

Одеський національний університет імені І.І.Мечникова

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерної алгебри та дискретної математики

(повна назва кафедри (предметної, циклової комісії))

## Дипломна робота

на здобуття ступеня вищої освіти «бакалавр»

(освітньо-кваліфікаційний рівень)

на тему: Розробка автоматизованої інформаційної системи формування розкладу занять для освітніх установ

(назва українською)

Development the automated information system to draw up the educational institutions schedule

(назва англійською)

Виконала: студентка денної форми навчання

спеціальності 123 Комп'ютерна інженерія

(шифр і назва напрямку підготовки, спеціальності)

Виноградов Дмитро Сергійович

(прізвище, ім'я, по-батькові)

Керівник д-р. фіз.-мат. наук проф. С. П. Варбанець

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

Рецензент канд. техн. наук, доц. Якімова Н.А.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ 12 від «09» 06 2023 р.

Завідувач кафедри

(підпис)

П. Д. Варбанець

(прізвище, ініціали)

Захищено на засіданні ЕК №     

протокол № 12 від «09» 06 2023 р.

Оцінка      /      /     

(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

(підпис)

А. А. Кобозєва

(прізвище, ініціали)

Одеса - 2023

## АНОТАЦІЯ

**Об'єкт:** Навчальні заклади, освітні установи.

**Предмет:** Автоматизована інформаційна система формування розкладу занять.

**Мета:** Розробка автоматизованої інформаційної системи формування розкладу занять для освітніх установ Також є полегшення процесу складання розкладу занять, зниження трудомісткості та покращення ефективності цього процесу.

В роботі розглядаються головні проблеми, пов'язані зі складанням багаторівневого розкладу учбових занять, такі як велика кількість обмежень, розбіжності у вимогах студентів та викладачів, проблеми сумісності часу, а також непередбачувані зміни в розкладі.

Щоб вирішити ці проблеми, розроблена автоматизована система, яка базується на алгоритмах генетичного програмування та використовує one+one еволюційну стратегію. Система враховує обмеження, задані в програмі, генерує розклад та здійснює оптимізацію розкладу, забезпечуючи максимальну задоволеність студентів та викладачів.

Моя дипломна робота включає опис процесу розробки інформаційної системи, впровадження та тестування її функціональності. Ефективність рішення оцінюється за виконанням заданих обмежень і критеріїв. Тестування алгоритму підтверджують, що всі головні обмеження дотримуються, а додаткові критерії максимально оптимізовані аби задовольнити попереднім умовам.

В результаті виконання дипломного проекту отримана готова ІС формування розкладу занять з дотриманням стандартів програмування та патернів проектування.

## ABSTRACTS

**Object:** Educational institutions, educational institutions.

**Subject:** An automated information system for the formation of class schedules.

**Objective:** To develop an automated information system for creating class schedules for educational institutions. It also aims to facilitate the process of creating class schedules, reduce labor intensity and improve the efficiency of this process.

The paper considers the main problems associated with the preparation of a multi-level class schedule, such as a large number of constraints, differences in the requirements of students and teachers, time compatibility problems, and unpredictable changes in the schedule.

To solve these problems, we developed an automated system based on genetic programming algorithms and using a one-to-one evolutionary strategy. The system takes into account the constraints set in the program, generates a schedule, and optimizes the schedule, ensuring maximum student and teacher satisfaction.

My thesis includes a description of the process of developing an information system, implementing and testing its functionality. The effectiveness of the solution is evaluated by fulfilling the set constraints and criteria. Testing of the algorithm confirms that all the main constraints are met, and additional criteria are optimized to meet the preceding conditions.

As a result of the graduation project, a ready-made IS for generating class schedules in compliance with programming standards and design patterns was obtained.

**ЗМІСТ**

|                                                      |    |
|------------------------------------------------------|----|
| ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ..... | 5  |
| ВСТУП.....                                           | 6  |
| 1 ПОСТАНОВКА ЗАДАЧІ.....                             | 8  |
| 1.1 Головні проблеми.....                            | 9  |
| 1.2 Основні обмеження.....                           | 9  |
| 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....           | 11 |
| 3 ІНФОРМАЦІЙНА МОДЕЛЬ ПРЕДМЕТНОЇ ОБЛАСТІ .....       | 15 |
| 4 МОДЕЛЬ ПРОГРАМНОГО ДОДАТКУ .....                   | 16 |
| 5 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....                | 17 |
| 6 АЛГОРИТМ ДЛЯ РОЗВ’ЯЗАННЯ ПОСТАВЛЕНИХ ЗАДАЧ.....    | 19 |
| 6.1 Огляд алгоритма роботи ІС.....                   | 19 |
| 6.2 Рішення та еволюційна стратегія.....             | 20 |
| 7 ПРОГРАМНА РЕАЛІЗАЦІЯ.....                          | 21 |
| 8 ІНСТРУКЦІЯ КОРИСТУВАЧА.....                        | 29 |
| ВИСНОВКИ .....                                       | 34 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                     | 35 |

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ**

JSON – JavaScript Objective Notation.

API (Application Programming Interface) – програмний інтерфейс додатку.

JSON – JavaScript Objective Notation.

JWT – JSON Web Token.

REST (Representational State Transfer) – передача “самоописуваного” стану.

UX/UI (User experience/user interface) – взаємодія користувача та інтерфейса.

ІС – інформаційна система.

ПЗ – програмне забезпечення.

## ВСТУП

Протягом існування учбових закладів і особливо у двадцять першому сторіччі, де освіта відіграє ключову роль у формуванні успішної та конкурентоспроможної суспільної системи, ефективне управління ресурсами освітнього процесу стає надважливою необхідністю.

Одним з головних аспектів управління є складання розкладу занять, яке є трудоемким, складним та часо-витратним процесом. Неправильно складений розклад може призвести до перевантаження викладачів або студентів, конфліктів у графіку викладачів та низької ефективності навчання в цілому.

Метою даної дипломної роботи є розробка автоматизованої системи генерації розкладу занять для освітніх установ, яка допоможе полегшити процес складання розкладу, знизити трудомісткість та покращити його якість.

При цьому всі обмеження та оптимізаційні питання достатньою мірою є вирішеними. ІС базується на сучасних методах оптимізації та інтелектуальних алгоритмах з використанням еволюційної стратегії, що дозволять уникнути традиційних проблем та забезпечити оптимальне розподілення ресурсів.

Для досягнення зазначеної мети необхідно вирішити такі завдання:

- 1) виконати аналіз предметної області;
- 2) розробити зручне програмне забезпечення з зрозумілим інтерфейсом, що дозволяє взаємодіяти з розкладом;
- 3) визначити і сформулювати вимоги та обмеження до створеної ІС;
- 4) вибрати архітектуру і шаблон проектування створюваної ІС;
- 5) спроектувати кодову базу;
- 6) вибрати технології та засоби реалізації ІС;
- 7) з урахуванням вибраних засобів створити клієнтську програму, яка дасть можливість користувачу ефективно маніпулювати даними предметної області відповідно до їх повноважень.

Моя дипломна робота має важливе значення, оскільки вона сприятиме

покращенню якості навчання та забезпеченню ефективного використання ресурсів освітнього процесу. Фінальні результати роботи можуть бути корисними для університетів, коледжів та інших освітніх закладів, які мають проблеми складання розкладу занять.

Користувачі мають змогу користуватися автоматизованою системою для швидкого та точного складання розкладу, що відповідатиме їх потребам та вимогам. Важливо зазначити, що використання інтелектуальних алгоритмів дозволить системі адаптуватись до змінних умов та забезпечувати оптимальне використання доступних ресурсів.

В подальшому в дипломній роботі будуть розглянуті основні проблеми, пов'язані зі складанням розкладу занять, описана методологія розробки автоматизованої системи та проведений аналіз її функціональності. Зафіксовані та отримані результати і висновки будуть підкреслювати важливість та ефективність розробленої ІС.

## 1 ПОСТАНОВКА ЗАДАЧІ

З метою полегшення процесу складання розкладу занять для закладів освіти та покращення його ефективності, до основних задач поставлено наступні завдання дипломної роботи:

1) Розробка алгоритмів та методів автоматичної генерації розкладу занять використовуючи еволюційну стратегію [1]. Система повинна бути здатною враховувати велику кількість обмежень, таких як доступність аудиторій, ресурси викладачів, пріоритети предметів та перевіряти їх сумісність;

2) Розробка інтуїтивно зрозумілого та зручного інтерфейсу користувача для введення вхідних даних, встановлення обмежень та отримання розкладу. Це дозволить адміністраторам освітніх закладів та викладачам легко взаємодіяти з системою та вносити необхідні зміни;

3) Забезпечення оптимізації розкладу занять. Система має пропонувати найбільш оптимальні варіанти розкладу, враховуючи вимоги та пріоритети всіх зацікавлених сторін, таких як студенти та викладачі;

4) Врахування гнучкості та зміни в розкладі. Система повинна бути здатна адаптуватися до змін у вимогах, непередбачуваних подіях та змінах в наявних ресурсах. Вона повинна швидко регенерувати розклад та пропонувати оптимальні варіанти, що враховують нові обмеження та умови;

5) Проведення ефективного тестування системи та оцінка її продуктивності. Вимоги до ІС повинні бути перевірені шляхом виконання тестових сценаріїв та збором статистики щодо часу генерації розкладу та якості отриманих результатів.

Вирішення цих завдань дозволить створити автоматизовану систему генерації розкладу занять, яка значно спростить процес розподілу ресурсів та покращить якість освітнього процесу в закладах освіти.

## 1.1 Головні проблеми

Кожне заняття на факультеті представлено у вигляді блоку тривалості від однієї до чотирьох годин. Для проведення кожного заняття потрібні викладач, аудиторія, час початку, тривалість і групи, які беруть участь у занятті. Також передбачено, які групи відвідують який клас, і всі класи мають однаковий розмір (тобто кожна група може вміститися в будь-який клас). Навчання проводиться на факультеті з 8:00 до 21:00 кожного робочого дня.

Основні проблеми при розробці ІС генерації розкладу занять для закладів освіти включають:

- 1) Велика кількість обмежень;
- 2) Розбіжності у вимогах;
- 3) Непередбачувані зміни;
- 4) Великий обсяг даних.

## 1.2 Основні обмеження

ІС має певні обмеження, що впливають на кінцевий результат який отримує користувач. Обмеження є логічно виваженими, деякі з них є обов'язковими, а деякі умовними та ситуативними. Повний перелік обмежень включає в себе:

- 1) Ресурси не можуть бути одночасно використані:
  - а) Жодна аудиторія не може бути зайнята двома заняттями одночасно;
  - б) Жоден викладач не може проводити два уроки одночасно;
  - с) Жодна група не може брати участь у двох заняттях одночасно.

Примітка: тут "одночасно" означає не лише однаковий час початку, а й враховує тривалість занять. Якщо ресурс зайнятий у момент часу  $T1$  і заняття триває  $t1$ , то ресурс можна використати знову тільки у момент часу  $T2 = T1 + t1$ .

- 2) Заняття повинні проводитися лише у дозволених аудиторіях;
- 3) Якщо предмет має кілька форм навчання, пріоритетним є порядок проведення таких:

- a) лекції;
- b) практичні заняття;
- c) лабораторні заняття.

Обмеження 1 і 2 повинні бути дотримані, тоді як обмеження 3 є “ситуативним” і його можна порушити.

Додаткові критерії для оцінки рішення (використовуються також як функція вартості):

- 1) Дотримання жорстких обмежень (1 і 2);
- 2) Дотримання слабких обмежень (3);
- 3) Мінімізація загального часу “простою” для кожної групи (мінімізація перерв між заняттями);
- 4) Мінімізація загального часу “простою” для кожного викладача (мінімізація перерв між заняттями);
- 5) Забезпечення однієї години на навчальний тиждень, коли ніхто не має занять.

## 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

При проектуванні архітектури інформаційної системи було обрано триланкову клієнт-серверну архітектуру та архітектурний стиль REST [2].

REST (Representational State Transfer) - це архітектурний стиль, що застосовується в системах веб-служб і використовує HTTP [3] протокол для взаємодії між системами. REST характеризується простотою, масштабованістю та гнучкістю, він відповідає принципам декларативного доступу до ресурсів та статичності клієнт-серверної архітектури.

REST API складається з декількох основних компонентів: ресурсів, операцій та методів HTTP. Ресурси представлені у вигляді URL-адрес, а операції відповідають за створення, отримання, оновлення та видалення (CRUD). Методи HTTP, такі як GET, POST, PUT, DELETE та інші, використовуються для взаємодії з ресурсами (див. рисунок 2.1).

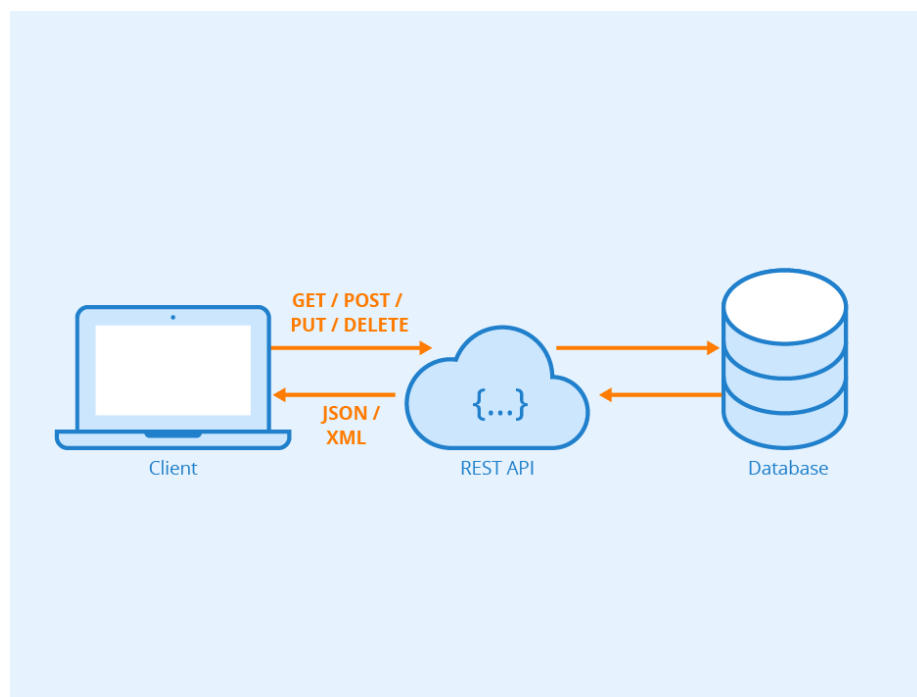


Рисунок 2.1 – Схема архітектури REST API [4]

В загальному випадку, REST API працює таким чином: клієнт надсилає запит до сервера за допомогою певного методу HTTP, сервер обробляє запит і надсилає відповідь у форматі JSON або XML документу (див. рисунок 2.2). У відповіді міститься статус виконання операції та, при необхідності, дані.

|                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                  |               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <h2>XML</h2>                                                                                                                                                                                                                                                                                                                                                                                | <p>vs.</p>                                                                                                                                                                                                                                       | <h2>JSON</h2> |
| <pre> 1 &lt;?xml version="1.0" encoding="UTF-8"?&gt; 2 &lt;endereco&gt; 3   &lt;cep&gt;31270901&lt;/cep&gt; 4   &lt;city&gt;Belo Horizonte&lt;/city&gt; 5   &lt;neighborhood&gt;Pampulha&lt;/neighborhood&gt; 6   &lt;service&gt;correios&lt;/service&gt; 7   &lt;state&gt;MG&lt;/state&gt; 8   &lt;street&gt;Av. Presidente Antônio Carlos, 6627&lt;/street&gt; 9 &lt;/endereco&gt; </pre> | <pre> 1 { 2   "endereco": { 3     "cep": "31270901", 4     "city": "Belo Horizonte", 5     "neighborhood": "Pampulha", 6     "service": "correios", 7     "state": "MG", 8     "street": "Av. Presidente Antônio Carlos, 6627" 9   } 10 } </pre> |               |

Рисунок 2.2 – Формат XML та JSON файла [5]

REST відрізняється від інших технологій, наприклад SOAP, тим, що у нього немає жорстко визначеної схеми взаємодії, але він має певні принципи та правила, які необхідно дотримуватися.

У цілому, REST відомий як ефективний та простий спосіб взаємодії між системами в Інтернеті і вважається практичним підходом для розробки API.

REST API може використовуватися для доступу до даних та функціональності веб-сервісу, виконання операцій над даними та передачі даних між різними системами.

Наприклад, компанія може розробити REST API для свого веб-сервісу електронної комерції, що дозволить іншим компаніям використовувати дані та функціональність цього сервісу у своїх додатках.

Ресурси, які можна отримати через REST API, представлені у вигляді URL. Користувач може здійснювати різні дії, такі як отримання даних, додавання, оновлення та видалення даних, використовуючи HTTP запити, наприклад GET, POST, PUT та DELETE.

REST API може бути реалізований у багатьох мовах програмування, таких як Java, Python, Ruby та PHP, і може використовуватися з різними платформами, включаючи веб-додатки, мобільні додатки та десктопні додатки.

REST API має значні переваги у гнучкості та простоті використання. Воно не залежить від конкретної платформи або мови програмування, тому його можуть використовувати різні системи та додатки.

Проте, оскільки REST API використовує протокол HTTP для передачі даних, це може створювати обмеження для додатків, які потребують високої пропускної здатності та низької затримки.

У цілому, REST API є широко використовуваним та ефективним способом надання доступу до веб-сервісів та даних, який може бути застосований в різних сферах.

При проектуванні додатку на основі архітектури REST, також важливо обрати відповідний шаблон проектування. Один з таких шаблонів - Model-View-Controller (MVC) [6], який можна використовувати у розробці веб-додатків та мобільних додатків. Шаблон MVC дозволяє розділити додаток на три основні компоненти: модель, представлення та контролер.

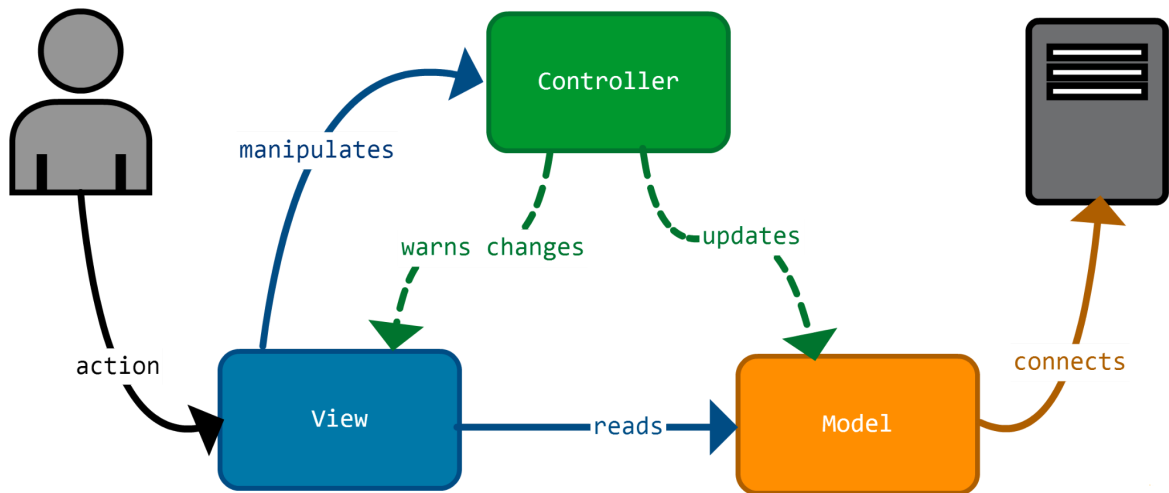


Рисунок 2.3 – Схема шаблону Model-View-Controller [7]

Модель (Model) представляє зовнішні сервіси або дані, які відображаються на екрані. Це може включати дані з бази даних, API або інших джерел.

Представлення (View) відповідає за візуальне відображення даних користувачу на екрані. Його реалізують за допомогою HTML, CSS та JavaScript.

Контролер (Controller) взаємодіє з моделлю та представленням. Він обробляє дії користувача, наприклад, кліки на кнопки, і викликає відповідні методи моделі або оновлює відображення на екрані.

Застосування шаблону MVC дозволяє розділити додаток на основні компоненти, що полегшує підтримку та розвиток додатку. Якщо потрібно змінити частину додатку, необхідно втручатися лише в цю частину, не впливаючи на інші. Крім того, MVC дозволяє розробникам вносити зміни в додаток, не впливаючи на інші його частини. Це сприяє більш ефективному розвитку додатку, економить час і зусилля. Використання шаблону MVC допомагає створити простий та легко підтримуваний додаток, а розробникам - працювати з ним ефективно й з меншими зусиллями.

### 3 ІНФОРМАЦІЙНА МОДЕЛЬ ПРЕДМЕТНОЇ ОБЛАСТІ

В ІС «Автоматизована інформаційна система формування розкладу занять для освітніх установ» слід виділити наступні сутності:

- розклад (Schedule) – містить інформацію про розклад який має бути згенерований наданий користувачем;
- навчальна кімната (Classroom) – містить інформацію про типи навчальних кімнат за різними характеристиками, та список навчальних кімнат за номерами які за цими типами закріплені;
- розклад-звіт (ScheduleReport) – містить інформацію про згенерований ІС готовий до аналізу, доробок та використання розклад;
- тип-навчання (LessonType) – містить інформацію про тип навчання який можна використовувати під час створення розкладу;
- сервіс-пресет (PresetService) – сервіс роботи з пресетами розкладу занять та робочих класів;
- розклад-звіт-сервіс (ScheduleReportService) – сервіс роботи та генерації звітів розкладу занять, їх обробка, збереження у потрібному форматі та видачу;
- представлення-генерації-розкладу (GenerateScheduleView) – представлення яке відповідає за комунікацію між користувачем та сервісом генерації розкладу;
- представлення-розклад-звіт (ScheduleReportView) – представлення яке відповідає за комунікацію.

#### **4 МОДЕЛЬ ПРОГРАМНОГО ДОДАТКУ**

З погляду моделі, система автоматизованого формування розкладу занять для освітніх установ може бути розглянута як єдиний додаток з єдиною точкою входу, що використовує монолітну архітектуру.

Додаток не має механізму авторизації, тому після запуску користувач одразу потрапляє на головну сторінку, де відображається функціонал для виконання завдань у вигляді кнопок та форм.

Якщо сервер додатку буде розміщено на віддаленому сервері (замість localhost), то з будь-якого пристрою можна без затримок використовувати додаток. Ця система також надає можливість легко розширювати функціонал, додаючи нові можливості.

Система організована у вигляді директорій та файлів, кожен з яких виконує певну окрему функцію. Клієнтська частина з інтерфейсом користувача надсилає запит контролеру (який описаний в файлі urls), а контролер передає інформацію по ланцюжку системи, де вже виконуються операції з об'єктами (моделями).

## 5 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Інформаційна система розроблена у формі веб-додатку.

Для створення програмного додатку використовуються Python [8] із фреймворком Django [9] (на серверній стороні), а також використовується готовий бібліотечний компонент django-bootstrap (на клієнтській стороні).

Python є високорівневою мовою програмування загального призначення. Він був створений Гуїдо ван Россумом в кінці 1980-х років і став однією з найпопулярніших мов програмування.

Python відрізняється простим і зрозумілим синтаксисом, що полегшує читання і написання коду. Він підтримує різні стилі програмування, включаючи об'єктно-орієнтоване, функціональне та скриптове програмування. Python використовується в різних галузях, таких як веб-розробка, наукове моделювання, аналіз даних та машинне навчання.

Основні переваги Python включають:

- 1) Простий синтаксис: Python має простий і зрозумілий синтаксис, що полегшує його читання і написання.
- 2) Багато бібліотек і інструментів: Python має широкий вибір встановлених бібліотек і інструментів, що допомагають вирішувати різноманітні задачі.
- 3) Платформонезалежність: Python може працювати на різних операційних системах, таких як Windows, macOS і Linux.
- 4) Підтримка різних парадигм програмування: Python підтримує об'єктно-орієнтоване, функціональне та процедурне програмування, що дозволяє використовувати його для різних типів проєктів.

Django є веб-фреймворком на Python, який допомагає розробникам швидко і ефективно створювати веб-додатки. Він надає готову структуру для розробки, включаючи управління базами даних, маршрутизацію URL та систему переглядів.

Django є одним з найпопулярніших (рис. 5) веб-фреймворків і відомий своєю безпекою та розширюваністю. Він має широкий спектр інструментів для захисту від веб-атак, шифрування даних та аутентифікації. Також існує велика кількість додаткових пакетів, які допомагають розширити функціональність веб-додатків.

Однією з великих переваг Django є його висока продуктивність і легкість використання. Він дозволяє швидко реалізувати ідеї з меншим обсягом коду порівняно з іншими фреймворками. Django також відомий своєю модульністю і гнучкістю, що дозволяє легко додавати або видаляти компоненти в проєкті та працювати з різними типами баз даних, щоб вибрати найбільш підходящий для проєкту.

| Framework     | Github Score | Stack Overflow Score | Overall Score |
|---------------|--------------|----------------------|---------------|
| React         | 99           | 97                   | 98            |
| ASP.NET MVC   |              | 95                   | 95            |
| Angular       | 91           | 96                   | 93            |
| Ruby on Rails | 87           | 99                   | 93            |
| AngularJS     | 90           | 97                   | 93            |
| Vue.js        | 100          | 87                   | 93            |
| Django        | 89           | 97                   | 93            |
| Laravel       | 90           | 93                   | 91            |
| ASP.NET       | 80           | 100                  | 90            |
| Spring        | 86           | 94                   | 90            |
| Express       | 88           | 87                   | 87            |
| Flask         | 89           | 83                   | 86            |
| Meteor        | 86           | 80                   | 83            |
| Symfony       | 81           | 86                   | 83            |
| CodeIgniter   | 79           | 86                   | 82            |
| JSF           |              | 81                   | 81            |
| Ember.js      | 80           | 78                   | 79            |
| .NET Core     | 77           | 80                   | 78            |

Рисунок 5 – Рейтинг веб фреймворків [10]

## 6 АЛГОРИТМ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНИХ ЗАДАЧ

### 6.1 Огляд алгоритма роботи ІС

- 1) Отримання та обробка даних користувача.

Дані з файлів excel зчитуються і зберігаються у відповідних структурах даних.

- 2) Ініціалізація псевдо моделей та початкових класів.

Ініціалізуються базові класи, псевдо моделі, пусті матриці та структури даних.

- 3) Налаштування псевдо моделі.

Порожні поля у матриці заповнюються відповідними класами та групами. Задається порядок класів для кожного типу класу та групи. Встановлюються порожні розклади для вчителів та груп.

- 4) Створення початкового варіанту розкладу.

Класи розміщуються у порожніх місцях матриці вертикально та послідовно, так щоб вони відповідали навчальним кімнатам.

5) Використання one + one стратегію еволюції для жорстких обмежень. Вибираються класи, які будуть змінювати своє розташування в розкладі на основі їх вартості, враховуючи лише жорсткі обмеження. Класи з найвищою вартістю мають шанс на мутацію. Мутація полягає у зміні положення класу в матриці, пошуку вільного поля, що задовольняє всім жорстким обмеженням.

6) Оптимізація та пошук найкращого варіанту який буде задовольняти легким обмеженням. Розклад оптимізується шляхом мінімізації часу простою для груп на всі дні та наявності вільної години без занять. У кожній ітерації випадкова кількість класів мутує, тобто їх положення в матриці змінюється, щоб задовольняти жорстким обмеженням.

7) Формування розкладу та статистики і вивантаження розкладу у форматі excel.

## 6.2 Рішення та еволюційна стратегія

Кожне заняття на факультеті представлено у вигляді блоку тривалості від однієї до чотирьох годин. Для проведення кожного заняття потрібні викладач, аудиторія, тривалість заняття та групи учнів, які беруть участь у занятті.

Також передбачено, які групи відвідують який клас, і всі класи мають однаковий розмір, тобто, кожна група може вміститися в будь-який клас. Навчання проводиться на факультеті з 7:00 до 18:00 кожного робочого дня.

Алгоритм розкладу можна представити у вигляді таблиці, де стовпці відповідають навчальним кімнатам, рядки вказують доступний час для занять, а комірки таблиці містять конкретні заняття, які проводяться у відповідних аудиторіях протягом заданого часу. Така таблиця є матрицею розміром 60 x кількість аудиторій, де 60 відповідає загальній кількості можливих часів протягом тижня (5 робочих днів, 12 годин на день).

Елементи матриці можуть бути порожніми, якщо в певний час аудиторія не використовується для занять, або вони можуть містити відповідний індекс класу. Індекс заняття визначається за допомогою прізвища викладача, предмету, типу заняття (лекція, практичне заняття або лабораторна робота), тривалості, доступних аудиторій та груп.

Оскільки заняття можуть мати різну тривалість, одні й ті ж індекси можуть з'являтися в таблиці кілька разів.

## 7 ПРОГРАМНА РЕАЛІЗАЦІЯ

Кожна модель реагує на команди контролерів в яких йде звернення до цієї моделі, виклик до контролерів приходить з представлення, таким чином і працює архітектурна модель проекту MVC (Model-View-Controller). Перелік моделей, представлень та контролерів наведено в таблиці нижче (табл. 7.1).

Список контролерів:

- `GenerateScheduleViewAPI` – забезпечує зв'язок з головною сторінкою, де проходить генерація розкладу та перегляд вже згенерованих репортів розкладу;
- `AdminAPI` – забезпечує зв'язок з `DefaultAdminSiteView`, де проходить зв'язок з усім додатком збоку адміністратора та доступ до адмін панелі який є вже предустановленим за замовчуванням у Django;
- `ScheduleReportViewAPI` – забезпечує зв'язок з завантаженням репорту розкладу занять для користувача.

Додаток на Django складається із модулів View (які попри назву, виконують функцію контролерів), які не залежать один від одного. Нижче можна побачити код модуля `views.py` (лістинг 7.1) який складається з представленням, які отримують та повертають дані у форматі HTTP.

Таблиця 7.1 – Структура ПЗ

| View                              | Controller                           | Model                                                                                                                 |
|-----------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>GenerateScheduleView</code> | <code>GenerateScheduleViewAPI</code> | <code>LessonType</code> , <code>Schedule</code> , <code>Classroom</code> , <code>ScheduleReport</code>                |
| <code>AdminView</code>            | <code>AdminAPI</code>                | All models ( <code>LessonType</code> , <code>Schedule</code> , <code>Classroom</code> , <code>ScheduleReport</code> ) |
| <code>ScheduleReportView</code>   | <code>ScheduleReportViewAPI</code>   | <code>ScheduleReport</code>                                                                                           |

```

import os
from typing import Any

import openpyxl
from django.http import HttpResponse, HttpResponseNotFound
from django.shortcuts import render
from django.views import View

from university_schedule_generator.forms import
GenerateScheduleForm
from university_schedule_generator.models import LessonType
from university_schedule_generator.services import PresetService,
ScheduleReportService
from university_schedule_generator.settings import BASE_DIR


class GenerateScheduleView(View):
    def __init__(self, **kwargs: Any) -> None:
        super().__init__(**kwargs)

        self.schedule_report_service = ScheduleReportService()
        self.preset_service = PresetService()

        self.data = {
            "preset_schedule":
self.preset_service.get_schedule_by_path(),
            "preset_classrooms":
self.preset_service.get_classrooms_by_path(),
            "lesson_types": LessonType.list(),
        }

        template_name = "university_schedule_generator/index.html"

    def get(self, request):
        data = self.data

        data["form"] = GenerateScheduleForm()
        data["schedule_reports"] =
self.schedule_report_service.get_all()

        return render(request, self.template_name, data)

```

Лістинг 7.1 – Програмний код модуля views.py

```

def post(self, request):
    data = self.data

    print(request.POST)
    form = GenerateScheduleForm(request.POST, request.FILES)
    data["form"] = form

    print(form.is_valid())
    print(form.errors)
    if form.is_valid():
        workbook_path =
self.schedule_report_service.generate(form)
        data["workbook_fname"] =
os.path.basename(workbook_path)

        data["schedule_reports"] =
self.schedule_report_service.get_all()

    return render(request, self.template_name, data)

class ScheduleReportView(View):
    def get(self, request, file_name):
        file_location = os.path.join(
            BASE_DIR,
            "university_schedule_generator",
            "data",
            "schedule_report",
            file_name,
        )

        try:
            workbook = openpyxl.load_workbook(file_location)
        except IOError:
            response = HttpResponseRedirect("File doesnt exist")
            response = HttpResponseRedirect(

content_type="application/vnd.openxmlformats-officedocument.spread
sheetml.sheet"
        )
        response["Content-Disposition"] = f'attachment;
filename="{file_name}"'
        response.write(workbook)

    return response

```

Лістинг 7.1 – Програмний код модуля views.py, лист 2

Код модуля `models.py` (лістинг 7.2) складається з моделей, які виконують роль зберігача даних в “runtime” часі, та полегшують маніпуляції з даними завдяки структуризації та абстракції.

```
from datetime import datetime
import enum
import os

# Enums
class LessonType(enum.Enum):
    LECTURES = "L"
    PRACTICE = "P"
    EXERCISES = "E"

    @staticmethod
    def list():
        return list(map(lambda c: c, LessonType))

# Data structures
class Schedule:
    def __init__(self, *args, **kwargs):
        self.subject = kwargs.get("Subject", None)
        self.type = kwargs.get("Type", None)
        self.lecturer = kwargs.get("Lecturer", None)
        self.classroom = kwargs.get("Classroom", None)
        self.duration = kwargs.get("Duration", None)
        self.groups = kwargs.get("Groups", [])
        if self.groups:
            self.groups = self.groups.split(" ")

    @property
    def raw_groups(self):
        return " ".join(self.groups)

    @property
    def to_json_request(self):
        return {
            "Subject": self.subject,
            "Type": self.type,
            "Lecturer": self.lecturer,
            "Classroom": self.classroom,
            "Duration": self.duration,
            "Groups": self.groups,
        }
```

Лістинг 7.2 – Програмний код модуля `models.py`

```

class Classroom:
    def __init__(self, *args, **kwargs):
        self.classroom = kwargs.get("Classroom", None)
        self.rooms = kwargs.get("Rooms", [])
        if self.rooms:
            self.rooms = self.rooms.split(" ")

    @property
    def raw_rooms(self):
        return " ".join(self.rooms)

class ScheduleReport:
    def __init__(self, *args, **kwargs):
        self.file_path = kwargs.get("file_path", None)

    @property
    def file_name(self):
        return os.path.basename(self.file_path) if self.file_path
    else ""

    @property
    def created_at(self):
        date_string =
self.file_name.split("|")[2].removesuffix(".xlsx")
        datetime_object = datetime.strptime(date_string, "%Y-%m-%d
%H:%M:%S")
        return datetime_object

```

## Лістинг 7.2 – Програмний код модуля models.py, лист 2

Код модуля services.py (лістинг 7.3) складається з сервісів, які виконують роль “виконувача” різного типу операцій та маніпуляцій з даними, він є зв’язком між представленням та моделями, який виконує операції в обидві сторони, як отримання так і збереження. Завдяки цьому розділенню ІС є дуже гнучкою та може одночасно працювати як на мобільних так і десктопних платформах маючи змогу оперувати однаковими даними у реальному часі.

```

from datetime import datetime
import os
from algorithm.main import run

from university_schedule_generator.forms import
GenerateScheduleForm
from university_schedule_generator.models import Classroom,
Schedule, ScheduleReport
from university_schedule_generator.settings import BASE_DIR
from university_schedule_generator.utils import (
    file_csv_to_dict,
    file_path_csv_to_dict,
    save_as_json,
)

class PresetService:
    preset_dirpath = os.path.join(
        BASE_DIR, "university_schedule_generator", "data",
        "preset"
    )
    schedule_path = os.path.join(preset_dirpath, "schedule.csv")
    classrooms_path = os.path.join(preset_dirpath,
        "classrooms.csv")

    def get_schedule_by_path(self, schedule_path=None):
        if schedule_path is None:
            schedule_path = self.schedule_path
        data_dict = file_path_csv_to_dict(schedule_path)

        return [Schedule(**dict) for dict in data_dict]

    def get_schedule_by_csv_file(self, csv_file):
        data_dict = file_csv_to_dict(csv_file)

        return [Schedule(**dict) for dict in data_dict]

    def get_classrooms_by_path(self, classrooms_path=None):
        if classrooms_path is None:
            classrooms_path = self.classrooms_path
        data_dict = file_path_csv_to_dict(classrooms_path)

        return [Classroom(**dict) for dict in data_dict]

```

Лістинг 7.3 – Програмний код модуля services.py

```

def get_classrooms_by_csv_file(self, csv_file):
    data_dict = file_csv_to_dict(csv_file)

    return [Classroom(**dict) for dict in data_dict]

class ScheduleReportService:
    def __init__(self) -> None:
        self.preset_service = PresetService()

        schedule_report_dirpath = os.path.join(
            BASE_DIR, "university_schedule_generator", "data",
            "schedule_report"
        )
        schedule_request_dirpath = os.path.join(
            BASE_DIR, "university_schedule_generator", "data",
            "schedule_request"
        )

    def get_all(self):
        file_paths = []
        for root, directories, files in
os.walk(self.schedule_report_dirpath):
            for file_name in files:
                file_path = os.path.join(root, file_name)
                file_paths.append(file_path)

        schedule_reports = sorted(
            [ScheduleReport(file_path=fp) for fp in file_paths],
            key=lambda x: x.created_at,
            reverse=True,
        )
        return schedule_reports

    def generate(self, form: GenerateScheduleForm):
        schedule_request_path = self._save_as_json(form)
        workbook_path = run(schedule_request_path)

        return workbook_path

    def _save_as_json(self, form: GenerateScheduleForm):
        preset_schedule_csv_file =
form.cleaned_data["preset_schedule_csv_file"]
        preset_classrooms_csv_file =
form.cleaned_data["preset_classrooms_csv_file"]

```

```

        preset_schedules =
self.preset_service.get_schedule_by_csv_file(
    preset_schedule_csv_file.file
)
        preset_classrooms =
self.preset_service.get_classrooms_by_csv_file(
    preset_classrooms_csv_file.file
)

        schedule_request_dict = {
            "Schedule": [ps.to_json_request for ps in
preset_schedules],
            "Classroom": {pc.classroom: pc.rooms for pc in
preset_classrooms},
        }
        schedule_request_path =
self._get_schedule_request_path(form)

        save_as_json(schedule_request_dict, schedule_request_path)

        return schedule_request_path

    def _get_schedule_request_path(self, form:
GenerateScheduleForm):
        preset_schedule_csv_file =
form.cleaned_data["preset_schedule_csv_file"]
        preset_classrooms_csv_file =
form.cleaned_data["preset_classrooms_csv_file"]

        current_datetime = datetime.now()
        current_datetime_str = current_datetime.strftime("%Y-%m-%d
%H:%M:%S")

        preset_schedule_str =
str(preset_schedule_csv_file).removesuffix(".csv")
        filenamepreset_classrooms_str =
str(preset_classrooms_csv_file).removesuffix(
            ".csv"
        )

        return os.path.join(
            self.schedule_request_dirpath,

f"{str(preset_schedule_str)}|{str(filenamepreset_classrooms_str)}|{c
urrent_datetime_str}.json",
        )

```

## 8 ІНСТРУКЦІЯ КОРИСТУВАЧА

В даному розділі розглядається розв’язання користувацьких задач за допомогою розробленого UX/UI (інтерфейсу).

Після підключення до веб-адреси додатку, користувач бачить головну сторінку генерації розкладу занять (рис 8.1).

**Schedule Generator**

Generate new schedule

Select a preset schedule CSV file\*

Choose file No file chosen

Select a preset classrooms CSV file\*

Choose file No file chosen

Generate

**FAQ for presets CSV**

**Preset Schedule template**

| # | Subject            | Type | Lecturer         | Classroom | Duration | Groups                                                  |
|---|--------------------|------|------------------|-----------|----------|---------------------------------------------------------|
|   | Computer Science   | L    | Dennis Ritchie   | L         | 2        | 123_1c 123_2c 123_3c 123_4c 126_1c 126_2c 126_3c 126_4c |
|   | Ukrainian Language | P    | Taras Shevchenko | P         | 1        | 123_1c 126_1c                                           |

**Preset Classrooms template**

Рисунок 8.1 – Головна сторінка

З лівою сторони екрану, на панелі навігації, користувач може побачити список згенерованих раніше репортів розкладу занять (рис 8.2).

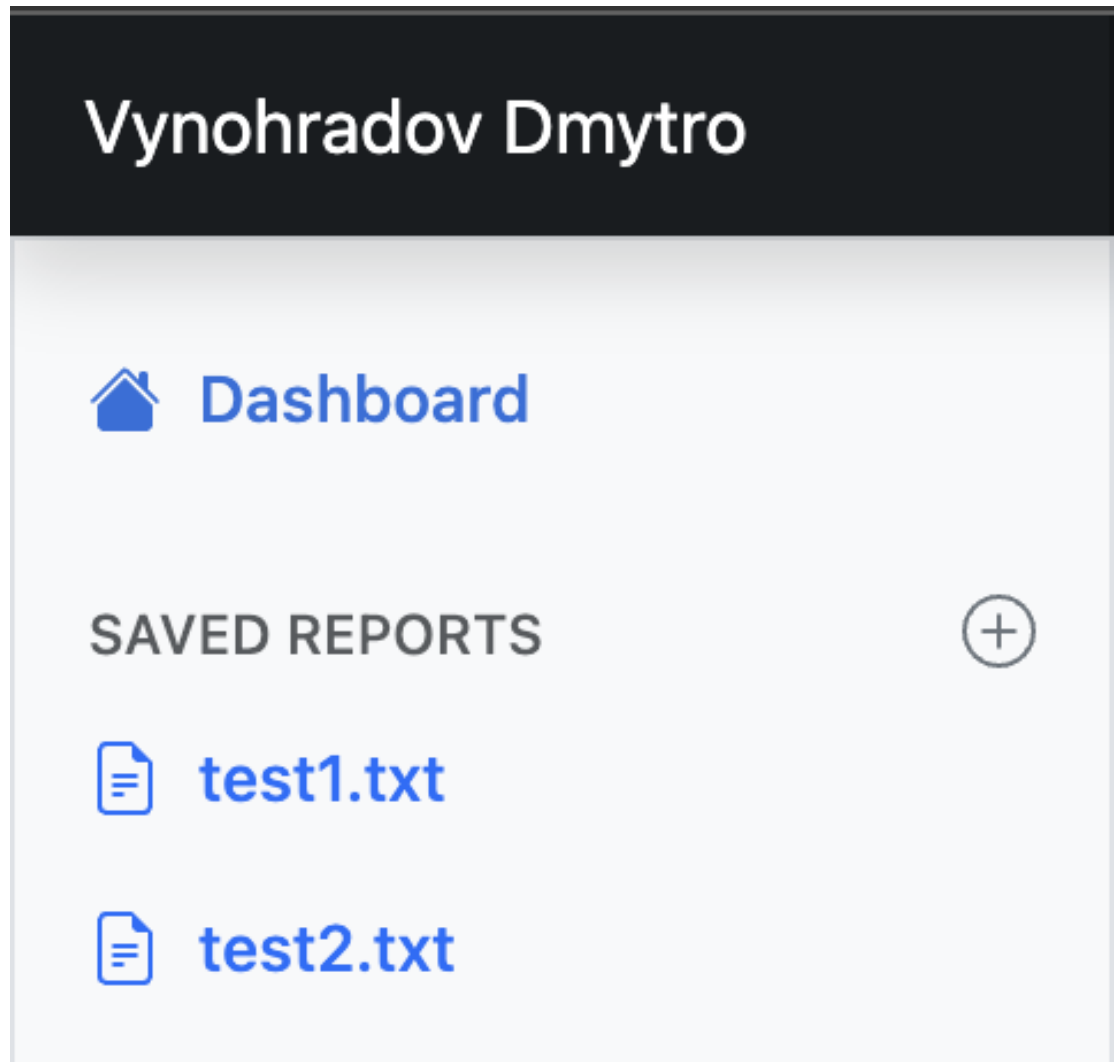


Рисунок 8.2 – Панель навігації

Нижче користувач бачить FAQ (підказки та опис) щодо правил користування та CSV файлів, які потрібно надати системі, аби згенерувати потрібний розклад учбових занять (рис 8.3).

### FAQ for presets CSV

#### Preset Schedule template

| # | Subject            | Type | Lecturer         | Classroom | Duration | Groups                                                  |
|---|--------------------|------|------------------|-----------|----------|---------------------------------------------------------|
|   | Computer Science   | L    | Dennis Ritchie   | L         | 2        | 123_1c 123_2c 123_3c 123_4c 126_1c 126_2c 126_3c 126_4c |
|   | Ukrainian Language | P    | Taras Shevchenko | P         | 1        | 123_1c 126_1c                                           |

#### Preset Classrooms template

| # | Classroom | Rooms          |
|---|-----------|----------------|
|   | L         | 1l 2l 3l 4l 5l |
|   | P         | 1p 2p 3p 4p 5p |

#### Type of lessons available

| Type | Defination  |
|------|-------------|
|      | L LECTURES  |
|      | P PRACTICE  |
|      | E EXERCISES |

Рисунок 8.3 – Правила користування та CSV файлів

Після того як користувач натиснув кнопку “Generate”, розклад буде згенерований і користувач побачить повідомлення “Schedule was successful generated and available here” (рис 8.4).

## Generate new schedule

Select a preset schedule CSV file\*

|             |                |
|-------------|----------------|
| Choose file | No file chosen |
|-------------|----------------|

Select a preset classrooms CSV file\*

|             |                |
|-------------|----------------|
| Choose file | No file chosen |
|-------------|----------------|

**Generate**

**Schedule was successful generated and  
available  
 here**

Рисунок 8.4 – Повідомлення успішної генерації розкладу

Натиснувши на текст “here”, розклад буде завантажений на пристрій користувача. Згенерований розклад має вигляд таблиці та формат файлу excel (рис 8.5), (рис 8.6).

| Monday |                            |      |                      |           |          |        |            |
|--------|----------------------------|------|----------------------|-----------|----------|--------|------------|
| Index  | Subject                    | Type | Lecturer             | Classroom | Duration | Groups | Time       |
| 39     | Senior Editor              | L    | Arly Posnett         | 3l        | 2        | 123_1c | 7, 8       |
| 73     | Senior Sales Associate     | P    | Barb Auchterlonie    | 9p        | 2        | 126_4c | 7, 8       |
| 56     | Pharmacist                 | P    | Corbin Birley        | 2p        | 2        | 126_2c | 7, 8       |
| 90     | Senior Financial Analyst   | P    | Jack Jowett          | 3p        | 2        | 123_2c | 7, 8       |
| 26     | Chemical Engineer          | P    | Jemmy Mallam         | 1p        | 3        | 123_3c | 7, 8, 9    |
| 72     | Compensation Analyst       | E    | Sergent Northill     | 1e        | 3        | 126_3c | 7, 8, 9    |
| 38     | Quality Engineer           | L    | Gerek Dunwoody       | 4l        | 1        | 126_2c | 9          |
| 24     | Quality Control Specialist | L    | Ginnifer Wilmot      | 5l        | 3        | 123_4c | 9, 10, 11  |
| 11     | Quality Control Specialist | P    | Marylee Touse        | 3p        | 3        | 126_4c | 9, 10, 11  |
| 16     | Help Desk Operator         | L    | Sumner Wesker        | 10l       | 1        | 123_2c | 10         |
| 21     | Design Engineer            | E    | Corri Freeman        | 3e        | 2        | 123_3c | 10, 11     |
| 10     | Biostatistician II         | P    | Kassie Edridge       | 8p        | 2        | 126_1c | 10, 11     |
| 18     | Compensation Analyst       | P    | Scott Jura           | 10p       | 3        | 126_3c | 10, 11, 12 |
| 48     | Actuary                    | L    | Chelsey Corneliussen | 2l        | 3        | 126_2c | 10, 11, 12 |
| 47     | Actuary                    | P    | Gannon Kester        | 2p        | 3        | 123_1c | 11, 12, 13 |
| 23     | VP Accounting              | P    | Amanda Solway        | 4p        | 1        | 123_2c | 12         |
| 69     | Cost Accountant            | L    | Richy Scotter        | 5l        | 2        | 123_3c | 12, 13     |
| 5      | VP Product Management      | L    | Mitchell Breacher    | 9l        | 2        | 126_1c | 12, 13     |
| 17     | Engineer II                | E    | Whit Janman          | 7e        | 3        | 126_4c | 12, 13, 14 |
| 54     | Programmer Analyst III     | P    | Moore Killiner       | 4p        | 1        | 123_2c | 13         |
| 4      | Account Coordinator        | L    | Rudiger Fuentez      | 6l        | 2        | 126_3c | 13, 14     |
| 88     | Account Executive          | P    | Rand Brede           | 9p        | 3        | 123_3c | 14, 15, 16 |
| 37     | Chief Design Engineer      | P    | Faustine Furmenger   | 4p        | 3        | 126_1c | 14, 15, 16 |
| 43     | Web Developer IV           | L    | Derek Simons         | 8l        | 2        | 126_2c | 15, 16     |
| 15     | Account Coordinator        | P    | Pamela Zima          | 1p        | 1        | 126_4c | 17         |
| 33     | Project Manager            | L    | Grady Placide        | 4l        | 1        | 123_3c | 18         |

Рисунок 8.5 – Приклад згенерованого розкладу на понеділок

| Tuesday |                               |      |                   |           |          |        |            |
|---------|-------------------------------|------|-------------------|-----------|----------|--------|------------|
| Index   | Subject                       | Type | Lecturer          | Classroom | Duration | Groups | Time       |
| 98      | Senior Quality Engineer       | P    | Kizzie Chapman    | 6p        | 2        | 126_3c | 7, 8       |
| 84      | Graphic Designer              | L    | Raimundo Grise    | 1l        | 2        | 126_2c | 7, 8       |
| 45      | Safety Technician I           | L    | Conny Barnaclough | 5l        | 3        | 123_4c | 7, 8, 9    |
| 31      | Registered Nurse              | L    | Kathleen Coxall   | 3l        | 1        | 126_2c | 9          |
| 22      | Sales Associate               | P    | Denny Hercules    | 3p        | 2        | 126_3c | 9, 10      |
| 97      | Dental Hygienist              | L    | Marthe Frostdicke | 1l        | 2        | 123_1c | 10, 11     |
| 50      | Assistant Media Planner       | L    | Lanita Lemanu     | 3l        | 3        | 126_2c | 10, 11, 12 |
| 53      | Analog Circuit Design manager | L    | Isidora Retchless | 1l        | 3        | 126_4c | 12, 13, 14 |
| 70      | Sales Representative          | L    | Mord Ogbourne     | 4l        | 2        | 126_3c | 13, 14     |
| 12      | Help Desk Technician          | P    | Murray Piotrowski | 4p        | 3        | 126_2c | 13, 14, 15 |
| 68      | Safety Technician IV          | E    | Rudy Carmichael   | 2e        | 3        | 126_1c | 15, 16, 17 |
| 67      | Assistant Media Planner       | P    | Nancy Atack       | 5p        | 3        | 126_3c | 15, 16, 17 |
| 19      | Pharmacist                    | L    | Joel Gyde         | 2l        | 2        | 126_2c | 16, 17     |
| 20      | Recruiter                     | P    | Lulu Whittingham  | 3p        | 1        | 123_3c | 18         |

Рисунок 8.6 – Приклад згенерованого розкладу на вівторок

## ВИСНОВОК

У даній дипломній роботі поставлена мета розробити автоматизовану систему генерації розкладу занять для закладів освіти з метою полегшення процесу складання розкладу, зниження трудомісткості та покращення його якості. Для досягнення цієї мети були розглянуті основні проблеми, пов'язані зі складанням розкладу, такі як велика кількість обмежень, розбіжності у вимогах, непередбачувані зміни та великий обсяг даних.

На основі проведених досліджень та аналізу було розроблено алгоритми та методи, що дозволяють автоматизувати процес генерації розкладу занять. Враховуючи обмеження та вимоги різних зацікавлених сторін, система забезпечує оптимальне використання ресурсів та забезпечує якісний розклад, відповідний потребам освітнього закладу.

Результати досліджень показали, що розроблена система є ефективним інструментом для управління розкладом занять. Вона дозволяє зменшити час, затрачений на складання розкладу, знижує ймовірність помилок та конфліктів, а також сприяє покращенню якості навчання та задоволеності всіх учасників освітнього процесу.

Отже, розроблена система є цінним інструментом для університетів та інших закладів освіти, які стикаються з проблемами складання розкладу занять. Її впровадження сприятиме оптимізації використання ресурсів, підвищенню ефективності навчального процесу та поліпшенню задоволеності всіх сторін.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. M. Emmerich, O.M. Shir, and H. Wang: Evolution Strategies. In: Handbook of Heuristics, 1-31. Springer International Publishing (2018). [Електронний ресурс] – Режим доступу: [https://link.springer.com/referenceworkentry/10.1007/978-3-319-07153-4\\_13-1](https://link.springer.com/referenceworkentry/10.1007/978-3-319-07153-4_13-1)
2. Richardson, Leonard; Amundsen, Mike; Ruby, Sam (2013). [Електронний ресурс] – Режим доступу: <https://archive.org/details/restfulwebapis0000rich>
3. Fielding Roy. Architectural Styles and the Design of Network-based Software Architectures [Електронний ресурс] – Режим доступу: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
4. Схема архітектури REST API [Електронний ресурс] – Режим доступу: <https://www.datamounts.com/difference-rest-api-restful-api/>
5. Формат XML та JSON файла [Електронний ресурс] – Режим доступу: [https://commons.wikimedia.org/wiki/File:JSON\\_vs.\\_XML.png](https://commons.wikimedia.org/wiki/File:JSON_vs._XML.png)
6. Adam Freeman. Pro ASP.NET MVC 4, 4th edition, 2013. — 688 с. [Електронний ресурс] – Режим доступу: SBN 978-5-8459-1867-3
7. Схема шаблону Model-View-Controller. [Електронний ресурс] – Режим доступу: <https://upload.wikimedia.org/wikipedia/commons/f/fd/MVC-Process.png>
8. Programming language Python [Електронний ресурс] – Режим доступу: <https://www.python.org/>
9. Django high-level web framework [Електронний ресурс] – Режим доступу: <https://www.djangoproject.com/>
10. Рейтинг веб фреймворків [Електронний ресурс] – Режим доступу: <https://hotframeworks.com/>

