



**Tribhuvan University  
Institute of Science and Technology**

**A Project Report  
On  
“Rental Service Nepal”**

**Submitted to:  
Department of Computer Science and Information Technology  
Amrit Science Campus**

*A project submitted in partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Computer Science and Information Technology awarded by IOST,  
Tribhuvan University*

**Submitted by  
Ason Gautam [25616/77]  
Shristi Chand [25701/77]**

**Under the Supervision of  
Abhimanyu Yadav**

JAN 2025

## **Disclaimer**

I hereby declare that this study, titled **Rental Service Nepal**, is the outcome of my original research and dedicated efforts. While conducting this project, I have thoroughly reviewed and acknowledged the contributions and related works of other researchers in the field. Every piece of information, data, and analysis included in this study has been carefully verified to ensure its accuracy and relevance. I take full responsibility for the authenticity and reliability of the information presented herein and affirm that any errors or discrepancies, if found, are solely my accountability. This project reflects my sincere commitment to maintaining the highest standards of academic integrity and originality.

### **Name and Signature of the Students:**

Ason Gautam

Shristi Chand

Date: 1/21/2025

## Recommendation

This is to certify that the project entitled ***Rental Service Nepal***, prepared and submitted by **Ason Gautam** and **Shristi Chand**, has been carried out under my supervision. This project has been completed as a partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Computer Science and Information Technology, awarded by Tribhuvan University. The work presented in this project demonstrates their dedication, research skills, and ability to apply theoretical knowledge to practical implementation. I hereby recommend this project for acceptance by Tribhuvan University, as it meets the required academic, technical, and research standards.

**Name of the Supervisor:** Abhimanyu Yadav

**Signature:**

**Designation:**

**Date Signed:**

## Certificate

This project, entitled **Rental Service Nepal**, prepared and submitted by **Ason Gautam** and **Shristi Chand**, has been thoroughly examined and evaluated by us. We hereby confirm that the project meets the academic standards and requirements necessary for the award of the degree of **Bachelor of Science (B.Sc.)** in Computer Science and Information Technology, as conferred by Tribhuvan University. The work presented in this project reflects the students' dedication, research capabilities, and practical application of their knowledge. Therefore, we accept this project for the fulfillment of the degree requirements.

.....  
**External Examiner**  
.....

.....  
**Project Coordinator**  
.....

.....  
**Supervisor**  
.....

.....  
**Head/Program Coordinator**  
.....

## Acknowledgement

We would like to express our sincere gratitude to our supervisor, **Mr. Abhimanyu Yadav**, for his guidance and support throughout this project, ***Rental Service Nepal***.

We are also thankful to the **Institute of Science and Technology (IOST)**, Tribhuvan University, for providing the platform and resources to carry out this work.

A heartfelt thanks to our friends and our families for their encouragement and support, which kept us motivated to complete this project successfully.

**Ason Gautam**

**Shristi Chand**

## Abstract

The technology-driven rental platform enhances efficiency in real estate, vehicle, and equipment rentals using the Waterfall Model for structured development. It integrates the Haversine Algorithm for precise geolocation-based searches, Smart Search & Filtering for optimized item retrieval, and Dynamic Sorting for improved user navigation. Features like automated notifications, real-time updates, secure transactions, and rigorous testing ensure system reliability, performance, and scalability. The project achieved successful results, improving rental management, user experience, and service accessibility for both service providers and customers.

**Keywords:** *Rental, Waterfall, Haversine, real-time, transaction, navigation*

## TABLE OF CONTENTS

### **SUPERVISOR'S RECOMMENDATION**

### **LETTER OF APPROVAL**

### **ACKNOWLEDGEMENT**

### **ABSTRACT**

### **TABLE OF CONTENTS**

### **LIST OF FIGURES**

### **LIST OF TABLES**

### **LIST OF ABBREVIATIONS**

## **1. INTRODUCTION ..... 1**

1.1. Introduction .....	1
1.2. Problem Statement.....	1
1.3. Objectives .....	2
1.4. Scope and Limitation.....	2
1.5. Development Methodology .....	3
1.6. Report Organization .....	4

## **2. BACKGROUND STUDY AND LITERATURE REVIEW ..... 5**

2.1. Background Study .....	5
2.2. Literature Review .....	6
2.3. Current Systems .....	8
2.4. The problem with Current System .....	10

## **3. SYSTEM ANALYSIS .....11**

3.1. Requirement Analysis .....	11
3.1.1. Functional Requirement .....	11
3.1.2. Non-Functional Requirement .....	14
3.2. Feasibility Analysis .....	15
3.2.1. Technical Feasibility .....	15
3.2.2. Operational Feasibility .....	15
3.2.3. Economic Feasibility .....	16
3.2.4. Schedule Feasibility .....	16
3.3. Analysis .....	18

## **4. SYSTEM DESIGN ..... 20**

4.1. Design .....	20
4.2. Workflow Diagram .....	23
4.3. Algorithm Details .....	24

<b>5. IMPLEMENTATION AND TESTING .....</b>	<b>26</b>
5.1. Implementation .....	26
5.1.1. Tools Used .....	28
5.1.2. Implementation Details of Modules .....	29
5.2. Testing .....	32
5.3. Result Analysis .....	34
<b>6. CONCLUSION AND FUTURE RECOMMENDATION .....</b>	<b>35</b>
6.1. Conclusion .....	35
6.2. Future Progress .....	35
<b>7. REFERENCES .....</b>	<b>36</b>

## LIST OF FIGURES

Figure 1 : Waterfall Model .....	3
Figure 2 : Rental Platform .....	7
Figure 3 : HamroBazar .....	8
Figure 4 :AirBnB .....	9
Figure 5 :UseCase .....	13
Figure 6 :Gantt Chart .....	17
Figure 7 :Sequence Diagram .....	18
Figure 8 :Class Diagram .....	20
Figure 9 :WorkFlow Diagram .....	23
Figure 10 :Database Schemas .....	29

**LIST OF TABLES**

Table 1 : Evaluation of current system .....	9
Table 2 : Basic Functions .....	14
Table 3 : Testing Cases .....	32
Table 4 : Result Analysis .....	33

## LIST OF ABBREVIATIONS

JWT	JSON Web Token
AJAX	Asynchronous JavaScript and XML
MVC	Model-View-Controller
API	Application Programming Interface
UI	User Interface
UX	User Experience
jQuery	JavaScript Query
PHP	Hypertext Preprocessor

# **Chapter 1 : INTRODUCTION**

## **1.1. INTRODUCTION**

In Nepal, renting is becoming a practical solution for many people who want to save money and access products and services without the high cost of ownership. The country has a diverse population where people from different backgrounds live together peacefully. Many individuals move to developed areas for better job opportunities, leading to a higher demand for affordable housing, food, and essential services.

In today's fast-paced world, people are always looking for convenient and cost-effective options. Rental services offer a variety of short-term products and services, making life easier and more budget-friendly. While the rental service industry in Nepal is still relatively new, it is growing rapidly. As the cost of living increases, buying expensive items and services becomes difficult. Rental providers help by offering affordable alternatives that meet customer needs.

In recent years, Nepal's rental market has expanded significantly, with a wide range of rental services available. The industry has grown to the point where it can cater to almost all customer demands. Studies suggest that more people are choosing to rent instead of purchasing, mainly due to high ownership costs, limited availability of products and services, and the convenience of leasing.

## **1.2. PROBLEM STATEMENT**

In Nepal, owning products and services can be expensive, and not everyone can afford to buy everything they need. As cities grow and more people move for jobs and better opportunities, the demand for affordable living, goods, and services is increasing. However, buying things outright is not always practical, especially for short-term needs.

Rental services can be a good solution, but in Nepal, they are still not very well structured. Many people do not know where to find rental services, and businesses offering rentals often struggle to reach customers. There is also a lack of trust in rental agreements, making it difficult for both renters and service providers to connect easily.

To solve this problem, Nepal needs a better system for rental services. A well-organized rental market would help people save money, make better use of available resources, and offer a more convenient alternative to buying. information about product availability, ensuring clients can conveniently obtain and reserve the items they require.

### **1.3. OBJECTIVES**

- To conduct research into developing a service system that offers comprehensiveness
- To gain a thorough understanding of similar systems and structures that could provide valuable insights for building an effective rental platform.
- To establish a direct connection between customers and rental providers, allowing renters to communicate without intermediaries and reducing the risk of misuse and fraudulent activities.
- To implement seller verification procedures to enhance the credibility of rental service providers and minimize risks associated with unreliable sellers.
- To introduce a user reporting system to identify and blacklist troublesome users, helping to reduce biased or harmful activities within the rental platform.

### **1.4. SCOPE AND LIMITATION**

#### **Scope:**

- The platform will help people easily find and rent products.
- It will allow direct communication between renters and service providers without needing middlemen.
- A verification system will be in place to ensure that sellers are trustworthy.
- Renting will offer a more affordable option compared to buying expensive items.

#### **Limitations:**

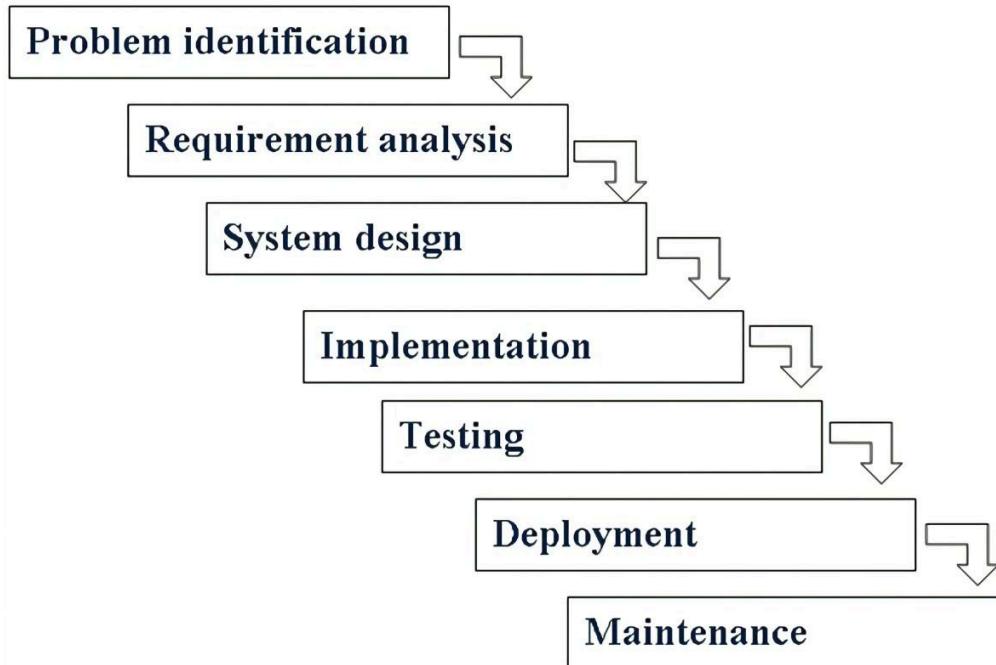
- The system will require internet access, which may not be available to everyone.
- Even with verification, there may still be trust issues between renters and providers.
- Some areas might have limited rental options due to fewer service providers.
- There is a chance that some users may misuse the platform, so monitoring will be needed.

## 1.5.DEVELOPMENT METHODOLOGY

The **Waterfall Model** will be used for developing the **rental service system** in Nepal, ensuring a structured and sequential development process. The first phase, **Requirement Analysis**, will define the system's features, including **user registration**, **item listings**, **rental requests**, and **payment integration**. The **System Design** phase will focus on planning the **database schema**, user interfaces, and defining the relationships between **users**, **rental items**, and **transactions**. **Laravel** will be used for backend development, **jQuery** for frontend interactions, and **MySQL** for database management.

In the **Implementation** phase, **jQuery** will be used to develop a dynamic frontend, while **Laravel** will handle backend logic and **MySQL** will store necessary data. Once implemented, the system will undergo **Testing**, including unit, integration, and security testing to ensure reliability and safety. After successful testing, the system will be **Deployed** on a cloud platform for public access.

Finally, the **Maintenance** phase will involve continuous updates and improvements based on user feedback and system performance. By following the Waterfall Model, the rental service system will be developed in a clear, organized, and efficient manner.



**FIGURE 1: WATERFALL MODEL OF RENTAL SERVICE NEPAL**

## **1.6 REPORT ORGANIZATION**

This report is structured as follows:

### **Chapter 1: Introduction**

- Provides an overview of the project, including background, problem statement, objectives, scope, and significance.

### **Chapter 2: Literature Review**

- Reviews existing rental service platforms, their challenges, and how this system improves upon them.
- Discusses related technologies and frameworks used in rental service development.

### **Chapter 3: System Analysis**

- Identifies and analyzes functional and non-functional requirements.
- Examines the challenges in the current rental market and how the proposed system addresses them.

### **Chapter 4: System Design**

- Details the architectural design, database schema, and component interactions.
- Provides UI/UX wireframes and system workflow diagrams.

### **Chapter 5: Implementation and Testing**

- Describes the development process, technologies used.
- Explains the testing approach, including unit testing, integration testing, and security testing.

### **Chapter 6: Conclusion and Future Enhancements**

- Summarizes key findings and discusses the system's limitations.
- Provides recommendations for future improvements and scalability.

## CHAPTER 2 : BACKGROUND STUDY AND LITERATURE REVIEW

### 2.1.BACKGROUND STUDY

Over the past 15 years, the number of rental housing households in Nepal has grown from 2.2% to 7.8% (National Shelter Policy, 2012). Despite this growth, the rental market faces many challenges that make it inefficient and difficult to access. Limited rental options and concerns about the reliability of rental products and services discourage potential renters. Additionally, buying new products is expensive, prompting individuals and businesses to look for affordable rental solutions.

The slow adoption of technology in the rental sector further limits its growth. Many services still lack online booking and payment options, making the process inconvenient. To address these problems, this project aims to create a trustworthy rental service provider in Nepal. The goal is to offer a wide range of reliable and affordable rental options, including flats, houses, and vehicles. By implementing strict quality measures and using technology to simplify operations, the platform will provide a user-friendly experience for customers across the country.

However, the current rental system faces several challenges. Outdated technology and low-cost systems make it difficult to update services frequently. Theft of rental equipment and vehicles increases financial burdens due to insurance costs, reducing the availability of rental options. Some listings include false information, such as poorly maintained vehicles, which puts renters at risk. Discriminatory practices in house rentals, such as caste-based biases, also create barriers. Additionally, listings for stolen properties and rental fraud cause further problems.

Rent prices often increase in areas with high maintenance costs, making rentals unaffordable for many. A lack of regulation in pricing structures makes it difficult to find high-quality rental options. Misleading advertisements can cause confusion when the rented product does not match what was promised. The system also struggles with complex bookings, sometimes listing unavailable properties as available, leading to issues when customers arrive to rent a property that has already been taken.

High agent fees make renting expensive, and fraudulent agents further reduce trust in the system. Theft and loss of rental items also create major issues for providers, making user verification and tracking necessary.

To solve these problems, rental services must maintain products properly and provide detailed, transparent information. Unreliable users should be blacklisted, and strict policies should be enforced during registration. Filtering services based on budget and renter preferences can help streamline price negotiations. In house rentals, clear communication about rent increases is essential, especially during uncertain times like pandemics. A strong

system with good customer service and improved product availability will ensure a better rental experience.

## **2.2.LITERATURE REVIEW**

*A Rental Platform Service Supply Chain Network: The Role of Digital Detection Technology and Big Data Marketing:*

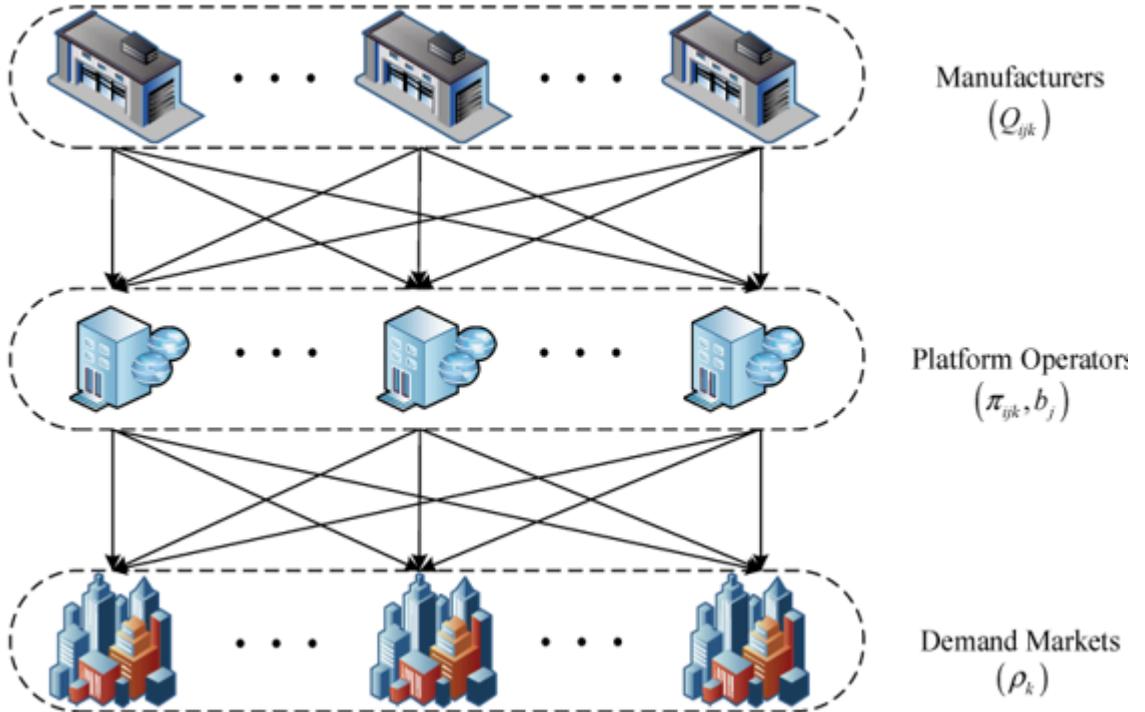
Technology is transforming various industries by reshaping decision-making processes, particularly in the rental service sector. Digital advancements have significantly influenced how rental platforms operate and make decisions in manufacturing. This study highlights the crucial role of big data marketing and digital detection technology in improving rental service supply chains.[8]

The research introduces a supply chain equilibrium model based on variational inequality and Nash equilibrium theory. This model outlines the ideal conditions for manufacturers and rental platform operators to optimize their services. The findings emphasize that digital detection technology enhances supply chain coordination and revenue growth, while big data marketing enables platforms and manufacturers to share costs effectively. This combination improves stability and efficiency across the rental network.

However, the study also acknowledges some limitations, particularly in terms of budget constraints. It assumes that all participants in the supply chain behave rationally, which may not always be the case. These limitations suggest areas for future research to refine and expand the model.[7]

Manufacturers provide rental equipment, while platform providers (such as Caterpillar rental locations or Ctrip automobile rentals) act as middlemen, facilitating transactions and earning commissions. These platforms allow users to choose from a variety of manufacturers. However, balancing supply and demand is essential for smooth operations. If marketing expenses become too high, platform operators may lose motivation. To maintain profitability, manufacturers invest in digital detection technologies and share marketing costs with platforms, ensuring long-term sustainability.

Despite its valuable insights, the study does not fully explore certain key challenges in the rental industry. For example, it does not address service reliability issues or the difficulties arising from infrequent rental demands among specific user groups. Events like weddings, where rentals are needed for short periods, present unique supply-and-demand dynamics that require further investigation. These gaps highlight opportunities for future research to improve the rental platform supply chain further.[2]

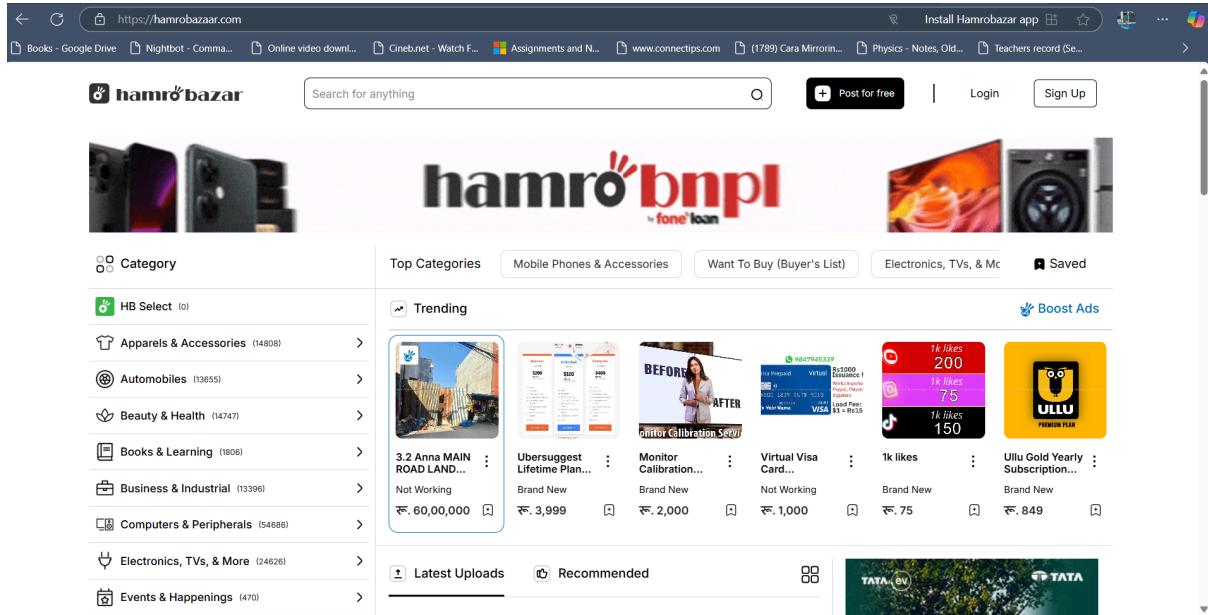


**FIGURE 2:RENTAL PLATFORM SERVICE SUPPLY CHAIN  
(PENG AND LI, 2023)**

The rental market includes various platforms that facilitate renting goods and properties, catering to both local and global audiences. In Nepal, platforms like Hamro Bazar serve as popular marketplaces for renting and buying various products, including real estate, vehicles, and electronic appliances. Meanwhile, on a global scale, platforms such as Airbnb provide vacation and home rental services, enabling homeowners to lease their properties to travelers. These platforms play a crucial role in connecting renters with service providers, offering a convenient and efficient way to access rental services. However, many of these systems often lack proper regulation, security measures, and service reliability, leading to concerns such as fraudulent listings, payment disputes, and inconsistent service quality. Addressing these challenges through better verification processes, secure payment systems, and improved user trust mechanisms could significantly enhance the overall rental experience.

### *HAMRO BAJAR(NEPAL):*

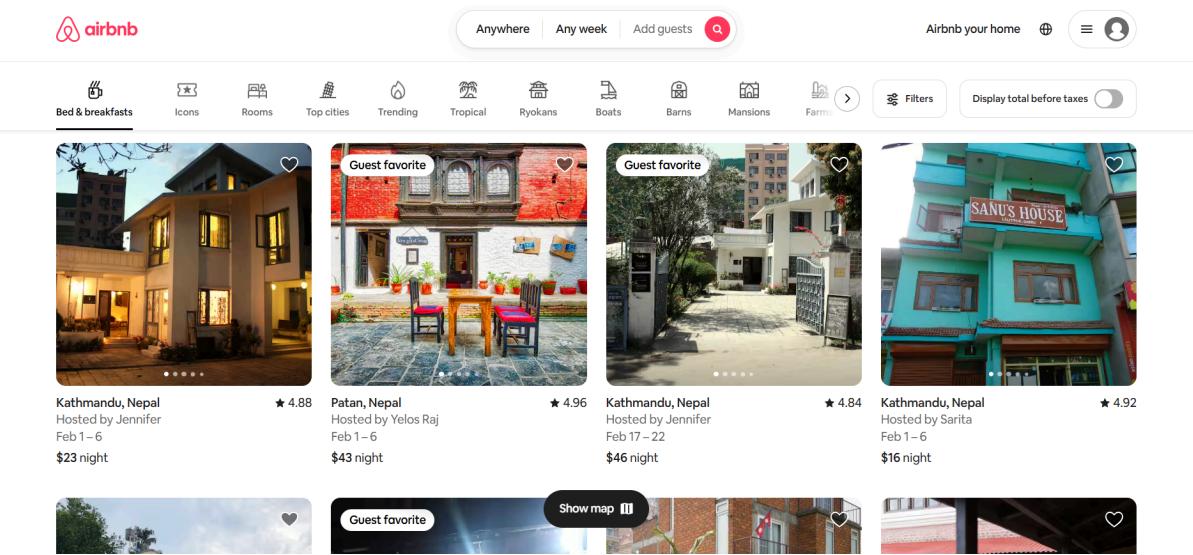
Hamro Bazar is a popular online marketplace in Nepal where users can buy, sell, and rent various products, including houses, vehicles, electronics, and furniture. It operates as a classifieds platform, allowing direct deals between buyers and sellers without middlemen. However, it lacks strict security measures and verification, leading to potential fraud risks.



**FIGURE 3:HAMROBAZAR**

### *Airbnb(Globally):*

Airbnb is a global online marketplace that allows people to rent homes, apartments, and unique stays directly from hosts. It connects travelers with property owners, offering short-term and long-term rental options. The platform provides secure payments, reviews, and customer support, but pricing fluctuations and hidden fees can sometimes be concerns for users.



**FIGURE 3:AIRBNB**

Features	Hamro Bazaar	Expedia	Zillow	Mero Property
Functionality and Implementation	4	3	3	3
User Interface	3	3	4	3
Adaptability and design	4	3	3	2
Reliability and efficiency	3	3	5	3
Flexibility and versality	2	4	3	2
Ratings and Feedback	4	4	4	3

**TABLE 1:EVALUATIONS OF CURRENT SYSTEM**

## **2.4. PROBLEM WITH CURRENT SYSTEM**

The current rental systems face several challenges, including a lack of regulation, which leads to inconsistent pricing and policies. Security risks and fraud are common, with fake listings and unverified users creating trust issues. Many platforms provide outdated or misleading rental information, making it difficult for users to find reliable options. Pricing inconsistencies, high agent fees, and fluctuating rates add to the inconvenience. Additionally, limited customer support and poor dispute resolution further reduce the efficiency and reliability of these systems, making the rental process less trustworthy and inconvenient for users.

## 3.SYSTEM ANALYSIS

### **3.1.REQUIREMENT ANALYSIS**

Requirement Analysis is a critical phase in software development where the needs and expectations of stakeholders are gathered, analyzed, and documented to ensure the system meets its purpose. It helps in identifying what the system should do (functional requirements) and how it should behave or operate (non-functional requirements).

#### **3.1.1.FUNCTIONAL REQUIREMENT**

*ACTORS:*

1. Customer User
2. Seller
3. Admin

#### **Customer User Use Cases:**

##### *Login / Register*

- *Register*
- *Login*
- *Update Profile*

##### *Browsing & Searching*

- *Search Rental Items*
- *Browse / Filter Rental Items*
- *View Item Details*

##### *Wishlist Management*

- *Manage Wishlist*
- *View Wishlist*
- *Add to Wishlist*

##### *Booking & Communication*

- *Request Booking*
- *Contact Seller*

##### *Seller Onboarding*

- *Request to become a Seller*
- *Wait for Request to be Accepted*

##### *Support*

- *Write Queries*

## **Seller Use Cases:**

### ***Item Management***

- *Create Item Listing*
- *Manage Item Listing*
- *Archive Listings*

### ***Customer Interaction***

- *Handle Item Request*
- *Contact Customer*

### ***Premium Seller Features***

- *Become a Premium Seller (extends Online Payment)*

## **Admin Use Cases:**

### ***User & Listing Management***

- *Manage Users*
- *Manage Listings*
- *Blacklist Users*

### ***Advertisements***

- *Create / Approve Advertisement*

### ***User Requests***

- *Review Verification Request*
- *Analyze User Data*

### ***Support***

- *Manage User Queries*

### ***Admin Login***

**FIGURE 5: USE CASE DIAGRAM**



Action	Admin	Seller	Customer/user
Add rental item	Yes	Yes	No
Edit rental item	Yes	Yes / self-added	No
Delete	Yes	Yes/ self-added	No
View listing	Yes	Yes	Yes
Book listing	Yes	Yes	Yes
Archive listing	Yes	Yes	No

**TABLE 2: BASIC FUNCTIONS****3.1.2.NON FUNCTIONAL REQUIREMENTS**

- The system should handle 1000+ concurrent users with <3s page load time.
- Search results and booking requests should process within 2-5 seconds.
- JWT/OAuth2 authentication with hashed passwords.
- SSL/TLS encryption for online payments, compliant with PCI-DSS.
- Blacklist mechanism for fraudulent users.
- 99.9% uptime, recovery within 30 minutes, with load balancing.
- Responsive UI, key actions in <3 clicks, supports multiple languages.
- Scalable to 100,000+ users, cloud-based with efficient database handling.
- RESTful APIs for mobile app integration, modular development for expansion.
- Zero downtime updates, logs for logins, admin actions, errors.
- Compliance with GDPR and local e-commerce regulations.

### 3.2.FEASIBLE ANALYSIS

#### 3.2.1. TECHNICAL FEASIBILITY

- Frontend: jQuery for UI interactions and AJAX requests.
- Backend: Laravel for handling business logic and API requests.
- Database: MySQL for structured data storage and retrieval.
- Authentication: JWT for secure user authentication.
- Password Security: bcrypt for hashing passwords.
- API Integration: RESTful APIs for frontend-backend communication.
- Payment Gateway: FonepayDev for secure online transactions.

 Technically feasible with jQuery, Laravel, MySQL, JWT, bcrypt, API, and FonepayDev.

#### 3.2.2. OPERATIONAL FEASIBILITY

- User-Friendly Interface: jQuery ensures a responsive UI, making it easy for customers, sellers, and admins to navigate.
- Efficient Management: Laravel provides structured workflows for user registration, item listings, payments, and authentication.
- Secure Transactions: JWT ensures secure logins, bcrypt protects passwords, and FonepayDev handles reliable payments.
- Scalability & Performance: MySQL optimizes data handling, and Laravel's API structure ensures smooth operations.
- Administrative Control: Admin can manage users, listings, payments, and blacklist fraudulent accounts.
- Market Adaptability: The system aligns with online rental service needs, ensuring usability for customers and sellers.

 Operationally feasible—secure, user-friendly, and scalable for real-world deployment.

### **3.2.3. ECONOMIC FEASIBILITY**

- The required technologies like Laravel, MySQL, jQuery, JWT, bcrypt, and FonepayDev API are either free or affordable to use.
- Deployment can be managed on low-cost cloud hosting platforms.
- Maintenance costs are minimal due to the availability of community support and open-source frameworks.

 Economically feasible with cost-effective tools and resources.

### **3.2.4. SCHEDULE FEASIBILITY**

#### **Planning (Week 1-2)**

- Define system requirements.
- Identify key features and user roles.

#### **Wireframing (Week 3)**

- Design system layout and user flows using jQuery components.

#### **Design Process (Week 4-5)**

- Create UI components and set up Laravel views.

#### **Front-End Development (Week 6-10)**

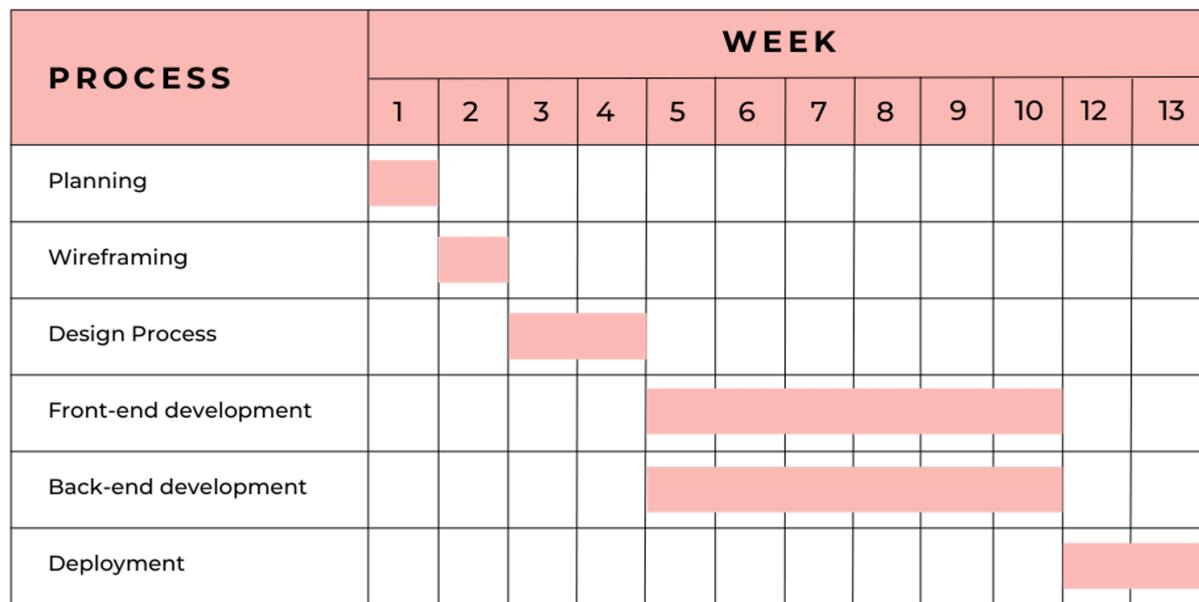
- Develop interactive features using jQuery and integrate with Laravel APIs.

#### **Back-End Development (Week 7-11)**

- Implement JWT authentication, bcrypt password hashing, MySQL database, and API connections.
- Integrate FonepayDev for payments.

#### **Deployment (Week 12-13)**

- Host the system on a cloud platform.
- Perform final testing and go live.

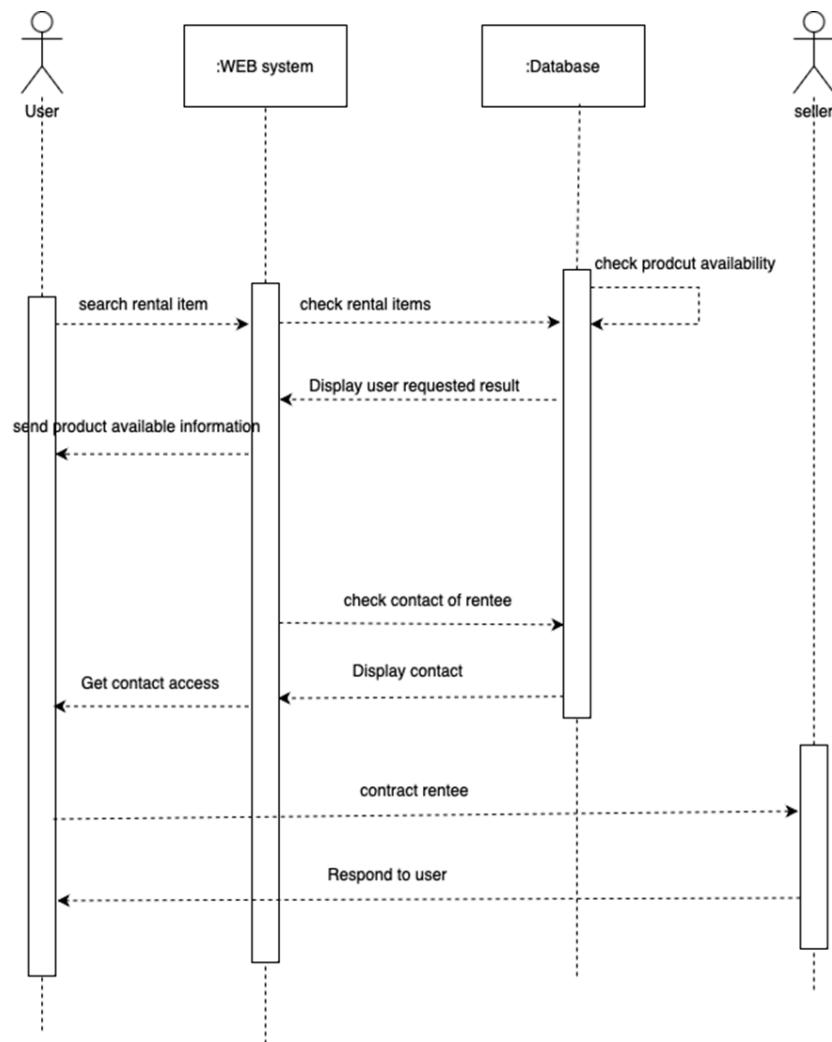


**FIGURE 6: GANTT CHART**

- ✓ The project is schedule-feasible, aligning with a structured 13-week timeline.

### 3.3.ANALYSIS

The sequence diagram illustrates the interaction flow between the User, Web System, Database, and Seller in a rental system. It follows a structured process where a user initiates a search request for a rental item. The Web System processes this request by querying the Database, which checks the availability of rental products. Once verified, the database sends the relevant results back to the web system, which then displays them to the user.



**FIGURE 7: SEQUENCE DIAGRAM**

After reviewing the available items, the user can request detailed product information and access the seller's contact details. The Web System retrieves the necessary contact details by querying the Database, which responds with the requested information. The Web System then displays the seller's contact details to the user, enabling direct communication. The user can then proceed to contact the seller to finalize the rental agreement. The seller acknowledges and responds to the user, completing the interaction.

This sequence diagram highlights a clear, step-by-step process of request and response between the different components, ensuring an efficient workflow for searching, retrieving, and renting products. The Web System serves as the main intermediary, managing user requests, database queries, and displaying results, while the Database ensures data integrity and availability. Finally, the seller plays a passive role until the user reaches out for further discussion.

## 4.SYSTEM DESIGN

### 4.1.DESIGN

This Class Diagram represents the structure of a rental system, showing entities (classes), their attributes, methods, and relationships.

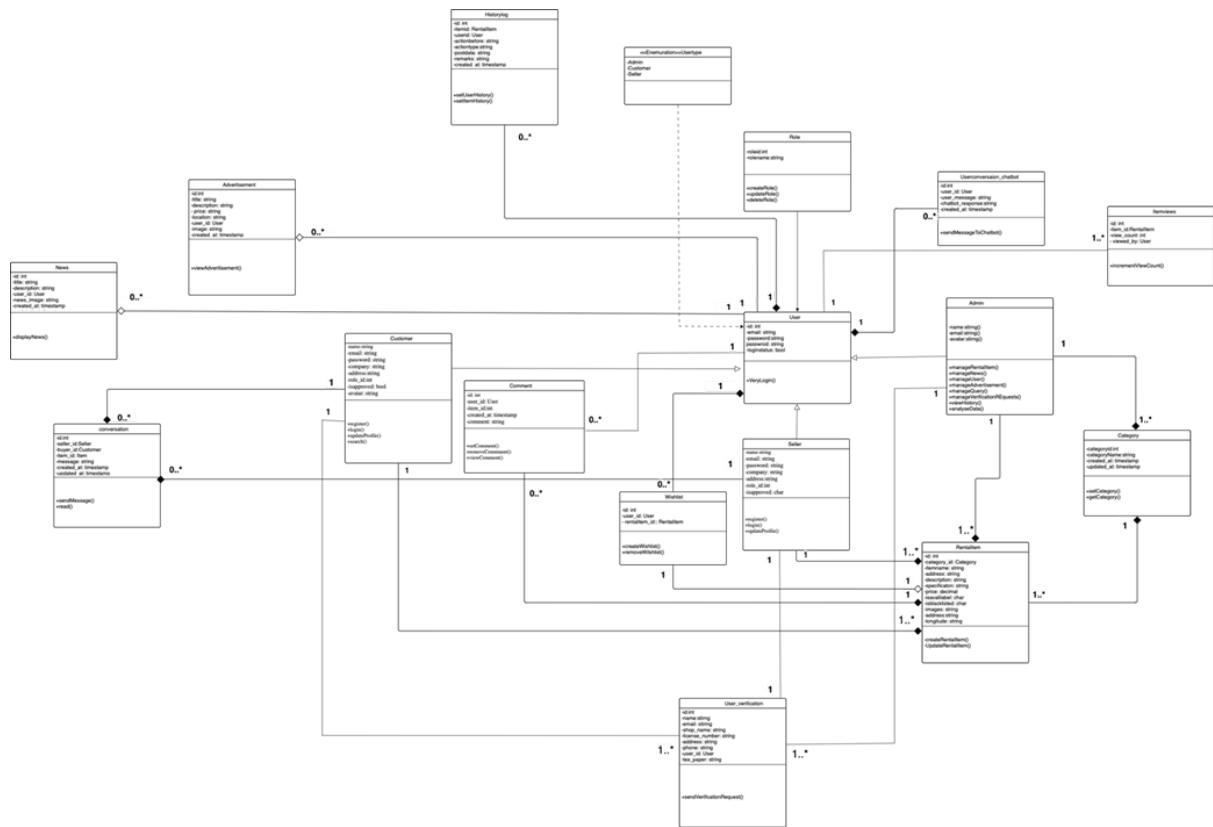


FIGURE 8:CLASS DIAGRAM

#### 1. User

- Attributes: **user\_id**, **name**, **email**, **password**, **phone**
- Methods: **viewApp()**, manages login and user activities.

#### 2. Customer (inherits from User)

- Attributes: **address**, **contact\_number**, **isVerified**

- Methods: `register()`, `search()`, allows customers to register and search for rental items.

### 3. Seller (inherits from User)

- Attributes: `company_name`, `contact_char`
- Methods: `register()`, enables sellers to register and list items.

### 4. Admin (inherits from User)

- Attributes: `admin_id`, `name`, `email`
- Methods: `manageRentalItems()`, `manageUsers()`, handles platform moderation.

### 5. RentalItem

- Attributes: `rental_id`, `title`, `price`, `description`, `category_id`
- Methods: `viewRentalItem()`, `updateRentalItem()`, represents rental items.

### 6. Category

- Attributes: `category_id`, `category_name`
- Methods: `getCategory()`, groups rental items into categories.

### 7. Wishlist

- Attributes: `wishlist_id`, `user_id`, `rentalItem_id`
- Methods: `createWishlist()`, allows users to save preferred rental items.

### 8. Comment

- Attributes: `id`, `user_id`, `rental_id`, `content`, `timestamp`
- Methods: `addComment()`, enables users to add comments.

### 9. Communication

- Attributes: `id`, `sender_id`, `receiver_id`, `message`, `timestamp`

- Methods: `sendMessage()`, manages chat between users.

## 10. User\_verification

- Attributes: `verification_id`, `user_id`, `status`, `document_proof`
- Methods: `sendVerificationReport()`, handles seller verification.

## 11. News

- Attributes: `id`, `title`, `content`, `timestamp`, `author`
- Methods: `displayNews()`, provides updates and announcements.

## 12. Advertisement

- Attributes: `ad_id`, `title`, `description`, `image`, `seller_id`
- Methods: `viewAdvertisement()`, displays promotional ads.

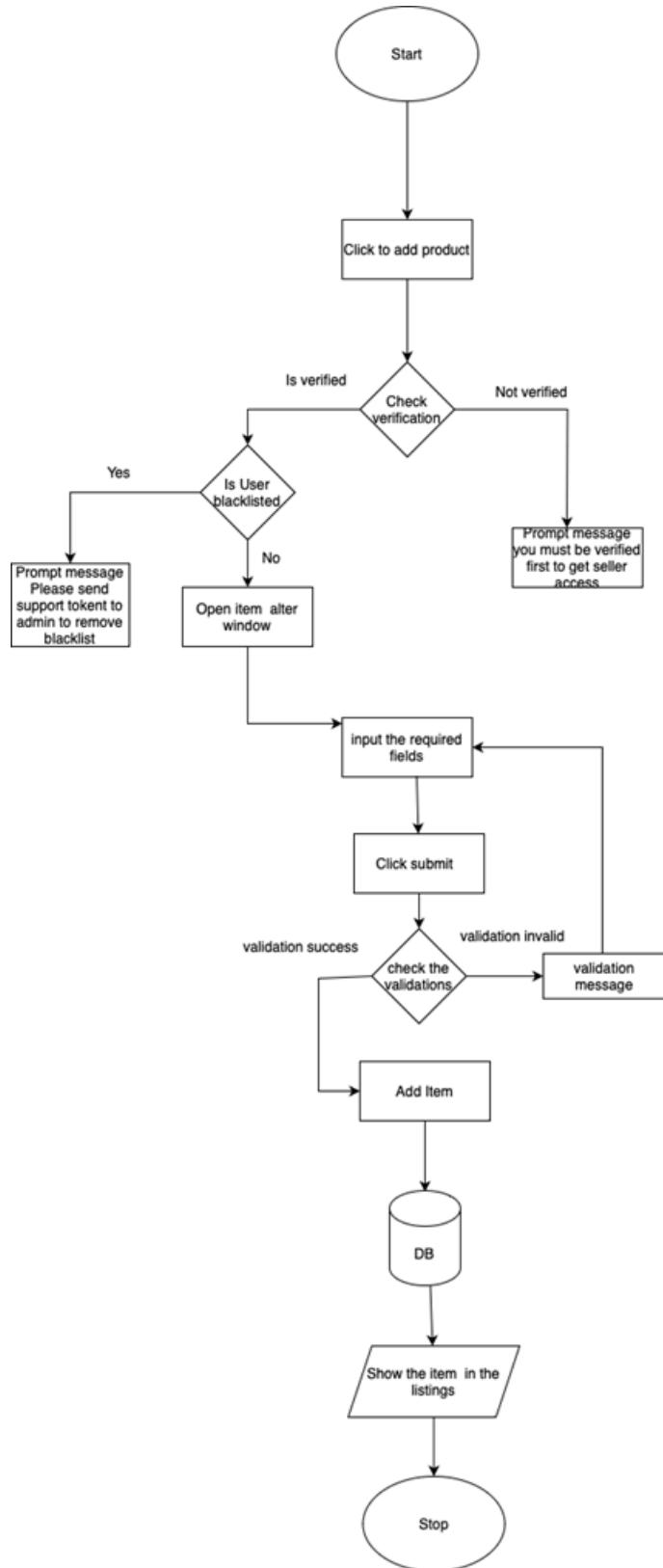
## 13. Role

- Attributes: `role_id`, `role_name`
- Methods: `createRole()`, `assignRole()`, defines system roles.

## 14. UserConversation ChatBot

- Attributes: `message_id`, `user_id`, `message_text`, `timestamp`
- Methods: `sendMessageToChatbot()`, provides automated chat support.

## 4.2.WORKFLOW DIAGRAM



## **FIGURE 10:Workflow of the user adding the rental item to the system**

### **4.3.ALGORITHM DETAILS**

#### **1. Haversine Algorithm (Location-Based Filtering)**

We have implemented the Haversine formula in JavaScript to calculate the distance between two geographical points (latitude and longitude). This helps in determining how far a rental property is from the user's location. The function `calculateDistance(lat1, lon1, lat2, lon2)` ensures that if any coordinate is missing, it returns Infinity, effectively excluding such items from the nearby search. The function `displayNearbyProperties(properties)` dynamically updates the frontend to show only relevant rental listings based on their proximity. Additionally, it checks if the property has images, and if not, a default placeholder image is displayed.

#### **2. Searching Algorithm**

In our Laravel-based backend, we have implemented a database query with multiple conditions to retrieve relevant rental items. The `$searchInput` is first sanitized using `trim()` to ensure that unnecessary spaces do not affect the query. The query searches for rental items by matching the item name, location, item type (SALE), and category name using a combination of LIKE operators for partial matching. We have ensured that only active (non-archived) listings are retrieved. Additionally, the query joins multiple tables, including `rentalitem`, `users`, `category`, and `wishlists`, ensuring that relevant metadata (such as seller details and wishlist status) is fetched. The filtered results are then passed to the sales/sales view for rendering.

#### **3. Sorting Algorithm (Organizing Search Results)**

We have implemented a sorting mechanism that allows users to sort rental listings dynamically based on their preference. The sorting is done based on `$selectedFilter`, with the following options:

Newest: Orders items by their creation date in descending order (latest first).

Oldest: Orders items by creation date in ascending order (oldest first).

Price Low to High: Sorts items based on price in ascending order.

Price High to Low: Sorts items based on price in descending order.

Finally, the sorted items are paginated (`paginate(10)`) to limit the number of results displayed per page, improving performance and user experience.

## Conclusion

We have successfully integrated location-based filtering using the Haversine algorithm, advanced searching with multiple criteria, and dynamic sorting for better result organization. These implementations enhance user experience by ensuring relevant rental listings based on location, preferences, and budget while maintaining system efficiency.

The Haversine algorithm allows users to find nearby rental properties, improving accessibility and convenience. Our search algorithm refines results by filtering through item names, locations, categories, and availability, making searches more precise. Additionally, the sorting feature ensures users can browse listings in their preferred order, whether by price, date, or availability.

To optimize performance, we have structured the system to handle searches efficiently, ensuring fast response times and scalability. Pagination and filtering mechanisms further enhance the browsing experience. Moving forward, we can integrate AI-driven recommendations, advanced filtering options, and enhanced communication features to make the platform even more user-friendly and intelligent. 

## 5. IMPLEMENTATION AND TESTING

### 5.1. IMPLEMENTATION

The implementation of the rental system involves integrating several core technologies to ensure efficient and scalable operation. The following core technologies were used in the project:

#### 1. Laravel (Backend Framework)

- **Purpose:** Laravel is used for building the backend of the system, handling all server-side logic, API endpoints, and database interactions.
- **Key Features:**
  - Provides easy routing and controller management for API requests.
  - Utilizes **JWT (JSON Web Tokens)** for secure user authentication.
  - Integrates **bcrypt** for secure password hashing.
  - Handles database interactions using **MySQL** with optimized queries for fast access and updates.

#### 2. MySQL (Database)

- **Purpose:** MySQL is used to store all system data, including user profiles, rental items, transactions, and communication logs.
- **Key Features:**
  - **Data Integrity:** Ensures relational integrity with normalized schemas.
  - **Performance:** Optimized for high-volume queries to retrieve rental item listings and transaction history.

#### 3. jQuery (Frontend)

- **Purpose:** jQuery is used to handle dynamic front-end interactions, such as updating the UI with AJAX requests without needing to reload the entire page.
- **Key Features:**
  - Simplifies DOM manipulation for tasks like filtering rental items or displaying dynamic content.
  - Enhances the user experience by making the interface more responsive and faster.

#### 4. JWT (Authentication)

- **Purpose:** JSON Web Tokens (JWT) are used to authenticate users and provide secure access to protected routes.

- **Key Features:**
  - Tokens are generated upon successful login, ensuring only authenticated users can access certain pages.
  - JWTs are stored securely on the client side (in cookies or localStorage).

## 5. bcrypt (Password Hashing)

- **Purpose:** bcrypt is used to securely hash user passwords, ensuring that sensitive data is not stored in plain text.
- **Key Features:**
  - Provides strong encryption for passwords, making them difficult to decrypt even if the database is compromised.

## 6. FonepayDev (Payment Integration)

- **Purpose:** FonepayDev API is integrated for secure online payments.
- **Key Features:**
  - Handles all financial transactions related to rental payments.
  - Ensures PCI-DSS compliance for handling credit card and transaction data securely.

### 5.1.1. TOOLS USED

For the rental system project, a combination of modern tools and technologies were leveraged to create a seamless, scalable, and efficient platform. Below are the tools used:

#### 1. Framework: Laravel

- **Purpose:** Laravel is used for the backend development of the rental system, providing an efficient and elegant framework for routing, database management, and user authentication.
- **Key Features:**
  - **Routing:** Simplifies routing for dynamic web pages.
  - **MVC Architecture:** Utilizes Model-View-Controller (MVC) to separate business logic, making the codebase cleaner and more maintainable.
  - **Authentication:** Provides built-in user authentication using **JWT** for secure login.

#### 2. Database Platform: MySQL

- **Purpose:** MySQL serves as the relational database to store and manage data such as user profiles, rental items, transactions, and communication logs.
- **Key Features:**
  - **Data Integrity:** Ensures data consistency across the system.
  - **Performance:** Optimized queries to retrieve rental items and transaction history efficiently.

#### 3. Frontend Library: jQuery

- **Purpose:** jQuery is used to enhance the front-end experience by enabling asynchronous interactions, such as filtering rental items and updating content dynamically without refreshing the page.
- **Key Features:**
  - **DOM Manipulation:** Simplifies HTML document traversal and manipulation.
  - **AJAX Requests:** Facilitates smooth user interactions by updating content dynamically.

### 5.1.2. IMPLEMENTATION DETAILS OF MODULE

The major modules implemented in the rental system project are described below:

#### Database Schema and MySQL:

The project's database schema is implemented using **MySQL**, with key tables to store data such as **users**, **rental\_items**, **transactions**, **wishlist**, and **messages**. Relationships between these tables are designed to maintain efficient data querying and integrity. For example:

- The **rental\_items** table is linked to the **categories** table to classify rental items.
- **Users** are connected to **rental\_items** for managing rentals and wishlist items.
- **Foreign key constraints** ensure data integrity, such as cascading updates or deletions when an item is removed from the system.

see



FIGURE 11:DATABASE SCHEMA

#### The Backend:

##### 1. User Authentication and Authorization:

- **JWT** is used for secure user authentication, ensuring only logged-in users can access protected routes and perform actions like renting items.
- **bcrypt** is used to securely hash user passwords before storing them in the database.

- Functions like `loginUser` and `registerUser` handle user authentication by validating credentials and issuing tokens.

## 2. Rental Item Management:

- Methods such as `getRentalItems` and `getItemDetails` load rental items based on the selected category and their availability status.
- The system performs dynamic filtering, sorting, and pagination to improve user experience and reduce data load.

## 3. Booking and Availability Tracking:

- Functions like `checkAvailability` are used to verify whether a rental item is available during the user's requested period. If the item is unavailable, the system informs the user and suggests alternatives based on their search history.

## 4. Wishlist Management:

- Users can add rental items to their `wishlist` using the `addToWishlist` function and view their saved items using `viewWishlist`.
- This data is tied to the user account, making it persistent across sessions.

## 5. Admin Management:

- Admin functionalities like user management and rental item listing control are handled by functions such as `manageUsers`, `manageListings`, and `banUser`.
- Admins can review, approve, or remove rental items and manage user accounts.

## 6. Payment and Subscription Management:

- The **FonepayDev API** is used to process payments for rentals. Functions like `processPayment` handle the transaction process, ensuring users are charged appropriately for their rentals.
- Queries like `checkPaymentStatus` confirm successful payments before confirming a booking.

## Challenges and Solutions:

### 1. Performance Issues:

- To address performance issues when querying large datasets, `indexing` was applied to key columns (e.g., `user_id`, `item_id`, `rental_period`). This optimization improved the speed of queries when retrieving rental items and transaction histories.
- Pagination and filtering were implemented to avoid retrieving excessive data and improve response time during browsing.

### 2. Asset Optimization:

- While the system primarily handles text data, optimizing `images and media assets` for rental item listings was essential. Images were compressed and

resized to ensure faster load times, improving user experience, particularly for mobile users.

### 3. Payment Gateway Issues:

- Initially, there were challenges with handling **FonepayDev's webhook events**, leading to discrepancies in payment statuses. Additionally, **irregular server downtimes** on **FonepayDev** caused delays in processing payments, resulting in temporary issues with transaction confirmation and status updates. This was resolved by properly debugging and fine-tuning the webhook handling, as well as implementing fallback mechanisms to handle server downtimes. These adjustments ensured accurate transaction tracking, reliable payment status updates, and better handling of unexpected downtime scenarios.

## 5.2.TESTING

S. N	Test	Required	Activity	Expected Result	Actual Result
1.	Registration	The user should not be present in the session.	Fill in the credentials and request registration.	User registered successfully.	Wait for admin approval.
2.	Login user	Users must be registered and can register if not verified also.	Fill in the login credentials.	Redirect to homepage.	Redirect to homepage.
3.	Adding Category	Must be admin login.	Add the category credentials.	Successfully added.	Success message.
4.	Adding Category Null	Must be admin login	Click submit without credentials	All the fields are required.	Validation message.
5.	Adding Item	Users Must be verified by admin.	Fill the credentials and submit.	Success message.	Item added message.
6.	Update / Delete item	Should be self-added or admin login.	Click the update/ delete button in the dashboard of the user.	Success message.	Success message.
7.	Send verification to Admin for seller access.	Should be logged in to user dashboard	Fill credentials and submit.	Request verification sent.	Request verification sent successfully.

8.	Adding Item with required details null	Users must be verified by admin.	Remove important credentials and submit.	Necessary fields are missing.	Error. All the credentials must be filled.
9.	Request item/contact seller.	Users must be logged.	View contact details and add a message to the message box and click on the mail button.	Message sent successfully.	Mail sent to the seller.
10.	Become premium member	Users must be in the user dashboard.	Click on premium membership.	Redirect to payment page.	Redirect to payment page.
11.	Filter Items	Users can be guests.	Click on map, location, type location, price range, category, item type filter options according to user need.	Show data according to location or any other applied filter.	Fetch the requested data.
12.	Map wise item description.	Users can be guests.	Click on the map hover icon.	Show the detail of the item of the area clicked in the	Show the detail of the item of the area clicked in the

**TABLE 3 : TESTING CASES**

### 5.3.RESULT ANALYSIS

The rental service system incorporates a feature where users can browse and rent items based on availability and their preferences. Users can add items to their wishlist, which allows them to save favorite rentals for easy access later. Upon booking an item, users proceed through a simple checkout process, which includes selecting the rental duration and confirming payment.

The system uses an integrated payment gateway, allowing users to securely complete transactions for rental bookings. If there are any issues with the payment, such as invalid card details, users are notified with an appropriate error message and can retry. Additionally, users are allowed to manage their profiles, update personal information, and track the status of current and past rentals.

The admin panel enables administrators to manage items, approve or reject listings, and review user reports or feedback. This ensures that the system remains organized and efficient. Preliminary testing has shown that users can easily navigate the platform, successfully rent items, and communicate with sellers. The flexibility of item management and the seamless payment process ensure a smooth and user-friendly rental experience.

Agenda	Achieved
Advertisement Add/ Update/ Remove	Yes
News Add/ Update/ Remove	Yes
Item Add/ Update/ Remove	Yes
User verification Add/ Update/ Remove	Yes
History log Add/ View	Yes
Archive Add/ Update/ Remove	Yes
Blacklist Add/ Update/ Remove	Yes
Chatbot save response	Yes
Item view Increment	Yes
Wishlist Add/ Update/ Remove	Yes
Item Request	Yes
Query Add/ update/ remove	Yes
Files Add/ Update/ Remove	Yes
User update/Remove	Yes
Profile update	Yes
Filtering	Yes

**TABLE 4:RESULT ANALYSIS**

## 6.CONCLUSION AND FUTURE RECOMMENDATION

### 6.1. CONCLUSION

This project aimed to improve the rental service experience by developing a platform that facilitates seamless item rental, efficient user management, and secure payment processing. The core objective was to create a user-friendly environment that allows customers to easily browse, rent, and manage their rentals while providing sellers with a streamlined platform to list and manage their items. By utilizing Laravel for backend development, jQuery for dynamic frontend interactions, and MySQL for data management, the project successfully implemented a system that supports rental item listings, user registration, and secure transactions.

The rental management algorithm powers key features, such as item search, booking, and payment handling. Users can browse items, add them to their wishlist, and proceed with a simple checkout process, making the rental experience quick and intuitive. Additionally, the system's design allows users to view their rental history, track current bookings, and manage personal details, ensuring a comprehensive and personalized experience.

This project contributes to the rental service sector, providing a robust platform with flexible features for both customers and sellers. Further improvements can include enhanced search algorithms and additional payment options, along with a mobile-friendly design for wider accessibility.

### 6.2. FUTURE PROGRESS

While the current rental service system provides a solid foundation, future work could address the following areas to enhance its performance and user experience:

Enhanced Search and Recommendation Algorithms: Introducing advanced search filters, recommendation systems, and dynamic pricing based on user preferences and rental history could refine the user experience and make item discovery more personalized.

Performance Optimization: Optimizing database queries and integrating caching mechanisms (e.g., using Redis) could improve the system's responsiveness, especially for data-heavy operations like retrieving rental item listings, processing bookings, and managing large transaction histories.

Expanded Payment Gateway Integration: Although FonepayDev is currently integrated, the system has experienced occasional server downtimes, which have affected payment processing. Future updates should focus on improving payment gateway reliability, possibly integrating backup payment gateways or enhancing error handling to address these issues and ensure smoother transactions.

Scalability and Load Balancing: As the user base grows, implementing load balancing and auto-scaling solutions would ensure the system can handle increased traffic without compromising performance.

User Engagement Features: Adding user reviews, ratings, and recommendation systems could increase engagement and trust between customers and sellers, improving the overall experience.

## 7. REFERENCES

1. Cocola-Gant, A., Jover, J., Carvalho, L., and Chamusca, P. (2021). *Corporate Hosts: The Rise of Professional Management in the Short-Term Rental Industry*. Tourism Management Perspectives, 40, p.100879. <https://doi.org/10.1016/j.tmp.2021.100879>.
2. Ghekiere, A., D'hondt, F., Derous, E., Schelfhout, S., and Verhaeghe, P.-P. (2023). *Reducing Ethnic Discrimination in Rental Applications: The Development of a Training Intervention*. Housing Studies, pp. 1–20. <https://doi.org/10.1080/02673037.2023.2170989>.
3. Peng, Y. and Li, H. (2023). *A Rental Platform Service Supply Chain Network Equilibrium Model Considering Digital Detection Technology Investment and Big Data Marketing*. Sustainability, 15(13), p.9955. <https://doi.org/10.3390/su15139955>.
4. Adams, B., Loewenstein, L., Montag, H., and Verbrugge, R.J. (2022). *Disentangling Rent Index Differences: Data, Methods, and Scope*. Working Paper (22-38). <https://doi.org/10.26509/frbc-wp-202238>.
5. Sittic, G.V. (2015). *The Problem With Rental Marketplaces*. TechCrunch. Available at: <https://techcrunch.com/2015/09/24/the-problem-with-rental-marketplaces/>.
6. National Shelter Policy, 2012 (2068). (2012). *National Shelter Policy, 2012 (2068)*. Ministry of Urban Development, Government of Nepal. Available at: [https://books.google.com.np/books/about/National\\_Shelter\\_Policy\\_2012\\_2068.html?id=rdeOngEACAAJ&redir\\_esc=y](https://books.google.com.np/books/about/National_Shelter_Policy_2012_2068.html?id=rdeOngEACAAJ&redir_esc=y).
7. FATbit Blog. (2020). *11 Challenges Faced by Entrepreneurs When Starting an Online Rental Business – How to Solve Them*. Available at: <https://www.fatbit.com/fab/roadblocks-in-starting-online-rental-business-and-how-to-overcome-them/>.
8. Investopedia. (n.d.). *3 Things to Consider When Renting by the Room*. Available at: <https://www.investopedia.com/articles/personal-finance/102015/3-things-consider-when-renting-room.asp>.
9. Dinnie Muslihat. (2018). *Agile Methodology: An Overview*. Zenkit. Available at: <https://zenkit.com/en/blog/agile-methodology-an-overview/>.
10. PM-Partners. (2021). *What Is Scrum? | The Agile Journey with PM-Partners*. PM Partners. Available at: <https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/>.
11. Asana. (n.d.). *The Hidden Art of Sprint Retrospective Meetings [2022]*. Available at: <https://asana.com/resources/sprint-retrospective#>.
12. Anon. (n.d.). *R1 Rental Workflow – Liberconsult*. Available at: <https://liberconsult.com/home/r1-rental-workflow/>.