоматизованих тестів, що допомагає забезпечити якість коду та його надійність.

Виходячи з вищезазначеного, можемо сказати, що Laravel дозволяє розробникам створювати потужні та масштабні вебдодатки швидко та ефективно за допомогою високоякісних інструментів та розширень. Laravel наразі є одним з найпопулярніших фреймворків – він заснований на поширеній і доступній мові PHP, має відкритий програмний код (open source), надає детальну та зручну документацію, постійно оновлюється та спирається на величезне міжнародне ком'юніті розробників.

## 1.2 Огляд готових кейсів

Огляд готових кейсів використання фреймворку Laravel може допомогти зрозуміти, як цей інструмент використовується в різних галузях та для різних типів проєктів. Нижче розглянемо декілька прикладів кейсів використання Laravel [5, 11, 12].

*Платформи SaaS (Software as a Service).* Компанія Buffer використовує Laravel для розробки своєї платформи з управління соціальними медіа. Laravel дозволив їм швидко розвивати та масштабувати свій продукт, забезпечуючи ефективність та надійність.

*Електронна комерція.* Flare by Laravel – це додаток для моніторингу та відстеження помилок у Laravel-додатках. Власники використовують цей продукт для отримання повної картини про взаємодію їх додатка з користувачами.

*CMS (Content Management System).* Statamic – це гнучкий та потужний CMS, який базується на Laravel. Statamic надає можливості управління вмістом, які зрозумілі та легкі в освоєнні.

*CRM (Customer Relationship Management).* Perfex CRM – це повнофункціональний CRM з рядом інтегрованих модулів, який реалізований з використанням Laravel. Цей продукт дозволяє компаніям ефективно вести облік клієнтів та взаємодіяти з ними.

*Соціальні мережі.* Kiborush – це платформа для автоматизації соціальних медіа, яка побудована на Laravel. Вона дозволяє користувачам автоматизувати публікації та взаємодіяти з аудиторією в різних соціальних мережах.

*Аналітика та звітність.* Laravel Analytics – це пакет для Laravel, який дозволяє зручно взаємодіяти з різними сервісами вебаналітики, такими як Google Analytics, та отримувати детальні звіти про трафік та поведінку користувачів.

Розглянуті приклади демонструють різноманітність використання

Laravel у різних галузях та для різних типів проєктів. Фреймворк надає потужні інструменти для швидкої та ефективної розробки різноманітних вебдодатків.

### 1.3 Створення сайту на Laravel

Процес створення сайту на Laravel може бути досить простим та ефективним завданням, особливо для тих, хто має базові знання фреймворку та розуміє основні концепції веброзробки. Нижче розглянемо основні кроки, які допоможуть у створенні сайту на Laravel.

*Крок 1.* Встановлення Laravel на комп'ютері або сервері. Це можливо зробити за допомогою Composer, який є пакетним менеджером для PHP. Просто вводимо команду «composer create-project --prefer-dist laravel/laravel [назва-папки]», де [назва-папки] – це назва проєкту.

*Крок 2.* Конфігурація бази даних: Laravel використовує конфігураційний файл .env для налаштування параметрів, таких як з'єднання з базою даних. Необхідно відредагувати цей файл, щоб вказати параметри своєї бази даних.

*Крок 3.* Створення маршрутів. Laravel використовує файл «routes/web.php» для визначення URL-адрес (маршрутів) та їх обробки відповідними контролерами або функціями. Потрібно визначити необхідні маршрути для свого сайту.

*Крок 4.* Створення контролерів та моделей для сайту. Контролери в Laravel відповідають за обробку логіки запитів, а моделі – за взаємодію з базою даних.

*Крок 5.* Шаблонізація. Створюємо шаблони для різних сторінок сайту. Laravel використовує шаблонізатор Blade для відображення HTML-шаблонів.

*Крок 6.* Взаємодія з базою даних. Створюємо необхідні моделі та виконуємо запити до бази даних. Для цього використовуємо Eloquent,

вбудований ORM, для взаємодії з базою даних.

*Крок 7.* Аутентифікація та авторизація.

*Крок 8.* Реалізація функціональності сайту, включаючи сторінки, форми, обробку запитів, взаємодію з базою даних тощо.

*Крок 9.* Тестування сайту, щоб впевнитися, що всі функції працюють як очікується, і виправлення помилок.

*Крок 10.* Розгортання сайту на виробничому сервері та перевірка, щоб він працював належним чином.

Це загальний огляд процесу створення сайту на Laravel. Кожен конкретний проєкт може мати свої власні особливості та вимоги, але ці кроки допоможуть розпочати роботу та створити базовий функціональний сайт [5, 12].

#### **1.4 Призначення та мета створення інформаційної системи**

Будь-яка база даних створюється для вирішення маси різнотипних проблем, але основна мета створення інформаційної системи – це пошук, аналіз та збереження інформації. Такі інформаційні масиви, функціонування яких підтримується сучасними комп'ютерами, дозволяють зберігати величезні кількості даних та швидко надають доступ до будь-якого необхідного файлу. Крім цього існує велика кількість інших цілей та завдань, зокрема:

- забезпечення доступу до різноманітної інформації для користувачів (це може бути інформація про товари, послуги, клієнтів, операції, наукові дані, тощо);
- автоматизація рутинних або складних процесів, що сприяє збільшенню ефективності роботи та зменшенню людського втручання;
- підтримка прийняття рішень на основі обробки даних;

- оптимізація використання ресурсів, таких як час, праця, гроші, матеріали тощо, за допомогою інформаційної системи;
- покращення комунікації між різними частинами організації або між різними сторонами бізнес-процесів;
- задоволення потреб користувачів, незалежно від того, чи це клієнти, співробітники, менеджери чи адміністратори;
- підвищення продуктивності та ефективності бізнес-процесів та роботи користувачів;
- забезпечення безпеки та захисту конфіденційної та важливої інформації від несанкціонованого доступу та зберігання.

Отже, призначення та мета створення інформаційної системи полягають у забезпеченні доступу до інформації, автоматизації процесів, підтримці прийняття рішень, оптимізації ресурсів та покращенні комунікації, що призводить до підвищення продуктивності та ефективності організації [11, 22].

### **1.5 Вимоги до системи**

Загальні вимоги, виокремлені та реалізовані в ході виконання кваліфікаційної роботи, розподілені за категоріями: управління розкладом, відображення розкладу, автоматичне оновлення та сповіщення, безпека та доступність [16]. Розглянемо детальніше кожну з них.

Управління розкладом занять:

- можливість створення, зміни та видалення розкладу для різних груп студентів, викладачів та аудиторій;
- підтримка різних типів занять, таких як лекції, семінари, практичні заняття, лабораторні роботи тощо;
- можливість планування розкладу на різні періоди, наприклад, на тиждень, семестр або навчальний рік.

#### Відображення розкладу:

- зручний інтерфейс для відображення розкладу занять для студентів;
- можливість фільтрувати розклад за різними параметрами, такими як група студентів, курс, день тижня, час тощо;
- підтримка можливості відображення розкладу на мобільних пристроях та інших платформах.

#### Автоматичне оновлення та сповіщення:

- можливість автоматичного оновлення розкладу при змінах або додаванні нових занять;
- підтримка сповіщень студентів про зміни у розкладі через повідомлення на платформі.

#### Можливість пошуку та фільтрації:

- пошук занять за різними критеріями, такими як назва предмету, викладач, аудиторія тощо;
- можливість фільтрувати розклад за часом, місцем проведення, типом заняття та іншими параметрами.

#### Безпека та доступність:

- захист від несанкціонованого доступу до розкладу занять студентів;
- забезпечення доступності системи у будь-який час для користувачів.

Ці вимоги допоможуть забезпечити ефективне функціонування інформаційної системи «Розклад занять студентів» та забезпечити зручну та ефективну організацію навчального процесу для всіх учасників освітнього процесу.

Інформаційна система, створена для даної роботи, представлена у вигляді ресурсу (сайту). Тому, для її коректного застосування, необхідно реалізувати дотримання лише двох вимог:

- наявність технічного засобу – ПК, планшету чи смартфона;
- наявність інтернет-підключення.

## 1.6 Огляд технологій

Для досягнення мети роботи та розв'язання поставлених задач, були застосовані різні технології [21, 25]. Короткий огляд технологій для інформаційної системи «Розклад занять студентів» представлено нижче.

*База даних MySQL.* Це потужна реляційна база даних, яка широко використовується в веброзробці для зберігання структурованих даних.

*Фреймворки та бібліотеки.*

Laravel – це PHP-фреймворк з елегантним та експресивним синтаксисом, що спрощує розробку вебдодатків.

jQuery – JavaScript-бібліотека, яка спрощує взаємодію з HTML-документами, обробку подій, анімацію та AJAX.

*Мови програмування.*

PHP (Hypertext Preprocessor) – скриптова мова програмування, що часто використовується для створення вебдодатків.

JavaScript – мова програмування, яка використовується для реалізації динамічного контенту на вебсторінках.

*HTML та CSS.*

HTML (Hypertext Markup Language) – мова розмітки, яка використовується для створення структури та вмісту вебсторінок.

CSS (Cascading Style Sheets) – мова стилів, яка використовується для задання вигляду та форматування вебдокументів.

Інструментом розгортання є Docker – платформа для розгортання, керування та запуску вебдодатків в контейнерах.

Вебсервер Nginx з відкритим вихідним кодом, який використовується для обробки запитів HTTP та HTTPS.



## 2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Дослідження предметної області

Розвиток сучасного суспільства характеризується процесом інформатизації і, у першу чергу, інформатизації освіти.

Інформатизація – сукупність взаємопов'язаних організаційних, правових, політичних, соціально-економічних, науково-технічних, виробничих процесів, які спрямовані на створення умов для задоволення інформаційних потреб громадян та суспільства на основі створення, розвитку і використання інформаційних систем, мереж, ресурсів та інформаційних технологій, які побудовані на основі застосування сучасної обчислювальної та комунікаційної техніки [6]. Одним із пріоритетних завдань програми визначено створення загальнодержавної мережі інформаційного забезпечення науки, освіти, культури, охорони здоров'я.

Однозначного розуміння та єдиної стратегії розвитку процесу інформатизації освіти немає, тому інформатизацію освіти слід розглядати як процес зміни змісту, методів та організаційних форм підготовки студентів на етапі переходу її до життя в умовах інформаційного суспільства, створення і використання інформаційних технологій для підвищення ефективності видів діяльності, що здійснюються в системі освіти.

Концепція інформатизації сфери освіти найповніше відображена в нормативних документах: Державній національній програмі «Освіта. Україна XXI століття», Законах України «Про загальну середню освіту», «Про вищу освіту», Державній програмі «Інформаційні та комунікаційні технології в освіті і науці» на 2006–2010 роки, «Про Основні засади розвитку інформаційного суспільства в Україні на 2007–2015 роки», Державній цільовій науково-технічній програмі, Постанові Кабінету Міністрів України від 17 березня 2004 р. № 326 «Про затвердження Положення про

Національний реєстр електронних інформаційних ресурсів», Національній доктрині розвитку освіти України у XXI столітті, Державній програмі «Вчитель», «Концептуальні засади розвитку педагогічної освіти України та її інтеграції в Європейський освітній простір»; концепції розвитку дистанційної освіти в Україні [20], де на основі інформаційних технологій представлено основний задум вирішення низки проблем національної сфери освіти України, а також систему взаємопов'язаних цілей, завдань, напрямів, методів і засобів інформатизації освіти, та в інших нормативних документах, які спрямовані на подолання відставання України від розвинених країн Заходу в становленні інформаційного суспільства. Проте, всі вони не внесли до реформування освіти цілісних структурних змін, а мали переважно фрагментарний характер [7].

Макаренко Л. Л., аналізуючи різні концепції інформатизації освіти виділяє такі її напрями [10]:

- зміна бази освіти на основі інформаційних технологій (матеріальної, навчальної, методичної, інформаційної) і створення сервісних центрів її обслуговування;
- зміна цілей і змісту освіти, викликана становленням фундаментального загальноосвітнього курсу інформатики на всіх щаблях освіти, необхідністю формування інформаційної культури студентів в умовах інформаційного суспільства;
- підвищення ефективності, доступності та якості освіти – досягнення якісно нового рівня освіти за рахунок інтеграції інформаційних технологій у навчання, розвиток і виховання;
- підготовка і перепідготовка педагогічних і управлінських кадрів освіти до здійснення інформаційно-педагогічної діяльності на інформаційній основі;
- інтенсифікація науково-дослідної і науково-методичної діяльності в освітніх установах;
- використання інформаційних технологій в управлінні освітніми

- установами на різних рівнях, аж до всієї системи освіти загалом;
- створення єдиного загальноосвітнього простору України та інтеграція національної системи освіти в інформаційну освітню інфраструктуру світової та євроспільноти;
  - вдосконалення нормативно-правової бази науково-освітньої і навчальної діяльності на основі інформаційних технологій;
  - уточнення змісту освітніх програм, моделей і методик;
  - розвиток індустрії освітніх послуг;
  - забезпечення контролю якості інформаційно-освітніх технологій;
  - забезпечення інформаційної безпеки єдиного освітнього інформаційного середовища [10].

Інформатизація освіти сприяє підвищенню ефективності та інтенсифікації навчального процесу за рахунок використання інформаційних технологій і впровадження нових методичних розробок в процес навчання, дає змогу ефективно витрачати державні ресурси й надавати якісні знання, формувати навички, компетентності відповідно до викликів суспільства на певному етапі його розвитку [10].

Інтеграція інформаційних технологій в освіту дозволить створити нову якість інформаційно-педагогічної діяльності, а саме:

- покращити організаційні умови навчального процесу: підвищити ефективність навчання шляхом впровадження нових механізмів наочності й інтерактивності навчального процесу, використовувати різноманітні джерела навчальної інформації, оптимізувати темп роботи студентів, ефективно реалізувати міжпредметні зв'язки;
- поліпшити психолого-педагогічні умови навчальної діяльності, створивши у студента інтерес і позитивний мотив через природний інтерес до комп'ютера, включенням механізму розвитку у студентів дослідницьких, творчих якостей шляхом рефлексії, самореалізації, самопізнання;
- сприятиме індивідуалізації, диференціації, інтенсифікації освіти і,

як наслідок, її оптимізації і вдосконаленню.

У результаті інформатизації освіти Україна інтегрується у світовий освітній простір, отримає ефективну єдину інформаційну освітню мережу, яка забезпечить якісне навчання та підготовку фахівців, що потрібні країні.

Сьогодні дослідники [10, 16] виділяють різні тенденції сучасного етапу інформатизації освіти, з яких найбільш важливими є:

- оснащення освітніх закладів засобами інформаційних технологій та їх використання в якості нового педагогічного інструменту та підтримки процесу навчання;
- зміна змісту освіти внаслідок розвитку інформатизації суспільства, а також об'єднання переваг традиційної освіти з можливостями інформаційних технологій;
- створення єдиного інформаційного освітнього простору, який забезпечує доступність якісної інформації.

Сучасний етап інформатизації вищої школи сприяв виникненню такої стратегічної цілі, як глобальна раціоналізація інтелектуальної діяльності за рахунок використання інформаційних технологій, радикальне підвищення ефективності і якості підготовки спеціалістів. Досягнення цієї мети залежить від масової комп'ютерної грамотності і сформованості інформаційної культури, тобто вміння використовувати засоби інформаційних технологій від телефону, комп'ютера та комп'ютерних мереж до користування найбільш розповсюдженими програмними продуктами, знання особливостей потоків інформації в своїй сфері діяльності, вміння виявляти її та ефективно використовувати [14].

Сучасний процес інформатизації освіти зумовив появу ряду завдань, що потребують вирішення [18]:

- підвищення рівня підготовки фахівців за рахунок покращення технологій навчання (впровадження в учбовий процес інформаційних, телекомунікаційних засобів, зокрема можливостей мережі Інтернет);

- адаптація інформаційних технологій навчання цілям та умовам навчального процесу;
- розробка програмно-методичної основи застосування інформаційних технологій навчання, які сприяють активізації пізнавальної діяльності студентів та підвищенню їх мотивації;
- формування у студентів навичок обробки, аналізу та перерозподілу інформації за допомогою інформаційних технологій з метою їх подальшого ефективного використання в самостійній професійній діяльності.

Таким чином, проблема ефективного використання викладачами інформаційних технологій особливо актуальна в наш час.

## **2.2 Проектування інформаційної моделі**

Проектування інформаційної моделі є важливим етапом у розробці будь-якої інформаційної системи, включаючи систему «Розклад аудиторних занять студентів». Інформаційна модель описує структуру та організацію даних, що використовуються в системі, а також взаємозв'язки між ними. Для зручності представлення інформації, використаємо ER діаграму.

ER-діаграми – це візуальний інструмент, який корисний для представлення моделі ER. Пітер Чень запропонував у 1971 р. Створити єдину конвенцію, яка може бути використана для реляційних баз даних та мережі. Він прагнув використати модель ER як концептуальний підхід до моделювання [3].

Перше, що необхідно зробити, визначити сутності, які необхідно зберігати в базі даних.

Друге – визначаємо типи зв'язків між сутностями.

Визначеної інформації досить для того, щоб побудувати модель даних верхнього рівня – рівня сутностей. Метою такої діаграми є формування

загального погляду на систему для її подальшої деталізації.

У редакторі сутності можна ввести текстові описи сутностей. Ці визначення дозволяють зрозуміти, який об'єкт представляє сутність і зробити більш ефективно обговорення структури даних з експертами предметної області на ранніх етапах створення моделі.

Атрибути виражають визначені властивості сутності. Визначаємо набори атрибутів кожної сутності за допомогою редактора атрибутів. При описі атрибутів сутностей один або кілька атрибутів повинні бути обрані як первинний ключ сутності. Первинний ключ дозволяє однозначно ідентифікувати екземпляр сутності. На ER діаграмі первинні ключі відображаються у верхній частині сутності, а інші атрибути – у нижній частині. Будемо ідентифікувати екземпляри всіх сутностей по їхньому номері.

Сутності, які необхідно зберігати в базі даних дослідження: адміністратор, викладач, предмет, група, аудиторія. Розглянемо сутності та їхні атрибути

Група – сутність, яка представляє групу студентів, які навчаються разом. Атрибути сутності «група»: номер, курс, факультет.

Предмет – сутність, яка визначає окремий навчальний предмет або курс. Атрибути сутності «предмет»: назва, номери груп, в яких викладається.

Викладач. Сутність, яка представляє викладача, який проводить заняття. Атрибути сутності: ПІБ, предмет, науковий ступінь, форма занять.

Аудиторія. Сутність, яка визначає конкретну аудиторію, в якій проводяться заняття. Атрибути сутності – номер аудиторії.

Розклад: Сутність, яка містить інформацію про розклад занять для певної групи студентів. Атрибути: дата, номер заняття, викладач, предмет, тип заняття, аудиторія.

ER-діаграма інформаційної системи «Розклад аудиторних занять студентів» приведена на рисунку 2.1.

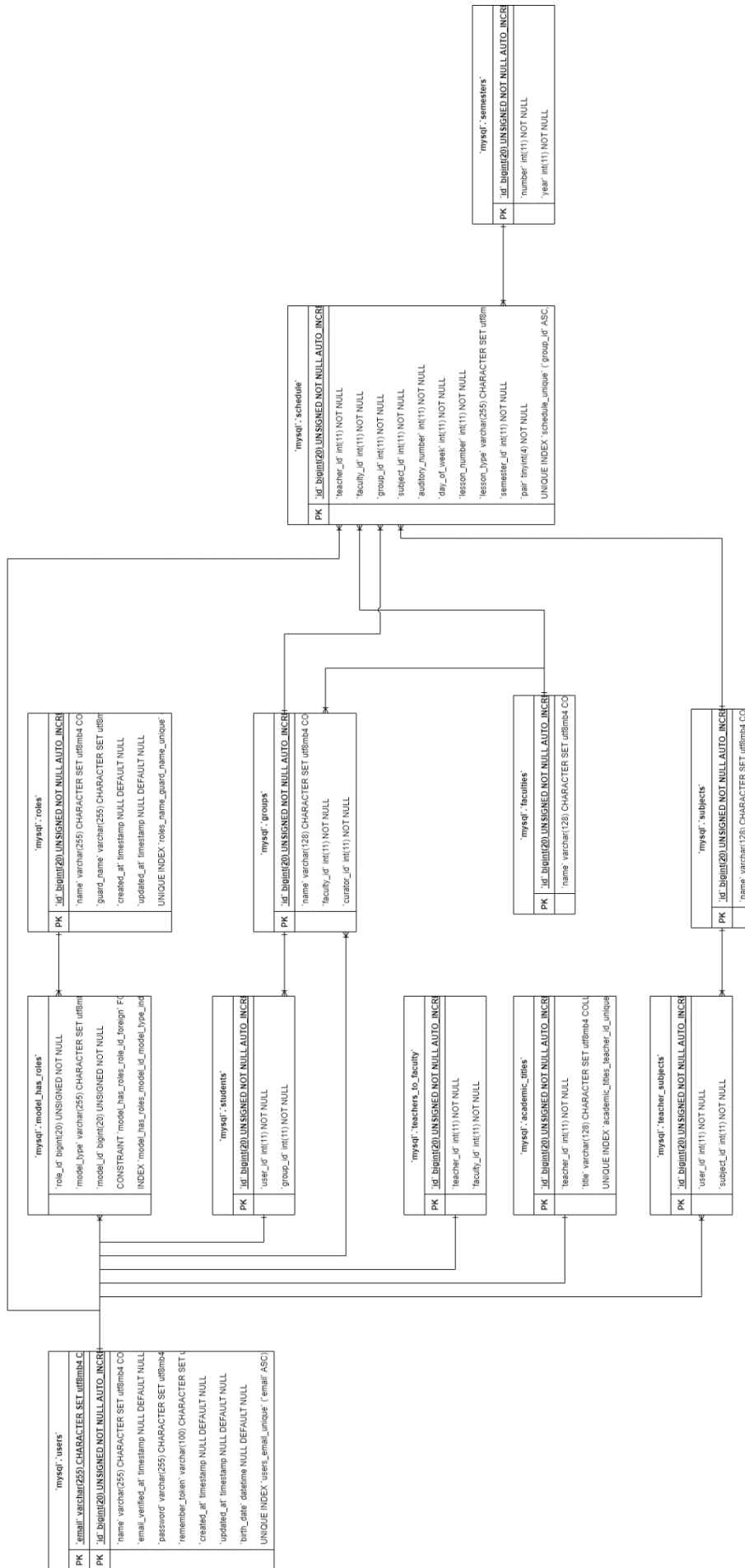


Рисунок 2.1 – ER-діаграма «Розклад аудиторних занять студентів»

ER-діаграма зображена за допомогою трьох конструктивних елементів:

- сутність – конструктивний елемент ER-моделі, призначений для завдання класу однорідних і помітних між собою об'єктів предметної області (класу відповідає тип сутності, конкретного об'єкта з класу – екземпляр сутності);
- атрибут – конструктивний елемент ER-моделі, призначений для завдання властивості сутності (або зв'язку);
- зв'язок – конструктивний елемент ER-моделі, призначений для вказівки відносини між сутностями.

### **2.3 Схема варіантів використання**

Схема варіантів використання, або діаграма прецедентів – це діаграма, на якій зображено відношення між акторами та прецедентами в системі.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених межею системи, асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання [24].

Схема варіантів використання інформаційної системи «Розклад аудиторних занять студентів» приведена на рисунку 2.2.

Суть діаграми прецедентів полягає в тому, що проєктована система подається у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Кожен варіант використання визначає деякий набір дій, який виконує система під час діалогу з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів із системою.



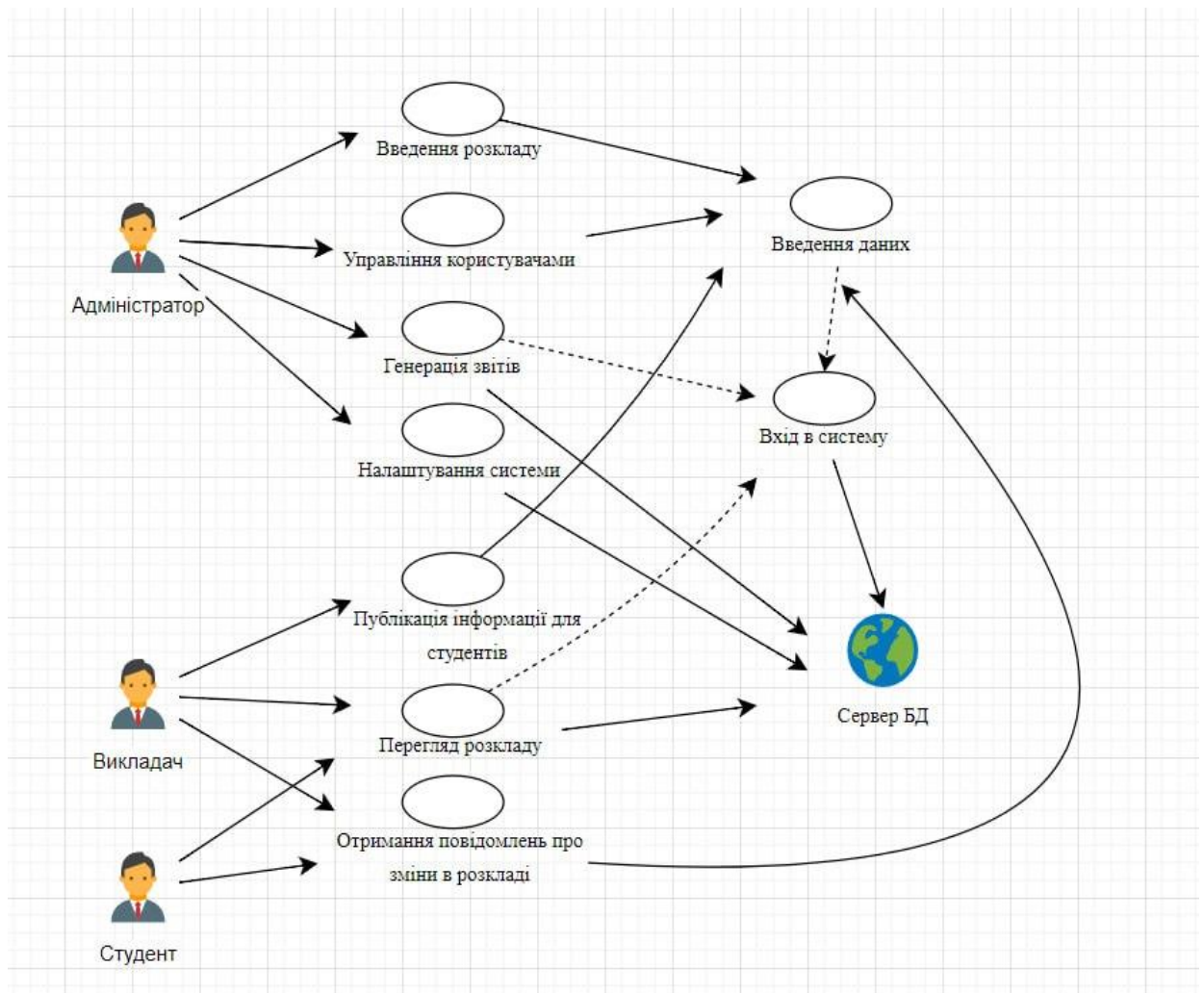


Рисунок 2.2 – Схема варіантів використання інформаційної системи «Розклад аудиторних занять студентів»

## 2.4 Схема взаємодії та послідовності

Діаграма послідовності – діаграма, на якій показані взаємодії об'єктів, упорядковані за часом їхнього прояву.

Створимо діаграму послідовності для інформаційної системи «Розклад аудиторних занять студентів», щоб проілюструвати взаємодію між класами при перегляді розкладу занять.

Взаємодію класів інформаційної системи «Розклад аудиторних занять студентів» представлено на рис. 2.3 та 2.4.

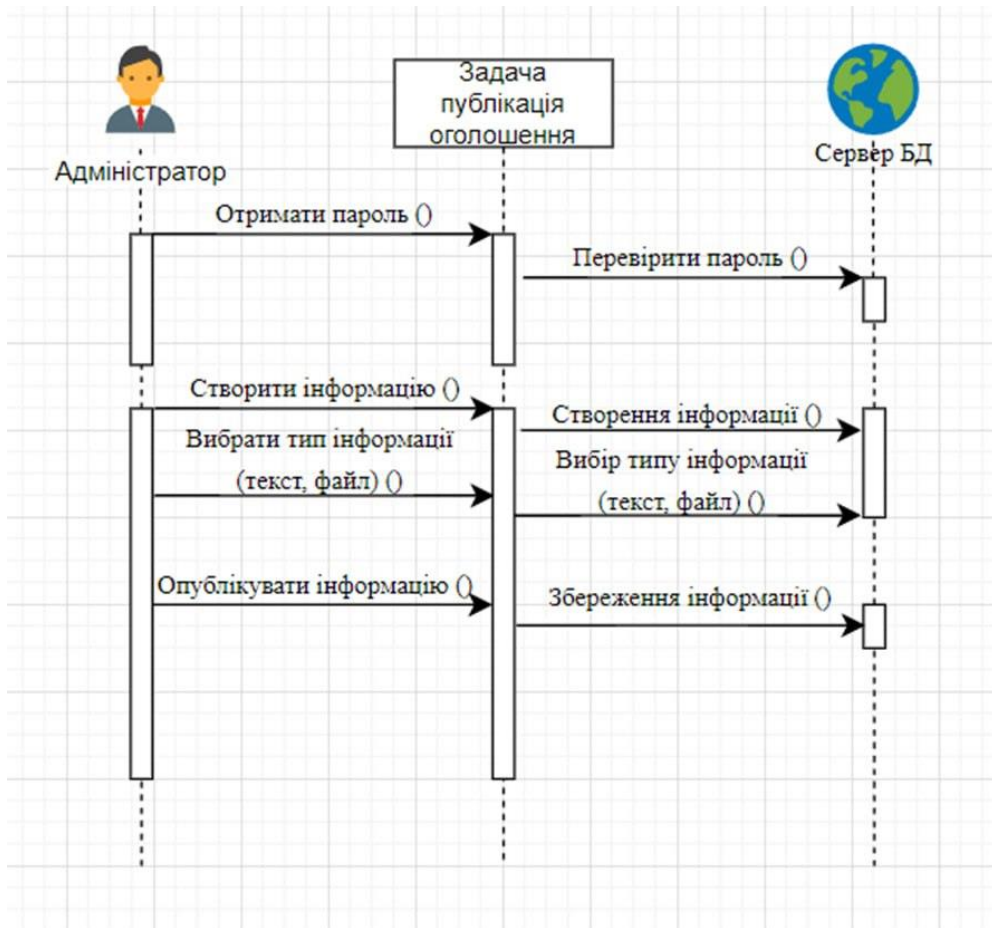


Рисунок 2.3

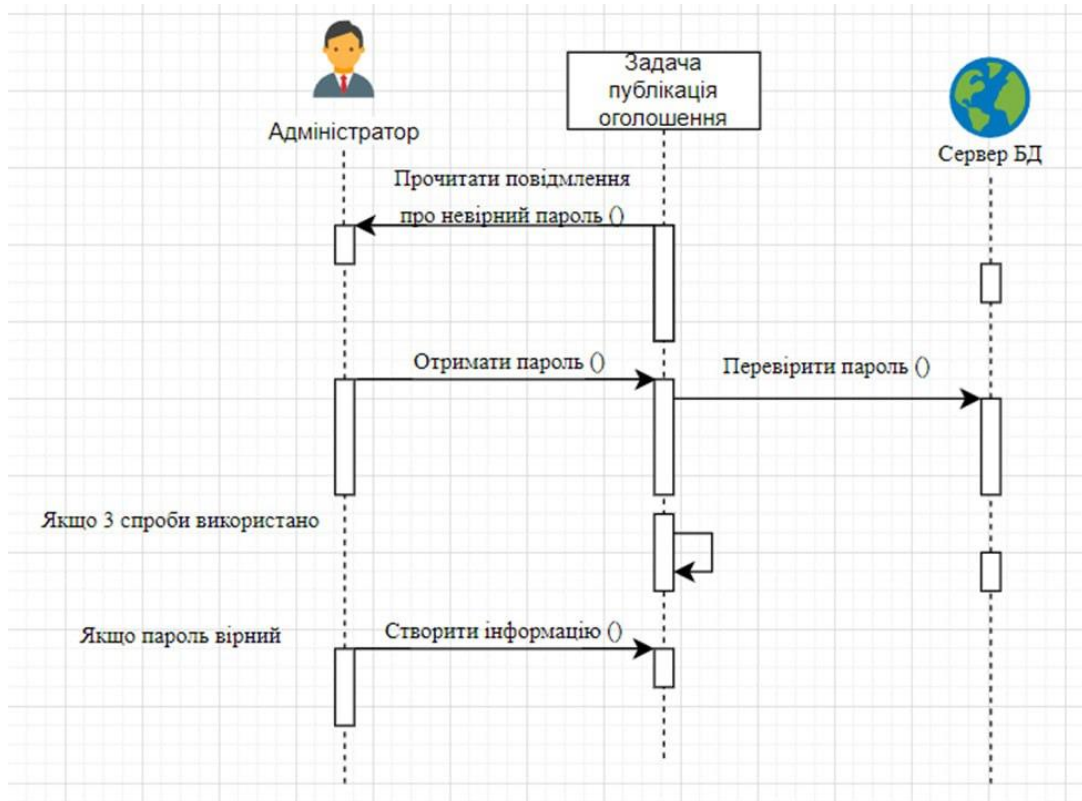


Рисунок 2.4

На рисунку 2.3 представлено взаємодію між користувачем (адміністратором) та системою (сервер БД).

Адміністратор починає запитом на публікацію, вводить пароль для авторизації. Система перевіряє пароль. У разі його підтвердження, користувач створює інформацію, обираючи тип (текст або файл) та публікує її. Система зберігає інформацію.

У разі введення невірному паролю, користувач отримує повідомлення від системи про невірний пароль. Користувач має повторно ввести пароль та дочекатися підтвердження від системи. Після його підтвердження системою, відбуваються дії, описані вище (створення та публікація інформації). Всього є три спроби введення паролю. Після того, протягом 30 хвилин неможливо долучитись до системи – вона заблокується. Для відновлення паролю необхідно авторизуватись через електронну поштову скриньку.

Процес взаємодії між адміністратором та системою під час виконання задачі «публікація оголошення» для студентів представлено на рис. 2.5 та 2.6

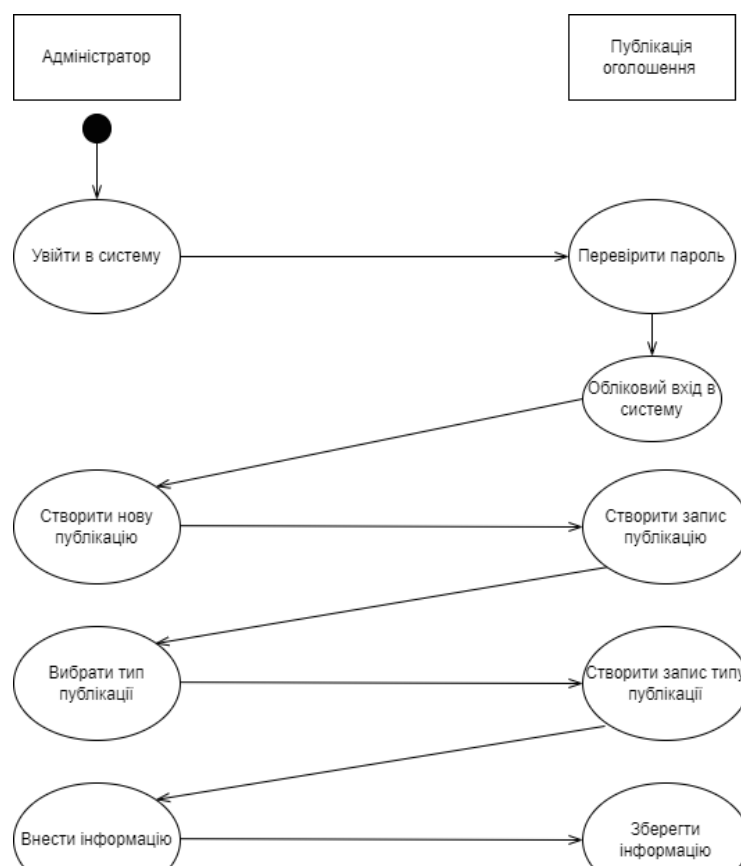


Рисунок 2.5

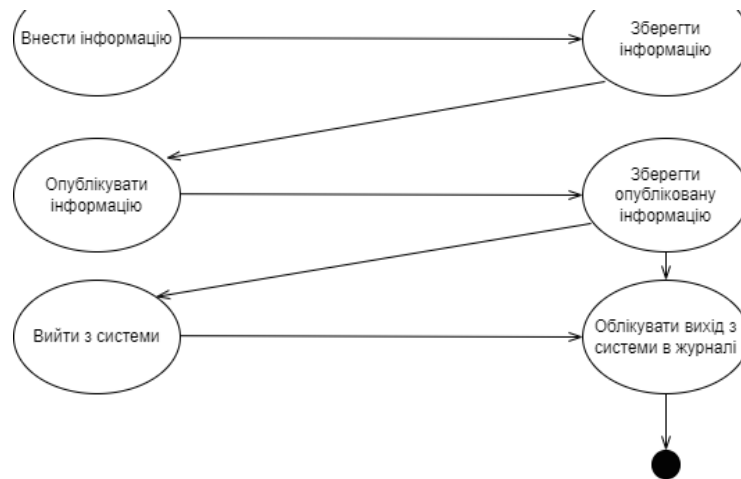


Рисунок 2.6

Розглянуті діаграми послідовностей дають змогу візуалізувати процес взаємодії між об'єктами або ролями в системі і відобразити послідовність дій, які вони виконують. В свою чергу, це допомагає нам розуміти, як працює система та як взаємодіють її складові частини.

## 2.5 Діаграма IDEF4

IDEF4 – графічно орієнтована методологія для проєктування об'єктно-орієнтованих програмних систем. Об'єктно-орієнтована парадигма програмування надає розробникам абстрактне бачення їх програми, у вигляді утвореної з певної кількості станів основних об'єктів, що визначають поведінку програми протоколом їх взаємодії. Об'єкт містить набір внутрішніх станів, що визначаються атрибутами та набором методів (процедур), які, в свою чергу, визначають поведінку конкретних об'єктів і їх взаємозв'язків з іншими об'єктами, що утворюють систему.

Метод IDEF4 багатовимірного підходу до об'єктно-орієнтованого проєктування програмного забезпечення системи складається з наступних пунктів:

- проєктування шарів;
- стан проєктування артефактів;

- проектування моделей і компонент обґрунтування дизайну;
- проектування особливостей, починаючи від загальних до конкретних, включаючи відкладення прийняття рішень.

Для побудови діаграма IDEF4 інформаційної системи «Розклад аудиторних занять студентів» розглянемо основні класи та методи.

Система є базовим класом, який містить методи для обробки запитів та генерації звітів.

«Користувач» та «Адміністратор» є підкласами «Системи» з власними методами.

«Студент» та «Викладач» є підкласами «Користувача» з методами для взаємодії з розкладом.

«Менеджер розкладу» є підкласом «Адміністратора» з методами для управління розкладом.

«Заняття» представляє академічні події з методами для отримання та зміни інформації. Підкласи «Лекція» та «Практичне заняття» успадковують ці методи.

Діаграма, подана на рисунку 2.7 допомагає візуалізувати структуру інформаційної системи, її класи та методи, а також зв'язки між ними.

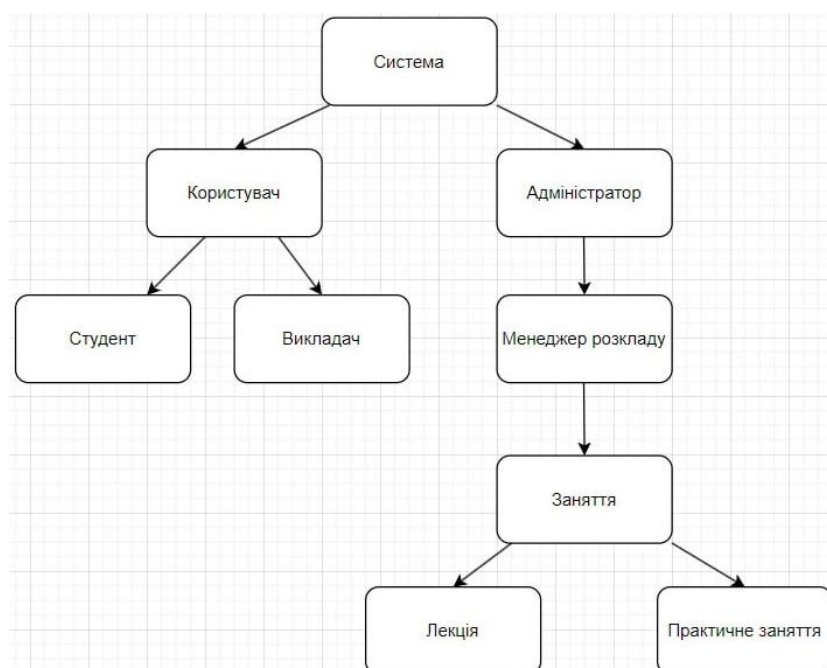


Рисунок 2.7 – Діаграма IDEF4 «Розклад аудиторних занять студентів»

*Клас:* «Система». *Методи:* обробляти запити, генерувати звіти.  
*Підкласи:* «Користувач», «Адміністратор».

*Клас:* «Користувач». *Методи:* переглядати розклад, фільтрувати заняття. *Підкласи:* «Студент», «Викладач».

*Клас:* «Адміністратор». *Методи:* додавати заняття, редагувати заняття, видаляти заняття. *Підкласи:* «Менеджер розкладу».

*Клас:* «Студент». *Методи:* переглядати розклад, підписатися на заняття.

*Клас:* «Викладач». *Методи:* переглядати розклад, оцінювати студентів

*Клас:* «Менеджер Розкладу». *Методи:* управляти заняттями.

*Клас:* «Заняття». *Методи:* отримати інформацію, змінити інформацію.  
*Підкласи:* «Лекція», «Практичне заняття».

*Клас:* «Лекція». *Методи:* отримати інформацію.

*Клас:* «Практичне заняття». *Методи:* отримати інформацію.

## 2.6 Діаграма розгорткування та компонентів

Діаграма розгорткування ілюструє фізичні зв'язки між програмними та апаратними компонентами системи. Вона відображає маршрути переміщення об'єктів і компонентів у розподіленій системі. Кожен вузол на діаграмі розгорткування представляє конкретний тип обчислювального пристрою, який у більшості випадків є частиною апаратного забезпечення. Діаграму розгорткування наведено на рисунку 2.8.

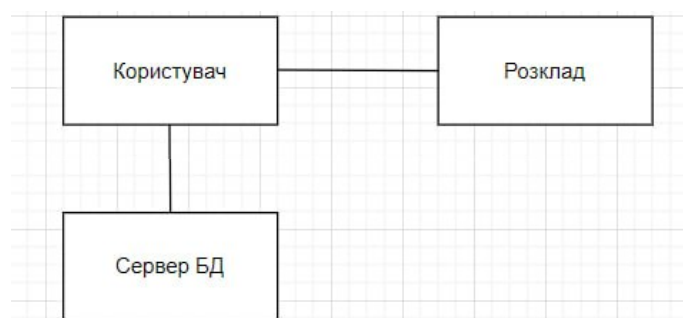


Рисунок 2.8 – Діаграма розгорткування

Діаграма компонентів демонструє структуру системи та взаємозв'язки між різними програмними компонентами. Це ефективний засіб для візуалізації архітектурного дизайну системи. Кожен компонент на діаграмі може представляти окрему функціональну одиницю програми, модуль, бібліотеку або сервіс. Зв'язки між компонентами показують, як вони взаємодіють між собою, наприклад, через виклики методів, обмін повідомленнями чи спільне використання даних. Діаграма компонентів допомагає розуміти логічну структуру системи та сприяє кращому розумінню взаємозв'язків між її складовими частинами. Діаграма компонентів інформаційної системи «Розклад аудиторних занять» включає в себе 7 компонентів: головна сторінка, обробка запиту користувача, БД викладач, БД предмет, БД факультет, БД семестр, БД група, БД студент, візуальний компонент з відображенням запиту. Наочно діаграма компонентів подана на рисунку 2.9.

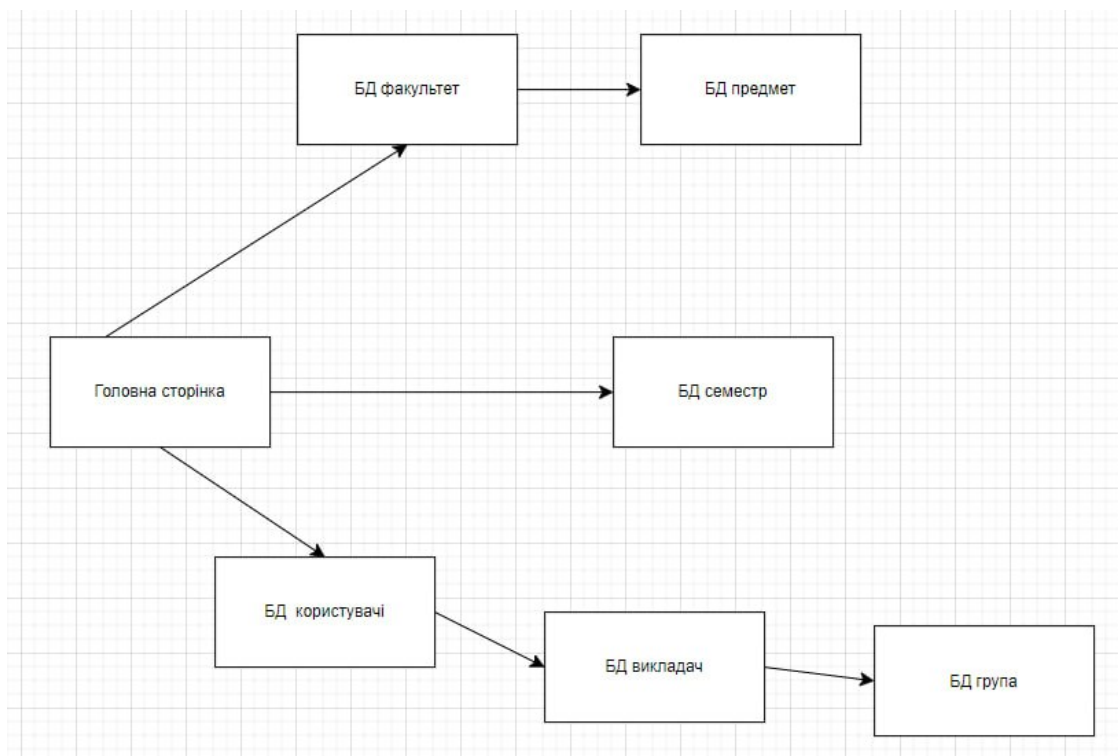


Рисунок 2.9 – Діаграма компонентів

Компонент Головна сторінка – є основним – надає змогу надати користувачам необхідну інформацію.

Компонент Обробка запиту користувача – служить для обробки запиту, отриманого від головної сторінки, через яку надходить інформація в базу даних груп, базу даних викладачів, базу даних студентів і на екран виводиться інформація, що цікавить користувача.

Компонент БД викладачів – містить в собі всю інформацію про зареєстрованих викладачів: ПІБ, науковий ступінь, предмети, що викладають, групи, в яких викладають.

Компонент БД предмет – містить в собі всю інформацію про дисципліну: назва, факультет, типи занять.

Компонент БД група – містить інформацію про студентів, що належать до цієї групи.

Компонент БД студент група – містить інформацію про студентів: ПІБ, факультет, курс.



## 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1 Розробка дизайну. Графічний інтерфейс

Розробка дизайну – це процес створення вигляду, структури та інтерфейсу продукту або системи, який може охоплювати багато аспектів, зокрема: вебдизайн, графічний дизайн, UX/UI дизайн, дизайн інтерфейсів користувача та інші [23].

Проаналізувавши основні вимоги до інформаційної системи, переходимо до реалізації [2, 4]. Важливим аспектом є графічний інтерфейс.

Графічний інтерфейс представляє собою ієрархією об'єктів, кожен з яких містить і впорядковує дочірні об'єкти. Об'єкти, в свою чергу, є елементами управління та іншими віджетами, наприклад, кнопки, текстові поля, через які користувач взаємодіє з ресурсом, залежно від своєї ролі [13].

Наочно ієрархію об'єктів ролі «адміністратор» наведено на рисунку 3.1.

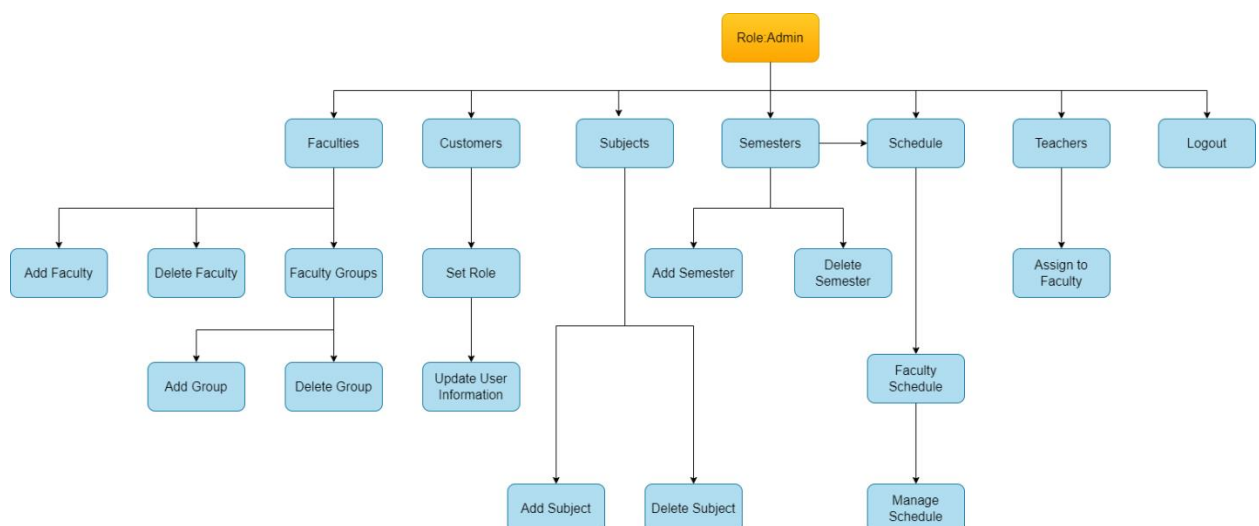


Рисунок 3.1 – Ієрархія об'єктів ролі «адміністратор»

Ієрархія об'єктів ролі «викладач» та «студент» представлені на рис. 3.2 та 3.3 відповідно.

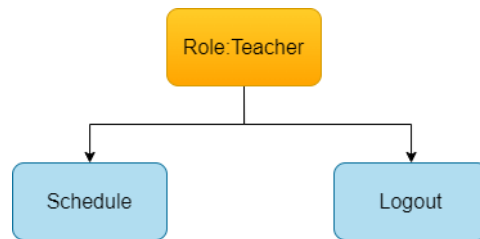


Рисунок 3.2 – Ієрархія об'єктів ролі «викладач»

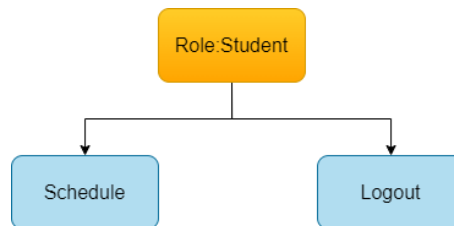


Рисунок 3.3 – Ієрархія об'єктів ролі «студент»

Інтерфейс інформаційної системи має бути зручним та інтуїтивно зрозумілим [9]. Реєстраційна форма для студентів та викладачів є однаковою та заповнюється менеджером розкладу (адміністратором). Форма реєстрації подана на рисунку 3.4.

Рисунок 3.4 – Форма авторизації

При переході на вебресурс відбувається відображення форми для авторизації, представлена на рисунку 3.5.

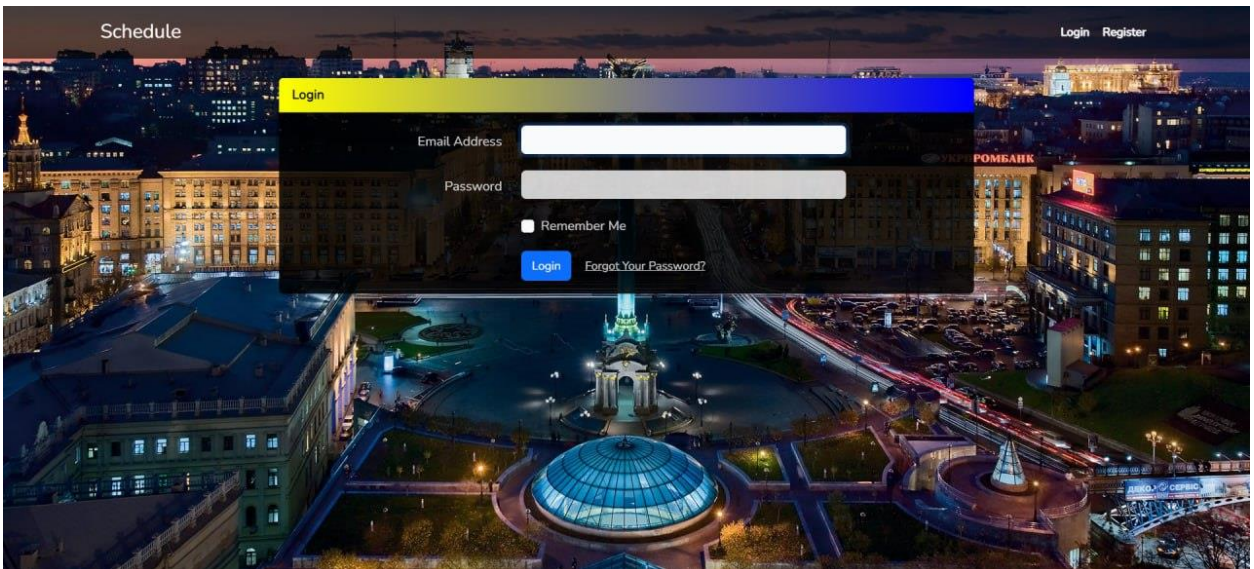


Рисунок 3.5 – Форма авторизації

Адміністратор додає, видаляє, редагує та оновлює інформацію про викладачів та студентів, вносити зміни в ролі.

Інтерфейс системи є простим, зрозумілим та розподілений за вкладками.

Всі студенти та викладачі в адміністративній панелі відображаються у вигляді списку (вкладка Customers).

Для зручності є можливість пошуку за прізвищем та роллю, зміст вкладки Customers наведено на рис. 3.6.

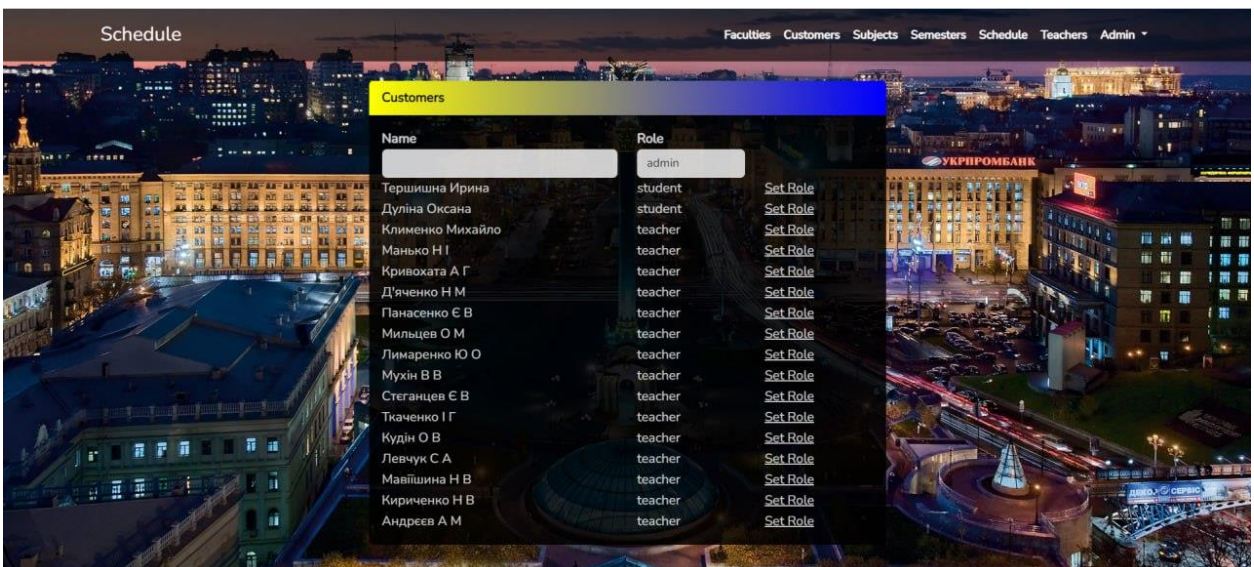


Рисунок 3.6 – Зміст вкладки Customers

Призначення та зміна ролей відбувається шляхом вибору ролі у відповідному полі, як наведено на рисунку 3.7.

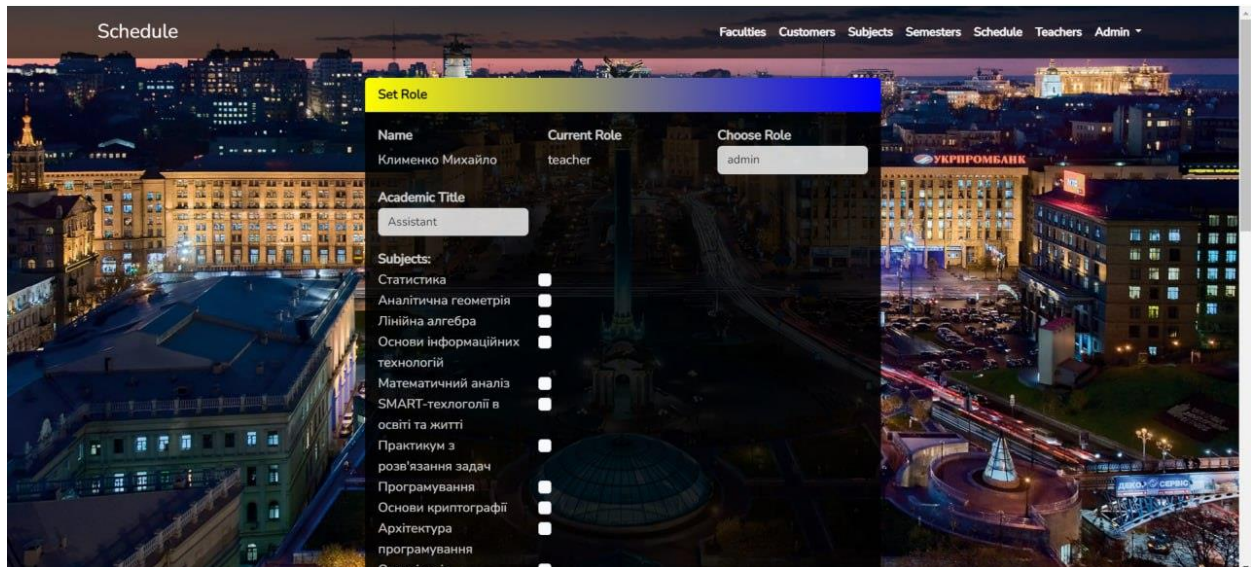


Рисунок 3.7 – Призначення та зміна ролей

Вкладка Факультет дає змогу додавати, редагувати та видаляти факультети. Її зміст та відображення наведені на рисунку 3.8.

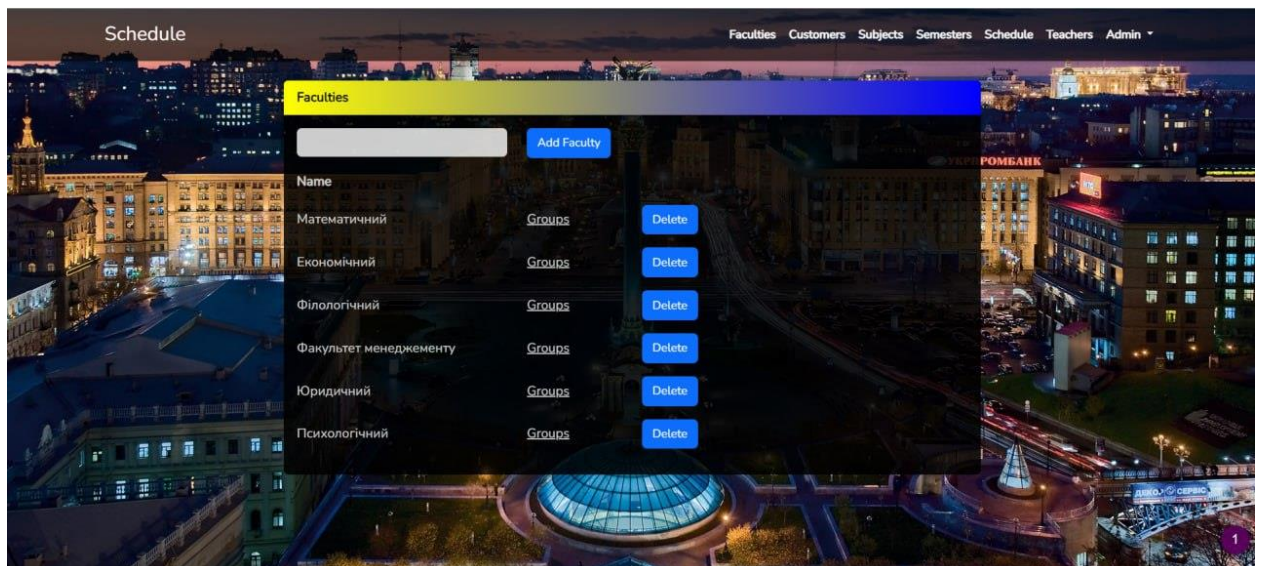


Рисунок 3.8 – Вкладка Faculties

Натиснувши назву факультету, побачимо список всіх викладачів, які включені до цього факультету, як наведено на рисунку 3.9.

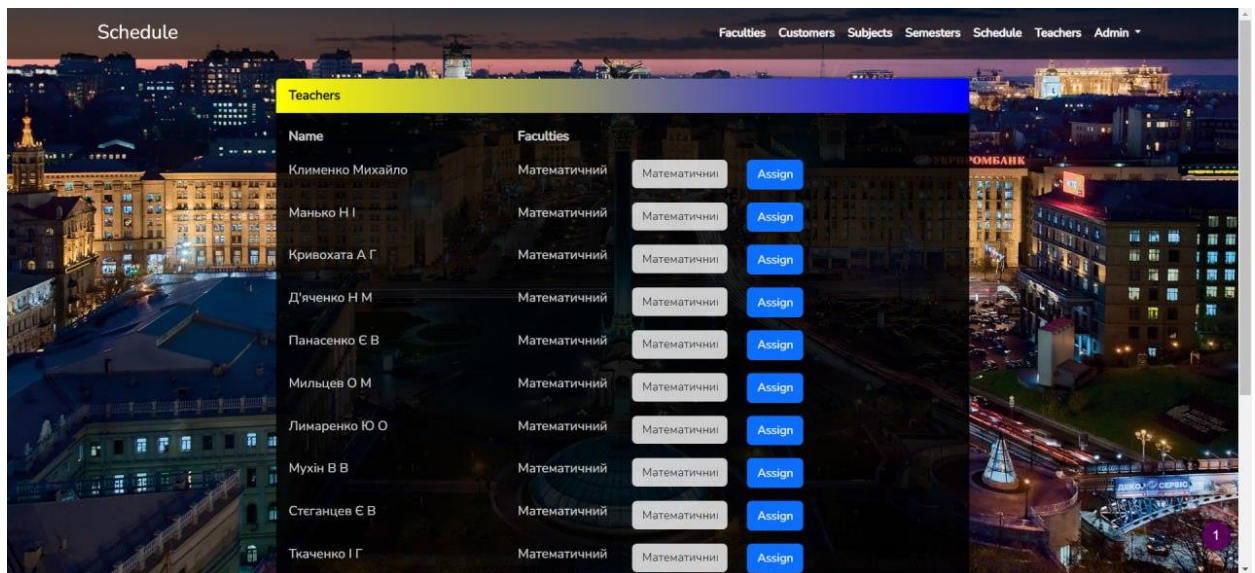


Рисунок 3.9 – Список викладачів факультету

Можливість створення, редагування та видалення дисциплін здійснюється менеджером розкладу (адміністратором) засобами вкладки Subjects, яка представлена на рисунку 3.10.

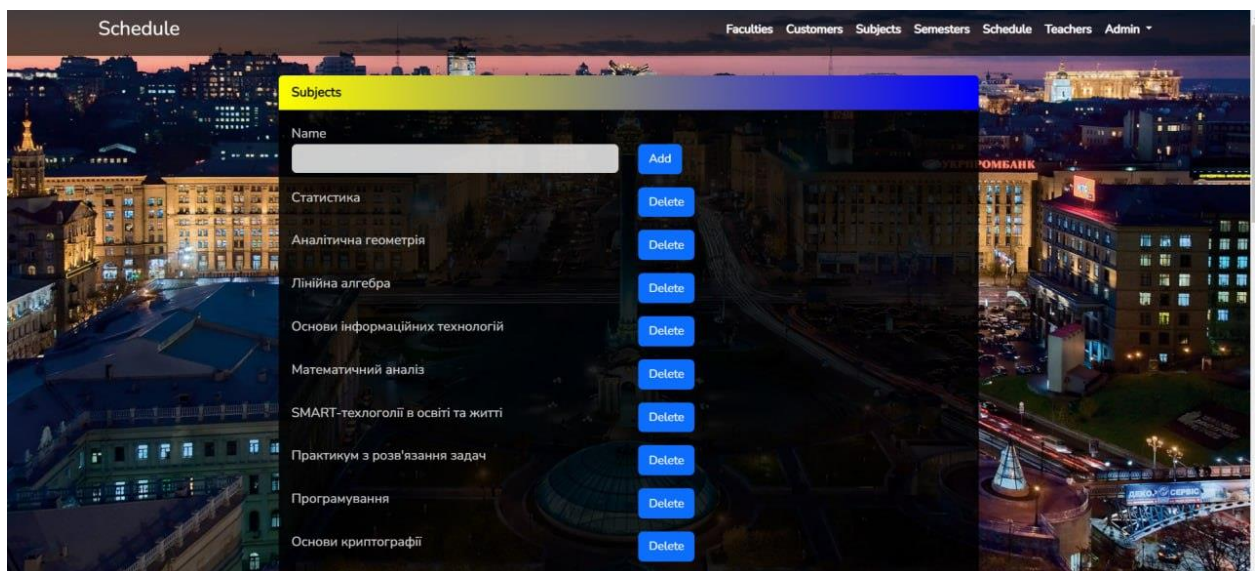


Рисунок 3.10 – Вкладка Subjects

Вкладка Semesters визначає керування семестрами: створення, редагування та фільтрація за семестром та навчальним роком (рис. 3.11)

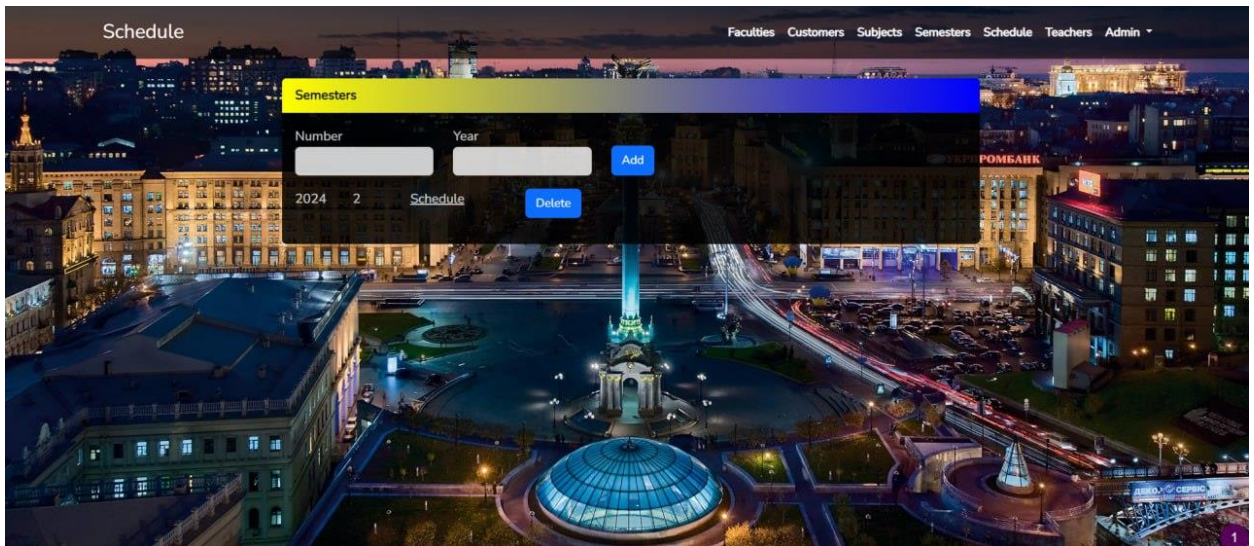


Рисунок 3.11 – Вкладка Semesters

### 3.2 Створення інформаційної системи

Для розробки та реалізації інформаційної моделі були використані: ER-модель, UML-діаграми, методологія реляційних баз даних тощо.

Процес створення включав:

- аналіз існуючих підходів до управління розкладом у навчальних закладах, їхні переваги та недоліки;
- формування вимог до інформаційної системи;
- проектування схеми даних з урахуванням відношень та зв'язків між таблицями;
- реалізацію баз даних;
- тестування інформаційної системи.

В процесі тестування були створені об'єкти, визначені інформаційною системою та представлені у розділі 2 цієї роботи, розподілені їхні ролі. Обрано дизайн в національному стилі. Результати є наступними: авторизувавшись, викладачі та студенти факультетів можуть ознайомитись з розкладом (за чисельником та знаменником) на тиждень або обрати перегляд розкладу на семестр.

### 3.3 Створення бази даних та робота з даними

Створення бази даних та робота з даними є ключовими аспектами розробки будь-якої інформаційної системи. Процес включає кілька основних етапів: аналіз вимог до даних та системи, проєктування схеми БД, вибір системи управління базами даних, створення моделі БД [8].

Аналіз вимог до даних, які потрібні для системи, визначили на початку роботи у розділі 1. Проаналізували можливість доступу та обробки даних.

Проєктування схеми бази даних включало створення моделі даних, яка містить визначення сутностей та їх атрибутів. Визначили зв'язки між сутностями та ключі для зв'язування сутностей. Наочно ці дані подано у розділі 2.

Система управління базами даних – MySQL.

Створення фізичної моделі бази даних включало SQL-скрипти для створення таблиць, визначення зв'язків, індексів та обмежень. Ці скрипти були виконані для створення фізичної бази даних. Приклад застосування SQL-скриптів подано на рисунку 3.12.

```
CREATE TABLE students (  
    student_id INT PRIMARY KEY,  
    name VARCHAR(100),  
    birth_date DATE  
);  
  
CREATE TABLE courses (  
    course_id INT PRIMARY KEY,  
    course_name VARCHAR(100)  
);  
  
CREATE TABLE schedule (  
    schedule_id INT PRIMARY KEY,  
    student_id INT,  
    course_id INT,  
    start_time DATETIME,  
    end_time DATETIME,  
    FOREIGN KEY (student_id) REFERENCES students(student_id),  
    FOREIGN KEY (course_id) REFERENCES courses(course_id)  
);
```

Рисунок 3.12 – Приклад застосування SQL-скриптів

## ВИСНОВКИ

У сучасному світі, де швидкість і доступність інформації є критичними для ефективного функціонування будь-якої системи, інформаційні технології стали невід'ємною частиною всіх аспектів життя. Це особливо актуально в сфері освіти, зокрема, вищої, де важливо забезпечити оптимальне використання часу та ресурсів для якісного навчання.

Предметною областю дослідження визначена організація освітнього навчального процесу, а саме, створення розкладу аудиторних занять студентів. При створенні розкладу працівники навчальної частини (менеджери розкладу) повинні враховувати інформацію про викладачів кафедри: анкетні дані, відомості про вчений ступень, адміністративна посада (науковий ступінь), яку займає, на яких факультетах та в яких аудиторіях відбуваються заняття. До того ж, викладачі кафедри мають забезпечити проведення занять по деяким предметам, відповідно до відведених годин (лекційні та практичні).

В процесі реалізації проєкту було розроблено та реалізовано інформаційну систему «Розклад аудиторних занять студентів» для формування розкладу занять, відповідно до навчального навантаження. Вона полягає в створенні зручного та функціонального інструменту, який допоможе автоматизувати процес планування та розподілу навчальних занять, забезпечуючи оптимальне використання ресурсів та задоволення потреб користувачів.

При розробці інформаційної системи був проведений аналіз предметної області, обрана архітектура і платформа реалізації, розроблена функціональна модель, спроектована архітектура системи, реалізована інформаційна система з використанням мови програмування PHP, здійснене тестування. Створена інформаційна система є простою та зручною, що дозволить її використання навіть тим, хто лише знайомиться зі світом



інформаційних технологій.

Відповідно до поставленої мети, розв'язані наступні задачі:

- проведено аналіз існуючих підходів до управління розкладом у навчальних закладах, переваги та недоліки цих підходів;
- проаналізовано переваги та недоліки існуючих інформаційних систем;
- сформульовано вимоги до інформаційної системи;
- спроєктовано схему даних з урахуванням відношень та зв'язків між таблицями;
- реалізовано бази даних;
- протестовано інформаційну систему, результати реалізації наведені у розділі 3.

Результати роботи можуть бути використані для подальшого застосування в освітній сфері при створенні розкладу аудиторних занять студентів, а також автоматизованого розподілу навчального навантаження викладачів.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Береза А. М. Основи створення інформаційних систем : навч. посібник, 2 видання, перероблене і доповнене. Київ : КНЕУ, 2001. 624 с.
2. Голуб Б. Л., Ветрова Д. В., Пронішина К. О. Програмна система формування розкладу занять у закладі вищої освіти. *Математичні машини і системи*. 2019. № 4. С. 100–109.
3. Діаграма ER: Модель діаграми взаємовідносин суб'єктів – Приклад СУБД. URL : <https://uk.csstricks.net/8224622-er-diagram-entity-relationship-diagram-model-dbms-example> (дата звернення: 10.04.2024).
4. Димова Г. О. Розробка моделі складання розкладу занять методом еволюційного пошуку. *Таврійський науковий вісник. Серія: Технічні науки*. 2022. № 2. С. 3–9. URL: <https://doi.org/10.32851/tnv-tech.2022.2.1> (дата звернення: 10.02.2024)
5. Документація Laravel. URL: <https://laravel.com/docs/11.x/readme> (дата звернення: 18.02.2024).
6. Жук Ю. О. Комп'ютерно-орієнтовані засоби навчальної діяльності: проблеми створення та впровадження. *Науковий вісник Ізмаїльського держ. пед. ін-ту*. 2004. Вип. 16. С. 11–15.
7. Закон України «Про Національну програму інформатизації» від 01.12.2022 № 2807-IX. URL: <https://zakon.rada.gov.ua/laws/show/74/98-%D0%B2%D1%80#Text> (дата звернення: 02.03.2024).
8. Коваленко В. В., Гуменюк В. О. Аналіз найактуальніших серверних систем управління базами даних. URL: [https://vlp.com.ua/files/14\\_22.pdf](https://vlp.com.ua/files/14_22.pdf) (дата звернення: 20.03.2024).
9. Кемарський М. О. Методи та програмні засоби складання розкладів занять у вищих навчальних закладах : магістерська дис. : 121 Інженерія програмного забезпечення. Київ, 2024. 147 с.
10. Макаренко Л. Л. Інформатизація освіти як пріоритетний напрям

модернізації освіти в умовах інформаційного суспільства. *Науковий часопис НПУ імені М. Драгоманова. Серія 5. Педагогічні науки і перспективи.* 2013. № 43. С. 118. URL: <http://enpuir.npu.edu.ua/bitstream/handle/123456789/18041/Makarenko.pdf?sequence=3&isAllowed=y> (дата звернення: 27.03.2024).

11. Марченко А. В. Проектування інформаційних систем. URL: [https://elearning.sumdu.edu.ua/free\\_content/lectured:de1c9452f2a161439391116eef364dd8ce4d8e5e/20160125164438/content-20160125164438.pdf](https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391116eef364dd8ce4d8e5e/20160125164438/content-20160125164438.pdf) (дата звернення: 22.03.2024)
12. Мінухін С. В., Беседовський О. М., Знахур С. В. Методи і моделі проектування на основі сучасних CASEзасобів. Харків : Вид. ХНЕУ, 2008. 272 с.
13. Пасічник В. В., Резниченко В. А. Організація баз даних та знань. Київ : Видавнича група BHV, 2006. 384 с.
14. Математичні основи теорії телекомунікаційних систем / Поповський В. В. та ін.; за загал. ред. В. В. Поповського. Харків : ТОВ «Компанія СМІТ», 2006. 564 с.
15. Портфоліо на тему Laravel. URL: <https://webnauts.pro/uk/portfolio/laravel/> (дата звернення: 14.03.2024).
16. Коваленко О. С., Добровська Л. М. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій). : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки». Київ : КПІ ім. Ігоря Сікорського, 2020. 192 с.
17. Проектування програмного забезпечення засобами UML. URL: [http://mmsa.kpi.ua/sites/default/files/disciplines/%D0%A0%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D1%96%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC/didkovska\\_m\\_v\\_testing\\_lecture\\_6.pdf](http://mmsa.kpi.ua/sites/default/files/disciplines/%D0%A0%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D1%96%20%D1%82%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC/didkovska_m_v_testing_lecture_6.pdf) (дата звернення: 14.03.2024).

18. Реінжиніринг бізнес-процесів. URL: <https://library.if.ua/book/28/1899.html> (дата звернення: 14.03.2024).
19. Розробка сайтів на Laravel URL: <https://redchameleon.com.ua/ua/individualnaya-razrabotka/razrabotka-saytov-na-laravel/> (дата звернення: 14.02.2024).
20. Сайт Міністерства освіти і науки України. URL: <http://mon.gov.ua> (дата звернення 25.02.2024).
21. Система вибору NoSQL сховища даних на основі бажаних характеристик інформаційної системи. URL: <http://naukam.triada.in.ua/index.php/konferentsiji/49-dev-yatnadtsyata-vseukrajinska-praktichno-piznavalna-internet-konferentsiya/438-sistema-viboru-nosql-skhovishcha-danikh-na-osnovi-bazhanikh-kharakteristik-informatsijnoi-sistemi> (дата звернення: 22.03.2024).
22. Список систем керування базами даних. URL : [https://www.wikiwand.com/uk/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA\\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC\\_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F\\_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85](https://www.wikiwand.com/uk/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) (дата звернення: 22.03.2024).
23. Столярчук О. В., Шувар Р. Я., Продивус А. М. Веб інтерфейси доступу до баз даних. Електроніка та інформаційні технології : зб. наук. пр. Вип. 1 / Львівський нац. ун-т ім. І. Франка. Львів : Видавництво ЛНУ ім. І. Франка, 2011. 262 с.
24. Схема використання елементів. URL : <https://dou.ua/forums/topic/40575/> (дата звернення: 22.03.2024).
25. Laracasts. URL: <https://laracasts.com/> (дата звернення: 18.02.2024).
26. PHP фреймворк Laravel та його можливості. URL: <https://www.youtube.com/watch?v=oKRC8b2ZEn8> (дата звернення: 28.02.2024).

# ДОДАТОК А

## Програмна реалізація бази даних

Чисельник	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
08:00-09:20	Статистика 6.1269 11	Теорія ймовірності 6.1212-1 15	Аналітична геометрія 6.1253 38		Теорія ймовірності 6.1212-1 15	
09:30-10:50	Аналітична геометрія 6.1253 12	Статистика 6.1051 11	Алгебраїчні методи геометрії 6.1052 38		Статистика 6.1051 11	
11:10-12:30	Алгебраїчні методи геометрії 6.1052 24		Статистика 6.1269 11			
12:40-14:00						
14:30-15:50						
16:00-17:20						
17:30-18:50						
19:00-20:20						

Рисунок А.1 – Розклад аудиторних занять викладача за чисельником

Знаменник	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
08:00-09:20			Алгебраїчні методи геометрії 6.1052 38			
09:30-10:50			Статистика 6.1269 11			
11:10-12:30	Статистика 6.1269, 6.1052 11		Статистика 6.1269 11			
12:40-14:00						
14:30-15:50						
16:00-17:20						
17:30-18:50						
19:00-20:20						

Рисунок А.2 – Розклад аудиторних занять викладача за знаменником

Чисельник	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
08:00-09:20	Статистика Клименко Михайло 11					
09:30-10:50	Комп'ютерна графіка Мухін В В 12	Нейронні системи і мережі Кудін О В 24			Основи криптографії Кириченко Н В 36	
11:10-12:30			Статистика Клименко Михайло 11		Безпека програм та даних Стеганцев Є В 28	
12:40-14:00	Комп'ютерна графіка Мухін В В 12	Основи криптографії Кириченко Н В 37	Нейронні системи і мережі Кудін О В 25			
14:30-15:50		Безпека програм та даних Стеганцев Є В 29				
16:00-17:20						
17:30-18:50						
19:00-20:20						

Рисунок А.3 – Розклад аудиторних занять студентів

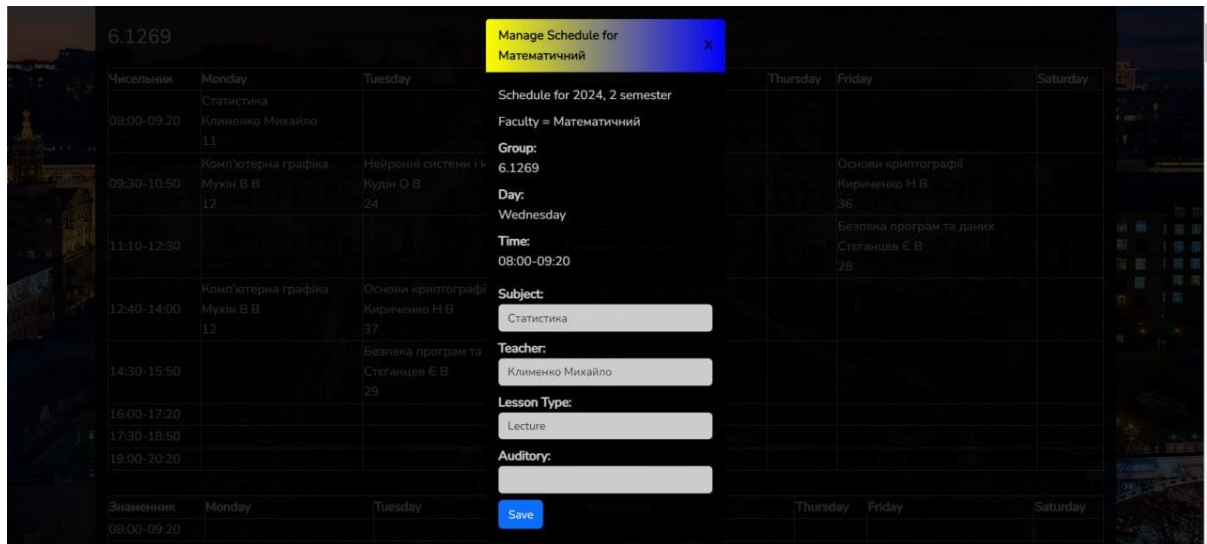


Рисунок А.4 – Детальна інформація про заняття

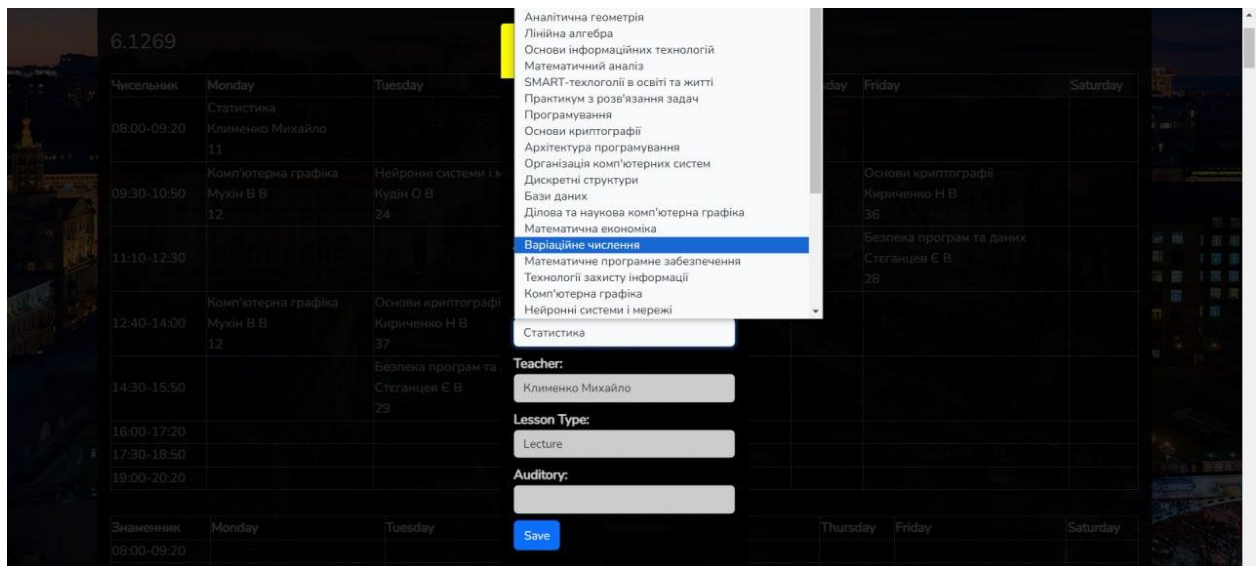


Рисунок А.5 – Вибір дисциплін та формування розкладу

## ДОДАТОК Б

### Програмний код

```

@extends('layouts.app')

@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-12">
                <div class="manage-schedule hidden">
                    <div class="card">
                        <div class="card-header">
                            <div class="text-header">Manage Schedule for
                            {{ $facultyName }}</div>
                            <div class="close">X</div>
                        </div>
                        <div class="card-body p-3">
                            {{ html()->form('POST', '/schedule/save')->open() }}
                            <div class="row">
                                <span>Schedule for {{ $year }}, {{ $number }}
                                semester</span>
                            </div>
                            {{ html()->hidden('semester-id', $semesterId) }}
                            <div class="row my-2">
                                <span>Faculty = {{ $facultyName }}</span>
                            </div>
                            {{ html()->hidden('faculty-id', $facultyId) }}
                            <div class="row my-2">
                                <label><b>Group:</b></label>

```

```

        <div class="group-number-text"></div>
    </div>
    <input type="hidden" name="group-id" id="group-id"
class="form-control">
    <div class="row my-2">
        <label><b>Day:</b></label>
        <div class="selected-date-number"></div>
    </div>
    {{ html()->hidden('day-number') }}
    <div class="row my-2">
        <label><b>Time:</b></label>
        <div class="selected-time"></div>
    </div>
    {{ html()->hidden('lesson-number') }}
    <div class="row my-2">
        <div class="pair-text"></div>
    </div>
    {{ html()->hidden('pair') }}
    <div class="row form-group subject-select my-2">
        <label for="subject-id"><b>Subject:</b></label>
        <div>
            <select name="subject-id" id="subject-id" class="form-
control">
                @foreach($subjects as $id => $subject)
                    <option value="{{ $id }}">{{ $subject }}</option>
                @endforeach
            </select>
        </div>
    </div>
    <div class="row my-2 form-group teacher-select">

```



```

<label for="teacher-id"><b>Teacher:</b></label>
<div>
  <select name="teacher-id" id="teacher-id" class="form-
control">
    @foreach($teachers as $id => $teacher)
      <option value="{{ $id }}">{{ $teacher }}</option>
    @endforeach
  </select>
</div>
</div>
<div class="row form-group my-2">
  <label for="lesson-type"><b>Lesson Type:</b></label>
  <div>
    <select name="lesson-type" id="lesson-type" class="form-
control">
      @foreach($types as $id => $type)
        <option value="{{ $id }}">{{ $type }}</option>
      @endforeach
    </select>
  </div>
</div>
<div class="row my-2 form-group auditory-select">
  <label for="auditory-number"><b>Auditory:</b></label>
  <div>
    <input type="text" name="auditory-number" id="auditory-
number" class="form-control">
  </div>
</div>
<div class="row form-group">
  <div>

```

```

        <button      type="submit"      class="btn      btn-
primary">Save</button>
    </div>
</div>
    {{ html()->form()->close() }}
</div>
</div>
</div>
<div class="card">
    <div class="card-header">Schedule for {{ $year }} {{ $number }}
Faculty {{ $facultyName }}</div>
    <div class="card-body">
        @foreach($groupsData as $group)
            <div      class="faculty-data      my-3"      data-semester-
id="{{ $semesterId }}" data-group-id="{{ $group->id }}">
                <h3>{{ $group->name }}</h3>
                <div class="schedule row justify-content-between">
                    @foreach([0, 1] as $pair)
                        <div class="col-12      my-3      schedule-container      pair-
{{ $pair }}" data-pair="{{ $pair }}">
                            <table>
                                <tr>
                                    <th>{{ $pair ?'Знаменник': 'Чисельник' }}</th>
                                    @foreach($days as $number => $day)
                                        <th class="day-{{ $number }}">
                                            {{ $day }}
                                        </th>
                                    @endforeach
                                </tr>
                                @foreach($lessons as $lessonNumber => $lesson)

```

```

        <tr          class="lesson"          data-lesson-
number="{{ $lessonNumber }}">
            <td class="time-label">
                {{ $lesson }}
            </td>
            @foreach($days as $dayNumber => $day)
                <td          class="schedule-cell"          data-day-
number="{{ $dayNumber }}">

<div>{{          ScheduleHelper::getSubject($group->schedules,          $dayNumber,
$lessonNumber, $pair, $group->id) }}</div>

<div>{{          ScheduleHelper::getTeacher($group->schedules,          $dayNumber,
$lessonNumber, $pair, $group->id) }}</div>

<div>{{          ScheduleHelper::getAuditory($group->schedules,          $dayNumber,
$lessonNumber, $pair, $group->id) }}</div>
                </td>
            @endforeach
        </tr>
    @endforeach
</table>
</div>
    @endforeach
</div>
</div>
</div>
</div>

```

```

    </div>
</div>
@endsection

@extends('layouts.app')
@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">Schedule for {{ $year }}
                    {{ $number }}</div>
                    <div class="card-body">
                        @foreach($faculties as $faculty)
                            <div class="row">
                                <div class="col-4">
                                    <h3>{{ $faculty->name }}</h3>
                                </div>
                                <div class="col-4">
                                    <a href="/schedule/{{ $semester_id }}/faculty/{{ $faculty-
                                    >id }}">Schedule</a>
                                </div>
                            </div>
                        @endforeach
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection

```

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Laravel</title>

    <!-- Fonts -->
    <link rel="preconnect" href="https://fonts.bunny.net">
    <link
href="https://fonts.bunny.net/css?family=figtree:400,600&display=swap"
rel="stylesheet" />
```