

A SYNOPSIS ON

HOSTEL BOOKING SYSTEM

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

Submitted by:

Shivam Kaushik 2261526

Navneet Pandey 2261384

Rohit Joshi 2261487

Abhinav Singh Bohra 2261063

Under the Guidance of

Mr. Anubhav Bewerwal

Assistant Professor, Department of Computer Science & Engineering

Project Team ID: 12



Department of Computer Science & Engineering

Graphic Era Hill University, Bhimtal, Uttarakhand

March-2025

CANDIDATES' DECLARATION

I/We hereby certify that the work which is being presented in the Synopsis entitled “**Hostel Booking System**” in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science & Engineering of the Graphic Era Hill University, Bhimtal campus and shall be carried out by the undersigned under the supervision of **Mr. Anubhav Bewerwal, Assistant. Professor**, Department of Computer Science & Engineering, Graphic Era Hill University, Bhimtal.

Team Members	University Roll Numbers	Signature
Shivam Kaushik	2261526	
Navneet Pandey	2261384	
Abhinav Singh Bohra	2261063	
Rohit Joshi	2261487	

The above mentioned students shall be working under the supervision of the undersigned on the “**Hostel Booking System**”

Signature

Supervisor

Mr. Anubhav Bewerwal

Signature

Head of the Department

Dr. Ankur Singh Bist

Internal Evaluation (By DPRC Committee)

Status of the Synopsis: Accepted / Rejected

Any Comments:

Name of the Committee Members:

Signature with Date

- 1.
- 2.

Table of Contents

Chapter No.	Description	Page No.
Chapter 1	Introduction and Problem Statement	4
Chapter 2	Background/ Literature Survey	6
Chapter 3	Objectives	8
Chapter 4	Hardware and Software Requirements	9
Chapter 5	Possible Approach/ Algorithms	10
	References	14

Chapter 1

Introduction and Problem Statement

Introduction

The rapid growth of educational institutions has amplified the need for efficient hostel management systems. Hostels are a critical component of campus life, providing accommodation to students from diverse backgrounds. However, traditional hostel management methods often rely on manual processes, such as paper-based applications, physical record-keeping, and in-person payment collection. These outdated practices lead to inefficiencies, errors, and delays, creating frustration for both students and administrators.

Institutions like Graphic Era Hill University host thousands of students annually, making manual management unsustainable. A **Hostel Booking System** addresses these challenges by automating key processes such as room allocation, payment tracking, and real-time availability updates. Modern web technologies like **HTML**, **CSS**, **JavaScript (frontend)**, **PHP (backend)**, and **MySQL (database)** offer a robust framework to build such a system. These tools enable seamless integration of user interfaces, server-side logic, and secure data storage, ensuring scalability and reliability.

Existing solutions often lack critical features such as:

1. **Real-time updates** on room availability.
2. **Priority-based room allocation** (e.g., seniority or merit).

This project aims to bridge these gaps by developing a **web-based Hostel Booking System** that prioritizes user experience, security, and administrative efficiency. By leveraging PHP's server-side capabilities and MySQL's relational database management, the system will streamline operations, reduce human error, and enhance transparency.

Problem Statement

The current hostel management process at many institutions involves the following challenges:

1. Manual Application and Allocation:

- Students submit physical forms to the hostel office, which are manually verified and processed.
- Administrators use spreadsheets or paper records to assign rooms, leading to errors like overbooking or duplicate entries.
- Example: A student might receive a room assignment that was already allocated to another student due to outdated records.

2. Lack of Real-Time Updates:

- Students cannot check room availability in real time, leading to confusion and repeated visits to the hostel office.
- Administrators struggle to update records promptly, resulting in mismatches between actual and recorded room statuses.

3. Inefficient Communication:

- Notifications about payment deadlines, room changes, or policies are disseminated via notice boards or word of mouth, causing delays and missed information.

4. Security Risks:

- Physical records are vulnerable to damage, loss, or unauthorized access.
- Financial transactions (e.g., hostel fees) are handled in cash, increasing the risk of fraud.

5. Scalability Issues:

- Manual systems cannot handle surges in demand during admission seasons, leading to bottlenecks.

Chapter 2

Background/ Literature Survey

Traditional Hostel Management Systems

Historically, hostel management in educational institutions relied on **manual processes**, including paper-based applications, physical ledgers for record-keeping, and face-to-face interactions for room allocation. Studies such as [1] highlight that these systems are prone to human error, data redundancy, and inefficiency. For example, a survey conducted by Kumar et al. [2] revealed that 65% of institutions using manual systems reported frequent overbooking incidents due to outdated records.

Manual systems also lack transparency. Students often have no visibility into room availability or allocation criteria, leading to dissatisfaction and disputes [3]. Administrators, meanwhile, face challenges in auditing financial transactions, as cash-based payments are difficult to track and secure.

Early Digital Solutions

With the advent of digital tools, institutions began adopting **spreadsheet-based systems** (e.g., Microsoft Excel) to manage hostel data. While these systems reduced paperwork, they introduced new limitations:

- **Data Fragmentation:** Multiple spreadsheets for rooms, students, and payments led to inconsistent records.
- **Limited Accessibility:** Files stored on local machines hindered remote access and collaboration.
- **Security Risks:** Unencrypted files were vulnerable to unauthorized access or corruption.

A case study by Sharma and Patel [4] on a college in Maharashtra demonstrated that spreadsheet-based systems failed during peak admission periods due to their inability to handle concurrent updates.

Web-Based Hostel Management Systems

The rise of web technologies in the 2010s spurred the development of **online hostel management systems**. Below is a review of key studies and existing solutions:

PHP-MySQL Systems

Several institutions have adopted PHP and MySQL for hostel management due to their cost-effectiveness and ease of integration:

- **Study by Gupta et al. [5]:** Developed a PHP-MySQL system for room allocation but lacked real-time updates. Administrators had to manually refresh the interface to reflect changes.
- **HostelX System [6]:** A commercial tool offering basic booking features. However, it used plaintext password storage, posing security risks.

Responsive Interfaces

Recent projects emphasize **user-friendly interfaces** using HTML, CSS, and JavaScript:

- **CampusHostel [7]:** A JavaScript-driven system with dynamic forms and live validations. However, it relied on a flat-file database, limiting scalability.
- **Study by Lee et al. [8]:** Implemented AJAX for real-time room availability checks but excluded payment gateway integration.

Security Practices

Secure authentication remains a critical focus:

- **Research by Singh and Rao [9]:** Advocated for PHP's `password_hash()` function to encrypt user credentials, reducing vulnerabilities.
- **Work by Al-Mamun et al. [10]:** Highlighted SQL injection risks in PHP-MySQL systems and recommended prepared statements for mitigation.

Chapter 3

Objectives

The objectives of the proposed work are as follows:

1. To develop a web-based Hostel Booking System using HTML, CSS, JavaScript (frontend), PHP (backend), and MySQL (database) to automate the hostel booking process, replacing error-prone manual workflows.
2. To design a responsive and user-friendly interface that enables students to:
 - View real-time room availability.
 - Book accommodations with instant confirmation.
 - Track payment status and booking history.
3. To ensure robust security through:
 - Secure user authentication using PHP sessions.
 - Password hashing (password_hash()) for credential protection.
 - Prepared SQL statements to prevent injection attacks.
4. To equip administrators with a centralized dashboard for:
 - Managing room allocations and cancellations.
 - Monitoring payments and generating financial reports.
 - Updating hostel policies and student records.

Chapter 4

Hardware and Software Requirements

4.1 Hardware Requirements

Sl. No	Name of the Hardware	Specification
1	Processor	Intel Core i3 (Minimum) / Intel Core i5 or higher (Recommended)
2	RAM	4GB (Minimum) / 8GB or higher (Recommended)
3	Storage	At least 20GB of free disk space
4	Network	Active internet connection for remote log transmission
5	Operating System	Windows 10/11, Linux (Ubuntu, Kali, Debian)

4.2 Software Requirements

Sl. No	Name of the Software	Specification
1	Frontend	HTML, CSS, JavaScript
2	Backend	PHP 8.x
3	Database	MySQL 8.0

Chapter 5

Possible Approach/ Algorithms

5.1 System Architecture Overview

The Hostel Booking System is structured as a **three-tier architecture**:

1. **Frontend**: User interface built with HTML, CSS, and JavaScript.
2. **Backend**: PHP scripts handling business logic (e.g., authentication, room allocation).
3. **Database**: MySQL for storing user data, room details, and transactions.

5.2 Core Algorithms

5.2.1 Priority-Based Room Allocation

Objective: Assign rooms to students based on seniority and preferences.

Pseudo-Code:

1. Retrieve list of registered students with confirmed payments.
2. Sort students by:
 - a. Enrollment year (descending order: senior students first).
 - b. Room preference (e.g., AC rooms prioritized).
3. For each student in the sorted list:
 - a. Query the database for available rooms matching their preference.
 - b. If a matching room is found:
 - i. Mark the room as "occupied" in the database.

- ii. Create a booking record linking the student and room.
 - iii. Send a confirmation email to the student.
- c. If no matching room is found:
 - i. Assign the next available room of any type.
- 4. Log allocation results for administrative review.

5.2.2 Secure Authentication Workflow

Objective: Ensure only authorized users access the system.

Pseudo-Code:

1. Registration Process:

- a. Capture user details (name, email, password).
- b. Encrypt the password using a secure hashing algorithm.
- c. Store the user's email and hashed password in the database.

2. Login Process:

- a. Capture user email and password input.
- b. Retrieve the stored hashed password for the given email.
- c. Compare the input password's hash with the stored hash.
- d. If matches:
 - i. Generate a session token for the user.
 - ii. Redirect to the booking dashboard.

e. If no match:

i. Display "Invalid credentials" error.

5.2.3 Real-Time Room Availability Check

Objective: Update room status dynamically without page reloads.

Pseudo-Code:

1. Client-Side (Frontend):

a. Periodically request room data from the server.

b. Update the UI to reflect current availability.

2. Server-Side (Backend):

a. Receive request for room status.

b. Query the database for all rooms and their current status.

c. Return the data in a structured format (e.g., JSON).

5.2.4 Payment Transaction Handling

Objective: Securely process online payments.

Pseudo-Code:

1. Payment Initialization:

a. Collect booking details (student ID, amount).

b. Generate a unique transaction ID.

c. Redirect the user to a third-party payment gateway.

2. Payment Verification:

- a. Receive payment confirmation from the gateway.
- b. Validate the transaction signature.
- c. If valid:
 - i. Update the booking's payment status to "completed".
 - ii. Send a payment confirmation email.
- d. If invalid:
 - i. Mark the transaction as "failed" in the database.

Conclusion

The Hostel Booking System leverages a three-tier architecture to streamline room allocation, authentication, and payments. With a responsive frontend, secure PHP backend, and real-time MySQL database, it ensures efficient, automated hostel management. Key algorithms prioritize students based on seniority and preferences, while robust authentication and payment workflows enhance security and user experience.

References

- [1] R. Thompson, "Challenges in Manual Hostel Management," *J. Edu. Admin.*, vol. 15, pp. 22–35, 2018.
- [2] A. Kumar et al., "Efficiency Analysis of Hostel Management Systems," *Proc. Int. Conf. on Edu. Tech.*, 2020.
- [3] S. Verma, "Student Satisfaction in Hostel Allocation," *IEEE Trans. Educ. Syst.*, vol. 7, no. 2, 2019.
- [4] P. Sharma & R. Patel, "Digital Transformation in Hostel Management," *J. Inf. Syst. Educ.*, 2021.
- [5] M. Gupta et al., "A PHP-MySQL Hostel System," *Int. J. Comp. Appl.*, vol. 45, no. 3, 2022.
- [6] *HostelX Software Documentation*, 2021. [Online].
- [7] *CampusHostel Project, GitHub Repository*, 2022. [Online].
- [8] J. Lee et al., "AJAX in Hostel Booking Systems," *Proc. IEEE Conf. on Web Dev.*, 2023.
- [9] V. Singh & K. Rao, "Security in PHP-MySQL Systems," *Comp. Sec. J.*, vol. 12, pp. 50–65, 2023.
- [10] S. Al-Mamun et al., "SQL Injection Mitigation," *J. Cyber Sec.*, vol. 8, no. 1, 2022.
- [11] *MDN Web Docs*, "AJAX Introduction," 2023. [Online].
- [12] A. Joshi et al., "Student Preferences in Digital Hostel Systems," *Edu. Tech. Res.*, vol. 30, 2023.