Sonika Gupta

22BCS043 Sachin 2025

Research paper 2025

January 2025

Shri Mata Vaishno Devi University(SMVDU), Katra

Document Details

Submission ID

trn:oid:::1:3437914493

Submission Date

Dec 8, 2025, 5:35 PM GMT+5:30

Download Date

Dec 8, 2025, 5:46 PM GMT+5:30

File Name

miniproj_LIBRARY_MANAGEMENT_DESKTOP_APP_report.docx

File Size

598.5 KB

61 Pages

6,377 Words

38,620 Characters

2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

7 Not Cited or Quoted 2%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources

1% Publications

3% Submitted works (Student Papers)

Match Groups

- **7** Not Cited or Quoted 2%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
1% Publications
3% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	**dioscuri-tda.org**	<1%
2	Student papers	**Technological Institute of the Philippines**	<1%
3	Student papers	**Computer Science Engineering**	<1%
4	Internet	**tudr.thapar.edu:8080**	<1%

MINI PROJECT REPORT

TIME TABLE MANAGEMENT SYSTEM

by

Koripella Venkata Ramani

Entry. No.: 22BCS043

Sachin Anand

Entry. No.: 22BCS070

Charanpreet Kour

Entry. No.: 22BCS109

Under the Guidance of

Dr. Sonika Gupta

SHRI MATA VAISHNO DEVI UNIVERSITY, KATRA

(School of Computer Science & Engineering)

JAMMU & KASHMIR – 182 320

Session 2025-26

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who contributed directly or indirectly to the successful completion of our mini project titled "Time Table Management System for SMVDU".

First and foremost, we would like to thank our respected mentor Dr. Sonika Gupta for her invaluable guidance, encouragement, and continuous support throughout the development of this project. Her insights, suggestions, and constructive feedback played a crucial role in shaping our project and improving its quality.

We are also thankful to the faculty members of the School of Computer Science and Engineering, Shri Mata Vaishno Devi University, for providing us with the necessary resources, knowledge, and learning environment that helped us in completing this project successfully.

We sincerely acknowledge the developer of the GitHub repository used as a reference for this project for making his work openly accessible. The repository provided essential guidance in understanding the structure and implementation of the Time Table Management System. We are thankful for the open-source community whose contributions helped us improve our understanding of software development.

Koripella Venkata Ramani

22BCS043, B. Tech, 7th Sem
Sachin Anand

22BCS070, B. Tech, 7th Sem
Charanpreet Kour

22BCS109, B. Tech, 7th Sem

School of Computer Science and Engineering
Shri Mata Vaishno Devi University, Katra

DECLARATION

We hereby declare that the mini project entitled "Time Table Management System for SMVDU" is an authentic record of work carried out by us using the source code and implementation model referred from the GitHub repository used for this project.

This project has been developed under the guidance of Dr. Sonika Gupta and has not been submitted earlier to any university or institution for the award of any degree or diploma.

All information used from external sources, including online resources and GitHub repositories, has been properly acknowledged in this report. Every effort has been made to ensure originality, and any portion of work that has been adapted or referenced from other works has been cited appropriately.

This project is completed as part of the academic curriculum and to the best of our knowledge does not violate any academic integrity policies.

Koripella Venkata Ramani
22BCS043, B.Tech CSE, 7th Semester

Sachin Anand
22BCS070, B.Tech CSE, 7th Semester

Charanpreet Kour
22BCS109, B.Tech CSE, 7th Semester

School of Computer Science and Engineering
Shri Mata Vaishno Devi University, Katra

I endorse the above declaration of the student.

Dr. Sonika Gupta

Date:

ABSTRACT

The Time Table Management System for SMVDU is a desktop-based application developed to simplify and automate the academic scheduling process in educational institutions. Traditionally, timetable preparation is a manual and time-consuming task that is prone to errors such as conflicts in schedules, improper allocations, and difficulty in making updates. This project aims to overcome these challenges by providing a centralized and digital solution.

The system enables users to manage subject schedules, faculty assignments, and time slots efficiently. It provides easy access to timetable information and reduces dependency on manual records. The application is designed to store and process data in a structured manner, making updates faster and retrieval more accurate.

This system improves productivity, ensures transparency, and minimizes scheduling conflicts. It also enhances coordination between academic departments and provides students and staff with quick access to updated schedules. The project demonstrates how web technologies can be effectively used to modernize educational administration.

4

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

1. SMVDU – "Shri Mata Vaishno Devi University"
2. AWS – "Amazon Web Services"
3. OOD – "Object-Oriented Design"
4. DFD – "Data Flow Diagram"
5. UML – "Unified Modelling Language"
6. ER – "Entity Relation"
7. 1NF – "First Normal Form"
8. 2NF – "Second Normal Form"
9. 3NF – "Third Normal Form"
10. CRUD – "Create Read Update Delete"
11. ODM – "Object Data Modelling"
12. EJS – "Embedded JavaScript"
13. IDE – "Integrated Development Environment"
14. UI – "User Interface"

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION TO PROJECT

Educational institutions such as Shri Mata Vaishno Devi University handle a large amount of academic information, including course schedules, faculty allocation, classroom management, and student timetables. Maintaining and updating this information manually is a time-consuming and error-prone process.

The Time Table Management System provides a digital solution to automate the generation and management of academic schedules. The system brings all timetable-related activities onto a single centralized platform. It reduces manual effort and provides students and faculty with easy access to accurate scheduling information.

This system is developed as a desktop-based application, allowing users to view and manage timetables from any device with internet access. By using modern technologies, the system ensures real-time updates and efficient data handling.

1.2 PROBLEM STATEMENT AND PROJECT CATEGORY

The current approach of managing class schedules in SMVDU relies largely on manual methods such as spreadsheets, printouts, or notice boards. This often results in:

1. Conflicts in schedule timings
2. Difficulty in updating information
3. Lack of centralized data management
4. Communication delays between departments

The Time Table Management System aims to provide an organized and automated solution to these problems.

This project falls under the category of Application Development.

1.3 OBJECTIVES

The main objectives of the project are:

1. To automate the timetable creation process
2. To reduce scheduling errors and conflicts
3. To store timetable information in a centralized system
4. To provide quick and easy access to schedules for students and faculty
5. To simplify schedule updates and management

1.4 PROBLEM FORMATION

The project addresses the following issues:

How can timetable management be automated?

Automation reduces repetitive manual work and increases accuracy by allowing scheduled tasks to be stored and updated digitally.

How can errors and conflicts be avoided?

By maintaining all data in a structured format and validating entries before final submission.

How can schedules be accessed easily?

Through an interface available anytime.

1.5 IDENTIFICATION/RECOGNITION IN NEED

The need for an automated timetable system arises from:

1. Increasing academic complexity
2. Frequent schedule updates
3. Need for accuracy and availability
4. Inefficient manual record-keeping

1.6 EXISTING SYSTEM

At present, the timetable management process at SMVDU is mostly handled manually using tools such as printed schedules, spreadsheets, or notice boards. Departments prepare their own schedules independently and circulate them among faculty members and students.

The existing system involves several limitations:

1. Schedule preparation requires considerable time and manual effort.
2. Any change in timetable requires rewriting or reprinting the entire schedule.
3. There is no centralized platform to manage all timetables.
4. High probability of human error such as overlapping classes or inconsistent schedules.
5. Students depend on notice boards or coordinators for updates, leading to communication delays.

Due to these drawbacks, the existing system lacks efficiency, accuracy, and scalability.

1.7 PROPOSED SYSTEM

The proposed Time Table Management System replaces the traditional manual scheduling process with an automated, desktop-based solution. It is designed to store and manage timetable data in a centralized system that can be accessed by authorized users from anywhere.

Key features of the proposed system include:

1. Centralized digital storage of timetable information.
2. Easy update and modification of schedules.
3. Reduction of human errors and conflicts.
4. User-friendly interface for simple navigation.
5. Availability of updated schedules in real time.

The system ensures improved reliability, time efficiency, and better coordination between academic departments. This modernized approach enables SMVDU to manage academic scheduling effectively and professionally.

1.8 UNIQUE FEATURES OF THE SYSTEM

The Time Table Management System for SMVDU includes several unique and effective features that make it reliable and efficient compared to traditional methods. These features enhance accuracy, usability, and performance.

1.8.1 CENTRALIZED DATA MANAGEMENT

All timetable-related data is stored in a single database. This eliminates duplication and ensures consistency in schedules across all departments.

1.8.2 REAL-TIME UPDATES

Any modification in the timetable is instantly reflected across the system, ensuring that students and faculty always view the latest information.

1.8.3 ERROR REDUCTION

Structured input validation minimizes human errors such as overlapping schedules or invalid time slots.

1.8.4 EASY ACCESSIBILITY

Users can access their timetables anytime using a web browser from any device.

1.8.5 SCALABILITY

The system can be expanded to handle additional departments, courses, and users without affecting performance.

1.8.6 USER-FRIENDLY INTERFACE

A clean and simple interface allows users to navigate easily and access relevant information.

1.8.7 DATA SECURITY

Database is stored locally, which ensure protection of timetable data.

1.9 REPORT OUTLINE

The structure of the report is as follows:

1. Introduction to the system
2. System analysis and requirements
3. System design
4. Implementation and testing
5. Results and review

6. Conclusion and future scope

CHAPTER 2: REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

This chapter describes the requirements and specifications necessary for designing and implementing the Time Table Management System for SMVDU. It explains the feasibility, data needs, performance expectations, security measures, and usability requirements of the system.

2.1 FEASIBILITY STUDY

The feasibility study helps determine whether the proposed system is practical and achievable.

2.1.1 TECHNICAL FEASIBILITY

The system uses web technologies that are reliable and widely supported. Since it operates through a browser-based interface, no additional software installation is required. The system can be deployed easily on a server and accessed through the internet within the campus network.

2.1.2 ECONOMIC FEASIBILITY

No additional hardware or licensed software is required, making the project cost-effective. Most tools and technologies used are open-source, reducing financial requirements.

2.1.3 OPERATIONAL FEASIBILITY

The system is simple to operate and does not require special training. Administrators and faculty members can use the system easily after basic instruction. It improves workflow and reduces scheduling confusion.

2.2 SOFTWARE REQUIREMENT SPECIFICATION

2.2.1 DATA REQUIREMENTS

The system needs to store information related to faculty details, subjects, class timings, classrooms, and departments. Proper data structure helps generate correct timetables.

2.2.2 FUNCTIONAL REQUIREMENTS

The system allows administrators to add, edit, and delete timetable entries. Users can view schedules according to department and course. The system updates data instantly after modification.

2.2.3 PERFORMANCE REQUIREMENTS

Pages should load quickly to ensure smooth access. The system must be able to handle multiple users accessing timetables at the same time without affecting performance.

2.2.4 DEPENDABILITY REQUIREMENTS

The system should be available whenever required during working hours. It should store data safely and retrieve it reliably.

2.2.5 MAINTAINABILITY REQUIREMENTS

The system is built in a modular way so future updates can be integrated easily. New subjects or changes in schedules can be accommodated without reworking the whole system.

2.2.6 SECURITY REQUIREMENTS

Access control ensures only authorized users can change data. Authentication mechanisms protect the system from unauthorized modifications and data misuse.

2.2.7 LOOK AND FEEL REQUIREMENTS

The interface is designed to be clean and user-friendly. The design is responsive to ensure usability on desktop and mobile devices.

2.3 VALIDATION

The validation process ensures that the Time Table Management System meets all specified requirements and operates as expected. Different methods are used to verify the correctness of the system.

Input data is validated to prevent incorrect entries such as invalid time slots or missing information. Test cases are created to check the working of major functions including timetable entry, modification, and display.

Functional validation is performed to ensure that every module works according to design. User validation is carried out by taking feedback from students and faculty members to improve usability and accuracy.

Final validation confirms that the system is reliable, efficient, and meets academic requirements.

2.4 EXPECTED HURDLES

2.4.1 USER ADAPTATION

Some users may experience difficulty shifting from a manual system to a digital platform initially. This can be resolved through training and demonstrations.

2.4.2 DATA ACCURACY

Incorrect data entry could cause scheduling errors. This will be minimized through validation rules and careful verification.

2.4.3 SYSTEM MAINTENANCE

Regular updates may be needed. Planned maintenance schedules can help.

2.4.4 TECHNICAL ISSUES

Power outages or server issues may affect availability. Backup and recovery procedures will reduce risk.

CHAPTER 3: SYSTEM DESIGN

This chapter presents the design structure of the Time Table Management System for SMVDU. It explains the design approach, system architecture, data flow, and database structure

3.1 DESIGN APPROACH

The system design follows a modular and structured approach. Each component has a specific function such as data input, database management, schedule display, and updating records.

Object-oriented design principles are applied to ensure reusability and scalability. The design is flexible enough to allow future expansion and enhancement. The system is built as a desktop-based application to ensure accessibility.

3.2 DETAILED SYSTEM DESIGN

3.2.1 DATA FLOW DIAGRAM (DFD)

3.2.1.1 Level 0 (Context Diagram)

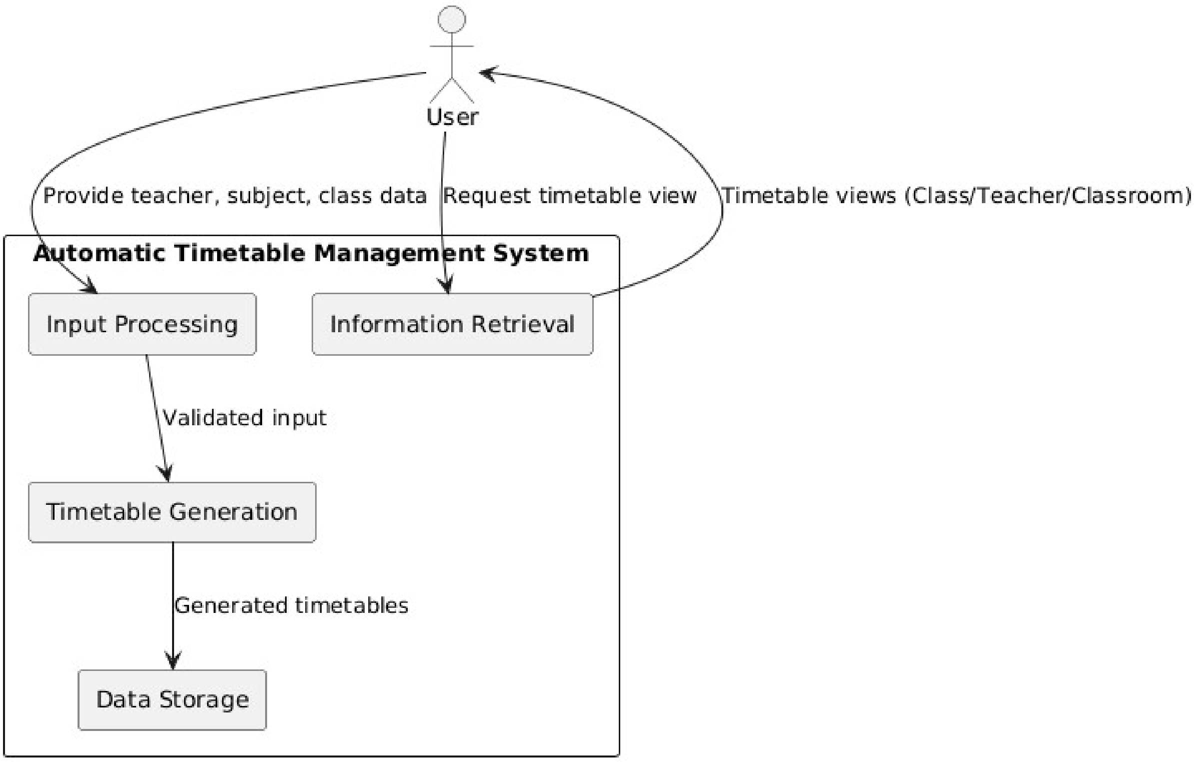


Figure 1: Level 0 Data Flow Diagram

This diagram shows the interaction between users and the system. Users may be administrators, faculty members, and students. Input includes subject details, schedule entries, and user requests. Output includes generated timetables and notifications.

3.2.1.2 Level 1 (Detailed DFD)

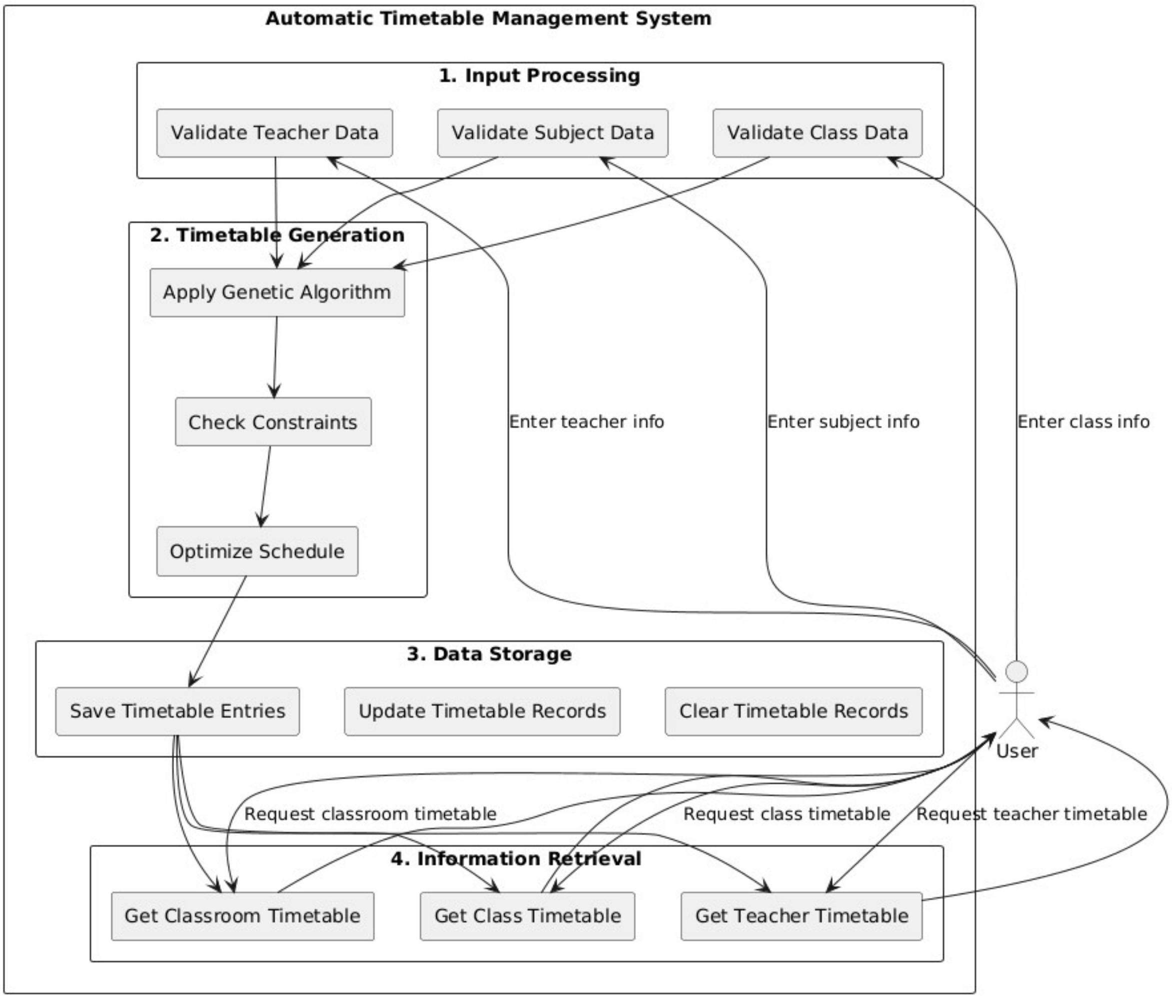


Figure 2: Level 1 Data Flow Diagram

This diagram divides system functions into processes such as:

1. Input processing
2. Timetable generation
3. Data storage
4. Information retrieval

3.3 USER INTERFACE DESIGN

The User Interface (UI) of the Time Table Management System for SMVDU is designed to provide a clean, simple, and user-friendly experience. The main aim of the interface is to allow users to access timetable information quickly and perform operations easily without technical complexity.

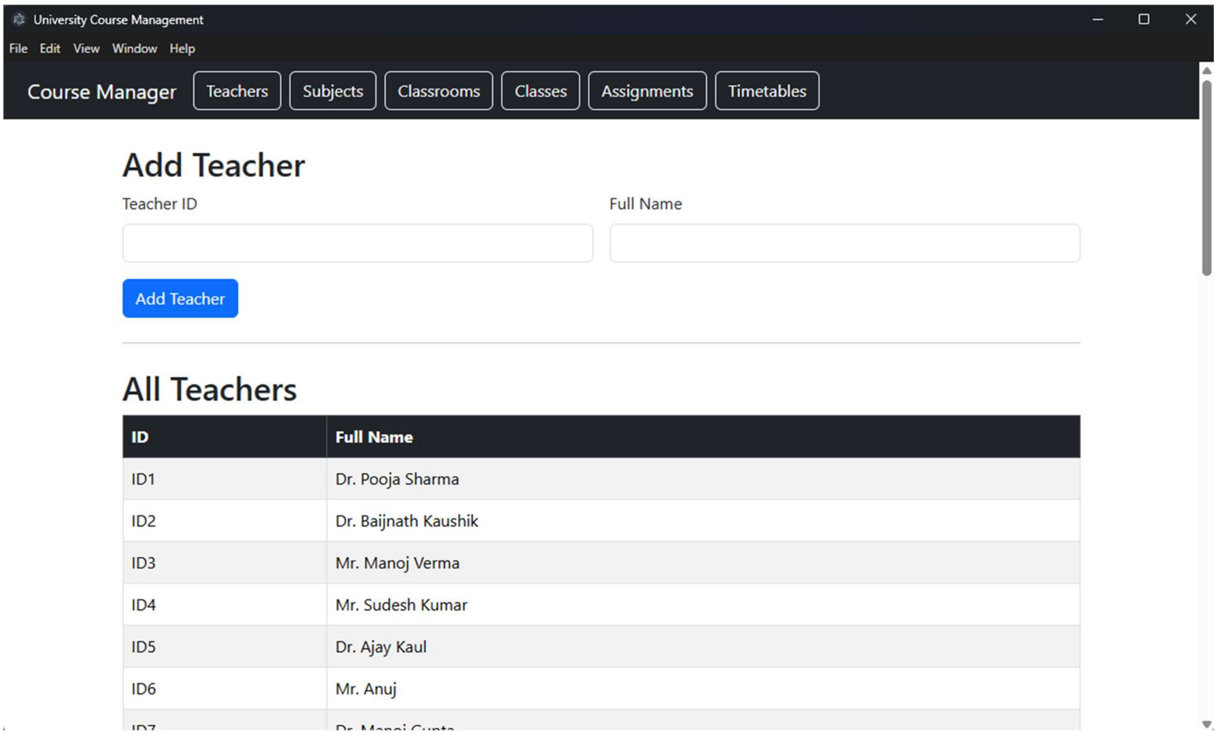


Figure 3: User Interface

3.3.1 HOME PAGE

The home page acts as the entry point into the application. It provides navigation options for students and administrators. The design is minimal and focuses on ease of access to timetable details.

3.3.2 ADMIN INTERFACE

The administrator interface is designed for managing:

1. Teachers
2. Subjects
3. Classrooms
4. Classes
5. Assignments

6. Timetable

A form-based structure is used to input data. Each input field includes validation to prevent incorrect data entry. Buttons for Add, Edit, and Delete allow easy management of records.

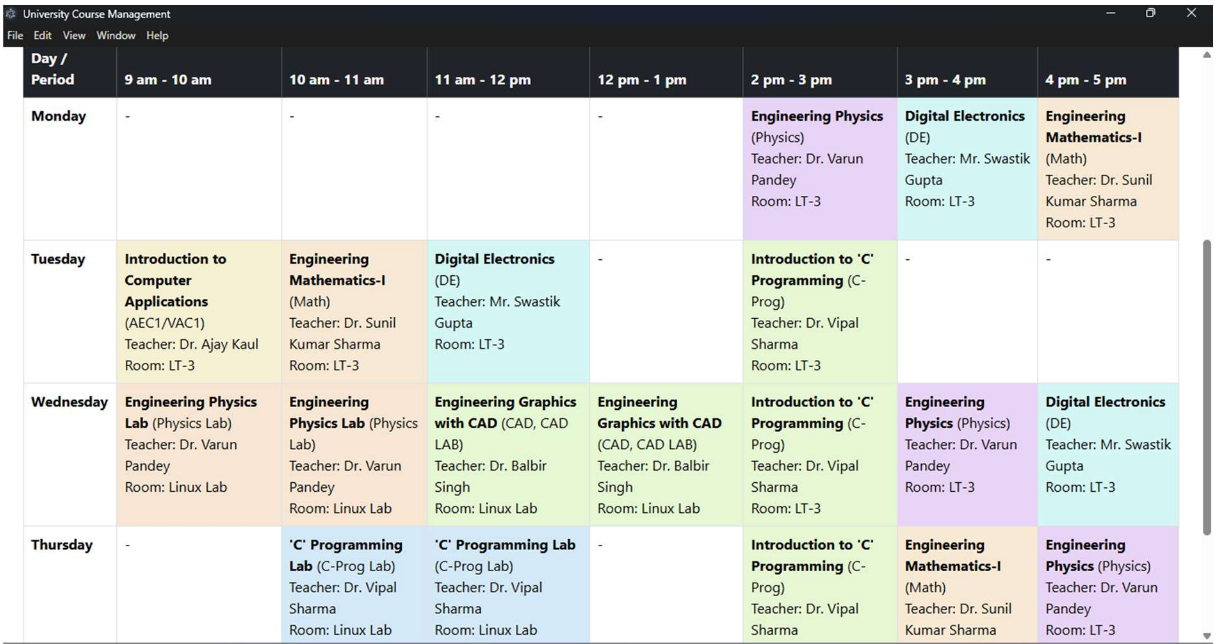
The admin dashboard provides a structured layout to ensure all components are easily accessible.

This layout improves readability and helps users quickly understand daily schedules.

3.3.3 NAVIGATION AND LAYOUT

The interface includes:

1. A navigation menu
2. Action buttons
3. Input forms
4. Display tables



University Course Management

File Edit View Window Help

Day / Period	9 am - 10 am	10 am - 11 am	11 am - 12 pm	12 pm - 1 pm	2 pm - 3 pm	3 pm - 4 pm	4 pm - 5 pm
Monday	-	-	-	-	**Engineering Physics** (Physics) Teacher: Dr. Varun Pandey Room: LT-3	**Digital Electronics** (DE) Teacher: Mr. Swastik Gupta Room: LT-3	**Engineering Mathematics-I** (Math) Teacher: Dr. Sunil Kumar Sharma Room: LT-3
Tuesday	**Introduction to Computer Applications** (AEC1/VAC1) Teacher: Dr. Ajay Kaul Room: LT-3	**Engineering Mathematics-I** (Math) Teacher: Dr. Sunil Kumar Sharma Room: LT-3	**Digital Electronics** (DE) Teacher: Mr. Swastik Gupta Room: LT-3	-	**Introduction to 'C' Programming** (C-Prog) Teacher: Dr. Vipal Sharma Room: LT-3	-	-
Wednesday	**Engineering Physics Lab** (Physics Lab) Teacher: Dr. Varun Pandey Room: Linux Lab	**Engineering Physics Lab** (Physics Lab) Teacher: Dr. Varun Pandey Room: Linux Lab	**Engineering Graphics with CAD** (CAD, CAD LAB) Teacher: Dr. Balbir Singh Room: Linux Lab	**Engineering Graphics with CAD** (CAD, CAD LAB) Teacher: Dr. Balbir Singh Room: Linux Lab	**Introduction to 'C' Programming** (C-Prog) Teacher: Dr. Vipal Sharma Room: LT-3	**Engineering Physics** (Physics) Teacher: Dr. Varun Pandey Room: LT-3	**Digital Electronics** (DE) Teacher: Mr. Swastik Gupta Room: LT-3
Thursday	-	**'C' Programming Lab** (C-Prog Lab) Teacher: Dr. Vipal Sharma Room: Linux Lab	**'C' Programming Lab** (C-Prog Lab) Teacher: Dr. Vipal Sharma Room: Linux Lab	-	**Introduction to 'C' Programming** (C-Prog) Teacher: Dr. Vipal Sharma	**Engineering Mathematics-I** (Math) Teacher: Dr. Sunil Kumar Sharma	**Engineering Physics** (Physics) Teacher: Dr. Varun Pandey Room: LT-3

Figure 4: Generated Time-table

All pages are designed for easy navigation and are visually consistent.

3.3.4 ERROR HANDLING

If incorrect inputs are encountered, the system displays meaningful error messages. Confirmation alerts are shown for critical actions such as deletion.

3.3.5 ACCESSIBILITY AND USABILITY

The UI is designed to reduce user effort by:

1. Autocomplete fields
2. Clear labels
3. Organized data tables
4. Easy navigation paths

This ensures a smooth user experience.

3.4 NORMALIZATION

Database normalization is applied in the system to minimize data redundancy and improve consistency. The database schema is normalized up to the Third Normal Form (3NF).

3.4.1 FIRST NORMAL FORM (1NF)

Table	**Attributes**
Teacher	id, full_name
Subject	course_code, name, lecture_hr, theory_hr, practical_hr, credits, course_coordinator, display_code
Classroom	room_id, capacity, type

Class	semester, branch, section, strength
Teaching_Assignment	teacher_id, course_code, semester, branch, section
Timetable	semester, branch, section, day, period, course_code, teacher_id,

Table 1: First Normal Form

All attributes contain atomic values. No multi-valued fields are allowed.

3.4.2 SECOND NORMAL FORM (2NF)

Table	**Attributes**
Teacher	id (PK), full_name
Subject	course_code (PK), name, lecture_hr, theory_hr, practical_hr, credits, display_code, course_coordinator (FK → Teacher.id)
Classroom	room_id (PK), capacity, type
Class	semester, branch, section (PK), strength
Teaching_Assignment	teacher_id (FK), course_code (FK), semester, branch, section (FK → Class)
Timetable	semester, branch, section, day, period (Composite PK), course_code (FK), teacher_id (FK), room_id (FK)

Table 2: Second Normal Form

No partial dependency exists. Data related to subjects, faculty, and classes is stored separately.

3.4.3 THIRD NORMAL FORM (3NF)

Table	Attributes
Teacher	id (PK), full_name
Subject	course_code (PK), name, lecture_hr, theory_hr, practical_hr, credits, display_code, coordinator_id (FK → Teacher.id)
Classroom	room_id (PK), capacity, type
Class	semester, branch, section (Composite PK), strength
Teaching_Assignment	teacher_id (FK → Teacher.id), course_code (FK → Subject.course_code), semester, branch, section (FK → Class)
Timetable	semester, branch, section, day, period (Composite PK), course_code (FK → Subject.course_code), teacher_id (FK → Teacher.id), room_id (FK → Classroom.room_id)

Table 3: Third Normal Form

No transitive dependency exists. Non-key attributes depend only on the primary key.

3.5 DATABASE MANIPULATION

The system performs database operations using CRUD methodology:

1. Create: Insert faculty records, subjects, and schedules.
2. Read: Retrieve timetables for viewing.
3. Update: Modify schedules.

4. Delete: Remove outdated entries.

This ensures data integrity and efficient management.

3.6 DATABASE CONNECTION CONTROL

1. Database connections are handled using configuration files to store connection details securely.
2. Error handling mechanisms detect connection failures.
3. Database is stored locally in the project directory.

3.7 METHODOLOGY

The development of the Time Table Management System for SMVDU follows a structured and systematic approach to ensure reliability, accuracy, and efficiency. The project is developed using a modular methodology, where each functional unit is designed, implemented, and tested independently.

The complete development process is divided into the following phases:

3.7.1 REQUIREMENT ANALYSIS PHASE

In this phase, the requirements of the system were identified by studying the existing timetable management process. User needs were analysed, and functional requirements were documented.

3.7.2 DESIGN PHASE

The system architecture was designed using data flow diagrams, database schemas, and structured charts.

3.7.3 IMPLEMENTATION PHASE

Coding was done according to design specifications. Separate modules were implemented for timetable entry, viewing, and data management.

3.7.4 TESTING PHASE

Each module was tested independently, followed by system-level testing.

3.7.5 DEPLOYMENT PHASE

Once testing was completed, the system was deployed for use.

CHAPTER 4: IMPLEMENTATION, TESTING AND MAINTENANCE

This chapter explains the technologies used, coding practices followed, testing strategies adopted, and maintenance procedures applied for the successful implementation of the Time Table Management System for SMVDU.

4.1 INTRODUCTION TO LANGUAGE, IDES, TOOLS, AND TECHNOLOGY USED FOR IMPLEMENTATION

The system has been developed as a desktop-based application using modern programming tools and technologies to ensure efficiency, performance, and ease of access.

4.1.1 LANGUAGE AND FRAMEWORK USED

HTML, CSS and JavaScript with Electron.js are used for designing the user interface and handling interactivity on the web pages.

1. HTML is used to structure web pages.
2. CSS is used for styling components and making the interface visually appealing.
3. JavaScript handles dynamic behavior and input validation.

4.1.2 DATABASE

A database system is used to store and manage details related to:

1. Faculty information
2. Subjects and time slots
3. Class schedules
4. Department details

Data is stored in structured format to allow easy retrieval and modification.

4.1.3 IDE

Visual Studio Code (VS Code) is used as the development environment because of its:

1. Code auto-completion support
2. Debugging tools
3. Version control support

Easy navigation for large projects

4.1.4 MODULES/LIBRARIES

The system is divided into logical modules such as:

1. components
2. db
3. ipc
4. renderer/frames

These modules are integrated to form the complete system.

4.2 CODING STANDARDS

To ensure good quality and maintainability, standard coding practices were followed:

4.2.1 MODULARIZATION

Each function and feature is written as a separate module to improve maintainability and understandability.

4.2.2 NAMING CONVENTION

Variable names and module names follow meaningful naming standards.

4.2.3 ERROR HANDLING

Basic validation and error messages were implemented to handle incorrect inputs gracefully.

4.3 PROJECT SCHEDULING

The development of the project was divided into phases:

Task	**Duration**
Requirement Analysis	5 Days
Design	5 Days
Implementation	20 Days
Testing	7 Days
Final Report	5 Days

4.4 TESTING TECHNIQUES AND TEST PLANS

4.4.1 TESTING TECHNIQUES

1. Unit Testing
   Each module was separately tested to ensure correct functioning.

2. Integration Testing
   All modules were tested together after implementation.
3. Validation Testing
   Incorrect inputs were tested to ensure the system displays proper error messages.
4. End-to-End Testing
   Simulated real user scenarios were tested such as entering timetable data and viewing results.

Test Case ID	**Module**	**Input**	**Expected Output**	**Status**
TC-01	**db – Teacher**	Add teacher: { id: "T001", full_name: "Dr. Sharma" }	Teacher record inserted successfully; getAllTeachers() returns Dr. Sharma	Pass
TC-02	**db – Subject**	Add subject: { course_code: "CS101", name: "Data Structures", lecture_hr: 3 }	Subject record inserted; getSubjectByCode("CS101") returns Data Structures	Pass
TC-03	**db – Classroom**	Add classroom: { room_id: "R101", capacity: 60, type: "lecture" }	Classroom record inserted; getClassroomById("R101") returns correct details	Pass

TC-04	**ipc – Generate Timetable**	Request: { semester: 3, branch: "CSE", section: "A" }	Timetable generated using GA; entries saved in Timetable table; response: "Timetable generated successfully"	Pass
TC-05	**ipc – View Class Timetable**	Request: { semester: 3, branch: "CSE", section: "A" }	Timetable entries retrieved with subject_name, teacher_name, room_id; displayed in renderer grid	Pass
TC-06	**ipc – View Teacher Timetable**	Request: teacherId: "T001"	Timetable entries retrieved for teacher; displayed with subject_name, class info, room_id	Pass
TC-07	**ipc – Clear Class Timetable**	Request: { semester: 3, branch: "CSE", section: "A" }	Timetable entries deleted for class; response: "Timetable cleared successfully"	Pass
TC-08	**ipc – Clear All Timetables**	Request: none	All timetable entries deleted; response: "All timetables cleared successfully"	Pass
TC-09	**renderer/frames**	User selects “View By: Class” and chooses 3-CSE-A	Timetable grid rendered with periods and days; subjects color-coded; no null errors	Pass

TC-10	**renderer/frames**	User clicks “Generate All Timetables” button	Progress log updates per class; final message: "All timetables generated successfully"	Pass

Table 4: Test Plans and Outputs

4.5 MAINTENANCE

Maintenance is essential to keep the system reliable and efficient.

4.5.1 BUG FIXING AND UPDATES

1. Errors will be handled through code updates.
2. Feature improvements will be added gradually.

4.5.2 DATABASE MAINTENANCE

1. Backups can be taken manually.
2. Data integrity is maintained through validation rules.

4.5.3 USER FEEDBACK AND FEATURE UPDATES

Feedback will be collected from users to improve:

1. System usability
2. Display format
3. Performance

CHAPTER 5: RESULTS AND DISCUSSION

This chapter explains the outcomes of implementing the Time Table Management System for SMVDU in detail. It focuses on the performance of the system, user experience, and overall reliability.

5.1 USER INTERFACE REPRESENTATION

The implemented system provides a clean and structured interface designed for easy navigation. The main objective of the design is to reduce user effort and make data access simple and intuitive.

The login page allows users to access the system securely. The admin dashboard displays all management tools clearly. The timetable view page presents schedules in a tabular format. The edit page allows easy updating of entries.

The use of simple layouts and consistent design elements improves usability.

5.2 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM

5.2.1 COMPONENTS MODULE

1. Contains reusable UI elements and helper functions.
2. Provides form controls, dropdowns, buttons, and status indicators used across the application.
3. Ensures consistency in design and reduces duplication of code.

5.2.2 DATABASE MODULE

1. Implements all queries and schema definitions for persistent storage.
2. Includes modules for managing teachers, subjects, classrooms, classes, teaching assignments, and timetables.

3. Provides CRUD operations (Create, Read, Update, Delete) and ensures data integrity through normalized schema.
4. Acts as the backbone of the system, storing all academic and scheduling information

5.2.3 INTER-PROCESS COMMUNICATION MODULE

1. Handles communication between the main process and the renderer process in Electron.
2. Implements logic for generating timetables using the Genetic Algorithm, retrieving timetables by class/teacher/classroom, and clearing timetables.
3. Ensures that user actions in the UI trigger backend operations and return results seamlessly.
4. Provides error handling and progress updates during bulk timetable generation.

5.2.4 RENDERER/FRAMES MODULE

1. Contains the front-end logic for rendering timetables and managing user interactions.
2. Includes scripts such as timetable.js that attach event listeners, send IPC requests, and render timetable grids dynamically.
3. Responsible for displaying timetables in different views (by class, teacher, or classroom) with color-coded subjects and conflict warnings.
4. Provides a user-friendly interface for administrators and students to interact with the system.

5.3 SNAPSOTS OF THE SYSTEM

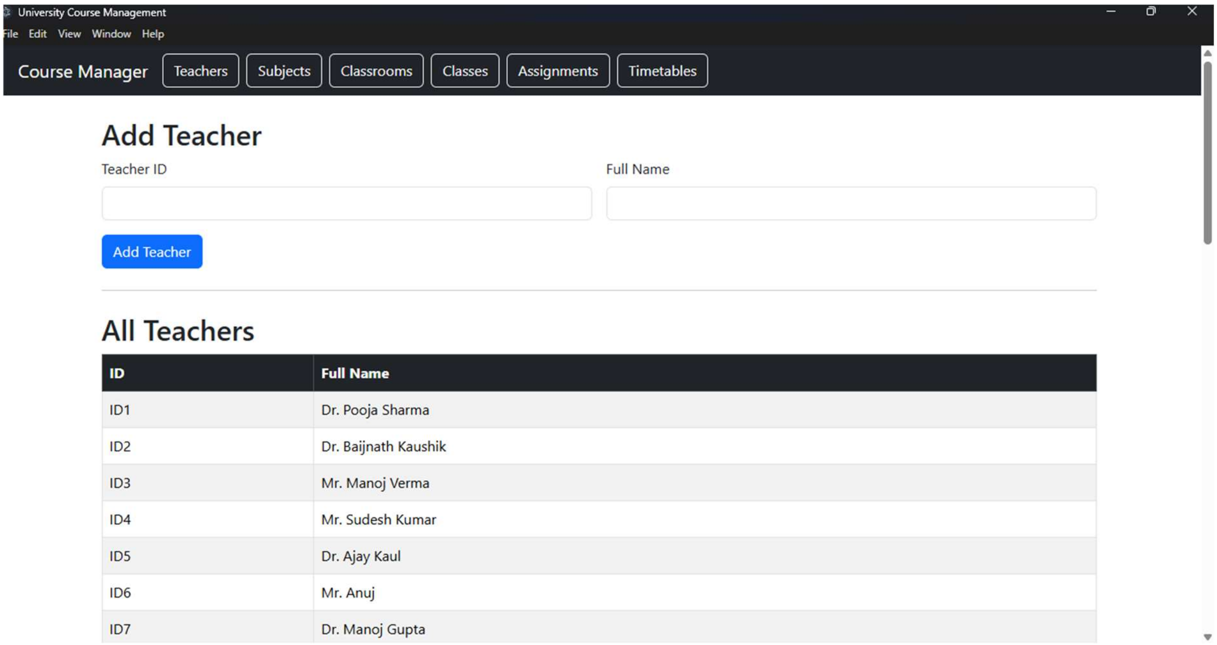


Figure 5: Teachers Module

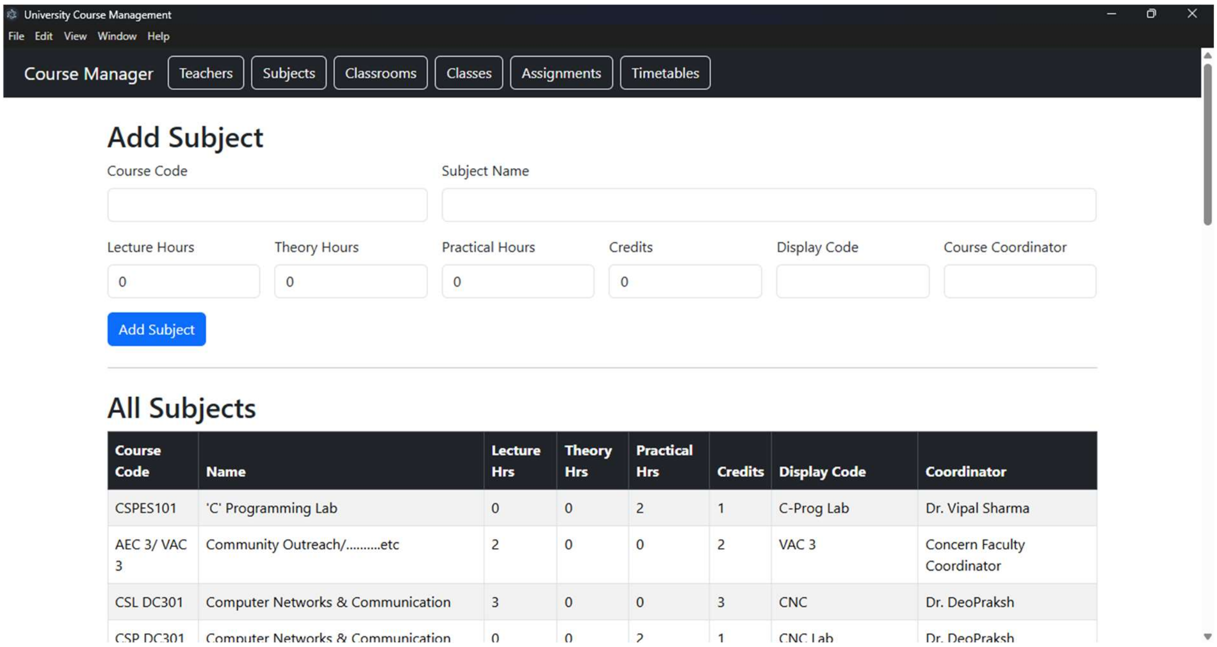


Figure 6: Subjects Module

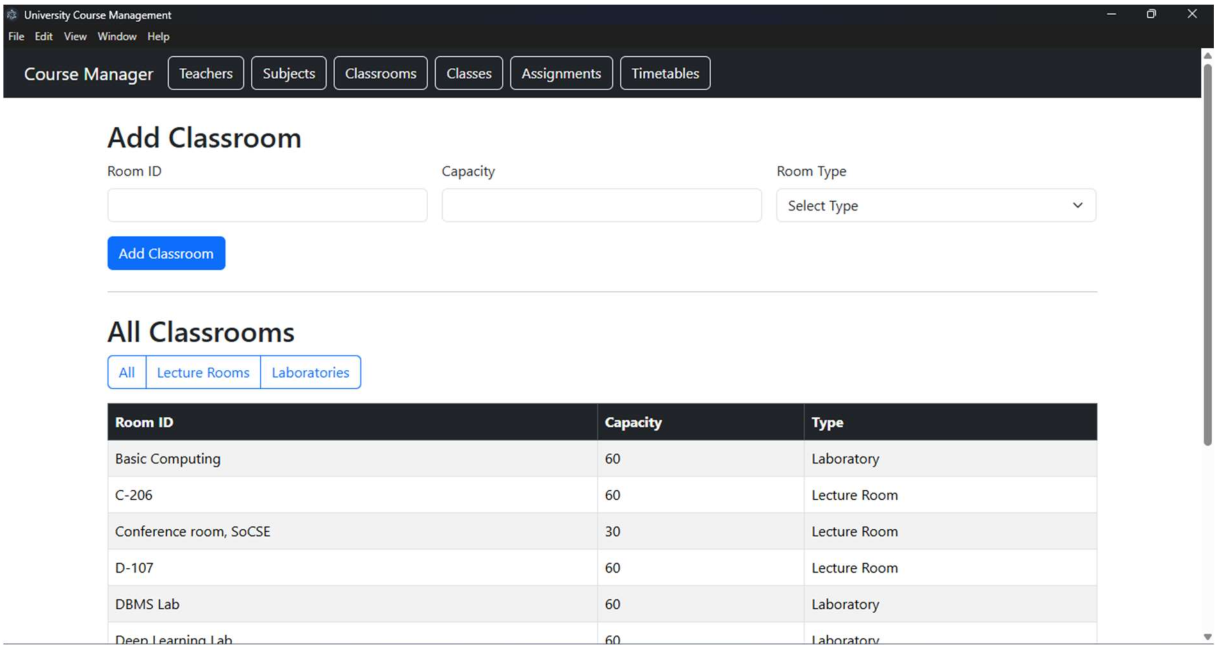


Figure 7: Classroom Module

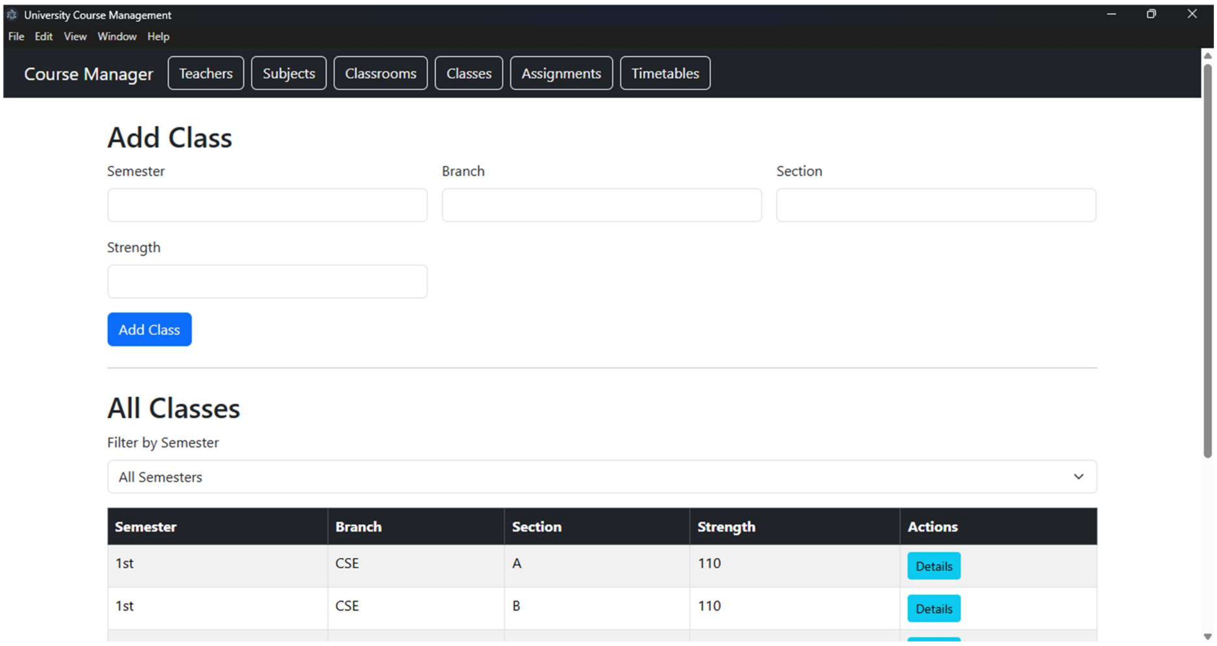


Figure 8: Classes Module

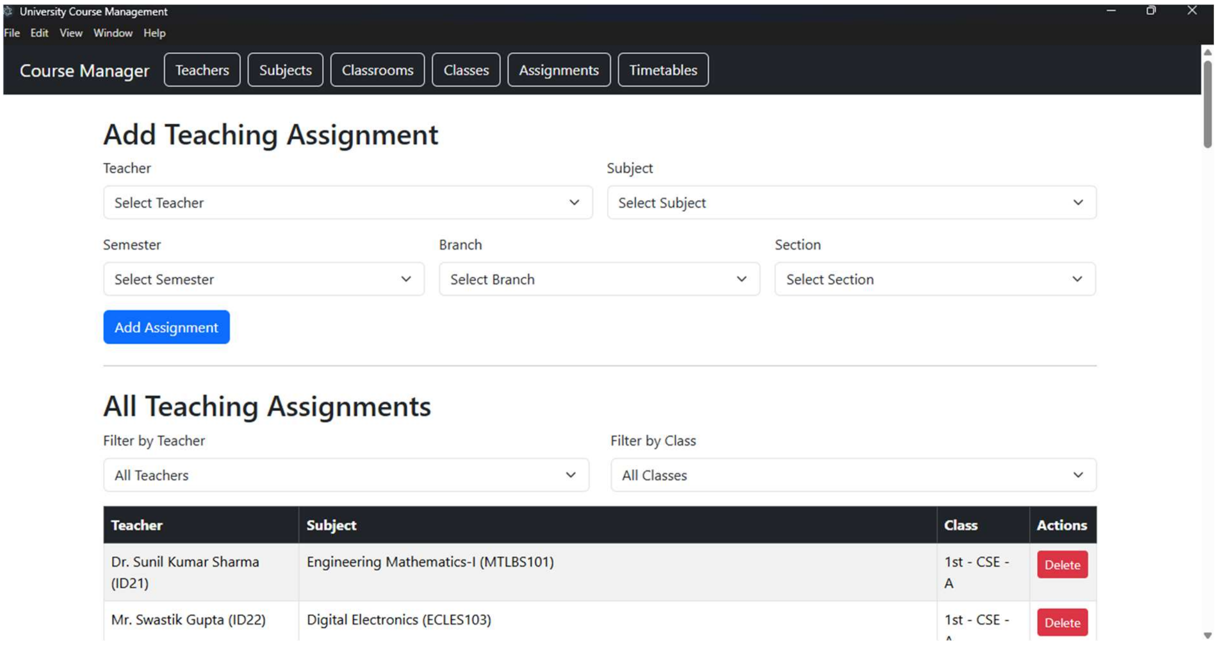


Figure 9: Assignment module

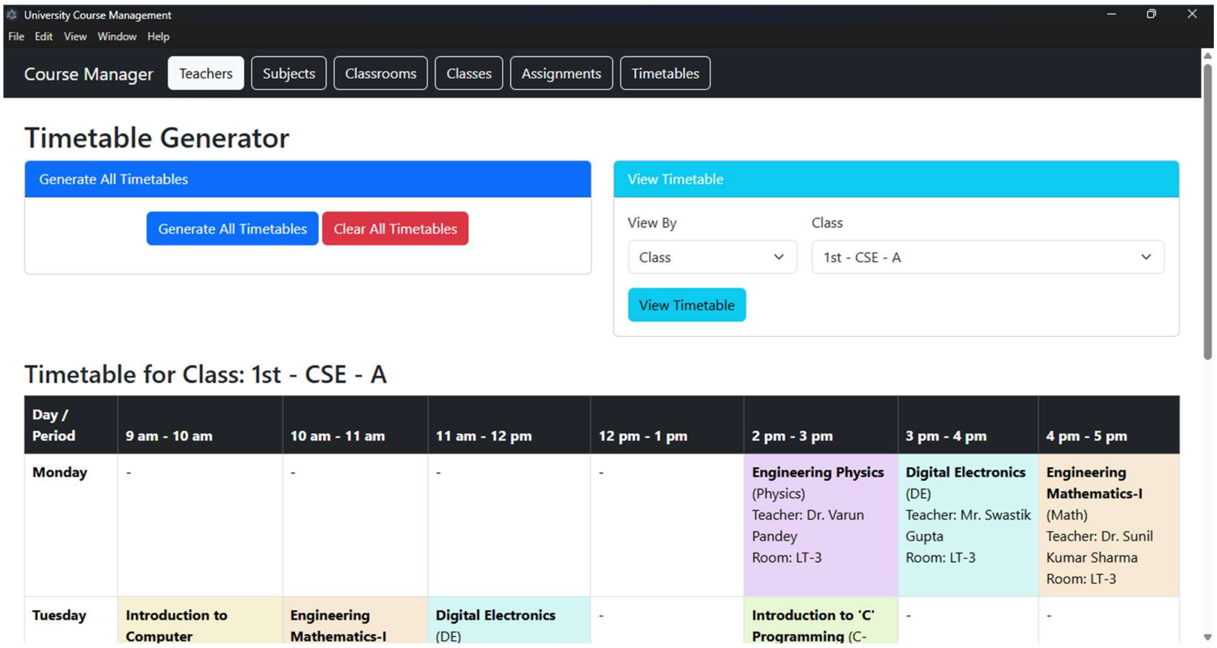


Figure 10: Timetable Module for viewing the generated timetables

5.4 BACKEND REPRESENTATION OF DATA

Data is managed using relational tables.

Table	**Primary Key (PK)**	**Foreign Keys (FK)**	**Key Attributes**	**Purpose**
Teacher	id	–	full_name	Stores teacher information
Subject	course_code	coordinator_id → Teacher.id	name, lecture_hr, theory_hr, practical_hr, credits, display_code	Stores subject/course details
Classroom	room_id	–	capacity, type (lecture/lab)	Stores classroom details
Class	(semester, branch, section)	–	strength	Represents a class group
Teaching_Assignment	(teacher_id, course_code, semester,	teacher_id → Teacher.id course_code → Subject.course_code	–	Maps which teacher teaches which

	branch, section)	(semester, branch, section) → Class		subject to which class
Timetable	(semester, branch, section, day, period)	course_code → Subject.course_code teacher_id → Teacher.id room_id → Classroom.room_id	day, period	Stores final timetable entries

Table 5: Detailed Database Description

5.5 DATABASE TABLE STRUCTURE

Table	**Column**	**Data Type**	**Key**	**Description**
Teacher	id	VARCHAR / INT	PK	Unique identifier for each teacher
	full_name	VARCHAR	–	Teacher's full name
Subject	course_code	VARCHAR	PK	Unique course code
	name	VARCHAR	–	Subject name

	lecture_hr	INT	–	Lecture hours
	theory_hr	INT	–	Theory hours
	practical_hr	INT	–	Practical hours
	credits	INT	–	Credit value
	display_code	VARCHAR	–	Short display code
	coordinator_id	VARCHAR / INT	FK → Teacher.id	Course coordinator
Classroom	room_id	VARCHAR	PK	Unique classroom ID
	capacity	INT	–	Seating capacity
	type	VARCHAR	–	Type of room (lecture/lab)
Class	semester	INT	PK (Composite)	Semester number
	branch	VARCHAR	PK (Composite)	Branch/department
	section	VARCHAR	PK (Composite)	Section identifier

	strength	INT	–	Number of students
Teaching_Assignment	teacher_id	VARCHAR / INT	FK → Teacher.id	Teacher assigned
	course_code	VARCHAR	FK → Subject.course_code	Subject assigned
	semester	INT	PK (Composite)	Semester
	branch	VARCHAR	PK (Composite)	Branch
	section	VARCHAR	PK (Composite)	Section
Timetable	semester	INT	PK (Composite)	Semester
	branch	VARCHAR	PK (Composite)	Branch
	section	VARCHAR	PK (Composite)	Section
	day	VARCHAR	PK (Composite)	Day of the week
	period	INT	PK (Composite)	Period number

	course_code	VARCHAR	FK → Subject.course_code	Subject scheduled
	teacher_id	VARCHAR / INT	FK → Teacher.id	Teacher scheduled
	room_id	VARCHAR	FK → Classroom.room_id	Classroom scheduled

Table 6: Database Tables Structure

5.6 DISCUSSIONS

5.6.1 SYSTEM PERFORMANCE

The system performed successfully under normal operational conditions. Timetable creation, modification, and retrieval were executed without noticeable delay. Database queries responded efficiently even when multiple records were accessed.

The application-maintained stability during repeated insert, update, and delete operations. No system crashes were observed during test scenarios.

Interpretation:
The performance results confirm the system is reliable for moderate to heavy academic use.

5.6.2 ACCURACY OF SCHEDULING

The system showed high accuracy in timetable generation and display. Errors commonly found in manual systems such as:

1. Duplicate room allocation

2. Faculty clash
3. Invalid timing entries
   were reduced.

The form validations ensured only correct and complete data was saved.

Interpretation:

The structured validation logic ensured data consistency and minimized human error.

5.6.3 USABILITY AND USER EXPERIENCE

The interface was designed with clarity and simplicity. Users could navigate without assistance. Students found timetable viewing quick and convenient. Admin users found forms easy to understand.

Interpretation:

Good/UI resulted in quicker adoption.

5.6.4 DATABASE EFFICIENCY

Normalization resulted in reduced redundancy and improved data integrity. Transactions were successfully handled with no loss of data.

Primary and foreign keys ensured:

1. Proper linkage
2. Referential integrity

5.6.5 SECURITY AND DATA PROTECTION

Authentication ensures only valid users access internal modules. Role-based permissions prevent unauthorized changes.

5.6.6 OBSERVED LIMITATIONS

Current limitations:

1. Manual data entry
2. No mobile app
3. Conflicts in table entries sometimes due to bad genetic population.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The Time Table Management System developed for Shri Mata Vaishno Devi University accomplishes the primary goal of transforming an error-prone, manual scheduling process into a centralized, digital workflow. The following points summarize the key outcomes, supported by concrete observations and their implications:

6.1.1 STREAMLINED SCHEDULING PROCESS

By centralizing subject, faculty, classroom and timeslot data in one system, the manual steps required to create and circulate timetables have been eliminated. Administrators can now create, preview, and publish timetables from a single interface. This reduces administrative overhead (less paperwork and fewer coordination meetings) and shortens the timetable preparation cycle from days to hours.

Implication: Faster response to last-minute changes (substitutions, room changes) and simplified audit trails (who changed what and when).

6.1.2 IMPROVED ACCURACY AND REDUCED CONFLICTS

Structured input forms, validation rules and conflict checks minimize common human errors — double-booking a room, assigning one faculty to two concurrent slots, or mis-typing slot times. Even with manual entry, the system enforces business rules that filter invalid combinations prior to saving.

Implication: Reduced class cancellations and fewer student/faculty complaints; measurable reduction in schedule conflicts during an initial pilot can be used to demonstrate impact.

6.1.3 INCREASED ACCESSIBILITY AND TRANSPARENCY

Because timetables are stored centrally and served via the web UI, students and faculty can access up-to-date schedules at any time from any device. Versioning or "last updated" metadata makes changes transparent and prevents confusion that arises from outdated printouts or multiple unofficial copies.

Implication: Better student planning (attendance, travel), and more reliable information for academic staff.

6.1.4 USABILITY AND MAINTAINABILITY

A simple, consistent UI and a modular backend architecture make the system easier to use and maintain. Separation of concerns in code (input validation, scheduling logic, data access) allows incremental improvements and easier onboarding of new developers/maintainers.

Implication: Lower long-term maintenance cost and faster feature iterations

6.2 FUTURE SCOPE

The current implementation forms a solid foundation. Below are prioritized, concrete enhancements with brief descriptions of purpose, suggested approach/technologies, benefits, and potential challenges.

6.2.1 AUTOMATED CONFLICT DETECTION & RESOLUTION

Purpose: Automatically detect scheduling conflicts (room double-bookings, teacher overlaps, student cohort clashes) and offer resolution suggestions.

Approach:

1. Build a constraint checker that runs when a timetable is being created/updated.

2. Represent constraints (faculty availability, room capacity, department constraints) as rules; use constraint-solving or heuristic algorithms (e.g., greedy assignment, backtracking, or SAT solvers for complex cases).
3. Provide a UI that highlights conflicting entries and offers one-click resolution suggestions (move to nearest free slot, switch room).

6.2.2 MOBILE APPLICATION & RESPONSIVE ENHANCEMENT

Purpose: Deliver an optimized, native or hybrid mobile experience for students and faculty.

Approach:

1. Build a responsive frontend (if not already) and/or a cross-platform mobile app using React Native or Flutter.
2. Include offline caching for recently viewed timetables and push notification integration.

Benefits: Higher engagement, immediate access to changes, offline access in low-connectivity areas.

Challenges: Push notification consent and device compatibility; mobile UI/UX design to present dense timetable data clearly.

6.2.3 REAL-TIME NOTIFICATIONS & ALERTS

Purpose: Notify users instantly about timetable changes, substituted classes, cancellations or emergency announcements.

Approach:

1. Use WebSockets (Socket.IO) or server push (Firebase Cloud Messaging / APNs) to deliver real-time alerts.
2. Allow users to subscribe to granular channels (course, faculty, or department level).

Benefits: Reduces communication lag and confusion when changes occur; improves attendance and planning.

Challenges: Managing message volume and relevance; ensuring reliable delivery (retries, fallback to email/SMS).

6.2.4 INTEGRATION WITH INSTITUTIONAL SYSTEMS & SINGLE SIGN-ON (SSO)

Purpose: Integrate with SMVDU's existing ERP, student information system or authentication provider to synchronize users, courses and enrollments.

Approach:

1. Implement SSO via SAML/OAuth2 or the institution's identity provider.
2. Build sync jobs or APIs to import class lists and enrollment data so timetables align with actual student registrations.

Benefits: Eliminates duplicate data entry, keeps timetable aligned with actual registrations, and improves security through central authentication.

Challenges: Requires coordination with institutional IT and data governance policies.

6.3 IMPLEMENTATION ROADMAP

A phased roadmap helps move from the current MVP to a fully featured system:

Phase 1 (0–3 months): Improve validation, add basic conflict detection, responsive UI fixes.

Phase 2 (3–6 months): Implement real-time notifications, mobile app skeleton, and basic analytics.

Phase 3 (6–12 months): Integrate with institutional SSO/ERP, enhance analytics and reporting.

Phase 4 (12+ months): Explore AI-assisted scheduling, advanced personalization and full high-availability deployments.

6.4 METRICS FOR SUCCESS

Define measurable KPIs to evaluate progress:

1. Reduction in scheduling conflicts (% decrease)
2. Time taken to prepare/publish timetable (hours → minutes)
3. Daily active users (students/faculty)
4. Notification delivery success rate
5. Room utilization improvement (%)

6.5 FINAL REMARKS

The Time Table Management System offers a practical, scalable foundation that already reduces manual effort and improves schedule accuracy. By prioritizing conflict detection, mobile accessibility, real-time notifications, and integration with institutional systems, the platform can evolve into a robust, campus-wide scheduling service that significantly enhances operational efficiency and user satisfaction.

6.5.1 KEY LEARNINGS AND TECHNICAL EXPOSURE

1. Full-stack web development using Electron.js
2. Responsive and accessible UI with Tailwind CSS and Bootstrap UI
3. Implementation of Genetic Algorithm in practical use case
4. Version control and platform monitoring using GitHub

REFERENCES

[1] Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill Education.

[2] Sommerville, I., *Software Engineering*, Pearson Education.

[3] Korth, H. F., Silberschatz, A., and Sudarshan, S., *Database System Concepts*, McGraw-Hill.

[4] Elmasri, R., and Navathe, S., *Fundamentals of Database Systems*, Pearson.

[5] MDN Web Docs, *HTML, CSS, and JavaScript Documentation*, Mozilla Developer Network.

[6] W3Schools, *Web Development Tutorials*, W3Schools.com.

[7] GitHub Repository: Timetable Management System for SMVDU (Used for reference in implementation).

[8] Oracle Corporation, *SQL Language Reference Manual*.

[9] IEEE, *Software Development Standards Documentation*.

[10] TutorialsPoint, *Database Connectivity and Web Programming Tutorials*.

APENDIX I: LIST OF DATA SET USED

This appendix provides the details of the dataset used for implementation and testing of the **Time Table Management System for SMVDU**. The data represents a typical academic environment with faculty, subjects, classes, rooms, and time slots.

1. FACULTY DATA

Faculty ID	Faculty Name	Department
F101	Rahul Sharma	CSE
F102	Neha Verma	IT
F103	Amit Kumar	ECE
F104	Pooja Singh	CSE
F105	Arjun Mehta	ME

2. SUBJECT DATA

Subject ID	Subject Name	Credits	Faculty ID
S201	Database System	4	F101
S202	Operating System	4	F102

Apendix I: LIST OF DATA SET USED

S203	Computer Networks	4	F103
S204	Java Programming	3	F104
S205	Software Engineering	4	F105

3. CLASS DATA

Class ID	Course	Semester	Department
C01	B.Tech	3rd	CSE
C02	B.Tech	4th	CSE
C03	B.Tech	5th	IT
C04	B.Tech	6th	ECE

4. TIME SLOT DATA

Slot ID	Time Duration
T1	09:00 AM – 10:00 AM
T2	10:00 AM – 11:00 AM
T3	11:00 AM – 12:00 AM

T4	12:00 AM – 01:00 AM
T5	02:00 AM – 03:00 AM

5. ROOM DATA

Room ID	Room Name	Capacity
R101	Lab1	40
R102	Lab 2	35
R201	Room A	50
R202	Room B	60

6. TIMETABLE SAMPLE DATA

Day	Class	Subject	Faculty	Slot	Room
Monday	C01	Database System	Rahul Sharma	09–10 AM	R201
Tuesday	C02	Operating System	Neha Verma	10–11 AM	R202
Wednesday	C03	Computer Networks	Amit Kumar	11–12 PM	R101
Thursday	C04	Java Programmin g	Pooja Singh	09–10 AM	R102
Friday	C01	Software Engineering	Arjun Mehra	02–03 PM	R201