

A PRELIMINARY PROJECT REPORT ON
“HOME AWAY”
Your Ultimate Housing Solution

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

Submitted by

Roll No	Name	Seat No.
BEA06	Anupam Pandey	B190594206
BEA16	Yash Pramod Bambal	B190594218
BEA19	Sakshi Baburao Bhor	B190594221
BEA40	Karn Datta Godse	B190594245



“येथे बहुतांचे हित”

“Towards Ubiquitous Computing Technology”

DEPARTMENT OF COMPUTER ENGINEERING

“Techno – Social Excellence”

Marathwada Mitra Mandal's Institute of Technology (MMIT)

Lohegaon, Pune- 411 047

SAVITRIBAI PHULE PUNE UNIVERSITY

(2023-24)



“येथे बहुतांचे हित”

CERTIFICATE

This is to certify that the project report entitles

“HOME AWAY”

Your Ultimate Housing Solution

Submitted by

Roll No	Name	Seat No.
BEA06	Anupam Pandey	B190594206
BEA16	Yash Pramod Bambal	B190594218
BEA19	Sakshi Baburao Bhor	B190594221
BEA40	Karn Datta Godse	B190594245

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of Mr. **Ashish Bhise** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Mr. Ashish Bhise
Guide
Head, of AI & DS Department
of Engineering

Dr. Subhash G. Rathod
Head,
Department of Computer Engineering

Dr. R. V. Bhortake
Principal,
Marathwada Mitra Mandal's Institute of Technology, Pune

Place: Pune

Date:— _____

SPONSORSHIP LETTER

SAMARTHA CONSTRUCTION

Office No. F-02, Sr no 3/2, CTS No 984, Inspiria Mall

Near Bhakti-Shakti Chowk, Nigadi, Pune-411044

Email- samarthconstruction.nd@gmail.com

Cell : +91-9823054173

GSTIN No: 27AFYPD641K1Z2

Date – 22 Oct 2023

Re: Official Sponsorship Approval Letter.

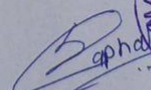
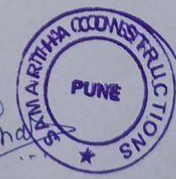
Dear Students,

This is to inform official sponsorship approval for an Web application given to the students of Computer Engineering at “Marathwada Mitra Mandal Institute of Technology, Pune”.

We would to confirm that **Anupam Pandey, Yash Bambal, Sakshi Bhor, Karn Godse** are official recipient of sponsorship under “Samartha Constructions”.

Project Name: **HOME AWAY- Your Ultimate Housing Solution**

The approval consists of total expenditure required for completion of the project.



For SAMARTHA CONSTRUCTIONS

ACKNOWLEDGEMENT

We invested a lot of time and energy into this project. However, a lot of people contributed to and encouraged us as we finished our tasks. We want to express our sincere gratitude to each one of them.

We are grateful to **Mr. Ashish Bhise** for acting as our mentor and for always being available to supervise us and provide the project with the information and insights we needed. We are appreciative of his guidance.

We would like to express our gratitude to our parents and friends at Marathwada Mitra Mandal's Institute of Technology for their support and cooperation in helping us finish this job.

Mr. ANUPAM PANDEY
Mr. YASH BAMBAL
Ms. SAKSHI BHOR
Mr. KARN GODSE

ABSTRACT

In an increasingly interconnected world, finding the perfect home, whether it's for a long-term lease or a short-term stay, can be a daunting task. Enter "HomeAway," your one-stop web application that revolutionizes the way owners and tenants connect, streamlining the process of finding the ideal living space for both individuals and bachelors. "HomeAway" is more than just a housing search platform; it's a dynamic solution that facilitates direct communication between property owners and potential tenants. Whether you're a property owner looking for responsible occupants or a bachelor in search of the perfect bachelor pad, our platform empowers you to connect seamlessly, making the housing hunt a hassle-free experience. Not only does "HomeAway" help you find the right match, but it also specializes in short-term housing searches. Whether you're in need of a temporary abode for a few weeks or a couple of months, our website's intuitive features will assist you in discovering the perfect short-term housing solution, connecting you with property owners willing to accommodate your needs. Welcome to "HomeAway" – where finding your ideal home and connecting with property owners or tenants has never been easier. Join us on this exciting journey of simplifying the housing search process, and make your next move with confidence.

TABLE OF CONTENTS

SR NO.	CHAPTER TITLE	PAGE NO.
01	Introduction	1
1.1	Overview	2
1.2	Motivation	3
1.3	Problem definition and objectives	4
1.4	Project scope and limitations	5
1.5	Methodologies of problem solving	6
02	Literature Survey	8
03	Literature Review	9
3.1	Online Survey	11
04	System Requirement Specifications	12
4.1	Introduction	13
4.1.1	Project Scope	13
4.2	Functional Requirement	14
4.2.1	Performance Requirements	14
4.2.2	Security Requirements	14
4.3	Existing Systems	15
4.3.1	Manual	15
4.3.2	Scope	15
4.3.3	Limitations	16
4.4	Project Perspectives	17
4.4.1	Tenant Perspective	17
4.4.2	House Owner Perspective	17
4.4.3	Admin Perspective	17
4.5	Project Features	17
4.6	User Interfaces	18
4.7	Software Quality Attributes	20
4.8	System Requirements	22
4.8.1	Database Requirements	23
4.8.2	Software Requirements	23
4.8.3	Hardware Requirement	25
4.9	Analysis Models System Design System	26
4.10	Implementation plan	27

05	System Design	28
	5.1 System Architecture	29
	5.2 Data Flow Diagrams	30
	5.3 UML Diagram	38
	5.4 Class Diagram	39
	5.5 State Diagram	40
	5.6 Sequence Diagram	43
	5.7 Deployment Diagram	45
	5.8 Object Diagram	46
	5.9 Component Diagram	47
06	Other Specifications	48
	6.1 Advantages	48
	6.2 Disadvantages	48
	6.3 Applications	49
07	Project plan	50
	7.1 Project understanding	51
	7.2 Project schedule	52
	7.2.1 Project Timeline Chart	52
	7.2.2 Project Task Set	53
	7.3 Task Network	54
	7.4 Risk Management	54
	7.4.1 Risk identification	54
	7.4.2 Risk Analysis	55
	7.5 Organization	56
08	Project Implementation Details	
	8.1 Project Modules	57
	8.2 Algorithm details	58
	8.2.1 What is an algorithm	58
	8.2.2 Characteristics of an algorithm	58
	8.2.3 Complexity	61
09	Testing	64
10	Results	70
	10.1 Snapshots of outcome	79
11	Conclusions and Future Work	80
	11.1 Future scope	81
	11.2 Plagiarism Report	82
	11.3 References	83

LIST OF FIGURES

FIGURE	ILLUSTRATION	PAGE NO.
4.9	Analysis Model System Design	26
5.1	System Architecture	28
5.2	Data Flow Diagram	29
5.2.1	Data Flow Diagram for owner level-1	30
5.2.2	Data Flow Diagram level-2	30
5.2.3	Data Flow Diagram level-3	31
5.3	UML Diagram	38
5.4	Class Diagram	39
5.5	State Diagram	42
5.6	Sequence Diagram	43
5.7	Deployment Diagram	44
5.8	Object Diagram	45
5.9	Component Diagram	46 40

LIST OF TABLES

TABLE	ILLUSTRATION	PAGE NO.
2	Literature Review	9
7.2.1	Timeline Chart	52
9	Testing	64

Chapter 1

INTRODUCTION

Chapter 1: INTRODUCTION

1.1 OVERVIEW

Home Away is an innovative web application designed to revolutionize the way individuals seek rental accommodations and streamline the rental process. This platform is a one-stop solution for all your housing needs, offering an array of services. At its core, Home Away enables users to search and rent houses, be it for long-term leases or short-term stays.

One of its standout features is the ability to create and manage rental agreements online, simplifying the often complex and paper-intensive legal aspects of renting. Users can draft, review, and sign digital contracts, enhancing transparency and security. Additionally, Home Away fosters seamless communication between landlords and tenants through its built-in messaging system, making it easy to discuss property details, negotiate terms, and address concerns in real-time.

With an intuitive interface and a comprehensive database of rental listings, Home Away is poised to redefine the way people find, rent, and manage their homes, offering convenience, efficiency, and peace of mind in the housing market.

1.2 MOTIVATION

Motivation to embark on the journey of creating a web application like Home Away stems from a profound desire to reshape and optimize the landscape of rental housing. In a world where finding the ideal place to live often involves a labyrinth of obstacles and the negotiation of intricate rental agreements, Home Away emerges as a beacon of simplicity, efficiency, and convenience. It is driven by the vision of offering a comprehensive solution for renters and landlords alike. At the heart of Home Away's innovation is the transformation of the traditional rental agreement process. By introducing the capability to create, review, and sign rental agreements online, Home Away enhances transparency and simplifies the often perplexing legal aspects of renting. This feature empowers users with the tools to draft and negotiate digital contracts, fostering a more secure and straightforward transaction.

Another vital aspect of this motivation lies in facilitating seamless communication. Home Away provides an integrated messaging system that enables real-time conversations between landlords and tenants. This function is not merely about technology; it's about fostering better landlord-tenant relationships, reducing misunderstandings, and ensuring that issues are addressed promptly. Moreover, Home Away adapts to the evolving housing market by catering to the surging demand for short-term rentals. This flexibility opens doors to various lifestyles, where individuals can explore new homes and locations with ease, making it suitable for a diverse range of users. The motivation to create Home Away is also rooted in the empowerment of users. Its user-friendly interface and extensive database of rental listings offer individuals the means to make informed decisions about their living spaces. This empowerment transcends the confines of the digital platform, giving people the autonomy to shape their housing choices in an informed and decisive manner.

In essence, Home Away is not just a web application; it is a catalyst for positive change in the realm of real estate. It is a mission to simplify complexities, promote transparent transactions, enhance communication, and, most importantly, offer a more enjoyable and efficient way for individuals to find, rent, and manage their homes. The motivation is embedded in the opportunity to redefine the housing market, making it more accessible and harmonious for everyone involved.

1.3 PROBLEM STATEMENT AND OBJECTIVES

Traditional property transactions are marred by inefficiencies. Real estate agents, brokers, and third-party listing websites have long been the intermediaries through which individuals seek rental or sale properties. This middleman-heavy approach not only adds unnecessary costs but also creates barriers in communication between property seekers and owners. These obstacles can lead to confusion, miscommunication and extended timelines.

- Direct Contact
- Comprehensive Listings
- User Profiles
- Real-Time Updates
- Transparency
- Reviews and Ratings

1.4 PROJECT SCOPE AND LIMITATIONS

SCOPE :

The proposed system is concerned searching a home rental, Payment and account controls in online way. The system will remove the tedious task of customers for searching home for rental. The system will help to the home administrative staff i.e. the Home rental to keep the daily and the history record details of the customers need in proper database.

LIMITATIONS:

When there are many advantages of a system there is also many problems also with the system. In performing this project there may face many constraints like lack of availability of enough secondary data and it may be inconsistent with the project.

- The error may occur if the transaction cancels or if the server is down.
- Lack of adequate time in order to get the additional secondary data.
- Lack of resources like Hardware and Software
- Lack of reference materials
- Lack of proficiency person in the product management system Our system could be:
 1. electric dependent
 2. if there is no internet service our system can't operate.

The rental system is not entirely online as users might be hesitant to upload confidential documents online.

1.5 METHODOLOGIES OF PROBLEM SOLVING

1. **Analysis:** Before starting the development process, a thorough analysis of the problem domain and requirements is conducted. This involves understanding the objectives, identifying the target audience, and defining the scope of the project. The analysis helps in gaining a clear understanding of the problem and setting the foundation for the subsequent steps.
2. **Design:** The design phase involves creating a blueprint for the application's structure and functionality. This includes designing the user interface, planning the interactive visualizations, and defining the algorithms' implementation logic. Design decisions are made based on the project objectives, user needs, and best practices in UI/UX design.
3. **Development:** The development phase involves the actual implementation of the web-based application. HTML, CSS, Java, Script, and third-party UI libraries are used to build the user interface. Blender is used to create 3D model of virtual space and mozilla spoke is use to deploy model to hubs. Echo 3D platform is used to store 3D objects in its cloud and is use to view it in3D.
4. **Testing:** Throughout the development process, rigorous testing is conducted to ensure the functionality, usability, and reliability of the application. Testing involves unit testing, integration testing, and user testing. The application is tested on different devices and browsers to identify and fix any bugs, errors, or compatibility issues.

5. **Iterative Approach:** The project follows an iterative approach, where the development process is divided into multiple iterations or sprints. Each iteration focuses on implementing a set of functionalities, testing them, and gathering feedback. This allows for continuous improvement, feedback incorporation, and adaptation of the application based on user requirements and testing outcomes.
6. **User Feedback Incorporation:** User feedback plays a crucial role in refining and enhancing the application. Feedback is collected through user testing, surveys, or direct user interaction. The feedback is analyzed and used to identify areas of improvement, address usability issues, and make necessary adjustments to the application's design and functionality.
7. **Evaluation and Refinement:** The project includes an evaluation phase to assess the effectiveness of the application in achieving its objectives. The evaluation involves measuring the impact on learners' understanding, engagement, and problem-solving skills. Based on the evaluation results, refinements and enhancements are made to improve the application's effectiveness and user experience.

CHAPTER 2

LITERATURE SURVEY

Chapter 2 : LITERATURE REVIEW

Sr. No.	Title	Publish Year	Authors	Findings
1.	TEXT-BASED PRICE RECOMMENDATION SYSTEM FOR ONLINE RENTAL HOUSE	2020	Qianjun Liu, Gong Chen, and Shouling Ji	We analyzed the relationship between the description of each listing and its price, and proposed a text-based price recommendation.
2.	THE ROLE OF THE PRIVATE SECTOR IN HOUSE MANAGEMENT	2020	Lujia Shen, Qianjun Liu, Gong Chen, and Shouling Ji	In this paper, we presented TAPE, a text-based price recommendation system for predicting a reasonable price for newly added listings.

3.	SHOP GO : AN IOT BASED SOLUTION 2021 FOR SMART SHOPPING	2021	Munchetty Ashutosh Rao	Supermarkets and shopping complexes contain a variety of products at the same place to help customers find their items easily without travelling to different places and get their items in one place.
4.	HUMAN RESOURCE MANAGEMENT PORTAL	2022	Jesada kittivaraporg	Project aims to enhance the design and development of a Human Resource application by empowering it with an efficient and innovative communication channel using web portal.
5.	EXPLORING AIRBNB USERS' CONCERNS WITH LDA- BASED TOPIC MODEL AND SENTIMENT ANALYSIS	2022	Jingya Liu X	Techniques such as text management and sentiment analysis have been applied widely to extract useful information from social network comments.

3.1 ONLINE SURVEY

We have gone through various popular similar web applications to do the current market analysis to find the best rental property for tenants or for owner to get the best tenant with faithfulness at its best and analyze the scope of improvements to the field. We propose these enhancements that we hope to make to our website after analyzing other online projects comparable to our project:

- Better User Interface and User Experience
- A responsive website that will deliver an excellent user experience
- Satisfying the conditions for owner as well as for tenant
- Easy way of payment
- Track of previous records and transactions
- Transparency in documents

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

Chapter 4: SOFTWARE REQUIREMENTS SPECIFICATION

4.1 INTRODUCTION

Software Requirements Specifications (SRS) are created during the requirements stage of the software development process. It is also called a Requirements document. When all requirements are gathered and analyzed, this report is constructed, laying the groundwork for software engineering efforts. SRS is a formal report that serves as a software representation, allowing clients to assess if the report (SRS) satisfies their needs. It also includes specific descriptions of system requirements as well as user needs for the system.

4.1.1Project Scope

The project scope defines the description of the work that is required in delivering the rental house management system. The following are the scopes of work during the project :

1. Study and understand the requirement of this project.
2. Construct Software Requirement Specification document of the system.
3. Construct Software Design Document of the system.

4.2 FUNCTIONAL REQUIREMENT

This is a necessary task, action or activity that was accomplished. The proposed system can:

1. Allow administrator to add a house, tenant, and defaulter's details.
2. Allow the administrator to delete houses, tenants, and defaulter's details.
3. Allow the administrator to search data in the database.
4. Allow the administrator to edit data in the database

4.2.1 PERFORMANCE REQUIREMENTS

This section describes the major performance requirements of the system.

1. Stable internet connection is required to use the application as it has some functionalities which requires a stable internet connection.
2. As the applications contain a large amount of data in the database, sufficient memory is required.

4.2.2 SECURITY REQUIREMENTS

This section describes the major security requirements of the system.

3. The database and the physical view of the system will remain hidden from the users, only an admin or the developer can view the back-end of the system.
4. User authorization and data encryption are important security requirements of the system. User stream should not be available to anyone who is not authorized by the user of the device.
5. The database is encrypted by a password. Any unauthorized user cannot access, update or manipulate the database.

4.3 EXISTING SYSTEMS

4.3.1 MANUAL :

Currently most property managers manage property and tenants 'details on papers. Once customers find a vacant house, they can call or email manager of the houses indicating the size of the house they would like rented to them. The property manager can email them back giving them all the details about the house they are requesting. The details include:

- Rent per month
- Deposit paid
- Terms and conditions to follow acceptance.

4.3.2 SCOPE:

User i.e. tenant can interact with the house owner in person. Some individuals might find online systems complicated & hard to operate.

Physical inspection of the property can be made at the first instance.

1. One of the most important benefits of manual house renting system is that physical inspection of the property can be done thoroughly and check the presence as well as working condition of electrical equipment.
2. While renting the houses no virtual conversation is needed. Face to face conversations are done.

4.3.3. LIMITATIONS:

When there are many advantages of a system there is also many problems also with the system. In performing this project there may face many constraints like lack of availability of enough secondary data and it may be inconsistent with the project.

- The error may occur if the transaction cancels or if the server is down.
- Lack of adequate time in order to get the additional secondary data.
- Lack of resources like Hardware and Software
- Lack of reference materials
- Lack of proficiency person in the product management system Our system could be:
 1. electric dependent
 2. if there is no internet service our system can't operate.

4.3PROJECT PERSPECTIVES

4.3.3 TENANT PERSPECTIVE

The tenant perspective can extract maximum benefits from Home Away since any person registering into Home Away can be considered as a user. A user must register on the site and by the credentials, he/she can login into the portal. He/she can view the properties. User can then proceed with a particular property by carrying out the payment

4.3.4 HOUSE OWNER PERSPECTIVE

An owner can add the properties onto the website, thus increasing the property availability, consequently expanding his/her own access to tenants. An owner can register as an owner and login with the registered credentials. An owner can add new properties as well as delete previously registered properties into the existing list, thereby providing users a better access to different properties.

4.3.5 ADMIN PERSPECTIVE

An admin is the person who manages the web application. The admin should already have a username and a password. He/she can login with those credentials. Admin can manage users, add a new admin, add an owner, add a property, and check out the reports about the property details.

4.4 PROJECT FEATURES

1. User Authentication:

This application provides the user authentication functionality. A user needs to register himself/herself before logging in to the portal. The portal validates the username and password. Only if the validation is successful, the user will be able to login in to the application.

2. Search Properties:

Home Away offers the user various properties. It also allows users (Tenants) to check out the specifications of the properties along with multiple images for a thorough viewing experience

3. Add/Delete Properties:

A user logged in as the owner can add & delete properties to the existing property list. The property list will automatically be updated..

4. View Report:

An admin of the application can view the report about the transactions. It contains the order details like the name of the tenant interested, the property details, the cost of the property, etc. The report also contains the total yield earned from the transactions.

5. Feedback Page:

A user can give feedback to the admin through a contact us feedback form. Hence, it helps the admin to improve any functionality of the website thus providing maximum customer satisfaction.

4.5 USER INTERFACES

The user interface of the system is very friendly, easy to understand, without the need of a user manual. The details of the user interfaces of our system are as follows:

1. User Interface
2. Distributor Interface
3. Admin Interface

The user has to go through following:

i) Registration Page:

1. Username
2. Password
3. Email ID
4. Type of User

ii) Login Page

- a) Username
- b) Password

NOTE: The above two pages are common in all user interfaces.

iii) Home Page: It is the welcome page containing a navigation bar to various webpages and links to cities.

- a) Home
- b) About
- c) Search
- d) Feedback
- e) Services
- f) Log out

iv) Property: It contains the list of properties available for the user to rent.

- a) Property Type
- b) Rent Amount
- c) Get Details

v) Feedback: It is a link for a Google form wherein the user can fill various fields, these details will be received by the admin, and the admin will perform the necessary actions, if any.

4.6 SOFTWARE QUALITY ATTRIBUTES

- **Scalability**

Scalability measures the highest workloads that the system can handle while still delivering the required levels of performance. The Web Application will be able to handle multiple clients simultaneously .

- **Usability**

Usability standards come in a variety of forms.

- **Learnability**

Users will be able to learn to operate our system very easily as our user interface is very user-friendly. We will also provide a tutorial about how to use the application when the user opens the application for the first time.

- **Efficiency**

We have only focused on essential topics and relevant components to ensure the user does not fiddle around.

- **Satisfaction**

Our design is pleasing to use with appropriate colors and animations that look good and don't strain the eyes of the user. Users will be able to easily use the Web Application without any help from others.

- **Reliability**

The dependability of a system or component indicates the likelihood that it will function well for a predetermined amount of time under specific circumstances. The application will always provide the correct information and intended visualization to the user when the input is provided to it.

- **Responsiveness**

The Web Application can adapt to various screen sizes and will also work properly on mobile devices.

4.7 SYSTEM REQUIREMENTS

4.7.1 Database Requirements PostgreSQL:

It provides quick storage and retrieval methods.

4.7.2 Software Requirements (Platform Choice)

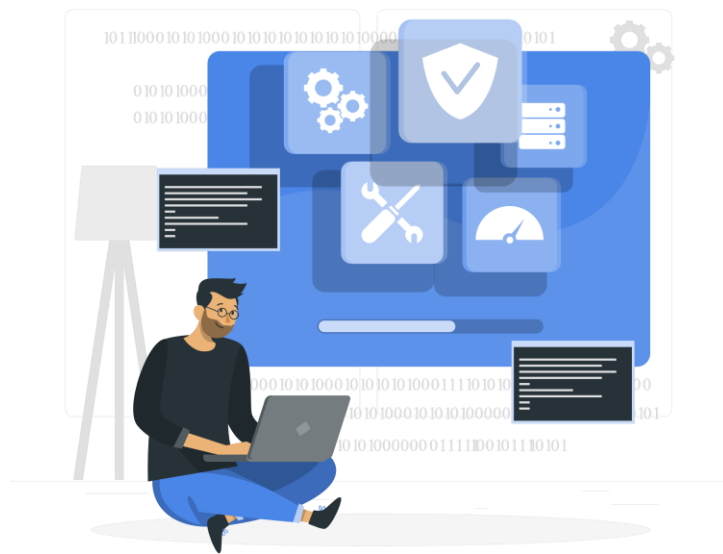


Fig. 4.1 System Requirements Analysis

Software Requirements

1. **Operating System: Windows/ Linux/Mac OS:**

It works on Windows, Mac OS as well as Linux OS with minor changes in implementation.

2. **Server: APACHE TOMCAT v10**

Apache Tomcat, also known as Tomcat Server, proves to be a popular choice for web developers building and maintaining dynamic websites and applications based on the Java

3. Programming language: JAVA

One of the most widely used programming languages, Java is used as the server-side language for most back-end development projects, including those involving big data and Android development.

4. Scripting language: JavaScript, HTML, CSS

HTML and CSS is the best designing language for web applications. With the help of JavaScript, we have made the web page more dynamic and interactive.

5. Database: PostgreSQL:

It provides quick storage and retrieval methods.

PostgreSQL is a powerful open source relational database management system (RDBMS) known for its reliability, extensibility, and adherence to SQL standards. Let's dive into its various aspects in detail:

➤ History and development:

PostgreSQL originated from the POSTGRES project at the University of California, Berkeley in the mid-1980s. It has a long history of development, the first version was released in 1989. Over the years, it has evolved into a robust and feature-rich database system thanks to the contributions of a large and active open-source community.

➤ **Features:**

- **ACID Compliant:** PostgreSQL provides ACID (Atomicity, Consistency, Isolation, Durability) properties, making it suitable for transactional applications.
- **Data Types:** Supports a wide variety of built-in and user-defined data types, including primitive types, geometric types, network address types, JSON, and XML.
- **Extensibility:** PostgreSQL allows users to define their own data types, functions, and procedural languages using its extensibility framework.
- **Scalability:** Supports various scalability features such as table partitioning, parallel query processing, and asynchronous replication.
- **High availability:** PostgreSQL offers built-in replication and clustering solutions such as streaming replication, logical replication, and Patrons for high availability setups.
- **Full-Text Search:** Provides robust full-text search capabilities via the ts vector and ts query types, along with the pg_trgm extension for trigram matching.
- **Geospatial support:** PostgreSQL supports geospatial data through extensions such as PostGIS, which enables the storage, indexing, and querying of geographic data.
- **JSON support:** PostgreSQL includes native support for JSON and JSONB (Binary JSON), allowing JSON data to be stored and queried efficiently.
- **Security:** Provides features such as SSL support, role-based access control (RBAC), and row-level security (RLS) for data security.
- **Rich Ecosystem:** PostgreSQL has an extensive ecosystem of extensions, tools, and libraries, making it suitable for a wide variety of use cases.

➤ **Architecture:**

- **Client-Server Architecture:** PostgreSQL follows a client-server model where clients connect to the PostgreSQL server over a network.
 - **Process Architecture:** Each client connection is handled by a separate server process that manages its own memory and resources.
 - **Storage architecture:** PostgreSQL stores data in tables organized into databases. It uses the Multi-Version Concurrency Control (MVCC) mechanism to manage concurrent data access.
 - **Query processing:** Queries are analyzed, planned and executed by the PostgreSQL query launcher, which generates an optimized query execution plan.
 - **Background processes:** PostgreSQL includes various background processes for tasks such as automatic vacuuming, WAL (Write-Ahead Logging) archiving, and replication.
- **SQL Compliance:** PostgreSQL adheres closely to SQL standards and supports a wide range of SQL features including complex queries, subqueries, joins, and advanced analysis functions. It also provides several extensions and features specific to PostgreSQL, such as common table expressions (CTEs), windowing functions, and table inheritance.
- **Community and Support:** PostgreSQL has a vibrant and active open-source community including developers, users, and contributors from around the world. It has extensive documentation, mailing lists, forums, and community-driven resources that provide support and assistance to users of all skill levels. Commercial support and consulting services are also available from various companies specializing in PostgreSQL.
- **Use Cases:** PostgreSQL is suitable for a wide range of use cases, including web applications, mobile applications, data warehouses, geospatial applications, and enterprise systems. Its flexibility, scalability and extensibility make it a popular choice for both small startups and large enterprises.

6. Browser: Microsoft Edge/ Google Chrome, Mozilla Firefox:

A browser helps to run the localhost server on our device and in turn, helps to execute the entire system.

4.7.3 Hardware Requirements



Fig 4.2 System Hardware Requirement Analysis

Hardware Requirements :

- CPU Processor: 2.0 Ghz or higher speed.
- RAM: Minimum of 2 GB
- ROM: Up to 10-30 GB of storage space

4.8 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

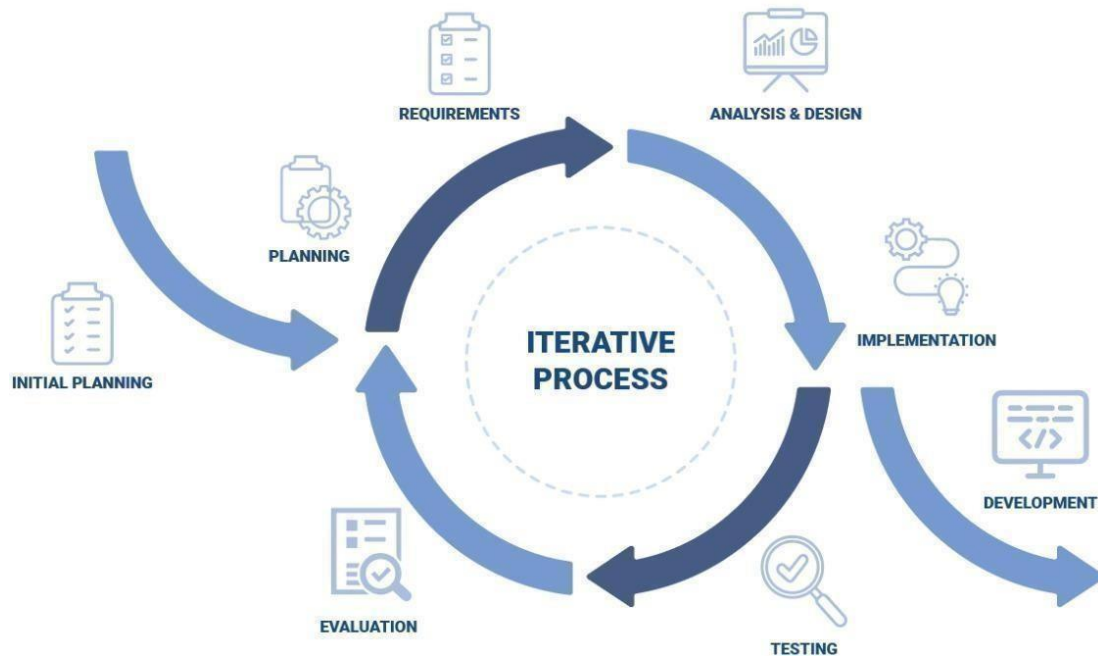


Fig 4.3 Iterative Software Development Model

In the Iterative Model, we begin with some software specs and work our way up to the initial version of the software. If changes to the software are required after the original version, a new version is made with a new iteration. The Iterative Model completes each release over an exact and predetermined time period known as an iteration.

The Iterative Model enables access to prior stages where changes were made accordingly. The following are various stages in the Iterative Model:-

1. **Planning:** We formed our group and communicated our topic to our mentor.
 2. **Requirement gathering & analysis:** In this phase, requirements are gathered from customers and checked by an analyst whether requirements will be fulfilled or not. The analyst checks whether that need will be achieved within budget or not.
- We gathered information about similar existing projects and analyzed them to find their advantages and disadvantages and improvements that can be done to them. We also decide which technologies to use to develop our system.

3.**Design:** In the design phase, the team designs the software by the different diagrams like Data Flow diagrams, UML diagrams etc. We created data flow diagrams, class diagrams, activity diagrams, and use case diagrams for our application.

4.**Implementation:** In the implementation, requirements are written in the coding language and transformed into computer programs which are called Software.

5.**Testing:** After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

6.**Deployment:** After completing all the phases, the software is deployed to its work environment.

4.9 SYSTEM IMPLEMENTATION PLAN

Task Name	July(2023)	Aug	Sept	Oct	Nov	Dec	Jan(2024)	Feb	Mar	April	May
Group Formation											
Ideation											
Literature Survey											
Topic Inquisition											
Mathematical Modelling											
Requirement Analysis											
UML Diagrams											
Software Development											
Testing											
Deployment											
Maintainance and development											

Fig 4.4 Project Plan

CHAPTER 5

SYSTEM DESIGN

Chapter 5 : SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

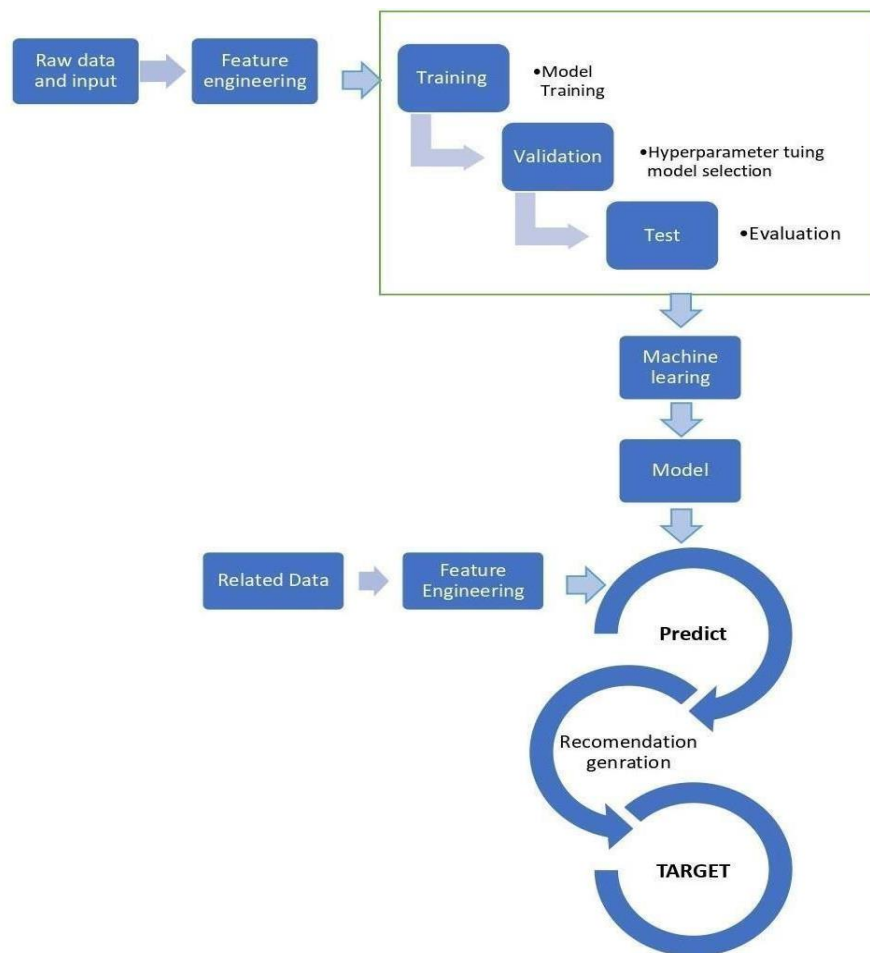


Fig. 5.1 System Architecture

5.2 DATA FLOW DIAGRAMS

The classic visual representation of how information moves through a system is a data flow diagram (DFD). A tidy and understandable DFD can graphically represent the appropriate quantity of the system demand. It can be done manually, automatically, or both.

It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.

A DFD's main goal is to outline a system's overall scope and bounds. It can be used as a tool for communication between a system analyst and anyone involved in the process of establishing the order that will serve as the basis for a system redesign. A bubble chart or data flow diagram are other names for the DFD.

A Data Flow Diagram (DFD) for predicting the best rental house, apartment, or flat in a nearby area based on location can be represented with the following components:

1. External Entities:

- User: The person looking for a rental property.
- Database: The data source containing information about available rental properties.

2. Processes:

- Search Algorithm: This process is responsible for predicting the best rental property based on the user's location and preferences.

3.Data Stores:

- Rental Property Database: Contains information about available rental houses, apartments, or flats, including details like location, price, size, amenities, and user reviews.

4.Data Flows:

- User Input: Data input from the user, including their location and preferences.
- Search Query: Data representing the user's search criteria, such as location, budget, and property type.
- Search Results: Data containing the best rental property options based on the user's search criteria.
- Property Details: Information about individual rental properties, including location, price, size , and amenities.

1. Define goals and target audience:

Identify the main goals and objectives of the application (eg property search, rental management, property listings). Identify your target audience (eg home buyers, renters, landlords, real estate agents) and understand their needs and preferences.

2. Research and planning:

Conduct market research to understand existing solutions, competitive applications and industry trends. Define the features, functionality and key differentiators of the application based on user needs and market gaps. Create user personas and user stories that guide development and prioritize features.

3. Design phase:

Develop wireframes or mockups to visualize your app's layout, navigation, and user interface. Design user experience (UX) and user interface (UI) to ensure ease of use, accessibility and visual appeal. Include feedback from stakeholders and potential users to improve the design.

4. Development iterations:

Start with a Minimum Viable Product (MVP) that contains the basic functionality to proof-of-concept your app. Implement features in iterative development sprints, focusing on one or several features at a time. Thoroughly test each iteration for functionality, performance and usability.

Gather feedback from users and stakeholders after each iteration to identify improvements and prioritize future development.

5. Feature Improvements:

Prioritize improvements and feature enhancements based on user feedback and analytics. Iterate on existing features to improve usability, performance and user satisfaction. Consider adding new features or integrations based on evolving user needs and market trends.

6. Quality Assurance and Testing:

Conduct rigorous testing at every stage of development to identify and fix bugs, errors, and usability issues. Perform functional testing, usability testing, compatibility testing (across devices and platforms), and security testing. Where possible, incorporate automated testing to streamline the testing process and ensure consistency.

7. Deployment and launch:

Prepare for deployment by optimizing performance, resolving any remaining issues, and ensuring compatibility with target platforms. Deploy the app to production or app stores according to the guidelines and requirements of each platform (e.g. Apple App Store, Google Play Store). After launch, monitor the app closely to fix any issues immediately and get user feedback for future updates.

8. Post-launch optimization:

Continuously monitor user feedback, application analytics and performance metrics to identify areas for improvement. Release regular updates and improvements based on user feedback, feature requests and market trends. Engage with users through customer support, community forums, and social media channels to foster loyalty and gain insights for future iterations.

9. Scale and Growth:

Plan for scalability as your user base grows and ensure your application infrastructure can handle increased traffic and data volume. Explore opportunities for expansion, such as adding new features, expanding into new markets, or partnering with complementary services. Stay agile and responsive to evolving user needs, technological advances and market dynamics to stay competitive and relevant.

Data Flow diagram for tenant

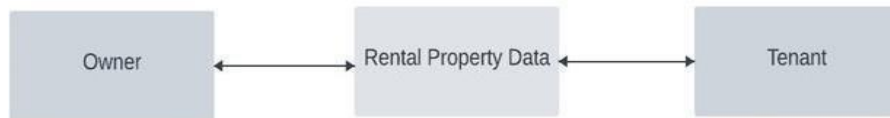


Fig. 5.2 Data Flow Diagram for level-0

In Data Flow Diagram Level 0, the complete system is shown as a single bubble. It is sometimes referred to as the core system model, with input and output data marked by incoming and exiting arrows. The system components that each of these bubbles represents are then broken down and documented as ever-more-detailed DFDs. In the above Data Flow Diagram Level - 0, we have represented our system as a single process and the external entities for our system are both Tenant and Owner.

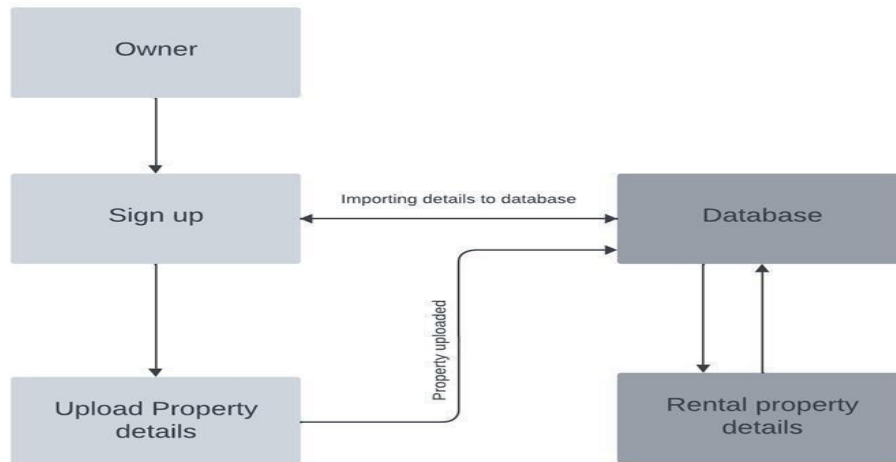


Fig. 5.3 Data Flow Diagram for owner level-1

A context diagram is processes in the Level 1 Data Flow Diagram. At this stage, we draw attention to the system's primary go and deconstruct the high-level DFD process into its component parts. Parts of Level 0 DFD are further processed in Level 1 DFD. It can be utilized to project or document precise/important information about how the system operates. In Level 1 DFD, each of the processes which are shown in the DFD Level 0 Diagram is broken down into sub-processes. In the above Level 1 DFD we have shown all the processes .

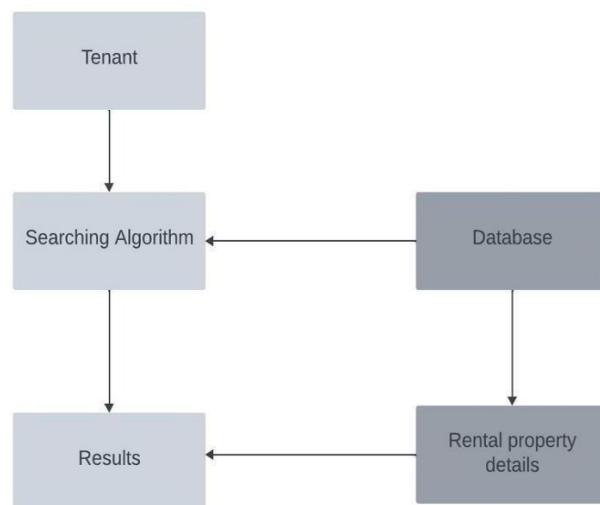


Fig. 5.4 Data Flow Diagram for tenant level-2

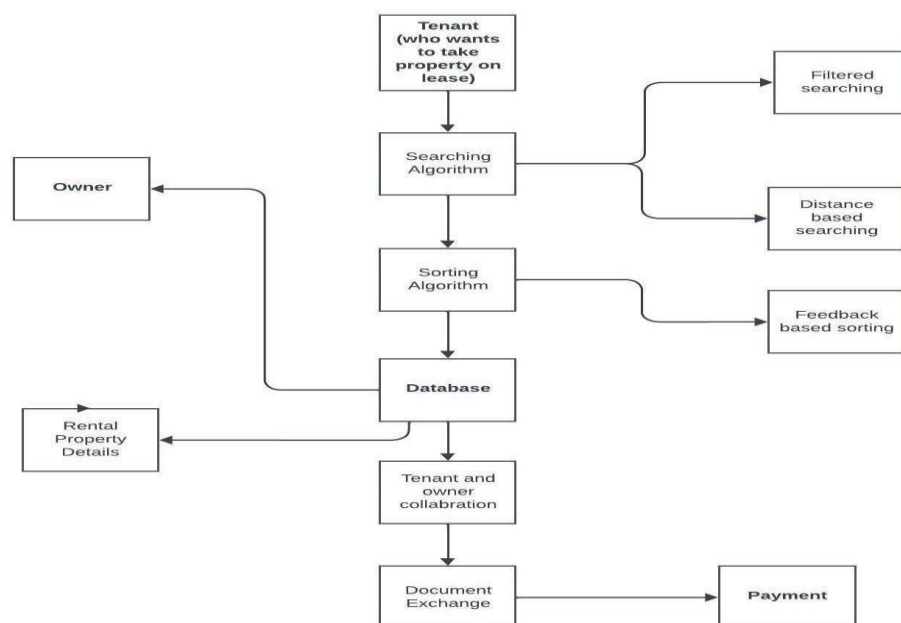


Fig. 5.5 Data Flow Diagram for level-3

The user provides input in the form of their location and preferences. The Search Algorithm processes this input, queries the Rental Property Database, and returns a list of suitable rental properties (Search Results). The user can then review these results and make a selection.

Please note that this is a simplified representation, and in practice, the system would involve more complexity, including user registration, recommendation algorithms, and a more comprehensive database of rental properties.

The data flow in the system for predicting the best rental property based on location and user preferences can be described step by step, from start to end, as follows:

1. User Input:

- The process begins when the user provides input. This input typically includes their location, budget, property type (house, apartment, flat), desired size, and other preferences.
- This user input is considered as the starting point of the data flow.

2. Search Algorithm (linear search):

- The user's input is processed by the Search Algorithm.
- The algorithm creates a search query based on the user's input, including location and preferences.

3. Rental Property Database:

- The Search Algorithm queries the Rental Property Database using the search query.
- The database contains information about available rental properties, such as location, price, size ,amenities, and user reviews.

4. Search Results:

- The Search Algorithm receives the results from the Rental Property Database. These results are a list of rental properties that match the user's criteria.

5. User Review and Selection:

- The Search Results are presented to the user for review.
- The user can then select a rental property from the list of results, based on their preferences and needs.

6. Property Details:

- Once the user selects a rental property, the system retrieves detailed information about the chosen property from the Rental Property Database.
- This information includes specifics like the property's location, price, size, and amenities.

1. User Decision:

- The user reviews the detailed information about the selected property and makes a decision about whether to proceed with renting it.

2. End Result:

- Depending on the user's decision, the data flow reaches its endpoint.
- If the user decides to rent the property, the process may involve further steps, such as completing a rental agreement and payment. If not, the process ends with the user's choice.

5.3 UML DIAGRAMS

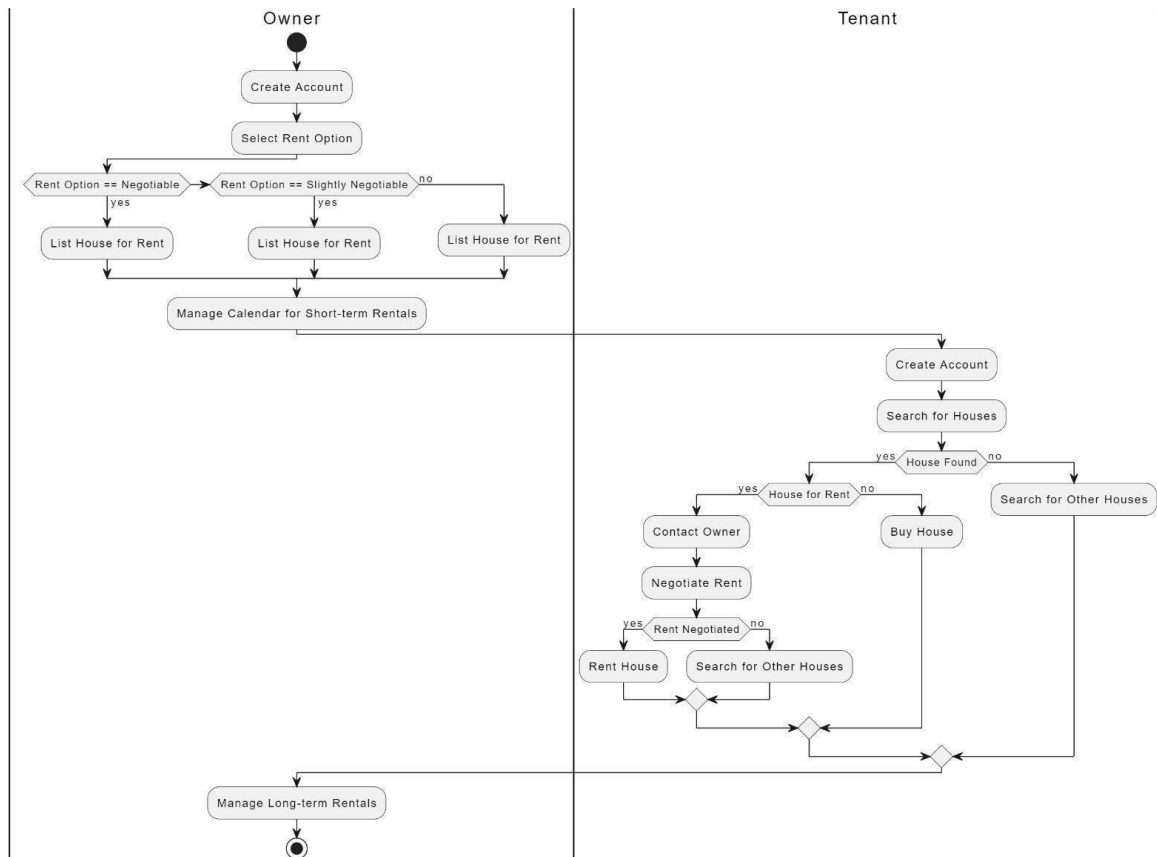


Fig 5.6 State Diagram

Our system has 2 actors: Owner and tenant.

CLASS DIAGRAM:

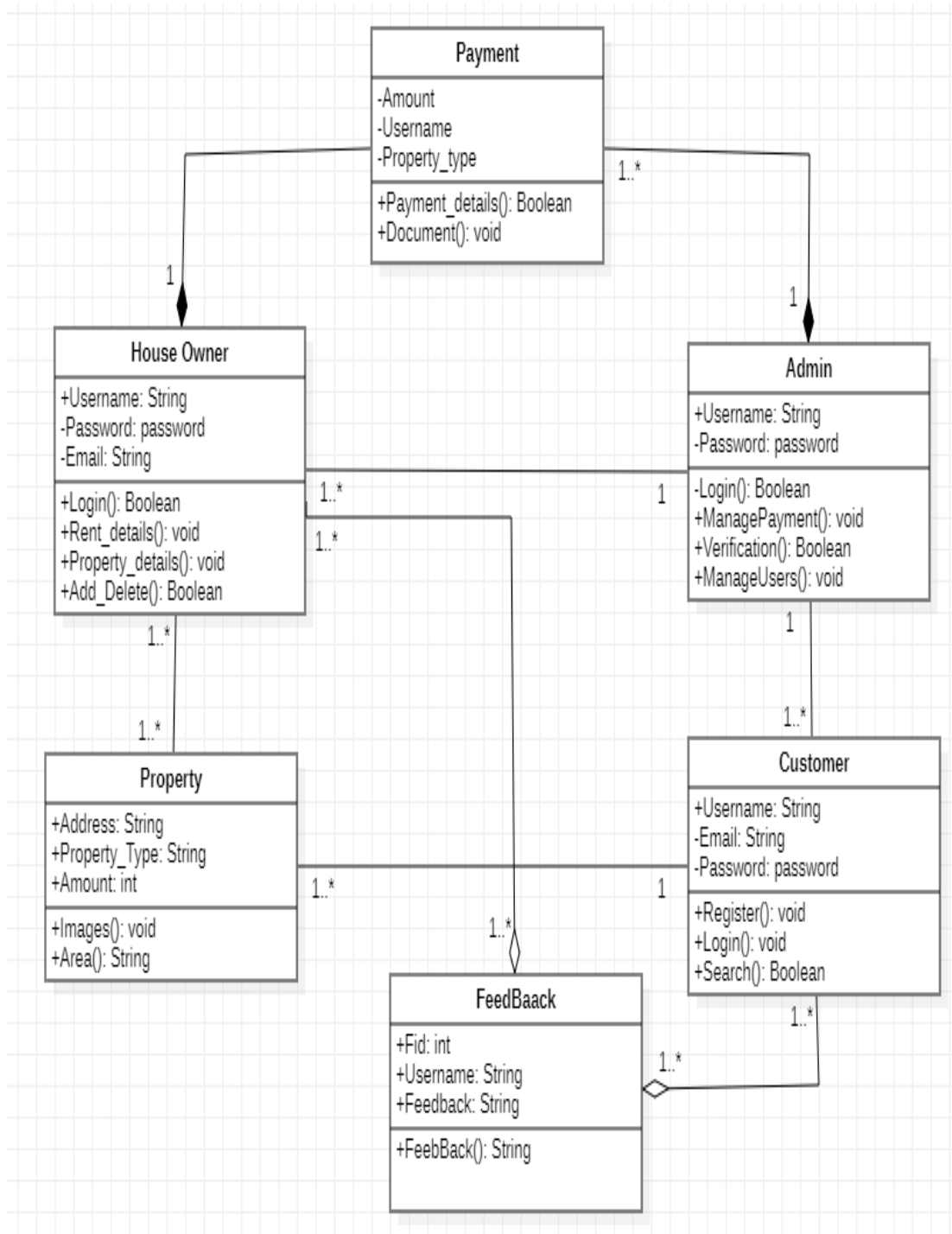
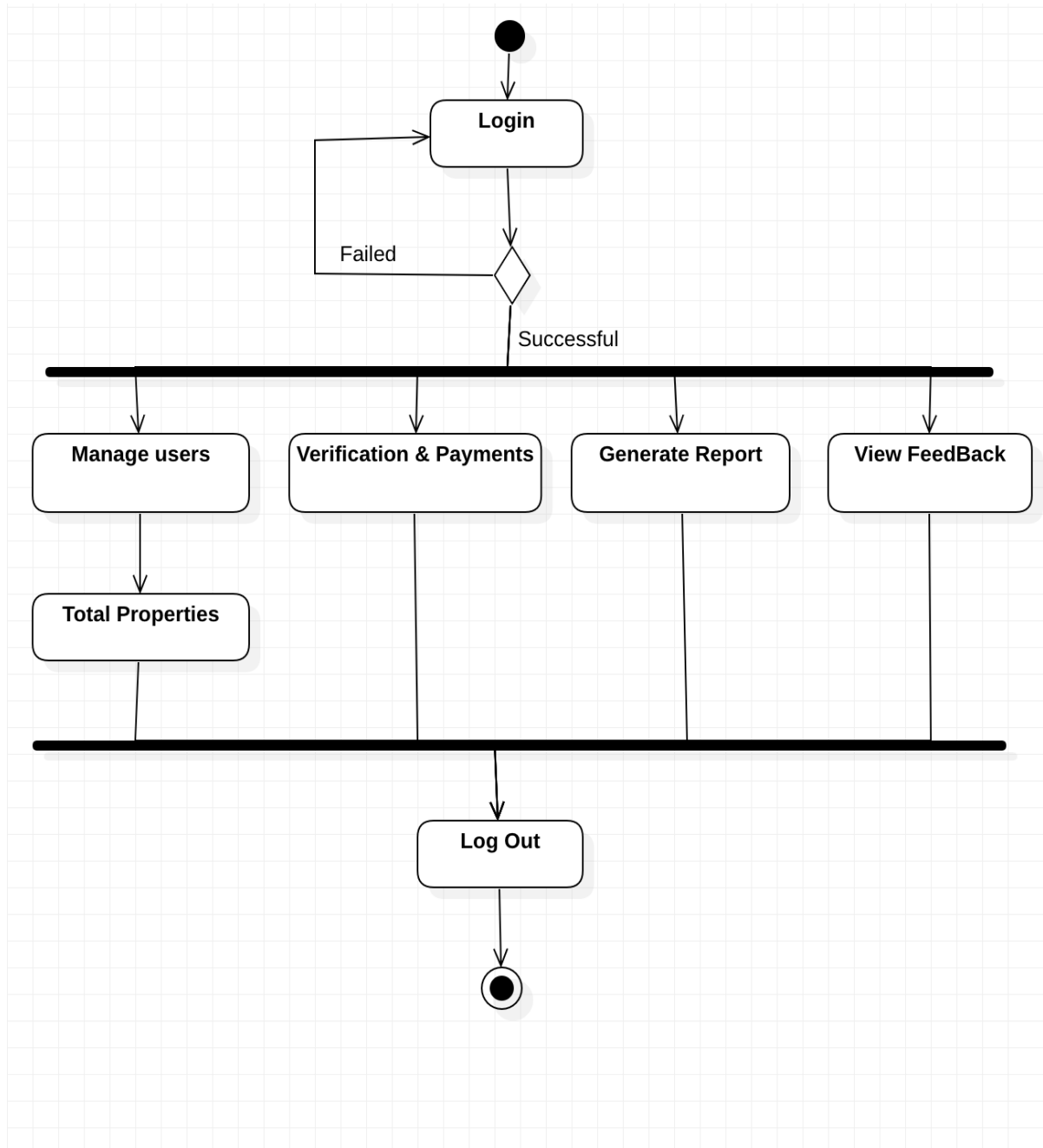


Fig. 5.7 Class Diagram

STATEDIAGRAM:

Fig 5.8 Acitivity diagram for admin



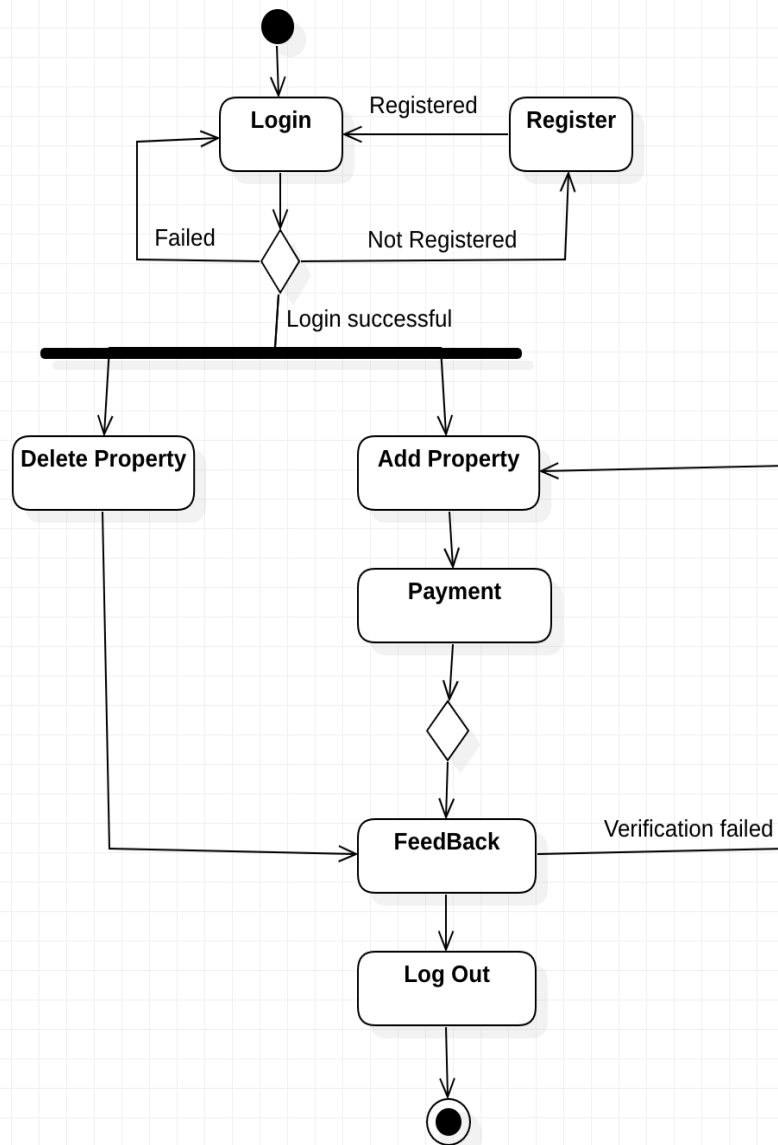


Fig 5.9 State diagram for Owner

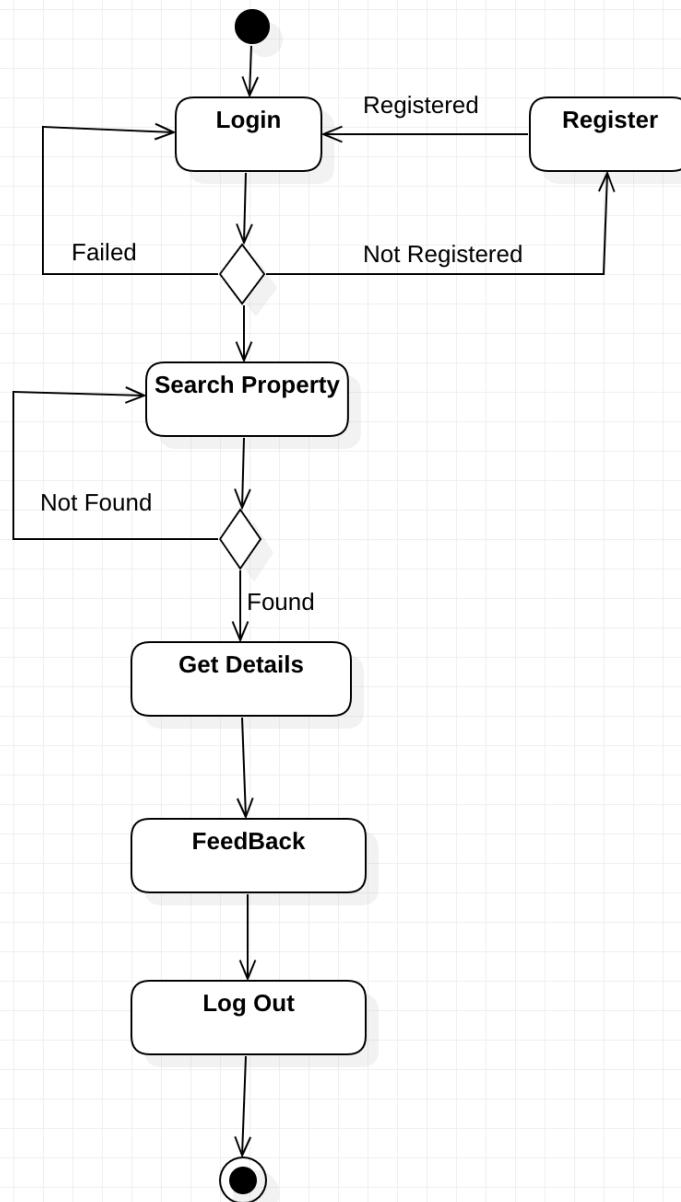


Fig 5.10 State Diagram for Tenant

Another crucial UML diagram for describing the system's dynamic elements is the activity diagram. An activity diagram is essentially a flowchart that shows how one action leads to another. The action might be referred to as a system operation.

One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control.

SEQUENCE DIAGRAM:

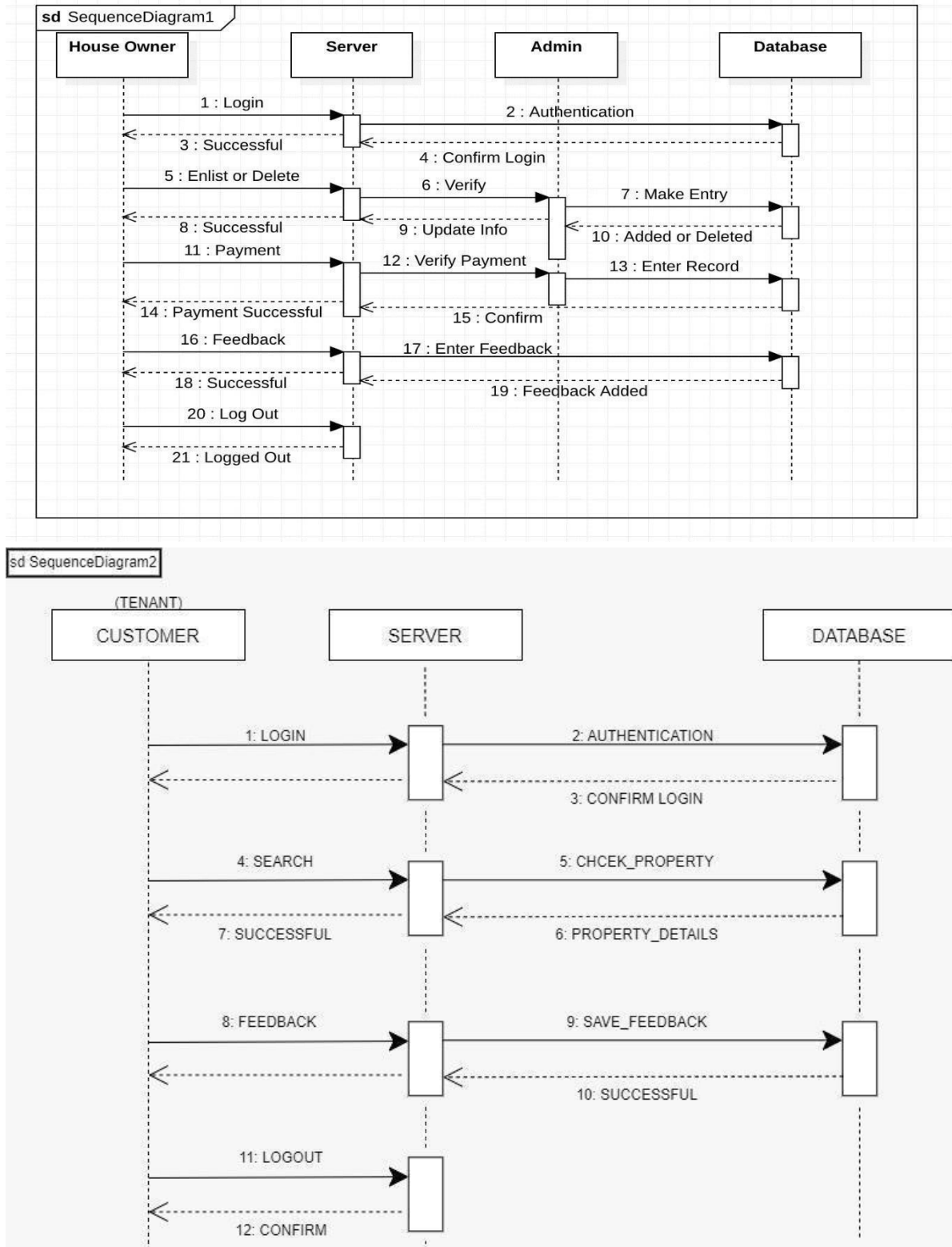


Fig 5.11 Sequence Diagram

Interaction diagrams called UML Sequence Diagrams describe the steps used to complete an operation. They depict how items interact within the framework of cooperation.

Deployment Diagram :

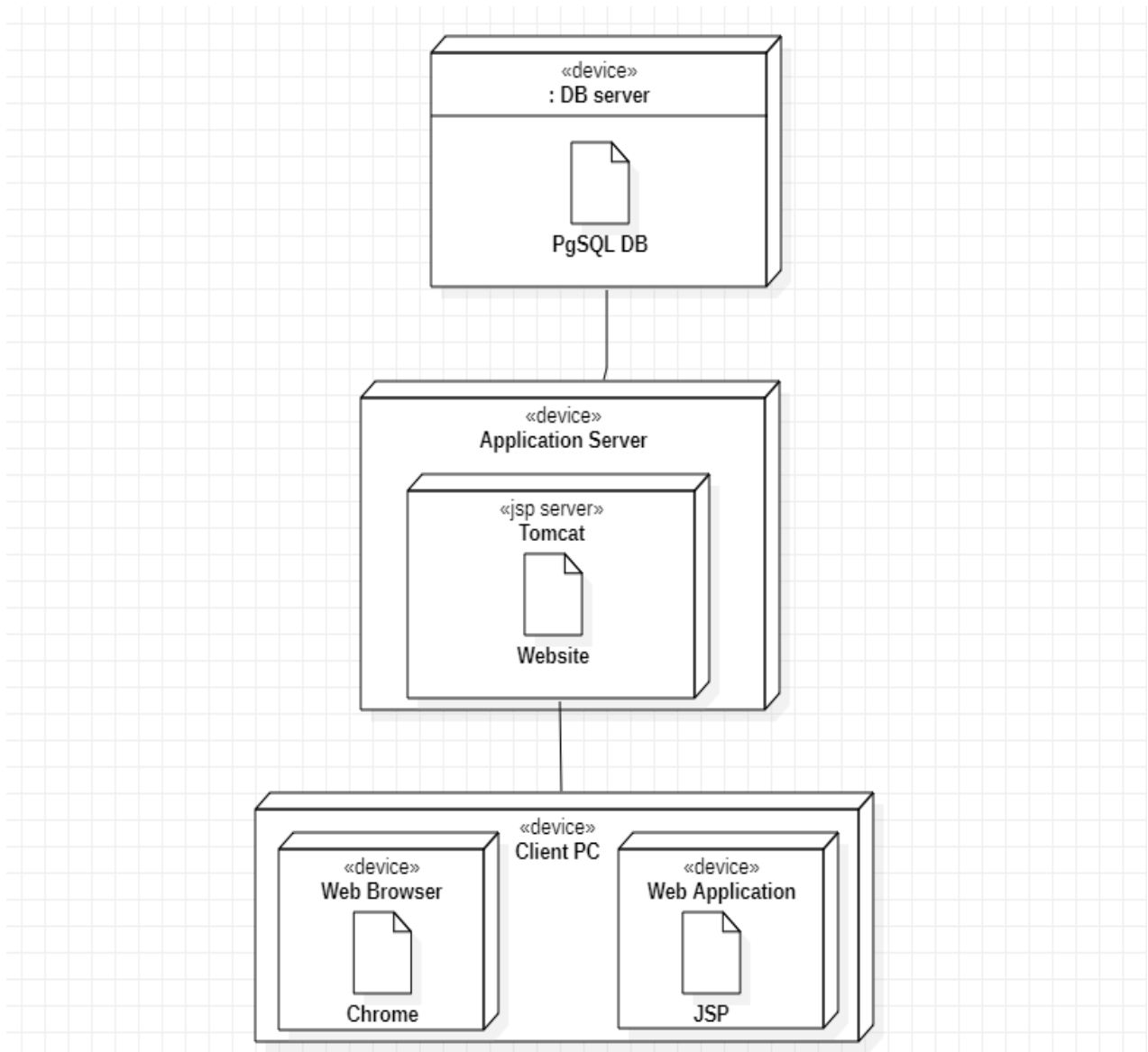


Fig 5.12 Deployment Diagram

Object Diagram :

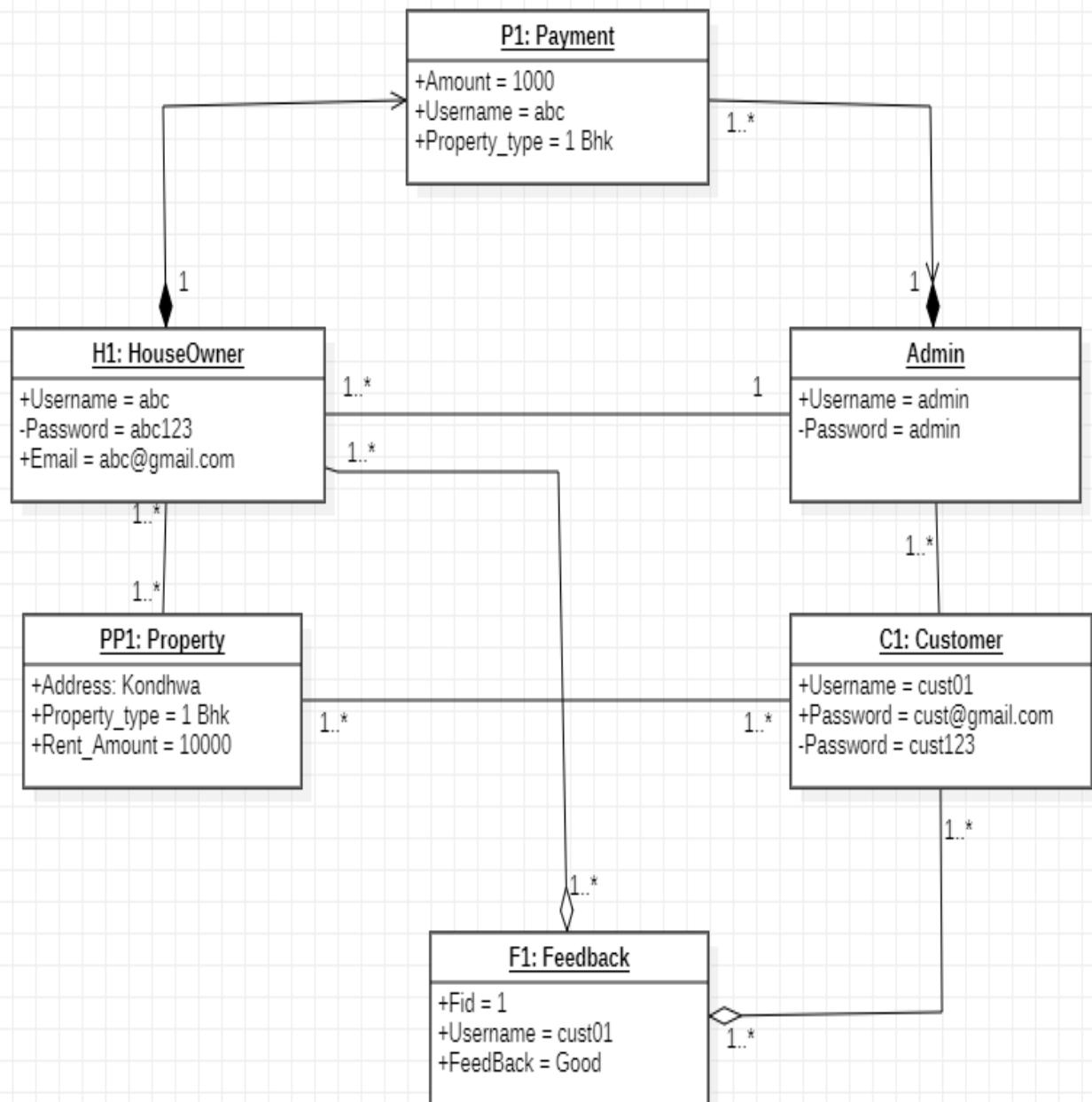


Fig 5.13 Object Diagram

Component Diagram :

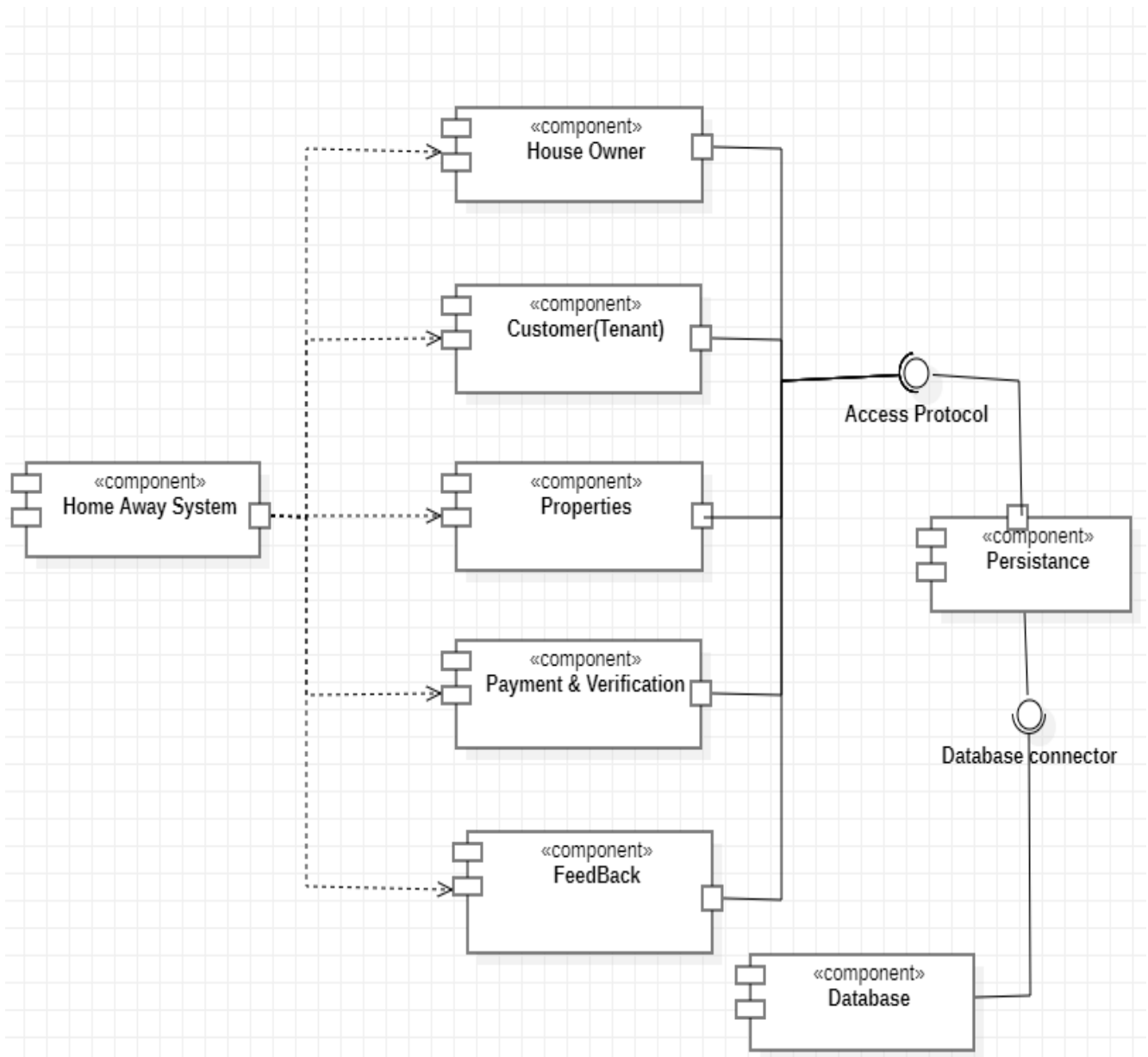


Fig 5.14 Component Diagram

CHAPTER 6
ADVANTAGES DISADVANTAGES
AND APPLICATIONS

Chapter 6: OTHER SPECIFICATION

6.1 ADVANTAGES

1. Provides most up-to-date content
2. Effective than conventional methods
3. Dynamic working of distinct algorithms
4. Accommodates everyone's needs
5. Users can understand the topic without prior knowledge.

6.2 DISADVANTAGES

1. Works only with Internet
2. The system can be more efficient if deployed on distributed servers

6.3 APPLICATIONS

- This application will engage and keep learners motivated with their subjects of study.
- Adding interactive games and mazes with algorithms will result in Gamified Algorithm Learning.
- Students and Corporate Trainees will be able to get knowledge on various algorithms.
- This application will make it easy for trainers or teachers to demonstrate the above-mentioned algorithms.
- This application will help users to better understand the real-world applications of these algorithms.

CHAPTER 7

PROJECT PLAN

CHAPTER 7 : PROJECT PLAN

7.1 PROJECT UNDERSTANDING

"Home Away" is envisioned as a comprehensive rental service catering primarily to the housing needs of students. Understanding the unique requirements and challenges faced by students when seeking accommodation, this project aims to provide a seamless and tailored solution to ensure comfortable living away from home. By offering a range of houses and flats specifically curated for student living, Home Away strives to become the go-to platform for students seeking affordable, convenient, and student-friendly rental options.

Key Objectives:

- **Student-Centric Approach:** Recognizing the distinct preferences and constraints of students, HomeAway prioritizes their needs in every aspect of its service offerings.
- **Diverse Housing Options:** Curate a wide selection of houses and flats, ranging from budget-friendly shared accommodations to premium studio apartments, to cater to various preferences and budgets.
- **Location Convenience:** Ensure that all listed properties are strategically located near educational institutions, public transportation hubs, and essential amenities to enhance accessibility and convenience for students.
- **Safety and Security:** Implement stringent safety measures and conduct thorough background checks to provide students with a secure living environment, fostering peace of mind for both students and their families.
- **Transparent Pricing:** Adopt a transparent pricing model with no hidden costs or fees, allowing students to make informed decisions based on their financial capabilities.
- **Seamless Booking Process:** Streamline the rental process through an intuitive online platform, enabling students to browse, select, and book their preferred accommodations effortlessly.
- **Responsive Customer Support:** Offer dedicated customer support services to address any queries, concerns, or issues promptly, ensuring a smooth and satisfactory experience users .

- **Community Engagement:** Foster a sense of community among tenants by organizing social events, facilitating interactions, and providing access to shared spaces and amenities to encourage collaboration and networking opportunities.
- **Project Deliverables: Comprehensive Market Research:** Conduct extensive research to identify the prevailing housing trends, preferences, and pain points among student renters.
- **Property Acquisition and Listing:** Acquire a diverse portfolio of rental properties and meticulously curate listings on the Home Away platform, highlighting key features, amenities, and proximity .
- **Website and Mobile Application Development:** Develop a user-friendly website and mobile application with intuitive navigation, advanced search filters, secure payment gateways, and interactive features to enhance the overall user experience.
- **Marketing and Promotion:** Launch targeted marketing campaigns across various channels, including social media, online forums, and student communities, to raise awareness and attract potential tenants.
- **Partnership and Collaboration:** Establish strategic partnerships with educational institutions, student organizations, and local businesses to expand reach, enhance credibility, and unlock exclusive benefits for Home Away users.
- **Continuous Improvement:** Gather feedback from users and stakeholders to identify areas for improvement and implement necessary enhancements to optimize the platform's functionality, reliability, and user satisfaction.
- By aligning its offerings with the specific needs and preferences of student renters, Home Away aspires to redefine the rental experience, making it more convenient, secure, and enjoyable, ultimately becoming the ultimate housing solution for students seeking a home away from home

TIMELINE CHART



7.1.1 PROJECT TASK SET

To schedule a project, we need to look after three main points which will decide the direction of our project.

- Work to be done.
- Time span of that project.
- Team/people assigned to it.

Once the project has been started it is important to stick to it and follow the schedule because if you don't do so then there is no point in planning the project. For this you need to schedule tasks and execute them. And scheduling people is as important as scheduling tasks. Because ultimately it's people who execute the tasks. There are many software's which helps you to schedule your tasks and also keep track of it on specific time basis. Many of these software's use Gantt Chart which schedule teams assignments and tasks.

7.2 TASK NETWORK

An implementation methodology must be well documented. The documentation should be content and audience-specific and normally comes in the form of:

1.Procedure – An overview of the methodology, its phases, milestones and deliverables. This type of documentation also consider templates that help promote efficiencies and streamline the implementation process. This type of documentation is aimed at the project team.

2.Technical – Describes the technical & installation requirements for the information system. This type of documentation is purposed at a technical audience.

3.End-User – Refers to manuals for the end-user, system administrator and support staff. They include resources like training manuals, training aides (PowerPoint slides) and automated training material (such as animations).

7.3 RISK MANAGEMENT

Risk is related with almost everything we do and is definitely associated with software implementation projects. A risk is something that “may happen”, implying a probability of less than 100%, and if it does transpire, will have an adverse impact on the project. If it has a probability of 100%, in other words, it occurs – then it becomes an issue. Such an issue is handled differently to a risk. An impressive methodology approach addresses risk management in four stages:

7.3.1 Risk Identification

Risk identification is a critical step in the risk management process for the algorithm visualization project. It involves identifying and documenting potential risks that may have an impact on the project's objectives, timeline, resources, or deliverables. By thoroughly understanding and documenting these risks, the project team can develop appropriate mitigation strategies to minimize their impact.

During the risk identification phase, the project team should consider various sources of risks, including:

1. **Technical Risks:** Risks related to the implementation and functionality of the visualization algorithms, such as algorithm complexity, compatibility issues, or performance limitations.
2. **Operational Risks:** Risks associated with the operational aspects of the project, such as inadequate data or inputs, lack of user understanding, or insufficient testing and validation processes.
3. **Schedule Risks:** Risks that may impact the project timeline, such as delays in development, unexpected dependencies, or changes in requirements

7.3.2 RISK ANALYSIS

- **Market Volatility:**

Risk: Fluctuations in the real estate market, changes in student demographics, or economic downturns could impact demand for rental properties.

Mitigation: Conduct thorough market research to understand trends and anticipate potential shifts. Diversify property portfolio to mitigate the impact of market volatility.

- **Legal and Regulatory Compliance:**

Risk: Non-compliance with local housing regulations, tenant rights laws, or data privacy regulations could lead to legal repercussions and damage the company's reputation.

Mitigation: Stay updated on relevant laws and regulations. Consult legal experts to ensure all operations and contracts adhere to legal requirements.

- **Property Maintenance and Safety:**

Risk: Inadequate maintenance or safety issues in rental properties may lead to accidents, injuries, or property damage, resulting in liability claims and tenant dissatisfaction.

Mitigation: Implement rigorous property inspections and maintenance schedules. Invest in safety features and ensure compliance with building codes and safety standards.

- **Financial Risks:**

Risk: Insufficient funding, unexpected expenses, or payment defaults from tenants could lead to financial strain and hinder business operations.

Mitigation: Maintain a robust financial plan with contingency funds. Conduct thorough credit checks on tenants and enforce strict payment policies to minimize defaults.

- **Technology Risks:**

Risk: Cybersecurity breaches, system failures, or technical glitches in the website or mobile application could compromise user data, disrupt services, and erode trust.

Mitigation: Implement robust cybersecurity measures, such as encryption protocols and regular security audits. Have backup systems in place to ensure continuity of operations.

- **Reputation Management:**

Risk: Negative reviews, poor customer experiences, or incidents of tenant dissatisfaction could damage the company's reputation and deter potential renters.

Mitigation: Prioritize customer satisfaction and promptly address any issues or complaints. Establish transparent communication channels and actively seek feedback to identify and resolve issues proactively.

- **Competition:**

Risk: Intense competition from existing rental platforms or new market entrants could impact market share and revenue generation.

Mitigation: Differentiate the Home Away brand through unique value propositions, exceptional customer service, and targeted marketing strategies. Continuously innovate and adapt offerings to stay ahead of competitors.

- **Natural Disasters and External Events:**

Risk: Natural disasters, pandemics, or other external events beyond the company's control could disrupt operations, damage properties, or affect rental demand.

Mitigation: Develop contingency plans and emergency response protocols. Ensure properties are adequately insured against natural disasters and other unforeseen events.

By proactively identifying and addressing potential risks, Home Away can effectively mitigate threats and safeguard its business operations, ensuring a resilient and successful venture in the student rental market.

7.4 ORGANIZATION

This project was divided into 4 modules so one team had to be assigned for each module. The team was chosen based on the interests of the students. As a team responsibilities and duties were assigned to each person. Initially everyone's job was to do research and based on the foundation of that research the further development was of project started.

CHAPTER 8

PROJECT IMPLEMENTATION

CHAPTER 8 : PROJECT IMPLEMENTATION

8.1 PROJECT MODULES

- 1) Log In page : where users can sign in in their accounts created.
- 2) Sign Up/ Registration page : where new user can register themselves using Aadhar card .
- 3) Home Page : Home page where you will found the about and service of the project.
- 4) Feedback Page : Users can give review .
- 5) Property Page : Properties will be enlisted here.
- 6) Search Page : According to need user can search properties .
- 7) Service Page : where there will be several services enlisted and contacts will be given for users .

8.1 ALGORITHM DETAILS

8.1.1 WHAT IS AN ALGORITHM?

We must ensure that the fundamentals of algorithms are understood before continuing on to the further sections, which describes algorithms and the software. The definition of the algorithm will be our first step.

We can simply define an algorithm as a series of steps or instructions that addresses a certain problem. Problems are some tasks that need to be solved. We face many different problems everyday like :- quickest way to work or home etc. However, all these problems may not fit our algorithm definition. Problems need to be specified with its inputs and all the inputs should have an output. Solving a problem means producing an output for all the inputs.

8.1.2 CHARACTERISTICS OF ALGORITHMS

We defined an algorithm as a series of steps or instructions that can solve a problem. However, not any series of steps will be able to solve a problem. A well defined algorithm will have the following characteristics:

- **Input:** An algorithm should have zero or more well defined inputs.
- **Output:** An algorithm should produce some output which are related to the inputs.
- **Definiteness:** It means that the steps required to solve the problem must be specified in proper sequence to produce the desired output.
- **Finiteness:** An algorithm should completely finish executing in a finite amount of time.
- **Unambiguity:** An algorithm should be unambiguous which means that each instruction of the algorithm should only have a single meaning.
- **Effectiveness:** The steps required to produce the output should be feasible with the available resources. It should not contain any steps which are not required which may make the algorithm ineffective.

8.1.3 TYPES OF ALGORITHMS

Currently a large number of algorithms exist. However, many of them can be classified into the following different types of algorithms:

- **Recursive Algorithm:** A recursive algorithm calls itself repeatedly until a problem is solved. In these algorithms, the input becomes smaller and smaller as the algorithm calls itself. Some common problems which are solved using this approach are: Tower of Hanoi and Searching in a Binary Search Tree.

- **Divide and Conquer Algorithm:** In Divide and conquer algorithm, a large problem is divided into smaller sub problems and then each sub problem is solved individually. After solving each sub problem, the solutions of all the subproblems are combined to form the solution of the whole problem. Merge Sort and Quicksort are major Divide and conquer algorithms.
- **Dynamic Programming Algorithm:** Some algorithms may take a large amount of time to execute completely. To avoid this and reduce the time taken by the algorithm to execute completely, more space is used by the algorithm in order to store the already calculated results. In Dynamic Programming Algorithm, additional space is used to store the results of already calculated sub problems for future use. This reduces the time required to calculate the same results again. problem, locally optimal solution is found at various steps of the algorithm. Popular Greedy algorithms include Huffman Coding and Dijkstra's algorithm.
- **Brute Force Algorithm:** This is the simplest of all the algorithms. It generates all the possible solutions and to search for one or more possible solutions of the given problem. Linear Search is an example of Brute Force Algorithm.
- **Backtracking Algorithm:** It is used to solve a problem in an incremental way. We keep moving forward with the solution until we find the correct solution. If the solution fails we backtrack, i.e. we move backwards, and continue finding the solution from the previous step. N Queens Problem is solved using the Backtracking Approach.

8.2 COMPLEXITY

Complexity of an algorithm refers to the amount of time and space it takes for the algorithms to run till completion for a given input size n . There are two types of complexity for an algorithm: Time complexity and Space complexity.

8.2.1 Time Complexity

Time complexity is the amount of time an algorithm takes to run as a function of the input length. It calculates how long it takes for each algorithm's code statement to run. It won't look at an algorithm's overall execution time. Instead, it will provide details on the variance (increase or decrease) in execution time when an algorithm's number of operations (increase or decrease). Yes, as stated in the definition, the time required depends solely on the length of the input.

8.2.2 Space Complexity

Space complexity is a measure of how much memory an algorithm requires depending on the size of the input. It quantifies the amount of memory space (in terms of additional variables, data structures, etc.) that an algorithm needs to run, and is essential to understanding the algorithm's memory usage and resource requirements.

➤ Importance of Cosmic Complexity:

Understanding the spatial complexity of the algorithm is essential for several reasons:

- **Memory Usage:** Helps us analyze how much memory the algorithm is consuming, which is key to optimizing memory usage, especially in resource-constrained environments.
- **Resource Constraints:** Allows us to evaluate whether an algorithm can run efficiently within the available memory constraints of the system or platform.
- **Performance Analysis:** Complements time complexity analysis by providing a comprehensive understanding of an algorithm's resource requirements, allowing us to make informed decisions about algorithm selection and optimization.

➤ How is the complexity of the universe analyzed?

Space complexity analysis involves determining how an algorithm's memory usage varies with the size of the input. Here are the key steps in analyzing the complexity of a space:

- **Identify additional memory usage:** Identify any additional memory used by the algorithm in addition to the input data itself. This includes variables, data structures, recursion stacks, and any other memory allocations made during algorithm execution.
- **Quantifying memory usage:** Determine the amount of memory used by each additional component of the algorithm as a function of input size. This step often involves analyzing the size and structure of the data structures used by the algorithm, as well as any temporary variables or buffers created during execution.
- **Express Complexity:** Express the space complexity of the algorithm using big O or other suitable notation. Similar to time complexity, space complexity is usually expressed in terms of the most significant term, which dominates memory usage as the input size increases.

➤ Types of Space Complexities:

When analyzing algorithms, we encounter several common types of spatial complexities:

1. Constant Space ($O(1)$): Algorithm's memory usage remains constant regardless of input size. The size of the input data does not matter.
2. Linear space ($O(n)$): The algorithm's memory usage grows linearly with the size of the input. It scales up or down linearly as the input size changes.
3. Quadratic space ($O(n^2)$): Algorithm memory usage grows quadratically with input size. It increases quadratically with increasing input size.
4. Exponential Space ($O(2^n)$): Algorithm memory usage grows exponentially with input size. It increases exponentially as the input size increases.

➤ Example:

Consider the following algorithm for calculating the factorial of a number using recursion:

Python copy the

Def factorial (n) : if n == 0 :

Return n* factorial (n-1)

This algorithm has a space complexity of $O(n)$, where n is the input to the function.

In the worst case, recursive calls to the algorithm consume memory in the call stack, resulting in a linear increase in memory usage with the size of the input. Analyzing the spatial complexity of an algorithm allows us to understand its memory requirements and resource usage. By quantifying additional memory usage as a function of input size, we can assess algorithm scalability, identify potential memory bottlenecks, and optimize memory usage for better performance and efficiency.

CHAPTER 9

TESTING

CHAPTER 9: TESTING

TEST PLAN

A test plan in software testing is a document which outlines the what, when, how, who, and more of a testing project. It contains the details of what the scope of testing is, what the test items are, who will do which testing task, what the items test/pass criteria will be, and what is needed to set up the test environment and much more.

The details of the test plan of the software testing areas follows:

1) Mode: Manual Testing

2) Resource Requirements: Developed System, Tabular structure for updating the test cases.

3) Testing Tasks: Graphical User Interface, Module Functionalities, Data base Connectivity.

4) Type: White Box Testing and Black Box Testing

5) Test Plan Identifier: Home Away,TS_01,TS_02,TC-001,TC-002,TC-003, etc.

6) References :Images folder was taken as a reference folder .It is apart of the system, but externally stored into a folder. “font awesome-free-6.0.0- web” folder for storing icons of the project.

7) Test Items: It is a "test object" or a component/subsystem to be tested. The various test items in our plan would be listed as Username field, Password field, dropdown lists, buttons, etc.

8) Featurestobetested:

- a. User Registration
- b. User Login
- c. GUI of all webpages.
- d. Database connectivity

9) Test Deliverables :Test Cases ,Test Report ,Bug Report.

10) Responsibilities and Schedule:

BLACK BOX TESTING

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

Test Case ID	Test Case Name	Description	Execution Stages	Expected Result	Actual Result	Status
TC - 001	User Interface	Check all the text.	1) Click on dropdowns	UI should be Perfect	UI should be Perfect	Pass

TC- 002	Blank Fields	Check the fields by not filling any data	1) Do not enter any value in the field. 2) Click on the Register button	1) Registration page won't get redirected to the next page.	Registration page won't get redirected to the next page.	Pass
TC- 003	Required_Fields	Check user should Register by filling all the required fields	1) 2) Enter valid values in the required fields. Click on the Register button.	1) User should be registered successfully 2) A successful registration message should show It will get redirected to the login page. 3)	1) Users should be registered successfully. 2) A successful registration message should show. 3) Page gets redirected to Login page	Pass
TC- 004	Username	Check that the username is not blank or duplicate	1) 2) Enter any username. Click on Register button	It should show any validation message	It should show some validation message	Pass

TC - 003	Email_validation	Check Email text field that has an Email address without @ symbol. • Check Email text field that has random string instead of a real email ID.	1) Enter Invalid Email 2) Click the on-Register Button.	It should show incorrect email.	It should show incorrect email.	Pass
TC- 004	Email_validation	Check All the Valid Emails.	1) Enter valid Emails. Click 2) on the Register Button.	It should show some validation message	It should show some validation message	Pass
TC- 005	Password_validation	Check the password limit when enter value less than min At least 8	1) Enter value which is 2) less than 8. Click on Register button	It shows Error message.	It shows Error message.	Pass

TC- 006	Password_validation	Check the password limit when enter value greater than max	1) Enter value which is more than 12. 2) Click on Register button	It shows Error message.	It shows Error message.	Pass
TC- 007	Password_validation	Check the password is Between 8 to 12	1) Enter value between 8 and 12. 2) Click on Register button	It should show some validation message	It should show some validation message	Pass

Here, mainly the testing of the registration page will be carried out by black box testing.

WHITE BOX TESTING

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing. Here, mainly the testing of the login page will be carried out by white box testing.

Test Case ID	Test Case Name	Description	Execution Stages	Expected Result	Actual Result	Status
TC - 001	User Interface	Check all the text.	1) Click on dropdowns	UI should be Perfect	UI should be Perfect	Pass

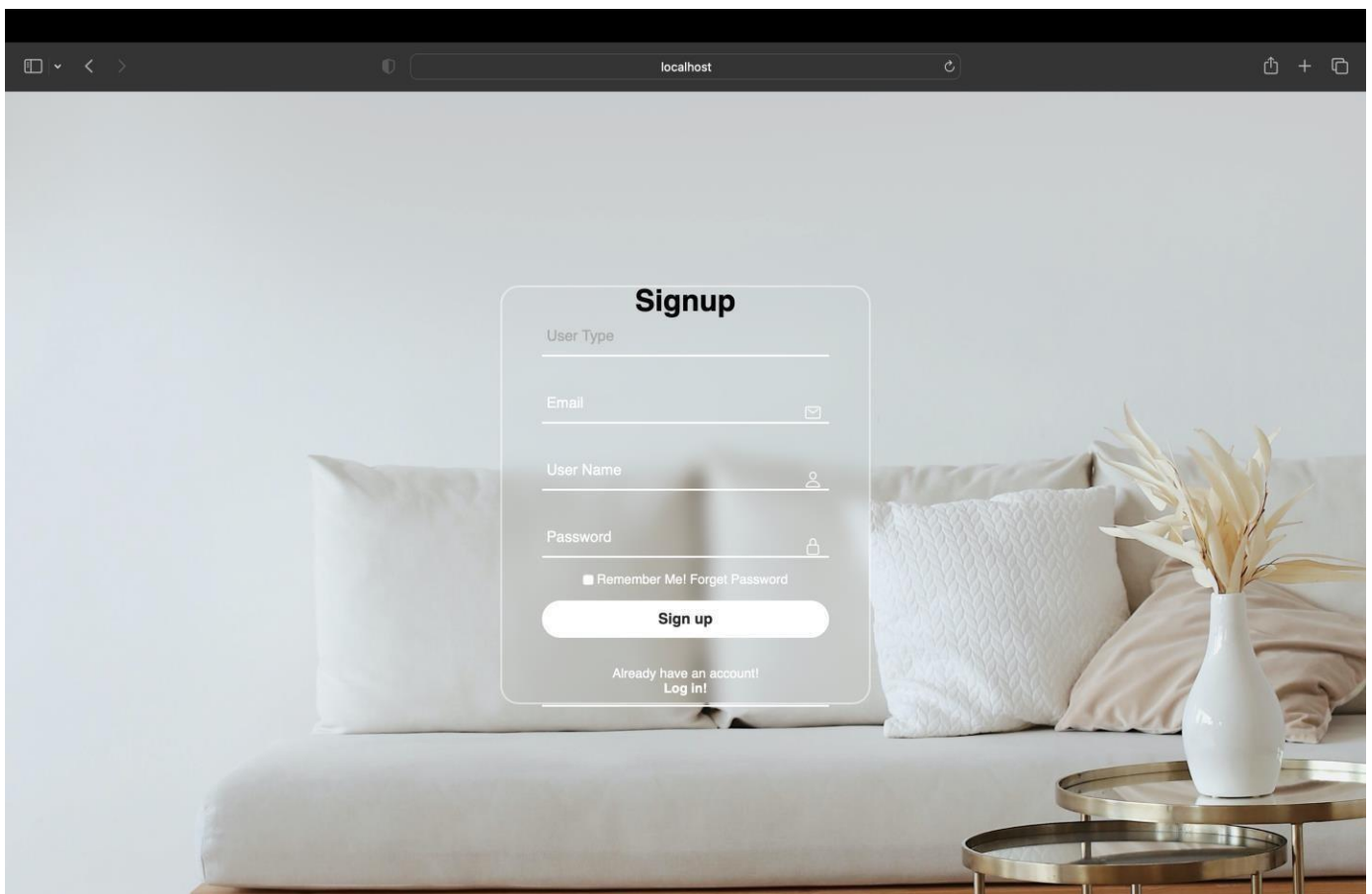
Test case ID	Test Case Name	Description	Execution Stages	Expected Result	Actual Result	Status
TC 008	User Interface	Check all the text boxes	Click on all buttons and dropdowns	UI should be perfect. Text boxes should be aligned.	UI is perfect. Text boxes are aligned.	Pass

TC 014	Signup	Check whether the register link for the new user is working	Click Register link	Clicking register link takes the user to signup page successfully	Clicking register link took the user to signup page successfully	Pass
TC 015	HomeOwner_Login	Check when type of user is admin and Selected user is home owner	1) Enter valid username 2) Enter incorrect password 3) Select Home owner 4) Click on Login Button	User should not log in and should show proper error message	User should not log in and should show proper error message	Pass
TC 016	Admin_Login	Check when type of user is Distributor and Selected user is Admin	1) Enter valid username 2) Enter incorrect password 3) Select Admin 4) Click on Login Button	User should not log in and should show proper error message	User should not log in and should show proper error message	Pass
TC 017	User_Login	Check when type of user is correct	1) Enter valid username 2) Enter incorrect password 3) Select correct type of user 4) Click on Login Button	User should log in	User should log in	Pass

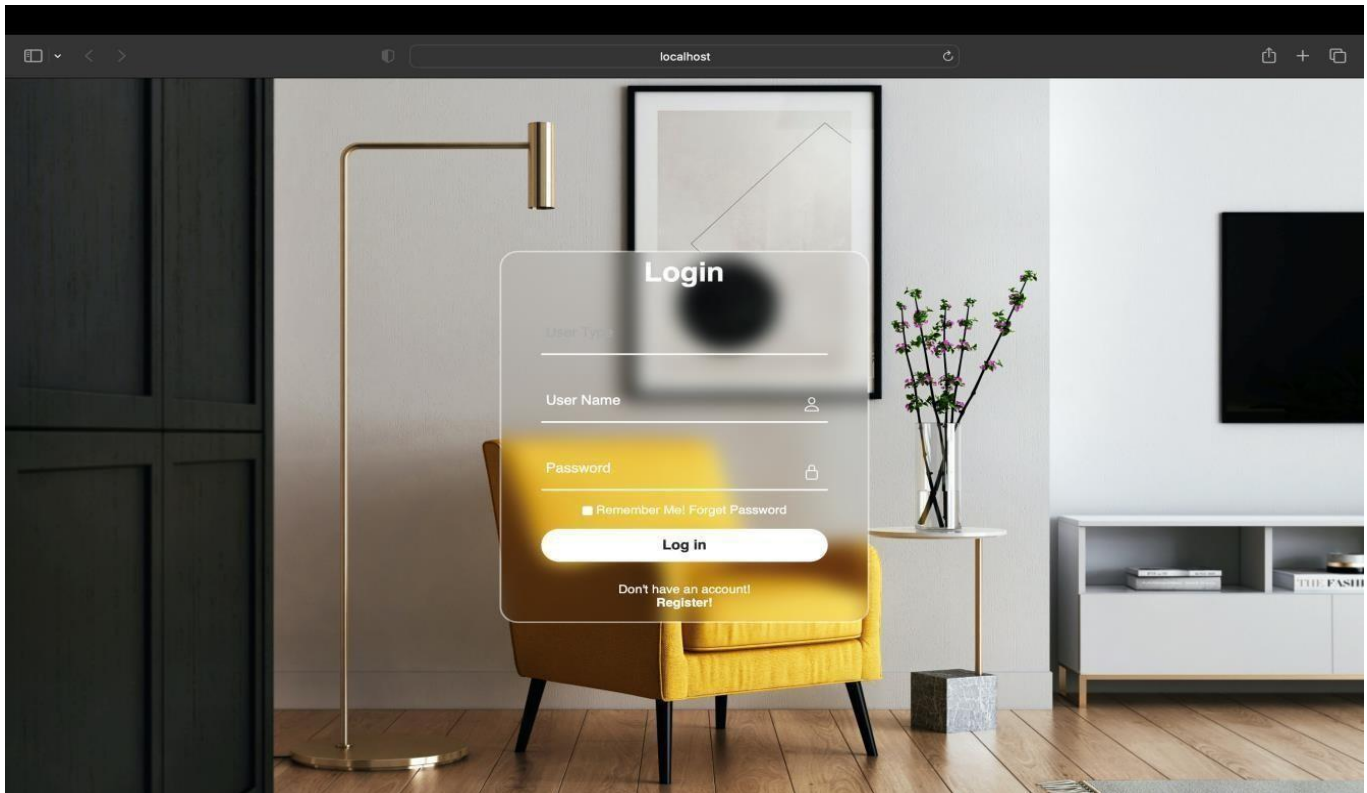
Chapter 10: Results

10.1 Snapshot of Outcome(Execution)

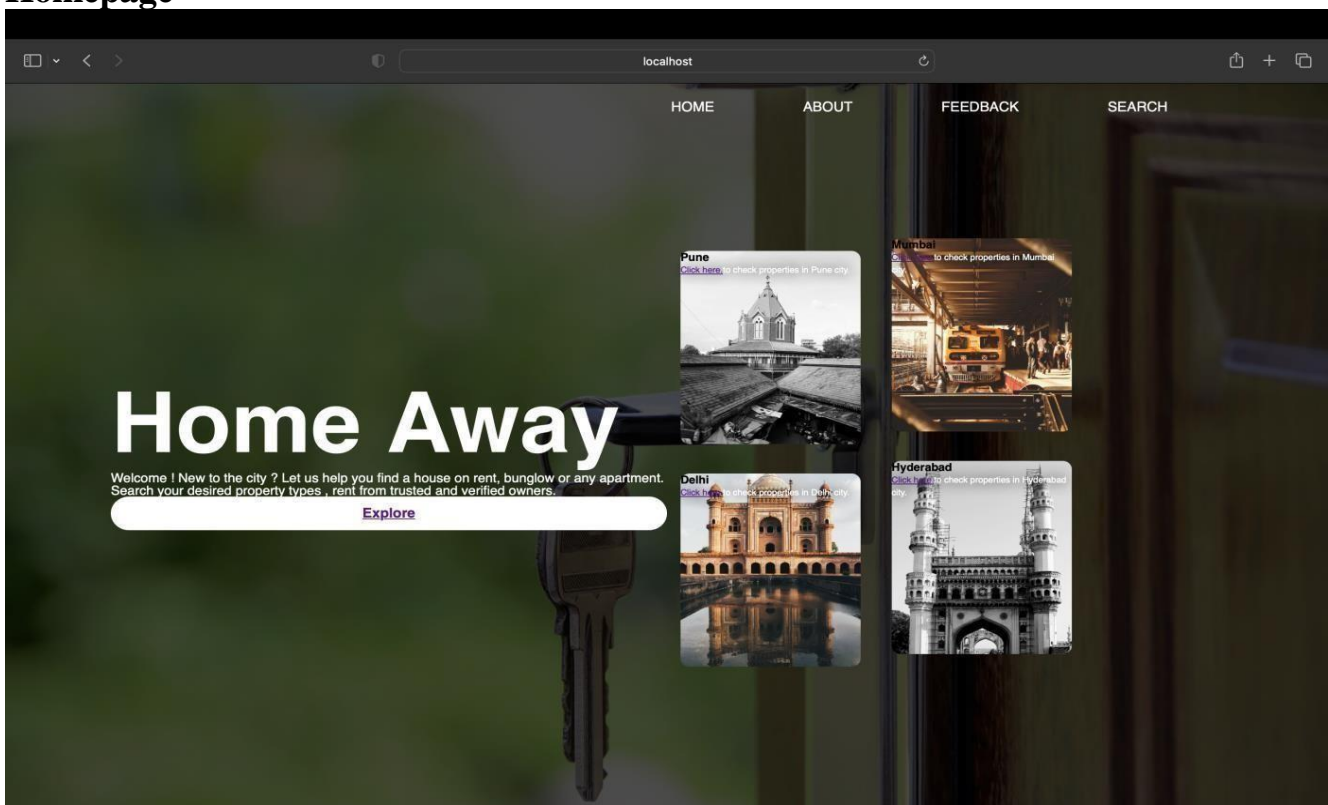
Registration



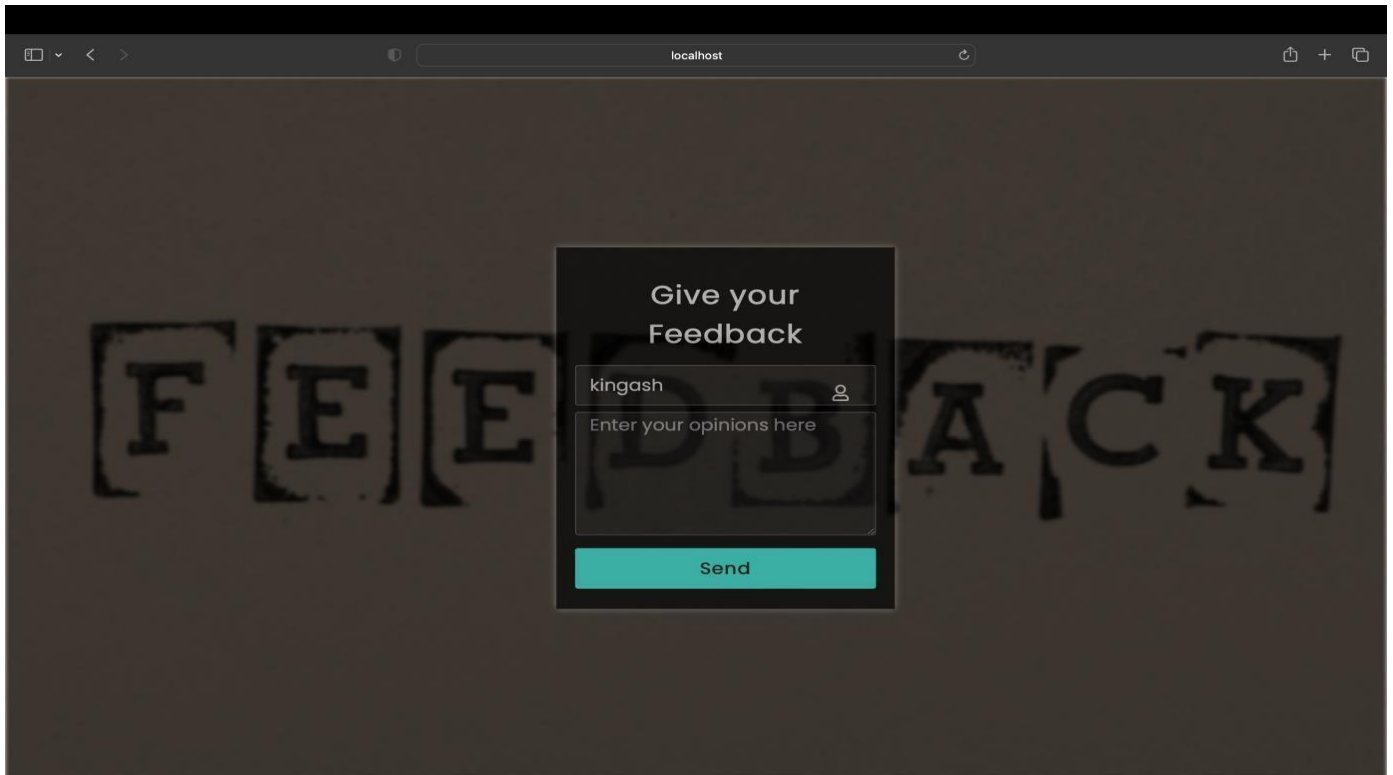
Login



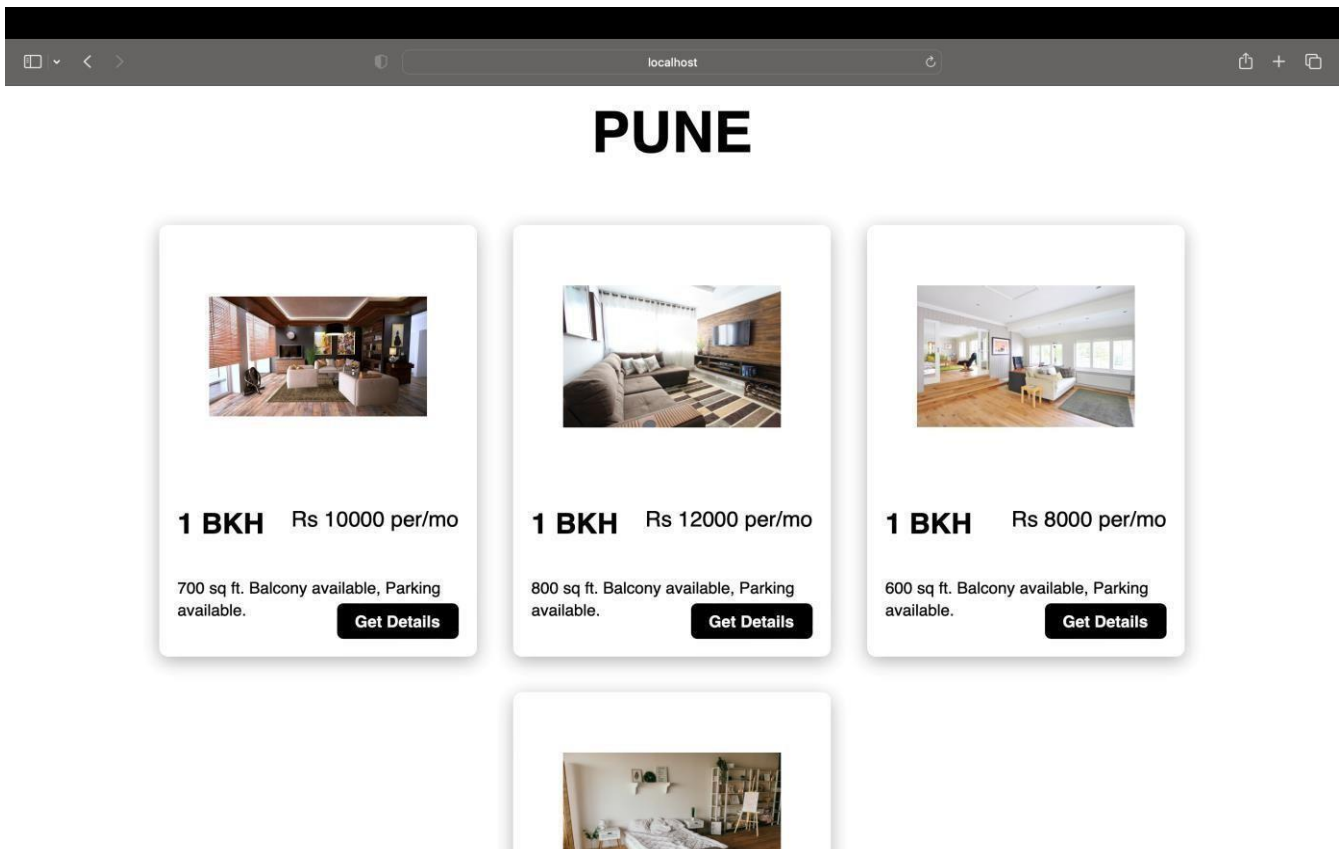
Homepage



Feedback



Property page



Search Page

Select location

Select Type

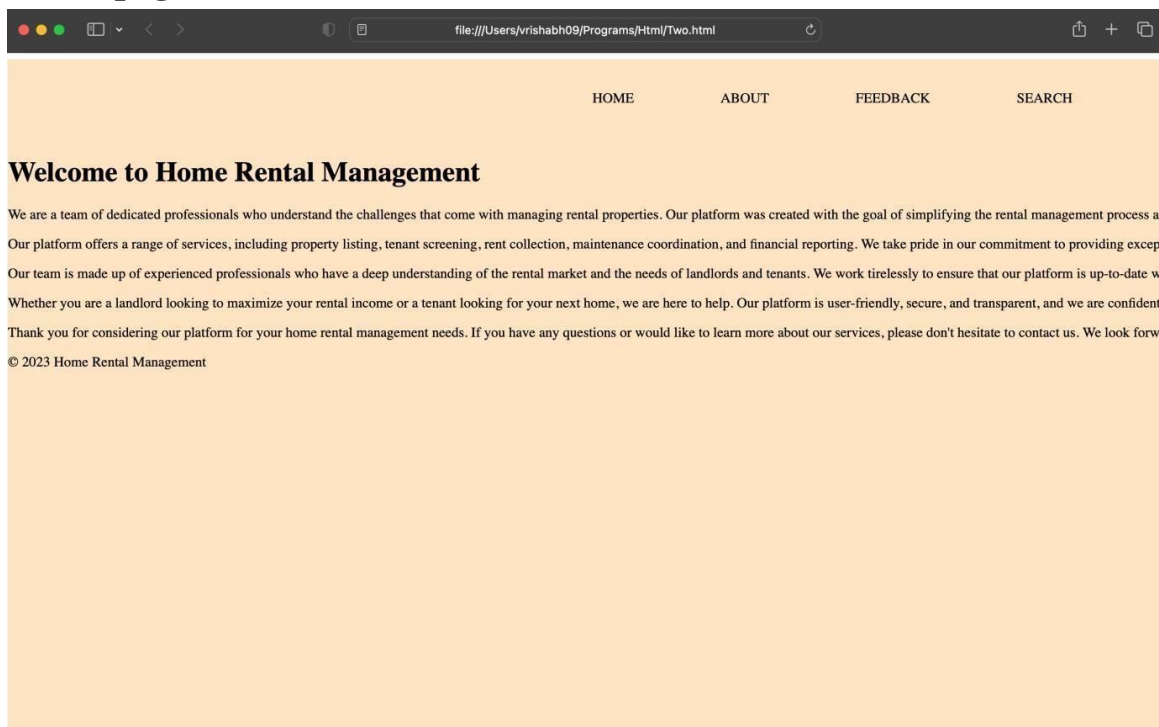
Select Price

Back

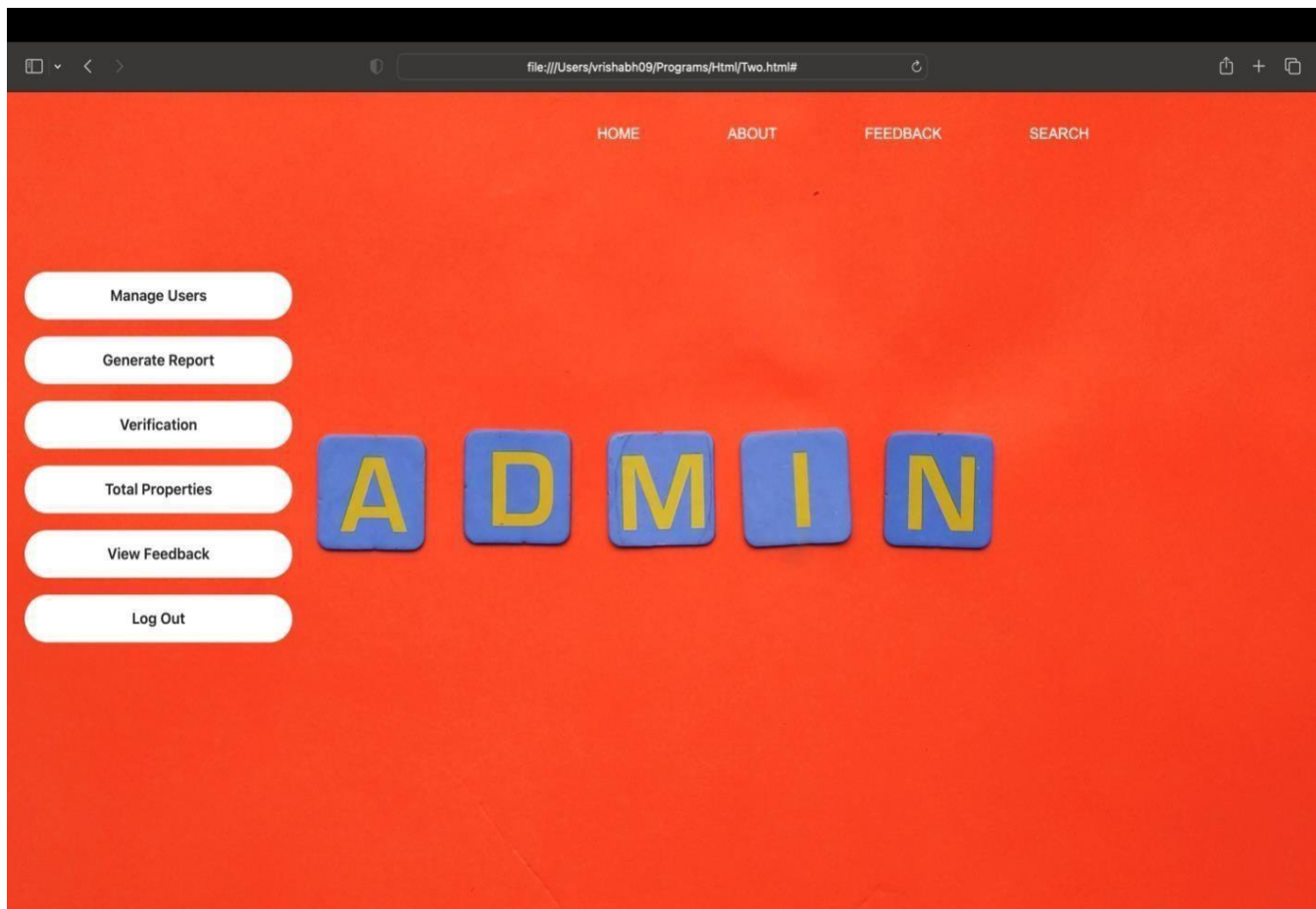
Apply

Name	Type	Location	Price	Jump
Patil	3BHK	Sahakar	200000	<div>Go</div>

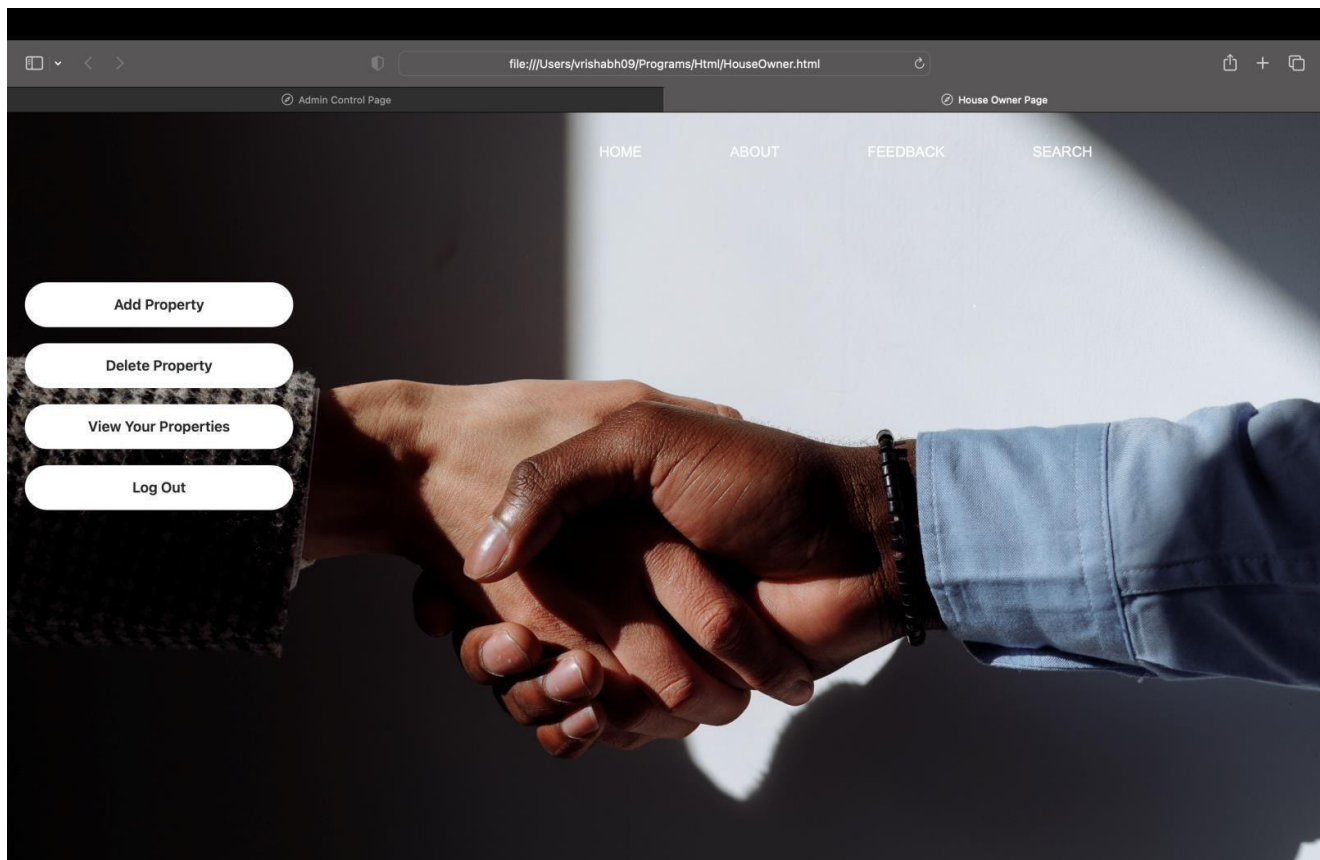
About page



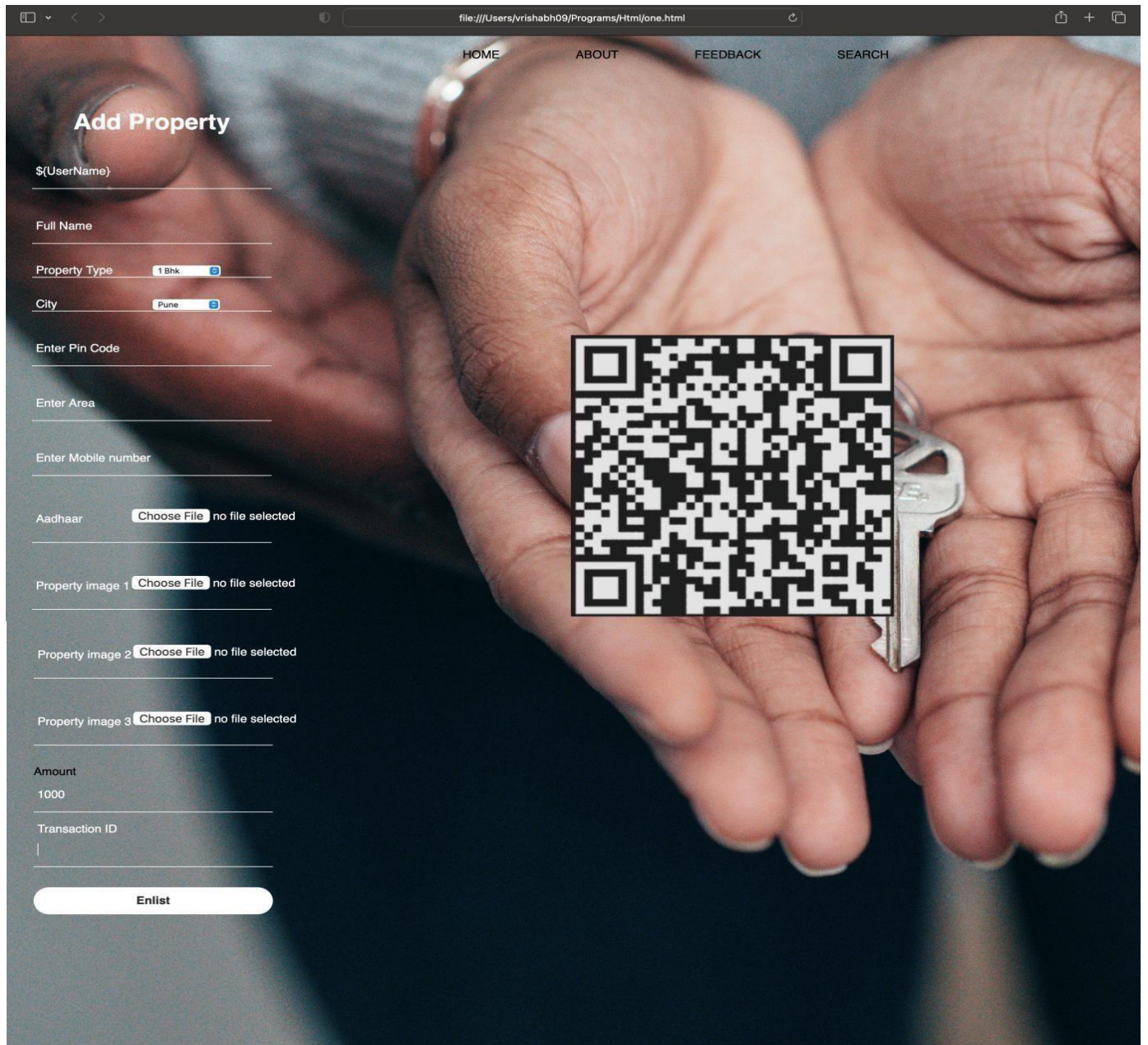
Admin Page



House owner Interface



Add Property Page



file:///Users/vrishabh09/Programs/Html/one.html

HOME ABOUT FEEDBACK SEARCH

Add Property

\$(UserName)

Full Name

Property Type 1 Bhk

City Pune

Enter Pin Code

Enter Area

Enter Mobile number

Aadhaar Choose File no file selected

Property image 1 Choose File no file selected

Property image 2 Choose File no file selected

Property image 3 Choose File no file selected

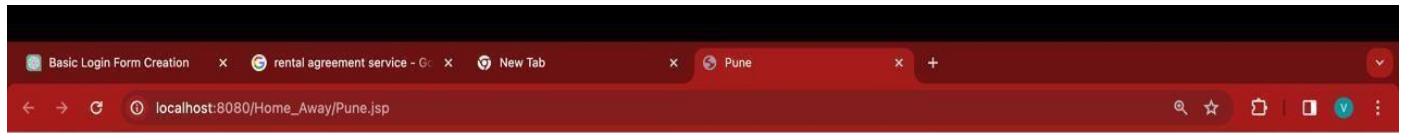
Amount

1000

Transaction ID

Enlist

Search page results



[Home](#) [About Us](#) [Feedback](#) [Search](#) [Services](#)



PUNE



1Bhk Rs12000 per/mo

700 sq ft. Balcony available, Parking available.

[Get Details](#)

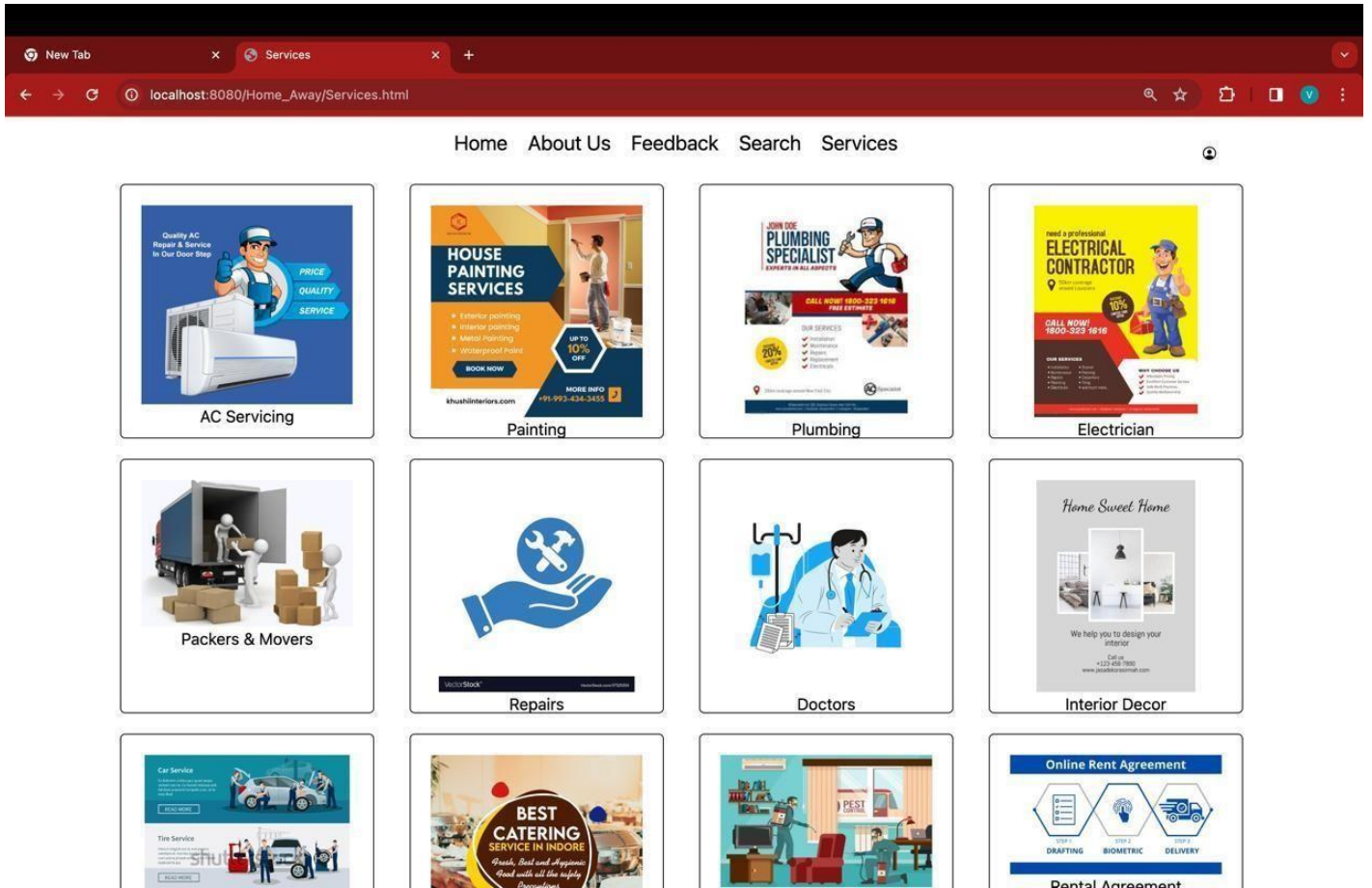


RowHouse Rs20000 per/mo

700 sq ft. Balcony available, Parking available.

[Get Details](#)

Services



Total users page

House

Name	Type	Location	Username	Price
Patil_Wada	3BHK	Sahakar Nagar	bob123	200000
Uttam_Townscape	3BHK	Yerawada	bhakti0205	150000
Sonigara_Township	3BHK	PUNE	bob123	100000
Skytree	2BHK	Koregaon Park	ketan123	100000
Lunkad	1BHK	Viman_Nagar	ketan123	300000
Peter_Wada	Bungalow	Vadgaon	ketan123	1500000

Customer

Name	Username	Password	Email	Phone
Omkar Nagnure	omkar1234	Osn@12345	omkarnagnure98@gmail.com	8390829546
Shruti	shruti123	Shruti@123	shruti@gmail.com	4354627897
Bhkati	bhakti123	Bhakti@1234	bhakti@gmail.com	4356789043

House Owners

Name	Username	Password	Email	Phone
Omkar Nagnure	osn123	123	omkarnagnure98@gmail.com	08390829546
Bob	bob123	bob12	bob@gmail.com	123456
Bhakti	bhakti0205	0987	bhakti@gmail.com	3245665472
Omkar Nagnure	hbbih	ybhub	omkarnagnure98@gmail.com	8390829546
Ketan	ketan123	Ketan@123	ketan@gmail.com	8390829546

[Back](#) [Print](#)

CHAPTER 11

CONCLUSIONS & FUTURE WORK

Chapter 11: Conclusion and future work

In conclusion, the software can be used as an inventory system to provide a framework that enables the managers to make reasonable transactions made within a limited time frame. Each transaction made on the system go hand in hand with the data being updated in the database which is the back end

11.1 Future Scope

1. The testing approaches can be automated for easy, efficient and accurate software testing.
2. The user interface can be made more user friendly after receiving the feedbacks and suggestions from the users who are currently using the application.
3. Security can be made more advanced by adding into more techniques.
4. The payment module can be converted into a real working module by connecting the admin account to a bank and completing the necessary procedures.
5. Backup can be automatically maintained.
6. Increasing the size of the database, by adding many more house owners and properties, thus widening the reach of the users.

7.1 PLAGIARISM REPORT



Plagiarism Checker X Originality Report

Similarity Found: 9%

Date: Monday, April 22, 2024

Statistics: 1899 words Plagiarized / 20464 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

A PRELIMINARY PROJECT REPORT ON "HOME AWAY" Your Ultimate Housing Solution
SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE BACHELOR OF
ENGINEERING (COMPUTER ENGINEERING) Submitted by Roll No Name Seat No. BEA06
Anupam Pandey BEA16 Yash Pramod Bambal BEA19 Sakshi Baburao Bhor BEA40 Karn
Datta Godse "Towards Ubiquitous Computing Technology" DEPARTMENT OF
COMPUTER ENGINEERING "Techno – Social Excellence" Marathwada Mitra Mandal's
Institute of Technology (MMIT) Lohegaon, Pune- 411 047 SAVITRIBAI PHULE PUNE
UNIVERSITY (2023-24)

PlagiarismCheckerX Summary Report



7.2 REFERENCES

- [1] K.G.Derpanis, Mean shift clustering, Lecture Notes, [http://www.cse.yorku.ca/%7Ekosta/CompVis/Notes/mean shift.pdf](http://www.cse.yorku.ca/%7Ekosta/CompVis/Notes/mean%20shift.pdf), 2005
- [2] D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [3] J. Pennington, R. Socher, and C. Manning, Glove: Global vectors for word representation, in *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.
- [4] J. Gallin, The longrun relationship between house prices and rents, *Real Estate Economics*, vol. 36, no. 4, pp. 635–658, 2008.
- [5] P. Choudhary, A. Jain, and R. Baijal, Unravelling Airbnb predicting price for new listing, *arXiv preprint arXiv:1805.12101*, 2018.
- [6] G. K. Tso and K. K. Yau, Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks, *Energy*, vol. 32, no. 9, pp. 1761–1768, 2007.
- [7] J. Gallin, The longrun relationship between house price and rents, *Real Economics*, vol. 36, no. 4, pp. 635–658, 2008
- [8] E. Tang and K. Sangani, Neighborhood and price prediction for San Francisco Airbnb listings, [http:// cs229.stanford.edu/proj2015/236 report.pdf](http://cs229.stanford.edu/proj2015/236%20report.pdf), 2018
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. of the Advances in Neural Information rocessing Systems*, Long Beach, CA, USA, 2017, pp. 5998–6008.
- [10] K. Fukunaga and L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.
- [11] S. Arora, Y. Liang, and T. Ma, A simple but tough-to-beat baseline for sentence embeddings, presented at *International Conference on Learning Representations*, Palais des Congres Neptune, Toulon, France, 2017.
- [12] K. Barron, E. Kung, and D. Proserpio, The sharin economy and housing affordability: Evidence from Airbnb, <https://ssrn.com/abstract=3006832>, 2018.

- [13] G. Quattrone, D. Proserpio, D. Quercia, L. Capra, and M. Musolesi, Who benefits from the sharing economy of Airbnb? in Proc. of the 25th International Conference on World Wide Web, Montreal, Canada, 2016, pp. 1385–1394.
- [14] C. Gibbs, D. Guttentag, U. Gretzel, L. Yao, and J Morton, Use of dynamic pricing strategies by Airbnb hosts, *International Journal of Contemporary Hospitality Management*, vol. 30, no. 1, pp. 2–20, 2018.
- [15] G. Zervas, D. Proserpio, and J. W. Byers, The rise of the sharing economy: Estimating the impact of Airbnb on the hotel industry, *Journal of Marketing Research*, vol. 54, no.5, pp. 687–705, 2017.
- [16] D. Lee, W. Hyun, J. Ryu,, W. J. Lee, W. Rhee, and B. Suh, An analysis of social features associated with room sales of Airbnb, in Proc. of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing, New York, NY, USA, 2015, pp. 219–222.
- [17] L. Cao, J. Luo, A. Gallagher, X. Jin, J. Han, and T. S. Huang A worldwide tourism recommendation system based on geotagged web photos, in Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 2010, pp. 2274–2277.
- [18] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997
- [19] Y. Li, Q. Pan, T. Yang, and L. Guo, Reasonable price recommendation on Airbnb using Multi-Scale clustering, in Proc. of the 35th Chinese Control Conference (CCC), Chengdu, China, 2016, pp. 7038–7041.
- [20] Y. Li, S. Wang, T. Yang, Q. Pan, and J. Tang, Price recommendation on vacation rental websites, in Proc. Of the 2017 SIAM International Conference on Data Mining Galleria Houston, TX, USA, 2017, pp. 399–407.