

**“SCHEDULE  
MANAGEMENT SYSTEM”**  
**Second year Mini Project Report**

Submitted in partial fulfillment of the requirements of the  
degree

**BACHELOR OF ENGINEERING IN COMPUTER  
ENGINEERING**

By

**Rohit Shahi/58**

**Umesh Tolani/64**

**Aditya Ajith/1**

Supervisor

**Prof. Richard Joseph**



**Department of Computer Engineering**

**Vivekanand Education Society's Institute of  
Technology HAMC, Collector's Colony, Chembur,  
Mumbai-40007 University of Mumbai (AY 2023-24)**

# CERTIFICATE

This is to certify that the Mini Project entitled '**Schedule Management System**' is a bonafide work of **Aditya Ajith (1)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Engineering**" in "**Computer Engineering**".

Prof. Richard Joseph  
Supervisor

Prof.NUPUR GIRI  
Head of Department

Prof.DR J.M.NAIR  
Principal

# Mini Project Approval

This Mini Project entitled "Schedule Management system" by  
**Aditya Ajith (1)** is approved for the degree of **Bachelor of**  
**Engineering in Computer Engineering.**

## Examiners

1.....

(Internal Examiner Name & Sign)

2.....

(External Examiner name & Sign)

Date:

Place:

# Contents

|   |            |
|---|------------|
| <b>Abstract</b>   | <b>ii</b>  |
| <b>Acknowledgments</b>  | <b>iii</b> |
| <b>List of Abbreviations</b>                                      | <b>iv</b>  |
| <b>List of Figures</b>  | <b>v</b>   |
| <b>List of Tables</b>   | <b>vi</b>  |
| <b>List of Symbols</b>  | <b>vii</b> |
| <br>  |            |
| <b>1 Introduction</b>   | <b>1</b>   |
| 1.1 Introduction  |            |
| 1.2 Motivation  |            |
| 1.3 Problem Statement & Objectives                                |            |
| 1.4 Organization of the Report                                    |            |
| <br>  |            |
| <b>2 Literature Survey</b>  | <b>11</b>  |
| 2.1 Survey of Existing System                                     |            |
| 2.2 Limitation Existing system or research gap                    |            |
| 2.3 Mini Project Contribution                                     |            |
| <br>  |            |
| <b>3 Proposed System (eg New Approach of Data Summarization )</b> | <b>18</b>  |
| 3.1 Introduction  |            |
| 3.2 Architecture/ Framework                                       |            |
| 3.3 Algorithm and Process Design                                  |            |
| 3.4 Details of Hardware & Software                                |            |
| 3.4 Experiment and Results  |            |
| 3.5 Conclusion and Future work.                                   |            |
| 3.6 References  |            |

## Introduction:

The "Schedule Management System" is a comprehensive web application developed using Django, Bootstrap, and SQL, designed to streamline and simplify the management of various aspects of educational institutions. This system provides a centralized platform for handling classes, exams, attendance records, and institute-related information, thereby enhancing efficiency and organization in educational settings.

## Motivation:

The motivation behind creating the "Schedule Management System" stemmed from the need to overcome the inherent challenges and complexities faced by educational institutions in managing their day-to-day operations. These challenges include class scheduling conflicts, manual attendance tracking, and exam management hassles. By developing this system, we aim to automate and optimize these processes, ultimately making life easier for both educators and students.

## Problem Statement & Objectives:

### Problem Statement:

Educational institutions often encounter inefficiencies in managing classes, exams, attendance, and other administrative tasks. These inefficiencies lead to scheduling conflicts, time wastage, and errors in attendance tracking.

### Objectives:

- **Efficient Class Management:** The primary objective of this project is to offer an efficient class scheduling system that eliminates conflicts and ensures optimal allocation of resources.
- **Streamlined Exam Management:** The system aims to simplify the process of creating, scheduling, and managing exams, reducing administrative overhead.
- **Automated Attendance Tracking:** Automation of attendance records ensures accurate and real-time tracking, reducing manual efforts and errors.
- **Comprehensive Institute Information:** The project will provide a central repository for managing essential institute-related data, such as student and faculty information.
- **User-Friendly Interface:** The system's user interface is designed to be intuitive and user-friendly, catering to both educators and students.
- **Data Security:** Data security is a priority, and measures are in place to ensure the confidentiality and integrity of the information stored.
- **Scalability:** The system is designed with scalability in mind, enabling it to adapt to the evolving needs of educational institutions.

## Literature Survey:

### Survey of Existing System:

In this section, we will review existing systems or methodologies related to schedule management and educational institution administration. We aim to provide a comparative analysis to understand the strengths and weaknesses of these systems. Some of the existing systems and areas of focus may include:

- **Manual Scheduling:** Many educational institutions still rely on manual scheduling, which is time-consuming and prone to errors.
- **Spreadsheet-based Solutions:** Some institutions use spreadsheets to manage classes and exams, which lack the automation and real-time features of a dedicated system.
- **Commercial Software:** Commercial software may offer scheduling and management solutions, but they often come with high costs and limited customization.
- **Research Publications:** Relevant research papers and publications in the field of educational management systems.
- **Open-Source Solutions:** An examination of open-source tools and frameworks used in educational management, if applicable.

### Limitations of the Existing System or Research Gap:

This section identifies the shortcomings of the existing systems and highlights the research gap that the "Schedule Management System" aims to address:-

- **Manual Errors:** Manual scheduling is error-prone and time-consuming. It can lead to scheduling conflicts, impacting the efficiency of educational institutions.
- **Lack of Automation:** Spreadsheet-based solutions lack the automation and realtime updates needed to efficiently manage class schedules and exams.
- **High Costs:** Commercial software can be expensive, making it inaccessible to smaller educational institutions.
- **Limited Customization:** Existing solutions may not provide the flexibility needed to adapt to the specific needs of different institutions.
- **Research Gap:** The research gap is the lack of an integrated, cost-effective, and highly customizable solution that can efficiently manage classes, exams, attendance, and institute information while providing a user-friendly interface.

## Mini Project Contribution:

This section outlines the specific contributions of your "Schedule Management System" project to address the limitations and research gap identified:

- **Automation:** The project automates class scheduling, exam management, and attendance tracking, reducing the manual effort and the potential for errors.
- **Cost-Effective:** It offers a cost-effective alternative to expensive commercial software, making it accessible to a broader range of educational institutions.
- **Customization:** The system allows for a high degree of customization, ensuring that it can be tailored to the unique needs of each institution.
- **User-Friendly:** The project focuses on providing an intuitive and user-friendly interface, enhancing the user experience for both educators and students.
- **Research Contribution:** This project contributes to the body of research in educational management systems by offering an open-source solution with a comprehensive feature set.

## Proposed System:

### Introduction:

The "Schedule Management System" represents a transformative solution designed to streamline and optimize the complex administrative tasks within educational institutions. Educational settings encompass a multitude of intricate processes, ranging from class management to exam scheduling, attendance tracking, and maintaining crucial institution-related data. The conventional methods employed for these tasks often lead to inefficiencies, errors, and time-consuming procedures.

In response to these challenges, the "Schedule Management System" has been conceived as a comprehensive and user-friendly web-based platform. It provides educators, administrators, and students with a centralized hub for managing their daily activities.

### Architecture/Framework:

The architecture of the "Schedule Management System" is carefully designed to provide a robust and efficient platform for managing educational institution resources. It is built on a combination of key technologies, frameworks, and components that work harmoniously to fulfill the project's goals.

## Overall System Architecture:

At its core, the system follows a client-server architecture. The client-side is built using Bootstrap, a popular front-end framework known for its responsive design and user-friendly interface. Bootstrap ensures that the user experience is seamless, irrespective of the device or screen size, making it a crucial component for the project's success.

On the server side, the system is powered by Django, a high-level Python web framework. Django's design principles of reusability, rapid development, and clean, pragmatic code make it an ideal choice for handling the back-end of the system. It enables us to manage user authentication, access control, database interactions, and the business logic of the application effectively.

## **Technology Stack and Frameworks:**

### **Django (Back-end):**

- Django is the backbone of the system, serving as the web framework responsible for managing the server-side logic.
- It provides a secure and efficient way to handle user authentication, database interactions, and request processing.
- Django's ORM (Object-Relational Mapping) simplifies database operations, allowing us to use SQL to interact with the database seamlessly.

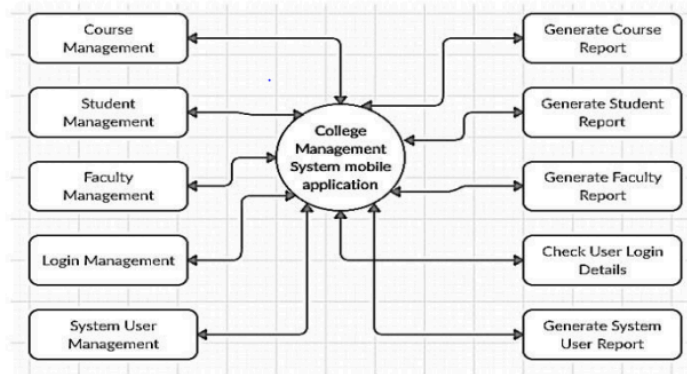
### **Bootstrap (Front-end):**

- Bootstrap is utilized for the front-end development, offering a comprehensive set of responsive and customizable design components.
- Its grid system and pre-built CSS styles enable us to create a user-friendly and visually appealing interface.
- Responsiveness ensures that the system works flawlessly on various devices, from desktops to mobile devices, enhancing accessibility for users.

### **SQL (Database):**

- SQL serves as the database management system, providing a structured and organized repository for storing and retrieving data.
- It plays a crucial role in managing class schedules, exams, attendance records, and institute-related information efficiently and reliably.





**Fig 1**  
SCHEDULE MANAGEMENT SYSTEM FLOWCHART

### Algorithm and Process Design:

The "Schedule Management System" is underpinned by carefully designed algorithms and logical processes that enable efficient class scheduling, exam management, and attendance tracking. Here's an overview of the algorithmic and process design for these core functions:

#### Class Scheduling:

- **Input Requirements:** The system receives input data, including class schedules, available classrooms, and faculty availability.
- **Algorithm for Conflict Resolution:** To prevent scheduling conflicts, the system employs a scheduling algorithm that considers factors like room availability, faculty workload, and preferred timings. It uses techniques such as constraint satisfaction and backtracking to find optimal solutions.
- **Generation of Schedule:** Based on the resolved conflicts, the system generates a class schedule for each academic term, considering factors like room allocation, faculty assignments, and student preferences.
- **Interactive User Interface:** The system provides an interactive interface where administrators can make manual adjustments if needed, ensuring the final schedule is acceptable to all stakeholders.

#### Exam Management:

- **Input Requirements:** Input data includes exam dates, available exam rooms, and student information.
- **Algorithm for Conflict Avoidance:** An algorithm is employed to avoid scheduling multiple exams for the same student at the same time. It considers constraints, such as room capacity and faculty availability.
- **Exam Scheduling:** The system uses a scheduling algorithm to allocate exam dates, times, and rooms. It prioritizes minimizing conflicts while ensuring fair distribution of exams.

### Attendance Tracking:

- **Input Data:** The system takes input data from faculty attendance records and student check-ins.
- **Real-time Attendance Updates:** It maintains real-time attendance records by capturing student check-ins and faculty attendance updates. The algorithm ensures that attendance data is up-to-date.
- **Automatic Notifications:** The system automatically notifies administrators and relevant parties if attendance falls below a certain threshold, triggering intervention.
- **Reporting and Analysis:** It provides tools for generating attendance reports and performing analysis to identify patterns and trends.

### Design Choices for Optimization:

- **Efficiency:** The system optimizes efficiency by using data structures and algorithms that allow for fast scheduling and real-time updates.
- **Scalability:** The design accommodates growth by ensuring that scheduling algorithms can handle an increasing number of classes, exams, and attendees.
- **User-friendliness:** The user interface is designed to be intuitive, reducing the learning curve for administrators and faculty members.
- **Error Handling:** Robust error-handling mechanisms are in place to address unexpected scenarios, such as server issues or data discrepancies.

The algorithmic and process design ensures that the "Schedule Management System" optimally manages core functions, making it a valuable tool for educational institutions. It not only prevents conflicts but also provides a user-friendly and efficient experience for all stakeholders.

### Details of Hardware & Software:

For the efficient hosting and operation of the "Schedule Management System," the following hardware and software requirements are necessary:

#### Hardware Requirements:

- **Server or Hosting Environment:** A server or hosting environment with adequate processing power, memory, and storage capacity to handle web application requests efficiently.
- **Storage:** Sufficient storage space to store database records and system files.

#### Software Requirements:

- **Operating System:** A server-grade operating system such as Linux (e.g., Ubuntu, CentOS) or Windows Server to host the web application.
- **Web Server:** A web server software like Apache, Nginx, or Microsoft Internet Information Services (IIS) to handle HTTP requests and serve web pages.
- **Database Management System:** SQL database management system (e.g., MySQL, PostgreSQL, or SQLite) for storing and retrieving data efficiently.
- **Development Tools:**
  - **IDE (Integrated Development Environment):** Visual Studio Code (VSCode) or any preferred IDE for coding and development.
  - **Python:** The Python programming language for developing the back-end using Django.
  - **Django:** The Django framework to build the web application.
  - **Front-end Technologies:** HTML, CSS, jQuery, Bootstrap, and JavaScript for creating the user interface and enhancing user experience.
  - **Version Control:** A version control system like Git to track code changes and collaborate with a development team.
  - **Additional Libraries and Dependencies:** Various Python libraries and packages that are specific to your project, such as those for database connectivity, user authentication, and any other project-specific functionalities.
  - **Security Tools:** Security tools and practices to protect the system from vulnerabilities and threats, including firewalls, SSL certificates, and best coding practices.
  - **Web Browser Compatibility:** Ensure that the system is compatible with modern web browsers, such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
  - **Testing Tools:** Tools for automated testing and quality assurance, such as Django's testing framework or third-party testing libraries.

## Conclusion:

In conclusion, the "Schedule Management System" has effectively addressed the challenges in educational management. It has streamlined operations, enhanced user experience, and improved data accuracy and security.

### **Future Work:**

Moving forward, the system should focus on additional features, scalability, security enhancements, user feedback integration, mobile accessibility, and integration with external systems to further elevate its role in educational institutions.

### **References:**

1. M.Mahalakshmi, S.Gomathi and, S.Krithika, "Event Management System", 2016
2. Thomas, O. Reference Model Management, In Kelley, G. (Ed.), Selected Readings on Information Technology Management: Contemporary Issues, Hershey, New York, 2008.
3. Filman, R.E. et al. (Eds.) . Aspect-Oriented Software Development, Addison – Wesley Professional, 2004.
4. Parnas, D. L. On the Criteria To Be Used in Decomposing Systems into Modules, Communications of the ACM, 1972
5. Sandeep Misal, Segar Jadhav, Tushar Jore , Archana Ugale , "Event Management System.
6. Roozbeh Derakhshan, Maria E. Orlowska and Xue Li, "RFID Data Management: Challenges and Opportunities", IEEE International Conference on RFID, 2007.
7. Theocharis, 2008, Special event management and event marketing: A case study of TKBL All Star 2011 in Turkey.
8. H. Sharma[2007-17] Event Education [Online].
9. Available:<http://www.eventeducation.com>.