

HOSTEL MANAGEMENT SYSTEM

A Project Report

Submitted in fulfilment of the
Requirement for the award of
the degree of

Master of Computer Applications
(Batch 2020- 2023)

To



By
Sandeep
(52010035)

Amdocs Training

DEPARTMENT OF COMPUTER APPLICATIONS
NATIONAL INSTITUTE OF TECHNOLOGY, KURUKSHETRA
May/June/July 2023

DECLARATION

Respected Sir/Mam,

I hereby declare that the work which is being presented in the attached project **report** entitled “***Hostel Management System***”, in partial fulfilment of the requirement for the award of the degree of ***MASTER OF COMPUTER APPLICATIONS*** submitted to the **Department of Computer Applications, National Institute of Technology, Kurukshetra** is an authentic work done by me during a period from **January 02,2023**, to till date **under the Guidance of Dr Kapil and AMDOCS learning partner Techademy.**

However, the internship will continue till **June 30th, 2023.**

The work presented in this project report has not been submitted by me for the award of any other degree of this or any other Institute/University.

Regards

Sandeep

52010035

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all individuals who have contributed to the ongoing development of the hostel management system. Firstly, we extend our heartfelt appreciation to our project supervisor for their continuous guidance and support throughout this endeavour. Their expertise and valuable suggestions have been instrumental in shaping the project and ensuring its progress.

We would also like to acknowledge the efforts and contributions of our dedicated team members who have worked tirelessly on different aspects of the project. From front-end handling to API generation and report generation, their dedication and hard work have significantly advanced the project.

We want to emphasise that this project is still in the developing mode, and we are committed to further enhancing its features and functionality. The continued support and collaboration of all involved parties are vital as we strive to create a robust and efficient hostel management system.

I am grateful to Dr Kapil for their precious guidance related to design system architecture of the project.

I would like to express my sincere gratitude to the following organisations for their invaluable contributions to my research:

Techademy/Amdocs provided me with a basic React course, which laid a solid foundation for my understanding of React and JavaScript. I am grateful to the instructors at Techademy/Amdocs for their clear and concise explanations, which made it easy for me to understand the concepts.

Thank you all for your valuable contributions to this project.

AMDOCS TRAINING PLANNER

Month	Week	Monday	Tuesday	Wednesday	Thursday	Fun Friday	Saturday	Sunday	
Jan, 2023	week 1	02-Jan-23	03-Jan-23	04-Jan-23	05-Jan-23	06-Jan-23	07-Jan-23	08-Jan-23	
						AFH 2:00pm-5:00pm			
	week 2	09-Jan-23	10-Jan-23	11-Jan-23	12-Jan-23	13-Jan-23	14-Jan-23	15-Jan-23	
		Learners Connect With CSM		ALH 5:00pm-6:00pm			Core Java VILT 10:00am-12:30pm 2:00pm-4:30pm	ALHT 10:00am-11:00am 11:30am-12:30pm	
	week 3	16-Jan-23	17-Jan-23	18-Jan-23	19-Jan-23	20-Jan-23	21-Jan-23	22-Jan-23	
				ALH 5:00pm-6:00pm		AFH 2:00pm-5:00pm			
	week 4	23-Jan-23	24-Jan-23	25-Jan-23	26-Jan-23	27-Jan-23	28-Jan-23	29-Jan-23	
		Learners Connect With CSM				AFH 2:00pm-5:00pm	Core Java VILT 10:00am-12:30pm 2:00pm-4:30pm	ALHT 10:00am-11:00am 11:30am-12:30pm	
	week 5	30-Jan-23	31-Jan-23	01-Feb-23	02-Feb-23	03-Feb-23	04-Feb-23	05-Feb-23	
				ALH 5:00pm-6:00pm		AFH 2:00pm-5:00pm			
Feb, 2023	week 6	06-Feb-23	07-Feb-23	08-Feb-23	09-Feb-23	10-Feb-23	11-Feb-23	12-Feb-23	
		Learners Connect With CSM		ALH 5:00pm-6:00pm		AFH 2:00pm-5:00pm	MySQL VILT 10:00am-12:30pm 2:00pm-4:30pm		
	week 7	13-Feb-23	14-Feb-23	15-Feb-23	16-Feb-23	17-Feb-23	18-Feb-23	19-Feb-23	
				ALH 5:00pm-6:00pm		AFH 2:00pm-5:00pm	MySQL VILT 10:00am-12:30pm 2:00pm-4:30pm	ALHT 10:00am-11:00am 11:30am-12:30pm	
	week 8	20-Feb-23	21-Feb-23	22-Feb-23	23-Feb-23	24-Feb-23	25-Feb-23	26-Feb-23	
	week 9	27-Feb-23	28-Feb-23	01-Mar-23	02-Mar-23	03-Mar-23	04-Mar-23	05-Mar-23	

	06-Mar-23	07-Mar-23	08-Mar-23	09-Mar-23	10-Mar-23	11-Mar-23	12-Mar-23	
Mar, 2023	week 10	Learners Connect With CSM				AFH 2:00pm-5:00pm	Unix VILT 10:00am-12:30pm 2:00pm-4:30pm	
	week 11	13-Mar-23	14-Mar-23	15-Mar-23	16-Mar-23	17-Mar-23	18-Mar-23	19-Mar-23
				ALH 5:00pm-6:00pm		AFH 2:00pm-5:00pm	Unix VILT 10:00am-12:30pm 2:00pm-4:30pm	ALHT 10:00am-11:00am 11:30am-12:30pm
	week 12	20-Mar-23	21-Mar-23	22-Mar-23	23-Mar-23	24-Mar-23	25-Mar-23	26-Mar-23
	week 13	27-Mar-23	28-Mar-23	29-Mar-23	30-Mar-23	31-Mar-23		
				ALH 5:00pm-6:00pm		ALHT 10:00am-11:00am 11:30am-12:30pm		

Course Name	Duration	Relevant YouTube Links	Start date	End date	Mentor Connect Session	Day	Slot
Python for Beginners	3h 23m		28-Mar-23	30-Mar-23			NA
Threading and Multiprocessing	1h 42m		30-Mar-23	3-Apr-23			
Apache Spark with Python - Big Data with PySpark and Spark	4h 17m		4-Apr-23	7-Apr-23	22-Apr-23	Saturday	10:00AM - 2:00PM
Introduction to Cloud Computing	3h 48m		10-Apr-23	12-Apr-23			
Fundamentals of JavaScript for App Development	4h 29m		13-Apr-23	19-Apr-23			
Learning React with Redux and Flux	4h 21m		20-Apr-23	26-Apr-23			
Microservices Architecture	2h 21m		27-Apr-23	5-May-23			
Email Etiquette	2h 26m		8-May-23	10-May-23			
Advanced English Grammar	16m	Tips to Improve Your Sentence Structure - https://youtu.be/DvxDiD9xWhe	11-May-23	11-May-23			NA
NA	23m	English Grammar - Error Identification & Correction - https://youtu.be/1Dax9Q0qXal	12-May-23	12-May-23			
	12m	How to Give and Receive Feedback in English - Business English Lesson - https://youtu.be/lIVBmMfflww	15-May-23	17-May-23			
	8m	The overlooked art of receiving feedback - https://youtu.be/fIyBmAIIiww	18-May-23	18-May-23			
			19-May-23	19-May-23			

Figure 1: Planner

ABSTRACT

The “Hostel Management System” is a web application designed to efficiently manage various activities in hostels. With the increasing number of educational institutions, there has been a corresponding rise in the number of hostels to accommodate students. However, manual management of hostels can be challenging and puts a strain on hostel administrators. To address these challenges, this project aims to develop a computerised system that overcomes the drawbacks of manual management and improves efficiency.

By identifying the limitations of the existing system, this software application is designed to be compatible with the current setup while offering a more user-friendly interface. The system aims to streamline hostel management processes and minimise errors that may occur when carried out manually.

Through the implementation of this online system, supervisors can improve their efficiency, overcome the drawbacks of the existing system, and enhance the overall management of hostels.

ABOUT COMPANY

Amdocs is a multinational corporation that was founded in Israel and is currently headquartered in Chesterfield, Missouri, with support and development centres located worldwide. The company specialises in software and services for communications, media, financial services providers and digital enterprises. It is a leading software and services provider to communications and media companies of all sizes, accelerating the industry's dynamic and continuous digital transformation. With a rich set of innovative solutions, long-term business relationships with 350 communications and media providers, and technology and distribution ties to 600 content creators, Amdocs delivers business improvements to drive growth.

Amdocs generates long-term business value and impact by leveraging academic collaboration from cutting-edge research and technology. In cooperation with leading academic institutions, we create thought leadership platforms that envision and evaluate future new directions while maximising joint innovation initiatives.

As Amdocs works mainly in the Telecom Domain. It has its own suite of products, in OSS and BSS. It does sell those Products across the globe and provides services to the vendors. Amdocs employees work on a plethora of technologies ranging from legacy systems to the latest day platforms. Technology is client and application specific. Amdocs BSS, OSS, network control and optimization solutions, coupled with its professional and managed services, have accelerated business value for their telecom customers by streamlining complex operating environments, reducing costs and speeding time to market for new products and services.

Our Intellectual property

Over the past three years, we've tripled our number of patents, placing us in the top 1,000 U.S. company patent owners. Our patents are spread across many domains in areas like NFV, artificial intelligence, IoT, operations support systems and business support systems.

Our Delivery Approach

The implementation of DevOps for our customers is no easy matter. It requires a deep understanding of the entire ecosystem, including backbone applications that must be seamlessly integrated with new innovative solutions. This is where we see the Amdocs advantage. Our comprehensive knowledge of the ecosystem, data models and innovative digital offerings – combined with DevOps capabilities – provide our customers with the right solution, with full accountability. Through our technologies and practices, we run projects with compelling results:

- Deployed full consumer digital stack within **4.5** months
- Increased release frequency from **3 to 7** times a year
- Achieved **20%** improvement in time to production
- Reduced defect density by up to **40%**
- Realised **~30%** reduction in escaping defects

Some of our clients

Amdocs' customer base includes communications and media companies. Through our relationships with over 600 content creators and 350 communications and media providers across 85 countries, we serve billions of consumers worldwide. We help our customers achieve their most audacious business goals and meet their everyday needs.



Figure 2: Some of our Clients

A design-led organisation

Our unique approach to problem-solving uses design thinking while focusing on the human aspect. With design thinking embedded into our processes, we begin with the needs of the end users as a first step to achieving the solution. The design-thinking methodology supports our customers with valuable new insights about the needs of their customers.

A content play unlike any other

With technology and distribution ties to over 600 content creators, Amdocs is uniquely positioned to help communications and media providers monetize premium content effectively.

Centre of excellence

The Amdocs Innovation Centre of Excellence addresses business challenges, creates an innovation culture, and emphasises the importance of innovation execution.

Demonstrated leadership across the Amdocs business

Analyst reports highlight Amdocs continued leadership in digital transformation, network virtualization and agile operations.

Amdocs was named as a leader in Gartner's 2019 Magic Quadrant for Integrated Revenue And Customer Management and Operations Support Systems.

Amdocs won the Leading Lights' 2018 Company of the Year (Public) award, a prize that recognizes "the publicly listed firm that stands out from its competitors, innovates constantly, makes investors proud, and makes employees happy."

Training at Amdocs:

Amdocs works on various technical domains like Java/J2EE, C, C++, and SQL as programming languages and hence provides students with training along with job offers on different technological platforms.

Each intern at Amdocs must implement a series of assignments to get hands-on with the technological platform he/she is trained on. These assignments proved to be really challenging and demanded great effort from my side. Each assignment is thoroughly reviewed by the supervisor. Apart from this, there are Mentors (technical persons of the team) who are assigned to each trainee for guidance throughout the project. After having a good hand on the assignments, the interns are provided with a real-life project.

LIST OF FIGURES

Figure 1. Planner	III
Figure 2. Some of our Customers	VI
Figure 5.3.1 Home Page	25
Figure 5.3.2 Student Registration	26
Figure 5.3.3 i) Student Successful Registration	28
Figure 5.3.3 ii) Student Database	29
Figure 5.3.4 Student Successful Login	30
Figure 5.3.5 Student Dashboard	31
Figure 5.3.6 Student Profile	33
Figure 5.3.7 Student Hostel Card	34
Figure 5.3.8 Student Rebate Form	36

TABLE OF CONTENTS

DECLARATION	I
ACKNOWLEDGMENT	II
AMDOCS TRAINING PLANNER	
ABSTRACT	
ABOUT COMPANY	
LIST OF FIGURES	IV
CHAPTER 1- INTRODUCTION	1-5
1.1 Introduction	1
1.2 Motivation	2
1.3 Objective	3
1.4 Organization of the Report	4
CHAPTER 2 – LITERATURE REVIEW	6-9
2.1 Existing System	6
2.2 Drawbacks of Previous System	7
2.3 Tools and Technology used in Previous Method	8
CHAPTER 3 - THE PROPOSAL	10-14
3.1 Proposed System	10
3.2 Proposed System Requirement	12
3.3 Project Planning	13
CHAPTER 4 - DESIGN	15-16

4.1 Program Structure	15
CHAPTER 5 - IMPLEMENTATION DETAILS & RESULTS	17-36
5.1 Tools and Technology Used	17
5.2 Installation and Setup	19
5.3 Discussion on Implementation	22
CHAPTER 6 - CONCLUSION AND FUTURE SCOPE	37
6.1 Conclusion	37
6.2 Future Scope	37
6.3 Contribution	37

Chapter 1: INTRODUCTION

1.1 INTRODUCTION

In the current era of automated systems, relying on manual systems for hostel management is not advisable. Manual systems often involve repetitive tasks that consume significant time and effort, leading to inefficiencies and potential errors. These systems heavily rely on paper-based processes for tasks such as registration form verification and data storage, which can result in difficulties when retrieving student records, managing bills, and accessing information about past students.

The drawbacks of manual systems are numerous. First, The repetitive nature of manual tasks not only consumes valuable resources but also increases the risk of human error. This can lead to data discrepancies, misplaced records, and inaccuracies in room allocations or payment tracking. Additionally, relying on paper-based processes makes it challenging to efficiently manage and retrieve student records when needed, particularly in large hostels with a significant number of residents.

Furthermore, manual systems often lack the capability to generate comprehensive reports or perform data analysis. This makes it difficult for hostel administrators to gain insights into occupancy rates, revenue trends, or students preferences, hindering effective decision-making and strategic planning.

To address these drawbacks, a computerised hostel management system is designed to streamline and automate various processes. This system leverages technology to eliminate repetitive tasks and reduce the reliance on paper-based processes. By transitioning to an automated system,hostel management can improve operational efficiency and overcome the limitations of manual systems.

The primary objective of a computerised hostel management system is to assist supervisors in maintaining accurate and up-to-date records of student details, room allocations, and

financial transactions. Through automation, the system simplifies tasks such as registrations, form verification, fee collection, and invoice generation. Additionally, it provides centralised access to student records, enabling swift retrieval and efficient management of mess bills, attendance, and disciplinary actions.

1.2 MOTIVATION

The motivation behind developing the Hostel Management Project stemmed from the challenges faced by universities in efficiently managing their hostel facilities. Traditional manual methods of managing hostels often resulted in various issues, such as administrative inefficiencies, difficulty in maintaining records, and a lack of transparency. The following factors served as the primary motivations for undertaking this project:

Streamlining Administrative Processes: Manual management of hostels involves a significant amount of paperwork and manual effort. By developing a digital solution, the project aimed to streamline administrative processes such as student registration, room allocation, fee management, and maintenance requests. Automating these tasks reduces administrative burden, minimises errors, and improves overall efficiency.

Enhancing Transparency: The project aimed to introduce transparency in the hostel management system. By providing students, parents, and administrators with access to relevant information, the project ensured a clear understanding of room availability, fee structures, and maintenance schedules. Improved transparency fosters trust and eliminates ambiguities.

Optimising Resource Allocation: Effective resource allocation is crucial in managing hostels. The project sought to optimise the allocation of rooms based on student preferences, availability, and specific requirements. By automating this process, the project aimed to minimise conflicts and maximise student satisfaction.

1.3 OBJECTIVE

The objectives of the Hostel Management System are designed to address key challenges and improve various aspects of hostel management. Let's understand each and every objective:

1. This objective aims to centralise and streamline the management of critical information related to rent, allottees, hostels, rooms and payments. This ensures accurate and up-to-date records, simplifying tasks such as rent calculation, allottee tracking, and managing payment information.
2. Reducing manual work involved in managing rent, allottees, beds, and hostels. One of the primary goals of the Hostel Management System is to minimise manual effort and increase operational efficiency. This objective aims to optimise resource utilisation. Eliminate repetitive work, and enhance overall productivity in hostel management operations.
3. Providing an application program for supervisors to handle various aspects of hostel management. This objective focuses on developing a user-friendly and comprehensive application program that caters to the specific needs of hostel supervisors.
4. The Hostel Management System aims to provide robust search capabilities that allow supervisors to retrieve information based on specific criteria. This includes searching for specific rent details, hostels, available rooms, payment records, and other relevant factors.
5. Facilitating the online availability of student details, employee details, and course information. This objective focuses on leveraging the system to make essential information easily accessible online. By enabling students to access their details, such as room assignments, fee statements, and attendance records, online, the system enhances convenience and transparency.
6. Monitoring information and transactions related to rooms. This objective aims to enable supervisors to monitor and track room-related information and transactions effectively. It includes features such as room occupancy tracking, maintenance

schedules, and room change requests. By having real-time visibility into room status and related activities, supervisors can proactively address issues, ensure timely maintenance, and optimise room utilisation.

1.4 ORGANISATION OF THE REPORT

Chapter-1 Introduction: The introduction section highlights the drawbacks of manual hostel management systems, emphasising the need for an automated solution. The primary objective of the project is to develop a computerised hostel management system.

Chapter-2 Literature Review: The literature review examines existing research and solutions in the field of hostel management systems. The review emphasises the importance of user interface design, scalability, data analytics, and security measures. It serves as a foundation for the proposed solution by providing insights into technologies, frameworks, and methodologies used in similar projects.

Chapter-3 Proposal: The proposal outlines the scope, objectives, and methodology of the hostel management project. It presents a comprehensive plan for developing a computerised system that addresses the identified challenges. The proposal includes details about the system architecture, user roles, functionalities and it also highlights the use of technologies.

Chapter-4 Design: In the design, The program structure of the hostel management system follows a modern and scalable architecture. It utilises ReactJS for the front-end, Node.js for the back-end, and MongoDB as the database. The system adopts a component-based approach, ensuring reusability and consistency in user interface design. It incorporates user authentication and authorization using JSON Web Tokens (JWT) and implements error handling and data validation mechanisms.

Chapter-5 Implementation Details and Results: In the implementation of the hostel management system involved the use of modern technologies such as ReactJS, Node.js, Express.js, and MongoDB. The implementation also focused on user experience and security. By utilising JSON Web Tokens (JWT) for authentication and authorization, the system ensured secure access to different user roles. Error handling, input validation, and modern frameworks further enhanced the system's performance and reliability.

Chapter-6 Conclusion and Future Scope: The conclusion emphasises the advantages of adopting a hostel management system, such as improved efficiency, communication, and security. It streamlines tasks, promotes collaboration, optimises room allocation, and ensures a secure environment. By implementing future enhancements like mobile applications, smart technology integration, data analytics, AI-powered chatbots, and online learning platform integration, the system aims to enhance convenience and efficiency while prioritising security and sustainability. Overall, a hostel management system offers significant benefits for administrators, staff, and students, providing an enhanced experience and paving the way for future advancements in the field.

Chapter 2: LITERATURE REVIEW

2.1 EXISTING SYSTEM

The existing system of a hostel management system typically involves a combination of manual processes and some level of automation. Here's an overview of the components typically found in the existing system:

- 1. Manual Record Keeping:** The existing system often relies on manual record keeping, where hostel administrators maintain physical registers or files to store information about students, room allocations ,fee payments, attendance records, and other relevant data.
- 2. Paper-Based Processes:** Various hostel-related processes,such as room allocation,leave requests, complaint submissions, and fee collection, are usually carried out using paper-based forms and documents.
- 3. Face-to-Face Communication:** Communication between students, supervisors, and administrators often occurs through face-to-face interactions. For example,students may visit the hostel office to submit requests, discuss issues, or inquire about hostel-related matters. This can lead to delays and inefficiencies in communication.
- 4. Manual Room Allocation:** Room allocation in the existing system is typically done manually,where supervisors assign rooms to students based on their preferences,availability, and other criteria.
- 5. Limited Reporting and Analytics:** The existing system may lack comprehensive reporting and analytics capabilities. Generating reports on occupancy rates,financials, attendance records,or other important metrics often requires manual compilation and analysis of data from various sources.
- 6. Lack of Self-Service Portals:** Students may have limited or no access to self-service portals to manage their personal information, make leave requests, view mess menus, or track their fee payments. This can result in a dependency on hostel staff for routine tasks and queries.

2.2 DRAWBACKS OF THE PREVIOUS SYSTEM

The previous hostel management system had several drawbacks due to its heavy reliance on manual processes and paper-based documentation. Some of the drawbacks include:

1. **Inefficiencies:** Manual processes in the previous system led to inefficiencies and time-consuming tasks. For example, manual registration, room allocation, and fee collection required significant manual effort and increased the processing time for these tasks.
2. **Errors and inaccuracies:** Manual data entry and record-keeping increased the chances of errors and inaccuracies in student information, room allocations, attendance records, fee payments. This could lead to discrepancies and difficulties in resolving issues.
3. **Difficulty in Data Management:** Managing and organising physical records and documents posed challenges in terms of storage, retrieval, and data management. It was time-consuming and prone to misplacement or loss of important information.
4. **Limited Accessibility and Convenience:** The previous system lacked convenient access to information and services for both students and administrators. Students had to visit the hostel office for various tasks, such as submitting forms, making inquiries, or requesting services.
5. **Security Risks:** Physical records and documents posed security risks as they were susceptible to damage, loss, or unauthorised access. There was a lack of robust measures to protect sensitive student information.
6. **Limited Reporting and Analytics:** Generating reports and analysing data in the previous system was a labour-intensive task. It required manual compilation and analysis of data from various sources, making it time-consuming and prone to errors.
7. **Lack Of Real-time Updates:** With manual processes, it was difficult to have real-time updates and access to the latest information. For example,

administrators may not have had instant visibility into room availability or fee payment status, leading to delays and potential confusion.

2.3 TOOLS AND TECHNOLOGIES USED IN THE PREVIOUS METHOD

In the previous hostel management system, the tools and technologies used were primarily traditional and offline in nature. Here are some of the common tools and technologies employed:

- 1. Paper-based Documentation:** The system heavily relied on physical documents and records for tasks such as student registration, room allocation, fee collection, attendance tracking, and complaint submissions. Paper forms, registers, files were used to store and manage this information.
- 2. Manual Registers and Ledgers:** Hostel administrators maintained manual registers and ledgers to record and track various data points, including student details, room assignments, fee payments, attendance records, and complaint logs.
- 3. Spreadsheets and MS Excel:** Some administrators may have used spreadsheet software like Microsoft Excel to manage certain aspects of the hostel management system. This could include creating basic reports, tracking fee payments, or performing simple calculations.
- 4. Physical notice Boards:** notice boards were commonly used to display important announcements, notifications, schedules, and other hostel-related information.
- 5. Manual Communication Channels:** Face-to-face communication was the primary mode of interaction between students, supervisors, and administrators. Students had to personally visit the hostel office to submit forms, make inquiries, or discuss any issues or requests.

- 6. Cash Transactions:** Fee collection was primarily done in cash. Students were required to visit the hostel office and make cash payments for their hostel fees.
- 7. Traditional Office Equipment:** Hostel offices were equipped with traditional office tools such as typewriters, calculators, file cabinets, and physical stationary like pens, pacers, and folders to carry out administrative tasks.

CHAPTER 3: THE PROPOSAL

3.1 PROPOSED SYSTEM

Our Project is divided into different working modules. Following are the description and functionalities of each and every modules:

3.1.1 Student Module

- i. The student module is designed to provide functionalities and services specific to students.
- ii. Students need to register in the student portal to access the system.
- iii. The module allows students to view their personal information, including details like name, contact information, and room allocation.
- iv. Students can submit leave requests through the module, specifying the reason, duration, and any supporting documents.
- v. They can also access features related to mess services, such as viewing mess menus, monthly mess bill and remaining amount..
- vi. Additionally, the student module may provide options for submitting complaints or maintenance requests, allowing students to report issues they encounter in the hostel or room.

3.1.2 Supervisors Module

- i. The supervisors module is responsible for authenticating and verifying student data.
- ii. Upon student registration, the supervisors module performs necessary checks to validate and authenticate the information provided by the student.
- iii. This module ensures that the data entered during the registration process is accurate and meets the required criteria.

- iv. Supervisors may have access to student records, allowing them to review and verify personal details, attendance records, mess bills and room allocations.
- v. Verification by the supervisors module enables students to use the facilities provided by the student module with authenticated and reliable data.

3.1.3 Admin Module

- i. The admin module serves as the central control and management hub for the entire hostel management system.
- ii. It ensures the proper handling and maintenance of the system and work ethics throughout all modules.
- iii. The admin portal typically requires authorised access for administrative staff or designated personnel.
- iv. It allows administrators to configure system settings, manage user roles and permissions, and oversee the overall functioning of the system.
- v. The admin module facilitates tasks like registering new students, assigning user roles, updating fee structures, and managing system-wide configurations.
- vi. It may also provide reporting and analytics features, allowing supervisors to generate various reports on hostel occupancy, financials, complaints, and other important metrics.

3.1.4 System Features

- i. User Authentication will provide secure login functionality for students, supervisors and administrators.
- ii. Each user role will have a dedicated dashboard displaying relevant information and quick access to key functionalities.
- iii. System will facilitate communication between users through messaging and notification features.

- iv. Our system will store and manage data related to students. Rooms fees. Attendance in a database.
- v. The system will implement appropriate security measures, including data encryption, access control, and regular backups.

Overall, the student module provides specific functionalities for students, such as personal information management, leave requests, mess services and compliant submissions. The supervisors module ensures the authentication and verification of student data, enabling students to use the system with validated information. The admin module governs the system's operation ensuring proper handling, work ethics, and overall management of all modules within the hostel management system.

3.2 PROPOSED SYSTEM REQUIREMENT

Following are the software requirements for Hostel Management System:

- 1. Operating System:** system should be compatible with the desired operating system such as Windows.
- 2. Web Development Framework:** Project system is web-based, so web development framework ReactJS is used to build the user interface.
- 3. Backend Framework:** A backend framework Node.js is used to handle server-side logic and data processing.
- 4. Database Management System:** In our project MongoDB is used to store and manage data related to students profile and supervisors login ID password.
- 5. Server Environment:** In future to host and deploy our project requires a server environment like Apache or Nginx.
- 6. Authentication and Authorization:** In our project to secure user access for authentication and authorization mechanisms we used JWT(JSON Web Tokens).

- 7. Development Tools and IDEs:** Visual Studio Code integrated development environment (IDE) is used for coding and debugging.
- 8. Version Control System:** In our project development Git is utilised for collaboration, code management.

3.3 PROJECT PLANNING

The project planning for a Hostel Management System using Javascript framework for client side ReactJS and server side Node.js:

- 1. Define Project Objectives:** Define the objectives and goals of the Hostel Management System, such as automating hostel operations and enhancing user experience.
- 2. Identify Stakeholders:** Identify the key stakeholders involved, including hostel administrators, supervisors and students.
- 3. Scope Definition:** Define the scope of the project by identifying the functionalities and features to be included, such as student registration and login, room allocation, fee management, messaging and reporting.
- 4. Creation of Project Team:** Assign roles and responsibilities to team members, including front-end developers, back-end developers.
- 5. Technology Selection:** Choose ReactJS as the client-side framework for building the user interface and Node.js as the server-side runtime environment. Advantages of ReactJS, such as component-based architecture, virtual DOM, and efficient rendering, and Node.js for its scalability and asynchronous programming capabilities.
- 6. Deployment and Hosting:** Plan for the deployment of the system, considering the hosting requirements for the ReactJS front-end and Node.js back-end. Choose the suitable hosting provider and configure the necessary infrastructure.
- 7. Project Monitoring and Control:** Establish mechanisms for monitoring and controlling the project's progress.

By a comprehensive project planning process, it can ensure the successful development and implementation of the Hostel Management System using ReactJS and Node.js. The project timeline, budget, and resources required would depend on the scope and complexity of the system and should be carefully planned and managed.

Chapter 4: DESIGN

4.1 PROGRAM STRUCTURE

1. **System Architecture:** Front-end will be developed using ReactJs. It will handle the presentation layer, user interactions, and data rendering. It will be developed using Node.js. It will handle the business logic, data processing, and integration with the database. The system will utilise a database management system MongoDB to store and manage data related to students registration, login, rooms, fees and other relevant information.
2. **Component-Based Approach:** The system will follow a component-based architecture where each logical unit or functionality will be encapsulated into reusable components. Common components like navigation menus, form elements, and notifications will be developed for consistent user experience.
3. **User Interfaces:** The system will have separate user interfaces for students, supervisors, and administrators, each to their specific needs and roles. UI will be designed using modern framework which is best compatible for ReactJs i.e. Material UI. Interfaces will include forms for registration, login, data input.
4. **Routing and Navigation:** React router will be used for handling client-side routing, allowing users to navigate between different pages and views within the application. Each Module(student, supervisor, admin) will have its own set of routes and access permissions to ensure appropriate access control.
5. **API and Server-side Development:** Back-end will expose RESTful APIs using Node.js and Express.js. It will handle requests from the front-end, perform necessary logic operations, and communicate with the database for data retrieval.
6. **Data Management:** Our Project uses an ORM (Object-Relational Mapping) library Mongoose for MongoDB to interact with the database.
7. **Authentication and Authorisation:** In our Project for User authentication JWT(JSON Web Tokens) is implemented. Authorisation rules will be defined

to control user access and permissions based on their roles (student, supervisor, admin).

8. **Error Handling and Validation:** Proper error handling mechanisms will be implemented to catch exceptions at both the front-end and back-end. Input validation will be performed on the client-side and server-side to ensure data integrity and prevent security vulnerabilities.
9. **Deployment and Maintenance:** System will be deployed on a suitable hosting platform like AWS, Heroku, or Azure, ensuring scalability, performance, and availability. Regular maintenance and updates will be performed to address bugs, security vulnerabilities, and feature enhancements.

CHAPTER 5: IMPLEMENTATION DETAILS & RESULTS

5.1 TOOLS AND TECHNIQUES USED

- i. **ReactJS:** ReactJS, a popular JavaScript library, was chosen for the client-side development of the Hostel Management Project. ReactJS follows a component-based architecture, allowing the creation of reusable UI components. Its efficient virtual DOM enables fast updates to the user interface, resulting in a responsive and dynamic application. ReactJS's declarative syntax and one-way data flow make it easier to understand and maintain the codebase. It also has a vast ecosystem of libraries and community support, enabling rapid development and integration with other tools and frameworks.
- ii. **Material-UI(UI Library):** Material-UI, a React-based UI library, was utilised to enhance the visual aesthetics and improve the user experience of the project. Material-UI provides a comprehensive set of pre-built and customizable components following the Material Design principles. These components, such as icons, boxes, calendars, buttons, forms, and navigation elements, helped create a consistent and visually appealing user interface for the hostel management system. Material-UI's theming capabilities allowed easy customization to match the institute's branding and design guidelines.
- iii. **Node.js:** Node.js is an open-source JavaScript runtime environment used for server-side development in the Hostel Management Project. Node.js enables running JavaScript on the server-side, employing a non-blocking, event-driven architecture. Its single-threaded, event-driven model ensures high scalability and performance, allowing the server to handle a large number of concurrent connections. Node.js's vast package ecosystem (npm) provides a wide range of libraries and modules that expedited the development process.

- iv. **Express:** Express is a minimal and flexible web application framework for Node.js, used in conjunction with Node.js for server-side development in the project. Express simplifies the creation of RESTful APIs and server-side logic by providing a lightweight and intuitive framework. Its middleware architecture allows for easy handling of HTTP requests and responses, including routing, request parsing, and error handling. Express's simplicity and extensibility made it an ideal choice for building the backend of the hostel management system.
- v. **Cors:** Cors (Cross-Origin Resource Sharing) is a middleware used in the project to handle cross-origin HTTP requests. It ensures secure communication between the frontend and backend servers by defining and enforcing access control policies. Cors enables the Hostel Management Project to handle requests from different domains securely, allowing the client-side application to interact with the server-side APIs. It provides flexibility in specifying the allowed origins, methods, and headers, preventing unauthorised access and protecting sensitive data.
- vi. **JSON Web Token (JWT):** JSON Web Token (JWT) is a compact and URL-safe means of representing claims between two parties. In the project, JWT was used for authentication and authorization purposes. After successful login, JWTs were generated and sent to the client, which could be included in subsequent requests for authentication. JWTs securely carried user information, enabling the server to verify the authenticity and permissions of each request. JWT's stateless nature eliminated the need for server-side session management, providing scalability and facilitating the implementation of a distributed system.
- vii. **MongoDB:** MongoDB, a NoSQL document database, was chosen as the database management system for the Hostel Management Project.

MongoDB's flexible document model allows for easy storage and retrieval of hostel-related data. It provides a scalable and high-performance solution for handling large volumes of data, making it suitable for a hostel management system with potentially thousands of students. MongoDB's query language and aggregation framework enable efficient data manipulation, querying, and reporting, enhancing the overall functionality and performance of the system.

- viii. **Mongoose:** Mongoose is an object data modelling (ODM) library for MongoDB and was utilised in the project. It provides a straightforward way to interact with MongoDB through a schema-based approach, allowing developers to define data models, perform validations, and create relationships between data entities. Mongoose simplifies the interaction with MongoDB by providing an additional layer of abstraction, making it easier to handle complex data operations and enforce data integrity in the hostel management system.

These tools and technologies, including ReactJS, Material-UI, Node.js, Express, MongoDB, Mongoose, Cors, and JSON Web Token (JWT), were instrumental in creating a modern and efficient architecture for the Hostel Management Project. The combination of these technologies provided a robust, scalable, and visually appealing solution, meeting the specific requirements of the institute's hostel management operations.

5.2 INSTALLATION AND SETUP

ReactJS:

- Install Node.js: Visit the official Node.js website (<https://nodejs.org>) and download the latest stable version compatible with your operating system. Follow the installation instructions provided.

- Create a React App: Open a terminal or command prompt and run the following command to create a new React app:

```
npx create-react-app hostel-management
```

- Navigate to the project directory:

```
cd hostel-management
```

- Start the development server:

```
npm start
```

- React app should be running at <http://localhost:3000>.

Material-UI:

- Install Material-UI: In the project directory, run the following command to install Material-UI and its dependencies:

```
npm install @mui/material @emotion/react @emotion/styled
```

- Import and Use Components: You can now import Material-UI components in your React components and start using them to enhance the user interface.

Node.js:

- Install Node.js: Visit the official Node.js website (<https://nodejs.org>) and download the latest stable version compatible with your operating system. Follow the installation instructions provided. Verify the installation: Open a terminal or command prompt and run the following command to verify that Node.js is installed

successfully:

```
node -v  
npm -v
```

Express:

- Initialise a Node.js project: Create a new directory for your project and navigate into it. Open a terminal or command prompt and run the following command:

```
npm init
```

- Install Express: Run the following command to install Express and its dependencies:

```
npm install express
```

- Create an Express server: Create a new file (e.g., server.js) and import Express. Define your server routes and logic in this file.

Cors:

- Install Cors: In the project directory, run the following command to install Cors:

```
npm install cors
```

- Enable Cors in Express: In your Express server file, import Cors and add it as middleware to enable cross-origin resource sharing.

JSON Web Token (JWT):

- Install JWT: In the project directory, run the following command to install the jsonwebtoken package:

```
npm install jsonwebtoken
```

- Generate and Verify JWTs: Implement the logic to generate and verify JWTs in your server-side code, ensuring secure authentication and authorization.

MongoDB:

- Install MongoDB: Visit the official MongoDB website (<https://www.mongodb.com>) and download the latest stable version compatible with your operating system. Follow the installation instructions provided.
- Start MongoDB: Run the MongoDB server using the command appropriate for your operating system.
- Connect to MongoDB: In your Express server file, import the mongoose package and establish a connection to your MongoDB database using the connection string.

Mongoose:

- Install Mongoose: In the project directory, run the following command to install Mongoose:

```
npm install mongoose
```

- Define Models and Schemas: Create Mongoose models and schemas to define the structure and behaviour of our hostel-related data.

5.3 DISCUSSION ON IMPLEMENTATION

5.3.1 Home Page

- The Home Page of our Hostel Management Project is designed to provide a comprehensive overview of the system's features and

- functionalities while incorporating various modules, MUI elements, and a Carousel to create an engaging and user-friendly interface.
- ii. At the top of the Home Page, the university logo and name is displayed and then a Navbar or Menu Bar is prominently displayed, allowing users to easily navigate to different sections of the website, including options like Home, About Us, and Contact. A login/signup button is there at the right side of navbar. This ensures convenient access to important information and functionalities.
 - iii. The HomeImage module occupies a prominent position and features an eye-catching image that represents the hostel or the institute. This visually appealing element serves as a focal point, capturing the attention of visitors and setting the overall tone and theme of the website.
 - iv. The About Us module provides detailed information about the hostel and the institute, including its history, vision, mission, and facilities. This section aims to give visitors a clear understanding of the purpose and values of the hostel, creating a sense of trust and familiarity.
 - v. The Contact module facilitates easy communication between visitors and the hostel management by providing essential contact information such as phone numbers, email addresses, and physical addresses. Visitors can use this section to reach out for inquiries, feedback, or any other communication needs.
 - vi. The Home Page also incorporates various MUI (Material-UI) elements to enhance its visual appeal and functionality. Typography offers consistent and visually appealing text styles, while the Box component serves as a container for organising and positioning other components, allowing for flexible layout arrangements. Additionally, the Icon component provides a wide range of icons that can be used to represent actions, functionalities, or visual cues, enhancing the user experience. An essential feature on the Home Page is the Login

Button, which provides access to personalised accounts for Students, Supervisors, and Admin. Users can log in to the website and access their respective dashboards or perform specific actions based on their roles and permissions. The Login Button ensures secure and convenient access to the hostel management system, enabling users to manage their data and access relevant information. Moreover, the Home Page incorporates a Carousel module, adding a dynamic and visually appealing element. The Carousel displays a series of images or content in a rotating manner, allowing for customization of transition speed, animation effects, and navigation controls. This module showcases different aspects of the hostel or highlights important announcements or events, capturing the attention of visitors and encouraging them to explore further.

- vii. By combining these modules, MUI elements, and the Carousel, the Home Page of the Project offers a visually appealing and intuitive interface. It provides easy navigation options, essential information about the hostel, a convenient login functionality, and an engaging Carousel that showcases various aspects of the hostel. This combination creates a positive user experience, encouraging visitors to explore more about the hostel and its management system.



Fig 5.3.1 Home Page

5.3.2 Signup

- i. There are three main modules: Student module, Supervisor module, and the Admin module. These modules cater to the different roles and functionalities required by the users of the hostel management system.
- ii. To start using the system, students need to register themselves. This can be done by accessing the login/signup page. Upon clicking the relevant button, a popup page will appear where new users can provide their necessary details for registration, such as their name, contact information, and academic information. The registration process ensures that only authorised students can access the system.
- iii. Once registered, students can then log in to the hostel management system using their login credentials, such as a roll number and password. Logging in allows students to access the various features and facilities provided by the system.

- iv. The registration and login/signup process ensures that the system can authenticate and verify the identity of the users, providing a secure and personalised experience. By logging in, students can benefit from the functionalities offered by the hostel management system, such as room allocation, fee management, leave requests, and access to important notifications and information related to the hostel.
- v. Overall, the modules in the project, along with the registration and login/signup processes, enable students to conveniently access and utilise the services provided by the hostel management system, enhancing their overall hostel experience and simplifying administrative tasks.

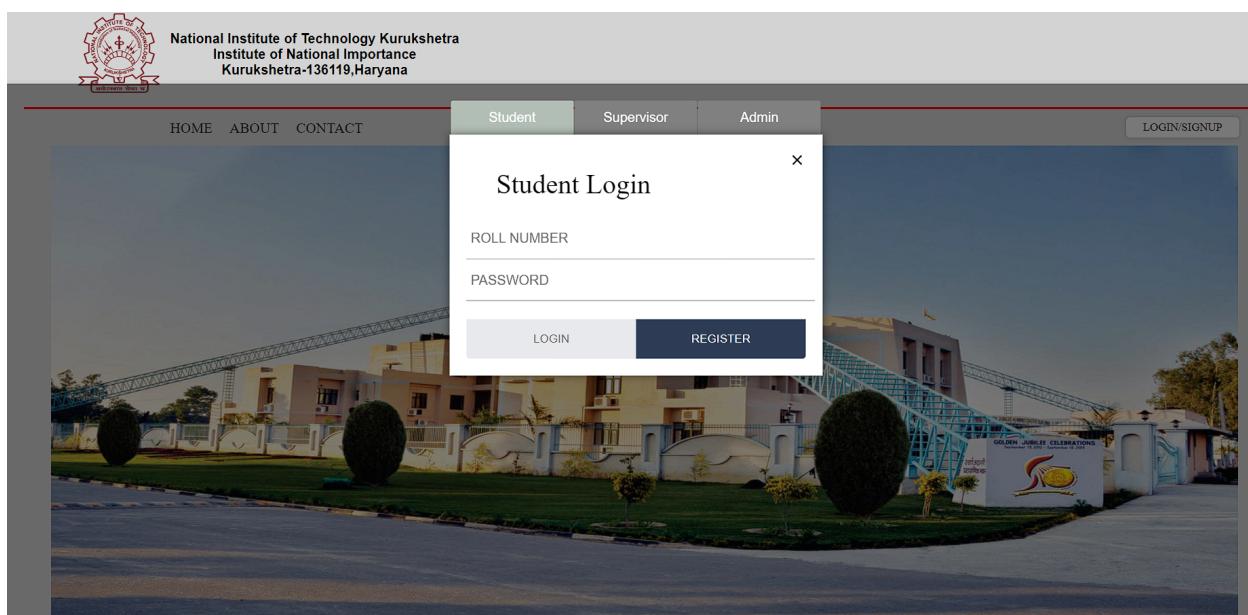


Fig 5.3.2 Student Registration

5.3.3 Students Registration

- i. In the popup page for registration, students are provided with a form where they can enter their basic details to complete the first phase of registration for their successful signup.
- ii. Once students have filled in all the required fields in the registration form, they can submit the form to complete the registration process.
- iii. This account will allow students to login and access the various features and functionalities provided by the hostel management system.
- iv. After students complete the registration process, their data is securely saved in the database. In this model for the hostel management system, MongoDB is utilised as the database management system to store and manage the data. MongoDB is a NoSQL database that offers flexibility, scalability, and ease of integration with both the client-side (ReactJS) and server-side (Node.js) components of the system.
- v. When a student submits the registration form, the entered data is collected and processed by the server-side code written in Node.js. This code interacts with the MongoDB database to store the student's information in a structured manner. Each registered student will have a dedicated entry or document in the database.

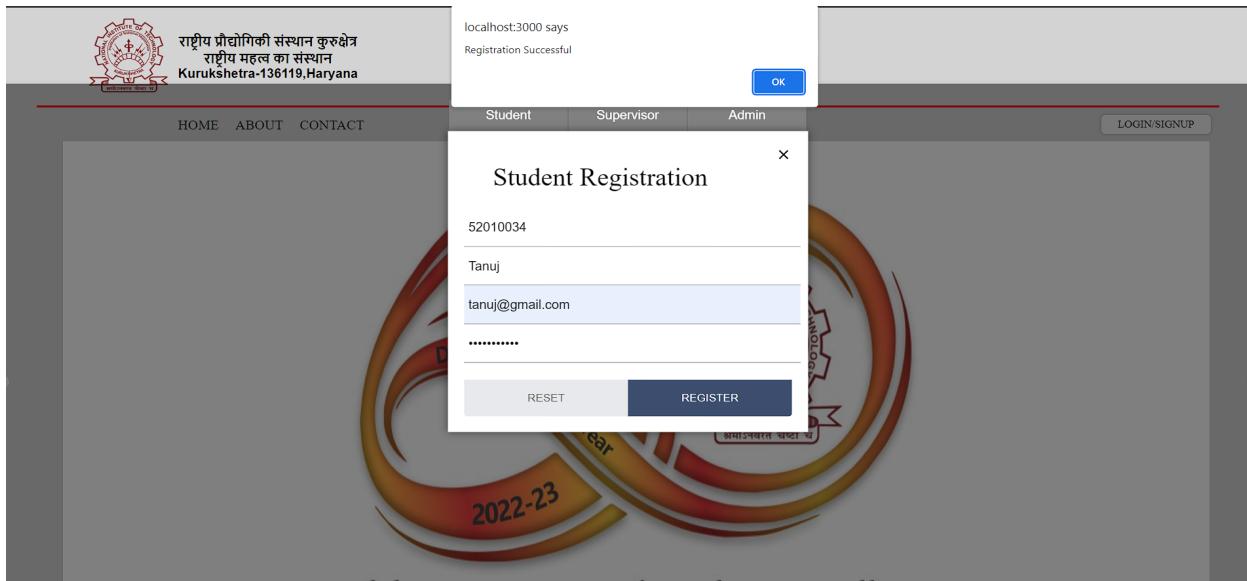


Fig 5.3.3 i) Student Successful Registration

JSON Web token is used for authentication and it is a popular method in web applications. It allows the server to securely verify the identity of a user and grant them access to protected resources.

By using JWT, the authentication process becomes stateless, as the server doesn't need to store session information. The client holds the token and includes it with each request, eliminating the need for server-side session management. JWTs are self-contained and carry all the necessary information to verify the user's identity. They also provide a secure way to transmit user-related data without exposing sensitive information.

It's important to note the JWTs should be transmitted over a secure channel(e.g. HTTPS) to protect them from interception and tampering. Additionally, the secret key used for signing and verifying JWTs should be kept confidential to ensure the token's integrity.

hostleRegister.UserInfo

The screenshot shows the MongoDB Compass interface with the database 'hostleRegister' selected. The 'UserInfo' collection is displayed under the 'Documents' tab. There are two documents listed:

```
_id: ObjectId('6474579fc6315863b48b34f1')
rollNumber: "52010010"
Name: "Manish"
email: "manishkt52@gmail.com"
password: "$2a$10$Bxj9S1hvGu2N7wwJW3rx/uWfo2fhwL1LQmxhQuagAoMCya8VkmxRu"
__v: 0

_id: ObjectId('647f129042aca9ed03c2a5ab')
rollNumber: "52010034"
Name: "Tanuj"
email: "tanuj@gmail.com"
password: "$2a$10$UOIfCBubUGeDTbLV6Eb3XORuOkR34wDFnwogqwrjyV50mq9e3Xx1y"
__v: 0
```

Fig 5.3.3 ii) Student Database

5.3.4 Student login

- i. After successfully registering with the hostel management system, students can proceed to log in using their valid credentials. Students are presented with a login form where they can enter their login credentials.
- ii. When students submit the login form, the system initiates the authentication process. The entered credentials are sent to the server-side for verification.
- iii. The server validates the entered credentials by checking if they match the records stored during the registration process.
- iv. If the credentials are valid and match the stored data, the authentication is successful. Otherwise, if the credentials are incorrect

or don't match, the authentication fails, and an error message is displayed.

- v. The system generates a JSON Web Token (JWT) to represent the student's authenticated session.
- vi. This JWT contains relevant information. For Example: in our project student roll number and password and it is securely sent back to the client-side.

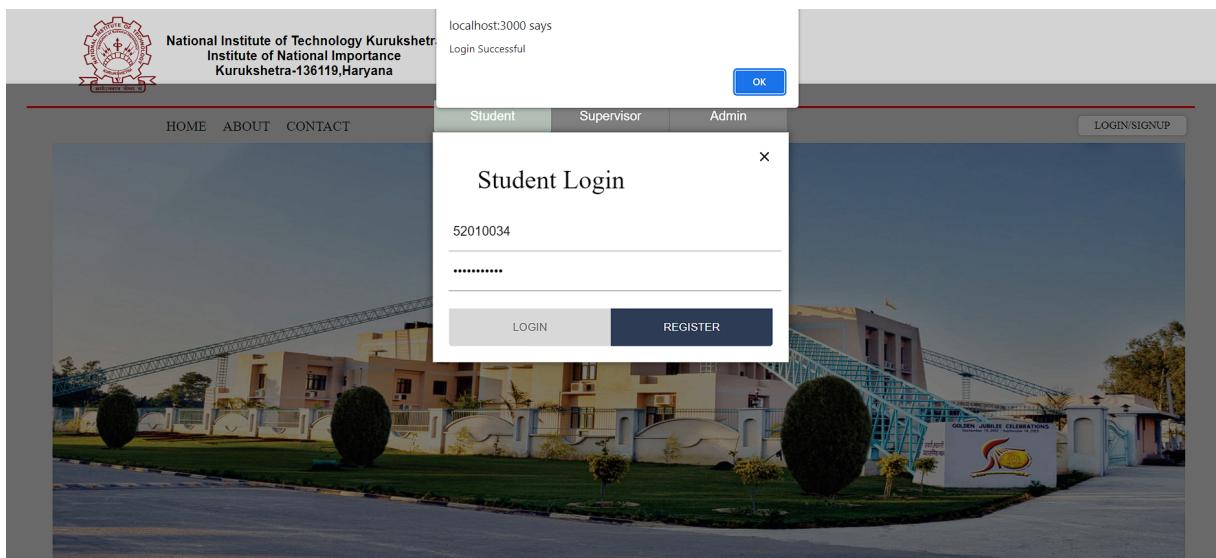


Fig 5.3.4 Student Successful Login

5.3.5 Student Dashboard

- i. After the successful login. The system recognizes the student's credentials as valid, and as a result, the student gains access to

their personalised dashboard. The dashboard serves as a central hub or interface where the Student can access various features, tools, and information related to their academic or educational journey.

- ii. The opening of the student dashboard indicates that the system has processed the login request and has loaded the relevant data and resources specific to the student's profile.

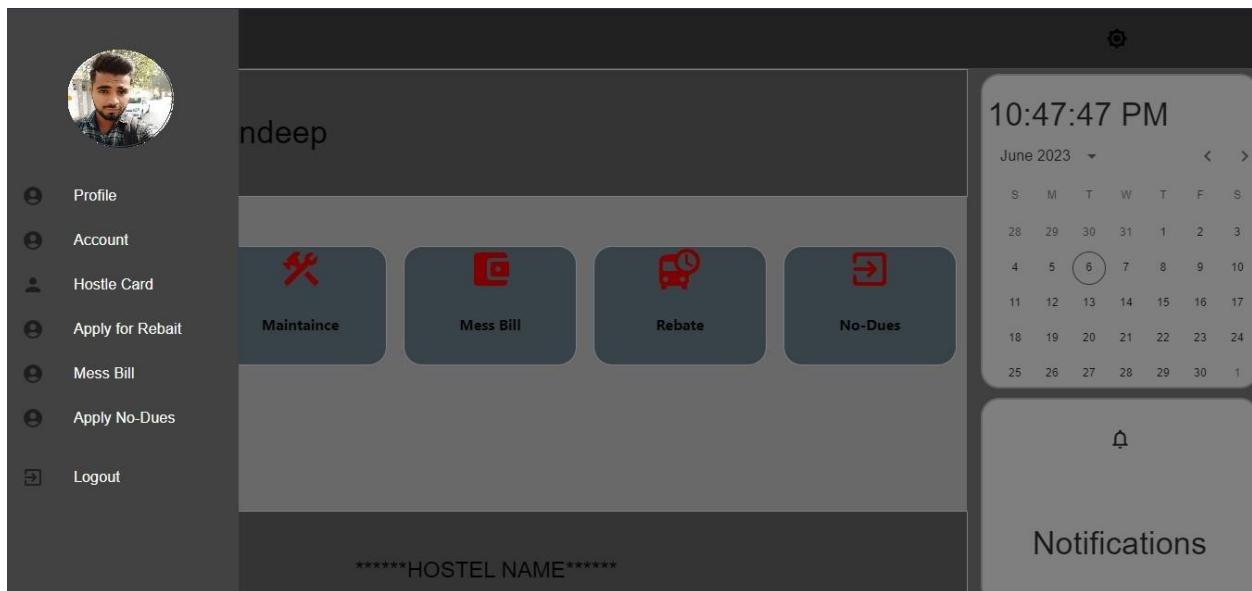


Fig 5.3.5 Student Dashboard

5.3.6 Profile:

- i. The Profile feature in the dashboard of the hostel management system provides students with a dedicated section to view and manage their personal information. It offers a comprehensive overview of the student's details, allowing them to verify the accuracy and update any incorrect or outdated information.
- ii. Within the Profile section, students can access various fields such as their name, roll number, contact information, branch, and other relevant data. They

- have the flexibility to make necessary edits to their profile, ensuring that the information remains up to date.
- iii. Furthermore, the Profile feature enables students to print and download their information. This functionality proves beneficial when students need to provide their details for administrative purposes, such as submitting required documents or fulfilling hostel requirements. The ability to generate a printable and downloadable version of their profile simplifies the process and enhances convenience.
 - iv. In addition to student access, supervisors also have the capability to review and verify student details within the Profile section. This helps supervisors in cross-checking the accuracy of information provided by students and ensuring the data aligns with the institute's records.

Overall, the Profile feature within the hostel management system's dashboard empowers students to manage their personal information efficiently. It enables them to update and maintain accurate records while providing convenient options for printing, downloading, and submitting their information as per hostel requirements. The feature also facilitates the verification process for supervisors, ensuring data integrity and reliability.

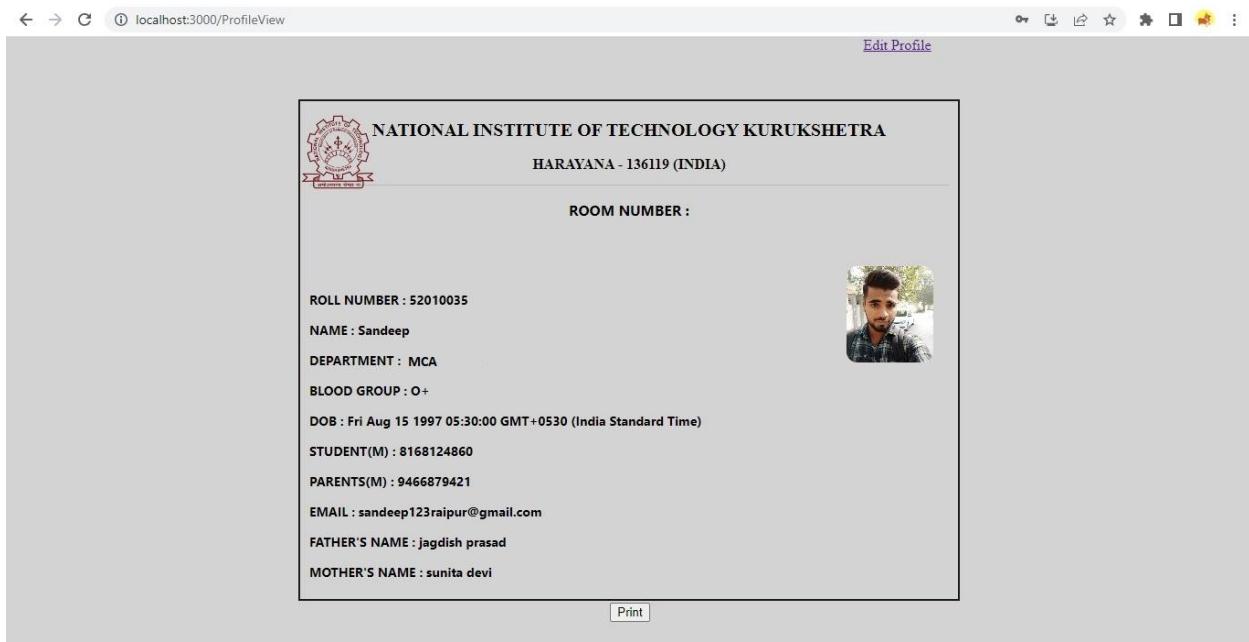


Fig 5.3.6 Student Profile

5.3.7 Hostel Card:

- i. The Hostel Card is one of the features of the dashboard in the hostel management system. It is a dynamic and informative card displayed in the side slider, providing essential details about the student's hostel accommodation. The card contains important information such as the student's name, roll number, contact details, branch, address, and other relevant data.
- ii. The Hostel Card is designed to present this information in a visually appealing and organised manner, making it easy for students to access and review. The card layout ensures that all the necessary details are displayed clearly, allowing users to quickly locate and reference the required information.

- iii. Additionally, the Hostel Card offers a convenient functionality that allows users to download and print the card as a PDF file only at once. This feature enables students to have a physical copy of their hostel details for reference or documentation purposes. It provides a seamless way to generate a printable version of the card, enhancing convenience and accessibility for the students.

Overall, the Hostel Card in the dashboard of the hostel management system serves as a comprehensive and user-friendly tool for accessing and managing student information. The option to download and print as a PDF add

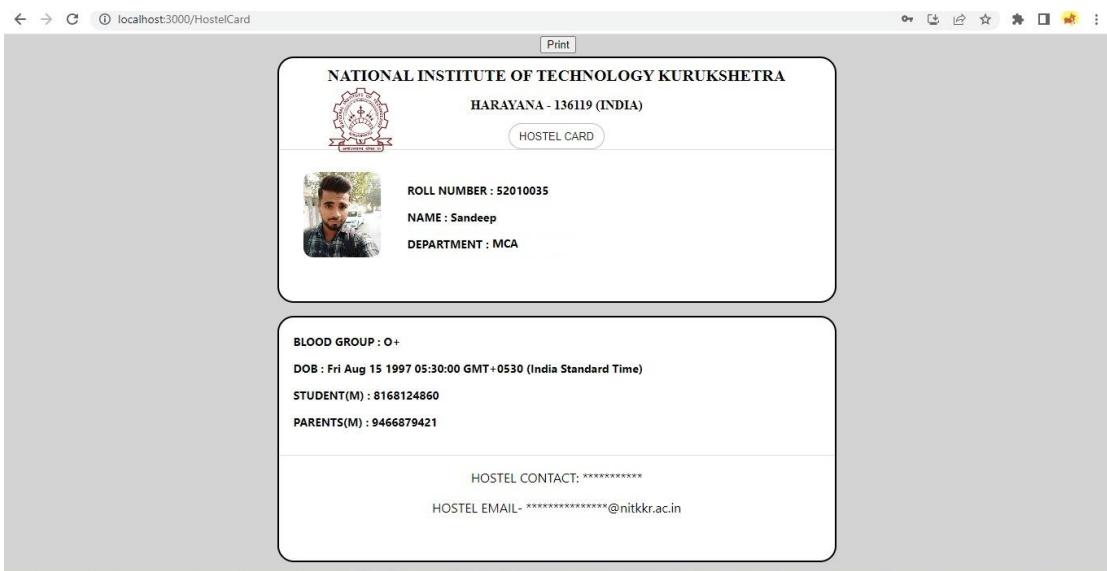


Fig 5.3.7 Student Hostel Card

5.3.8 Rebate Apply

- i. The Rebate feature in the hostel management system facilitates the process of applying for holiday leaves from the hostel, specifically concerning the mess arrangements. This feature allows students to request temporary leave from the hostel mess during their absence.

- ii. To apply for a mess rebate, students can conveniently access the Rebate section within the dashboard. They are required to provide detailed information, including the start and return dates of the leave, the reason for the leave, the destination or place they will be during the leave, and any additional notes or comments they may have.
 - iii. Once the leave application is submitted, it undergoes a thorough review and approval process by the supervisor. The supervisor carefully considers various factors such as the current mess occupancy, availability of resources, and the student's specific dietary needs to ensure that the necessary arrangements are made. Upon approval, the student is exempted from mess charges for the duration of their leave.
- iv. The Rebate feature not only streamlines the process of applying for a mess rebate but also ensures effective communication and coordination between students and supervisors. It allows for proper planning and resource allocation in the hostel mess, minimising wastage and optimising the overall mess management system.
- v. By incorporating the Rebate feature into the hostel management system, students can conveniently manage their mess arrangements during temporary absences, while supervisors can efficiently handle the mess operations and maintain a fair and transparent process for rebate approval.

Rebate Form

From Date:

□

Return Date:

□

Reason:

Contact:

Submit

Fig 5.3.8 Student Rebate Form

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

A hostel management system improves efficiency, communication, and security in hostel operations. It streamlines tasks like registration, room allocation, fee management, and enhances collaboration between administrators, staff, and students. The system optimises room allocation, facilitates fee collection and tracking, and ensures a secure environment through access control and monitoring. Overall, it enhances the hostel experience for all parties involved.

6.2 FUTURE SCOPE

The future scope of a hostel management system involves developing a mobile application, integrating with smart technologies, leveraging data analytics, implementing AI-powered chatbots, integrating with online learning platforms, enhancing security measures, and promoting sustainability initiatives. These advancements aim to improve convenience, efficiency, and the overall experience for hostel stakeholders.

6.3 CONTRIBUTION

Creation of schema according to the users requirement for the fulfilment for CRUD operation and API generation .

Helps to generate comprehensive reports.

REFERENCES

1. <https://www.ijraset.com/research-paper/hostel-management-system-hms>
2. <http://www.ifnoss.com/edu/college-university-software/hostel-management-system.aspx#>
3. <https://www.vidyalayaschoolsoftware.com/features/school-hostel-management-system>
4. <https://mui.com/>
5. <https://www.mongodb.com/>
6. <https://nitkkr.ac.in/wp-content/uploads/2023/01/Girls-Hostels-F.pdf>
7. <https://nitkkr.ac.in/wp-content/uploads/2023/01/All-Hostels-f.pdf>
8. <https://nitkkr.ac.in/wp-content/uploads/2023/01/Hostel-rules.pdf>