VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590 018

A Mini - Project Report

On

“SMART SCHEDULAR”

Submitted in partial fulfilment of the requirements for the **MINI PROJECT (BCD586)** course of the 5th semester

Bachelor of Engineering

In

Computer Science & Engineering (DATA SCIENCE)

Submitted by

Mr. Aaftab (4AI23CD001)	**Mr. Rohan H G** (4AI23CD042)
Mr. Shreyas M M (4AI23CD051)	**Mr. Tarun N V** (4AI23CD057)

Under the guidance of

Mrs. Shalini I S **B.E., M.Tech.**

Assistant Professor

Department of CS&E (DATA SCIENCE)
Adichunchanagiri Institute of Technology
CHIKKAMAGALURU - 577102
2025-26

ADHICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY

Jyothinagar, Chikkamagaluru-577102

DEPARTMENT OF CS&E (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini project work entitled **“SMART SCHEDULAR”** is a bonafied work carried out by **Mr. Aaftab (4AI23CD001), Mr. Rohan H G (4AI23CD042) Mr. Shreyas M M (4AI23CD051), Mr. Tarun N V (4AI23CD057),** in partial fulfillment for the **Mini Project (BCS586)** course of 5th semester Bachelor of Engineering in **Computer Science and Engineering (Data Science)** of the Visvesvaraya Technological University, Belagavi during the academic year **2025-2026**. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said Degree.

Signature of the Guide	**Signature of the Coordinator**
Mrs. Shalini I S B.E., M.Tech	**Mrs. Shilpa K V. B.E., M.Tech**
Assistant Professor	**Assistant Professor**

Signature of the HOD
Dr. Adarsh M J B.E., M.Tech., Ph.D
Associate Professor and Head

ABSTRACT

Academic scheduling and examination seating management are critical tasks in educational institutions, yet they are often handled manually, leading to errors, increased workload, and difficulty in maintaining consistency. Traditional scheduling struggles with managing multiple teachers, subjects, weekly hour variations, and strict seating constraints, resulting in class overlaps and inefficient arrangements. The **Smart Scheduler** project solves these issues by providing an intelligent automated system that generates optimized timetables and conflict-free seating plans. Using teacher inputs such as subject type, weekly hours, and year-handling details, the system applies advanced optimization techniques, including **Genetic Algorithms** (GA) and **Google OR-Tools**, to produce accurate, balanced, and rule-compliant schedules.

Built using the **Django framework**, the Smart Scheduler includes a secure relational database for storing academic data and uses **ReportLab** to generate clean, professional PDF reports. The system provides an intuitive dashboard where teachers can input data, view schedules, generate seating arrangements, and download results instantly. By reducing manual effort, eliminating scheduling conflicts, and improving planning accuracy, the Smart Scheduler significantly enhances institutional efficiency. It offers a scalable and reliable solution that modernizes academic administration and supports smooth, transparent, and productive operations in educational environments.

ACKNOWLEDGEMENTS

We express our humble pranamas to his holiness **Divine Soul Parama Poojya Jagadguru Padmabushana Sri Sri Sri Dr.Balagangadharanatha Maha Swamiji** and **Parama Poojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji Pontiff, Sri Adichunchanagiri Maha Samsthana Matt and Sri Sri Gunanatha Swamiji,** Chikkamagaluru branch, Sringeri who have showered their blessings on us.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of help and support from all quarters during the making of this project, so with gratitude, we take this opportunity to acknowledge all those whose guidance and encouragement helped us emerge successful.

We express our gratitude to **Dr. C K Subbaraya**, Director, Adichunchanagiri Institute of Technology.

We express our sincere thanks to our beloved principal**, Dr. C T Jayadeva** for having supported us in our academic endeavors.

We are also indebted to **Dr. Adarsh M J,** HOD of CS&E (DATA SCIENCE) Department, for the facilities and support extended towards us.

We thank our project coordinator **Mrs. Shilpa K V,** Asst. Professor, Department of CS&E (DATA SCIENCE), for her lively correspondence and assistance in carrying on with this project

We are thankful to the resourceful guidance, timely assistance and graceful gesture of our guide **Mrs. Shalini I S,** Asst. Professor, Department of CS&E (DATA SCIENCE), who has helped us in every aspect of our project work.

We would be very pleased to express our heart full thanks to all the teaching and non-teaching staff of CS&E (DATA SCIENCE) Department and our friends who have rendered their help, motivation and support.

Aaftab

Rohan H G

Shreyas M M

Tarun N V

CONTENTS

ABSTRACT ACKNOWLEDGEMENTS **i**

ii

CONTENTS **iii**

LIST OF FIGURES **iv**

LIST OF TABLES **v**

LIST OF SNAPSHOTS **vi**

CHAPTERS PAGE NO

1. Introduction

1.1 Background 01

1.2 Problem Statement 01

1.3 Objectives of the system 01

1.4 Significance of the system 02

1.5 Scope of the project 02

1.6 Methodology 02

1.7 Target Audience 03

1.8 Overview of the report 03

2. System Design

2.1 System Architecture 04

2.2 Module Design 05

2.2.1 Teacher & Subject Input Module 05

2.2.2 Timetable generation Module 05

2.2.3 Seating Arrangement Module 05

2.2.4 PDF Generation Module 05

2.3 Database Design 05

2.4 User Interface Design 05

2.5 Technology Stack 06

3. Implementation

3.1 Backend Implementation 07

3.2 Timetable Implementation 07

3.4 Seating Arrangement Implementation 08
3.4 Database Implementation 08
3.5 Frontend Implementation 09
3.6 PDF Generation Implementation 09
3.7 System Integration 09

4. Testing

4.1 Testing Objectives 10
4.2 Testing Environment 10
4.3 Type of testing 11
4.4 Test Cases 12

5. Results and Discussions

5.1 Results 13-14
5.2 Discussion 15

6. Conclusion and Future Enhancement

6.1 Conclusion 16
6.2 Future Enhancement 16

REFERENCES 17

List of Figures

Sl. No	Description	Page No
2.1	System Architecture	4

List of Tables

Sl. No	Description	Page No
3.1	Database Schema Overview	08
4.1	Test Cases	12

List of Snapshots

Sl. No	Description	Page No
5.1	Login Page	13
5.2	Teacher Dashboard Options	13
5.3	Seating Arrangement Results	14
5.4	Generated Timetables	14
5.5	Generated Class Timetable with Subject Allocation	15

Chapter 1

INTRODUCTION

1.1 Background

- **Context**: Timetable creation and examination seating arrangement are essential administrative tasks in educational institutions. Traditionally, these tasks are performed manually by department coordinators or faculty members. Manual preparation is time-consuming, prone to clashes, inconsistent, and difficult to update when changes occur.
- **Problem**: Manual scheduling often leads to multiple issues such as timetable clashes, uneven workload distribution, incorrect hour allocation, and errors in seating arrangements. When several teachers handle multiple classes, maintaining accuracy and consistency becomes challenging.
- **Opportunity**: The adoption of automated systems provides institutions with the ability to generate optimized timetables and seating plans efficiently. Modern technologies such as AI-based optimization, Genetic Algorithms, and OR-Tools make it possible to automate scheduling with high accuracy and real-time usability.

1.2 Problem Statement

- **Overview of the Problem**: Educational institutions face significant challenges in managing academic schedules due to the increasing complexity of handling multiple teachers, subjects, and year-wise classes. Manual timetable preparation is highly time-consuming and often leads to errors such as class overlaps, inconsistent hour allocation, and inefficient teacher distribution. Additionally, examination seating arrangements are usually created without a systematic approach, resulting in students from the same year being seated together, which compromises fairness.

- To address these challenges, an automated Smart Scheduler system is proposed that efficiently generates optimized, clash-free timetables and systematic examination seating arrangements. The system minimizes manual effort, eliminates scheduling errors, ensures fair student seating, and improves overall accuracy and efficiency in academic planning.

1.3 Objective of the System

- The primary objective of the **Smart Scheduler** is to provide an automated and intelligent system that generates class timetables and exam seating arrangements based on predefined academic rules and constraints.

- **Specific Goals**:
  - **Automate scheduling:** Reduce manual effort by automatically generating timetables and seating layouts.
  - **Ensure accuracy:** Eliminate timetable clashes, incorrect hour allocation, and manual calculation errors.
  - **Improve fairness:** Randomize seating nd ensure same-year students are not seated adjacent

- **Enhance usability:** Provide an easy-to-use dashboard for teachers and a view-only interface for students.
- **Generate reports:** Allow users to download professional PDF timetables and seating plans.
- **Support scalability:** Handle multiple teachers, subjects, and years without performance issues.
- **Centralize data**: Store all academic information in a unified database for quick retrieval.

1.4 Significance of the System

The *Smart Scheduler* offers several benefits that enhance academic

- **Time Efficiency:** Automates complex scheduling tasks, reducing manual workload and preparation time.
- **Accuracy & Reliability:** Prevents common human errors such as double-booking or incorrect hour allocation.
- **Optimized Workload Distribution:** Ensures fair weekly distribution of classes for teachers.
- **Fair Student Allocation:** Generates seating arrangements that promote discipline and reduce malpractice.
- **Consistency & Transparency:** Maintains centrally stored data for easy verification at any time.
- **Cost-Effective:** Reduces paper usage and administrative overhead.
- **Scalable & Future-Ready:** Can be expanded to include new features such as room allocation or attendance integration.

1.5 Scope of the Project

- **In Scope**:
  - Automated generation of class timetables based on subject type and weekly hours.
  - Automated seating arrangement with randomization and separation rules.
  - Teacher dashboard for entering subject details, class handling, and generating schedules.
  - Centralized database using SQLite.
  - PDF report generation using ReportLab.
- **Out of Scope**:
  - Biometric, RFID, or QR-based attendance systems.
  - Classroom or hall allocation modules (unless added later).
  - Student academic performance monitoring.
  - Integration with external LMS or ERP systems.

1.6 Methodology

- The Smart Scheduler is developed as a web-based system using a combination of backend and optimization techniques.
- Django acts as the backend framework for handling logic, routing, and data flow. The optimization core uses **Genetic Algorithms** and **Google OR-Tools** to generate conflict-free timetables and valid seating plans based on constraints such as subject hours, integrated subjects, and seating
- The system follows an **Agile development model**, allowing iterative development and continuous feedback from faculty members. Each module was developed, tested, and refined in cycles.
- **Testing**: The system will be tested through a combination of **unit testing**, **integration testing**, and **user acceptance testing** to ensure functionality and user satisfaction.

1.7 Target Audience

- **Teachers:** Enter academic details, generate timetables, and approve seating arrangements.
- **Students:** View and download the finalized schedules**.**

1.8 Overview of the Report

This report is structured into several chapters that detail the development and design of the **Smart Schedular**. The following chapters include:

- **Chapter 1:** Gives a brief introduction of the project.
- **Chapter 2: System Design** – Describes the architecture and design of the system.
- **Chapter 3: Implementation** – Discusses the system's development and the technologies used.
- **Chapter 4: Testing and Validation** – Details the testing process and results.
- **Chapter 5: Results and Discussions** – Presents and results obtained and discusses the limitations
- **Chapter 6: Conclusion and Future enhancement -** Summarizes the project and suggests future improvements.

Chapter 2

SYSTEM DESIGN

This chapter explains the complete system design of the Smart Schedular, describing how the timetable generation and seating arrangement modules are architected, integrated, and managed within the application. The design approach focuses on automation, conflict-free schedule generation, and efficient classroom seating allocation, ensuring the system meets the academic requirements of faculty and students. The system is built to be scalable, reliable, and user-friendly, enabling seamless interaction for teachers and administrators.

2.1 System Architecture

- Smart Schedular follows a client–server architecture, where the frontend interacts with the backend server that processes all timetable and seating logic and communicates with a centralized database. Users, such as teachers and administrators, access the system through a web dashboard, from which they enter subject details, class years, and constraints. The backend then uses optimization algorithms like **Genetic Algorithm** (GA) and **Google OR-Tools** to generate conflict-free timetables and valid seating plans.

The architecture consists of three primary components:

- **Frontend (User Interface):** A clean and responsive web interface built with HTML, CSS, and JavaScript (or React.js). It allows teachers to enter timetable requirements, edit seating, view results, and download PDFs.
- **Backend Server:** Developed using Django, the backend handles input validation, scheduling logic, optimization algorithms, and generation of final timetables and seating arrangements.
- **Components**: Student data, subjects, teacher workloads, room details, and seating preferences are stored in a relational database such as SQLite. This ensures structured storage, fast retrieval, and long-term record management.

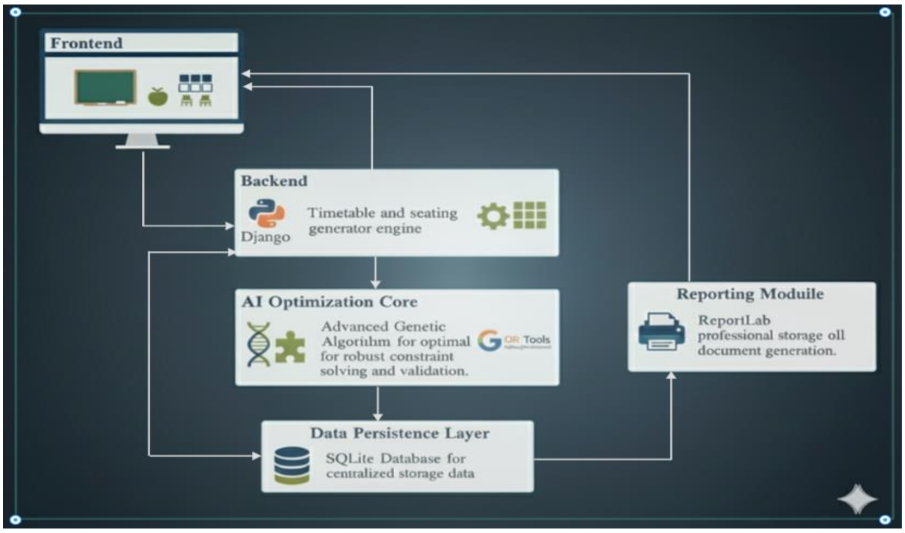

Frontend
Backend
Django
Timetable and seating generator engine
AI Optimization Core
Advanced Genetic Algorithm for optimal for robust constraint solving and validation.
OR Tools
Reporting Moduile
ReportLab professional storage oll document generation.
Data Persistence Layer
SQLite Database for centralized storage data

Figure 2.1 System Architecture

2.2 Module Design

To simplify development and ensure maintainability, the Smart Schedular is divided into several functional modules.

2.2.1 Teacher & Subject Input Module

This module collects all required information from teachers, including subject type (integrated or non-integrated), required weekly hours, lab requirements, and the number of years handled. The system uses this data to generate valid schedules.

2.2.2 Timetable Generation Module

This is the core module of the system. It processes all constraints such as:

- Teacher availability
- Year-based subject hours
- Integrated theory/lab hours
- No teacher year clash
- Balanced hourly distribution

2.2.3 Seating Arrangement Module

This module automatically generates seating charts for internal examinations.

It ensures:

- Students from the same year are not adjacent
- Randomized but valid placements
- Teacher override options to remove or replace students
- Final seating PDF export

2.2.4 PDF Generation Module

The system uses ReportLab templates to produce clean, formatted PDF files for:

- Final timetable
- Seating arrangement charts

2.3 Database Design

The database is designed to store structured academic data. Major tables include:

- Teacher Table: Stores teacher details, subjects, and workload.
- Subject Table: Stores subject type (integrated/non-integrated), hours required, lab details.
- Year/Class Table: Represents each academic year or section.
- Student Table: Stores student details for seating arrangement.
- Timetable Table: Stores generated timetable slots for each year.
- Seating Table: Stores exam seating positions for each student.

2.4 User Interface (UI) Design

The user interface is designed to be minimal, intuitive, and accessible for faculty.

- **Main Screens**:
    - **Login Page:** Secure login for teachers and admin.
    - **Dashboard:** A central screen that displays all features such as "Generate Timetable", "Generate Seating", and “Download PDF”.
    - **Timetable Input Screen:** Teachers enter subjects, year details, and constraints.
    - **Seating Input Screen:** Admin/teacher uploads student lists or selects classes.
    - **Preview Screen:** Shows generated timetables and seating arrangements before finalizing.
    - **Download Screen:** Allows exporting PDFs.

2.5 Technology Stack

- **Frontend**: HTML, CSS, JavaScript, or frontend frameworks like React for an interactive user interface.
- **Backend**: Python Django – handles scheduling algorithms, validation, business logic, and API services.
- **Database**: SQLite for secure, scalable storage.
- **Optimization Tools:**
    - **Genetic Algorithm (GA):** For evolving best-fit timetable solutions.
    - **Google OR-Tools:** For constraint solving and time-slot optimization
- **PDF Generation:** ReportLab for creating professional-quality timetable and seating PDFs.

Chapter 3

IMPLEMENTATION

This chapter describes the implementation of the Smart Scheduler system, focusing on backend logic, optimization workflow, database structure, frontend interface, and PDF export functionality. The goal of implementation was to convert the planned system design into a working solution capable of automatically generating class timetables and examination seating arrangements with minimal human effort.

3.1 Backend Implementation

- The backend is the core processing unit of the system, built using the **Django framework**. It manages input validation, applies scheduling constraints, communicates with the optimization algorithms, interacts with the database, and returns results to the frontend.

- The backend is structured into **Models**, **Views**, **Services**, and **Controllers**:
  - **Models:** Represent database tables for teachers, subjects, classes, students, timetables, and seating.
  - **Views/Controllers:** Handle requests from the frontend such as timetable generation or seating creation.
  - **Service Layer:** Contains business logic—GA operations, OR-Tools seating solver, constraint checks.
  - **Serializers (optional):** Used when data is sent/received through APIs.

- Key Backend Functionalities
  - Process subject input and weekly hour allocation.
  - Identify subject type (Theory, Lab, Integrated).
  - Validate teacher availability for multiple years.
  - Generate timetable without clashes.
  - Generate seating layout with separation constraints.
  - Store results to database & send formatted response to frontend.

3.2 Timetable Implementation

A Genetic Algorithm (GA) is used to generate optimized, clash-free timetables.

Workflow of GA in the system

- **Input Collection:** Teacher, subject, hours/week, year details.
- **Population Creation:** Random timetable combinations generated.
- **Fitness Evaluation:** Penalty given for:
  - Teacher handling two classes at same time
  - Unbalanced hour distribution
  - Wrong placement of lab/integrated hours

  - Excessive continuous sessions
- **Selection:** Best candidates chosen for next generation.
- **Crossover:** Two timetables exchange slots to produce new ones.
- **Mutation:** Random slot modifications to explore better solutions.
- **Termination:** Stops when no clash remains or after max iterations.
- **Final Output:** Best timetable stored in database & sent for PDF export.

3.3 Seating Arrangement Implementation

The seating module uses Google OR-Tools to automate student placement for exam halls.

Seating Rules Implemented

- No two students of same year sit adjacent (left-right/front-back).
- Randomized allocation to avoid predictable patterns.
- Supports manual edit/removal before final approval.
- Final output saved and downloadable as PDF.

Algorithm Workflow

- Create seat grid → assign one student per seat.
- Apply year-separation constraints.
- OR solver tries combinations until rules satisfied.
- Output structured seating plan.

3.4 Database Implementation

A relational database SQLite is used to store all system data. Efficient indexing ensures fast query execution while generating schedules.

Table 3.1 Database Schema Overview

Table Name	Description
Teacher	Stores teacher details, subjects handled
Subject	Stores subject name, type (theory/lab/integrated), weekly hours
Year/Class	Stores academic year mappings
Timetable	Final scheduled slots for each year
Students	Used for seating arrangement
Seating	Row-column mapping for each student

3.5 Frontend Implementation

The frontend is developed using **HTML, CSS & JavaScript**, with teacher-friendly UI interaction

Main Screens

- Login Screen – Role-based authentication.
- Teacher Dashboard – Create timetable and seating.
- Input Page – Enter subjects, hours, year handling.
- Generation Screen – View generated output preview.
- Student Interface – View/download timetable or seating.
- PDF Download Page – Export formatted reports**.**

3.6 PDF Generation Module

The ReportLab library is used to create printable PDF reports.

Features

- Professional layout for timetable and seating.
- Header, footer, and institutional branding.
- Clean tables, proper alignment, font consistency.
- One-click PDF save for faculty/students.

3.7 System Integration

All modules are bound through a smooth workflow:

- Teacher enters subject & class details.
- Backend validates input and sends it to GA.
- GA generates timetable → saved to DB.
- Seating generated using OR-Tools → saved to DB.
- User previews & exports PDF.

Chapter 4

TESTING

Testing is a crucial phase of software development, used to evaluate correctness, reliability, performance and confirm that the system behaves as expected. For the Smart Scheduler, testing was performed to ensure that timetable generation, seating arrangement, user access, and PDF generation worked correctly under various inputs and constraints. The main objective was to verify that the system produces clash-free timetables, valid seating layouts, prevents scheduling errors, and provides stable usage for teachers and students.

4.1 Testing Objectives

The primary goals of testing were:

- To ensure each module functions according to the requirements.
- To verify timetable generation is clash-free and meets weekly hour rules.
- To validate seating arrangement rules, ensuring no same-year adjacency.
- To ensure role-based access control works properly.
- To detect and correct functional/logical errors in early stages.
- To check PDF export functionality and database communication.
- To evaluate system performance under multiple input combinations.
- To ensure the interface remains responsive and stable.

4.2 Testing Environment

Testing was carried out using the following environment setup:

Hardware:

- Laptop/PC with minimum Intel i5/Ryzen equivalent
- 8GB RAM recommended for optimization algorithms Software:

Backend:

- Developed using Django (Python).
- Optimization: Uses Genetic Algorithm and Google OR-Tools.
- Database: SQLite / MySQL for storing system data.
- Frontend: Designed using HTML, CSS, and JavaScript.
- PDF Generation: Implemented using ReportLab.
- API/Logic Testing: Performed using Postman.
- Browser Testing: Tested on Chrome, Firefox, and Edge.

Operating System:

- Windows 10/macOS/Linux.

Browser:

- Google Chrome, Mozilla Firefox, and Microsoft Edge for cross-browser testing.

4.3 Types of testing

4.3.1 Unit Testing

Unit testing was performed to test individual functions such as subject input handling, GA fitness evaluation, student seating placement logic, and PDF generation.

Examples:

- Verify GA generates timetable slots without overlap.
- Ensure seating solver prevents adjacent year conflicts.
- Check timetable stored in database after generation.
- Confirm PDF generation formats data correctly.

Outcome: All core functions executed successfully.

4.3.2 Integration Testing

Integration testing verified smooth data flow between frontend, backend, database, PDF generator.

Scenarios Tested:

- Teacher input → Backend validation → GA scheduling.
- Seating module → OR-Tools solver → Database storage.
- Timetable retrieval → UI view → PDF download.

4.3.3 Functional Testing

Functional testing ensured that features aligned with every requirement of the system.

Test Scenarios:

- Teacher adds subjects and generates timetable correctly.
- System prevents teacher clash between year classes.
- Lab subjects allocate 2–3 consecutive hours.
- Seating generator separates same-year students.
- Students can view/download schedules only.
- Unauthorized user actions are blocked.

4.4 Test Cases

The Smart Scheduler system was tested using several test cases to ensure all major functions worked correctly. The login module was verified by checking successful access with valid credentials (TC-001) and proper error messages for invalid attempts (TC-002). The core features were also tested: timetable generation produced conflict-free schedules (TC-003), and the seating arrangement module correctly prevented same-year students from sitting together (TC-004). Additional tests confirmed that PDF downloads worked smoothly (TC-005) and that unauthorized users, such as students, could not access teacher-only features (TC-006). All test cases passed, showing that the system is functioning reliably and meets its requirements.

Below are sample test cases for various components:

Table 4.1 Test Cases

Test Case ID	**Description**	**Test Steps**	**Expected Result**	**Status**
TC-1	Login with valid credentials	Enter valid username & password → Login	Redirect to dashboard	Pass
TC- 2	Invalid login attempt	Enter wrong details → Login	Show error message & deny access	Pass
TC- 3	Generate timetable	Teacher enters subject/year details → Generate	System produces clash-free timetable	Pass
TC- 4	Seating arrangement generation	Upload student list → Generate seating	No same-year adjacency, valid seating matrix	Pass
TC- 5	PDF export	Click Download PDF	Timetable/Seating PDF downloads successfully	Pass
TC- 6	Unauthorized access	Student tries to access teacher module	Access denied message shown	Pass

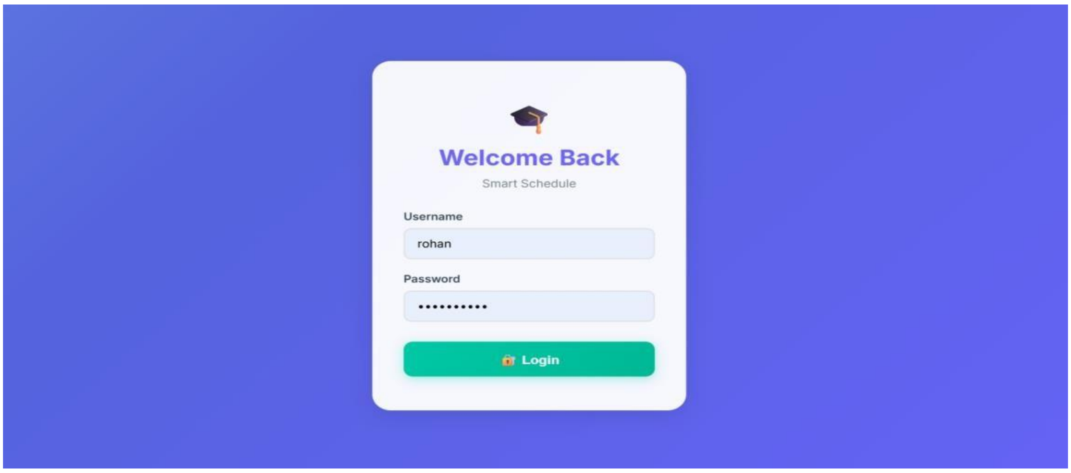
Chapter 5

RESULTS AND DISCUSSION

This chapter presents the output generated by the Smart Scheduler system, along with observations, performance evaluation, effectiveness, advantages, and challenges faced during development. The results validate how well the system meets project objectives such as automated timetable creation, seating arrangement generation, and PDF exporting.

5.1 Results

- **Login Interface:** The Figure 5.1 shows the login page of the Smart Scheduler system, where users securely authenticate using their username and password.



Welcome Back

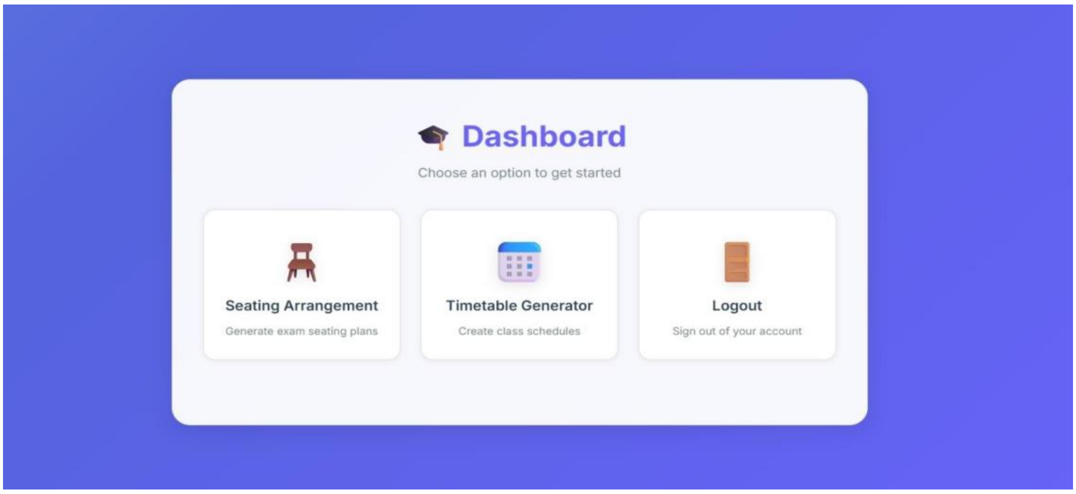
Smart Schedule

Username: rohan

Password: ••••••••••

Figure. 5.1 Login Page

- **Dashboard Interface:** The Figure. 5.2 serves as the central control panel of the Smart Scheduler system. It provides users with quick access to core functionalities such as timetable generation, examination seating arrangement, and secure logout, enabling easy and efficient navigation within the system.

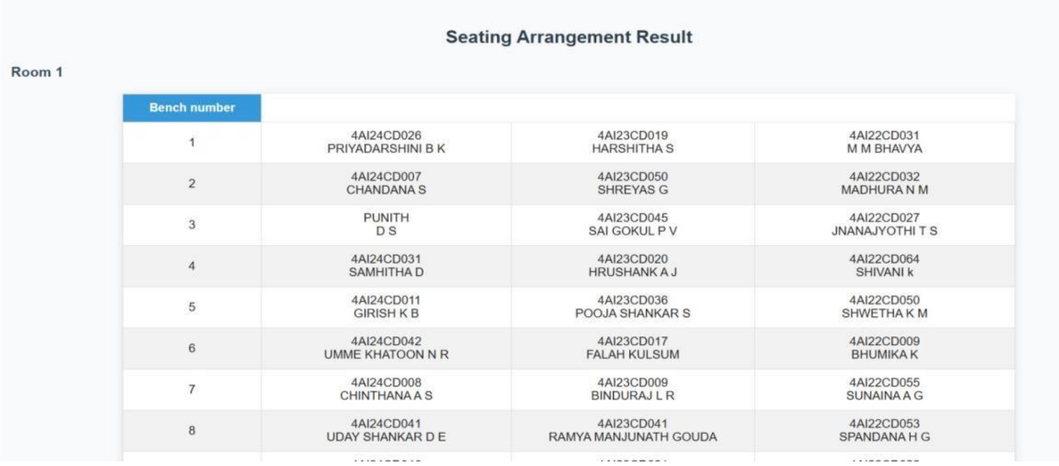


Dashboard

Choose an option to get started

Figure. 5.2 Teacher Dashboard Options

- **Seating Arrangement Output:** This Figure. 5.3 displays the automatically generated examination seating arrangement for a classroom. Students are assigned bench numbers in a structured format while ensuring seating rules are followed, such as avoiding adjacent seating of same-year students. The result provides a clear, organized, and ready-to-use seating plan.



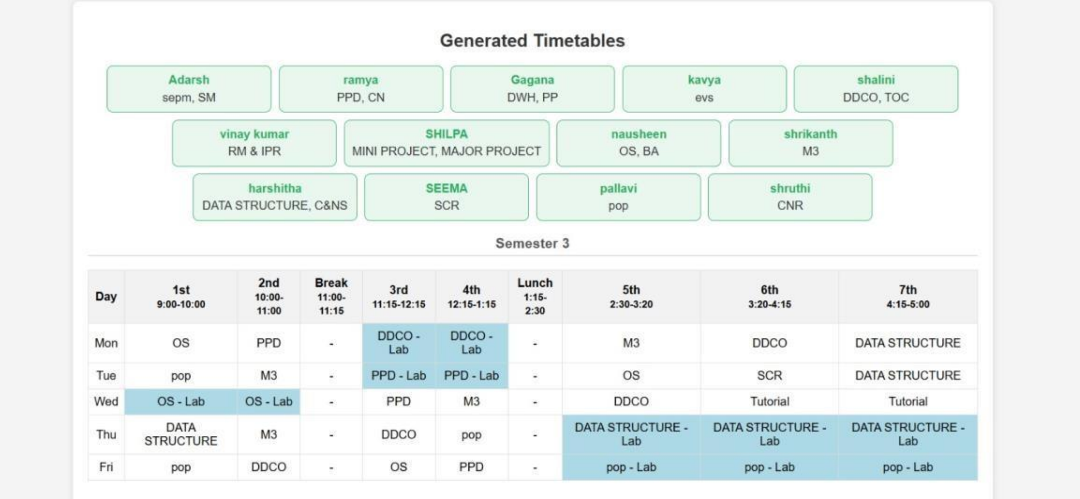
Seating Arrangement Result

Room 1

Bench number			
1	4AI24CD026 PRIYADARSHINI B K	4AI23CD019 HARSHITHA S	4AI22CD031 M M BHAVYA
2	4AI24CD007 CHANDANA S	4AI23CD050 SHREYAS G	4AI22CD032 MADHURA N M
3	PUNITH D S	4AI23CD045 SAI GOKUL P V	4AI22CD027 JNANAJYOTHI T S
4	4AI24CD031 SAMHITHA D	4AI23CD020 HRUSHANK A J	4AI22CD064 SHIVANI k
5	4AI24CD011 GIRISH K B	4AI23CD036 POOJA SHANKAR S	4AI22CD050 SHWETHA K M
6	4AI24CD042 UMME KHATOON N R	4AI23CD017 FALAH KULSUM	4AI22CD009 BHUMIKA K
7	4AI24CD008 CHINTHANA A S	4AI23CD009 BINDURAJ L R	4AI22CD055 SUNAINA A G
8	4AI24CD041 UDAY SHANKAR D E	4AI23CD041 RAMYA MANJUNATH GOUDA	4AI22CD053 SPANDANA H G

Figure. 5.3 Seating Arrangement Result

- The system generates class-wise timetables as well as individual teacher timetables for better academic planning.
- By clicking on a teacher's name, the system displays the complete individual timetable of that faculty member.



Generated Timetables

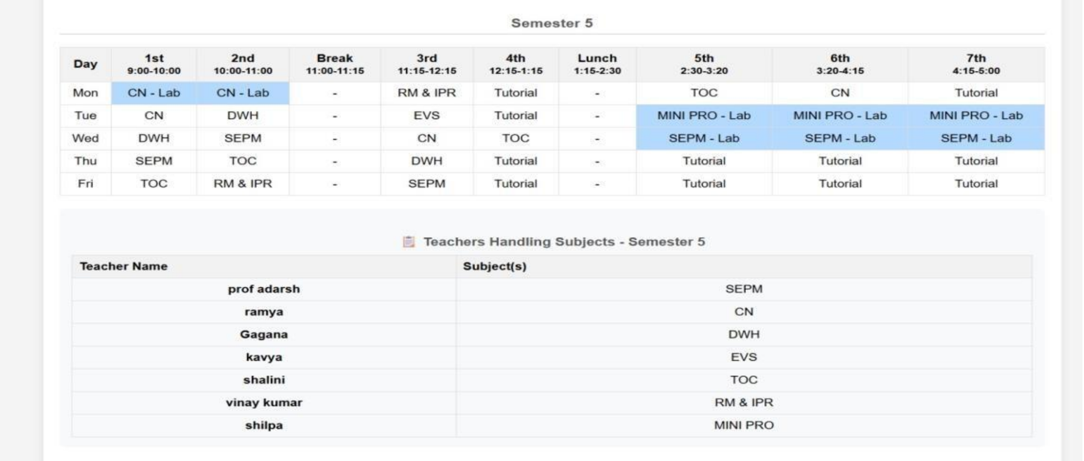
Adarsh	sepm, SM
ramya	PPD, CN
Gagana	DWH, PP
kavya	evs
shalini	DDCO, TOC
vinay kumar	RM & IPR
SHILPA	MINI PROJECT, MAJOR PROJECT
nausheen	OS, BA
shrikanth	M3
harshitha	DATA STRUCTURE, C&NS
SEEMA	SCR
pallavi	pop
shruthi	CNR

Semester 3

Day	1st 9:00-10:00	2nd 10:00-11:00	Break 11:00-11:15	3rd 11:15-12:15	4th 12:15-1:15	Lunch 1:15-2:30	5th 2:30-3:20	6th 3:20-4:15	7th 4:15-5:00
Mon	OS	PPD	-	DDCO - Lab	DDCO - Lab	-	M3	DDCO	DATA STRUCTURE
Tue	pop	M3	-	PPD - Lab	PPD - Lab	-	OS	SCR	DATA STRUCTURE
Wed	OS - Lab	OS - Lab	-	PPD	M3	-	DDCO	Tutorial	Tutorial
Thu	DATA STRUCTURE	M3	-	DDCO	pop	-	DATA STRUCTURE - Lab	DATA STRUCTURE - Lab	DATA STRUCTURE - Lab
Fri	pop	DDCO	-	OS	PPD	-	pop - Lab	pop - Lab	pop - Lab

Figure. 5.4 Generated Timetable

- The system displays **class-wise timetables along with subject allocation**



Semester 5

Day	1st 9:00-10:00	2nd 10:00-11:00	Break 11:00-11:15	3rd 11:15-12:15	4th 12:15-1:15	Lunch 1:15-2:30	5th 2:30-3:20	6th 3:20-4:15	7th 4:15-5:00
Mon	CN - Lab	CN - Lab	-	RM & IPR	Tutorial	-	TOC	CN	Tutorial
Tue	CN	DWH	-	EVS	Tutorial	-	MINI PRO - Lab	MINI PRO - Lab	MINI PRO - Lab
Wed	DWH	SEPM	-	CN	TOC	-	SEPM - Lab	SEPM - Lab	SEPM - Lab
Thu	SEPM	TOC	-	DWH	Tutorial	-	Tutorial	Tutorial	Tutorial
Fri	TOC	RM & IPR	-	SEPM	Tutorial	-	Tutorial	Tutorial	Tutorial

Teachers Handling Subjects - Semester 5

Teacher Name	Subject(s)
prof adarsh	SEPM
ramya	CN
Gagana	DWH
kavya	EVS
shalini	TOC
vinay kumar	RM & IPR
shilpa	MINI PRO

Figure. 5.5 Generated Class Timetable with Subject Allocation

5.2 Discussion

Effectiveness of the System

- The system automated scheduling processes that earlier required manual planning and reduced workload significantly.
- Optimization logic (GA + OR-Tools) ensured smart allocation without conflicting hours or seating violations.
- The UI was simple enough for non-technical faculty to use with minimal guidance.
- Role-based access increased security and data control, preventing unauthorized modifications.
- PDF generation improved record management and made sharing outputs easier.

Strengths Observed

- Fast generation of timetables and seating layout.
- Accurate clash detection and constraint handling.
- Scalable design capable of adding new departments/years.
- Centralized data storage with quick retrieval.

Challenges Encountered

- Building a fitness function for GA to avoid all clashes required multiple refinement cycles.
- Seating separation rules needed adjustment during initial testing.
- Frontend alignment for PDF export required formatting improvements.
- Performance slowed slightly when handling very large student batches.

Limitations of the Current System

- Seating is limited to defined classroom hall structure.
- Does not currently support *auto room allocation* (can be part of future scope).
- No attendance or marks integration modules at present.
- Requires faculty input for subjects & hours manually (not yet auto-linked to ERP).

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

- The **Smart Scheduler System** successfully accomplishes its main objective of automating the process of timetable generation and seating arrangement planning in educational institutions. By using **Genetic Algorithm (GA) and Google OR-Tools**, the system intelligently allocates subjects, teachers, and time slots without clashes, while also generating seating layouts that prevent students of the same year from sitting adjacent to each other.
- The developed platform provides:
  - Automated timetable and seating generation
  - Error-free scheduling with optimized resource utilization
  - Easy PDF export for reporting and record keeping
  - Role-based dashboard for teachers and students
  - Faster planning compared to manual scheduling methods
- Overall, the Smart Scheduler proves to be an efficient, reliable, and scalable system that reduces human workload, minimizes scheduling errors, and modernizes academic administration. It demonstrates how automation can significantly improve efficiency, transparency, and time management in educational institutions.

6.2 Future Enhancements

Although the system performs effectively in its current state, several improvements can enhance performance and usability in the future:

1. Auto Classroom Allocation
   - Automatically assign rooms based on capacity and class size.
   - Useful for large institutions handling multiple departments at once.
2. Live Web View / Portal Access
   - Students and faculty can view schedules online without downloading PDFs.
   - Real-time update reflection in case of timetable changes.
3. Notification & Alert System
   - Send reminders to teachers for pending scheduling approval.
   - Notify students about changes in timetable or room allocation.
4. Mobile Application
   - Android/iOS app for quick access to schedules anywhere.
   - Push notifications and offline PDF viewing support.

REFERENCES

[1] Rajaraman, V. *Fundamentals of Computers* (6th ed.). PHI Learning.

[2] Pressman, R. S. *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill.

[3] A. Kumar and R. Singh, "Automated Timetable Scheduling Using Optimization . Techniques," *IEEE Access*, vol. 9, pp. 112345–112356, 2021

[4] S. Khalil and M. Abdelrahman, "Optimization-Based Classroom Scheduling Systems," . *IEEE International Conference on Smart Computing*, pp. 215–220, 2021.

[5] Kim, J., & Kim, Y. (2020). *Automated Timetable Scheduling Using Optimization Algorithms*. IJCA, 175(32), 18–25.

[6] Django Software Foundation. (2024). *Django Web Framework*. https://www.djangoproject.com

[7] Google. (2024). *OR-Tools Optimization Suite*. https://developers.google.com/optimization

[8] Python Software Foundation. (2024). *Python Programming Language*. https://www.python.org

[9] SQLite Documentation. (2024). *SQLite Database Engine*. https://www.sqlite.org

[10] ReportLab Inc. (2024). *ReportLab PDF Toolkit*. https://www.reportlab.com