---

Project Report: University Timetable Generator Web Application

---

Introduction

This project involves developing a web application that generates university timetables using a genetic algorithm. The application allows users to upload CSV files containing course, instructor, and student data. It processes this data to create feasible timetables while adhering to constraints such as instructor availability, room scheduling, and student enrollments.

Project Objectives:

- Provide an intuitive interface for uploading data and generating timetables.

- Display progress updates during the generation process.

- Present generated timetables in an accessible format.

- Allow users to view and download timetables directly from the application.

---

Technologies Used

- Python: Backend programming language.

- Flask: Web framework for building the application.

- Pandas: For handling CSV and Excel files.

- NumPy: Numerical computation library.

- DEAP: Library for implementing genetic algorithms.

- Tailwind CSS: Styling the web interface.

- HTML/CSS/JavaScript: Frontend technologies.

---

Project Structure

1. Application Components

- File Upload Interface: For uploading necessary CSV files.

- Timetable Generation: Uses a genetic algorithm to create timetables.

- Progress Monitoring: Displays real-time progress.

- Timetable Viewing/Downloading: Generated timetables can be viewed online or downloaded as Excel files.

2. Code Structure

- `app.py`: The main Flask application file.

- `timetable\_generator.py`: Contains the genetic algorithm and timetable generation logic.

- Templates: HTML files for rendering web pages (`index.html`, `progress.html`, etc.).

- Static Files: CSS and JavaScript, including Tailwind CSS for styling.

---

Features and Functionality

1. File Uploading

- Users can upload three CSV files:

- `Instructors\_Courses.csv`

- `Backlog.csv`

- `Elective.csv`

- The application validates the uploaded files to ensure the correct format.

2. Timetable Generation

- The genetic algorithm considers constraints like instructor conflicts, room availability, student schedules, and lab sessions.

- The algorithm runs in a separate thread to ensure the web app remains responsive.

3. Progress Monitoring

- A progress page shows a progress bar and status messages, updated via AJAX.

4. Viewing Timetables

- Users can view timetables categorized by:

- Batch Timetables

- Instructor Timetables

- Student Timetables

- The timetables are displayed with improved table formatting, search functionality, and hover effects.

5. Downloading Timetables

- Timetables can be downloaded as Excel files.

---

Styling and User Interface

1. Tailwind CSS Integration

- Tailwind CSS provides a modern and clean look for the application.

- Utility-first CSS classes are used directly in HTML templates for rapid styling.

2. Table Formatting

- Tables are styled with borders, alternating row colors, and hover effects for better readability.

- Pandas `Styler` is used to apply these styles before rendering the tables as HTML.

3. User Experience

- The app features consistent navigation, including 'Back' buttons and easy access to timetables.

- Input fields and buttons are styled cohesively for an improved user experience.

---

How to Run the Application

Prerequisites:

- Python 3.6 or higher: Ensure Python is installed.

- Required Packages: Install using:

```bash

pip install flask pandas numpy xlsxwriter deap openpyxl werkzeug

```

Steps:

1. Clone the repository.

2. Navigate to the project directory:

```bash

cd path\_to\_project\_directory

```

3. Run the application:

```bash

python app.py

```

4. Access the web app via `http://localhost:5000/`.

---

Conclusion

The project successfully delivers a user-friendly web application for generating and managing university timetables. By combining a genetic algorithm with a responsive web interface, users can efficiently produce timetables while ensuring constraints are met. Tailwind CSS and AJAX updates contribute to a positive and interactive user experience.

---

Future Improvements

- Error Handling: Improve error messages and validation.

- Performance Optimization: Enhance the speed of the genetic algorithm.

- User Authentication: Add login functionality.

- Additional Features: Enable timetable editing and export options.

---

This version is now concise, logically structured, and free of redundancy. If you want a document version (like a `.docx` or `.pdf`), the next step is to convert this cleaned-up content into the appropriate file format using a tool or script. Let me know how you'd like to proceed!