Synopsis

On

TimelyFlow – Smart Timetable Generator

Submitted in the partial fulfilment of the requirement for the award of degree of
Bachelor of Technology Computer Science and Engineering

Batch (2022 - 2026)

Submitted To: Ms.Mehak Kohli
Assistant Professor

Submitted By: Meenakshi
Reg. No: 12200149

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DAV UNIVERSITY

JALANDHAR-PUNJAB 144012

ACKNOWLEDGEMENT

I express my heartfelt gratitude to all those who provided invaluable support and guidance throughout the development of this project.

Firstly, I would like to express my sincere gratitude and indebtedness to **Dr. Manik Sharma**, Dean Academics, and **Dr.Ridhi Kapoor**, Head of Training and Placement, Department of Computer Science and Engineering, for their encouragement and support during this journey.

I am also grateful to all the faculty members of the Department of Computer Science and Engineering for their inspiration, technical guidance, and assistance in preparing the final report and presentation.

Finally, I thank my peers, family, and friends for their unwavering support and motivation throughout the process.

DECLARATION

I, Meenakshi, hereby declare that the work presented in this project titled "**TimelyFlow – Smart Timetable Generator**" is an authentic record of my own efforts. This project has been completed in partial fulfilment of the requirements for the award of the Bachelor of Technology (B. Tech) Degree in Computer Science and Engineering under the guidance of **Dr. Rahul Hans** (Head of Department). To the best of my knowledge, the content of this report has not been submitted to any other University or Institute for the award of any degree or diploma .

Meenakshi(12200149)

ABSTRACT

Managing academic schedules in educational institutions can be a complex and time-consuming task, often involving conflicts between faculty availability, room capacities, and subject allocations. TimelyFlow – Smart Timetable Generator is a web-based application developed using Python and Streamlit to simplify this process by providing an interactive, automated, and user-friendly platform for timetable management.

The system allows users to automatically generate weekly timetables with intelligent assignment of subjects, faculty, and rooms while incorporating lunch breaks and other scheduling constraints. Additionally, it provides a manual entry interface for custom adjustments, and supports Excel/CSV file upload and download for ease of use.

TimelyFlow features an engaging dark-mode interface with animated backgrounds, interactive pivot-grid visualization of timetables, and responsive design to enhance usability. By integrating automation with flexibility, the project reduces administrative workload, improves efficiency, and ensures a smooth scheduling experience for students, faculty, and administrators alike.

This project demonstrates the application of data-driven and interactive techniques in solving real-world scheduling challenges, making timetable generation faster, smarter, and more accessible.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction to TimelyFlow – Smart Timetable Generator

TimelyFlow is a Python and Streamlit-based intelligent platform designed to automate timetable management for educational institutions. The system simplifies the process of scheduling classes, faculty assignments, and room allocations by combining data-driven algorithms, rule-based scheduling, and interactive UI components.

It addresses challenges such as scheduling conflicts, manual errors, and inefficient timetable management by providing automated generation, manual customization, and file-based timetable handling. The platform ensures a user-friendly experience with responsive tables, pivot-grid views, and interactive controls.

1.2 Advantages of TimelyFlow

1. Automated Timetable Generation – Assigns subjects, faculty, and rooms intelligently.
2. Manual Editing – Allows custom timetable entries and adjustments.
3. File Handling – Supports CSV/Excel upload and download for existing and generated timetables.
4. Interactive Visualization – Provides pivot-grid and table view for easy analysis.
5. Constraint Handling – Accounts for faculty availability, room capacity, and lunch breaks.
6. User-Friendly Interface – Dark-mode, animated background, and responsive buttons.
7. Scalable & Adaptable – Can be used in schools, colleges, or universities.
8. Time-Saving – Reduces administrative workload and scheduling conflicts.

1.3 Features of TimelyFlow

1. Faculty & Room Management – Add, edit, and view faculty and room details.
2. Automatic Timetable Generation – Creates weekly timetables based on constraints.
3. Manual Timetable Entry – Enter and edit custom schedules.
4. File Upload & Download – Import existing timetables and export generated ones.
5. Pivot-Grid Timetable View – Quickly visualize timetable by day and slot.

6. Dynamic Lunch Breaks – Automatically inserts lunch breaks in the schedule.

1.4 Technologies Used

1. Frontend & Interface
   - Streamlit → Interactive web interface
   - Custom CSS → Animated backgrounds, responsive buttons, and dark mode
2. Backend & Logic
   - Python → Scheduling algorithms and data processing
   - Pandas → Handling tabular data and Excel/CSV files
   - Random & Rule-Based Algorithms → Timetable generation with constraints
3. Utilities
   - Openpyxl → Read/write Excel files
   - Pathlib → File path management

Chapter 2

PROJECT TITLE: TimelyFlow – Smart Timetable Generator

2.1 Project Problem Statement

Creating and managing timetables manually is a tedious, error-prone, and time-consuming task in schools and colleges. Conflicts between faculty availability, room capacity, and subject allocation are common and lead to inefficient scheduling.

TimelyFlow solves this problem by providing a smart, automated timetable generation system with features for manual adjustment, file management, and interactive visualization.

2.2 Project Description

TimelyFlow automates the process of creating weekly class schedules, intelligently assigning faculty, subjects, and rooms, while considering lunch breaks and other constraints. Users can also manually add or modify entries. The platform supports CSV/Excel upload and download, providing flexibility and ease of integration with existing systems.

The system is designed for accessibility and interactivity, featuring dark-mode styling, animated backgrounds, and pivot-grid visualizations of the timetable. It reduces conflicts, saves administrative time, and ensures accurate and flexible scheduling.

2.3Project Hierarchy

1. Backend
   - Handles timetable logic, randomization, and constraint checking
   - Processes Excel/CSV files and manages faculty/room data
2. Frontend
   - Displays interactive tables and pivot-grid views
   - Provides sliders, buttons, and input fields for data entry
3. Algorithm Modules
   - Automatic timetable generation with randomization & constraints
   - Manual entry and editing
   - Dynamic lunch break handling

2.4 Aim

The aim of TimelyFlow is to provide an efficient, automated, and user-friendly timetable management system that reduces administrative workload and scheduling conflicts in educational institutions.

Specific Objectives:

- Automatically generate weekly timetables
- Provide manual timetable entry and editing
- Manage faculty and room data efficiently
- Offer Excel/CSV upload and download
- Visualize timetables in grid and table format
- Maintain scalability and adaptability for various institutions

Chapter 3

Feasibility Study

A feasibility study of the TimelyFlow – Smart Timetable Generator project ensures that the system is technically, operationally, and economically viable.

1. Technical Feasibility: - The project uses Python, Streamlit, Pandas, and other widely available libraries. - It requires only a standard computer system with internet access. - Open-source libraries reduce dependency on proprietary software.

2. Operational Feasibility: - The application simplifies scheduling tasks for administrators and faculty. - The intuitive Streamlit interface makes it easy to use, even for non-technical users. - Integration with Excel/CSV provides compatibility with existing processes.

3. Economic Feasibility: - The project is cost-effective as it leverages open-source tools. - It minimizes administrative workload, saving time and resources. - Scalable for schools, colleges, and universities without additional costs.

Thus, the project is feasible and beneficial in terms of cost, usability, and efficiency.

Chapter 4

OBJECTIVES

1. Centralized Timetable Management –
   All data about classes, faculty, and rooms is stored in one organized system, making it easy to access and manage schedules efficiently.
2. Automated Scheduling –
   The system intelligently assigns subjects to faculty and rooms based on availability and constraints, reducing manual effort and errors.
3. Manual Editing –
   Users can customize or adjust timetables manually, giving flexibility to handle special cases or last-minute changes.
4. File Management –
   Timetables can be imported from or exported to Excel/CSV files, allowing seamless integration with existing records and easy sharing.
5. Interactive Visualization –
   Pivot-grid and table views help users quickly understand schedules, check conflicts, and view the timetable in a clear format.
6. Constraint Handling –
   The system automatically considers lunch breaks, room capacities, and faculty availability to avoid scheduling conflicts.
7. Cross-Platform Usability –
   TimelyFlow runs on web platforms with a responsive design, ensuring accessibility on different devices like desktops, tablets, and mobiles.

Chapter 5

CODING AND IMPLEMENTATION

4.1Project Architecture

- Frontend: Streamlit with custom CSS and animations
- Backend: Python logic with Pandas, Random, and Openpyx
- Database: Excel/CSV files for storing faculty, rooms, and timetable data

4.2Backend Implementation

- Generates timetable using random selection and constraints
- Handles CSV/Excel file read/write operations
- Manages faculty and room data

4.3Automatic Timetable Module

- Assigns subjects, faculty, and rooms automatically
- Inserts lunch breaks dynamically
- Provides pivot-grid view for quick visualization

4.4Manual Timetable Module

- Allows user to add, edit, and view manual timetable entries
- Supports Excel/CSV export for offline use

4.5Frontend Implementation

- Responsive UI with sliders, input boxes, and download buttons
- Animated backgrounds and dark mode for enhanced usability

Chapter 6
CONCLUSION

TimelyFlow – Smart Timetable Generator successfully integrates automation, interactive visualization, and user-friendly design to provide an efficient solution for timetable management in educational institutions.

Key Outcomes:

- Automated timetable generation with minimal conflicts
- Manual timetable editing and customization
- File upload/download for flexibility
- Interactive and responsive visualization for better understanding

Future Enhancements:

- Integration with faculty calendars for real-time availability
- Conflict detection algorithms for complex institutions
- Mobile app deployment for better accessibility
- AI/ML-based timetable optimization for maximizing efficiency
- Integration with institution portals for seamless updates

Chapter 7

FUTURE SCOPE

The TimelyFlow – Smart Timetable Generator can be further enhanced with the following developments:

1. AI/ML Integration – Use machine learning to optimize scheduling patterns and predict conflicts.

2. Mobile Application – Develop an Android/iOS app for easier accessibility.

3. Cloud Deployment – Host the system on cloud platforms for real-time collaboration and remote use.

4. Integration with LMS – Connect with Learning Management Systems (LMS) like Moodle or Google Clas

5. Advanced Conflict Resolution – Incorporate advanced algorithms to handle large-scale institutional cons

6. Notification System – Send automatic notifications to faculty and students about timetable changes.

Chapter 8

REFERENCES

1. Streamlit Documentation – https://docs.streamlit.io
2. W3Schools CSS Tutorials – https://www.w3schools.com/css/
3. Towards Data Science – https://towardsdatascience.com
4. Pandas Documentation – *Powerful Python data analysis toolkit.*

   Available at: https://pandas.pydata.org
5. Openpyxl Documentation – *Read/Write Excel 2010 xlsx/xlsm files.*

   Available at: https://openpyxl.readthedocs.io
6. Python Software Foundation – *Python official documentation.*

   Available at: https://www.python.org
7. NumPy Documentation – *Fundamental package for scientific computing with Python.*

   Available at: https://numpy.org/doc/
8. Matplotlib Documentation – *Comprehensive library for creating visualizations in Python.*

   Available at: https://matplotlib.org/stable/contents.html
9. GeeksforGeeks – *Python Programming Language Tutorials.*

   Available at: https://www.geeksforgeeks.org/python-programming-language/
10. Real Python – *Practical Python tutorials and guides.*

    Available at: https://realpython.com
11. W3Schools CSS Tutorials – *Learn CSS for styling Streamlit apps.*

    Available at: https://www.w3schools.com/css/
12. Towards Data Science – *Articles on data science and machine learning.*

    Available at: https://towardsdatascience.com

13. Analytics Vidhya – *Data Science & Machine Learning Guides.*

Available at: https://www.analyticsvidhya.com

14. KDnuggets – *Resources for AI, Analytics, and Data Science.*

Available at: https://www.kdnuggets.com