Malnad College of Engineering

Under the auspices of M.T.E.S

(An Autonomous Institution of VTU, Belgaum)

P.B No.21, Hassan-573 202, Karnataka

Mini project (20IS507)

Report on

EXAM ROOM AND SEAT ALLOTMENT SYSTEM

Submitted by:

Name	USN
PAVAN B	4MC21IS077
S M YATHIN	4MC21IS090
SUGNYAN K	4MC21IS106
SUHAS K P	4MC21IS107

Under the guidance of

Dr. Ananda Babu J

Assoc. Prof & Assoc. Dean AA

Department of Information Science and Engineering

Malnad College of Engineering
Hassan, Karnataka-573 202
2023-2024

Malnad College of Engineering

Under the auspices of M.T.E.S

(An Autonomous Institution of VTU, Belgaum)

P.B No.21, Hassan-573 202, Karnataka

CERTIFICATE

Certified that the mini project work carried out by 4MC21IS077, 4MC21IS090, 4MC21IS106, and 4MC21IS107 is a Bonafede work, submitted during the academic year 2023-24, in partial fulfilment for the award of B.E degree in Information Science & Engineering. All the corrections suggested during the internal evaluation are incorporated in the project report. This report has been approved as it satisfies the academic requirements of mini project-3 prescribed for the Bachelor of Engineering degree.

Dr. Ananda Babu J
Assoc. Prof & Assoc. Dean AA

Dr. Chandrika J
Prof & Head of Dept

Name of the Examiners	Signature with Date
1.	
2.	

Department of Information Science & Engineering

Malnad College of Engineering

Hassan – 573202

2023-24

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We would like to express my gratitude to our respected director **Dr. S Pradeep** and principal **Dr. A J Krishnaiah** for providing a congenital environment and surrounding to work in. We would like to express our sincere gratitude to **Dr. Chandrika J**, Head of the department, information science and engineering for her continuous support and encouragement.

We are highly indebted **Dr. Ananda Babu J** for their guidance and constant supervision as well as for providing necessary information regarding the mini project & also for their support in completing the mini project.

We would like to express our gratitude towards my parents & member of Malnad College of Engineering for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

SUHAS K P

PAVAN B

SUGNYAN K

S M YATHIN

ABSTRACT

The project aims to automate the process of generating room allocation reports for educational institutions, facilitating efficient and error-free management of available resources. The system collects information on the number of available rooms, their capacities, and student details such as their unique serial numbers (USNs) and entry formats. Using this data, the script dynamically generates a PDF report utilizing the ReportLab library. The report includes room-wise allocation of seats to students, considering specified ranges of USNs and accommodating non-existing USNs. The user-friendly interface allows administrators to input relevant data, ensuring flexibility in adapting to different educational settings. The resulting PDF provides a comprehensive visual representation of room allocations, aiding institutions in organizing student seating arrangements systematically. This project streamlines administrative tasks, minimizes manual errors, and enhances the overall efficiency of room management in educational institutions.

CONTENTS

Chapter 1

Introduction

1.1 Introduction to the automated seat allocation system

Seat allotment systems play a pivotal role in several sectors including educational institutions, event management, and transportation, underpinning the organized and efficient utilization of space. These systems, leveraging algorithmic technological integration, ensure optimal allocation of seating to accommodate attendees, and students thus addressing the fundamental challenge of space management.

1.2 The Potential of the Problem

Manual seat allocation is time-consuming, prone to errors, and often fails to optimize space utilization. An automated system can streamline this process, ensuring fair and efficient allocations while accommodating specific requirements and preferences.

1.2.1 The Objective of the Present Work - Problem Description

The objective is to develop an automated Seat Allotment System that can dynamically allocate seats based on predefined criteria, ensuring optimal space utilization and addressing the limitations of manual processes.

1.2.2 Existing System and its drawbacks

For the seat allotment in our college. The staff of the examination section needs to manually distribute the students to their exam halls. This process needs to be repeated for every exam since different exam halls and rooms are available on different days.

Drawbacks

1. Time-Consuming - Manual processes are slow.

2. Error-Prone - Prone to human mistakes.

3. Data Redundancy - Multiple entries for similar data.

4Tracking Difficulty - Challenges in change tracking.

5. Allocation Delays - Dependent on faculty timing.

1.3 The objective of the present work - problem description:

1. Develop an Automated Seat Allotment System.

2. Dynamically allocate seats based on provided data.

3. Ensure optimal space utilization in the seat allocation process.

4. Address the limitations of manual seat allocation processes.

1.4 Platform and Tools Used:

Programming Language: Python

PDF Report Generation: ReportLab library

Database Connectivity: JDBC (Java Database Connectivity)

Database Management System: MySQL

Additional Library/API: iText API

The system incorporates JDBC for database connectivity, indicating compatibility across different platforms.

Chapter 2

System Analysis

2.1 Literature Survey:

A comprehensive literature survey was conducted, reviewing existing seat allotment systems, algorithms, and their implementations. This helped in understanding current challenges and technological solutions. The literature survey encompassed a broad spectrum of sources, including academic journals, industry reports, and case studies of existing seat allotment systems. The focus was on identifying best practices, common pitfalls, and technological trends in automated seat allocation. Notably, the survey highlighted the increasing reliance on algorithmic solutions to enhance space utilization and user satisfaction across sectors.

2.1.1. Exam Hall Management System Using PHP/MySQL

The Exam Hall Management System is a PHP project designed to automate exam allotment and seating arrangements. It utilizes PHP, HTML, CSS, JavaScript, MySQLi, and jQuery to assign students to classes and allocate seating to prevent conflicts. This system streamlines the examination process by managing locations through computer-generated hall arrangements. Additionally, it can generate reports related to student concerns during exams

2.1.2. Java JDBC API

The Java Database Connectivity (JDBC) API offers universal data access from Java to various data sources, including relational databases, spreadsheets, and flat files. It serves as a foundation for building tools and interfaces. The JDBC API consists of two packages: **java.sql** and **javax.sql**. To use the JDBC API with a specific database management system, a JDBC technology-based driver is required to facilitate communication between JDBC technology and the database

2.1.3. Best Python Libraries for Writing Reports to PDF

When writing structured data to PDF in Python, several libraries can be utilized. One approach involves using pandas, HTML, and weasyprint to write tables to PDF. Another method is creating PDF documents using the fpdf library. Choosing the right library depends on the specific requirements of the project. Libraries like reportlab, fpdf, and weasyprint offer different functionalities for writing structured data to PDF documents.

This literature survey covers diverse programming topics ranging from PHP/MySQL for exam management systems to Java JDBC APIs and Python libraries for generating PDF reports. Each topic provides valuable insights into different aspects of programming languages and tools used in software development.

2.2 Findings of Analysis

The analysis underscored a crucial demand for automated systems adept at managing intricate allocation criteria. Emphasizing the significance of generating user-friendly reports for administrative use, the project addresses this need by developing an Automated Seat Allotment System. The focus on simplicity and efficiency ensures a solution that not only streamlines complex processes but also enhances overall administrative effectiveness in handling exam-related tasks.

2.3 System Requirements Specification

The system requires Python and the ReportLab library. It should be capable of processing student data, managing room capacities, and generating allocation reports in PDF format.

2.3.1 Functional Requirements:

1. User LOGIN and RUN: Users can able to log in and run the program. If he does this, he can generate the pdf by providing some input data.
2. PDF Generation: A shareable PDF must be generated based on the user inputs.

2.3.2Non-Functional Requirements:

1. SOFTWARE REQUIREMENTS
   - Pycharm or Visual studio code
   - Os and Pdfgen library
   - mySQL database
   - APIs like ItextPdf and JDBC

2. HARDWARE REQUIREMENTS:
   - Memory: Minimum of 1GB RAM
     - For best results: 2GB RAM
   - Processor: Intel Core i5 Processor
   - Disk space: Minimum of 1GB disk space

Chapter 3

Design

3.1 Design of the Database

Data Schema: The schema includes a dictionary for available rooms where keys are room names and values are their capacities.

But for Java application we have created the database which led to data redundancy and helped to skip the user inputs. While designing the tables in the database we have maintained the constraints to the inserting tuples or relation sets to avoid duplicate values and invalid inputs. In our Java application, we've seamlessly integrated a MySQL database to manage various aspects of our system, including course offerings, departmental information, student records, faculty details, and room capacities. Leveraging the power of relational databases, we've meticulously designed tables to store this diverse array of data while minimizing redundancy and optimizing performance. Each table, from courses to rooms, has been crafted with precision to ensure data integrity and efficiency.

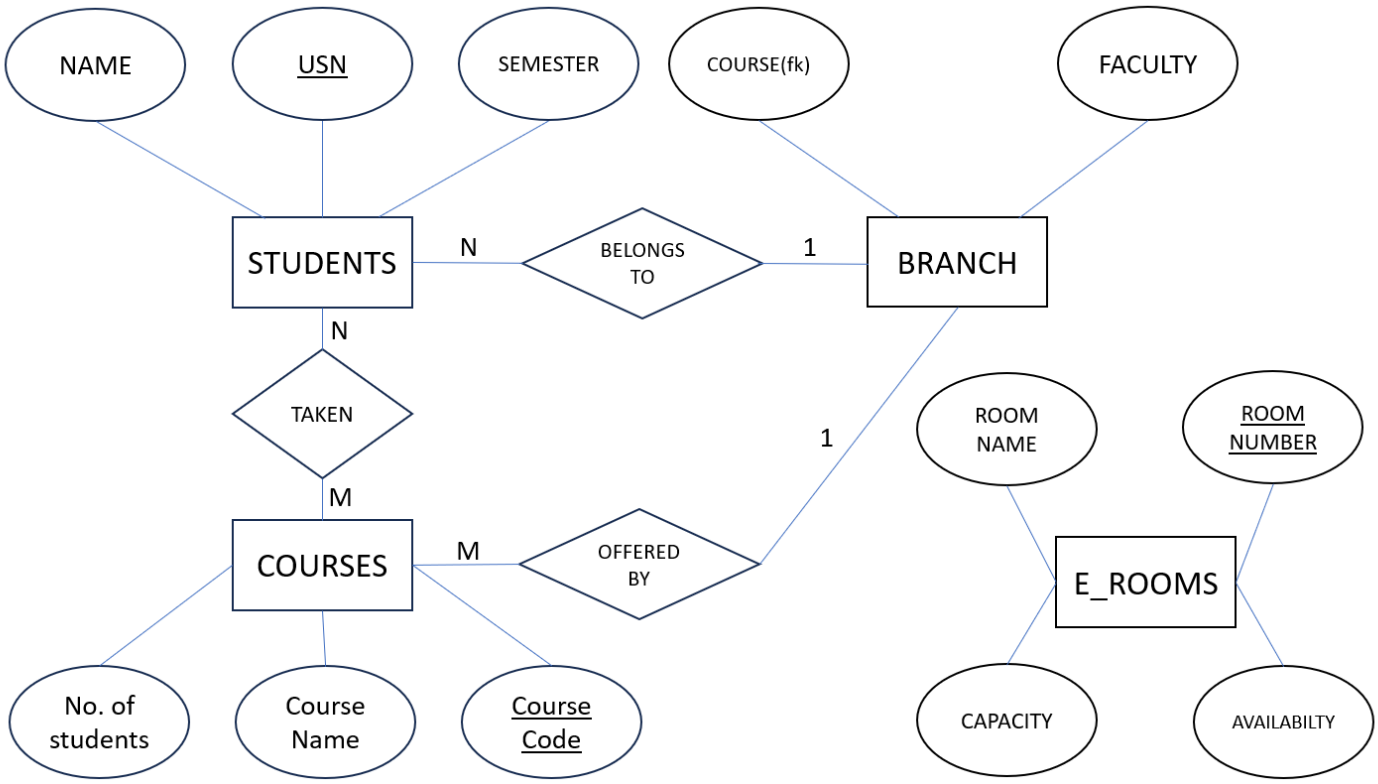
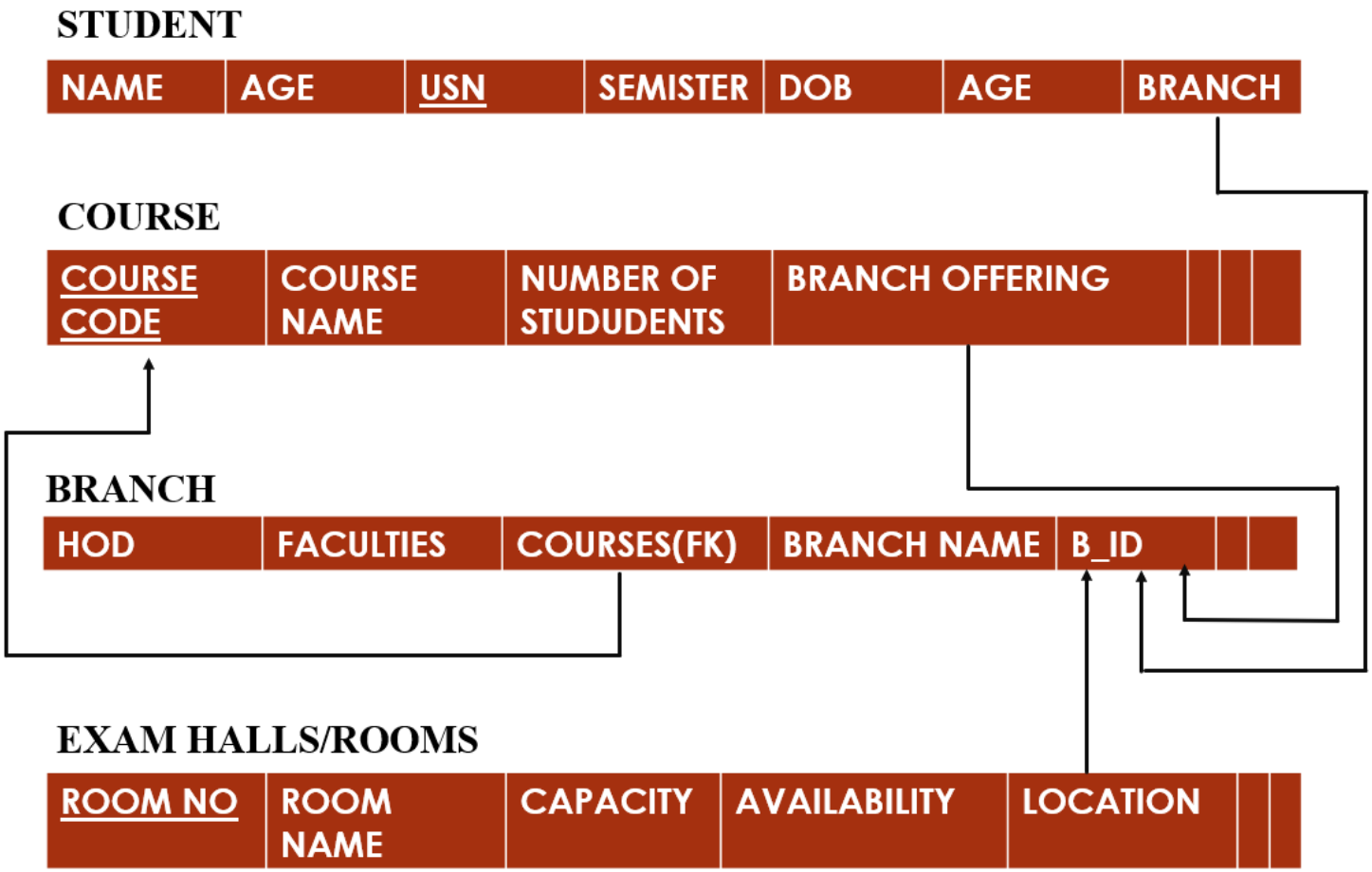


Fig.no.1 Entity relationship diagram of MCE database.

3.2 Schema of the database



3.2 Design of Functions

•Input Collection Functions: These functions prompt the user for all necessary inputs, including the number of rooms, room names and capacities, USN formats, and ranges. Input validation ensures data integrity.

•USN Generation Function: A dedicated function generates USNs based on specified formats and ranges, excluding any non-existing USNs, ensuring all potential attendees are accounted for.

•Seat Allocation Algorithm: This core function dynamically assigns seats to USNs, iterating through rooms based on their capacities and updating the seat numbers and available room data structures accordingly.

•PDF Report Generation Function: Utilizing the ReportLab library, this function creates a PDF document, iteratively adding allocated seat information in a structured and readable format.

3.3 Design of User Interface

•Command-Line Prompts: The system uses clear and concise prompts to guide the user through the data input process. Error handling mechanisms provide feedback for incorrect inputs, enhancing usability.

•Progress Indicators: During operations like USN generation and seat allocation, the system displays progress indicators to keep the user informed of the ongoing processes. 3.4 Design of Reports

Report Content: The PDF report includes a header with the report title and generation date, followed by the allocation details organized by room. Each entry includes the USN, allocated room, and seat number.

Formatting and Style: The report utilizes a clean, professional layout with adequate spacing and font sizes for readability. Room sections are separated, and a footer on each page includes page numbers.

Relevant code used for the room allocation process

```
 # Room Allocation Algorithm

import os

from reportlab.pdfgen import canvas

# Specify an absolute path for the PDF file in the current working directory

pdf_filename = os.path.join(os.getcwd(), "room_allocation_report.pdf")

# Function to generate PDF

def generate_pdf(existing_usn, available_rooms):

    pdf_canvas = canvas.Canvas(pdf_filename)

 # Set column widths and box dimensions

    box_width = 200

    box_height = 15

# Function to draw a cell (box)

    def draw_cell(x, y, width, height, text):

        pdf_canvas.rect(x, y, width, height)

        pdf_canvas.drawString(x + 5, y + 5, text)
```

```
# Add a pass statement if there is no content in the function yet
generate_pdf(None, None)  # Replace with actual parameters
    allocated_usn = set()
    # Iterate over rooms
    for room in available_rooms:
        pdf_canvas.drawString(100, 800, f"Room: {room}")
        draw_cell(100, 785, box_width, box_height, "Register Number (USN)")
        draw_cell(300, 785, box_width, box_height, "Seat Number")
        pdf_canvas.drawString(100, 775, "-" * 500)
        row_position = 770
        seat_number = 1
        capacity = available_rooms[room]
  # Iterate over students
        for usn in existing_usn:
                draw_cell(100, row_position, box_width, box_height, usn)
                draw_cell(300, row_position, box_width, box_height, str(seat_number))
                row_position -= box_height
                seat_number += 1
                capacity -= 1
                allocated_usn.add(usn)
   pdf_canvas.showpage()  # Add a new page for the next room
    pdf_canvas.save()
    print(f"\nPDF generated successfully: {pdf_filename}"
# Call the function to generate a PDF
generate_pdf(existing_usn, available_rooms)
```

Chapter 4

Implementation

The seat allotment system comprises several key modules, each designed to handle specific functionalities within the system. Here is a detailed description of the main modules:

User Input Module: This module is responsible for capturing user inputs through a command-line interface (CLI). Inputs include the number of available rooms, their names and capacities, the format and range of student identification numbers (USNs), and any exceptions in the sequence of USNs. This module ensures that the input is validated and stored correctly for further processing.

Seat Allotment Logic Module: At the heart of the system, this module implements the logic for allocating seats to students based on the available rooms and their capacities. It takes into consideration the USNs of regular and lateral entry students, ensuring that seats are allocated fairly and efficiently. The module also handles exceptions, such as non-existing USNs in the sequence, by skipping them during the allotment process.

PDF Report Generation Module: After the seat allotment process is complete, this module generates a comprehensive PDF report detailing the allocation results. Using the ReportLab library, it creates a document that lists each student's USN along with their allocated room and seat number. The report is formatted for clarity and ease of use during the actual seating arrangement.

Chapter 5

Testing

Testing is a crucial phase in the development of the seat allotment system, ensuring that the application is reliable, performs as expected, and meets the user requirements. Various testing methodologies were employed to thoroughly evaluate every aspect of the system. Here are the key testing strategies implemented:

•**Unit Testing:** This form of testing focuses on the smallest units of the software, such as individual functions or methods, to ensure they perform correctly in isolation. For the seat allotment system, unit tests were written for the user input validation functions, seat allotment logic, and PDF report generation routines. Mock objects were used to simulate database interactions where necessary. These tests helped in identifying and fixing bugs in the logic used for handling edge cases, such as non-existing USNs and full-room capacities.

•**Integration Testing:** After unit testing, integration testing was conducted to ensure that different parts of the application worked together seamlessly. This included testing the integration between the user input module and the seat allotment logic, as well as the integration between the seat allotment logic and the PDF report generation module. These tests uncovered issues with data passing between modules, particularly with the formatting of USNs and room allocations, which were subsequently corrected.

•**System Testing:** Conducted on a fully integrated system to evaluate the system's compliance with its specified requirements, system testing was performed to assess the entire seat allotment system's functionality. This included testing the system with various input scenarios, such as different numbers and configurations of rooms, and ranges of USNs, including edge cases with gaps in between the benches.

5.1 Test case and test results

Example input:

Current working directory: C:\Users\Suhas K P\PycharmProjects\pythonProject6\pythonProject40

Enter the number of available rooms:4

Enter the name of Room 1: CRB1

Enter the capacity of CRB1: 40

Enter the name of Room 2: CRB2

Enter the capacity of CRB2: 40

Enter the name of Room 3: CRB3

Enter the capacity of CRB3: 40

Enter the name of Room 4: CRB4

Enter the capacity of CRB4: 40

Enter the usn format of the students (Enter first 7 characters): 4MC21IS

Enter the usn format of lateral entry students: 4MC22IS

Only enter the integers:

Enter the starting usn: 001

Enter the ending usn: 123

Enter the starting usn of lateral entry: 400

Enter the ending usn of lateral entry students: 413

Enter the USNs (separated by commas) which do not exist in the middle:

{'CRB1': 40, 'CRB2': 40, 'CRB3': 40, 'CRB4': 40}

Pdf of Room allocation on the based input.

Room: CRB1

Register Number(USN)	Seat Number
4MC21ISOO1	1
4MC21IS002	2
4MC21IS003	3
4MC21IS004	4
4MC21IS005	5
4MC21IS006	6
4MC21IS007	7
4MC21IS008	8
4MC21IS009	9
4MC21IS010	10
4MC21IS011	11
4MC21IS012	12
4MC21IS013	13
4MC21IS014	14
4MC21IS015	15
4MC21IS016	16
4MC21IS017	17
4MC21IS018	18
4MC21IS019	19

Chapter 6

User Manual

6.1 Installation Procedure

To set up the seat allotment system, follow these step-by-step instructions. This guide assumes you have a basic understanding of your computer's operating system and have the necessary permissions to install software.

6.1.2 Prerequisites:

•Ensure Python (version 3.6 or later) is installed on your system. You can download it from the official Python website.

•Verify that pip, Python’s package installer, is installed by running pip --version in your command line or terminal.

6.1.3 Install Required Libraries:

•Open your command line or terminal.

•Navigate to the directory containing the system's source code.

•Install the required Python libraries by running pip install library name

Pip install reportlab.pdfgen

6.2. Running the Application:

•In the command line or terminal, while still in the project directory, start the application by running python main.py (assuming main.py is the entry point of the system).

•The system should now start, and you will be prompted to enter the necessary information via the command line or through a graphical user interface (GUI), if one is provided.

•the sequence of USNs. The system testing phase helped in identifying usability issues in the user input module, which were then refined to improve user experience

•While running the code make sure os is also imported.

•On left side you find the pdf document named room allocation report click on it , now you can see the allocation for students. You can share the document in a WhatsApp group or through mail.

Chapter 7

Conclusion

The seat allotment system project embarked on a mission to automate the process of allocating seats for various purposes, such as in educational institutions, event management, and transportation sectors. The objective was to create a system that not only simplifies the seat allocation process but also ensures efficient use of space, fairness in seat distribution and minimizes manual errors. The project has successfully achieved these objectives, marking a significant improvement over traditional manual allocation method.

Key achievements of the project include:

•**Automated Seat Allocation:** The system efficiently automates the allocation of seats, handling complex requirements and constraints with ease. It supports various scenarios, including regular and lateral entry students in educational settings, ensuring a fair and transparent allocation process.

•**Dynamic Room and Capacity Management:** The ability to input multiple rooms with different capacities allows the system to optimize space utilization, accommodating as many individuals as possible while adhering to specified constraints.

•**PDF Report Generation:** The generation of detailed PDF reports for each seat allocation session enhances the transparency and usability of the system. These reports serve as official documentation of the allocation process, which can be easily shared and reviewed.

•**User-Friendly Interface:** The system is designed with a focus on user experience, featuring an intuitive interface that guides users through the input and allocation process. This minimizes the learning curve and ensures that users can effectively utilize the system without extensive training.

•**Flexibility and Scalability:** Built with flexibility in mind, the system can easily be adapted to different contexts and scales, from small events to large institutions. Its modular design allows for easy updates and the addition of new features.

References

[1]. Project Article Submitted by Nikhil B on July 2, 2021 named "Exam Hall Management System Full Source Code Using PHP/MySQL". Available:https://www.sourcecodester.com/php/14205/exam-hall-management-system-full-source-code-using-phpmysql.html

[2]. Synopsis of Oracle Java JDBC application program interface documentation and guides for sequential databases. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/

[3]. Tech Blog named "Best Python Libraries to Write Reports to PDF" By Dinesh Kumar K B. Available: https://plainenglish.io/blog/best-python-libraries-to-write-reports-to-pdf-87771be815c9

[4]. You tube video tutorial playlist on "how to use itextPDF api" by Sarthi Technology. Available: https://www.youtube.com/watch?v=SZEhv8tpT6U&list=PLFh8wpMiEi8-Yo59DBCasuVi1M29kQrvn.

[5]. Geeks for geeks website for shuffling algorithm name "Shuffle a given array using Fisher–Yates shuffle Algorithm". Available: https://www.geeksforgeeks.org/shuffle-a-given-array-using-fisher-yates-shuffle-algorithm/

[6]. W3schools blog website named "Java Packages and Application program interfaces(APIs)". Available: https://www.w3schools.com/java/java_packages.asp