### PROJECT Synopsis

**On**

# HostelSync

Submitted in partial fulfillment of the requirement for the Course BEE (22CS026) of

**COMPUTER SCIENCE AND ENGINEERING**

**B.E. Batch-2022 in**

**Jan -2025**



**Under the Guidance of** **Submitted By**

**Mr. Reshab Kumar Akashdeep Roll. No.**  2210991215

**Akshit**  **Roll. No.** 2210991234

**Ananya Roll. No.**  2210991258

**Akriti**  **Roll. No.** 2210991220

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# CHITKARA UNIVERSITY

**PUNJAB**

|  |  |  |
| --- | --- | --- |
| Index | | |
| S.No. | Topic | Page No. |
| 1. | Title of Project | 3 |
| 2. | Problem Statement | 4 |
| 3. | Objective & Key Learning’s | 5 |
| 4. | Flow of Project | 6 |
| 5. | Execution of the Project | 7 |
| 6. | Advantages | 8 |
| 7. | References | 9 |

Title of project

HOSTELSYNC

A Comprehensive Hostel Management System

**Problem Statement**

* **Efficient Room Allocation:** The current manual process of assigning rooms to hostel residents is time-consuming and prone to errors. The hostel management system should streamline room allocation, ensuring that available rooms are assigned fairly and quickly.
* **Centralized Student Information Management:** Managing student information, including personal details, room assignments, and payment records, is often fragmented across different systems. A centralized system is needed to store and manage all student-related data efficiently, reducing the risk of data loss or duplication.
* **Automated Fee Collection and Tracking:** The manual collection of hostel fees can lead to discrepancies and delays in payment tracking. The system should automate fee collection and provide real-time updates on payment status, ensuring transparency and accuracy in financial records.
* **Streamlining Complaint Resolution:** Due to inefficient communication channels, students often face delays in resolving maintenance or service-related complaints. The hostel management system should provide a streamlined process for lodging and tracking complaints, ensuring timely resolution and improving student satisfaction.

**Objective & Key Learning’s**

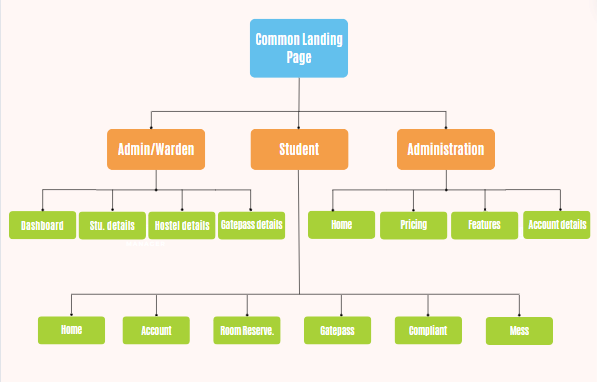
**Project Objective:**

1. **Enhance Operational Efficiency:** Develop a centralized system to streamline various hostel management processes, including room allocation, student information management, and fee collection, reducing manual effort and errors.
2. **Improve Student Experience:** Create a user-friendly interface that allows students to easily access their information, submit requests, and lodge complaints, ensuring quick responses and better overall satisfaction.
3. **Automate Administrative Tasks:** Implement automated systems for managing bookings, tracking payments, and generating reports, freeing up administrative staff to focus on more strategic tasks.
4. **Facilitate Data-Driven Decision Making:** Integrate tools for generating visual reports and analytics on key metrics like occupancy rates, revenue, and resource usage, enabling management to make informed decisions.
5. **Ensure Data Security and Integrity:** Build a secure system that protects sensitive student and financial data, ensuring compliance with data protection regulations and maintaining trust with users.

**Key Learnings:**

1. **Full-Stack Development:** Gain hands-on experience in building a full-stack application using technologies like React.js for the front end and possibly Node.js or another backend framework for server-side operations.
2. **Database Management:** Learn how to design, implement, and manage a database system that efficiently handles large volumes of data, ensuring quick retrieval and secure storage.
3. **User Interface and Experience Design:** Develop skills in designing and implementing a responsive, intuitive user interface that enhances the user experience for both students and administrators.
4. **API Integration:** Understand how to design and integrate RESTful APIs to enable smooth communication between the front end and backend of the application.
5. **Project Management and Collaboration:** Improve your ability to manage a software development project from conception to deployment, including collaboration with team members and handling version control using tools like Git.
6. **Security Best Practices:** Learn about best practices for securing web applications, including encryption, authentication, and protecting against common vulnerabilities

**Flow of the Project**



**Execution of the Project**

**Development Environment Setup**

* Local Development:
  + **Frontend:** Use React.js for building the user interface. Tools like create-react-app can help you get started quickly.
  + **Backend:** You can use Node.js with Express.js for creating RESTful APIs.
  + **Database:** Use a database like MySQL, MongoDB for storing data.
  + **Version Control:** Use Git for version control, and GitHub for collaboration and repository management.
* Platform-as-a-Service (PaaS):
  + **Netlify:** It offers easy deployment for static and serverless React apps.

**Project Management and Collaboration**

* Communication:
* WhatsApp or Google Meet for easy team communication and meetings.

**Testing and Quality Assurance**

* Manual Testing:
  + Perform manual testing using different browsers and devices to ensure your application is responsive and functional.

**Security Measures**

* Data Protection:
  + Ensure sensitive data like passwords are hashed before storage and use HTTPS for secure data transmission if deploying online.

**Advantages**

1. **Ease of Use:** Using simple tools makes the development process more straightforward, allowing you to focus on building your application without being overwhelmed by complex setups or configurations.
2. **Quick Learning Curve:** Since the tools are beginner-friendly, you and your team can quickly grasp how to use them, reducing the time needed for learning and allowing you to start working on the project sooner.
3. **Faster Development:** Simple tools often come with minimal setup, enabling you to develop and deploy your project more quickly, which is particularly useful if you’re working under tight deadlines.
4. **Cost-Effective:** Most simple tools are free or have free tiers, which makes them an economical choice for students or small teams with limited budgets.
5. **Focus on Core Features:** With fewer distractions from complex toolchains, you can concentrate more on developing the core functionalities of your project, ensuring that the essential features are well implemented.

**References**

**Websites and Online Resources:**

* **React.js Official Documentation (reactjs.org)** – For understanding React.js concepts, components, and hooks.
* <https://legacy.reactjs.org/docs/getting-started.html>
* **W3Schools (w3schools.com)** – For quick tutorials and references on web development basics.
* [React Tutorial (w3schools.com)](https://www.w3schools.com/react/)

**Articles and Tutorials:**

* **"Understanding REST APIs" on Medium** – For insights into building and consuming RESTful APIs.
* **"Building a Simple Node.js Backend" on Dev.to** – A step-by-step guide for setting up a basic backend with Node.js and Express.

**Tools and Libraries Documentation:**

* **Node.js Documentation (nodejs.org)** – For understanding the Node.js runtime and its capabilities.
* <https://nodejs.org/docs/latest/api/>
* <https://www.mongodb.com/docs/>
* <https://expressjs.com/>
* <https://nodemailer.com/>