**SAATHI**

**A PROJECT REPORT**

**for**

**Minor Project (KCA353)**

**Session (2024-25)**

**Submitted by :**

**Jatin Gupta**

**2300290140080**

**Ayushi Verma**

**2300290140047**

**Lalit Sharma**

**2300290140096**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Mr. Prashant Agrawal**

### Associate Professor



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**December 2024**

## **CERTIFICATE**

Certified that Jatin Gupta(2300290140080),Ayushi Verma(2300290140047) ,Lalit Sharma(2300290140096) have carried out the project work having “Saathi” (Minor-Project-KCA353) for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Mr. Prashant Agrawal** **Dr. Arun Kumar Tripathi**

**Associate Professor** **Dean & Professor**

**Department of Computer Applications** **Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad** **KIET Group of Institutions, Ghaziabad**

### **ABSTRACT**

Saathi is an innovative student-centric rental platform designed to address the financial challenges students face in accessing essential items. The platform leverages technology to promote resource-sharing and sustainability while offering cost-effective alternatives to purchasing. Saathi provides a streamlined experience with features such as social login, personalized dashboards, interactive maps for discovering nearby rental items, secure payment gateways, and a robust review system to foster trust within the community.

Unlike traditional rental systems, Saathi is tailored specifically for students, focusing on affordability, convenience, and environmental impact. By enabling peer-to-peer rentals, it not only helps students save money but also empowers them to earn additional income by renting out unused items. With an intuitive interface and a powerful admin panel for efficient management, Saathi ensures a seamless experience for both users and administrators.

Saathi aims to create a sustainable, resource-sharing community where students can tackle financial challenges, reduce waste, and contribute to an eco-friendly economy. It redefines how students access resources, promoting collaboration and sustainability in the student community.

**Keywords:** Student Rental Platform, Sustainability, Peer-to-Peer Sharing, Financial Affordability, Resource Optimization.

## **ACKNOWLEDGEMENTS**

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Prof.** **Mr. Prashant Agrawal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Dean & Professor, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Jatin Gupta

(2300290140080)

Ayushi Verma

(2300290140047)

Lalit Sharma

(2300290140096)

## **TABLE OF CONTENTS**

Certificate i

Abstract ii

Acknowledgements iii

Table of Contents iv

List of Tables vi

List of Figures vii

1. **Introduction**  1
   1. Project Description 1
   2. Literature Review 2
   3. Recommendation System 4
   4. Objective of the project 7
   5. Key features 8
   6. Scope of the project 8
   7. Software/Hardware Requirements 9
2. **Problem identification & feasibility study** 10
   1. Problem Identification 10
   2. Feasibility Study 12
      1. Technical Feasibility 12
      2. Operational Feasibility 12
      3. Economic Feasibility 13

**3** **Requirement Analysis** 14

* 1. Introduction 14
  2. Functional Requirements 15
     1. User Profile Module 15
     2. Recommendation Engine 15
     3. Communication Platform 15

3.3 Non-Functional Requirements 15

* + 1. Performance 16
    2. Security 16
    3. Scalability 16
    4. Usability 16

1. **Project Planning and Scheduling** 17
   1. Introduction 17
   2. Components of the project 18
   3. The 5 phases of a project 20
   4. Pert Chart 22
2. **Hardware and Software Specification** 24
   1. Hardware Specification 24
   2. Software Specification 25
3. **Design**  26
   1. FrontEnd Architecture 26
   2. BackEnd Architecture 27
   3. Data Flow Diagram 28
4. **ER DIAGRAM**  30
5. **Form Design**  32
   1. Home Page 32
   2. Contact Us 33
   3. Registration and Login 34
   4. Vendor Login 35
   5. Product 36
   6. Checkout 37

Summary 55

Conclusion 56

Future Scope and Further Enhancement of the Project 57

Bibliography 58

## **LIST OF TABLES**

Table No. Name of Table Page

7.1 ER Diagram 32

8.1 Home Page 35

* 1. Contact Us 36
  2. Registration and Login 34

8.4 Vendor Login 35

8.5 Product 36

8.6 Checkout 37

## LIST OF FIGURES

Figure No. Name of Figure Page No.

4.1 Introduction 18

4.2 Components of the project plan 20

4.3 5 Phases of project 21

4.4 Pert Chart 22

6.1 PHP 27

6.4 0 Level DFD 29

6.2 1 Level DFD 30

7.1 Entity-relationship Model 31

8.1 Home Page 38

8.2 Contact us 39

8.2 Login 40

8.3 Vendor Login 41

8.4 Product 42

8.4 Chcekout 43

**CHAPTER – 1**

### **INTRODUCTION**

**1.1 Project Description**

In a world where education has become increasingly competitive, students face not only academic pressures but also financial challenges. The rising costs of essential items like textbooks, electronics, and furniture often leave students in a difficult position, forcing them to make compromises that can impact their academic success. Recognizing this widespread issue, we have developed Saathi, a student-centric rental platform aimed at bridging the gap between students’ needs and their financial constraints.

**Saathi** is designed as a peer-to-peer rental ecosystem where students can rent items from their peers instead of purchasing them outright. It not only makes essential goods more affordable but also enables students to earn additional income by renting out items they no longer need. By promoting resource-sharing, Saathi addresses financial challenges while fostering a community-driven approach to sustainability.

What sets Saathi apart from traditional rental platforms is its tailored focus on students. The platform incorporates advanced features that cater specifically to their needs, including:

Social Login Integration: Quick and easy account creation using platforms like Google, Facebook, or Twitter, ensuring seamless access.

Personalized Dashboards: A centralized hub for users to manage their rentals, earnings, and transaction histories.

Interactive Maps: A dynamic feature that allows students to locate nearby rental items with ease, saving time and effort.

Secure Payment Gateways: Reliable and secure payment processing to ensure hassle-free transactions for both lenders and renters.

Community Trust Mechanisms: A robust review and rating system to foster trust and transparency among users.

In addition to meeting the immediate needs of students, Saathi promotes a sustainable approach to resource consumption. By encouraging students to rent instead of buy, the platform helps reduce waste and supports an eco-friendly lifestyle. Moreover, it empowers students to focus on their academic and personal growth without the stress of managing high expenses.

The concept of Saathi aligns with the growing trend of sharing economy platforms, which have demonstrated their potential to transform how people access and manage resources. Unlike generic rental platforms, Saathi is built with a deep understanding of the unique challenges and preferences of the student demographic. By combining affordability, convenience, and sustainability, Saathi aims to redefine the rental experience for students.

**1.2 Literature Review**

The idea of resource-sharing is not new. Platforms like ***OLX, RentHoop, and Unirent*** have successfully established the benefits of peer-to-peer exchanges. However, these platforms often cater to a general audience, leaving students with limited options tailored to their specific needs.

Research studies have shown the increasing adoption of sharing economy models among younger demographics, particularly students. Hamari et al. (2016) highlighted the economic and environmental benefits of such platforms, noting their ability to reduce costs and promote sustainable consumption. Similarly, Teubner & Flath (2015) explored the importance of trust-building mechanisms in enhancing user engagement on peer-to-peer platforms.

Existing platforms, while effective in some areas, lack features that address the core concerns of students. For instance:

Affordability: Many platforms do not prioritize cost-effectiveness, which is crucial for students.

Trust: Generic platforms often fail to build a strong sense of community trust, which is essential for peer-to-peer interactions.

Accessibility: Features like location-based item discovery are often missing, making it harder for users to find what they need.

Saathi is designed to overcome these limitations. By integrating advanced functionalities such as interactive maps, real-time notifications, and personalized dashboards, the platform offers a comprehensive solution tailored to students. Additionally, its emphasis on sustainability aligns with the broader goal of reducing waste and promoting responsible consumption.

Globally, the sharing economy has shown significant potential to address resource inefficiencies. Platforms like Airbnb and Uber have transformed their respective industries by leveraging the principles of resource-sharing and community engagement. Saathi applies these principles to the student rental market, creating a platform that not only meets immediate needs but also fosters long-term benefits for the community.

**1.3 Recommendation System**

One of the standout features of Saathi is its intelligent recommendation system. This feature simplifies the process of finding suitable rental items by analyzing user preferences, geographical location, and past interactions. For example, if a user frequently rents electronics, the platform prioritizes listings for gadgets and related items in their recommendations.

Future iterations of Saathi will leverage machine learning algorithms to enhance the accuracy and relevance of its recommendations. Algorithms such as clustering and collaborative filtering will be used to identify patterns in user behavior and group similar users. By doing so, the platform can provide highly personalized suggestions, further improving user satisfaction and engagement.

**1.4 Objective of the Project**

The objectives of Saathi are both immediate and long-term, aiming to create a platform that addresses the diverse needs of students. Key objectives include:

Affordable Access: Provide students with a cost-effective alternative to purchasing essential items.

Sustainability: Promote eco-friendly practices by encouraging the reuse of items.

Financial Empowerment: Enable students to earn additional income by renting out unused items.

Trust and Transparency: Establish a reliable and trustworthy community through secure transactions and user reviews.

Ease of Use: Simplify the rental process with intuitive technology and seamless interfaces.

**1.5 Key Features**

User-Friendly Interface: The platform is designed to be intuitive and easy to navigate, ensuring a hassle-free user experience.

Interactive Maps: Users can locate rental items in their vicinity, making the process quick and convenient.

Secure Payments: Integrated payment gateways provide reliable and safe transaction options.

Personalized Dashboards: Users can manage all aspects of their rentals in one place, including listings, earnings, and transaction histories.

Review and Rating System: A community-driven feature that fosters trust and transparency.

Admin Oversight: An efficient admin panel ensures smooth platform operations and user management.

**1.6 Scope of the Project**

The scope of Saathi extends beyond being a simple rental platform. It aims to create a sustainable, scalable, and efficient ecosystem that meets the evolving needs of students. Key aspects include:

Peer-to-Peer Ecosystem: Developing a community-driven platform for resource sharing.

Advanced Features: Incorporating functionalities like interactive maps, real-time notifications, and personalized dashboards.

Sustainability Goals: Encouraging responsible consumption and waste reduction.

Scalability: Ensuring the platform can accommodate a growing user base.

User Engagement: Enhancing user experience through intuitive design and robust features.

**1.7 Software/Hardware Requirements**

**Software Requirements:**

Frontend: HTML, CSS, Bootstrap, JavaScript.

Backend: Java Servlets, JSP, JDBC.

Database: MySQL.

Development Tools: NetBeans, Eclipse.

**Hardware Requirements:**

Processor: Intel Core i3 or higher.

RAM: Minimum 4 GB.

Storage: 1 TB HDD or SSD.

Network Interface: Reliable internet connectivity.

**Chapter 2**

**Problem Identification & Feasibility Study**

**2.1 Problem Identification**

The development and implementation of **Saathi**, a student-centric rental platform, involve several critical challenges that need to be addressed to ensure its success. These challenges span multiple areas, including security, algorithm fairness, user engagement, and resource constraints. Below are the key issues identified during the problem identification phase:

1. **Data Privacy and Security**:  
   Since **Saathi** will handle sensitive user data such as personal details, rental preferences, payment information, and location, ensuring robust data security is of utmost importance. The platform must comply with data protection regulations such as the **General Data Protection Regulation (GDPR)** in Europe and the **California Consumer Privacy Act (CCPA)** in the United States. Implementing encryption methods, secure user authentication mechanisms, and regular security audits will be crucial in safeguarding this data. Furthermore, the platform should adopt best practices in cybersecurity to protect user information from unauthorized access and data breaches.
2. **Mitigating Algorithmic Bias**:  
   As the recommendation system is at the core of **Saathi**, ensuring fairness and accuracy in the platform’s suggestions is essential. There is a risk of biased recommendations that could lead to unfair outcomes for certain groups of students, such as unequal access to available items based on location, gender, or other factors. Regularly monitoring the algorithms for biases and adjusting them to ensure diversity and inclusivity in recommendations is critical. To minimize bias, the development process should involve diverse datasets, including a broad spectrum of users with different backgrounds and needs. Continuous refinement and testing of the recommendation models are necessary to ensure that all students, regardless of background, can access items fairly.
3. **Maintaining Accuracy and Reliability in Recommendations**:  
   For **Saathi** to establish credibility and gain user trust, the recommendations provided by the platform must be accurate and relevant. This requires access to up-to-date information about available rental items, pricing trends, and geographic location. Regular updates to the platform’s database are needed to reflect changes in rental item availability, market trends, and user preferences. The algorithms should be designed to provide evidence-based and well-researched suggestions, thus improving reliability.
4. **User Engagement and Adoption**:  
   Encouraging students to use a new platform such as **Saathi** can be challenging, especially for users who are not familiar with technology-based solutions. To increase user adoption, the platform must feature a simple and intuitive interface. Offering tutorials, onboarding sessions, and customer support can help users familiarize themselves with the platform. Additionally, the platform should clearly communicate its benefits through success stories and testimonials from early adopters. As part of engagement, **Saathi** can also include features like reminders, promotions, and rewards for active users, incentivizing continued use.
5. **Dynamic and Evolving Job Market Integration**:  
   The rental market is constantly evolving, and so are students’ needs. Items rented by students may vary seasonally, or new trends may emerge as students' needs change. **Saathi** will need to stay updated with emerging trends by regularly reviewing rental data, user feedback, and market research. Integrating real-time data and collaborating with educational institutions, companies, and market analytics firms can help **Saathi** remain relevant and aligned with students’ ever-changing needs.
6. **Resource Constraints (Time, Expertise, Funding)**:  
   Developing a sophisticated platform such as **Saathi** requires significant resources, both human and financial. A lack of skilled personnel, including experts in areas such as software development, machine learning, and data science, could delay the project. Moreover, funding constraints might limit the ability to invest in necessary technologies, secure cloud services, and customer support teams. Therefore, securing adequate funding, forming strategic partnerships, and acquiring skilled personnel are crucial steps to overcome this challenge. Careful management of resources, including setting realistic project timelines and deliverables, is also essential for maintaining momentum and meeting deadlines.
7. **Accessibility and Inclusivity**:  
   Ensuring that **Saathi** is accessible to students from all backgrounds is vital. The platform should include accessibility features such as screen readers, voice recognition, and multi-language support to cater to students with disabilities and those who speak different languages. Additionally, the design of the platform should consider varying educational levels, cultural differences, and technical expertise, ensuring that it is easy to use for everyone. This inclusive approach will broaden the user base and increase the platform’s effectiveness in helping students.

By addressing these challenges comprehensively, **Saathi** can deliver a robust, user-centric solution that not only meets students’ needs but also ensures long-term success and sustainability.

**2.2 Feasibility Study**

A feasibility study is crucial in determining whether **Saathi** can be developed successfully within the scope of available resources, technology, and market needs. This study examines the critical aspects of technical feasibility, operational feasibility, and economic feasibility. Below is an analysis of each area:

**2.2.1 Technical Feasibility**

Technical feasibility assesses the ability to develop the **Saathi** platform using available technology and infrastructure. Key considerations include:

1. **Availability of Necessary Hardware and Software**:  
   To develop and deploy **Saathi**, the required hardware and software must be available or easily acquired. Hardware requirements include servers for data storage, high-speed networking infrastructure, and cloud-based systems to handle user traffic. The software stack includes web development tools (HTML, CSS, JavaScript, PHP), backend technologies (Java Servlets, MySQL, JDBC), and machine learning libraries (TensorFlow, Scikit-learn). Cloud platforms like AWS or Azure can provide scalable solutions for hosting and data management.
2. **Expertise and Skillsets**:  
   Developing **Saathi** requires a team with expertise in several areas:
   * **Web Development**: Frontend and backend developers proficient in the chosen tech stack.
   * **Data Science and Machine Learning**: Engineers with experience in developing recommendation algorithms and analyzing user data.
   * **Security Experts**: Professionals to ensure data privacy and protection.
   * **UX/UI Designers**: Designers to create an intuitive, user-friendly interface.

The feasibility of acquiring these resources and skills will determine the project’s success. Outsourcing or partnering with academic institutions may provide access to required expertise.

1. **Integration with Existing Systems**:  
   The platform must be able to integrate smoothly with existing IT infrastructure, especially in educational institutions. This includes integrating with student databases, career services, and possibly job boards or internship portals. The compatibility of **Saathi** with these systems should be assessed to ensure a seamless user experience.

**2.2.2 Operational Feasibility**

Operational feasibility determines if the platform can be practically implemented and effectively operated. Key factors include:

1. **User Acceptance**:  
   The system must align with students' expectations. **Saathi** should meet the needs of students for affordability, convenience, and trust. Testing the system with real users through surveys, feedback, and pilot programs will gauge acceptance and identify any operational flaws.
2. **Training and Support**:  
   Providing sufficient training for users and support staff is essential. Since many students may not be familiar with technology-driven solutions, offering tutorials, FAQs, and customer support will be necessary to facilitate easy adoption. Training materials, webinars, and user guides will help students better understand how to use the platform effectively.
3. **Scalability and Flexibility**:  
   As **Saathi** grows, the platform must be able to accommodate increased user traffic, data, and transactions. Designing the system to be scalable will ensure its longevity and effectiveness. Future upgrades and feature enhancements should be planned for to keep up with changes in technology and user needs.

**2.2.3 Economic Feasibility**

Economic feasibility evaluates the financial viability of the **Saathi** project. Key considerations include:

1. **Cost-Benefit Analysis**:  
   The initial and ongoing costs of developing and maintaining **Saathi** should be compared to the anticipated benefits. Costs include development, infrastructure, marketing, and customer support. Benefits could include increased efficiency, access to a broader market, and potential revenue from service fees.
2. **Revenue Generation Potential**:  
   **Saathi** has several potential revenue streams:
   * **Subscription Fees**: Charging users for premium services, such as listing more items or accessing exclusive features.
   * **Transaction Fees**: A small commission on each rental transaction.
   * **Advertising**: Revenue from targeted ads on the platform.
   * **Partnerships**: Collaborations with educational institutions or corporate sponsors.
3. **ROI Calculation**:  
   By calculating the **Return on Investment (ROI)**, **Saathi** can estimate the profitability of the platform over time. Metrics like **Payback Period**, **Net Present Value (NPV)**, and **Internal Rate of Return (IRR)** can help assess the financial success of the project.

**Chapter 3**

**Requirement Analysis**

**3.1 Introduction**

Requirement analysis is a critical phase in the development of the **Saathi** platform. This chapter focuses on gathering, documenting, and analyzing both functional and non-functional requirements essential for designing and implementing the platform.

Effective requirement analysis is crucial to ensure that **Saathi** meets the expectations of its users—students, rental service providers, and administrators. By thoroughly understanding the needs of these stakeholders, we can identify the features and functionalities that **Saathi** must include, while also considering any constraints and performance criteria that need to be fulfilled.

The requirement analysis process involves engaging with stakeholders through interviews, surveys, and workshops to understand their needs and expectations, which are then translated into detailed technical specifications. The chapter will cover both **functional requirements**, which describe the core features and functionalities of the system, and **non-functional requirements**, which define the quality attributes of the system such as security, performance, and usability.

**3.2 Functional Requirements**

Functional requirements specify the essential features and actions that **Saathi** must perform to meet the needs of its users. These features include key capabilities that ensure the platform functions efficiently for renting and lending items. Below are the primary functional requirements for **Saathi**:

**3.2.1 User Profile Module**

The User Profile Module is central to **Saathi**, allowing students and item providers to manage their personal profiles and rental listings. Key functionalities include:

* **User Registration and Login**: Users must be able to create accounts and log in securely using social logins like Google, Facebook, or Twitter.
* **Profile Management**: Users should be able to manage their profile details, including personal information, rented items, and rental history.
* **Personalized Recommendations**: The system will suggest relevant items for rent based on user preferences, location, and past interactions.

**3.2.2 Rental Management System**

The Rental Management System allows users to list and browse available items for rent. This includes:

* **Item Listings**: Users can list items they wish to rent, including uploading photos, setting rental prices, and specifying item details.
* **Search and Filters**: A robust search feature will allow users to filter items by category, location, price, and availability.
* **Transaction Management**: Users should be able to book items for rent, initiate transactions, and track rental histories.
* **Review and Rating System**: After each transaction, users can leave reviews and ratings for items and lenders to help build community trust.

**3.2.3 Payment Gateway Integration**

The platform must support secure payment methods to ensure smooth and trustworthy transactions.

* **Payment Processing**: **Saathi** should support multiple payment methods, including credit/debit cards, online wallets, and direct bank transfers.
* **Security**: The system should use encryption and secure protocols (SSL) to protect financial transactions and sensitive information.
* **Transaction History**: Users should be able to view and manage their transaction records, including rental payments, deposits, and refunds.

**3.2.4 Communication Platform**

The Communication Platform is crucial for facilitating interactions between users, lenders, and admins. It should include:

* **In-app Messaging**: A secure chat system that allows users to ask questions, negotiate terms, or resolve issues related to rentals.
* **Notification System**: Real-time notifications for new messages, rental approvals, item availability, and transaction updates.
* **Admin Communication**: Admins should be able to communicate directly with users for system updates, account issues, or policy enforcement.

**3.3 Non-Functional Requirements**

Non-functional requirements are essential for ensuring that **Saathi** performs reliably, securely, and efficiently. These requirements define the operational attributes of the system, including performance, security, scalability, and usability. Below are the key non-functional requirements for **Saathi**:

**3.3.1 Performance**

Performance is a critical non-functional requirement for ensuring a seamless user experience. The platform should meet the following performance criteria:

* **Response Time**: The system should respond to user actions, such as listing an item or processing a rental transaction, within a reasonable time (e.g., less than 3 seconds).
* **Concurrent Users**: The system must support simultaneous access from multiple users without significant delays or disruptions.
* **High Availability**: The platform should be available 99.9% of the time, ensuring minimal downtime for users.

**3.3.2 Security**

Security is paramount for protecting sensitive user data and ensuring the integrity of the system. Key security features include:

* **User Authentication**: Robust user authentication mechanisms, including multi-factor authentication (MFA), to prevent unauthorized access.
* **Data Encryption**: All sensitive data, including payment information and personal details, must be encrypted both in transit and at rest using industry-standard encryption algorithms.
* **Access Control**: The platform should implement role-based access controls (RBAC) to restrict access to sensitive features and data based on user roles (admin, student, etc.).
* **Regular Security Audits**: The system should undergo regular security audits to identify and fix vulnerabilities.

**3.3.3 Scalability**

Scalability is crucial for **Saathi** as it grows in terms of users, transactions, and data volume. The platform must:

* **Handle Growth**: The system should be designed to scale horizontally, allowing it to handle increasing traffic and new users without significant performance degradation.
* **Database Scaling**: The database should support the addition of new records as more students and items are added, and support scaling solutions such as sharding or partitioning for large data sets.
* **Flexible Architecture**: The platform should have a modular architecture that allows for adding new features, services, or third-party integrations as needed.

**3.3.4 Usability**

To ensure high user engagement and adoption, **Saathi** must offer an intuitive, easy-to-use interface. Key usability features include:

* **User Interface (UI)**: The UI should be clean, modern, and easy to navigate. Information should be presented in a clear and logical manner, with minimal clutter.
* **Mobile-Friendly**: Given the increasing reliance on mobile devices, the platform must be responsive and optimized for both desktop and mobile users.
* **Onboarding and Tutorials**: New users should be guided through the registration and usage process with a tutorial or onboarding guide, explaining how to use the platform effectively.
* **Accessibility**: The platform should incorporate accessibility features like text-to-speech, high-contrast modes, and multilingual support to cater to users with disabilities or those who speak different languages.

**3.3.5 Reliability**

The reliability of the **Saathi** platform is critical to ensuring consistent performance and maintaining user trust. This includes:

* **Error Handling**: The system should gracefully handle errors, displaying informative messages to users and logging issues for further analysis.
* **System Redundancy**: To prevent data loss or downtime, the platform should implement redundant systems for data storage and processing.
* **Backup and Recovery**: Automatic backups should be performed regularly to prevent data loss, and a disaster recovery plan should be in place.

By identifying these functional and non-functional requirements, **Saathi** is positioned to meet the needs of students and service providers while delivering a secure, scalable, and user-friendly platform. These requirements ensure that the platform will be efficient, effective, and adaptable, fostering long-term success and growth.

**CHAPTER 4**

### **PROJECT PLANNING AND SCHEDULING**

#### **4.1 Introduction**

Project planning refers to the phase in project management in which you determine the actual steps to complete a project. This includes laying out timelines, establishing the budget, setting milestones, assessing risks, and solidifying tasks and assigning them to team members.

Project planning is a discipline addressing how to complete a project in a certain timeframe, usually with defined stages and designated resources. One view of project planning divides the activity into these steps:

* Setting measurable objectives
* Identifying deliverables
* Scheduling
* Planning tasks

**Why is project planning important?**

Project planning is important at every phase of a project. It lays out the basics of a project, including the following:

* Scope
* Objectives
* Goals
* Schedule

Planning enables project managers to turn an intangible idea into reality. Key purposes of planning include the following:

* Facilitate communication and provide a central source of information for project personnel;
* Help the project sponsor and other key stakeholders know what is required;
* Identify who will perform certain tasks, and when and how those tasks will happen;
* Facilitate project management and control as the project progresses;
* Enable effective monitoring and control of a project;
* Manage project risk;
* Generate feedback useful for the next project planning phase.



Fig 4.1 Introduction

#### **4.2 Components of a project plan**

The three major parts of a project plan are the scope, budget and timeline. They involve the following aspects:

**Scope:**

The scope determines what a project team will and will not do. It takes the team's vision, what stakeholders want and the customer's requirements and then determines what's possible. As part of defining the project scope, the project manager must set performance goals.

Budget:

Project managers look at what manpower and other resources will be required to meet the project goals to estimate the project's cost.

**Timeline:**

This reveals the length of time expected to complete each phase of the project and includes a schedule of milestones that will be met.

**How do you create a project plan?**

Project planning includes the following 10 steps:

1. Define stakeholders. Stakeholders include anyone with an interest in the project. They can include the customer or end user, members of the project team, other people in the organization the project will affect and outside organizations or individuals with an interest.
2. Define roles. Each stakeholder's role should be clearly defined. Some people will fill multiple roles, however.
3. Introduce stakeholders. Hold a meeting to bring stakeholders together and unify the vision behind the project. The topics covered should include scope, goals, budget, schedule and roles.
4. Set goals. Take what is gleaned from the meeting and refine it into a project plan. It should include goals and deliverables that define what the product or service will result in.
5. Prioritize tasks. List tasks necessary to meet goals and prioritize them based on importance and interdependencies. A Gantt chart can be helpful for mapping project dependencies.
6. Create a schedule. Establish a timeline that considers the resources needed for all the tasks.
7. Assess risks. Identify project risks and develop strategies for mitigating them.
8. Communicate. Share the plan with all stakeholders and provide communications updates in the format and frequency stakeholders expect.
9. Reassess. As milestones are met, revisit the project plan and revise any areas that are not meeting expectations.
10. Final evaluation. Once the project is completed, performance should be evaluated to learn from the experience and identify areas to improve.



Fig 4.2 Component of project plan

#### **4.3 The 5 phases of a project**

Projects typically pass through five phases. The project lifecycle includes the following:

* Initiation defines project goals and objectives. It also is when feasibility is considered, along with how to measure project objectives.
* Planning sets out the project scope. It establishes what tasks need to get done and who will do them.
* Execution is when the deliverables are created. This is the longest phase of a project. During execution, the plan is set into motion and augmented, if necessary.
* Monitoring and management occur during the execution phase and may be considered part of the same step. This phase ensures that the project is going according to plan.
* Closing and review is the final Contracts are closed out and the final deliverables are given to the client. Successes and failures are evaluated.

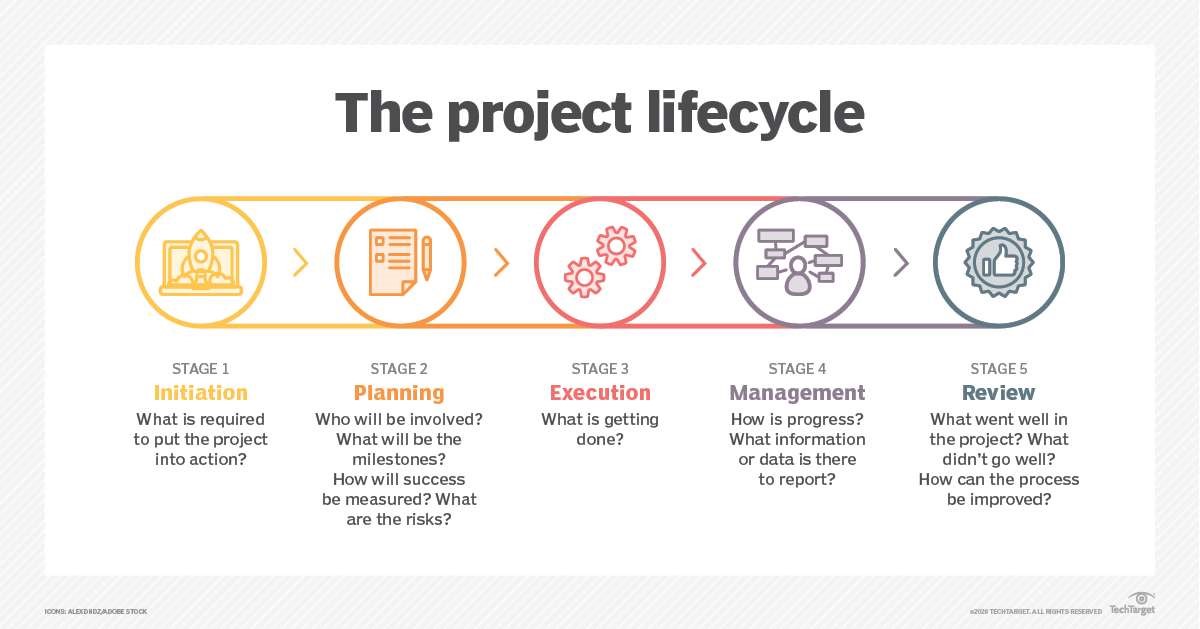


Fig 4.3 5 Phases of Project

#### Some project planning tools and software

Project planning and project management software facilitate the project planning process. The best tools support collaboration among stakeholders, have intuitive user interfaces and provide built-in time tracking and invoicing.

Some project planning software tools include the following:

* Asana offers different project views to suit a team's preferences.
* ClickUp comes with several Agile-based features, including a custom automation builder that lets users create reusable task templates.
* Freedcamp lets users organize their projects using a Gantt chart or Kanban
* Hive has a template creation tool in the task management feature that speeds up task creation.
* Scoro is a combination of tools and includes customer relationship management
* Trello provides Kanban features, budget management, resource management and progress tracking features.
* Wrike integrates with tools like Jira, Slack and Dropbox.

Here’s a revised version of the **PERT Chart** section tailored for your project **Saathi**, followed by a step-by-step guide to creating a PERT chart specific to your project:

**4.4 PERT Chart for Saathi**

A **PERT (Program Evaluation Review Technique)** chart is a valuable project management tool that helps organize, schedule, and coordinate tasks within the development process of **Saathi**. The PERT chart visually represents the project as a network diagram, where numbered nodes (typically circles or rectangles) represent project milestones or events, and labeled arrows or vectors represent tasks or activities that connect these events. The arrows indicate the sequence in which tasks must be completed.

In the context of **Saathi**, the PERT chart helps to plan and manage the entire development process, from the initial planning phase to the final deployment and ongoing maintenance of the platform. The tasks may include system design, database setup, developing key features (like the rental management system, recommendation engine, and payment gateway), integrating AI/ML models, testing, and user acceptance.

**Saathi’s** PERT chart will be particularly helpful in identifying the **critical path**, which is the longest sequence of tasks that must be completed on time for the entire project to be finished by its deadline. By pinpointing these tasks, project managers can focus on ensuring that these critical tasks are prioritized, minimizing delays and preventing disruptions to the overall project timeline.

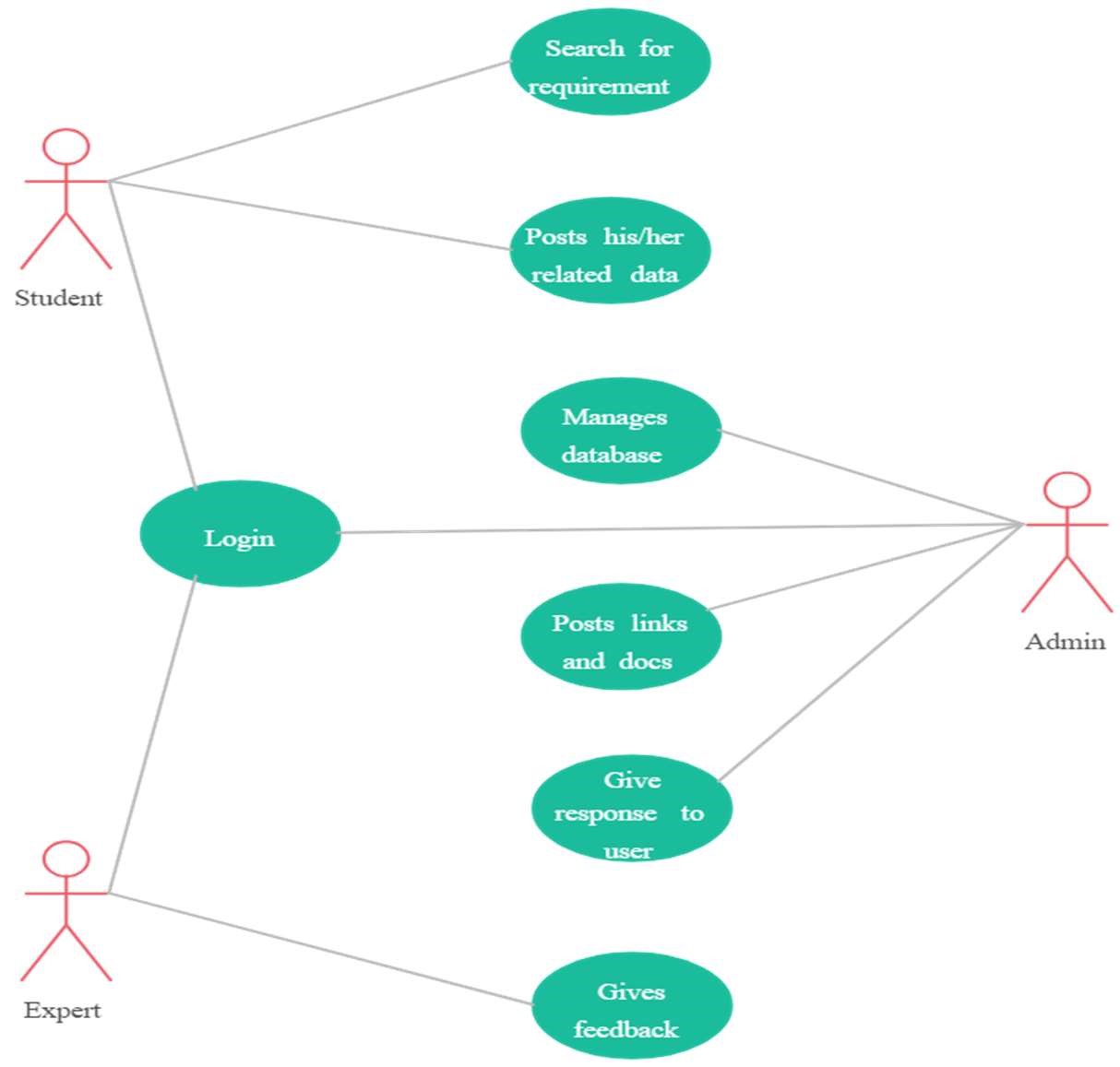


Fig 4.4 Pert Chart

Here’s the updated version of **Chapter 5: Hardware & Software Specification** tailored for **Saathi**, based on the revised hardware and software stack that uses Apache NetBeans IDE, GlassFish server, JSP/Java Servlets for backend, and SQLlog for the database:

**Chapter 5**

**Hardware & Software Specification**

**5.1 Hardware Specification**

The **Saathi** platform will be developed and deployed on a hardware infrastructure that ensures optimal performance, reliability, and scalability. The recommended hardware specifications for the server and client machines are as follows:

**Server:**

* **Processor**: Intel Core i5 or equivalent (or higher)
* **RAM**: 8 GB or higher
* **Storage**: 256 GB SSD or higher (SSD preferred for faster data access and improved performance)
* **Network Interface**: Gigabit Ethernet for high-speed data transfer and network reliability

**Database Server:**

* **Processor**: Intel Core i5 or equivalent (or higher)
* **RAM**: 8 GB or higher
* **Storage**: 256 GB SSD or higher
* **Network Interface**: Gigabit Ethernet

**Client Machines (For Users):**

* **Processor**: Intel Core i3 or equivalent (or higher)
* **RAM**: 4 GB or higher
* **Storage**: 128 GB SSD or higher
* **Network Interface**: 100 Mbps Ethernet or Wi-Fi

These specifications are designed to ensure that **Saathi** functions smoothly under expected usage scenarios, allowing users to interact with the platform seamlessly, and ensuring that server and database performance remains optimal.

**5.2 Software Specification**

The **Saathi** platform will be developed using a combination of server-side and client-side technologies. Below are the software specifications, covering both the development environment and runtime environment for the platform:

**Server-Side Technologies:**

* **Operating System**: Windows Server 2016 or later, or Linux (Ubuntu/CentOS) for flexible hosting options.
* **Web Server**: **GlassFish Server 5.1** or later. GlassFish is an open-source Java EE application server ideal for deploying Java-based applications, including JSP and Java Servlets.
* **Database Management System (DBMS)**: **SQLlog** (or MySQL/PostgreSQL). **SQLlog** will handle the storage and management of all user-related data, rental transactions, and platform-related information.
* **Server-Side Scripting Language**: **Java** (JSP & Java Servlets). JSP will generate dynamic web pages, while Servlets will handle user requests and interact with the backend (database and business logic).
* **Application Development**: The system will utilize **JSP (Java Server Pages)** for the frontend and **Java Servlets** for backend logic and database communication.

**Client-Side Technologies:**

* **Web Browser**: Latest versions of **Chrome**, **Firefox**, **Safari**, or **Microsoft Edge**, ensuring compatibility and smooth user experience across different browsers.
* **Client-Side Scripting**: **JavaScript** and **jQuery** will be employed for dynamic and interactive functionalities, including real-time updates and form validation without reloading the page.
* **Responsive Design Framework**: **Bootstrap 4/5** will be used for creating a mobile-responsive design that adjusts seamlessly to various screen sizes and device types.
* **HTML/CSS**: HTML5 and CSS3 will be used to structure and style the user interface, ensuring a clean, modern, and user-friendly design.

**Development Tools:**

* **Integrated Development Environment (IDE)**: **Apache NetBeans IDE 12.0** (or later) will be used for developing **Saathi**. NetBeans provides excellent support for Java-based web applications, JSP, and Servlets, making it the ideal tool for building and testing the platform.
* **Version Control**: **Git** will be used for version control to manage code and collaborate with other developers. The code repository will be hosted on **GitHub** or **Bitbucket** to track changes and ensure smooth team collaboration.
* **Build and Dependency Management**: **Apache Maven** or **Gradle** may be used for managing dependencies, compiling the project, and packaging it for deployment.

**Security Technologies:**

* **SSL/TLS Encryption**: **SSL/TLS** certificates will be used to secure all communication between the client and server, ensuring that sensitive information like login credentials and payment details are encrypted during transmission.
* **Firewall**: The **server firewall** will be configured to block unauthorized access and ensure that only legitimate traffic is allowed.
* **Anti-malware Software**: Up-to-date anti-malware software will be installed on both server and client machines to protect against cyberattacks and ensure the security of the platform.
* **User Authentication**: **Java Authentication and Authorization Service (JAAS)** will be used to implement secure user authentication, role-based access control, and session management.
* **Data Encryption**: All sensitive data (like passwords and payment details) stored in the **SQLlog** database will be encrypted using industry-standard encryption algorithms, such as **AES-256**.

**Deployment Tools:**

* **GlassFish Server**: **GlassFish 5.1** will be used for deploying the **Saathi** web application. GlassFish is highly suitable for Java EE applications and ensures that the platform can handle high traffic and complex requests with ease.
* **Cloud Deployment**: The **Saathi** platform will be hosted on a **cloud-based platform** like **AWS**, **Google Cloud**, or **Microsoft Azure**. This will provide scalability, high availability, and secure hosting for the application.
* **Content Delivery Network (CDN)**: A **CDN** like **Cloudflare** will be used to distribute static resources (images, CSS, JavaScript files) quickly and efficiently across different geographical locations, improving the load time for global users.

**Summary of Software Requirements :**

| **Component** | **Specification** |
| --- | --- |
| **Operating System** | Windows Server 2016 or later, or Linux (Ubuntu, CentOS) |
| **Web Server** | GlassFish 5.1 or later |
| **Database Management** | SQLlog (or MySQL/PostgreSQL) |
| **Server-Side Language** | Java (JSP, Java Servlets) |
| **Frontend Technologies** | HTML5, CSS3, Bootstrap 4/5, JavaScript, jQuery |
| **Development Tools** | Apache NetBeans IDE 12.0, Visual Studio Code, Git |
| **Version Control** | Git (GitHub/Bitbucket) |
| **Security** | SSL/TLS Encryption, JAAS, Firewall, Anti-malware Software |
| **Deployment Tools** | GlassFish, AWS/Google Cloud, CDN (Cloudflare) |

**CHAPTER 6**

### **Design**

#### **6.1 FrontEnd Architecture**

The frontend architecture of **Saathi** is designed to ensure an intuitive user experience, seamless interactivity, and maintainable code. The project utilizes a combination of HTML, CSS, and JavaScript, with the following architectural principles and structure:

1. **HTML (Structure)**:
   * The foundation of the frontend is built with well-structured HTML, ensuring semantic markup for accessibility and search engine optimization.
   * The layout is organized into reusable components like headers, footers, and dynamic sections, promoting modularity.
2. **CSS (Styling)**:
   * CSS is used to enhance the visual aesthetics and responsiveness of the application.
   * A modular approach is followed, with styles organized into separate files based on components or pages (e.g., header.css, forms.css).
   * Media queries ensure a responsive design, providing an optimized user experience across devices.
3. **JavaScript (Interactivity)**:
   * JavaScript powers dynamic functionalities such as form validation, user input handling, and real-time updates without reloading the page.
   * The project follows a structured approach, organizing JavaScript into distinct files for better maintainability and readability (e.g., login.js, dashboard.js).
4. **Integration with JSP**:
   * The frontend is tightly integrated with JSP to dynamically render data fetched from the MySQL database. JSP provides the server-side logic, while the frontend focuses on the presentation layer.
5. **Frameworks and Libraries**:
   * While the project does not rely on frontend frameworks like React or Angular, it is designed in a way that allows for future scalability and potential integration with such technologies.
6. **Performance Optimization**:
   * Techniques such as minified CSS and JavaScript, optimized images, and asynchronous data loading enhance the performance of the frontend.
7. **User-Centric Design**:
   * The frontend architecture is built with a user-first approach, ensuring ease of navigation and interaction, which is critical for a project like **Saathi**.

This architecture not only ensures a clean and maintainable codebase but also facilitates future enhancements and scalability, making **Saathi** a reliable and robust web application.

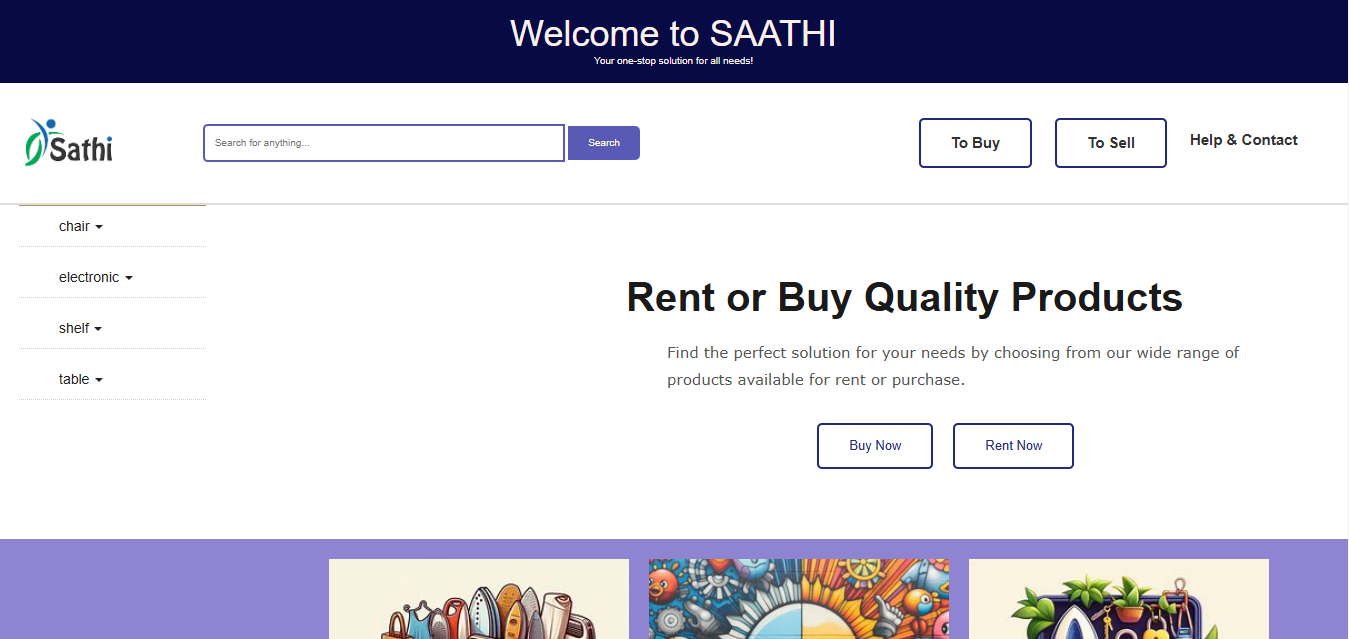


Fig 6.1 FRONT-END

#### **6.2 BackEnd Architecture**

The backend architecture of **Saathi** leverages **MySQL** as its relational database management system (RDBMS) for efficient data storage and retrieval. MySQL is widely recognized for its robustness, scalability, and ease of integration, making it a popular choice for dynamic web applications like Saathi.

In **Saathi**, MySQL handles critical data operations such as user information, product records, and other dynamic content. The database interacts seamlessly with **JSP (JavaServer Pages)** and **Java**, which form the core of the backend logic. This combination enables the application to process user requests, handle server-side business logic, and provide real-time responses.

The backend architecture avoids the use of traditional LAMP stack components like PHP. Instead, it focuses on **Java-based technologies**, which ensure scalability, modularity, and high performance. This architecture is designed to manage complex data transactions while maintaining a secure and reliable connection between the user interface and the database.

By integrating MySQL with JSP and Java, **Saathi** ensures a smooth and efficient backend workflow, delivering a responsive and reliable user experience

#### 6.3 Data Flow Diagram

The data flow diagram shows the flow of data within any system. It is an important tool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It’s also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD: - In the DFD, four symbols are used and they are as follows.

1. A square defines a source (originator) or destination of system data.
2. An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.

1. A circle or a “bubble “(Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.

1. An open rectangle is a data store-data at rest, or a temporary repository of data.

#### 6.4 Context Level Diagram

This level shows the overall context of the system and its operating environment and shows the whole system as just one process. Canteen Management System is shown as one process in the context diagram; which is also known as zero level DFD, shown below. The context diagram plays important role in understanding the system and determining the boundaries. The main process can be broken into subprocesses and system can be studied with more detail; this is where 1st level DFD comes into play.

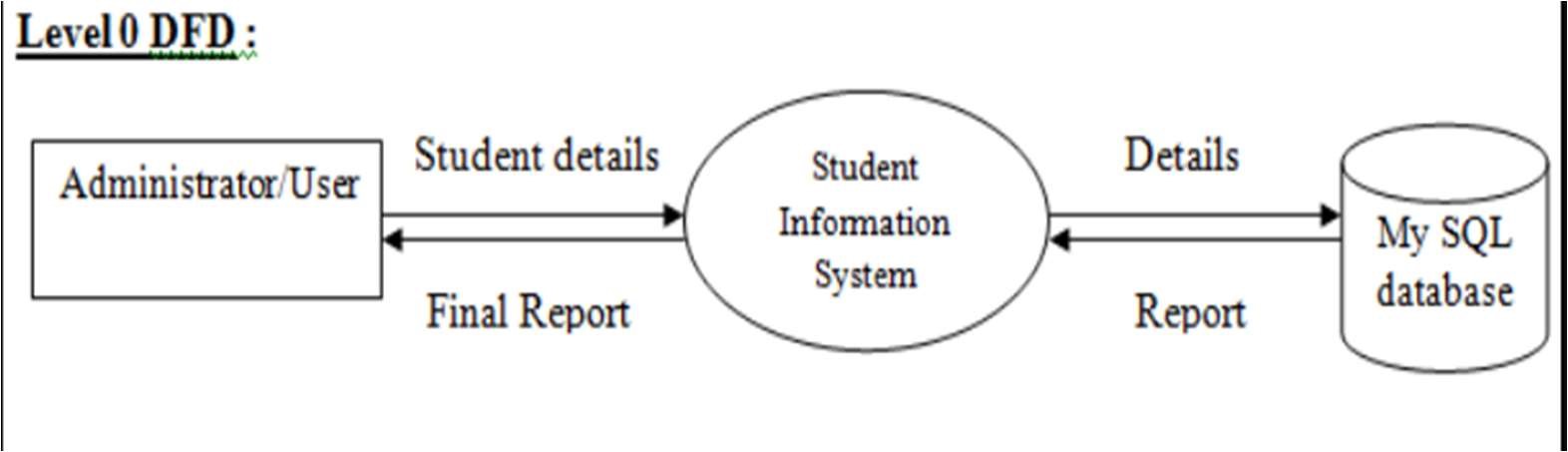


Fig 6.4 0-Level DFD

The Level 0 DFD shows that the Administrator/User interacts with the Student Information System by providing student details and receiving final reports. The Student Information System processes the data, stores it in the MySQL database, and retrieves necessary information from the database to generate reports. This high-level diagram provides a clear overview of the system's functionality and how data flows between the user, system, and database.

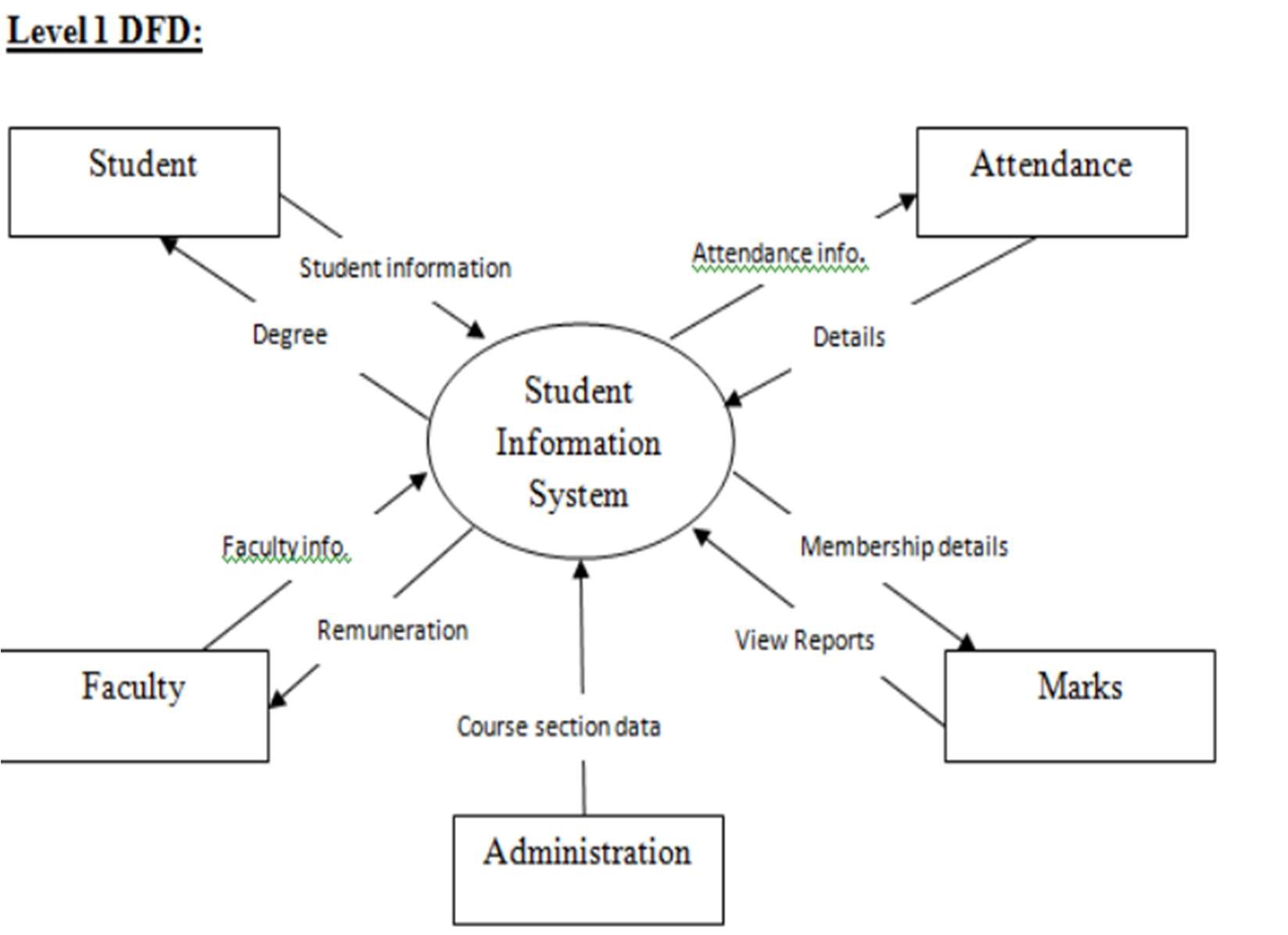


Fig 6.4- 1 Level DFD

The provided diagram is a Level 1 Data Flow Diagram (DFD) for a Student Information System. A Level 1 DFD breaks down the primary process of the system (shown in a Level 0 DFD) into its main sub-processes and shows the flow of information between these sub-processes and external entities. Here’s a detailed explanation of the components and the data flow in this Level 1 DFD:

This Level 1 DFD for the Student Information System details how different entities (students, faculty, administration) interact with the system, and how various data flows between them. The diagram shows the inputs provided by these entities and the outputs generated by the system. It helps in understanding the detailed working of the Student Information System by illustrating how data is processed, stored, and utilized within the system to manage student-related information effectively.

CHAPTER 7

### ER-DIAGRAM

#### 7.1 Entity-relationship model: -

The entity-relationship model or entity-relationship diagram (ERD) is a data model or diagram for high-level descriptions of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity-relationship diagrams.

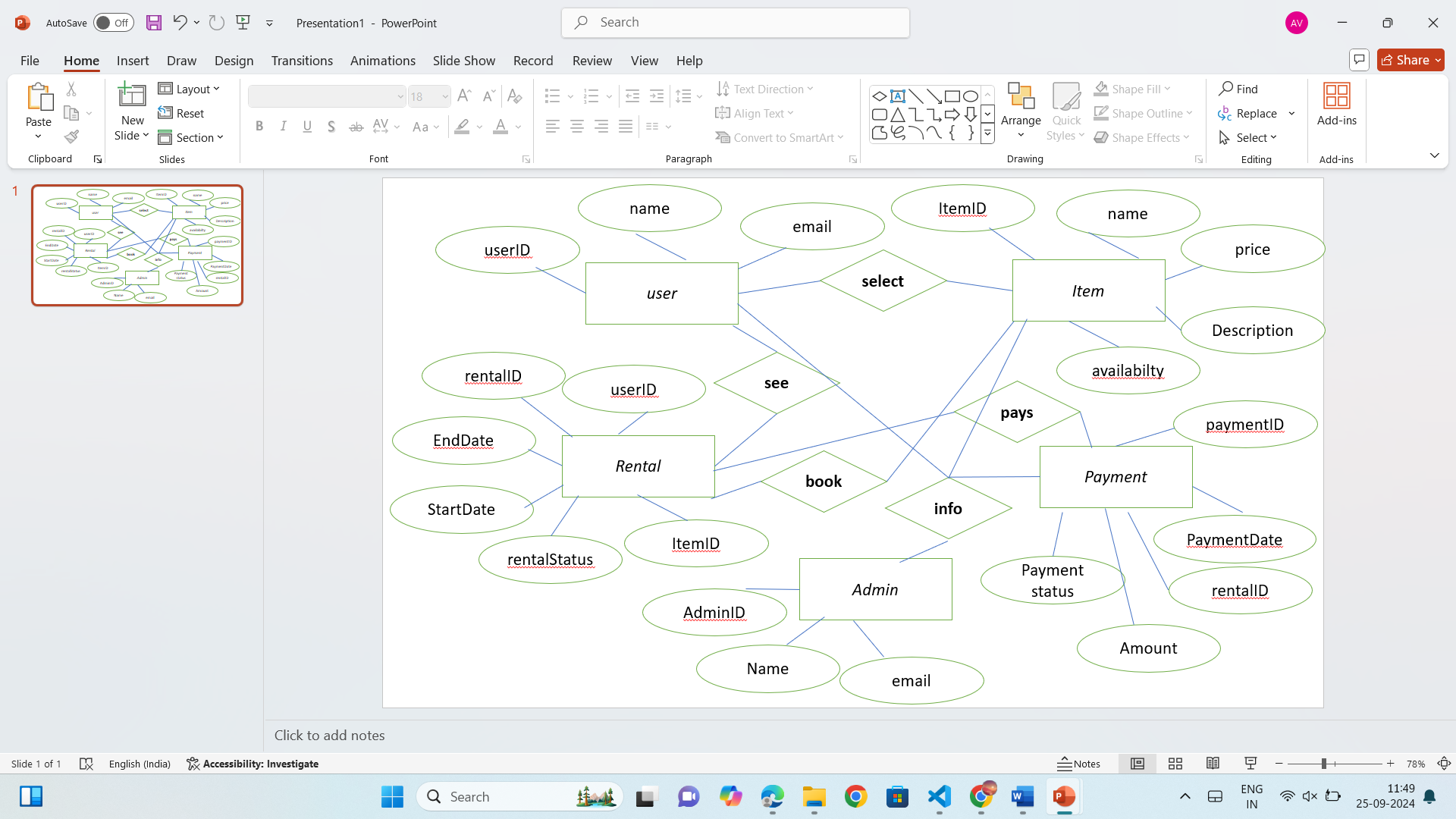
****

Fig 7.1 ER Diagram

The flow of the Saathi platform ensures that students can easily navigate through the various features and functionalities offered. Below is a detailed theoretical explanation of the platform's project flow, guiding users from the initial access to the completion of transactions, community building, and administrative management.

 **User Registration/Login**: Students can sign up using social login options (Google, Twitter, Facebook).

 **User Dashboard**: After logging in, users manage rentals, item listings, rental history, and payments from a personalized dashboard.

 **Item Listing**: Users list items (textbooks, electronics, furniture, etc.) with descriptions, prices, and availability.

 **Search and Interactive Map**: Users search for items using filters or view items on an interactive map to find nearby rentals.

 **Rental Request and Approval**: Renters request items for a specific period, and lenders approve or reject the requests.

 **Secure Payment Gateway**: Payments are processed securely, and deposits are held until the item is returned in good condition.

 **Item Pick-Up/Delivery**: Renters and lenders arrange item pick-up or delivery based on mutual convenience.

 **Return of the Item**: Once the rental period ends, the item is returned, and deposits (if applicable) are refunded.

 **Admin Panel**: Admins manage users, items, transactions, and resolve disputes.

 **Notifications**: Users receive real-time alerts for rental approvals, payments, and return reminders.

 **Community Engagement**: Saathi fosters resource sharing and sustainability, promoting eco-friendly practices and collaboration among students.

CHAPTER 8

### FORM DESIGN

8.1 Home Page:

The homepage serves as the primary entry point for users visiting the website. It typically showcases featured products, promotions, and various product categories to attract users. The homepage also includes navigation links to other sections like the product catalog, contact details, registration, and login pages, making it easy for users to explore the website further.

[Home] [Services] [About Us] [Contact Us] [Login] [Register]

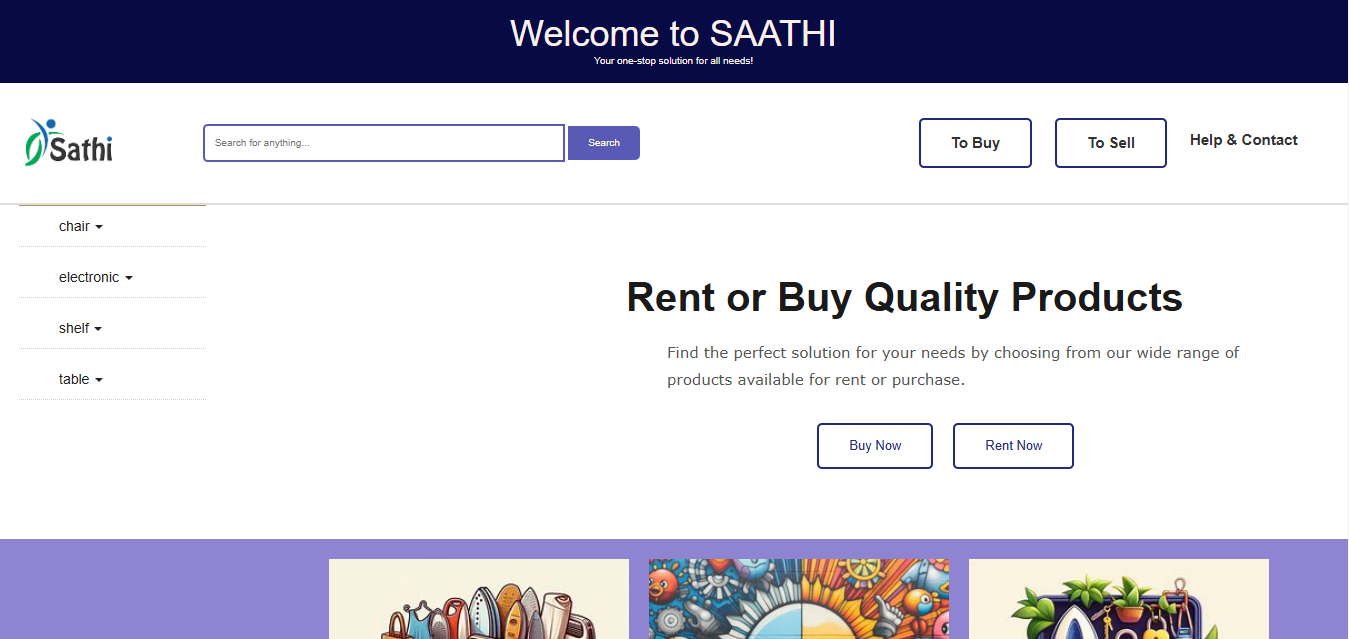


Fig 8.1 Home Page

8.1.1 Contact Us:

The Contact Us page allows users to communicate directly with the website's support or sales team. It usually contains a contact form with fields for the user's name, email, subject, and message, along with alternative contact methods like phone numbers and email addresses. In some cases, a map showing the company’s location is also provided to help users find the physical address.

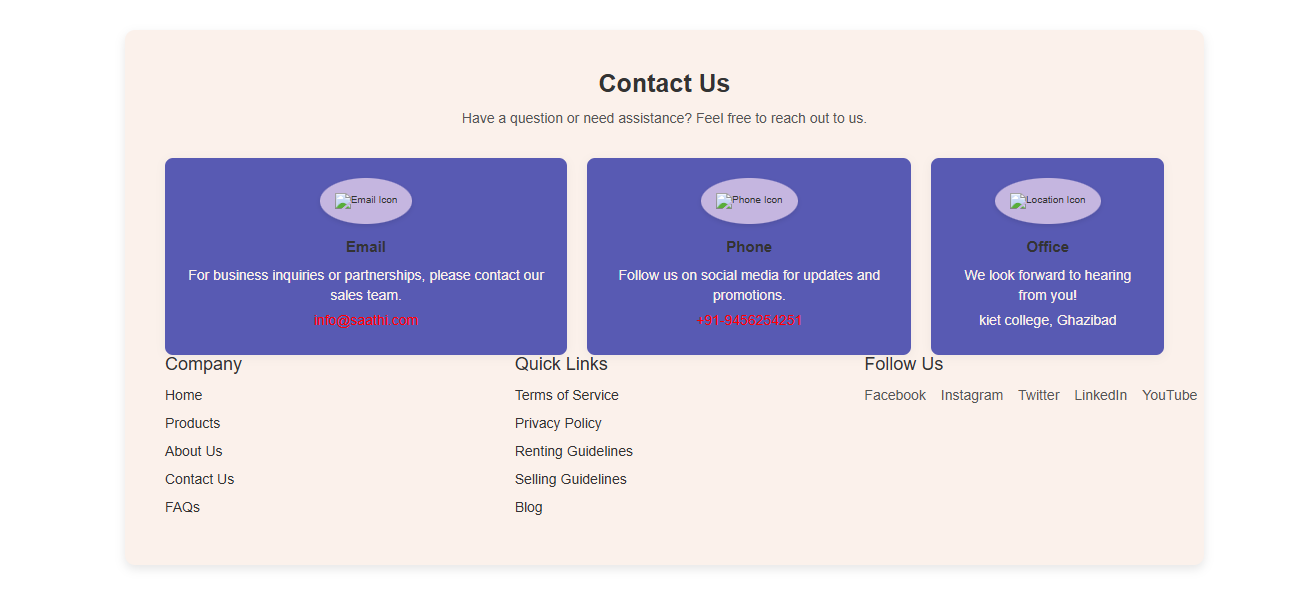


Fig 8.1.1 Contact us

8.2 Registration and Login:

The registration page enables new users to create an account by entering necessary details like name, email, password, and contact information. Once registered, users can log in using their credentials on the login page to access their account and perform actions like managing their orders or personal details. Both pages typically include security measures such as password strength validation and options for recovering forgotten passwords.

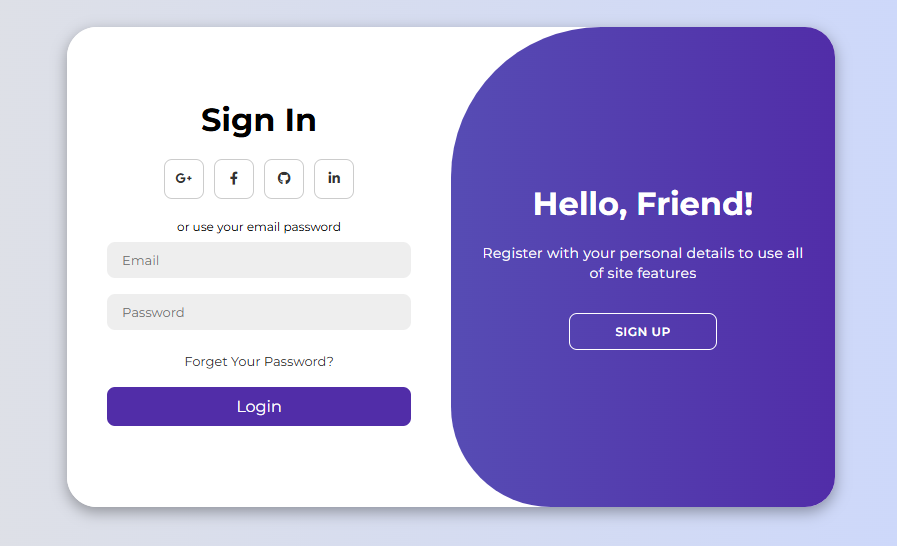


Fig. 8.2 Login

8.3 Vendor Login

The vendor login page is dedicated to merchants who want to manage their products on the platform. Vendors use this page to securely log in and access their dashboard, where they can update product listings, monitor orders, and manage their accounts. This page ensures that only authorized vendors have access to their business data and product management tools.

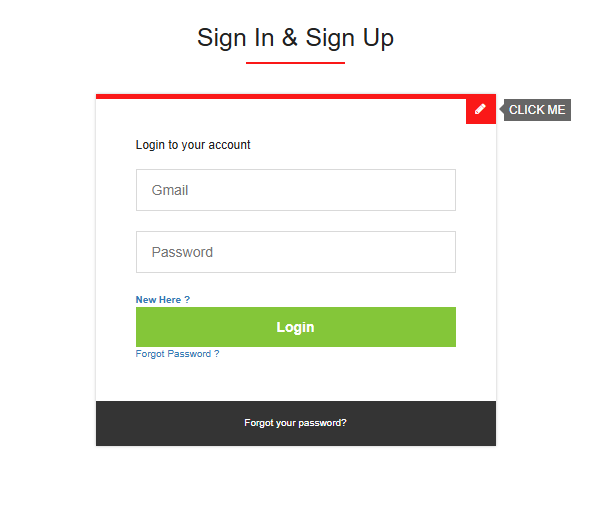


Fig 8.3 Vendor Login

Products:

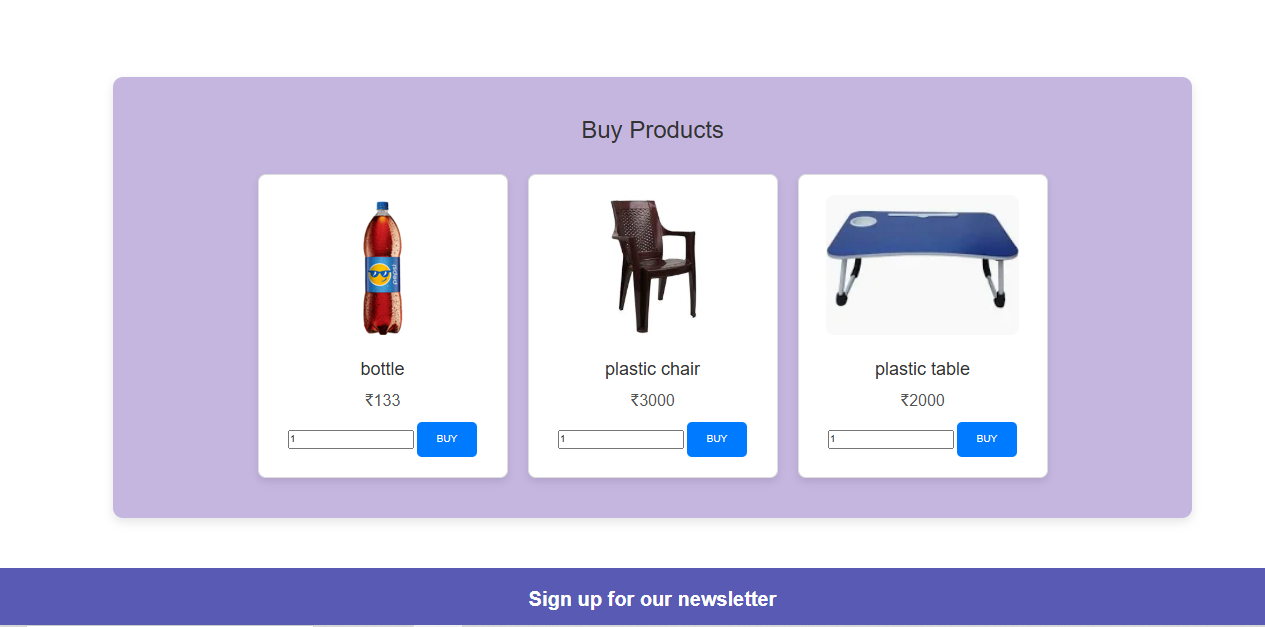
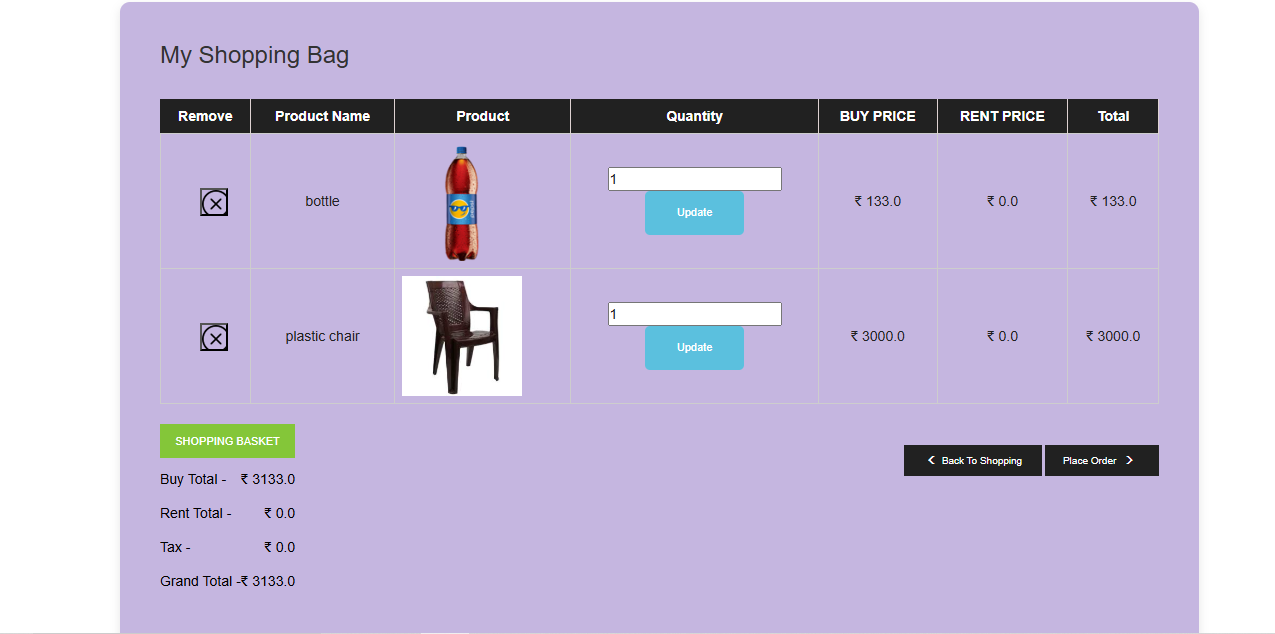
The product page displays detailed information about individual products, including images, descriptions, prices, and availability. Users can explore various product features, view reviews and ratings, and add items to their shopping cart or wishlist. The page may also offer related products or suggest alternatives, allowing users to browse similar items that might interest them. 

Fig 8.4 products

#### 9.5 checkout

The checkout page is where users finalize their purchases by reviewing their cart, providing shipping details, and selecting a payment method. It typically includes an order summary, a list of items being purchased, and options for applying discount codes or selecting delivery preferences. After confirming the details, users can complete the transaction and receive an order confirmation.



#### Fig 8.5 checkout

**Summary**

Saathi is a student-centric rental platform designed to promote sustainability and cost-effectiveness by enabling students to rent items they no longer need. Developed using Java Servlets, the platform integrates cutting-edge technologies and a user-friendly interface to create a seamless rental experience.

Key features include social login, personalized dashboards, interactive maps, secure payment systems, and an efficient admin panel. By fostering a resource-sharing community, Saathi empowers students to save money and earn by renting items, contributing to a sustainable lifestyle. This project reflects a commitment to solving real-world problems with innovative, student-focused solutions.

**Conclusion**

The Saathi platform represents a significant step towards creating a sustainable, student-focused economy. By enabling students to rent their unused items, the project promotes resource-sharing and provides an avenue for financial savings and earnings.

The development process focused on crafting a robust and scalable architecture, utilizing technologies like Java Servlets, MySQL, and APIs for enhanced functionality. With features like interactive maps, personalized dashboards, and secure payment gateways, Saathi ensures a seamless and trustworthy user experience.

The project’s successful deployment highlights its potential for addressing challenges like affordability and accessibility, making it a valuable addition to the rental platform ecosystem. Moving forward, Saathi is well-positioned to expand its features, reach a wider audience, and adapt to the evolving needs of its users, ensuring long-term relevance and impact.

**Future Scope and Further Enhancement of the Project**

* 1. Integration of Notification System

Implement a system to send real-time alerts to users about new listings, rentals, or updates related to their activity on the platform.

* 1. Expansion of User Base

Extend the platform to include non-student users while maintaining the core focus on students to broaden its reach and usability.

* 1. Development of a Mobile Application

Launch a mobile app to complement the web platform, providing users with on-the-go access to rental listings and platform features.

* 1. Feedback Mechanism

Introduce a feedback system for users to rate and review listings and services, ensuring continuous improvement in user satisfaction.

* 1. Advanced Analytics and Reporting

Integrate advanced analytics to track user engagement, monitor rental trends, and improve the efficiency of the platform through data-driven insights.

* 1. Integration with Additional APIs

Expand functionality by incorporating APIs for better payment options, enhanced location services, and more interactive features.

**Bibliography**

**Books :**

Java: The Complete Reference by Herbert Schildt

Head First Design Patterns by Eric Freeman

Online Resources

Oracle Java Documentation

Bootstrap Documentation

Google Maps API Documentation

**Tools :**

MySQL Documentation (https://dev.mysql.com/doc/)

GitHub for version control and project management (https://github.com)