A PROJECT REPORT

on

Rentify

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
Computer Science and Engineering

BY

Siddhartha Mukherjee 21052365

Submitted To
Dr. Sarita Mishra

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
November, 2024

A PROJECT REPORT

on

Rentify

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN

Computer Science and Engineering

BY

Siddhartha Mukherjee 21052365

Submitted To

Dr. Sarita Mishra

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

November, 2024

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024

Certificate

This certificate, awarded by **Celebal Technologies**, recognizes the successful completion of a Node.js Summer Internship, during which I developed a full-stack MERN project titled **Rentify**.

Throughout the project, I demonstrated proficiency in building a robust web platform that facilitates property transactions by connecting buyers and sellers directly.

The project allowed me to apply skills in Node.js, Express, MongoDB, and React to create a scalable solution where users can list, search, and contact property sellers with ease.

This certificate acknowledges both my technical and project management skills, showcasing my ability to deliver a comprehensive solution from concept to completion.

CELEBAL TECHNOLOGIES

CERTIFICATE

OF INTERNSHIP

This certificate is proudly awarded to

Siddhartha Mukherjee (CT_CSI_NJ_4667)

In recognition of your outstanding completion of Celebal Summer Internship'2024 Program at Celebal Technologies under **Node JS** for the duration of **03 Jun 2024** to **03 Aug 2024**

You are found to be hardworking, enthusiastic and diligent during your internship. We highly appreciate your accomplishments and wish you the best for the future.

Best Regards,

Sharthak Acharjee

Senior Manager

Acknowledgements

I would like to express my sincere gratitude to **Celebal Technologies** for providing me with the opportunity to be part of the **Celebal Summer Internship (CSI) Program**. The internship was exceptionally structured and detailed, allowing me to gain valuable hands-on experience in backend development, particularly in the domain of Node.js. The weekly assignments and tasks were thoughtfully designed to enhance my technical skills and ensure steady progress throughout the program. I am especially thankful to **Ms. Priya Soni** and **Mr. Sarthak Acharjee**, whose guidance and support as HR coordinators greatly contributed to my successful completion of the internship. Their encouragement and prompt assistance were instrumental in making this journey both productive and enjoyable.

Siddhartha Mukherjee

ABSTRACT

In the current real estate market, potential buyers often face challenges in finding suitable properties due to fragmented information, lack of direct communication channels with sellers, and dependency on intermediaries. Traditional property searching methods typically involve multiple intermediaries, leading to increased costs, delayed communication, and limited access to comprehensive property information. This project addresses these challenges by developing a web-based real estate marketplace platform using the MERN (MongoDB, Express.js, React.js, Node.js) stack.

The platform serves as a centralized solution where property sellers can directly list their properties (houses, villas, and apartments) with detailed information, while potential buyers can efficiently search, and connect with sellers without intermediary intervention. The implementation focuses on creating an intuitive user interface, robust backend architecture, and secure communication channels between parties.

Key features include user authentication, detailed property listings with multimedia support, and advanced search functionality with multiple filters. The platform significantly reduces the time and effort required in property searching by providing comprehensive property information, location-based searches, and immediate connection opportunities with property owners.

Results demonstrate improved efficiency in property discovery and communication, with users reporting easier access to property information and faster initial contact between buyers and sellers. The platform successfully eliminates the need for initial intermediary involvement while providing a transparent and secure environment for real estate transactions.

This solution contributes to the modernization of real estate property discovery and addresses the growing need for direct, efficient communication between property buyers and sellers in the digital age.

Keywords:
Real Estate Platform, MERN Stack, Property Marketplace, Direct Seller-Buyer Communication, Web Application.

Contents

List of Figures

Chapter 1

Introduction

Background

The real estate sector in India has traditionally operated through a network of brokers, property dealers, and word-of-mouth referrals. While this conventional system has served the market for decades, it presents several challenges in today's digital age. Property seekers often spend considerable time visiting multiple real estate offices, scanning through newspapers, or relying on various disconnected online listings to find suitable properties. Similarly, property sellers face difficulties in reaching potential buyers directly and often depend heavily on intermediaries, leading to increased costs and extended selling periods.

Current Market Challenges:

The traditional real estate market faces several key challenges:

1. **Information Asymmetry**: Buyers often lack access to comprehensive and accurate property information, making decision-making difficult.
2. **Intermediary Dependency**: The necessity of brokers and agents increases transaction costs and can lead to delayed communication.
3. **Limited Reach**: Sellers struggle to showcase their properties to a wider audience, often restricting themselves to local markets.
4. **Time Consumption**: The process of property searching and selling involves multiple physical visits and repeated communications through various channels.
5. **Lack of Standardization**: Property listings across different platforms lack uniformity in information presentation, making comparisons difficult.

Project Significance

This project aims to revolutionize the traditional real estate market by developing a comprehensive online platform that bridges the gap between property sellers and buyers. The platform leverages modern web technologies to create a user-friendly, efficient, and transparent marketplace for real estate transactions.

Chapter 2

Basic Concepts

This section serves as a foundational pillar, encompassing the fundamental concepts and essential knowledge that are imperative for a comprehensive understanding of the project. It delves into the intricate details of the related tools and techniques employed throughout the development process, shedding light on their significance and the pivotal roles they play in shaping the final outcome of the project.

2.1 React JS:

React.js is a powerful JavaScript library developed by Facebook for building user interfaces. In our real estate platform, React.js handles the frontend development, allowing us to create reusable UI components and manage the application state efficiently. The component-based architecture of React helps in breaking down complex UI elements into smaller, manageable pieces, making the code more maintainable and scalable. React's virtual DOM ensures optimal rendering performance, which is crucial for smooth user experience when browsing through multiple property listings.

2.2 Node.js:

Node.js is a JavaScript runtime environment that enables server-side execution of JavaScript code. In our project, Node.js powers the backend server, handling HTTP requests, processing data, and managing business logic. Its event-driven, non-blocking I/O model makes it particularly efficient for handling multiple concurrent connections, which is essential for a real estate platform where multiple users may be accessing property listings simultaneously.

2.3 MongoDB:

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. For our real estate platform, MongoDB provides a scalable and flexible database solution that can easily handle varied property listings with different attributes and characteristics. Its document-based structure is perfect for storing complex property data including images, amenities, and location details. MongoDB's query capabilities also enable efficient searching and filtering of properties based on multiple criteria.

2.4 Express JS:

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for building web applications. In our project, Express.js simplifies the creation of the backend API, handling routes, middleware integration, and HTTP requests. It serves as the middleware between our frontend React application and the MongoDB database, managing data flow and API endpoints efficiently.

2.5 Tailwind CSS:

Tailwind CSS is a utility-first CSS framework that enables rapid UI development through pre-built classes. In our real estate platform, Tailwind CSS provides a modern, responsive design system without the need for writing custom CSS. Its utility classes allow for quick styling and consistent design implementation across all components, ensuring a professional and responsive user interface across different devices and screen sizes.

2.6 Cloudinary:

Cloudinary is a cloud-based media management solution used in our project for handling property images and media files. It provides efficient upload, storage, optimization, and delivery of media assets. Cloudinary automatically handles image resizing, format optimization, and delivery through CDN, ensuring fast loading times for property images regardless of the user's location. This is crucial for a real estate platform where high-quality images are essential for property showcasing.

Each of these technologies plays a vital role in creating a robust, scalable, and user-friendly real estate platform. Their integration enables efficient development, smooth user experience, and reliable data management, making the platform suitable for handling numerous property listings and user interactions.

Chapter-3

Problem Statement

The real estate market in India continues to rely heavily on traditional methods of property buying and selling, creating significant inefficiencies in the process. Property sellers, particularly individual homeowners, face challenges in reaching potential buyers without depending on multiple intermediaries. This dependency not only increases the cost of property transactions but also extends the time taken to complete deals. Buyers, on the other hand, struggle with fragmented information spread across various platforms and sources, making it difficult to make informed decisions. The lack of standardization in property information presentation further complicates the comparison process.

The current system's reliance on physical visits and paper-based documentation creates unnecessary delays and increases the overall cost of property transactions. Additionally, the absence of a centralized, reliable platform for property listings makes it challenging for buyers to access comprehensive property information and for sellers to showcase their properties effectively to a wider audience.

3.1 Project Planning

1. Requirements Gathering:

Understanding the needs of all stakeholders is crucial for developing a successful real estate platform. Through extensive market research, user interviews, and analysis of existing solutions, we identified key requirements from different user perspectives.

Property sellers need a straightforward way to list their properties with detailed information and images. They require tools to manage their listings, update property status, with potential buyers. Individual sellers need guidance on pricing their properties and understanding market trends.

Property buyers seek reliable, detailed information about properties, including accurate locations, high-quality images, and transparent pricing. They need efficient search and filter mechanisms to find properties matching their specific requirements. The ability to save favourite properties, compare different options, and directly communicate with sellers is essential for making informed decisions.

2. Technology Feasibility Assessment:

The selection of appropriate technologies is crucial for building a scalable and maintainable platform. Our technology stack has been chosen after careful consideration of various factors including development efficiency, scalability requirements, and long-term maintenance.

The MERN (MongoDB, Express.js, React.js, Node.js) stack was selected as the primary technology stack due to its robust ecosystem and proven reliability in building modern web applications. React.js provides an efficient way to build interactive user interfaces with reusable components, making it ideal for handling dynamic property listings and search results. Its virtual DOM implementation ensures optimal performance even when dealing with large lists of properties.

MongoDB's flexible document-based structure is particularly suitable for storing property listings with varying attributes. Its querying capabilities enable efficient implementation of search and filter features, while also providing scalability as the platform grows. The combination of Node.js and Express.js creates a robust backend capable of handling multiple concurrent requests efficiently.

For media management, Cloudinary was chosen as it provides comprehensive image handling capabilities including automatic optimization, responsive delivery, and reliable storage. This is crucial for a real estate platform where high-quality images are essential for property showcasing.

3. Feature Specification and Acceptance Criteria:

The platform's features have been carefully defined to address the specific needs of both buyers and sellers while ensuring platform usability and efficiency.

The user authentication system incorporates registration options of including email/password. This system includes robust security measures such as email verification, and storing of password using various hashing algorithms, and protection against unauthorized access.

Property listing features are designed to make it easy for sellers to showcase their properties effectively. The listing process is broken down into logical steps, guiding sellers through information input while ensuring all necessary details are captured. Image upload capabilities are provided with automatic optimization for web delivery.

Search functionality is implemented with basic filtering options. Users can search by location, price range, property type, and various amenities. The search interface is designed to be intuitive while providing powerful filtering capabilities. Results are displayed with relevant information prominently shown, making it easy for buyers to quickly assess properties.

3.2 Project Analysis

Market analysis reveals a growing trend towards digital property search and transaction platforms. With increasing internet penetration and smartphone usage, particularly in urban areas, there is a significant opportunity for digital real estate platforms. The target market includes both individual property owners and real estate agencies, with a primary focus on residential properties. The competitive landscape includes several established property portals, but there remains room for innovation in terms of user experience and feature offerings. Our platform differentiates itself through its focus on direct buyer-seller communication, standardized property information presentation, and user-friendly interface.

1. **Technical Analysis**:

   The platform's architecture is designed to ensure scalability, maintainability, and performance. The frontend is built using a component-based architecture in React.js, promoting code reusability and maintainability. State management is handled through react-hooks, providing a predictable state container for managing application data.

   The backend implements a RESTful API architecture, with clearly defined endpoints for different functionalities. Database design follows MongoDB best practices, with appropriate indexing for optimizing search operations. The system includes caching mechanisms to improve performance for frequently accessed data.

2. **Risk Analysis**:

   Several potential risks have been identified and assessed. Technical risks include challenges in handling large volumes of image data, ensuring system performance under high load, and maintaining data security. These are addressed through implementation of content delivery networks, load balancing, and robust security measures.

   Operational risks primarily revolve around user adoption and data quality. To mitigate these, the platform includes features for verifying property information, user ratings and reviews, and regular data quality checks.

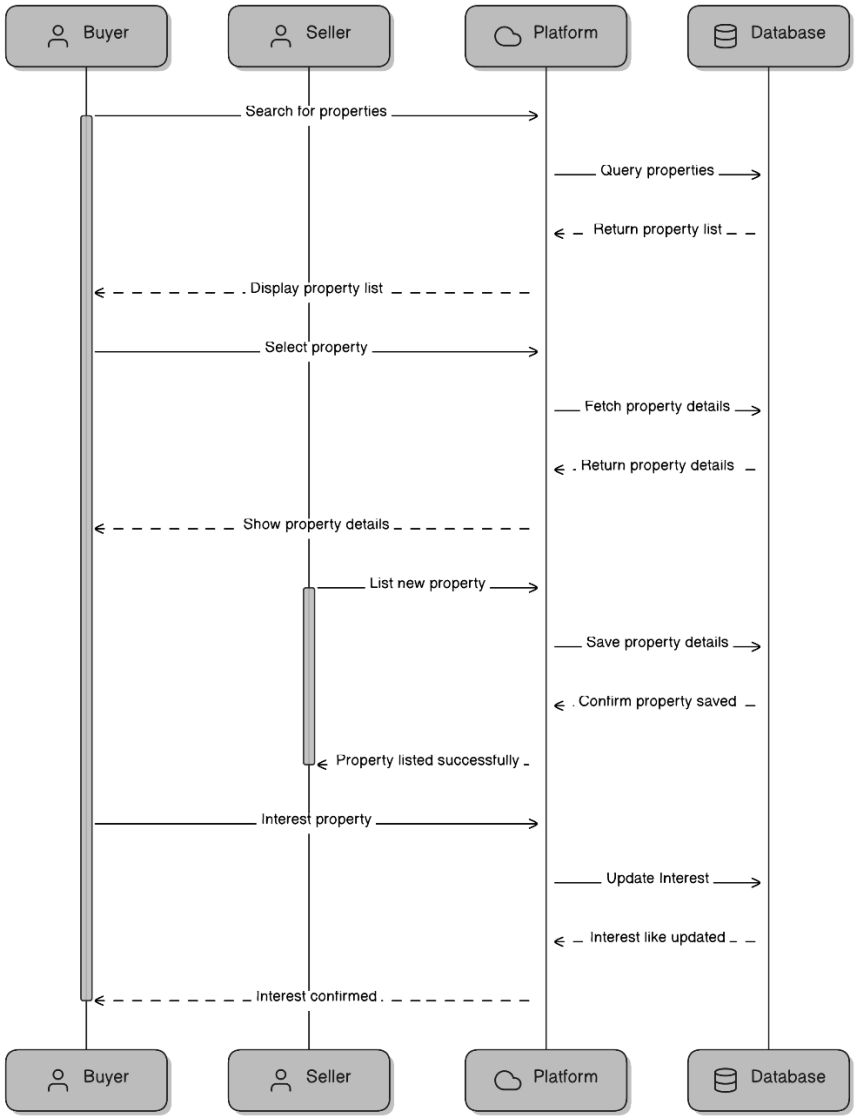
Marketing strategies are planned to drive user adoption and build trust in the platform.

3.3 System Design:

The system architecture follows a three-tier design with clear separation of concerns between the presentation, business logic, and data layers. The frontend implements a responsive design that works seamlessly across devices of different sizes. The backend is structured to handle authentication, data validation, and business logic processing efficiently.

Security is implemented at multiple levels, including secure communication protocols, data encryption, and access control mechanisms. The database design includes appropriate indexing strategies for optimal query performance and data integrity constraints to maintain data quality.

3.3.1 UML Sequence Diagrams:



Chapter 4

Implementation

For the implementation of the project, the following methodology was adopted:

4.1 Methodology:

- **Agile Development**: The project followed an agile development methodology, with iterative development cycles known as sprints. This approach allowed for flexibility in responding to changing requirements and enabled continuous improvement based on feedback received throughout the development process.

- **Feature Prioritization**: Features were prioritized based on user needs, project objectives, and technical feasibility. I team identified core functionalities such as listing properties, uploading properties, and managing properties as the primary focus areas for initial implementation. Additional features, were planned for subsequent iterations based on their importance and complexity.

- **Technology Stack Selection**: The chosen technology stack was carefully evaluated to ensure compatibility with project requirements and alignment with development and my expertise. React was selected for frontend development due to its ability to scalable single page applications with native-like performance. Node.js and Express were utilized for backend development to provide a lightweight and efficient server-side runtime environment.

- **Collaborative Development**: The development adopted a collaborative approach, with regular commits and checkpoints were established to facilitate collaboration and coordination. GitHub was used as the primary version control system, allowing me to manage on code, track changes, and manage project tasks using features such as issues and pull requests.

4.2 Testing Plan

After the completion of the project implementation, the following testing and verification plan was executed:

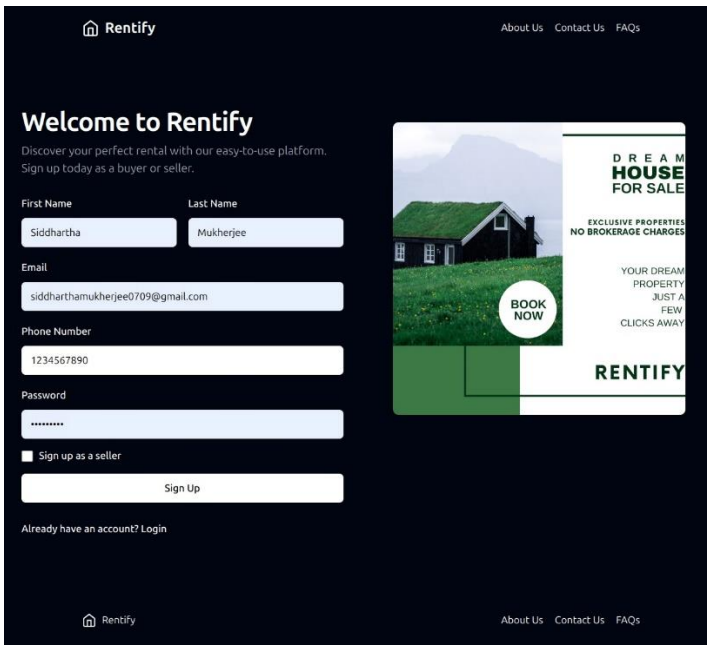
- **Unit Testing:** Unit tests were written for individual components and functionalities of the application, ensuring that each part of the system behaved as expected.

- **Integration Testing:** Integration tests were conducted to verify the interactions between different components and services, ensuring seamless communication and data flow.

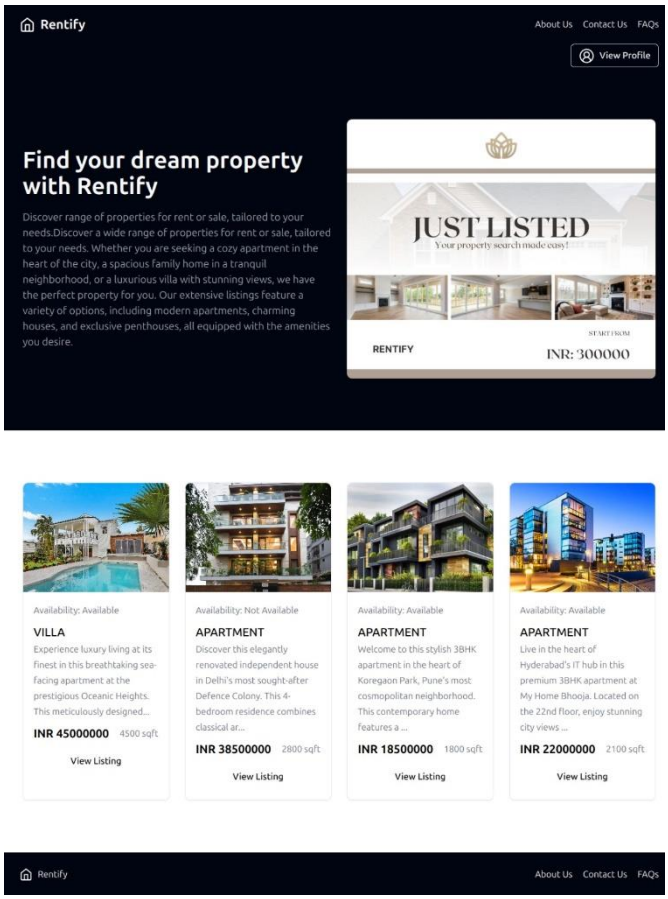
- **Validation Testing:** Various validation tests, including functional, usability testing was carried out, so that the application's features and functionalities were rigorously assessed to ensure they met the specified requirements and provided a seamless user experience.

- **Performance Testing:** Performance tests were conducted to assess the responsiveness and scalability of the application under different loads and usage scenarios. This ensured that the application could handle concurrent user activity and maintain acceptable performance levels.

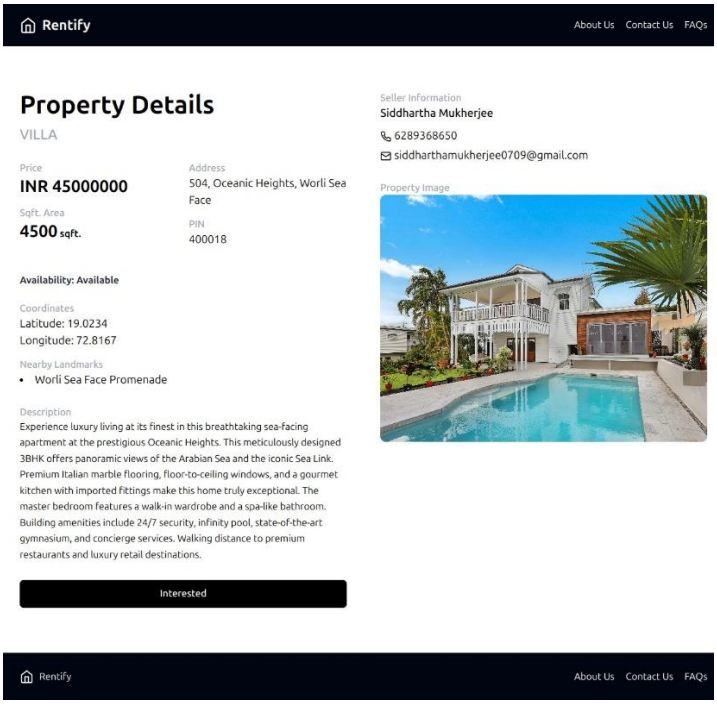
4.3 User Interface Screenshots:



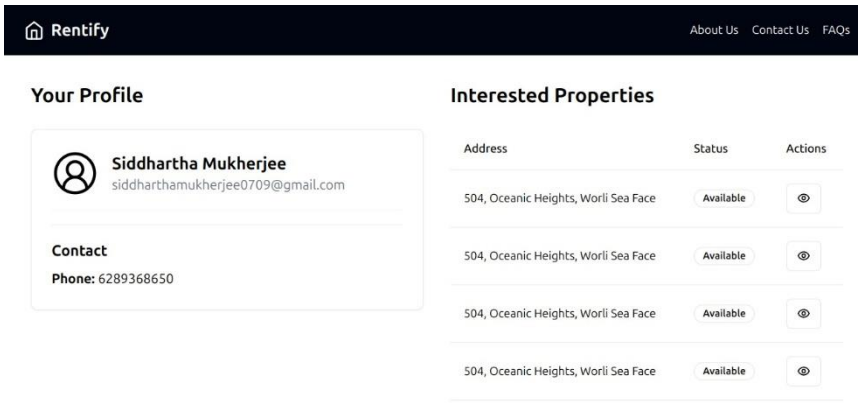
Buyer/Seller Signup UI



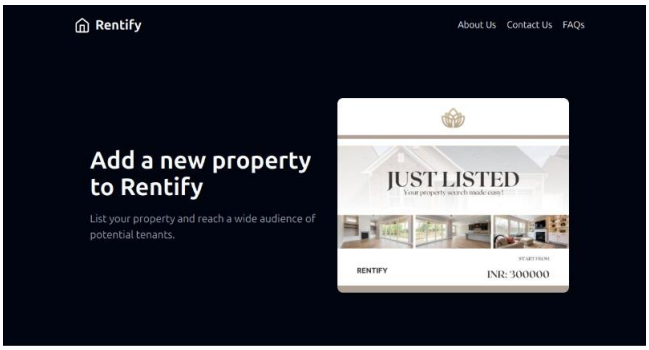
Property Listing UI

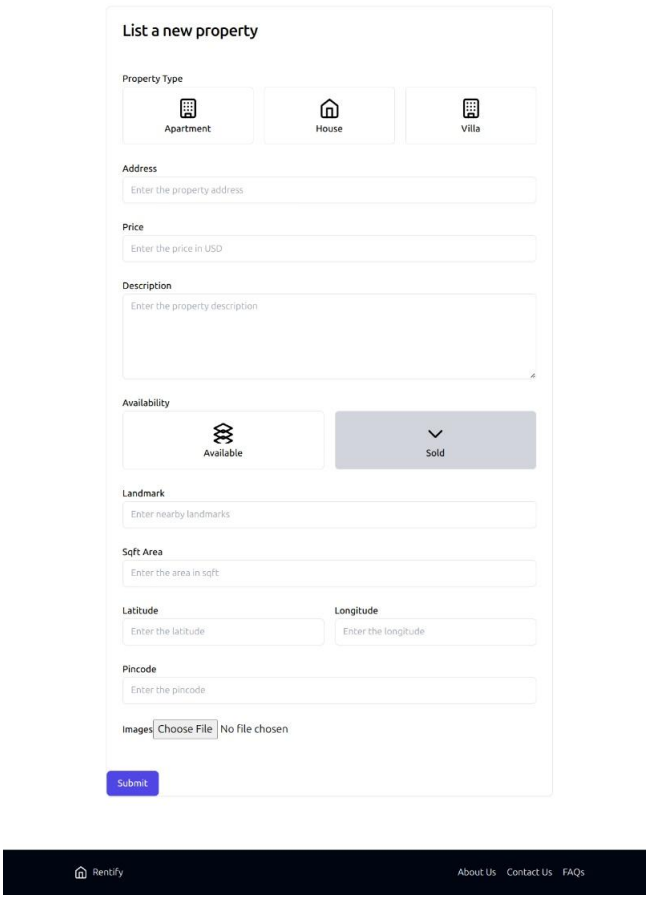


Property Details UI

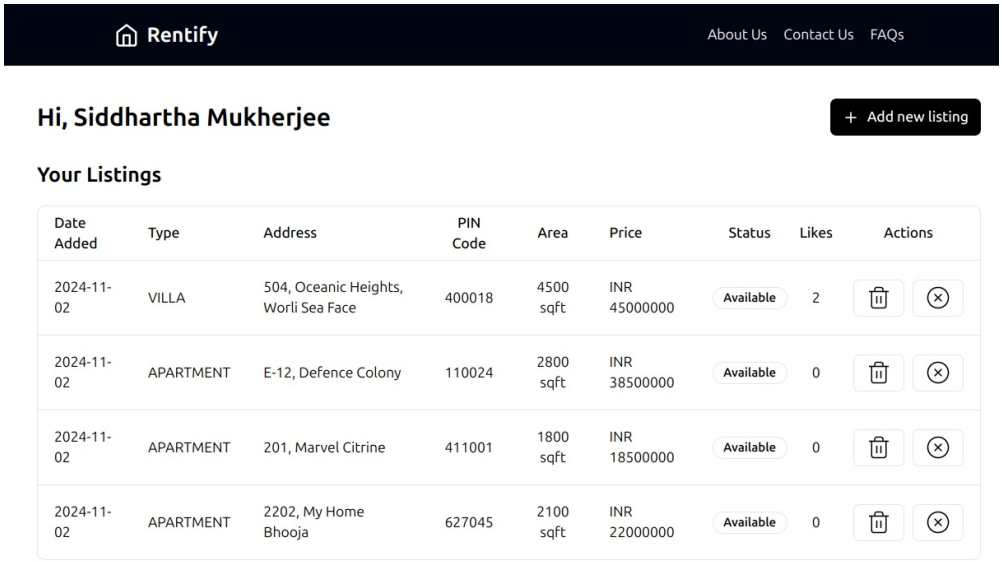


Buyer's Interested Properties UI





Seller's Portal for Adding New Property



Seller's Portal for Managing Properties

Chapter 5

Standards Adopted

5.1 Design Standards:

In adherence to established design standards, our project followed a systematic approach to architecting the software system. We employed the following recommended practices:

- **Utilization of UML Diagrams**: Unified Modelling Language (UML) diagrams were extensively used to visualize and document the system architecture, including use case diagrams, class diagrams. These diagrams provided a clear understanding of the system's structure, behaviour, and interactions.

- **Modular Design Approach**: The system was designed using a modular approach, with complex components broken down into smaller, manageable modules. This modular design facilitated reusability, maintainability, and scalability, allowing for easier management of the codebase.

- **Scalability Considerations**: During the design phase, we meticulously assessed scalability requirements, ensuring that the system was structured to seamlessly accommodate future expansion in user base, data volume, and functionality. We implemented scalable architecture patterns, including utilization of cloud computing services were deemed appropriate, to fortify the system's ability to grow and evolve.

5.2 Coding Standards:

The project adhered to established coding standards to ensure consistency, readability, and maintainability of the codebase. Key coding standards followed include:

- **Meaningful Naming Conventions**: Descriptive and meaningful names were used for variables, functions, classes, and other identifiers, following consistent naming conventions such as camelCase.

- **Structured Code Organization**: Code was organized into logical blocks and modules, with clear separation of concerns were employed to promote code modularity and maintainability.

- **Clear Code Readability**: Code was written with clarity and conciseness in mind, avoiding overly complex or convoluted logic. Comments and documentation were utilized to explain the purpose of code blocks, algorithms, and complex operations.

- **Consistent Indentation and Formatting**: Consistent indentation and formatting styles were applied throughout the codebase to enhance readability. Industry-standard formatting guidelines were followed, and code linters were utilized to enforce consistent coding styles.

- **Robust Error Handling**: Robust error handling mechanisms were implemented to gracefully handle exceptions, errors, and edge cases. Appropriate error codes, logging, and exception handling techniques were employed to aid in troubleshooting and debugging.

5.3 Testing Standards:

The project followed established testing standards to ensure the reliability, functionality, and quality of the software product. Key testing standards observed include:

- **Documentation as per IEEE 829**: Test documentation was prepared in accordance with IEEE 829 standard, outlining the format and content of various test documents, including test plans, test cases, and test reports.

- **Comprehensive Test Coverage**: Comprehensive test coverage was achieved through a combination of unit tests, integration tests, system tests, and acceptance tests.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion:

This real estate platform project successfully addresses the fundamental challenges in the traditional property buying and selling process by creating a digital ecosystem that connects buyers and sellers directly. The implementation of the MERN stack, combined with modern tools like Tailwind CSS and Cloudinary, has resulted in a robust, scalable, and user-friendly platform that effectively streamlines real estate transactions.

Key Achievements

1. **User Empowerment** The platform has successfully eliminated the mandatory dependency on intermediaries by providing tools for direct communication between buyers and sellers. Users can now make informed decisions based on comprehensive property information, high-quality images, and accurate location data.
2. **Digital Transformation** The project represents a significant step towards digitalizing real estate transactions in India. By providing a standardized format for property listings and implementing advanced search capabilities, the platform has made property discovery and comparison more efficient and transparent.
3. **Technical Implementation** The successful integration of various technologies demonstrates the platform's technical robustness:

6.2 Future Scope:

The real estate platform has significant potential for future enhancement and expansion. Here are the key areas identified for future development:

1. Advanced Technology Integration,

Virtual Reality (VR) Integration

- Implementation of virtual property tours
- 360-degree property views
- Interactive virtual walkthrough experiences
- VR-based property showcasing

Artificial Intelligence Features

- AI-powered property recommendations
- Price prediction algorithms
- Automated property valuation
- Chatbots for instant query resolution
- Image recognition for property feature detection

2. Enhanced User Features

Mobile Application Development

- Native mobile apps for iOS and Android
- Push notifications for property updates
- Location-based property alerts
- Offline property viewing capability

Financial Integration

- Online payment gateway integration
- EMI calculators
- Loan pre-approval integration
- Property insurance integration
- Real-time market rate analysis

References:

The following references were taken into considerations while preparing the project:

1. MongoDB Documentation: https://mongoosejs.com/docs/
2. Node JS Documentation: https://nodejs.org/docs/latest/api/
3. Express JS Documentation: https://expressjs.com/
4. React JS Documentation: https://react.dev/learn

These references provided valuable information and guidance during the development of the project.