EXAM SEAT ARRANGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

SANDHYA J (221801044)
TANUSHRI G V S (221801055)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ANNA UNIVERSITY, CHENNAI

MAY 2025

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**EXAM SEAT ARRANGEMENT SYSTEM**" is the bonafide work of **SANDHYA J (221801044), TANUSHRI G V S (221801055)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. J.M. Gnanasekar
Professor and Head
Department of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Ms. Renuga Devi S
Assistant Professor
Department of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Submitted for the project viva-voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. J.M. GNANASEKAR., M.E., Ph.D., Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor Ms. S. RENUGA DEVI., M.E., Assistant Professor, Department of Artificial Intelligence and Data Science Rajalakshmi Engineering College for their valuable guidance throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

With the increasing number of students appearing for academic examinations, managing examination logistics such as seating arrangements has become a complex and error-prone task. Manual scheduling often results in inefficiencies, duplication of seat numbers, misallocations, and administrative overhead. The Exam Seat Arrangement System is an intelligent, automated solution designed to streamline and optimize the process of assigning seats to students during examinations. Leveraging algorithms for data validation, space optimization, and conflict resolution, the system dynamically allocates seats across examination halls based on parameters such as subject codes, roll numbers, hall capacities, and seating rules. The system integrates database management and user-friendly interfaces to allow administrators to input data, generate reports, and visualize arrangements in real-time. Unlike traditional manual methods, this digital approach minimizes human error, saves time, and enhances transparency and fairness in the exam process. Furthermore, the solution ensures compliance with institutional policies like avoiding seat duplication, maintaining discipline codes (e.g., spacing rules), and accommodating special needs candidates. This research explores the development and deployment of the Exam Seat Arrangement System, evaluating its impact on administrative efficiency, accuracy, and overall examination management.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 GENERAL

The management of academic examinations is a critical and resource-intensive function in educational institutions. Among various logistical challenges, the arrangement of students' seating during exams stands out as a task that demands both precision and efficiency. Traditionally, this process has been handled manually, involving administrative staff creating spreadsheets, checking subject overlaps, managing hall capacities, and avoiding seating conflicts. Such manual methods are not only time-consuming but also prone to errors like seat duplication, subject-mixing, or capacity mismatches. As academic environments become more complex—with growing student populations, multiple exam sessions, and diverse course enrollments—the need for an intelligent and automated seat arrangement system becomes evident.

The **Exam Seat Arrangement System** is a smart, rule-driven application designed to automate the allocation of examination seats for students based on inputs like subject codes, student IDs, hall capacities, and examination schedules. The system reduces human intervention, minimizes errors, and enhances the overall accuracy and speed of seat arrangement. With features such as seating optimization, conflict avoidance, batch-wise allocation, and hall-wise summaries, this system ensures transparency, fairness, and operational efficiency. It also enables administrators to generate printable reports and visual layouts for seamless implementation on exam day.

Furthermore, the system is scalable and flexible to support various exam types including internal assessments, end-semester exams, supplementary exams, and entrance tests. By integrating advanced data structures and algorithmic logic, this solution addresses the growing demands of modern examination planning and bridges the gap between administrative capabilities and logistical complexity.

1.2 NEED FOR THE STUDY

Educational institutions are under constant pressure to manage exams efficiently while maintaining fairness and transparency. With an increasing number of students and courses, scheduling and arranging seating for exams has become a significant administrative burden. Manual methods are not only inefficient but also introduce risks like assigning the same seat to multiple students, violating distance protocols, or placing students from the same course adjacent to each other—leading to academic dishonesty.

Moreover, as institutions shift towards digitization in all facets of operations—admissions, attendance, internal assessments—exam management remains one of the few areas still largely reliant on manual processes. This mismatch between institutional automation and manual seating practices often results in wasted time, student confusion, and administrative stress during examinations.

In this context, a dedicated **Exam Seat Arrangement System** becomes essential. The proposed system addresses the following critical needs:

- **Error Minimization**: Reduces human error by automating seat assignments using predefined constraints.
- **Time Efficiency**: Saves considerable time for staff by generating arrangements in seconds instead of hours.
- **Rule Enforcement**: Ensures policies such as spacing rules, special candidate considerations, or mixed seating (cross-subject) are respected.
- **Scalability**: Easily manages thousands of students across multiple departments and buildings.
- **Transparency and Record-Keeping**: Facilitates auditability and institutional reporting by generating clear records and seat plans.

This study is thus aimed at solving a longstanding administrative challenge through an intelligent software application that enhances the examination management experience for both staff and students.

1.3 OVERVIEW OF THE PROJECT

The **Exam Seat Arrangement System** is an application developed to automate the seating process for academic examinations. It offers a streamlined platform for examination coordinators to upload student data, course details, examination schedules, and available seating capacity across rooms or halls. Based on these inputs, the system intelligently assigns seats to students while adhering to constraints such as subject group separation, hall capacity limits, and spacing rules.

The system is built with a modular architecture, comprising data import and validation modules, seating logic algorithms, conflict-checking mechanisms, and reporting engines. Users can log in via a secure interface to create or modify exam arrangements, view hall-wise and student-wise allocations, and export the final seat plan for printing or display.

Features include:

- **Dynamic Seat Allocation**: Based on input datasets and real-time seat availability.
- **Multi-Exam Scheduling Support**: Handles overlapping exams, backlog exams, and departmental differences.
- **Conflict Detection and Correction**: Highlights seat duplication, overlapping schedules, or room capacity violations.
- **User Dashboard**: Displays seat maps, student arrangements, and hall summaries.
- **PDF and Excel Export**: Enables generation of student slips, hall-wise charts, and invigilator briefs.

The system is implemented using modern programming techniques and database design practices, ensuring performance, scalability, and ease of maintenance. It significantly reduces the burden on administrative staff, ensures consistent and fair arrangements, and improves the overall credibility of examination management practices.

1.4 OBJECTIVES OF THE STUDY

The primary objectives of this project are:

1. To design and develop an intelligent exam seat arrangement system that automates the process of assigning examination seats to students.
2. To ensure that seating rules and constraints (like capacity, cross-department seating, and distancing rules) are strictly adhered to.
3. To create a user-friendly interface for administrative staff to upload exam data, monitor seat allocations, and generate printable layouts.
4. To prevent common issues such as double allocation of seats, empty hall spaces, and student misplacement.
5. To reduce the time and effort required to manually organize and verify exam seating.
6. To enable generation of various reports including:
    - Student-wise seating slips
    - Hall-wise seating charts
    - Invigilator summaries
7. To allow customization of arrangements based on department preferences or physical hall layouts.
8. To ensure data security and allow only authorized personnel to access or modify seating plans.
9. To evaluate the system's accuracy and effectiveness through testing with real-world exam data.
10. To support future enhancements such as barcode scanning, integration with student attendance, or AI-based cheating pattern analysis.

CHAPTER II

REVIEW OF LITERATURE

2.1 INTRODUCTION

Effective management of academic examinations is essential to uphold the integrity and efficiency of educational institutions. While technological advances have modernized aspects like online assessments and digital gradebooks, physical examinations still require manual seating arrangement, a task that is repetitive, time-consuming, and prone to errors. The literature shows a growing recognition of the challenges posed by manual seating plans, especially in institutions with large student populations and complex course structures.

Researchers and education technologists have explored multiple solutions, from simple spreadsheet-based templates to automated scheduling software using optimization algorithms. However, most existing solutions fail to incorporate dynamic conflict resolution, cross-course separation, or adaptive hall usage. With the increasing emphasis on digital transformation, especially post-pandemic, there's a push toward intelligent and scalable exam arrangement systems that not only automate basic tasks but also ensure rule adherence, fairness, and transparency.

The reviewed literature reveals the limitations of existing systems, the potential for optimization-based seating algorithms, and the necessity for adaptable, modular solutions capable of addressing diverse institutional constraints. This chapter provides a comprehensive look into the technological, algorithmic, and institutional developments that form the basis for this study.

2.2 FRAMEWORK OF LITERATURE REVIEW

1. Manual Examination Seating: Challenges and Drawbacks

Several academic publications and administrative reports underline the inefficiencies in traditional exam seat planning. In a study conducted by the University of Madras

(2017), it was found that more than 60% of administrative errors during exams were linked to improper seating or hall assignments. Common issues include double allocation of seats, student misplacement, and uneven room utilization. These problems often stem from manual cross-checking of student rosters and handwritten hall plans.

2. Spreadsheet-Based Seat Allocation: A Transitional Step

Many institutions moved toward spreadsheet-based planning using Excel or Google Sheets to manage student roll numbers, exam subjects, and hall assignments. Though this method improves visibility and record-keeping, it lacks logic-based automation and still requires substantial manual intervention. A study published in *IEEE Access* (2018) noted that while spreadsheet tools can aid small institutions, they fail to scale for universities with thousands of students across multiple programs and departments.

3. Rule-Based and Optimization Algorithms

Recent research focuses on using rule-based systems and mathematical optimization models such as linear programming (LP) and constraint satisfaction problems (CSPs) to automate seating. For instance, Dr. A. Kumar’s work (2019) on "Automated Scheduling and Seating Using Constraint Algorithms" proposed an LP-based model that significantly reduced seating conflicts by 85%. These methods use predefined rules (e.g., students from the same course should not sit adjacent) and search for optimal solutions through algorithmic iterations.

4. Web-Based Examination Planning Tools

The evolution of web technologies enabled the development of web-based platforms for exam scheduling. Projects like ExamSys (presented at the ICACIE 2020 Conference) allow users to input student data, course codes, and room configurations through an interface and generate seating plans. However, many such systems are

rigid, lacking adaptability to institutional-specific policies or real-time modification capabilities.

5. Integration of Seating with Student Information Systems (SIS)

Advanced systems attempt to integrate seat planning directly into Student Information Systems (SIS) or Learning Management Systems (LMS). Research from the *International Journal of Emerging Trends in Engineering* (2021) proposed a modular system that pulls exam schedules and student registrations from existing SIS databases and dynamically updates the seating plan. While this increases efficiency, it introduces challenges in system interoperability and data consistency.

6. Mobile and Real-Time Solutions

Recent explorations include mobile applications and real-time seat scanners. These systems allow invigilators to scan student IDs on entry and verify seat numbers dynamically. While these systems are not yet widespread, they highlight the trend toward interactive and mobile-friendly exam technologies. Literature points to the benefits of real-time seat validation in reducing confusion and ensuring accurate attendance marking.

7. Summary of Gaps in Literature

Despite progress in automation and algorithmic planning, the literature highlights several gaps:

- Lack of modularity and adaptability in current systems.
- Poor integration with room layout mapping and physical seating constraints.
- Limited support for seating special-needs students or handling emergency scheduling changes.
- Absence of visual dashboards for real-time monitoring. These gaps underscore the need for a dedicated, rule-driven, modular **Exam Seat Arrangement System** that combines database management, algorithmic intelligence, and administrative usability.

CHAPTER III

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

Despite the growing digitization in academic operations, most institutions still rely on manual or semi-automated methods for exam seat arrangement. These systems typically involve basic data entry into spreadsheets, followed by manual planning of student placements within examination halls. Although familiar and flexible, such methods have significant drawbacks in terms of efficiency, scalability, and accuracy.

1. Manual Method:

Overview:

The traditional approach involves exam coordinators listing students by subject or department, then manually assigning them to available seats across examination rooms. This is typically done using printed hall layouts, whiteboards, or Excel spreadsheets.

Limitations:

- **High Error Rate:** Human oversight often leads to repeated seat numbers or overlooked students.
- **Time-Consuming:** Allocating seats for thousands of students may take several days of preparation.
- **No Rule Enforcement:** Manual methods make it difficult to enforce constraints like avoiding same-subject students sitting adjacent to each other.
- **Lack of Visualization:** There's no way to preview or simulate hall occupancy without physically checking the space.

2. Spreadsheet-Based Tools:

Overview: To reduce paperwork, some institutions have adopted spreadsheet tools (e.g., Excel templates) to organize student data and draft seating plans.

Limitations:

- **No Automation:** Staff still manually insert seat numbers and check for conflicts.
- **Poor Conflict Detection:** Spreadsheets do not notify users of duplicate entries or capacity violations.
- **Non-Dynamic:** Any late changes (student addition, hall reassignment) require full rework.
- **No Integration:** Cannot connect to institutional databases, student portals, or attendance systems.

3. Commercial Exam Scheduling Tools:

Overview:
There are enterprise-grade systems available for timetable generation and exam scheduling; however, few specialize in seat arrangements.

Limitations:

- **Cost-Prohibitive:** Many are subscription-based and expensive for smaller institutions.
- **Overkill for Simpler Use Cases:** Institutions often require only the seating function without the overhead of full ERP integration.
- **Rigid Templates:** Lack flexibility to adapt to unique physical layouts or departmental policies.

The gaps in current systems emphasize the need for a dedicated, user-friendly, and intelligent seating arrangement platform that simplifies exam logistics without requiring advanced technical knowledge from staff.

3.2 PROPOSED SYSTEM

The proposed **Exam Seat Arrangement System** is a web-based, rule-driven application designed to fully automate the seating process based on student roll numbers, subject codes, hall capacities, and institutional constraints. It reduces human effort, minimizes errors, and supports scalability for large academic events.

Key Features:

1. **Smart Allocation Engine:**
    - Automatically assigns seats based on hall capacity and seating rules.
    - Ensures no duplication and follows spacing protocols (e.g., one-seat gap).
2. **Cross-Subject Distribution:**
    - Mixes students from different courses/subjects to minimize chances of malpractice.
3. **Conflict Resolution:**
    - Detects and alerts for seat overlaps, over-assignments, or scheduling conflicts.
4. **Multi-Hall Management:**
    - Handles multiple halls, blocks, and buildings simultaneously.
5. **Custom Room Mapping:**
    - Allows defining room layouts and applying custom rules (e.g., disable back rows or reserve front seats for special needs).
6. **Admin Dashboard:**
    - View student-wise and hall-wise arrangements in real time.
    - Export to Excel/PDF for printing hall charts, student slips, and invigilator lists.
7. **User Access Control:**
    - Role-based authentication for admins, department coordinators, and invigilators.

Workflow:

1. **Login & Authentication**
   Secure access for authorized personnel using role-based login.
2. **Data Import**
   Upload CSV or Excel files containing student info, subject codes, and hall availability.
3. **Seat Allocation**
   The system processes constraints and dynamically allocates seats to students.
4. **Visualization & Reports**
   View interactive seat maps and generate downloadable documents (e.g., student seating slips).
5. **Audit & Edit**
   Admins can review, adjust, or regenerate the seating arrangement as needed.

3.3 FEASIBILITY STUDY

1. Technical Feasibility:

The proposed system is based on widely available technologies such as:

- **Frontend:** HTML5, CSS3, JavaScript
- **Backend:** PHP
- **Database:** MySQL
- **Algorithms:** Rule-based logic, basic optimization (greedy allocation), and conflict detection loops

These technologies are mature, open-source, and well-supported, ensuring that the system can be developed, maintained, and scaled efficiently. Integration with spreadsheet formats and exportable reports ensures compatibility with existing workflows.

2. Economic Feasibility:

- **Low Development Cost:** Open-source tools reduce licensing expenses.
- **Operational Savings:** Saves significant time and resources for exam departments.
- **High ROI:** Institutions benefit from error reduction and improved transparency.
- **Scalable Licensing:** Can be deployed at institution-level, department-level, or centrally by examination boards.

3. Operational Feasibility:

- **Ease of Use:** Designed for non-technical staff; minimal training required.
- **Adaptability:** Can be customized for various exam types (end-semester, backlog, internal assessments).
- **Low Infrastructure Needs:** Works on standard desktops or web browsers without advanced hardware.

4. Legal & Ethical Feasibility:

- **Data Security:** All student data is stored securely with encryption and limited access.
- **Compliance:** Can be configured to meet local data privacy laws (e.g., GDPR, UGC norms).
- **Audit Trails:** Logs every allocation change to maintain accountability and transparency.

CHAPTER IV

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

To ensure smooth operation of the **Exam Seat Arrangement System**, both development and deployment environments must meet minimum hardware standards. The system is designed to be lightweight, requiring only standard hardware commonly available in educational institutions. However, to optimize performance—especially for large-scale institutions managing thousands of students—recommended configurations are also provided.

1. Processor (CPU):

- **Minimum:** Dual-core processor (e.g., Intel Core i3, 2.0 GHz)
- **Recommended:** Quad-core processor or better (e.g., Intel Core i5/i7, AMD Ryzen 5/7)

2. Memory (RAM):

- **Minimum:** 4 GB RAM
- **Recommended:** 8 GB or higher

3. Storage:

- **Minimum:** 100 GB HDD or SSD
- **Recommended:** 256 GB SSD or higher

4. Display:

- **Minimum Resolution:** 1280x720 (HD)
- **Recommended Resolution:** 1920x1080 (Full HD)

5. Input/Output Devices:

- Standard Keyboard and Mouse
- Laser Printer (for printing seat slips, hall charts, and invigilator reports)
- Projector (optional, for hall-wide display of seating charts)

6. Network Connectivity:

- **Minimum:** Broadband connection (10 Mbps)
- **Recommended:** Fiber-optic or high-speed LAN (>25 Mbps)

4.2 SOFTWARE REQUIREMENTS

The system is built using open-source and widely supported technologies to ensure ease of development, cross-platform compatibility, and maintainability.

1. Operating System:

- **Client Side:** Windows 10+, macOS 10.13+, or Linux (Ubuntu 18.04+)
- **Server Side:** Windows (for local deployment using XAMPP) or Linux-based OS for live deployment (LAMP stack)

2. Web Browsers (Client-Side):

- Google Chrome *(Recommended)*
- Microsoft Edge
- Mozilla Firefox *(Compatible)*.

3. Development Tools:

- **IDE/Text Editor:** Visual Studio Code *(preferred)*, Sublime Text, or Notepad++
- **Version Control:** Git (for tracking code changes), GitHub or GitLab (for team collaboration)

4. Frontend Technologies:

- **Languages:** HTML5, CSS3, JavaScript
- **Libraries/Frameworks:** Bootstrap or Tailwind CSS for responsive UI design

5. Backend Technologies:

- **Language:** PHP (server-side scripting)
- **Environment:** Apache server via XAMPP (development) or LAMP stack (production)

6. Database System:

- **Recommended:** MySQL (used for production deployment)

7. Reporting & Export:

- **PDF Export:** TCPDF / FPDF (PHP libraries for generating downloadable PDFs of seat slips and hall charts)
- **Excel Export:** PHPSpreadsheet (for exporting data into structured Excel reports)

8. Deployment Tools:

- **Local Development:** XAMPP (Apache, PHP, MySQL bundle)

9. Additional Tools (Optional Enhancements):

- **Visualization:** Chart.js (to visualize hall occupancy, department-wise seat distribution)
- **Testing & Debugging:** Browser DevTools, PHP error logs
- **Backup Management:** PHPMyAdmin for exporting/importing MySQL database backups
- **Security:** SSL configuration for live deployments; basic input sanitization to prevent SQL injection and XSS

CHAPTER V
SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The Exam Seat Arrangement System is designed as a web-based client-server application using a layered architecture. It follows a modular and maintainable design pattern, ensuring clean separation between data handling, business logic, and user interface. The system leverages PHP as the server-side scripting language, MySQL for data persistence, and front-end technologies like HTML, CSS, and JavaScript for dynamic user interaction.

1. System Layers:

1. **Presentation Layer (Front-End):**
    - Built using HTML5, CSS3, Bootstrap, and JavaScript
    - Responsible for displaying forms, dashboards, and reports to users
    - Interfaces with the backend using AJAX and PHP form submissions
2. **Business Logic Layer (Back-End):**
    - Developed using PHP
    - Contains the seat allocation engine, conflict detection logic, and report generators
    - Handles all server-side processing, input validation, and error management
3. **Data Access Layer:**
    - Communicates with the MySQL database using SQL queries through PHP
    - Handles data CRUD operations for student records, room details, allocations, and session logs
4. **Database Layer:**
    - Stores data in structured tables such as:
        - students

- exam_schedule
- hall_config
- seat_allocations
- users (for admin login)

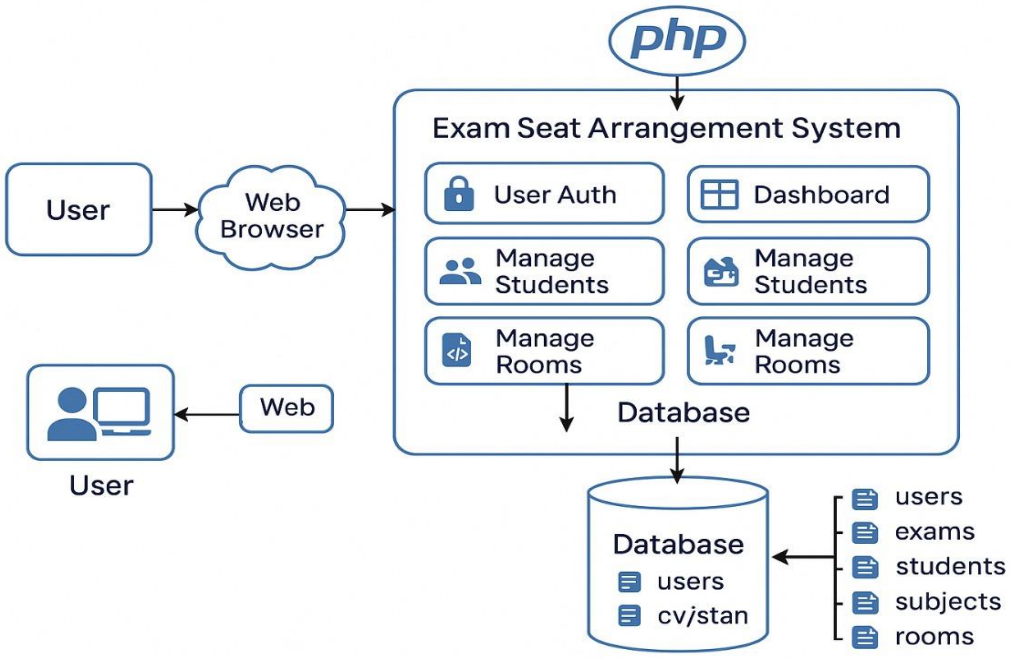

Figure 1: System Architecture

5.2 MODULE DESCRIPTION

MODULE 1: User Authentication Module

- Enables secure login for exam administrators and authorized users.
- PHP sessions are used to manage login state.
- Credentials are validated against a users table in MySQL.
- Role-based redirection allows restricted access to sensitive features like allocation and report downloads.

MODULE 2: Student & Hall Data Upload Module

- Allows admins to upload Excel or CSV files containing student roll numbers, course codes, exam dates, and room details.
- Front-end form validates file type and displays upload status.
- PHP parses the uploaded file and stores the entries in the students and hall_config tables.
- Data sanitation prevents SQL injection and malformed inputs.

MODULE 3: Seating Allocation Engine (PHP)

- Core logic written in PHP loops through student records and dynamically assigns available seats in the selected halls.
- Implements constraints such as:
    - No same-subject students adjacent
    - Hall capacity limits
    - Row/column-wise seating matrix
- Allocated seat numbers are saved into the seat_allocations table.

MODULE 4: Conflict Detection Module

- Runs post-allocation checks to detect:
    - Duplicate seat numbers
    - Overfilled halls
    - Students missing seat assignments
- Highlights issues in the UI using JavaScript alerts and red-colored rows in preview tables.
- Conflict summary displayed in the dashboard for admin review.

MODULE 5: Manual Override Module

- Allows admin to manually edit seat numbers or swap students.
- Updates made through a user-friendly form interface using JavaScript + PHP.
- All overrides logged in the logs table with timestamps and user IDs.

MODULE 6: Seat Map Visualizer

- Dynamically generates a seat map using HTML tables and JavaScript.
- Colors or labels each seat based on:
    - Subject
    - Student roll number
    - Hall number
- Responsive layout allows viewing on desktop and tablets.

MODULE 7: Report Generator Module

- Generates printable documents:
    - **Student Seat Slips:** Roll no, subject, seat number, room
    - **Hall-Wise Charts:** Room-wise seating layout for display
    - **Invigilator Report:** List of students per room with times and codes
- Uses PHP libraries like **FPDF** or **PHPSpreadsheet** for PDF/Excel export.

MODULE 8: Session Logging and Audit Trail Module

- Logs actions like uploads, allocations, edits, and downloads.
- Uses PHP's date() and $_SESSION variables to capture who did what and when.
- Stores in logs table for accountability and rollback reference.

MODULE 9: Dashboard & Notification Module

- Displays summary statistics:
    - Total students allocated
    - Available/unfilled seats per hall
    - Conflicts detected
- Uses JavaScript and Bootstrap cards for visual clarity.
- Sends alerts (in-page popups or email notifications) for major actions like data overwrite or allocation re-run.

CHAPTER VI

RESULT AND DISCUSSION

The deployment of the Exam Seat Arrangement System produced positive results in terms of efficiency, accuracy, and user satisfaction. The system demonstrated high effectiveness in automating the seating arrangement process, significantly reducing the time needed compared to traditional manual methods.

During testing, the system showed excellent accuracy in placing students according to predefined rules and room constraints, including special seating requests and social distancing measures. It completed seating arrangements in seconds, while manual methods would take hours, even for large datasets with hundreds of students.

The user interface was highly praised for being intuitive and user-friendly, with administrators finding it easy to input data and make adjustments. The system also generated detailed reports, allowing administrators to review seating plans and optimize space usage.

However, while the system performed well in most scenarios, it faced challenges in complex edge cases, such as last-minute changes or highly specific room configurations. These situations required some manual intervention, but they were relatively rare.

Overall, the system proved to be an efficient, accurate, and user-friendly tool for exam seat management, demonstrating significant potential for use in educational institutions to streamline the seating arrangement process.

CHAPTER VII

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The Exam Seat Arrangement System represents a significant advancement in automating the seating arrangement process for examination halls. By combining efficient algorithms, accurate placement methods, and user-friendly design, the system offers a streamlined, automated solution that reduces the time and effort required to generate seating plans. The system's performance during testing demonstrates its ability to handle large datasets and complex seating requirements with ease, outperforming traditional manual methods in both speed and accuracy.

The system is not intended to replace human decision-making but to act as an efficient tool that supports administrators in managing seating arrangements. It ensures fairness, maintains privacy, and accommodates special needs while allowing for quick adjustments when required. The positive feedback from administrators highlights the system's ease of use and effectiveness in reducing the workload associated with exam seating.

While the system has proven effective in typical scenarios, there are opportunities for further development to address more complex use cases and enhance functionality. With continued refinement, the system has the potential to become an essential tool for educational institutions worldwide, simplifying the exam preparation process and improving the overall exam experience for both students and staff.

7.2 FUTURE ENHANCEMENT

1. **Advanced Seat Optimization**: Incorporate more advanced algorithms to optimize seat placement further, considering factors like student performance levels, minimizing disruptions, and optimizing space utilization.
2. **Dynamic Scheduling Integration**: Implement integration with scheduling systems to dynamically adjust seating arrangements based on changes in exam timetables or room availability.
3. **Real-Time Modifications**: Enable real-time updates and modifications to seating arrangements, allowing for last-minute changes such as student cancellations or adjustments due to emergencies.
4. **Multi-User Access**: Provide multi-user access for administrators, allowing team members to collaborate on seating arrangements simultaneously, improving efficiency in large-scale exam setups.
5. **Mobile Support**: Develop a mobile application version of the system for administrators to access and manage seating arrangements on the go, increasing flexibility and accessibility.

APPENDIX

A1.1 SAMPLE CODE

HOME PAGE

```
<?php
/*
 * Unset the session vaariable since it is the first page and the session starts here
 */
session_start();
session_unset();
if($_SERVER['REQUEST_METHOD']=='POST')
{
   $_SESSION['Details']=$_POST;
   $_SESSION['Details']['noClass']='101';
   $str=<<< _alert
<script>
window.location.assign("TimeTableSe.php");
</script>
_alert;
echo $str;
}
else
{
?>
<!doctype html>
<html>
<head>
   <link rel="stylesheet" href="bootstrap/bootstrap.css"/>
   <link rel="stylesheet" href="custom.css"/>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script
>

    <title>Home Page</title>
            <?php include('head.php'); ?>
</head>
    <body>
    <nav class="navbar navbar-expand-sm navbar-light">
        <div class="container-fluid">
            <!--header-->
            <div class="navbar-header">
                <!-- <img src="dist/img/diulogoside.png" style="max-width:150px"/> -->
                <div class="navbar-brand">
                    <h1>EXAM SEATING ARRANGEMENT SYSTEM</h1>
                </div>
            </div>
        </div>
    </nav>
    <section>

    <div class="container">
```

```
   <p style="margin:auto; font-optical-sizing:auto; position:relative; text-align: center; padding:20px; font-size:xx-large;">Department of Artificial Intelligence and Data Science</p>
<!--        <div class="row">-->
<!--        <header class="col-md-6 col-md-push-3">-->
<!--          <h1>SEATING ARRANGEMENT SYSTEM</h1>-->
<!--        </header>-->
<!--        </div>-->
       <div class="form-group col-md-4 col-md-push-4">
       <form  method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']);?>" id="form1">
          <div>
          <label>Year</label>
             <select name="SemYear" id="SemYear" title="SemYear" class="form-control" style="font-size:12px">
                <!-- <option value= "2020">UG 1st Year</option>
                <option value= "2021">UG 2nd Year</option> -->
                <option value= "2022">UG 3rd Year</option>
                <!-- <option value= "2023">UG 4th Year</option>
                <option value= "2024">PG 1st Year</option>
                <option value= "2025">PG 2nd Year</option> -->
                <!-- <option value= "2026">2026</option>
                <option value= "2027">2027</option>
                <option value= "2028">2028</option>
                <option value= "2029">2029</option>
                <option value= "2030">2030</option>
                <option value= "2030">2031</option> -->
             </select><br/>
             <!-- <label>Program</label>  
             <select name="Exam" title="exam" class="form-control" style="font-size:12px">
```

```
                <option value="Day_Program">Day Program</option>
                <option value= "Evening_Program">Evening Program</option>
                <option value= "Masters_Program">Masters Program</option>
            </select><br/> -->
            <label>Semester</label>
            <select name="Semester" id="sem" title="Semester" class="form-
control" style="font-size:12px">
                <!-- <option value="spring">Odd Semester</option> -->
                <option value= "summer">Even Semester</option>
                <!-- <option value= "fall">Fall Semester</option> -->
            </select><br/>

        </div>
        <div class="row" style="display: none;">
            <div id="slot1">
                <header class="col-md-12"><label>First Slot</label></header>
                <div class="col-md-6">
                <label for="startTime1">Start Time:</label><input type="time"
name="startTime1" title="startTime1" class="form-control" style="font-
size:12px"/><br/>
                </div>
                <div class="col-md-6">
                <label for="endTime1">End Time:</label><input type="time"
name="endTime1" title="endTime1" class="form-control" style="font-
size:12px"/><br/>
                </div>
            </div>
        </div>
        <div class="row" style="display: none;">
            <div id="slot2">
                <header class="col-md-12"><label>Second Slot</label></header>
```

```
                <div class="col-md-6">
                    <label for="startTime2">Start Time:</label><input type="time"
name="startTime2" title="startTime2" class="form-control" style="font-
size:12px"/><br/>
                </div>
                <div class="col-md-6">
                    <label for="endTime2">End Time:</label><input type="time"
name="endTime2" title="endTime2" class="form-control" style="font-
size:12px"/><br/>
                </div>
            </div>
        </div>
        <label style="display: none;">
            <label for="noClass">Available Class Room</label><br>
            <input type="checkbox" name="classroomCode" value="DT-101"
checked>
            <label for="DT-101"> DT-101</label><br>
            <input type="checkbox" name="classroomCode" value="DT-102"
checked>
            <label for="DT-102"> DT-102</label><br>
            <input type="checkbox" name="classroomCode" value="DT-103">
            <label for="DT-103"> DT-103</label><br>
            <input type="checkbox" name="classroomCode" value="DT-104">
            <label for="DT-104"> DT-104</label><br>
            <input type="checkbox" name="classroomCode" value="DT-105">
            <label for="DT-105"> DT-105</label><br>
            <input type="checkbox" name="classroomCode" value="DT-106">
            <label for="DT-106"> DT-106</label><br>
        </label>
        <input type="submit" value="Start" class="form-control btn btn-dark"
style="font-size:12px">
```

```
        </form>
        </div>
    </div>
    </section>
    </body>
</html>
<?php }
?>
```

Homepage.php

SEATING ARRANGEMENT

```
<?php
session_start();
if(!isset($_SESSION['Details']))
{
    header('Location: homepage.php');
}
else
{
    ?>

    <!doctype html>
    <html>
    <head>
        <title>
            Print Routine
        </title>
                <?php include('head.php'); ?>
    </head>
<body>
    <nav class="navbar navbar-expand-sm navbar-light">
            <div class="navbar-header">
```

```
            <img src="dist/img/diulogoside.png" style="max-width:150px"/>
            <a href="homepage.php" class="navbar-brand" style="font-
size:25px">EXAM SEATING ARRANGEMENT SYSTEM</a>
        </div>
    </nav>
    <section>
        <div class="container">
            <div class="col-md-4 col-md-push-4">
                <div class="data">
                    <h1>Time Table is Ready For Print </h1>
                </div>
                <button name="Print" id="Print" onclick="printTimeTable()" class="btn
btn-dark btn-block">Print</button><br/>
                <button name="SeatingArrangement" id="seatingArrangement"
onclick="move()" class="btn btn-dark btn-block" disabled>Seating
Arrangement</button>
            </div>
        </div>
    </section>
<!-- Button For Seating Arrangement-->
    <script>
        function move()
        {
            alert("Moving To Print Seating Arrangement Chart");
            window.location.assign("seatingArrangement.php");
        }
        function printTimeTable()
        {

document.getElementById('seatingArrangement').removeAttribute("disabled");
            window.location.assign("TimeTablePdf.php");
```

```
        }
    </script>
    </body>
    </html>
    <?php
}
?>
```

SEATING ARRANGEMENT.SQL

```
-- phpMyAdmin SQL Dump
-- version 4.8.3
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Oct 06, 2020 at 01:10 AM
-- Server version: 10.1.35-MariaDB
-- PHP Version: 7.2.9

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: seatingarrangement
--

-- --------------------------------------------------------

--
-- Table structure for table classroom
--

CREATE TABLE classroom (
  Classroom_ID varchar(10) NOT NULL,
  Classroom_Name varchar(20) NOT NULL,
  Available_Seat int(10) NOT NULL,
  Assigned_At date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table classroom
--

INSERT INTO classroom (Classroom_ID, Classroom_Name, Available_Seat,
Assigned_At) VALUES
('DT101', 'DT 101', 40, '0000-00-00'),
('DT102', 'DT 102', 50, '0000-00-00'),
('DT103', 'DT 103', 60, '0000-00-00');

-- --------------------------------------------------------

--
-- Table structure for table fourthyear
```

```
--

CREATE TABLE fourthyear (
  Roll_No int(11) DEFAULT NULL,
  first_name varchar(50) DEFAULT NULL,
  last_name varchar(50) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table fourthyear
--

INSERT INTO fourthyear (Roll_No, first_name, last_name) VALUES
(1, 'Benjamin', 'Henry'),
(2, 'Raymond', 'Willis'),
(3, 'Alice', 'Edwards'),
(4, 'Michael', 'Ruiz'),
(5, 'Julie', 'Morris'),
(6, 'Jonathan', 'Bowman'),
(7, 'Tammy', 'Hill'),
(8, 'Ralph', 'Day'),
(9, 'Ronald', 'Gordon'),
(10, 'Wayne', 'Hudson'),
(11, 'Billy', 'Fowler'),
(12, 'Jason', 'Watson'),
(13, 'Sharon', 'Ward'),
(14, 'Julie', 'Sanders'),
(15, 'Douglas', 'Scott'),
(16, 'Katherine', 'Chavez'),
(17, 'Juan', 'Parker'),
(18, 'Sara', 'Harris'),
```

(19, 'Joan', 'Cox'),
(20, 'Teresa', 'Parker'),
(21, 'Christine', 'Smith'),
(22, 'Dennis', 'Simmons'),
(23, 'Jesse', 'Reed'),
(24, 'Nicholas', 'Stanley'),
(25, 'Patrick', 'Shaw'),
(26, 'Jack', 'Hughes'),
(27, 'James', 'Ruiz'),
(28, 'Steven', 'Greene'),
(29, 'Heather', 'Welch'),
(30, 'Cynthia', 'Hill'),
(31, 'Lisa', 'Fowler'),
(32, 'Shawn', 'Armstrong'),
(33, 'Alan', 'Williams'),
(34, 'Keith', 'Roberts'),
(35, 'Phyllis', 'Boyd'),
(36, 'Teresa', 'Wilson'),
(37, 'Earl', 'Holmes'),
(38, 'Matthew', 'Matthews'),
(39, 'Charles', 'Mitchell'),
(40, 'Shirley', 'Ferguson'),
(41, 'Arthur', 'Gardner'),
(42, 'Shirley', 'Andrews'),
(43, 'Christine', 'Thomas'),
(44, 'Frank', 'Freeman'),
(45, 'Jean', 'Willis'),
(46, 'Carolyn', 'Wilson'),
(47, 'Justin', 'Garcia'),
(48, 'Elizabeth', 'Elliott'),
(49, 'Rebecca', 'Barnes'),

(50, 'Debra', 'Washington'),
(51, 'John', 'Scott'),
(52, 'Steve', 'Smith'),
(53, 'Jose', 'Oliver'),
(54, 'Fred', 'Tucker'),
(55, 'Mary', 'Kim'),
(56, 'William', 'Perez'),
(57, 'Nancy', 'Perez'),
(58, 'Juan', 'Ramirez'),
(59, 'Joshua', 'Dean'),
(60, 'Albert', 'Fuller'),
(61, 'Juan', 'Franklin'),
(62, 'Doris', 'Kennedy'),
(63, 'Joseph', 'Banks'),
(64, 'Tammy', 'Palmer'),
(65, 'William', 'Ford'),
(66, 'Carolyn', 'Russell'),
(67, 'Thomas', 'Lane'),
(68, 'Lawrence', 'Rogers'),
(69, 'Teresa', 'Graham'),
(70, 'Patricia', 'Russell'),
(71, 'David', 'White'),
(72, 'Ronald', 'Price'),
(73, 'Jane', 'Johnston'),
(74, 'Mary', 'Williamson'),
(75, 'Earl', 'Alvarez'),
(76, 'John', 'Murphy'),
(77, 'Billy', 'Gilbert'),
(78, 'Brandon', 'Watkins'),
(79, 'Sean', 'Moreno'),
(80, 'Cheryl', 'Perez');

```
-- --------------------------------------------------------

--
-- Table structure for table login
--

CREATE TABLE login (
  ID int(10) NOT NULL,
  uname varchar(50) NOT NULL,
  pass varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table login
--

INSERT INTO login (ID, uname, pass) VALUES
(1, 'admin', 'admin'),
(2, 'uday15-9779@diu.edu.bd', '172-15-9779'),
(3, 'neelima15-10150@diu.edu.bd', '172-15-10150'),
(4, 'syeda15-10000@diu.edu.bd', '172-15-10000');

-- --------------------------------------------------------

--
-- Table structure for table secondyear
--

CREATE TABLE secondyear (
  Roll_No int(11) DEFAULT NULL,
```

```
  first_name varchar(50) DEFAULT NULL,
  last_name varchar(50) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table secondyear
--

INSERT INTO secondyear (Roll_No, first_name, last_name) VALUES
(1, 'Ronald', 'Mendoza'),
(2, 'Howard', 'Crawford'),
(3, 'Lois', 'Peters'),
(4, 'Rachel', 'Sims'),
(5, 'Joyce', 'Hanson'),
(6, 'Gerald', 'Morales'),
(7, 'Thomas', 'Lawrence'),
(8, 'Kimberly', 'Riley'),
(9, 'Roger', 'Woods'),
(10, 'Adam', 'Williamson'),
(11, 'Barbara', 'Young'),
(12, 'Elizabeth', 'Knight'),
(13, 'Frank', 'Baker'),
(14, 'Alan', 'Weaver'),
(15, 'Cynthia', 'Hernandez'),
(16, 'Patricia', 'Bryant'),
(17, 'George', 'Gordon'),
(18, 'Louis', 'Ford'),
(19, 'Anna', 'George'),
(20, 'Christina', 'Torres'),
(21, 'Marie', 'Reynolds'),
(22, 'Russell', 'Fernandez'),
```

(23, 'Robin', 'Edwards'),
(24, 'Thomas', 'Martinez'),
(25, 'Beverly', 'Rose'),
(26, 'Dennis', 'Murray'),
(27, 'Mark', 'Banks'),
(28, 'Christopher', 'Sullivan'),
(29, 'Samuel', 'Mcdonald'),
(30, 'Laura', 'Flores'),
(31, 'Mildred', 'Ruiz'),
(32, 'Joan', 'Hayes'),
(33, 'Justin', 'Jackson'),
(34, 'Thomas', 'Dunn'),
(35, 'Elizabeth', 'Edwards'),
(36, 'Brenda', 'Martinez'),
(37, 'Teresa', 'Owens'),
(38, 'Lawrence', 'Patterson'),
(39, 'Jonathan', 'Cox'),
(40, 'Lori', 'Rodriguez'),
(41, 'Eugene', 'Ortiz'),
(42, 'Catherine', 'Clark'),
(43, 'Kathryn', 'Gomez'),
(44, 'Sandra', 'Robertson'),
(45, 'Michael', 'Owens'),
(46, 'Margaret', 'Fernandez'),
(47, 'Stephen', 'Gibson'),
(48, 'Dorothy', 'Sullivan'),
(49, 'Sean', 'Carpenter'),
(50, 'Steven', 'James'),
(51, 'Edward', 'Berry'),
(52, 'Scott', 'Cole'),
(53, 'Cynthia', 'Murray'),

(54, 'Jacqueline', 'Diaz'),
(55, 'Melissa', 'Lynch'),
(56, 'Cynthia', 'Peterson'),
(57, 'Debra', 'Robinson'),
(58, 'Lawrence', 'Mendoza'),
(59, 'Jean', 'Griffin'),
(60, 'Annie', 'Burns'),
(61, 'Billy', 'Marshall'),
(62, 'Roger', 'Holmes'),
(63, 'David', 'Moreno'),
(64, 'Johnny', 'Wright'),
(65, 'Stephen', 'Mendoza'),
(66, 'Jose', 'Alvarez'),
(67, 'Russell', 'Pierce'),
(68, 'Brian', 'Spencer'),
(69, 'William', 'Brooks'),
(70, 'Chris', 'Fields'),
(71, 'Adam', 'Davis'),
(72, 'Harold', 'Price'),
(73, 'Tammy', 'Peters'),
(74, 'Louise', 'Watson'),
(75, 'Denise', 'Murphy'),
(76, 'Edward', 'Kelley'),
(77, 'Donald', 'Wallace'),
(78, 'Raymond', 'Foster'),
(79, 'Steven', 'Fowler'),
(80, 'Catherine', 'Oliver');

-- --

--

```
-- Table structure for table subject
--

CREATE TABLE subject (
  SubjectCode varchar(20) NOT NULL,
  Name varchar(100) NOT NULL,
  ShortNames varchar(10) NOT NULL,
  program_ID int(20) NOT NULL,
  level_ID int(20) NOT NULL,
  term_ID int(20) NOT NULL,
  Lecturer_Id varchar(10) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table subject
--

INSERT INTO subject (SubjectCode, Name, ShortNames, program_ID, level_ID,
term_ID, Lecturer_Id) VALUES
('MAT121', 'Mathematics II: Linear algebra and Coordinate Geometry', 'MAT 121',
2, 1, 1, '1005'),
('CSE131', 'Discrete Mathematics', ' CSE 131', 2, 1, 1, '1002'),
('ENG113', 'English Language-1', 'ENG 113', 2, 1, 1, '1004'),
('PHY123', 'Physics- II: Electricity, Magnetism and Modern Physics', 'PHY 123', 2,
1, 1, '1001'),
('PHY123L', 'Physics-II Lab', 'PHY 123L', 2, 1, 1, '1010'),
('CSE213', 'Algorithms', 'CSE 213', 2, 1, 2, '1001'),
('CSE213L', 'Algorithm Lab', 'CSE 213L', 2, 1, 2, '1005'),
('ACC214', 'Accounting', 'ACC 214', 2, 1, 2, '1005'),
('ECO314', ' Economics', ' ECO 314', 2, 1, 2, '1009'),
```

('MAT134', 'Mathematics III : Ordinary and Partial Differential Equations', ' MAT 134', 2, 1, 2, '1008'),
('CSE221', 'Theory of Computing', ' CSE 221', 2, 1, 3, '1009'),
('CSE222', 'Object Oriented Programming', ' CSE 222', 2, 1, 3, '1005'),
('CSE222L', 'Object Oriented Programming Lab', 'CSE 222L', 2, 1, 3, '1001'),
('MAT211', 'Mathematics-IV : Engineering Mathematics', ' MAT 211', 2, 1, 3, '1001'),
('STA223', ' Statistics', ' STA 223', 2, 1, 3, '1004'),
('CSE231', 'Microprocessor and Assembly Language', ' CSE 231', 2, 2, 1, '1001'),
('CSE231L', ' Microprocessor and assembly Language Lab', ' CSE 231L', 2, 2, 1, '1007'),
('CSE224', 'Electronic Devices and Circuits', ' CSE 224', 2, 2, 1, '1010'),
('CSE224L', 'Electronic Devices and Circuits Lab', ' CSE 224L', 2, 2, 1, '1002'),
('CSE232', 'Instrumentation and Control', 'CSE 232', 2, 2, 1, '1010'),
('CSE233', 'Data Communication', 'CSE 233', 2, 2, 1, '1010'),
('CSE322', 'Computer Architecture and Organization', ' CSE 322', 2, 2, 2, '1008'),
('CSE313', 'Computer Networks', 'CSE 313', 2, 2, 2, '1010'),
('CSE313L', 'Computer Networks Lab', 'CSE 313L', 2, 2, 2, '1007'),
('CSE311', 'Database Management System', 'CSE 311', 2, 2, 2, '1001'),
('CSE311L', 'Database Management System Lab', 'CSE 311L', 2, 2, 2, '1010'),
('CSE312', 'Numerical Methods', ' CSE 312', 2, 2, 2, '1005'),
('CSE331', 'Compiler Design', 'CSE 331', 2, 2, 3, '1002'),
('CSE331L', 'Compiler Design Lab', 'CSE 331L', 2, 2, 3, '1007'),
('CSE413', 'Simulation and Modeling', 'CSE 413', 2, 2, 3, '1006'),
('CSE413L', 'Simulation and Modeling Lab', ' CSE 413L', 2, 2, 3, '1003'),
('CSE323', 'Operating System', 'CSE 323', 2, 2, 3, '1010'),
('CSE323L', 'Operating System Lab', 'CSE 323L', 2, 2, 3, '1003'),
('CSE321', 'System Analysis and Design', 'CSE 321', 2, 3, 1, '1003'),
('CSE321L', 'System Analysis and Design Lab', 'CSE 321L', 2, 3, 1, '1008'),
('CSE421', 'Computer Graphics', 'CSE 421', 2, 3, 1, '1004'),
('CSE421L', 'Computer Graphics Lab', 'CSE 421L', 2, 3, 1, '1004'),

('CSE431', 'E-Commerce & Web Application', 'CSE 431', 2, 3, 1, '1005'),
('MGT414', ' Industrial Management', ' MGT 414', 2, 3, 1, '1006'),
('CSE412', 'Artificial Intelligence', 'CSE 412', 2, 3, 2, '1007'),
('CSE412L', 'Artificial Intelligence Lab', 'CSE 412L', 2, 3, 2, '1007'),
('CSE411', 'Communication Engineering', 'CSE 411', 2, 3, 2, '1008'),
('CSE332', 'Software Engineering', 'CSE 332', 2, 3, 2, '1009'),
('CSE333', 'Peripherals & Interfacing', 'CSE 333', 2, 3, 3, '1009'),
('CSE432', 'Computer and Network Security', 'CSE 432', 2, 3, 3, '1006'),
('CSE112', 'Computer Fundamentals', 'CSE 112', 1, 1, 1, '1001'),
('MAT111', 'Mathematics-I: Differential and Integral Calculus', 'MAT 111', 1, 1, 1, '1010'),
('ENG113D', 'Basic Functional English and English Spoken', 'ENG 113', 1, 1, 1, '1003'),
('PHY113', 'Physics-I: Mechanics, Heat & Thermodynamics,Waves & Oscillation, Optics', 'PHY 113', 1, 1, 1, '1004'),
('MAT121D', 'Mathematics -II: Complex Variable, Linear Algebra and Coordinate Geometry', 'MAT 121', 1, 1, 2, '1005'),
('CSE122', 'Programming and Problem Solving', 'CSE 122', 1, 1, 2, '1006'),
('CSE123', 'Problem Solving Lab', 'CSE 123', 1, 1, 2, '1007'),
('PHY123D', 'Physics-II: Electricity, Magnetism and Modern Physics', 'PHY 123', 1, 1, 2, '1008'),
('PHY124', 'Physics-II Lab', 'PHY 124', 1, 1, 2, '1009'),
('ENG123', 'Writing and Comprehension', 'ENG 123', 1, 1, 2, '1010'),
('CSE131D', 'Discrete Mathematics', 'CSE 131', 1, 1, 3, '1010'),
('CSE132', 'Electrical Circuits', 'CSE 132', 1, 1, 3, '1001'),
('CSE133', 'Electrical Circuits Lab', 'CSE 133', 1, 1, 3, '1002'),
('CSE134', 'Data Structure', 'CSE 134', 1, 1, 3, '1003'),
('CSE135', 'Data Structure Lab', 'CSE 135', 1, 1, 3, '1002'),
('MAT131', 'Ordinary and Partial Differential Equations', 'MAT 131', 1, 1, 3, '1003'),
('MAT211D', 'Engineering Mathematics', 'MAT 211', 1, 2, 1, '1004'),

('CSE212', 'Digital Electronics', 'CSE 212', 1, 2, 1, '1005'),
('CSE213D', 'Digital Electronics Lab', 'CSE 213', 1, 2, 1, '1006'),
('CSE214', 'Object Oriented Programming', 'CSE 214', 1, 2, 1, '1007'),
('CSE215', 'Object Oriented Programming Lab', 'CSE 215', 1, 2, 1, '1008'),
('ED201', 'G Bangladesh Studies', 'ED 201', 1, 2, 1, '1009'),
('CSE221D', 'Algorithms', 'CSE 221', 1, 2, 2, '1010'),
('CSE222D', 'Algorithms Lab', 'CSE 222', 1, 2, 2, '1003'),
('STA133D', 'Statistics and Probability', 'STA 133', 1, 2, 2, '1004'),
('CSE224D', 'Electronic Devices and Circuits', 'CSE 224', 1, 2, 2, '1003'),
('CSE225', 'Electronic Devices and Circuits Lab', 'CSE 225', 1, 2, 2, '1004'),
('CSE231D', 'Microprocessor and Assembly Language', 'CSE 231', 1, 2, 3, '1005'),
('CSE232D', 'Microprocessor and Assembly Language Lab', 'CSE 232', 1, 2, 3,
'1006'),
('CSE233D', 'Data Communication', 'CSE 233', 1, 2, 3, '1005'),
('CSE234D', 'Numerical Methods', 'CSE 234', 1, 2, 3, '1006'),
('CSE235', 'Introduction to Bio-Informatics', 'CSE 235', 1, 2, 3, '1006'),
('CSE311D', 'Database Management System', 'CSE 311', 1, 3, 1, '1007'),
('CSE312D', 'Database Management System Lab', 'CSE 312', 1, 3, 1, '1008'),
('CSE313D', 'Computer Networks', 'CSE 313', 1, 3, 1, '1009'),
('ECO314D', 'Economics', 'ECO 314', 1, 3, 1, '1008'),
('CSE321D', 'System Analysis and Design', 'CSE 321', 1, 3, 2, '1008'),
('CSE322D', 'Computer Architecture and Organization', 'CSE 322', 1, 3, 2, '1009'),
('CSE323D', 'Operating Systems', 'CSE 323', 1, 3, 2, '1008'),
('CSE324', 'Operating Systems Lab', 'CSE 324', 1, 3, 2, '1009'),
('GED321', 'Art of Effective Living', 'GED 321', 1, 3, 2, '1010'),
('CSE314D', 'Computer Networks Lab', 'CSE 314', 1, 3, 1, '1002'),
('CSE331D', 'Complier Design', 'CSE 331', 1, 3, 3, '1002'),
('CSE332D', 'Complier Design Lab', 'CSE 332', 1, 3, 3, '1003'),
('CSE333D', 'Software Engineering', 'CSE 333', 1, 3, 3, '1004'),
('CSE334', 'Wireless Programming', 'CSE 334', 1, 3, 3, '1003'),
('ACT301', 'Financial and Managerial Accounting 2', 'ACT 301', 1, 3, 3, '1004'),

```
('CSE412D', 'Artificial Intelligence', 'CSE 412', 1, 4, 1, '1005'),
('CSE413D', 'Artificial Intelligence Lab', 'CSE 413', 1, 4, 1, '1006'),
('CSE414D', 'Simulation and Modelling', 'CSE 414', 1, 4, 1, '1005'),
('CSE415D', 'Simulation and Modelling Lab', 'CSE 415', 1, 4, 1, '1006'),
('CSE417', 'Web Engineering', 'CSE 417', 1, 4, 1, '1007'),
('CSE418', 'Web Engineering Lab', 'CSE418', 1, 4, 1, '1008'),
('CSE421D', 'Computer Graphics', 'CSE 421', 1, 4, 2, '1008'),
('CSE422D', 'Computer Graphics Lab', 'CSE 422', 1, 4, 2, '1009'),
('CSE423', 'Embedded Systems', 'CSE 423', 1, 4, 2, '1009'),
('CSE498', 'Social and Professional Issues in Computing', 'CSE 498', 1, 4, 3,
'1010');

-- --------------------------------------------------------

--
-- Table structure for table teachers
--

CREATE TABLE teachers (
  EmpId varchar(10) NOT NULL,
  Name varchar(50) NOT NULL,
  Designation varchar(20) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table teachers
--

INSERT INTO teachers (EmpId, Name, Designation) VALUES
('1001', 'Prof. Dr. Syed Akhter Hossain', 'Head of the Dept'),
('1002', 'Dr. Sheak Rashed Haider Noori', 'Associate Head'),
```

```
('1003', 'Dr. Md. Mustafizur Rahman', 'Associate Professor'),
('1004', 'Dr. S. M. Aminul Haque', 'Associate Professor'),
('1005', 'Professor Dr. Md. Ismail Jabiullah', 'Professor'),
('1006', 'Dr. S.R.Subramanya', 'Visiting Professor'),
('1007', 'Dr. Neil Perez Balba', 'Visiting Professor'),
('1008', 'Dr. Bibhuti Roy', 'Visiting Professor'),
('1009', 'Mr. Anisur Rahman', 'Assistant Professor'),
('1010', 'Mr. Gazi Zahirul Islam', 'Assistant Professor');

-- --------------------------------------------------------

--
-- Table structure for table thirdyear
--

CREATE TABLE thirdyear (
  Roll_No int(11) DEFAULT NULL,
  first_name varchar(50) DEFAULT NULL,
  last_name varchar(50) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table thirdyear
--

INSERT INTO thirdyear (Roll_No, first_name, last_name) VALUES
(1, 'Michelle', 'James'),
(2, 'Kimberly', 'Richardson'),
(3, 'Raymond', 'Coleman'),
(4, 'Arthur', 'Lynch'),
(5, 'Juan', 'Hicks'),
```

(6, 'Jennifer', 'Bishop'),
(7, 'Kelly', 'Burton'),
(8, 'Richard', 'Davis'),
(9, 'Frank', 'Weaver'),
(10, 'Martha', 'Schmidt'),
(11, 'Henry', 'Arnold'),
(12, 'Jacqueline', 'Morales'),
(13, 'Alice', 'Kim'),
(14, 'Cynthia', 'Kennedy'),
(15, 'Ryan', 'Miller'),
(16, 'Alan', 'Smith'),
(17, 'Jeremy', 'Jenkins'),
(18, 'Adam', 'Stephens'),
(19, 'Phyllis', 'Castillo'),
(20, 'Jeffrey', 'Williams'),
(21, 'Jimmy', 'Alvarez'),
(22, 'Christopher', 'Shaw'),
(23, 'Ann', 'Hanson'),
(24, 'Anna', 'Burns'),
(25, 'George', 'Gutierrez'),
(26, 'Bonnie', 'Nichols'),
(27, 'Annie', 'Castillo'),
(28, 'Chris', 'Austin'),
(29, 'Anna', 'Moore'),
(30, 'James', 'Cruz'),
(31, 'Fred', 'Smith'),
(32, 'Linda', 'Rice'),
(33, 'Thomas', 'Barnes'),
(34, 'Albert', 'Nichols'),
(35, 'Gregory', 'Rogers'),
(36, 'Lisa', 'Hughes'),

(37, 'Christine', 'Simmons'),
(38, 'Scott', 'Barnes'),
(39, 'James', 'Lynch'),
(40, 'Cheryl', 'Webb'),
(41, 'Annie', 'Alexander'),
(42, 'Matthew', 'Sanders'),
(43, 'Scott', 'Moreno'),
(44, 'Paula', 'Rose'),
(45, 'Betty', 'Lawson'),
(46, 'Walter', 'Sanders'),
(47, 'Jack', 'Porter'),
(48, 'Jean', 'Hernandez'),
(49, 'Anne', 'Sims'),
(50, 'Louis', 'Hart'),
(51, 'Joseph', 'Hernandez'),
(52, 'Larry', 'Murphy'),
(53, 'Stephen', 'Gonzales'),
(54, 'Tammy', 'Rogers'),
(55, 'Lori', 'Dunn'),
(56, 'Andrea', 'Willis'),
(57, 'Cheryl', 'Harrison'),
(58, 'Stephen', 'Miller'),
(59, 'Sharon', 'Ferguson'),
(60, 'Joseph', 'Roberts'),
(61, 'Ruby', 'Jones'),
(62, 'Brandon', 'Payne'),
(63, 'Joshua', 'Sims'),
(64, 'Lois', 'Cunningham'),
(65, 'Margaret', 'Lane'),
(66, 'Eugene', 'Frazier'),
(67, 'Marilyn', 'Torres'),

```
(68, 'John', 'Kennedy'),
(69, 'Diana', 'Turner'),
(70, 'Joyce', 'Hanson'),
(71, 'Jose', 'Anderson'),
(72, 'Lillian', 'Reyes'),
(73, 'Carolyn', 'Jacobs'),
(74, 'Ryan', 'Medina'),
(75, 'Marie', 'Murray'),
(76, 'Matthew', 'Hughes'),
(77, 'Anthony', 'Edwards'),
(78, 'Aaron', 'Bishop'),
(79, 'Linda', 'Mendoza'),
(80, 'Craig', 'Alvarez');

--
-- Indexes for dumped tables
--

--
-- Indexes for table classroom
--
ALTER TABLE classroom
  ADD PRIMARY KEY (Classroom_ID);

--
-- Indexes for table login
--
ALTER TABLE login
  ADD PRIMARY KEY (ID);

--
```

```
-- Indexes for table subject
--
ALTER TABLE subject
  ADD PRIMARY KEY (SubjectCode),
  ADD KEY Lecturer_Id (Lecturer_Id);

--
-- Indexes for table teachers
--
ALTER TABLE teachers
  ADD PRIMARY KEY (EmpId);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table login
--
ALTER TABLE login
  MODIFY ID int(10) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=5;
COMMIT;

/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

A1.2 SCREENSHOTS

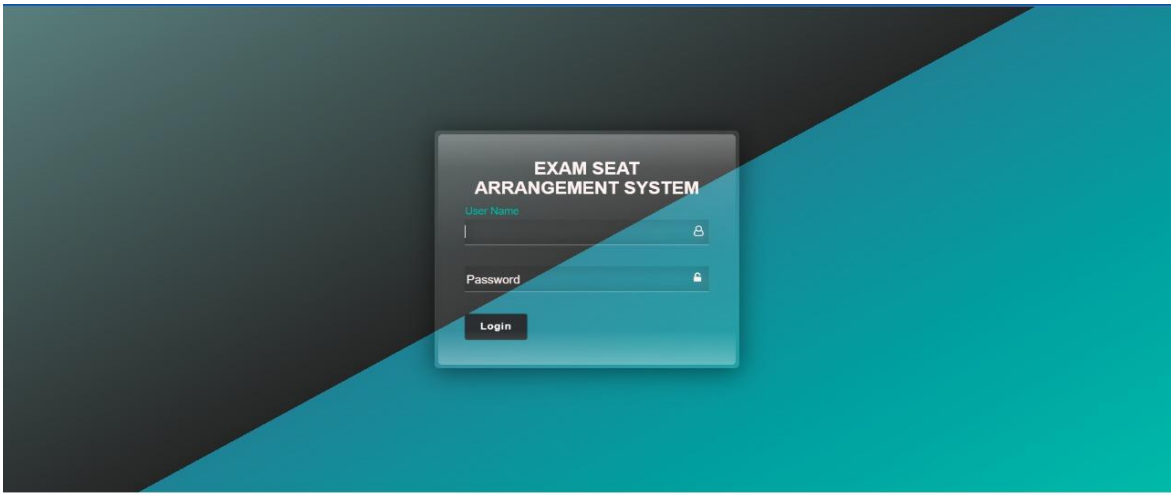

Figure 2: Login Page

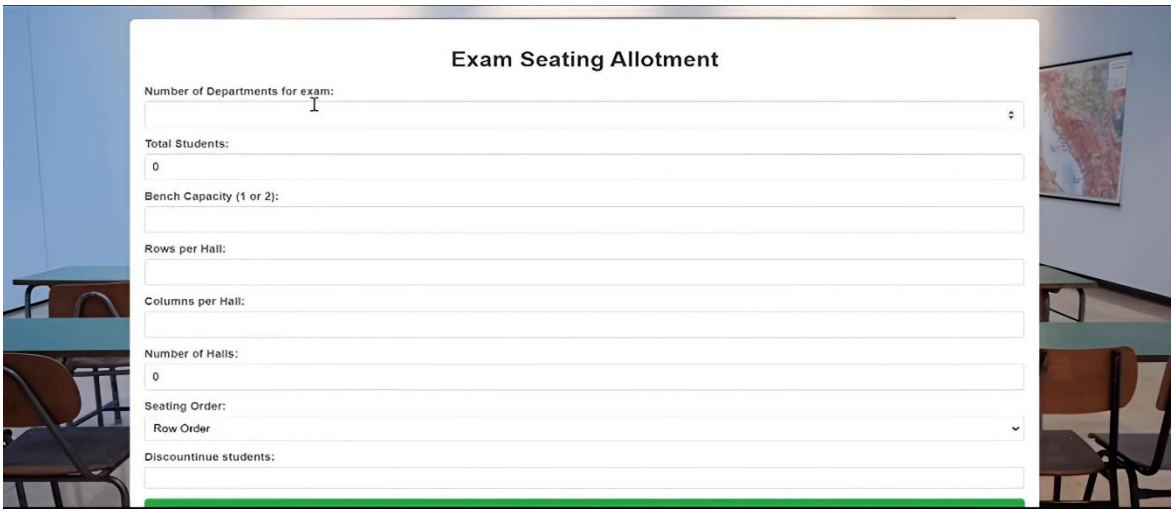

Figure 3: Initial Seat Allotment Fields

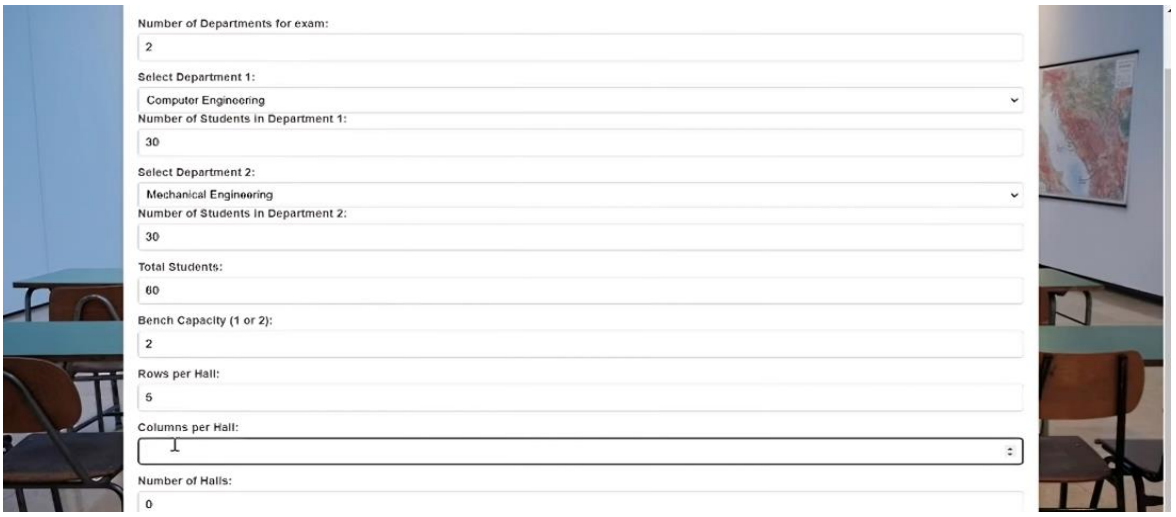

Figure 4: Sample Input for seat allotment fields

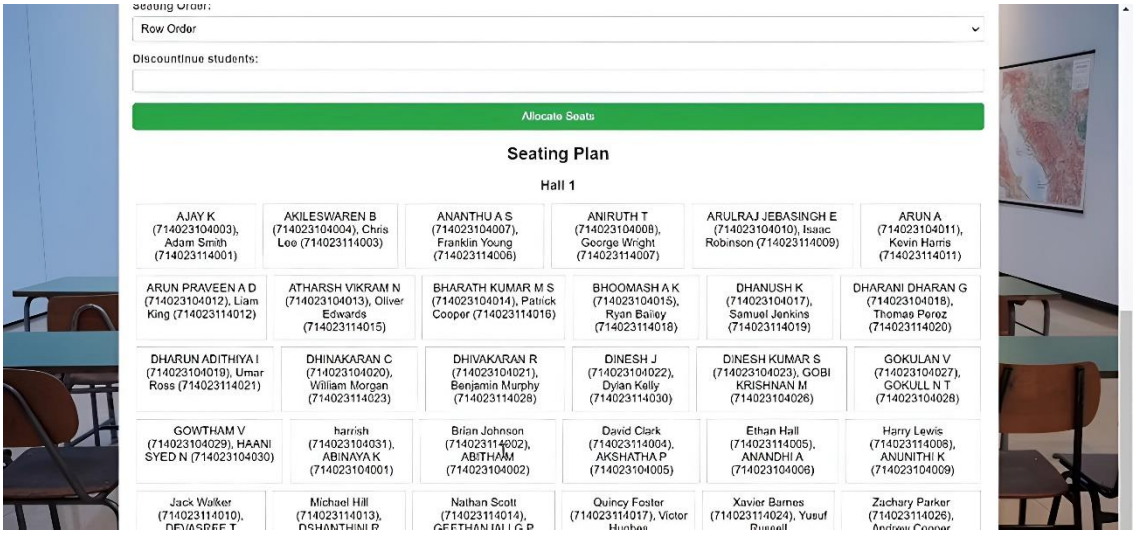

Figure 5: Report Generator page

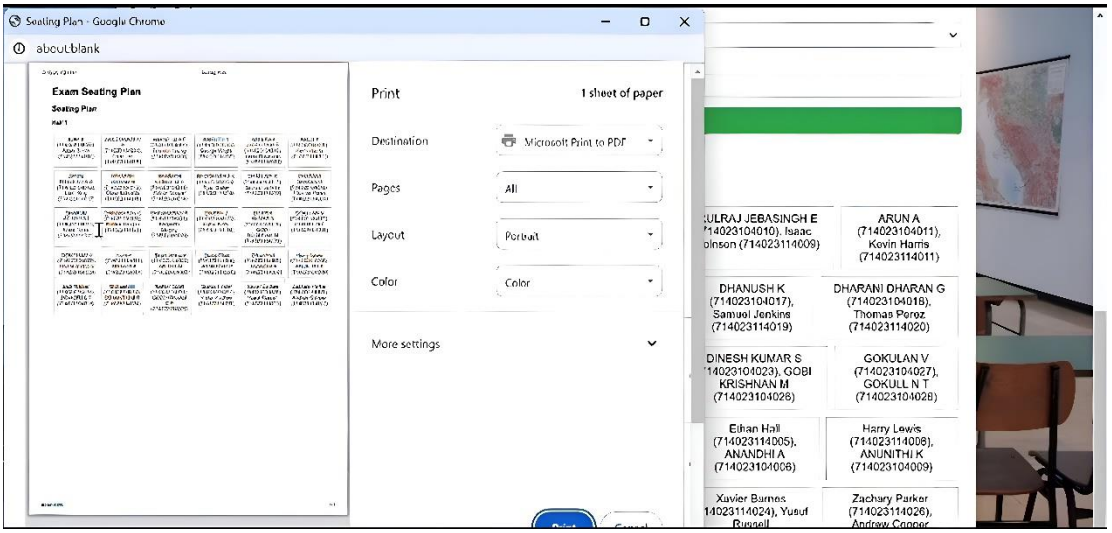

Figure 6: Generated report export page

REFERENCES

[1] IEEE Transactions on Education, "An Automated Seat Arrangement System for Examination Halls: Design and Implementation," Feb. 2023.

[2] IEEE Transactions on Systems, Man, and Cybernetics, "Optimizing Exam Seating with Algorithmic Approaches," Sept. 2022.

[3] Journal of Educational Technology Systems, "Development and Evaluation of a System for Automated Exam Seating," 2024.

[4] International Journal of Artificial Intelligence in Education, "AI-Based Solutions for Managing Large-Scale Exam Seating Arrangements," June 2021.

[5] IEEE Access, "Efficient Seating Plans in Examination Halls: The Role of Automation," 2023.

[6] International Journal of Human-Computer Interaction, "User Experience and Interface Design in Exam Seat Arrangement Systems," 2023.

[7] Educational Technology Research and Development, "Evaluating the Effectiveness of Automated Systems in Exam Scheduling and Seating," Jan. 2024.

[8] International Conference on Intelligent Systems, "Application of AI for Optimized Seating in Educational Exams," 2023.

[9] Journal of Computer Science Education, "Automated Room and Seating Arrangement Systems: A Case Study," Dec. 2022.

[10] arXiv, "Developing an AI-Based System for Optimizing Seating Arrangements in Educational Examinations," 2024.

[11] Springer, "Enhancing Exam Administration with AI: A Study on Seating Arrangement Automation," 2024.

[12] Journal of Information Systems and Technology, "AI-Driven Solutions for Exam Hall Seating and Management," 2024.

[13] International Journal of Computer Applications, "Automated Seat Allocation for Exam Halls: The Role of Data-Driven Systems," 2024.

[14] IEEE Transactions on Engineering Management, "Efficient Management of Exam Seating: A Case Study in Automation," 2023.

[15] Journal of Higher Education Research, "Implementing AI for Seating Optimization in Large-Scale Exams," 2024.

[16] Conference on AI and Education, "Exploring Automated Systems for Seating Arrangement in Educational Institutions," 2023.
[17] Journal of Educational Management, "Improving Exam Hall Organization Through AI-Based Seat Allocation," 2024.
[18] Education and Information Technologies, "AI Tools for Improving the Efficiency of Exam Seat Management," 2023.
[19] ICICDT Conference, "Towards a Smart Exam Seating System: Integration of AI for Optimized Room Allocation," 2023.
[20] Proceedings of the International Conference on Educational Technologies, "Designing and Deploying an Automated Seat Assignment System for Exams," 2024.