WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Exam Seating Arrangement System

Yukth P[1], Issa Syed Mahmood[2], Shashank K H[3], Thejus S G[4], Mrs. Darshini M S[5]

[1-4] Student, [5]Guide, Assistant Professor
Department of CSE ATME College of Engineering
13th Kilometer, Mysuru-Kanakapura-Bengaluru Road, Mysuru - 570 028

ABSTRACT

The Exam Seating Arrangement System is an advanced digital platform developed to automate and simplify the process of assigning seats to students during examinations in schools, colleges, and universities. Traditional seating allocation is often handled manually, which is labour- intensive, time-consuming, and prone to human errors such as duplication, unfair placement, and incomplete records. As student strength increases and examination halls become distributed across multiple locations, the need for an intelligent system to manage seating efficiently becomes essential.

This system integrates institutional data including student details, roll numbers, course information, hall capacities, and examination schedules to automatically generate accurate seating arrangements. Using structured algorithms, the system sorts students, distributes them evenly to prevent crowding, and allocates seats in a fair and transparent manner that minimizes opportunities for malpractice. The platform also supports multiple exam scenarios, such as semester exams, internal assessments, supplementary exams, and combined seating for students of different courses.

The Exam Seating Allotment System provides features such as real-time data updates, downloadable seating charts, hall-wise reports, and invigilator instructions. Administrators can easily modify seating plans, monitor hall capacity utilization, and print or share seat allotment lists. By reducing manual workload and eliminating common allocation errors, the system improves accuracy, saves time, and enhances the overall examination management process. Ultimately, this solution ensures a smooth, organized, and secure examination environment, promoting fairness and operational efficiency within academic institutions.

1. INTRODUCTION

1.1 Problem Statement

The manual process of preparing exam seating arrangements is inherently time-consuming, error-prone, and difficult to manage, especially when handling large student groups or multiple examination halls. Traditional methods, such as using paper charts or basic spreadsheets, are often rigid and unsuitable for dynamic changes like last-minute student additions or hall revisions. These manual systems also suffer from high risks of human error, including duplication and the misplacement of students, while simultaneously increasing the chances of malpractice due to poor planning in seat distribution. Consequently, there is a critical need for an automated system that can efficiently generate accurate seating plans, reduce administrative workload, and ensure fairness and organization during examinations.

1.2 Objectives

- Automation: To automate the process of assigning seats to students for examinations accurately and efficiently.
- Efficiency: To minimize manual work, human errors, and the time required in preparing seating arrangements.
- Resource Utilization: To utilize available hall capacities effectively and ensure fair and organized seating distribution.
- Security: To reduce opportunities for malpractice by generating systematic and unbiased seating layouts.
- Reporting: To provide easily printable and shareable seating charts and hall-wise reports for smooth exam management.
- Flexibility: To enable quick updates or modifications to seating plans when required.

1.3 Overview

- Digital Solution: The system is a digital solution designed to automate the assignment of seats to students during examinations.

- Data Utilization: It utilizes student data, hall capacities, and exam schedules to generate accurate and organized seating plans.
- Efficiency and Fairness: The system reduces manual workload and minimizes errors while helping to prevent malpractice through fair seat distribution.
- reporting and Management: It supports smooth exam administration by providing printable seating charts, hall-wise reports, and easy options for modification.

1.4 Scope

- Automatic Allocation: The system covers the automatic allocation of seats to students based on roll numbers, hall capacity, and exam schedules.
- Flexibility and Updates: It allows for the modification of seat plans, the addition of new rooms or students, and real-time updates when required.
- Versatility: The system can be used for various types of examinations, including internal assessments, semester exams, supplementary exams, and combined seating.

2 LITERATURE SURVEY

Literature survey is a critical analysis of a portion of the published body of knowledge available through the use of summary, classification, and comparison of previous research studies, reviews of literature, and journal articles. A literature survey examines the current scholarly work available on a particular subject, perhaps within a given time period. It is the summary and synthesis of material gathered from various sources and organized to address an issue, research objective, or problem statement.

2.1 Survey Papers

2.1.1 AI Powered Automated Exam Hall Allocation And Management System

- Source/Year: International Journal of Science and Research (IJSART), 2025.
- Proposed System: The authors developed a web-based system using Python (Flask) and MySQL. It automated the allocation of students to halls and assigned invigilation duties to staff. It used a deterministic constraint-based algorithm to ensure fairness.
- Limitation: The system focused heavily on staff allocation but lacked a feature for handling dynamic "Room Overflow" logic efficiently where students needed to be split across multiple partial rooms.

2.1.2 Web Application Based Exam Hall Seating Management System

- Source/Year: IEEE International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022.
- Authors: Jitha K, Fathima A.
- Proposed System: This paper proposed a centralized web portal where admins could upload student lists via CSV. It generated a static table view for students to check their seats online before the exam to avoid crowding at notice boards.
- Limitation identified: The system did not support "Interleaved Seating" (Zipper Logic). Students of the same branch were often seated sequentially, which did not fully address the risk of malpractice.

2.1.3 Smart Seating System for Universities

- Source/Year: Proc. IEEE International Conference on Computing, Communication and Automation, 2021.
- Author: M. Patel.
- Proposed System: A smart system that integrated with biometric attendance. It allowed students to scan a QR code to find their seat.
- Limitation identified: The hardware cost for implementation was too high for standard colleges, and the system required active internet connectivity for every student device, which is not always feasible.

2.1.4 Automated Students Examination Seat Allocation Using Linear Congruential Generator

- Source/Year: International Journal of Computer Trends and Technology (IJCTT), 2021.

- Proposed System: This study used a randomization algorithm (Linear Congruential Generator) to shuffle student seats completely randomly to prevent cheating.
- Limitation identified: Complete randomization often led to confusion for invigilators when collecting answer scripts, as roll numbers were not in any predictable order.

3 SYSTEM REQUIREMENTS AND SPECIFICATIONS

3.1 Hardware Requirements

The system is lightweight and can run on most standard computers.

- Processor: Intel Core i3 (5th Gen) or higher / AMD Ryzen 3 equivalent.
- RAM: 4 GB minimum (8 GB recommended for smoother performance).
- Hard Disk Space: 500 MB free space (for application code, database, and generated PDFs).
- Input Devices: Standard Keyboard and Mouse.
- Display: 1366 x 768 resolution or higher (LED/LCD Monitor).

3.2 Software Requirements

These are the tools and environments required to run the application.

- Operating System: Windows 10/11, Linux (Ubuntu), or macOS.
- Programming Language: Python 3.10 or higher.
- Web Framework: Flask (Micro-framework).
- Database: SQLite (Default with Python) or SQLAlchemy ORM.
- IDE / Text Editor: Visual Studio Code (VS Code) or PyCharm.
- Web Browser: Google Chrome, Microsoft Edge, or Mozilla Firefox (latest versions).

3.3 Functional Requirements

These define what the system actually *does*.

1. User Authentication:

- Secure Login system for Administrators and Staff.
- Role-Based Access Control (Admin has full access; Staff has limited access).
- Secure Logout functionality.

2. Input Management:

- Drag-and-Drop File Upload: Support for uploading Student Lists via PDF, Excel (.xlsx), and CSV formats.
- Manual Entry: A text area to copy-paste student names directly.

3. Seating Generation Logic:

- Automated Allocation: Automatically assigns seat numbers based on room capacity.
- Zipper Logic (Interleaving): Capable of mixing two different subjects (Subject A and Subject B) in an alternating pattern to prevent malpractice.
- Branch Constraints: Option to prevent students from the same branch (e.g., CSE) from sitting sequentially.

4. Resource Management (CRUD):

- Manage Rooms: Add, delete, and view exam halls with specific columns and capacities.
- Manage Exams: Define upcoming exams and dates.

- Manage Staff: Add invigilators to the database.

5. Output & Reporting:

- Visual Map: On-screen grid display of the seating arrangement.
- PDF Generation: One-click download of a professional seating chart PDF for printing.
- Public Search: A student-facing portal to find seat numbers using USN/Name without logging in.

4. SYSTEM ANALYSIS

4.1 Introduction

System analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. It helps in understanding the specific needs of the system and ensuring that the proposed solution meets the user's requirements effectively. This chapter details the analysis of the existing manual system, the proposed automated system, and the feasibility of the project.

4.2 Existing System

In the current manual system, the examination committee relies on spreadsheets (Excel) or physical registers to organize seating.

- Process: The admin collects student lists from various departments, manually counts the capacity of each room, and assigns roll numbers one by one.
- Disadvantages:
    1. Time-Consuming: Allocating seats for thousands of students takes days of effort.
    2. High Error Rate: There is a high probability of assigning duplicate seats or missing a student.
    3. Malpractice Vulnerability: Manually shuffling students to ensure different branches or subjects are seated adjacently is difficult and error-prone.
    4. Static Reporting: Making last-minute changes (e.g., a broken bench) requires re-doing the entire chart.

4.3 Proposed System

The proposed Exam Seating Arrangement System (ExamSeater) is a web-based automation tool. It replaces the manual workflow with an intelligent algorithm.

- Automation: The system accepts student lists via PDF/Excel and automatically distributes them across rooms.
- Zipper Logic: It implements an "Interleaved Seating Strategy" where students from two different subjects are mixed (Seat 1: Subject A, Seat 2: Subject B) to prevent malpractice.
- Centralized Database: All data regarding rooms, exams, and history is stored in a secure SQLite database.
- Accessibility: Includes a public portal for students to find their seats remotely.

4.4 Feasibility Study

A feasibility study determines if the system is viable to develop and implement.

- 4.4.1 Technical Feasibility: The project uses Python and Flask, which are open-source, stable, and widely supported technologies. The database (SQLite) requires no server configuration, making it lightweight and easy to deploy on any standard college computer. The technical requirements are minimal, ensuring high feasibility.
- 4.4.2 Operational Feasibility: The user interface is designed with a modern "Glassmorphism" aesthetic, making it intuitive for non-technical staff. The inputs are simple (dropdowns and file uploads), requiring no specialized training for the exam cell staff.
- 4.4.3 Economic Feasibility: The system is built using entirely open-source libraries (Flask, ReportLab, Pandas). There are no licensing costs involved, making it a highly cost-effective solution for the institution.

4.5 Functional Requirements

These define the specific behaviors and functions of the system.

1. Authentication Module:
    - The system must validate user credentials (Username/Password) against the database.
    - It must support role-based access (Admin has full control; Staff has limited access).
    - Passwords must be hashed (encrypted) before storage.
2. Input Processing Module:
    - File Parsing: The system must accept student lists in .pdf, .csv, or .xlsx formats.
    - Data Extraction: It must intelligently identify columns for "Name", "USN", and "Branch" automatically.
3. Core Generation Module (The Algorithm):
    - Zipper Logic: The system must be able to interleave two student lists to alternate subjects.
    - Branch Constraint: If enabled, the system must prevent students of the same branch from sitting sequentially.
    - Blocked Seats: The system must skip specific seat numbers defined as "blocked/broken" by the admin.
4. Reporting & Output Module:
    - PDF Generation: The system must generate a downloadable PDF with a header, exam details, and a formatted seating table.
    - Visual Map: The system must render a visual grid layout of the room on the screen for quick verification.
5. Search Module:
    - The system must provide a public-facing page where students can search by Name or ID to retrieve their Room and Seat Number.

4.6 Non-Functional Requirements

1. Performance: The seating allocation algorithm should process a batch of 100 students in under 3 seconds.
2. Scalability: The database design must allow for an unlimited number of exams and rooms to be added over time.
3. Reliability: The system must handle errors gracefully (e.g., if an uploaded file is empty, it should show a clear error message rather than crashing).
4. Security: Unauthorized users must be redirected to the login page if they attempt to access internal URLs.

5. METHODOLOGY

5.1 System Architecture

The Exam Seating Arrangement System (ExamSeater) follows the standard Model-View- Controller (MVC) architectural pattern, adapted for the Flask web framework. This architecture separates the application into three interconnected components, ensuring modularity and ease of maintenance.

The architecture consists of the following layers:

1. Presentation Layer (View):
    - User Interface: Built using HTML5, CSS3, and JavaScript with a custom "Glassmorphism" design.
    - Templates: Jinja2 templates (dashboard.html, seating.html) render dynamic data sent from the server.
    - Interaction: Users (Admins/Staff) interact with forms to upload student lists (PDF/Excel) and define exam parameters.
2. Application Logic Layer (Controller):
    - Web Server: The app.py (Flask) acts as the controller. It intercepts user requests (e.g., /generate, /login).
    - Data Processing: Uses the Pandas library to parse uploaded Excel/CSV files and PyPDF2 for PDF text extraction.
    - Core Algorithm: The "Zipper" logic processes the student lists, sorting them by branch or name, and interleaving them to prevent malpractice.
    - PDF Engine: The ReportLab library draws the final seating chart on a PDF canvas.
3. Data Storage Layer (Model):

- Database: A relational SQLite database stores all persistent data.
- ORM: SQLAlchemy is used to map Python classes (User, Room, Exam, History) to database tables, abstracting raw SQL queries.

5.2 Data Flow Diagram (DFD)

The data flows through the system in the following sequence:

4. Admin Input: The administrator logs in and submits the *Exam Date*, *Subject*, *Room*, and

Student List (via file upload).

5. Validation: The system validates the file format and checks for empty inputs.
6. Processing:
    - The system extracts names and USNs from the file.
    - It checks the "Blocked Seats" configuration for the selected room.
    - It applies the "Interleaved/Zipper" sort to mix subjects.
7. Storage: The final arrangement is committed to the SeatAssignment table for public search and the History table for archival.
8. Output: The system renders a Visual Map on the dashboard and generates a downloadable PDF.

5.3 Seating Algorithm Logic

(The "Zipper" Method) To ensure fairness and reduce malpractice, the system uses a custom algorithm based on Python's itertools.zip_longest.

- Step 1: The system accepts two distinct lists: *List A* (Subject 1) and *List B* (Subject 2).
- Step 2: Both lists are sorted alphabetically or by branch (if the "Prevent Branch Conflict" constraint is enabled).
- Step 3: The algorithm iterates through both lists simultaneously:
    - *Iteration 1:* Assigns Student A1 to Seat 1.
    - *Iteration 1:* Assigns Student B1 to Seat 2.
    - *Iteration 2:* Assigns Student A2 to Seat 3.
    - *Iteration 2:* Assigns Student B2 to Seat 4.
- Step 4: If a seat is marked as "Blocked" in the database, the algorithm skips that index and continues to the next available seat.

5.4 Modules Description

The project is divided into four primary modules:

9. Authentication Module: Manages secure user access. It utilizes pbkdf2:sha256 hashing to protect passwords and handles session management for Admins and Staff.
10. Management (CRUD) Module: Allows administrators to Create, Read, Update, and Delete records for Rooms, Exams, and Invigilators. It ensures the infrastructure data is always up-to-date.
11. Generator & Reporting Module: The core engine that processes student data. It handles the logic for visual grid rendering and PDF document generation.
12. Public Search Module: A lightweight, read-only interface that allows students to query their seat allocation using their Name or USN without requiring authentication.

5.5 Proposed System

The proposed Exam Seating Arrangement System is a web-based automation tool designed to replace the traditional manual process of exam planning. It is a centralized digital solution that utilizes student data, hall capacities, and exam schedules to automatically generate accurate and organized seating plans.

Unlike existing spreadsheet-based methods, the proposed system is built on a robust Python (Flask) framework with a secure SQLite database. It introduces an intelligent "Zipper

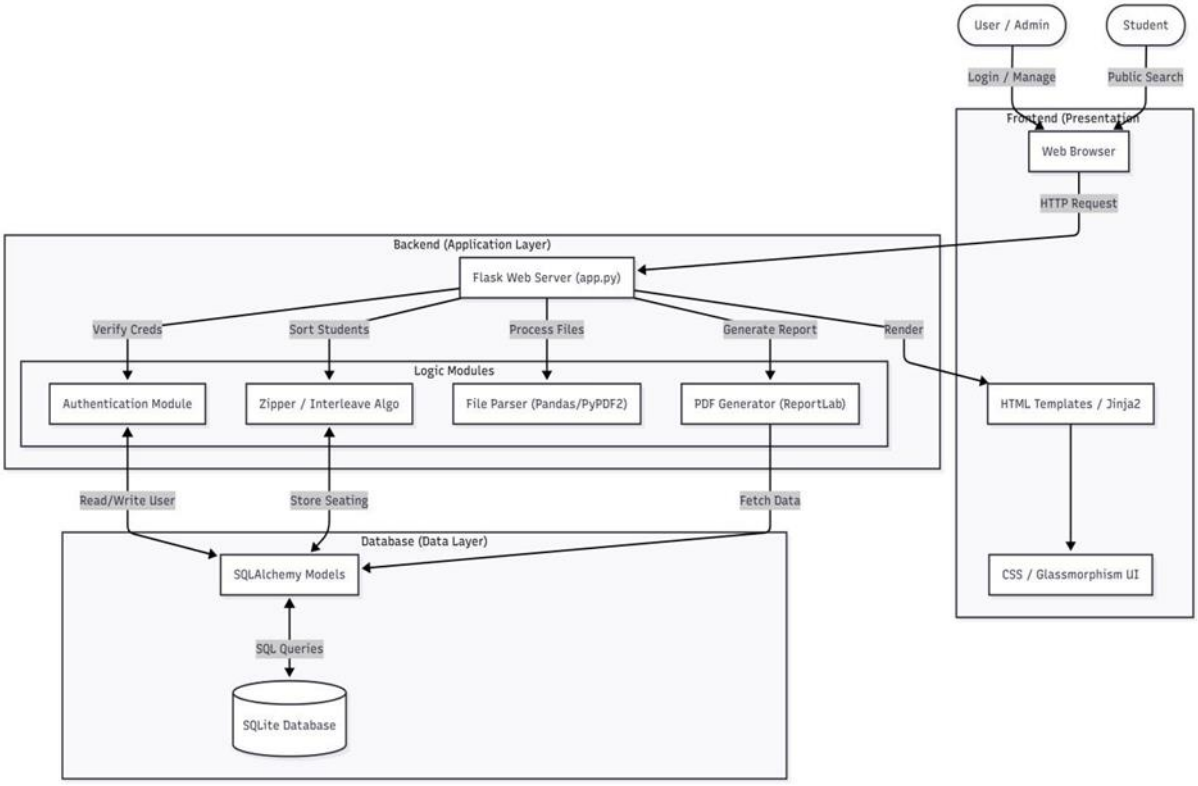
Algorithm" (Interleaved Seating) that automatically mixes students from two different subjects—placing a student of Subject B between two students of Subject A—to effectively minimize the risk of malpractice.

Key Features of the Proposed System:

- Automation: It completely automates the assignment of seats based on roll numbers and room capacity, significantly reducing manual workload and human errors.
- Dynamic Handling: The system allows for the modification of seat plans, the addition of new rooms, and the handling of "blocked" (damaged) seats in real-time.
- Visual Mapping: It provides a "Cinema-style" visual grid map of the exam hall, allowing administrators to verify layouts at a glance.
- Public Accessibility: A dedicated Student Search Portal enables students to find their allocated room and seat number remotely using their USN, preventing overcrowding at notice boards.
- Reporting: It generates professional, printable seating charts and hall-wise reports in PDF format for smooth exam administration.

By integrating these features, the system ensures a fair, secure, and efficient examination environment, addressing the scalability and management issues found in manual systems.

System architecture



6 SNAPSHOTS

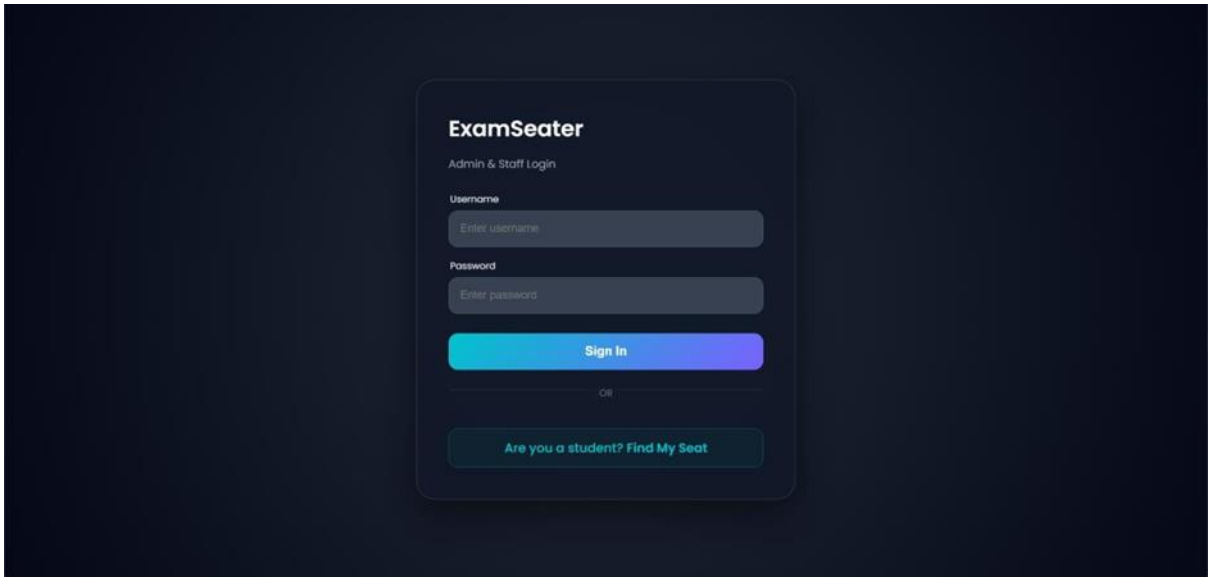


Fig 6.1: Login page

Allows multiple users to login in with PBKDF2 (Password-Based Key Derivation Function 2).

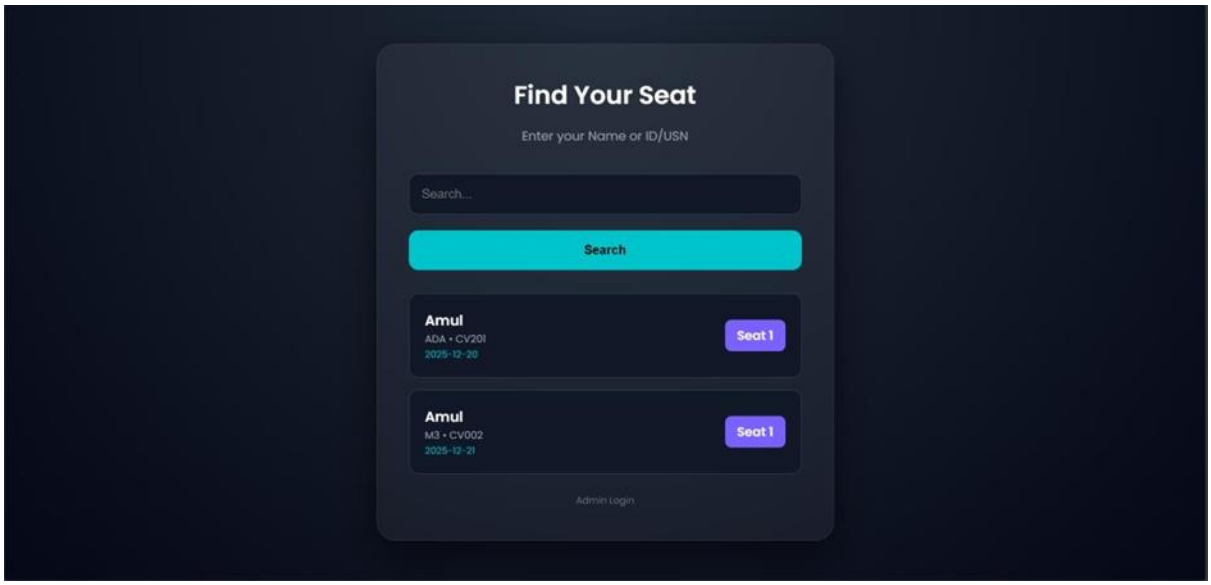


Fig 6.2: Find My Seat

Allows students to find their seating without any authentication.

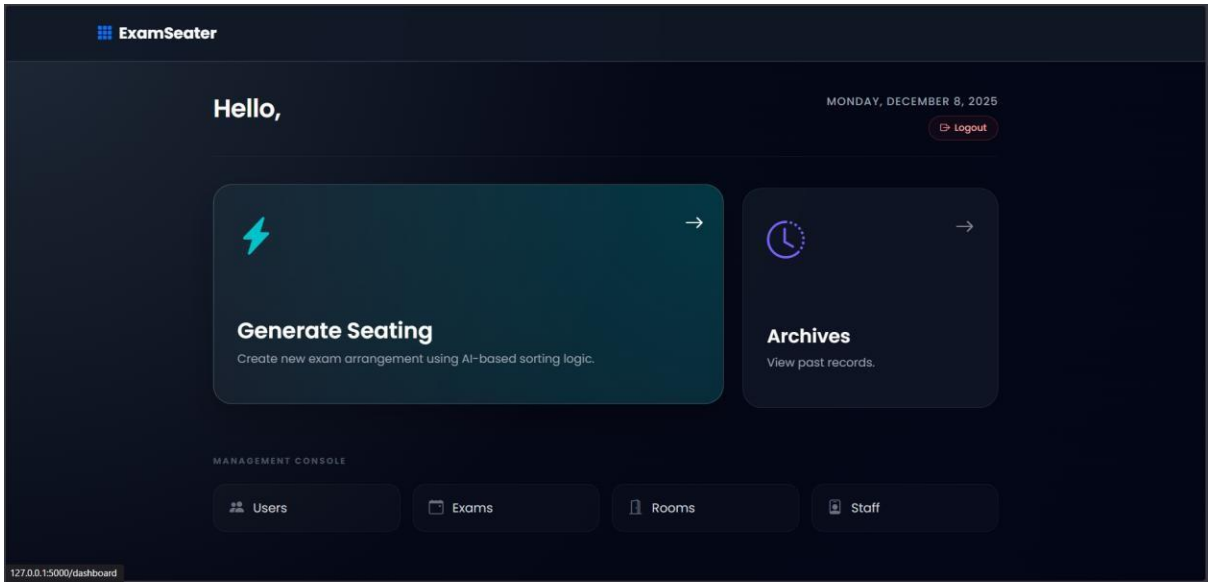


Fig 6.3: Home Page

Home page or the dashboard where all functions can be accessed.

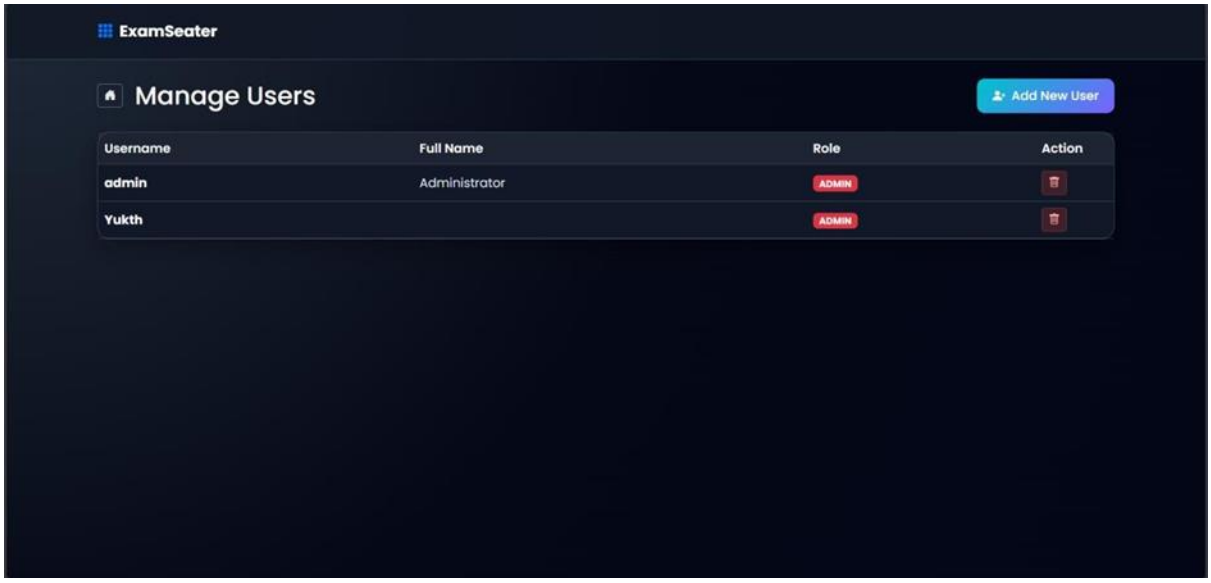


Fig 6.4: Manage Users

Allows us to have multiple users with various role such as admin and staff.

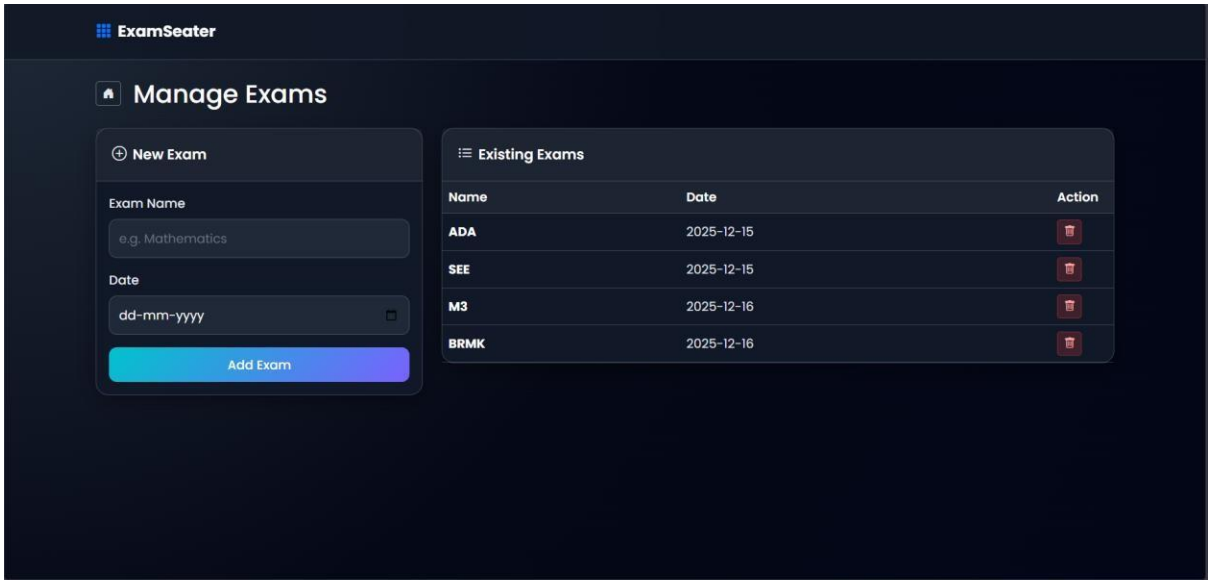


Fig 6.5: Manage Exams

Page where we can add or delete exams.

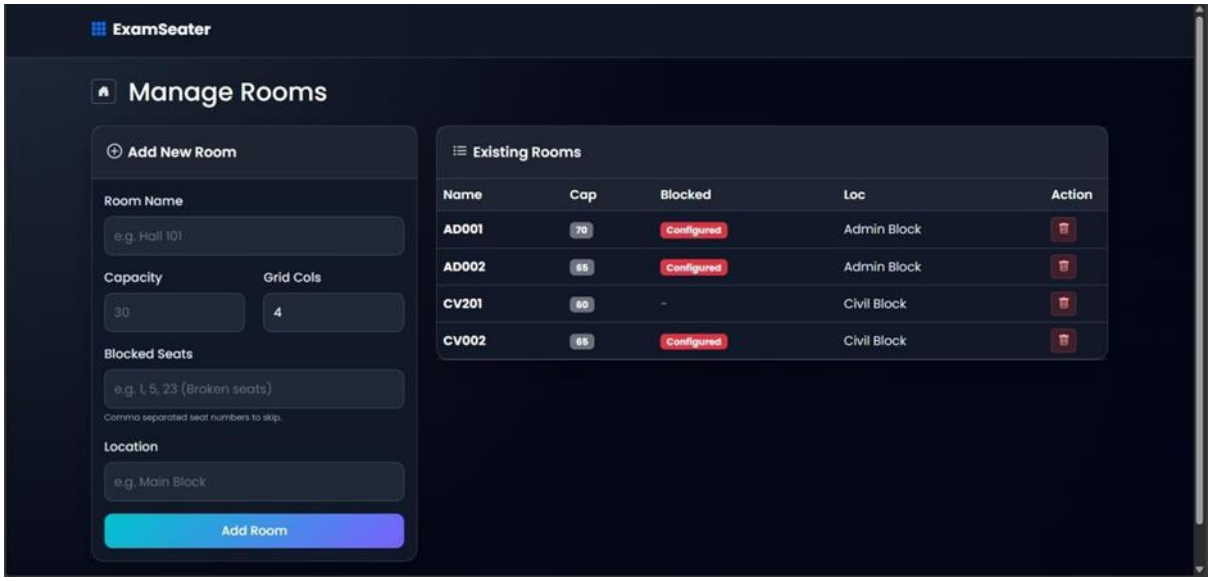


Fig 6.6: Manage Rooms

Allows us to manage exam rooms and also lets us to block damaged or unavailable seats.

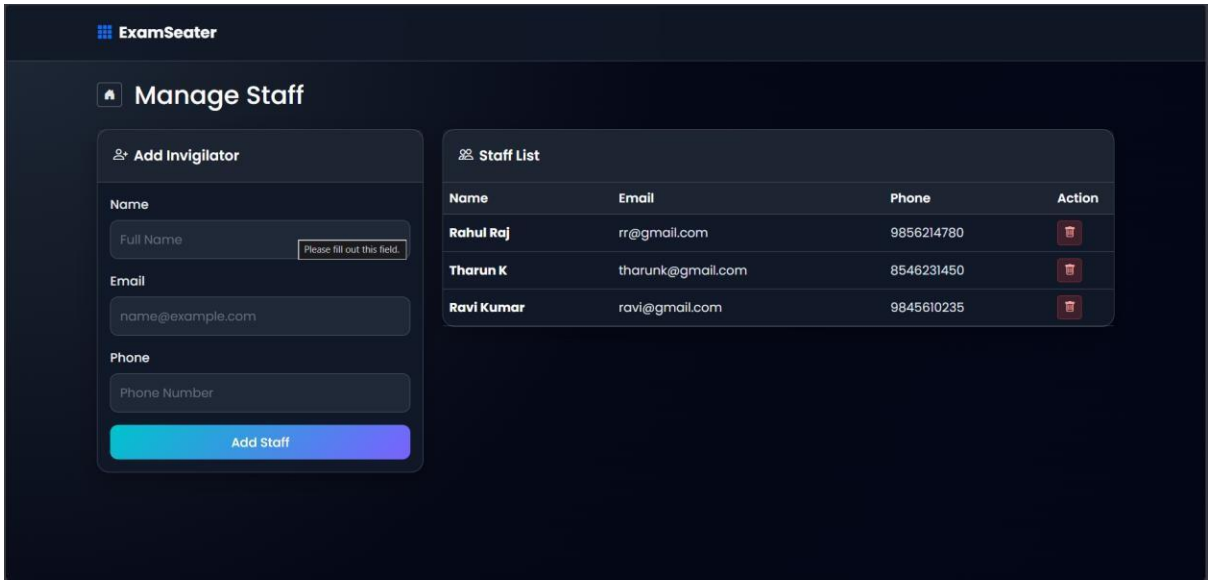


Fig 6.7: Manage Staff

Manage available invigilators for various exams with their details.

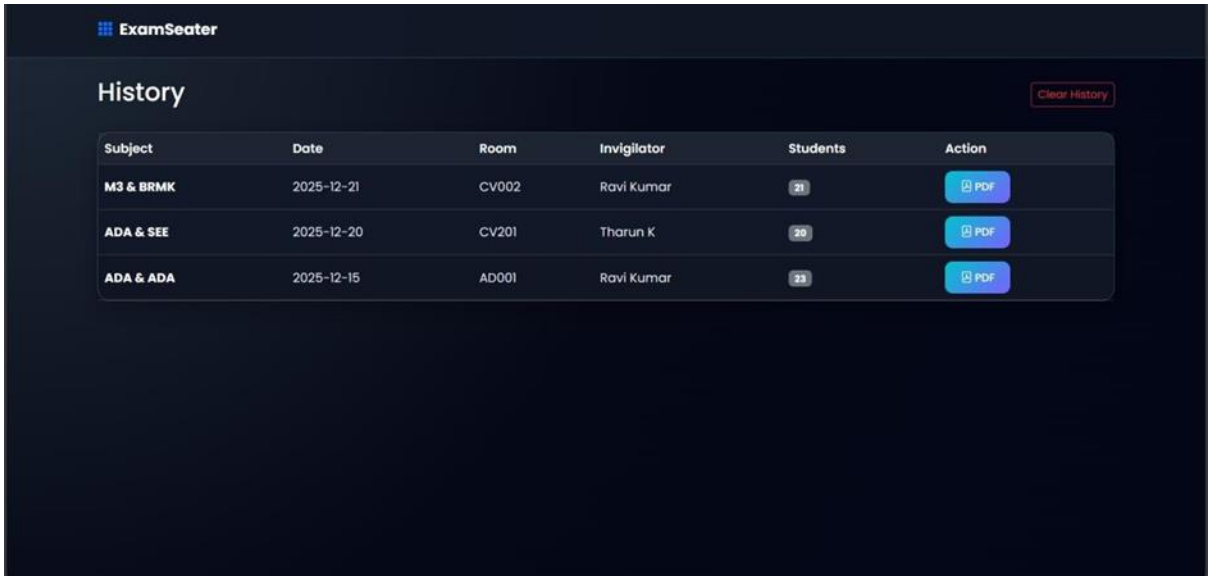


Fig 6.8: History

An archive of all the previous seating arrangements.

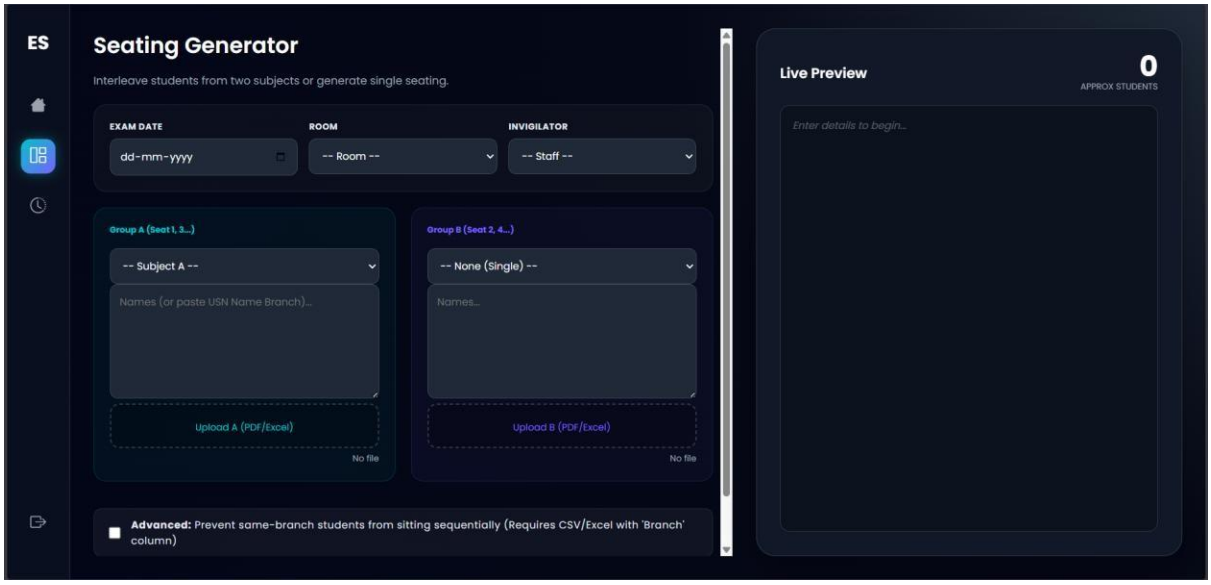


Fig 6.9: Seating Generator

Allows us to generate seating and accepts files in both .csv and .pdf formats. Also shows the live preview next to the seating.

EXAM SEATING ARRANGEMENT

DATE: 2025-12-20 **ROOM: CV201**

SUBJECT: ADA & SEE **INVIGILATOR: Tharun K**

SEAT	STUDENT	SEAT	STUDENT
1	Amul (ADA)	2	Ashwin (SEE)
3	Asim (ADA)	4	Diganth (SEE)
5	Bhuvan (ADA)	6	Govind (SEE)
7	Geera (ADA)	8	Kishan (SEE)
9	Hisam (ADA)	10	Pooja (SEE)
11	Jitin (ADA)	12	Raghav (SEE)
13	Joy (ADA)	14	Rohan (SEE)
15	Kumar (ADA)	16	Ujwal (SEE)
17	Manish (ADA)	18	Vedha (SEE)
19	Tejus (ADA)	20	Yashi (SEE)

Fig 6.10: Seating PDF

The PDF file of the allocated seats with all required information.

7. CONCLUSION AND FUTURE SCOPE

The Exam Seating Arrangement System (ExamSeater) was successfully designed, developed, and tested. The project effectively addresses the critical inefficiencies associated with manual examination planning. By digitizing the workflow, the system eliminates human errors such as double-booking and accidental omission of students.

The implementation of the "Zipper Algorithm" (Interleaved Seating) has proven to be a robust solution for minimizing malpractice by ensuring students of the same subject do not sit adjacent to each other. The system's centralized SQLite database ensures data integrity and security, while the Public Search Portal significantly improves the student experience by allowing remote access to seating details. Overall, the application creates a secure, organized, and stress- free examination environment for both administrators and students.

Future Scope

While the current system covers all primary requirements for exam management, there is significant potential for further enhancements:

1. Mobile Application: Developing a dedicated Android or iOS application would allow students to receive push notifications regarding their exam halls and seat numbers directly on their phones.
2. Email & SMS Integration: The system can be upgraded to automatically email the generated PDF seating charts to invigilators and send SMS alerts to students 30 minutes before the exam.
3. Facial Recognition: Integrating AI-based facial recognition could automate the attendance process, matching the student in the seat against their database photo to prevent impersonation.
4. Cloud Deployment: Hosting the application on cloud platforms like AWS or Azure would allow it to scale globally and handle thousands of simultaneous requests during peak exam periods.
5. Visual Drag-and-Drop: Implementing a drag-and-drop interface for the Visual Seat Map would allow administrators to manually swap students between seats with a simple mouse click.

REFERENCES

1. "AI Powered Automated Exam Hall Allocation And Management System," International Journal of Science and Research (IJSART), 2025.
2. Jitha K. and Fathima A., "Web Application Based Exam Hall Seating Management System," in IEEE International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022.
3. M. Patel, "Smart Seating System for Universities," in Proc. IEEE International Conference on Computing, Communication and Automation, 2021.
4. "Automated Students Examination Seat Allocation Using Linear Congruential Generator," International Journal of Computer Trends and Technology (IJCTT), 2021.
5. R. Kumar, S. Mehta, and P. Roy, "Automation of Seating Arrangement System," International Journal of Computer Applications, vol. 182, no. 25, pp. 1-5, 2019.
6. S. Gupta and A. Sharma, "Digital Exam Management System," International Research Journal of Engineering and Technology (IRJET), vol. 7, no. 6, pp. 2401-2405, Jun. 2020.
7. K. Singh and T. Reddy, "Web-Based Examination Management Module," International Journal of Emerging Technologies in Learning (iJET), vol. 17, no. 12, pp. 85-92, 2022.
8. Flask Documentation. (2024). Pallets Projects. Available: https://flask.palletsprojects.com/
9. ReportLab PDF Library. (2024). User Guide. Available: https://www.reportlab.com/
10. Pandas Development Team. (2024). Pandas Documentation. Available: https://pandas.pydata.org/