

Property Rental: A Web Application

BY

Mahmud Reza Chowdhury Ani

ID: 201-15-3090

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Mr. Saiful Islam

Assistant Professor

Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

Mr. Afjal Hossan Sarower

Lecturer (Senior scale)

Department of Computer Science and Engineering
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

June 2024

APPROVAL

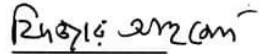
This Project titled "Property Rental: A Web Application", submitted by Mahmud Reza Chowdhury Ani, ID No: 201-15-3090 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on July 13, 2024

BOARD OF EXAMINERS



Dr. Md. Taimur Ahad (MTA)
Associate Professor & Associate Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Board Chairman



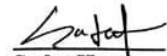
Dr. Fizar Ahmed (FZA)
Associate Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner 1



Mr. Rahmatul Kabir Rasel Sarkar (RKR)
Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner 2



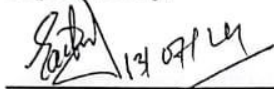
Sadat Hasah
Data Scientist (Senior Principal Officer)
Risk Management Division
BRAC Bank

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Saiful Islam, Assistant Professor, Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

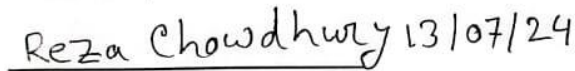


Mr. Saiful Islam
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Mr. Afjal Hossan Sarower
Lecturer (Senior scale)
Department of CSE
Daffodil International University

Submitted by:



Mahmud Reza Chowdhury Ani
ID: 201-15-3090
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty for His divine blessing making it possible for us to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Mr. Saiful Islam, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Web Design and Development*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Sheak Rashed Haider Noori Head of the Department of CSE**, for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

This web application, known as "Property Rental," is designed to simplify the process of finding and renting accommodation for students enrolled at DIU (Daffodil International University). Tailored to meet the specific needs of student renters seeking cost-effective housing solutions, the platform offers a user-friendly interface catering to both property owners and tenants. Landlords have the capability to register, list their properties, and manage their listings, while renters can utilize the platform to search for available properties, schedule viewings, and submit rental inquiries. By centralizing property management tasks and facilitating seamless interactions between landlords and tenants, "Property Rental" optimizes the rental experience for students, fostering efficient communication and enhancing accessibility to housing options within the DIU community.

TABLE OF CONTENTS

CONTENTS	PAGE NO
Approval	ii
Declaration	iii
Acknowledgement	iv
Abstract	v
Table of contents	vi
List of figures	viii
 CHAPTER 1: INTRODUCTION	 1-5
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Description of the system	3-4
1.5 Expected Outcomes	4-5
1.6 Summary	5
 CHAPTER 2: BACKGROUND	 6-9
2.1 Preliminaries	6
2.2 Related Work	6-7
2.3 Scope of the Problem	7-8
2.4 Challenges	8-9
 CHAPTER 3: REQUIREMENT SPECIFICATION	 10-21
3.1 Business Process Modeling	10
3.2 Requirement Collection and Analysis	11
3.3 Functional Requirements	11-12
3.4 Non-Functional Requirements	13-14
3.5 Use Case Diagram	14-18
 <i>©Daffodil International University, Bangladesh</i>	 vi

3.6 Entity Relationship Diagram	19-21
CHAPTER 4: DESIGN SPECIFICATION	22-25
4.1 Front-end Design	22-23
4.2 Back-end Design	23-25
CHAPTER 5: IMPLEMENTATION AND TESTING	26-33
5.1 Front-End Implementation	26
5.2 Back-End Implementation	26-27
5.3 Pages	27-31
5.4 Feature Testing	31-33
CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	34-36
6.1 Impact on Society	34-35
6.2 Environmental Impact	35
6.3 Sustainability	36
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	31-32
7.1 Conclusion	37
7.2 Future Scope	37-39
REFERENCES	40
APPENDIX	41-47
PLAGARISM REPORT	48

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1.1: Business Process Modelling	10
Figure 3.5.1: Use Case Diagram	15
Figure 3.6.1: ER Diagram of “Property Rental”	19
Figure 5.3.1: Home page of “Property Rental”	27
Figure 5.3.2: Login page of “Property Rental”	28
Figure 5.3.3: Landlord Profile page of “Property Rental”	29
Figure 5.3.4: Admin Profile page of “Property Rental”	29
Figure 5.3.5: Navigation Bar of “Property Rental”	30
Figure 5.3.6: Add Property Page of “Property Rental”	31

CHAPTER 1

Introduction

1.1 Introduction

In the realm of higher education, securing suitable accommodation can often pose a significant challenge for students, particularly those enrolled at institutions such as DIU (Daffodil International University). Recognizing the importance of accessible and affordable housing options, we introduce "Property Rental," a web application dedicated to simplifying the process of finding and leasing properties in the vicinity of Universities in Ashulia, Savar.

With "Property Rental," students gain access to a convenient platform for exploring available rental properties. Each property listing provides comprehensive details, including information about landlords and property specifics. Moreover, the application facilitates direct communication between tenants and landlords, streamlining the rental inquiry process.

1.2 Motivation

The development of "Property Rental" is driven by profound understanding of the challenges students face in finding rental accommodations near university areas. Many students, especially newcomers to the area face challenges navigating the local rental market. Students often struggle with accessibility, affordability, and convenience when searching for suitable housing. Traditional methods such as classified ads or word-of-mouth, can be cumbersome and time-consuming. By offering a user-friendly platform with comprehensive listings and direct communication with landlords, "Property Rental" aims to simplify the rental experience for students. Our motivation lies in empowering students with a reliable solution that meets their housing needs, ultimately enhancing their well-being and academic success.

1.3 Objective

The primary goal of "Property Rental" is to develop a user-friendly platform that simplifies the process of finding and leasing rental properties. This platform aims to provide easy access to a diverse selection of rental accommodations, tailored to the unique preferences and budgetary constraints of students and tenants. It achieves this through a wide range of filters and search options, enabling users to quickly locate properties that suit their specific needs.

Transparency is a crucial aspect, with the platform offering accurate and current information about properties, landlords, rental terms, and amenities. Each listing features detailed descriptions, high-quality images, virtual tours, and reviews from previous tenants. This thorough approach ensures users have all the necessary information to make well-informed decisions, fostering trust and reliability in the rental process.

Convenience is a top priority, achieved by streamlining the rental experience with intuitive search functionalities, comprehensive property listings, and direct communication channels with landlords. The platform includes features like online scheduling for property viewings and communication through direct contact links. These elements are designed to save time and reduce the hassle typically associated with renting a property.

Affordability is also a key focus, as the platform aims to help students find housing options that fit within their financial limits. By presenting a variety of price points and detailed cost breakdowns, the platform assists users in finding affordable options without sacrificing necessary amenities.

Ultimately, the platform seeks to enhance the overall rental experience, supporting students' well-being and academic success by providing a dependable and efficient solution for securing rental accommodations. Committed to continuous improvement, the platform regularly updates its features based on user feedback and market trends. This dedication ensures it remains a valuable resource for students seeking rental properties, promoting a stress-free and enjoyable living experience.

1.4 Description of the system

The "Property Rental" system is a web application built using Flutter and ExpressJS that facilitate the process of finding and leasing rental properties for renters and students.

User Roles:

1.Tenants/Renters: Users seeking rental accommodations.

2. Landlords: Property owners interested in renting out their properties.

3.Admin: Administrators responsible for managing the platform.

Key Features:

Property Listings: Landlords can register and list their properties, including details such as number of rooms, location, amenities, and rental prices.

Direct Communication: Tenants can view property listings and contact landlords for rental inquiries and further details.

User Authentication: Secure login functionality for landlords, and admin users

Admin Dashboard: Admin panel for managing user accounts, property listings, and verifying landlord information.

Search and Filter: Advanced search and filtering options for tenants to find properties based on their preferences.

Documentation Management: Storage and verification of property documents by admin before listing.

Data Management:

Property Data: Storage of property details, landlord information, and rental terms.

User Data: Secure storage of user accounts, including login credentials and contact information.

Document Verification: Admin verifies property documents submitted by landlords before listing properties on the platform.

User Interaction:

Property Viewing Requests: Tenants can view property listings and contact landlords directly for rental inquiries and confirmation.

Rental Inquiries: Tenants can directly message landlords for rental inquiries and additional information.

User Interface:

Intuitive Design: User-friendly interface accessible to tenants, landlords, and admin users.

Responsive Layout: Ensures compatibility across devices, allowing users to access the platform from desktops, tablets, and smartphones.

Security Measures:

User Authentication: Secure login process with password encryption to protect user accounts.

Document Verification: Admin verifies property documents to ensure authenticity before listing properties on the platform.

1.5 Expected Outcomes:

The anticipated outcomes of "Property Rental" encompass:

1. Increased accessibility to rental accommodations, enabling students/tenants to locate suitable housing options more effectively.
2. Improved transparency in the rental process, fostering trust between tenants and landlords and facilitating smoother rental transactions.
3. Simplified search functionalities and comprehensive property listings, leading to a more intuitive and streamlined rental search experience for students.
4. Expanded availability of affordable housing options that align with students' budget constraints while offering essential amenities and proximity to campus.
5. Elevated overall satisfaction and well-being among students, as they benefit from a dependable and user-friendly solution for finding rental accommodations tailored to their preferences.

1.6 Summary:

The goal of the project is developing a robust house rental web application using Flutter, designed to cater to renters, landlords. It aims to provide an intuitive and user-friendly platform where students and renters can find suitable property according to their needs. Landlords, on the other hand, can add properties on the platform ensuring reliability.

The web application also strives to promote affordability by offering budget friendly housing specially targeted towards student needs and helping users find suitable housing options within their financial means.

Overall, the project aims to enhance the rental experience by providing a reliable and efficient solution for both tenants and landlords. It emphasizes usability, security, and efficiency to create a seamless rental process that supports users in finding and leasing properties that meet their needs and preferences.

CHAPTER 2

Background

2.1 Preliminary

The "Property Rental" web application targets the housing needs of tenants, specifically aimed at students seeking accommodation. Utilizing a user-friendly interface and comprehensive features, the application simplifies the process of discovering and leasing rental properties. Developed using Flutter, tenants can efficiently search for properties tailored to their requirements through advanced filtering options. Landlords are empowered to add property details, which are meticulously verified by administrators upon submission of requisite documentation. Once tenants select a property of interest, they can directly communicate with landlords for property viewings and subsequent rental arrangements using the contact information provided within the property listing. This approach ensures a secure and convenient rental process, eliminating the necessity for physically scouting properties in rural areas.

2.2 Related Work

Property rental websites are very common in Bangladesh but their uses are mostly limited to residential properties around the upper areas of Dhaka. These websites mostly target people in need of family size apartments or people with high salary paying jobs. Because of this, the rental houses are very costly and not ideal for students who are looking for accommodation.

Here are a few related websites of house rental system:

1.Bproperty: Bproperty is a leading real estate marketplace in Bangladesh, offering a

comprehensive platform for buying, selling, and renting properties. With a user-friendly interface and extensive listings, Bproperty connects property seekers with a wide range of residential and commercial properties across the country.

2.Rents.com.bd: Rents.com.bd is a Bangladeshi online platform focused on rentals, specifically targeting Dhaka. They function as a dedicated rental portal, connecting landlords with tenants seeking apartments, houses, or office spaces.

3.The Tolet: Tolet.com is a Bangladeshi online platform concentrated on real estate rentals, particularly in Dhaka. They cater to various rental needs, including flats, messes, sublets, hostels, offices, shops, showrooms, and even garages. Listings also mention features like attached baths, balconies, and availability for families or bachelors.

These Web sites provide a snapshot of the features and business models of some notable platforms in the legal tech space. Studying these aspects can offer valuable insights as we shape and refine the features and strategies for our "Property Rental" project.

2.3 Scope of the problem

Understanding the scope of the problem that the "Property Rental" project aims to solve is crucial for defining its objectives and designing effective solutions. Several key aspects encompass this scope:

1. Limited Availability: Many students face challenges in finding rental properties that align with their preferences, such as location, amenities, and affordability.

2. Time and Effort: Traditional methods of searching for rental accommodations can be time-consuming and inefficient, requiring significant effort to navigate the local rental market.

3. Transparency Issues: Existing rental solutions often lack transparency, causing

uncertainty about rental terms, property conditions, and the reliability of landlords.

4. Communication Barriers: Inadequate communication channels between tenants and landlords can complicate the rental process, making it difficult for students to obtain necessary information or arrange property viewings.

Addressing these challenges is pivotal for the "Property Rental" project. By developing the web application, the goal is to streamline the rental process, enhance transparency, and improve communication between tenants and landlords. This approach aims to offer students a more efficient and reliable way to find suitable rental accommodations that meet their needs and preferences.

2.4 Challenges

Launching and sustaining a project like the "Property Rental" web application comes with several potential challenges that must be recognized and addressed early in the development process to ensure its success and effectiveness:

Data Accuracy and Verification: Ensuring the accuracy and authenticity of property listings and landlord information is critical. Robust verification processes are necessary to prevent fraudulent or misleading listings, maintaining trust and reliability among users.

User Engagement: Encouraging active participation from tenants and landlords on the platform is a challenge. Effective strategies are needed to attract users and demonstrate the benefits of using the application, such as comprehensive property listings, user reviews, and convenient communication tools.

Security and Privacy: Safeguarding user data and ensuring secure communication channels between tenants and landlords are paramount. Implementing strong security

measures protects sensitive information and complies with privacy regulations, fostering trust among users.

Platform Scalability: Designing the application to handle potential growth in user base and property listings while maintaining performance and usability is essential. Scalability considerations ensure the platform can expand seamlessly without compromising user experience.

Legal and Regulatory Compliance: Adhering to legal and regulatory requirements, such as rental agreements, property regulations, and data protection laws, is crucial. Thorough research and implementation of compliance measures ensure the platform operates within legal boundaries and safeguards user rights.

User Experience Design: Creating an intuitive and user-friendly interface that meets the diverse needs of tenants and landlords is vital. Effective user experience design, informed by user feedback and iterative testing, enhances usability and satisfaction, driving adoption and retention.

Addressing these challenges effectively during the development and deployment phases will contribute to the functionality, reliability, and usability of the "Property Rental" web application. By tackling these issues proactively, the platform can provide a seamless and trustworthy rental experience for all stakeholders involved.

CHAPTER 3

Requirement Specification

3.1 Business Process Modeling

Business process modeling can illustrate the entire workflow between tenants and landlords. Landlords begin by registering and logging into the platform. After completing their profile, they can submit their property for listing. The admin then reviews the submission and has the authority to approve or reject the property.

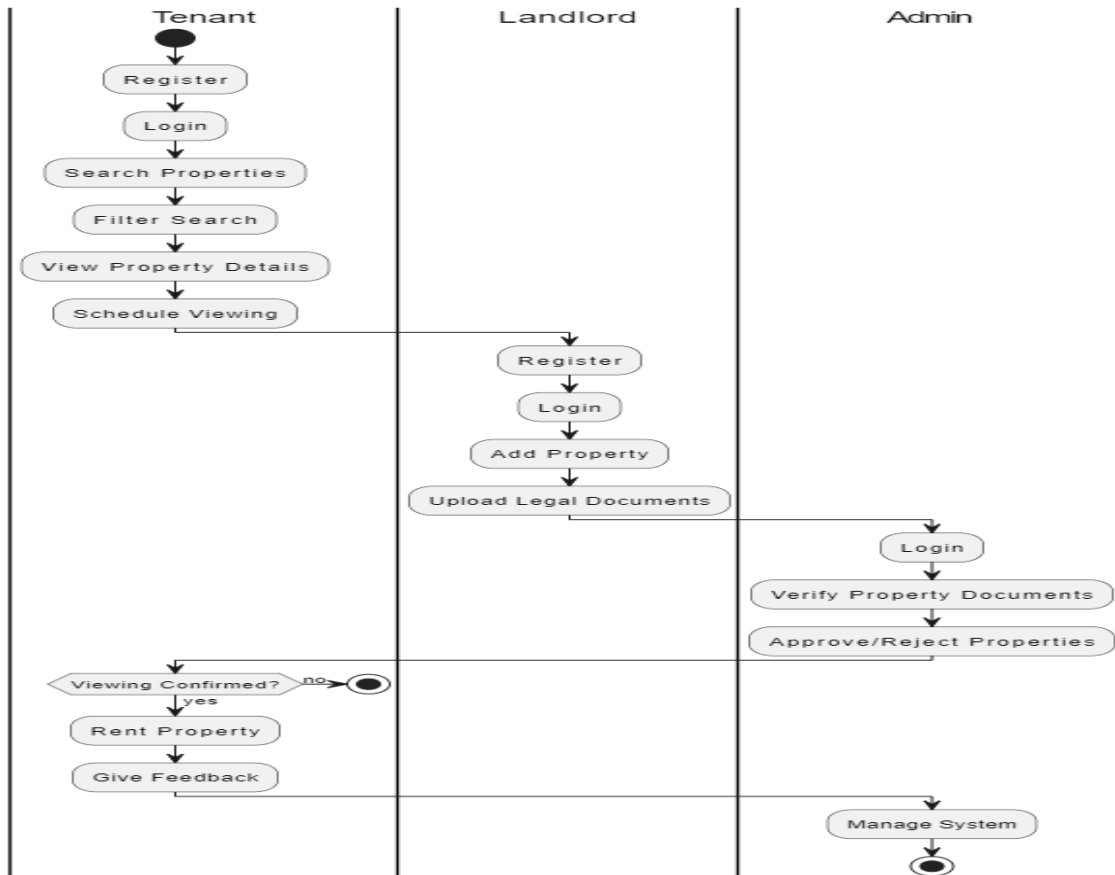


Figure 3.1.1: Business Process Modelling

3.2 Requirement Collection and Analysis

The process of requirement collection and analysis is crucial in understanding what the "Property Rental" web application must achieve to satisfy the needs of its users; students, landlords, and administrators. This phase involves gathering, analyzing, and specifying the necessary requirements to guide the design and development of the system.

3.3 Functional Requirements:

1. User Registration and Authentication:

Users (tenants, landlords, administrators) should be able to register securely with required details such as name, contact information, email, and national identification where applicable.

Authentication mechanisms should ensure secure login and session management.

2. Property Search and Filtering:

Tenants should have the ability to search for rental properties based on criteria such as location, number of bedrooms, bathrooms, rent amount, and amenities.

Advanced filtering options should be provided to refine search results effectively.

3. Property Listings Management:

Landlords must be able to add and manage property listings, including detailed information like address, description, bedrooms, bathrooms, kitchen details, rent amount, and amenities.

Legal documents related to properties uploaded by landlords should undergo verification by administrators.

4. Viewing and Rental Process:

Tenants should be able to schedule property viewings through the platform.

The rental process should enable tenants to communicate directly with landlords, inquire about properties, and finalize rental agreements.

5. User Profiles:

Each user type (tenant, landlord, administrator) should have a profile page displaying relevant information such as personal details, rented properties (for tenants), and owned properties (for landlords).

6. Feedback and Rating System:

Tenants should be allowed to provide feedback and ratings for properties and landlords.

Ratings and reviews should be visible on property listings to assist future tenants in decision-making.

7. Administrative Dashboard:

Administrators should have access to a dashboard for managing user accounts, verifying property listings, approving/rejecting properties based on legal documentation, and handling reported issues.

3.4 Non-Functional Requirements:

1. Performance:

The application must be responsive and capable of handling simultaneous user interactions efficiently.

Property search and listing retrieval processes should be optimized for speed, even with large datasets.

2. Security:

Robust security measures must be implemented to protect user data and ensure secure communication channels between tenants and landlords.

Data encryption should be applied to sensitive information such as passwords and personal details.

3. Reliability:

The platform should maintain high availability with minimal downtime, supported by backup and recovery mechanisms to safeguard against data loss.

4. Usability:

The user interface should be intuitive, user-friendly, and accessible to all users, including those with disabilities.

Navigation should be straightforward, and the design should prioritize a positive user experience.

5. Scalability:

The application architecture should support scalability to accommodate growth in user base and property listings without compromising performance.

Scalability measures should ensure seamless expansion and responsiveness under increased load.

6. Compliance:

The platform must adhere to relevant legal and regulatory requirements concerning property rental, data protection, and user privacy.

Compliance with standards and regulations should be maintained throughout the application's lifecycle.

3.5 Use Case Diagram

Property Rental is a Flutter web application aimed at finding a property for a student under proper accommodation, authenticating landlords, communicating with each other, helps clarify the functionalities required for each type of user and outlines how these users will interact with the system. It provides a clear, structured approach to understanding user requirements and how the system should behave in response to user actions

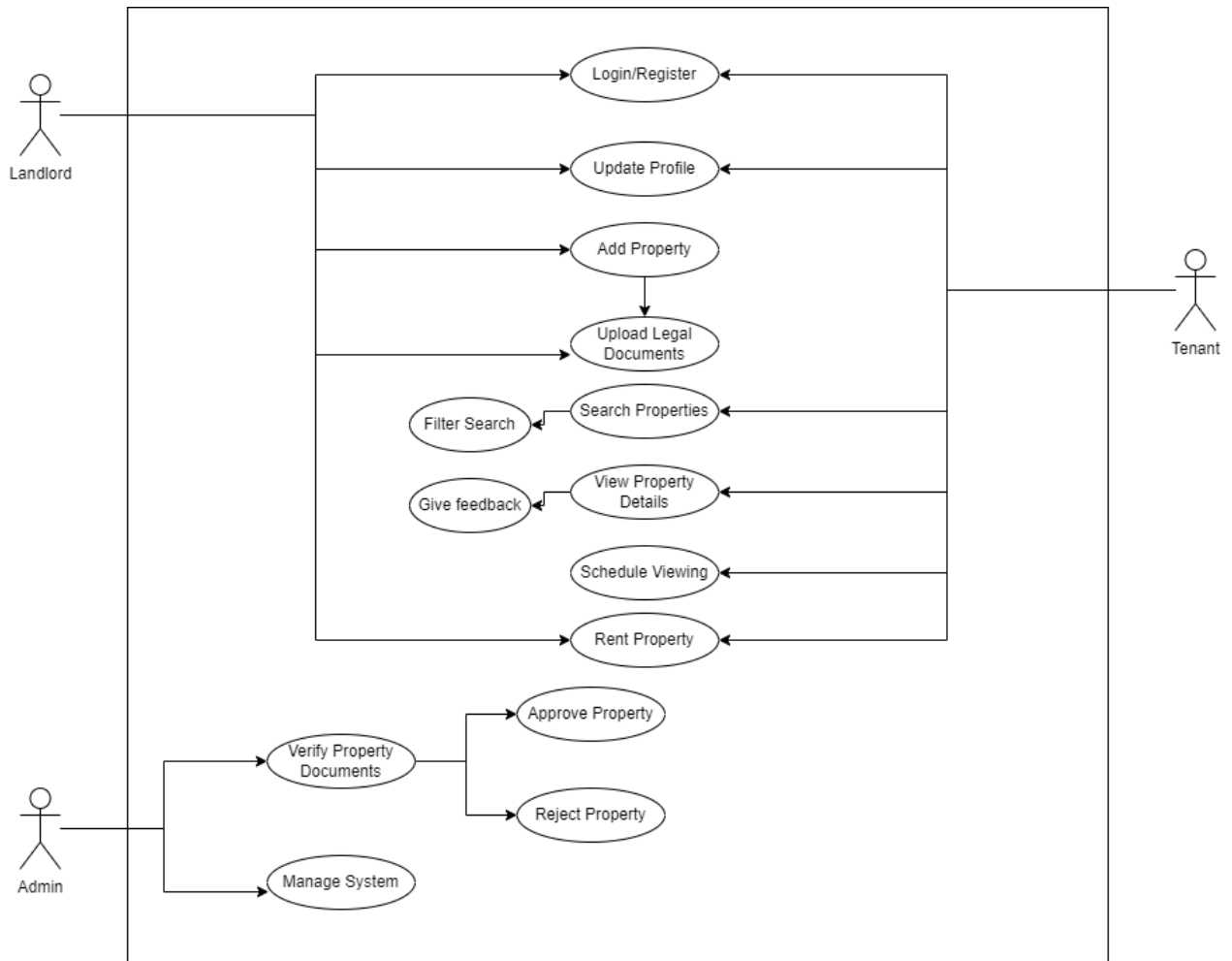


Figure 3.5.1: Use Case Diagram.

Description for Use Case is given below:

Actors:

1. **Tenant:** An individual looking to find and lease rental properties.
2. **Landlord:** An individual looking to list properties for rent.
3. **Admin:** An individual responsible for overseeing the system, verifying documents, and approving or rejecting property listings.

Use Cases:

Register:

1. **Actors:** Tenant, Landlord
2. **Description:** Allows tenants and landlords to create an account by providing necessary details such as name, contact information, email, and national identification.

Login:

1. **Actors:** Tenant, Landlord
2. **Description:** Enables tenants and landlords to securely log in to their accounts.

Search Properties:

1. **Actor:** Tenant
2. **Description:** Enables tenants to search for available rental properties.

Filter Search:

1. **Actor:** Tenant
2. **Description:** Provides tenants with options to refine their search results based on criteria such as location, number of bedrooms, bathrooms, rent amount, and amenities.

View Property Details:

1. **Actor:** Tenant
2. **Description:** Allows tenants to view detailed information about a property, including descriptions, images, number of bedrooms, bathrooms, kitchen details, rent amount, and amenities.

Schedule Viewing:

1. **Actor:** Tenant
2. **Description:** Allows tenants to schedule an appointment to view a property they are interested in.

Rent Property:

1. **Actor:** Tenant
2. **Description:** Facilitates the process for tenants to rent a property after viewing it, including communication with the landlord and finalizing rental agreements.

Update Profile:

1. **Actors:** Tenant, Landlord
2. **Description:** Allows tenants and landlords to update their profile information such as contact details, email, and other personal information.

Add Property:

1. **Actor:** Landlord
2. **Description:** Enables landlords to add new property listings to the platform.

Upload Legal Documents:

1. **Actor:** Landlord
2. **Description:** Requires landlords to upload legal documents for their properties to ensure authenticity and legality.

Approve Property:

1. **Actor:** Admin
2. **Description:** Admin approves property listings after verifying the uploaded legal documents.

Reject Property:

1. **Actor:** Admin
2. **Description:** Admin rejects property listings if the verification of legal documents fails or if there are issues with the property.

Verify Property Documents:

1. **Actor:** Admin
2. **Description:** Admin verifies the legal documents uploaded by landlords for their properties.

Manage System:

1. **Actor:** Admin
2. **Description:** Admin manages the overall system, including user accounts, property listings, and handling reported issues.

Relationships:

1. **Tenant** interacts with use cases: Register, Login, Search Properties, Filter Search, View Property Details, Schedule Viewing, Rent Property, Update Profile.
2. **Landlord** interacts with use cases: Register, Login, Add Property, Upload Legal Documents, Update Profile.
3. **Admin** interacts with use cases: Approve Property, Reject Property, Verify Property Documents, Manage System.
4. **Add Property** includes **Upload Legal Documents**.
5. **Verify Property Documents** includes both **Approve Property** and **Reject Property**.

3.6 Entity Relationship Diagram

The entity-relationship diagram (ERD) for the property rental web application captures the key entities and their relationships within the application. The main entities include admin, landlord, tenant and property. The ER chart of our framework is given underneath:

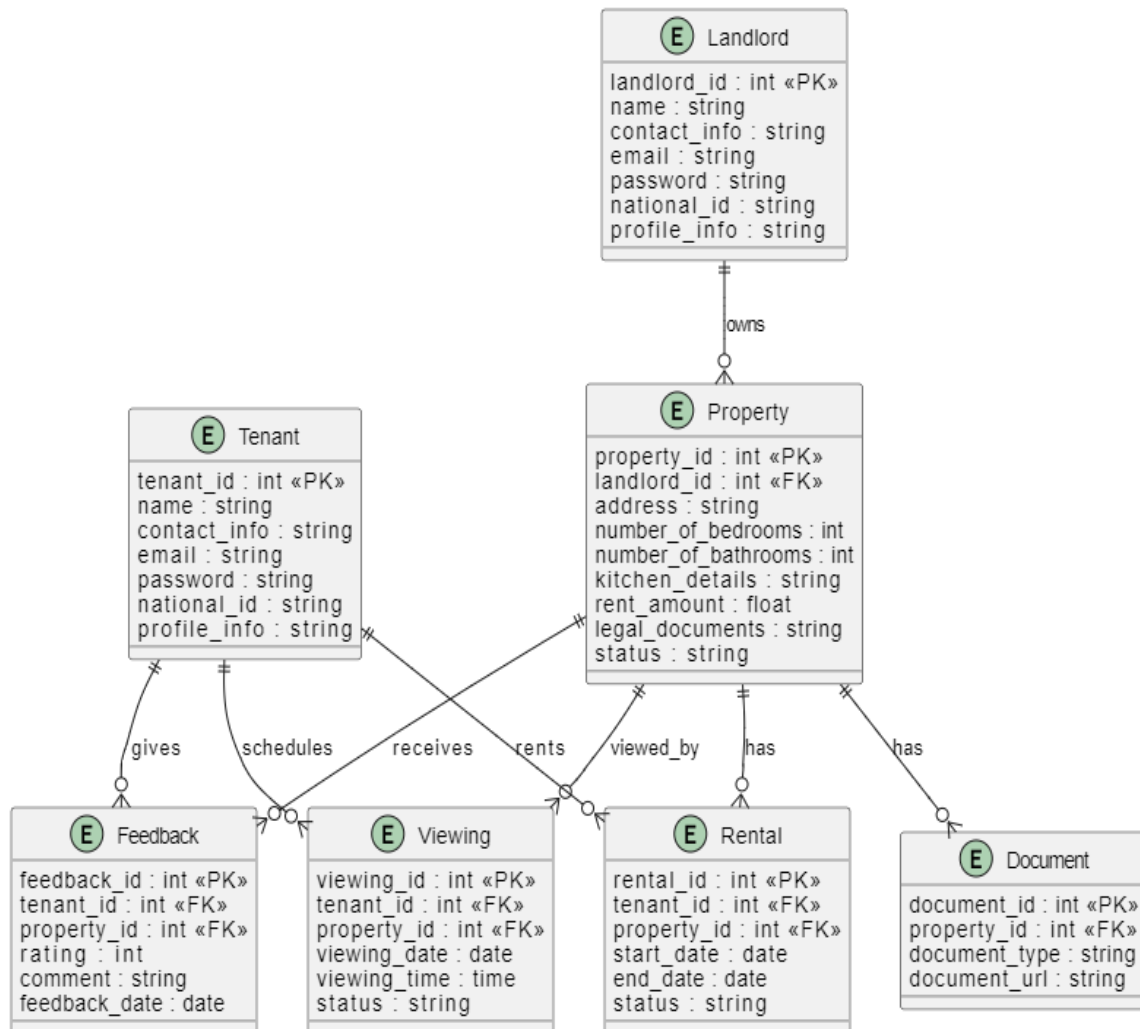


Figure 3.6.1: ER Diagram of “Property Rent

Relationship between keys and attributes of “Property Rental” is given below:

Relationship	Primary Key (PK)	Foreign Key (FK)	Description
Landlord - Property	landlord_id	landlord_id in Property	A landlord can own multiple properties. The landlord_id in the Property entity links back to the landlord_id in the Landlord entity.
Tenant - Rental	tenant_id	tenant_id in Rental	A tenant can rent multiple properties over time. The tenant_id in the Rental entity links back to the tenant_id in the Tenant entity.
Tenant - Viewing	tenant_id	tenant_id in Viewing	A tenant can schedule multiple viewings. The tenant_id in the Viewing entity links back to the tenant_id in the Tenant entity.
Tenant - Feedback	tenant_id	tenant_id in Feedback	A tenant can give multiple feedback entries. The tenant_id in the Feedback entity links back to the tenant_id in the Tenant entity.
Property - Rental	property_id	property_id in Rental	A property can have multiple rental agreements. The property_id in the Rental entity links back to the property_id in the Property entity.
Property - Viewing	property_id	property_id in Viewing	A property can be viewed by multiple tenants. The property_id in the Viewing entity links back to the property_id in the Property entity.

Relationship	Primary Key (PK)	Foreign Key (FK)	Description
Property - Document	property_id	property_id in Document	A property can have multiple associated documents. The property_id in the Document entity links back to the property_id in the Property entity.
Property - Feedback	property_id	property_id in Feedback	A property can receive multiple feedback entries. The property_id in the Feedback entity links back to the property_id in the Property entity.

This table illustrates the relationships within the "Property Rental" system. Landlords can own multiple properties, tenants can rent multiple properties over time, and tenants can also schedule multiple viewings and provide feedback on properties. Each property can have several rental agreements, viewings by different tenants, associated documents, and feedback entries. The relationships are defined through primary and foreign keys linking these entities.

CHAPTER 4

Design Specification

4.1 Front-end Design

The front-end of the "Property Rental" web application is designed using Flutter, a popular UI toolkit known for its ability to create natively compiled applications for mobile, web, and desktop from a single codebase. The use of Flutter ensures a seamless and responsive user experience across different devices and platforms.

User Interface:

1. User Registration and Login:

1. **Screens:** The application includes distinct registration and login screens for tenants and landlords.
2. **Forms:** Forms for user input validation, including fields for name, email, password, contact information, and national ID.
3. **User Experience:** Simple and intuitive layout to ensure a smooth onboarding process.

2. Property Listings:

1. **Layout:** A list view of available properties with filters for criteria such as location, number of bedrooms, rent amount, and property type.
2. **Details:** Each listing shows a brief summary of the property, including a thumbnail image, rent amount, and key features.

3. Property Details:

1. **Display:** A detailed view of selected properties showing images, number of bedrooms, bathrooms, kitchen details, rent amount, landlord information, and contact details.
2. **Actions:** Options to schedule a viewing or contact the landlord directly.

4. Profile Management:

1. **Tenant Profile:** Allows tenants to view and update their personal

information, rental history, and feedback given.

2. **Landlord Profile:** Provides landlords with the ability to manage their property listings, view inquiries, and update their profile information.

5. Navigation:

1. **Drawer Menu:** A side navigation drawer for easy access to different sections of the application, such as home, profile, property listings, and settings.
2. **Bottom Navigation Bar:** A persistent navigation bar at the bottom for quick access to frequently used features.

4.2 Back-end Design

The back-end of the "Property Rental" web application is developed using a combination of modern technologies to ensure scalability, performance, and security. The stack includes TypeScript, Node.js, Express.js, MongoDB, Mongoose, Vercel, Cloudinary, and Firebase.

Server-Side Logic:

1. Language and Framework:

1. **TypeScript:** Ensures type safety and easier maintenance of the codebase.
2. **Node.js:** Utilized for its capability to handle asynchronous operations and build a scalable server.
3. **Express.js:** Provides a robust framework for building RESTful APIs, handling routing, middleware, and HTTP requests.

2. Database:

1. **MongoDB:** A NoSQL database chosen for its flexibility in handling unstructured data, scalability, and ease of integration with JavaScript-based technologies.
2. **Mongoose:** An ODM (Object Data Modeling) library for MongoDB and Node.js, used for schema definition, data validation, and business logic.

3. **API Design:**

1. **Endpoints:** RESTful API endpoints for various functionalities such as user registration, login, property management, scheduling viewings, and feedback.
2. **Authentication:** JWT (JSON Web Tokens) for secure authentication and authorization.
3. **Data Validation:** Middleware for validating incoming data to ensure integrity and consistency.

4. **File Handling:**

1. **Image Upload:** Cloudinary is used for efficient image storage, transformation, and delivery.
2. **Video and File Upload:** Firebase handles video and other file uploads, providing secure storage and easy access.

5. **Deployment:**

1. **Server:** The back-end is deployed on Vercel, offering fast and scalable serverless deployment, continuous integration, and automated deployments.

6. **Data Flow:**

1. **User Actions:** User interactions from the front-end are sent to the back-end API, which processes the requests and interacts with the database.
2. **Responses:** The back-end returns appropriate responses, including data retrieval, error messages, or success confirmations, which are then rendered on the front-end.

Integrations:

1. **Cloudinary:**

1. **Purpose:** Used for storing and managing images uploaded by landlords when adding property details.
2. **Integration:** Direct integration with the back-end to handle image uploads and transformations efficiently.

2. **Firebase:**

1. **Purpose:** Handles video uploads and file storage for property documentation.

2. **Integration:** Seamlessly integrated with the back-end to ensure secure and efficient handling of large files.

4. Security

1. Authentication and Authorization:

1. **JWT Tokens:** Secure authentication mechanisms to ensure only authorized users can access and modify data.
2. **Password Encryption:** Secure storage of user passwords using hashing algorithms.

2. Data Validation:

1. **Middleware:** Validation of incoming requests to protect against malicious inputs and ensure data integrity.

3. Access Control:

1. **Role-Based Access:** Different access levels for tenants, landlords, and admins to control permissions and data access.

The design of the "Property Rental" web application combines a powerful front-end using Flutter with a robust back-end powered by TypeScript, Express.js, and MongoDB. The integration of Cloudinary and Firebase ensures efficient handling of images, videos, and other files. This design provides a secure, scalable, and user-friendly platform that meets the needs of both tenants and landlords, making the rental process more accessible and streamlined.

CHAPTER 5

Implementation and Testing

5.1 Front-End Implementation

The front-end implementation of the "Property Rental" web application is developed using Flutter, ensuring a responsive and visually appealing user interface across both web and mobile platforms. This involves creating various screens for user registration, login, property listings, property details, profile management, and communication between tenants and landlords. Custom Flutter widgets are employed to ensure consistency and reusability throughout the application. State management is handled using the Provider package, which efficiently manages application state and user interactions. Flutter's Navigator is utilized to manage screen transitions and user navigation. Integration with back-end APIs is achieved through HTTP requests to fetch and display data, while real-time updates for property availability and user actions are implemented using WebSockets or Firebase Cloud Messaging.

5.2 Back-End Implementation

The back-end of the "Property Rental" web application is developed using TypeScript for enhanced type safety and Node.js, leveraging Express.js to build a RESTful API. The back-end architecture adopts a modular design, with distinct modules dedicated to user management, property management, rental processing, and document handling. MongoDB is employed as the database, organized to store user profiles, property listings, rental agreements, and user feedback. Mongoose is utilized for schema definition, data validation, and seamless interaction with the database.

Authentication and authorization are secured through JSON Web Tokens (JWT), with role-based access control ensuring appropriate permissions for tenants, landlords, and administrators. For handling media uploads, the application integrates Cloudinary for

images and Firebase for video and file storage. The back-end is deployed on Vercel, providing a scalable and serverless deployment environment. Additionally, CI/CD pipelines are configured to automate testing and deployment processes, ensuring continuous integration and delivery.

5.3 Pages

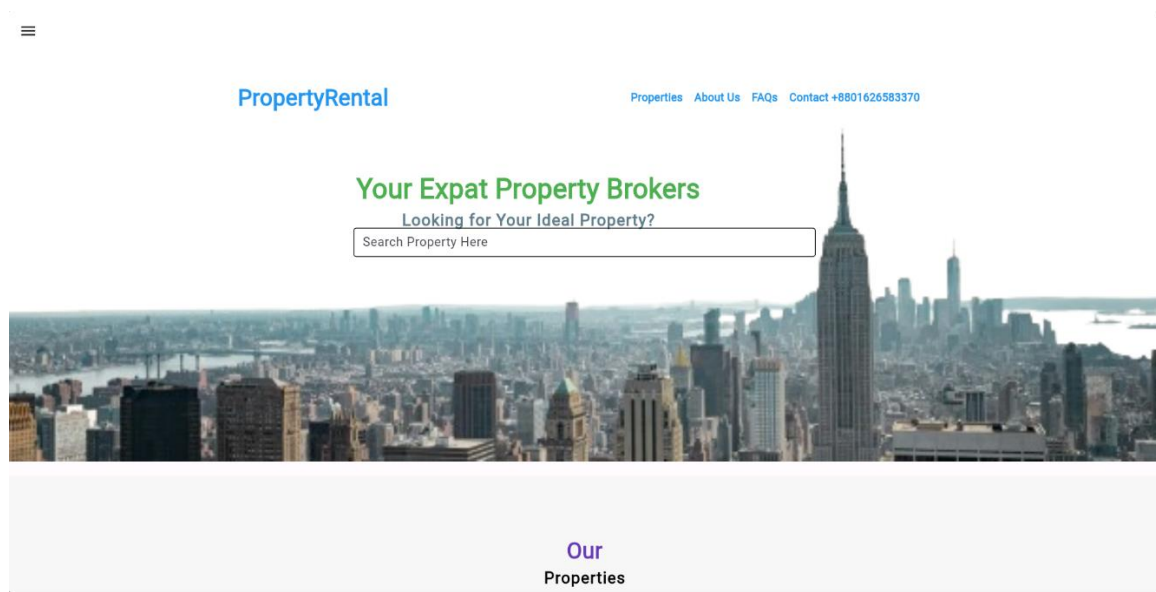


Figure 5.3.1: Home page of “Property Rental”

This is the home page of Property Rental. If anyone wants to access the system then he/she must go through the landing page.

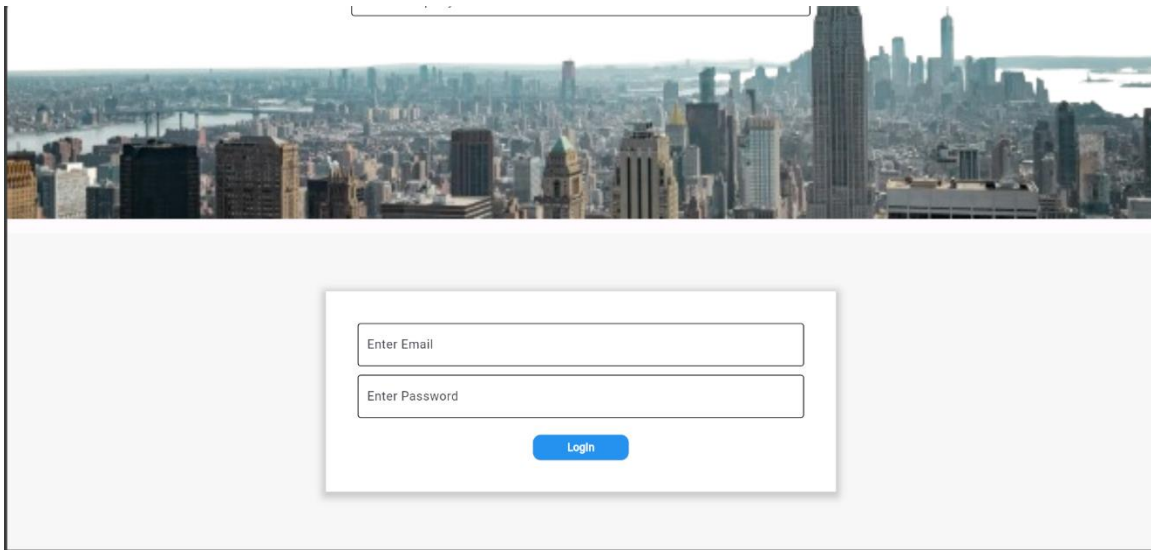


Figure 5.3.2: Login page of “Property Rental”

In this section we can see the login page for both landlords and admin.

The image shows a user profile verification page. At the top center, there is a blue circular button with a white person icon and the text 'Upload Your Image'. Below this button, a red error message reads: 'Your profile is not verified yet! Please verify your profile to add properties.' To the right of this message is a grey button labeled 'Editable Mode'. Below the error message, there are five stacked input fields: 'Enter Your Name', 'Say something about you' (a larger text area), 'Enter Your Mobile Number', 'Enter Your WhatsApp Number', and 'Enter Your Office Mobile Number'. At the bottom left of the form area, the text 'Your Email' is visible.

Figure 5.3.3: Landlord Profile page of “Property Rental”

In this section, Landlords can view and update their profile with necessary details. Landlord profile needs to be completed to submit properties.

Figure 5.3.4: Admin Profile page of “Property Rental”

This section allows admins to view and update their profiles with required details.

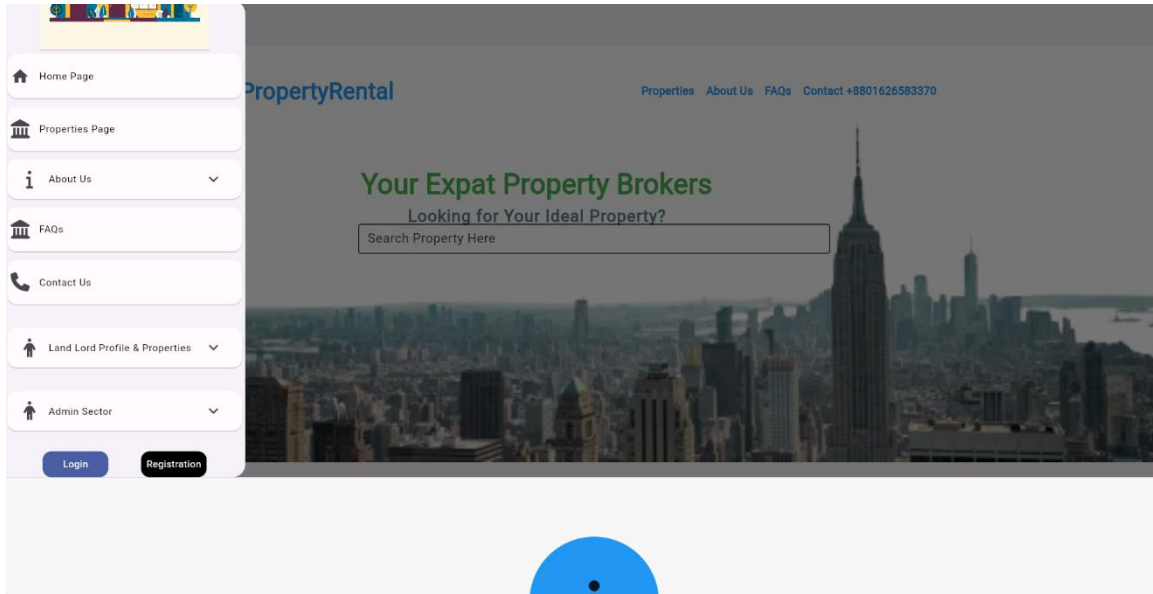


Figure 5.3.5: Navigation Bar of “Property Rental”

The image shows a form for adding a new property listing. At the top is a light blue rectangular area with a camera icon and the text 'Upload minimum 10 images'. Below this are several input fields and selection options. The first is a text field labeled 'What is the property name?' with the placeholder 'Property Name'. The second is a text field labeled 'How much is the rent?' with the placeholder 'Property Price'. Below these, it shows 'Property Uploading Date 2024-06-29'. Then, under 'Select property status', there are two radio buttons: 'Rented' (which is selected) and 'Available'. Finally, under 'Select property type', there are two radio buttons: 'Flat' (which is selected) and 'Home'.

How many bathrooms?

Property Bathrooms

Total Property Area

Property Area

Say something about your property

Property Bio

Property Location Google Map Link

Property location google map link

Property Documents

Property Video Link

Upload Property

Figure 5.3.6: Add Property Page of “Property Rental”

5.4 Feature Testing

The "Property Rental" platform is a web application designed to connect students with landlords, facilitating the process of finding and renting accommodations. Thorough feature testing is essential to ensure a seamless user experience and the proper functioning of this platform. Here are some crucial feature tests to consider:

1. User Registration and Authentication

Validate the user registration process for both students and landlords.

Confirm that users can log in using their credentials without any issues.

2. Property Upload and Management

Test the property listing process to ensure landlords can provide titles, descriptions, pricing, and upload photos and videos.

Ensure landlords can categorize and tag their properties by type (e.g., apartment, house, dorm).

3. Property Search and Filters

Verify that students can search for properties using keywords, location, and property type.
Test the filtering options for rent amount, number of bedrooms, bathrooms, and amenities.

4. User Profile Management

Test the creation and updating of user profiles for both students and landlords.
Verify that users can access their listed properties, bio, profile pictures, and favorite properties.

5. User Interaction

Ensure that students can schedule viewings, send inquiries, and communicate with landlords.
Test the ability of users to follow or unfollow landlords and receive updates on their dashboard.

6. Notification System

Test notifications for new messages, scheduled viewings, and property updates.
Ensure that email notifications are sent as required.

7. Security Measures

Conduct vulnerability assessments to identify potential security issues such as cross-site scripting or SQL injection.
Verify that user information and passwords are encrypted and stored securely.

8. Responsive Design

Test the platform's usability on various devices and screen sizes.
Ensure the user interface adapts properly for tablet and mobile devices.

9. Performance Testing

Evaluate the application's response time under different loads to ensure it can handle multiple users simultaneously.

Identify and optimize any slow-loading pages or components.

10. Accessibility Compliance

Conduct accessibility testing to ensure the platform meets WCAG standards and is usable by people with disabilities.

11. Browser Compatibility

Test the application on different web browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure consistent functionality.

12. Data Backup and Recovery

Ensure that user-generated data, such as profiles, properties, and messages, are regularly backed up and recoverable in case of data loss.

Verify that users can report inappropriate or offensive content.

13. User Reporting and Support

Ensure users can report inappropriate content or contact support for assistance.

14. API Testing with Postman

Test all API endpoints using Postman to ensure they return the correct responses and handle various scenarios (e.g., valid data, invalid data, missing fields).

Verify the integration of front-end and back-end components through API testing.

Ensure proper handling of error messages and status codes in the API responses.

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

6.1 Impact on Society

The "Property Rental" web application has a significant impact on society, particularly within the context of student life and the housing market in university areas like DIU (Daffodil International University). The platform addresses several societal challenges and contributes positively in various ways:

Enhanced Accessibility to Housing

By offering a centralized platform for rental properties, "Property Rental" makes it easier for students to find and secure accommodations. This improved accessibility helps reduce the stress and uncertainty often associated with searching for housing, especially for newcomers to the area.

Improved Student Well-being

Finding suitable and affordable housing is crucial for students' overall well-being. "Property Rental" provides a user-friendly and efficient solution, ensuring that students can focus more on their academic and social lives rather than worrying about housing issues. This can lead to better academic performance and a more balanced lifestyle.

Economic Benefits

The platform supports the local economy by facilitating the rental market. Landlords benefit from increased visibility and the ability to reach a wider audience, leading to higher occupancy rates. Students, in turn, benefit from a competitive market where they can find housing that fits their budget.

Community Building

By connecting students with landlords and helping them find homes within the university area, "Property Rental" fosters a sense of community. Students living near their university are more likely to participate in campus activities and form social connections, enhancing the overall university experience.

6.2 Environmental Impact

The "Property Rental" web application also has implications for the environment, both positive and negative. Addressing these impacts is crucial for promoting sustainability:

Reduction in Travel and Resource Use

The digital nature of the platform reduces the need for physical travel when searching for properties. Students can view listings and communicate with landlords online, significantly cutting down on the carbon footprint associated with traditional house-hunting methods.

Promotion of Efficient Use of Resources

The platform encourages efficient use of existing housing resources. By filling vacancies promptly and ensuring that properties are rented out continuously, "Property Rental" helps prevent wastage of housing resources. This efficient use of resources aligns with sustainable living practices.

Potential for Encouraging Sustainable Practices

The platform can highlight properties with sustainable features, such as energy-efficient appliances, solar panels, or proximity to public transportation. By doing so, "Property Rental" can incentivize landlords to adopt sustainable practices and students to choose environmentally friendly living options.

6.3 Sustainability

Sustainability is a core consideration in the design and operation of the "Property Rental" web application. The platform aims to promote long-term positive outcomes for both users and the environment:

Sustainable Business Model

The platform's business model focuses on continuous improvement and user satisfaction. By regularly updating features based on user feedback and market trends, "Property Rental" ensures it remains a valuable and reliable resource. This approach supports the platform's long-term viability and success.

Energy Efficiency and Digital Infrastructure

Hosting the application on energy-efficient servers and using green hosting providers can minimize the environmental impact of the platform's digital infrastructure. Additionally, optimizing the application's code and performance can reduce energy consumption.

Encouraging Sustainable Living

"Property Rental" can incorporate educational content about sustainable living practices. By providing tips and information on energy conservation, waste reduction, and sustainable commuting, the platform can empower students to adopt eco-friendly habits in their daily lives.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The "Property Rental" web application has been designed to address the housing challenges faced by students, particularly those at Daffodil International University (DIU). By streamlining the rental process and providing a user-friendly interface, the platform significantly simplifies the search for affordable and suitable accommodation. It enhances the rental experience by offering comprehensive property listings, intuitive search functionalities, and direct communication channels between tenants and landlords.

The application fosters a more efficient, transparent, and accessible housing market. It reduces the time and effort required for students to find housing, ensures that landlords can easily manage their property listings, and provides a secure and trustworthy environment for both parties. The focus on affordability, convenience, and user experience ensures that the platform meets the unique needs of student renters while also benefiting landlords by increasing property visibility and occupancy rates.

Moreover, the "Property Rental" application contributes positively to the community by enhancing student well-being, supporting academic success, and promoting a sense of community among students living near their university. The platform's commitment to transparency and reliability builds trust among users, further solidifying its role as a vital resource for student housing.

7.2 Future Scope

While the "Property Rental" web application addresses many current challenges in the student housing market, there are several areas for future development and improvement:

1. **Enhanced User Features:** Future updates can include additional features such as advanced filtering options, personalized property recommendations, and integration with mapping services to provide insights into nearby amenities and transportation options.
2. **Mobile Application Development:** Developing a mobile app version of the platform would provide greater accessibility and convenience for users. Mobile apps can offer push notifications for new listings, updates, and reminders, enhancing the overall user experience.
3. **Integration with Payment Systems:** Integrating secure online payment systems for rental transactions could streamline the rental process even further. This feature would allow tenants to pay rent directly through the platform, simplifying financial transactions and ensuring timely payments. A robust payment gateway would provide a seamless and secure method for handling rent payments, security deposits, and other financial transactions related to property rentals.
4. **Virtual Reality Tours:** Incorporating virtual reality (VR) tours of properties can provide a more immersive experience for potential tenants, allowing them to explore properties in detail without needing to visit them in person. This can be especially useful for international students or those living far from the university.
5. **Community and Social Features:** Adding social features such as forums, discussion boards, and tenant reviews can create a sense of community among

users. These features can provide valuable insights, foster connections, and enhance the overall user experience.

6. **Sustainability Initiatives:** Future developments could include features that promote sustainable living, such as highlighting eco-friendly properties, providing tips on energy conservation, and encouraging landlords to adopt green practices.
7. **Data Analytics and Insights:** Leveraging data analytics can provide valuable insights into market trends, user preferences, and property performance. This information can help landlords optimize their listings and assist the platform in continually improving its services.
8. **Expanded Geographic Coverage:** Initially focused on DIU and its surrounding areas, the platform can expand to include other universities and regions. This expansion would broaden the user base and provide housing solutions to a larger student population.

Reference

- [1] Google. (n.d.). *Flutter*. Retrieved from <https://flutter.dev>
- [2] Microsoft. (n.d.). *TypeScript*. Retrieved from <https://www.typescriptlang.org>.
- [3] Express.js. (n.d.). *Express - Node.js web application framework*. Retrieved from <https://expressjs.com>
- [4] MongoDB, Inc. (n.d.). *MongoDB - The database for modern applications*. Retrieved from <https://www.mongodb.com>
- [5] Mongoose. (n.d.). *Mongoose - Elegant MongoDB object modeling for Node.js*. Retrieved from <https://mongoosejs.com>
- [6] Vercel Inc. (n.d.). *Vercel - Develop. Preview. Ship*. Retrieved from <https://vercel.com>
- [7] Cloudinary Ltd. (n.d.). *Cloudinary - Image and Video Management*. Retrieved from <https://cloudinary.com>
- [8] Google. (n.d.). *Firebase - App success made simple*. Retrieved from <https://firebase.google.com>
- [9] Postman, Inc. (n.d.). *Postman - API Platform for Building and Using APIs*. Retrieved from <https://www.postman.com>

Appendix A

Course Outcomes, Complex Engineering Problems (EP) and Complex Engineering Activities (EA) Addressing

Title: Property Rental – A Web Application

Student ID: 201-15-3090

CO Description for FYDP

CO	CO Descriptions	PO
Phase -I		
CO1	Integrate recently gained and previously acquired knowledge to identify a web application problem for the Final Year Design Project (FYDP)	PO1
CO2	Analyze different aspects of the goals in designing a solution for this FYDP	PO2
CO3	Explore diverse problem domains through a literature review, delineate the issues, and establish the goals for the FYDP	PO4
CO4	Perform economic evaluation and cost estimation and employ suitable project management procedures throughout the development life cycle of the FYDP	PO11
Phase -II		
CO5	Design and develop technical solutions and system components or processes that meet specified requirements, ensuring compliance with public health and safety standards, as well as considering cultural, socioeconomic, and environmental factors in this FYDP	PO3
CO6	Choose and apply appropriate methodologies, resources, and contemporary engineering and IT technologies to address complex engineering processes, encompassing prediction and modeling, while adhering to relevant constraints in this FYDP	PO5
CO7	Analyze societal, health, safety, legal, and cultural considerations, along with associated responsibilities, in the context of professional engineering practice and the resolution of this problem, employing logical reasoning guided by contextual understanding.	PO6
CO8	Comprehend and evaluate the enduring sustainability and impact of professional engineering endeavors in addressing intricate engineering challenges within social and environmental frameworks.	PO7
CO9	Implement ethical principles and adhere to professional standards and norms in this FYDP	PO8
CO10	Capable of operating proficiently both individually and as a team member or leader across diverse teams and interdisciplinary settings in this FYDP.	PO9

CO11	Proficiently communicate with the engineering community and broader society regarding complex engineering endeavors, including the ability to comprehend and generate comprehensive reports and design documentation, as well as provide and receive clear instructions throughout this FYDP.	PO10
CO12	Acknowledge the importance of self-directed and life-long learning within the evolving landscape of technology, and possess the readiness and capability to engage in lifelong learning endeavors.	PO12

Addressing CO (1 to 8), Knowledge Profile (K), Attainment of Complex Engineering Problems (EP), and Attainment of Complex Engineering Activities (EA)

Addressing CO (1 to 8), Knowledge Profile (K), Attainment of Complex Engineering Problems (EP):

SN	EP Definition	Attainment	CO	Justification (with Knowledge Profile)	References
1.	EP1: Depth of Knowledge required	Yes	CO1, CO2, CO3, CO5, CO6, CO7 and CO8	The project covers Engineering Fundamentals (K3) through process modeling, data design, and front-end development, showcasing strong application of engineering principles. The project addresses Specialist Knowledge (K4) by integrating front-end and back-end design, interaction design, and user experience (UX). Utilizing Flutter for the front end, and TypeScript, Express.js, and MongoDB for the back end, the project demonstrates expertise in developing a comprehensive property rental web application. Additionally, the use of Vercel for deployment, Cloudinary for image uploads, and Firebase for video and file storage highlights the application of modern web technologies in creating a robust and scalable solution.	Page no: [16,19-26,29], Section: [3.1, 3.5, 3.6, 5.1, 5.2] Page no: [27-36], Section: [4.1, 4.2, 5.1, 5.2, 5.4]
				The project applies engineering practice & design (K5) by integrating design requirements such as user interface, security, scalability, customization, compatibility, accessibility, and reliability, ensuring a robust and user-centric property rental web	Page no: [18-26], Section: [3.1, 3.5,

				application aligned with design principles. The project addresses engineering practice & technology (K6) by employing Flutter for the front end, and TypeScript, Express.js, MongoDB, Vercel, Cloudinary, and Firebase for the back end, emphasizing secure, scalable development. The use of Mongoose for database management and Vercel for deployment further underscores the project's commitment to modern, efficient, and scalable development practices.	3.6] Page no: [28-30], Section: [4.1, 4.2]
				The project covers research literature (K8) by reviewing analogous platforms, UI/UX studies, security protocols, personalization algorithms, and industry trends. This comprehensive review has enabled informed decision-making for integrating innovative features into the property rental web application, ensuring it meets the needs of users while staying current with industry best practices.	Page no: [15-17], Section: [2.2]
2.	EP2: Range of Conflicting Requirements	Yes	CO2, and CO7	The project confronts Engineering Practice EP-2 by addressing challenges such as integrating diverse features while maintaining user experience, balancing security with functionality, ensuring scalability, achieving real-time data synchronization, adhering to evolving industry standards, complying with privacy regulations, and implementing effective recommendation systems. These challenges are met through careful planning, flexible development, and continuous adaptation, ensuring the property rental web application remains robust, secure, and user-friendly.	Page no: [13] Section: [2.3, 2.4]

3.	EP3: Depth of analysis required	Yes	CO2, and CO6	The project tackles Engineering Practice EP-3 by prioritizing solutions such as contemporary front-end design for enhanced user experience and selecting Flutter for streamlined front-end development. For the back-end, technologies like TypeScript, Express.js, and MongoDB are used, addressing challenges such as user interaction optimization, platform accessibility, and scalability. These efforts ensure the creation of a superior property rental web application, providing a seamless and efficient user experience.	Page no: [28-30], Section: [4.1, 4.2]
4.	EP4: Familiarity of Issues	Yes	CO8	The project fulfills Engineering Practice EP-4 by integrating insights from property rental management into the development process. This includes optimizing workflows, enhancing user experience, ensuring data security, and contributing to industry efficiency and effectiveness. By bridging Computer Science and Engineering with the property rental domain, the project demonstrates an impact that extends beyond traditional CSE boundaries.	Page no: [15-17], Section: [2.2, 2.4]
5.	EP5: Extends of application codes	No	CO5	N/A	N/A
6.	EP6: Extends of stakeholders involved and conflicting requirements	No	CO8	N/A	N/A

7.	EP7: Interdependence	Yes	CO5	The project meets Complex Engineering Problems (CEP), EP-7 by addressing high-level challenges across various stages of development. It ensures data integrity, security, and performance in database implementation, optimizes user experience through intuitive front-end design, and conducts comprehensive testing to guarantee a robust and efficient property rental web application.	Page no: [31-36], Section: [5.1, 5.2, 5.3]
----	-----------------------------	-----	-----	--	---

Addressing CO11 with Complex Engineering Activities (EA) [Some or all of the following]:

SN	EA Definition	Attainment	CO	Justification	References
1.	EA1: Range of resources	Yes	CO11	The development of this system leverages a skilled team across multiple domains, meticulously allocates financial resources for infrastructure, marketing, and development, and utilizes tools like Android Studio along with various programming languages, frameworks, and databases for effective execution. This comprehensive approach ensures a robust and scalable property rental web application.	Page no: [30], Section: [4.2]

2.	EA2: Level of interaction	No		N/A	N/A
3.	EA3: Innovation	No		N/A	N/A
4.	EA4: Consequences for society and the environment	Yes		This project emphasizes social impact and environmental sustainability by enhancing the efficiency of finding rental accommodations, promoting sustainable living practices, and ensuring ethical considerations such as strong security, privacy, and transparent business practices. These efforts foster societal well-being and contribute to environmental stewardship.	Page no: [37-38], Section: [5.4]
5.	EA-5: Familiarity	Yes		The Comparative Analysis for "Property Rental" evaluates market competitors to highlight its unique features, security measures, personalization, and customer engagement strategies. This analysis enriches market understanding and informs strategic decisions, enhancing the application's competitiveness and value proposition.	Page no: [15-17], Section: [2.1, 2.3]

Addressing CO (4, 9, 10, and 12):

SN	COs	Attainment	Justification	References
1	CO4	Yes	The project captures CO4 by integrating financial management practices such as budgeting, resource allocation, cost management, and risk mitigation. This approach ensures efficient fund distribution, monitors expenditures, and identifies optimization areas while maintaining high quality throughout the development process.	Page no: [12-13], Section: [1.4]
2	CO9	Yes	The project upholds ethical practices by prioritizing user privacy, maintaining transparency in data usage, providing fair suggestions, respecting cultural sensitivities, and promoting accessibility. These efforts foster trust, transparency, and integrity among stakeholders involved in the property rental web application.	Page no: [39], Section: [6.2]
3	CO10	No	N/A	N/A
4	CO12	Yes	The project's implementation phase emphasizes lifelong learning through diverse testing protocols, automated frameworks, and CI/CD pipelines. These aspects ensure ongoing improvement and adaptation to emerging technological trends, with results analysis refining the property rental platform for enhanced effectiveness over time.	Page no: [31-37], Section: [5.1, 5.4]

Turnitin Originality Report

Processed on: 29-Jun-2024 12:57 +06

ID: 2410158316

Word Count: 7732

Submitted: 1

Property Rental: A Web Application By
201-15-3090 Mahmud Reza
Chowdhury Ani

Similarity Index

19%

Similarity by Source

Internet Sources: 15%
Publications: 4%
Student Papers: 17%

2% match (student papers from 29-Jun-2024)

[Submitted to Daffodil International University on 2024-06-29](#)

2% match (student papers from 24-May-2024)

[Submitted to Daffodil International University on 2024-05-24](#)

2% match (student papers from 23-Jan-2024)

[Submitted to Daffodil International University on 2024-01-23](#)

1% match (student papers from 24-Jun-2024)

[Submitted to Daffodil International University on 2024-06-24](#)

1% match (Internet from 17-Jun-2024)

<https://open-innovation-projects.org/blog/page/112>

< 1% match (student papers from 04-Apr-2018)

[Submitted to Daffodil International University on 2018-04-04](#)

< 1% match (student papers from 09-Jul-2023)

[Submitted to Daffodil International University on 2023-07-09](#)

< 1% match (student papers from 14-Jan-2021)

Class: Fall 2020 - Fall 2023

Assignment: SI

Paper ID: [1487304266](#)

< 1% match (student papers from 02-Apr-2019)

[Submitted to Daffodil International University on 2019-04-02](#)

< 1% match (student papers from 04-Apr-2018)

[Submitted to Daffodil International University on 2018-04-04](#)

< 1% match (student papers from 23-Jan-2024)

[Submitted to Daffodil International University on 2024-01-23](#)

< 1% match (student papers from 07-Apr-2018)

[Submitted to Daffodil International University on 2018-04-07](#)

< 1% match (student papers from 20-Jan-2021)

Class: Fall 2020 - Fall 2023

Assignment: SI

Paper ID: [1490623285](#)

< 1% match (student papers from 02-Apr-2019)

[Submitted to Daffodil International University on 2019-04-02](#)

< 1% match (Internet from 25-Oct-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6954/172-15-10074%20%2c%207%25.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 21-Nov-2022)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7111/171-15-9026%20%2822%25%29clearance.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 29-Jun-2023)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/9853/22853.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 11-Oct-2022)