Online Rental Service

BY

Pranta Roy

ID: 151-15-4979

Md. Mobassir Al Masud Basunia

ID: 151-15-4899

Hreday Sagar Chakraborty

ID: 151-15-5017

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

Supervised By
Moushumi Zaman Bonny
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised By
Mr. Anup Majumder
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

NOVEMBER 2018

APPROVAL

This project titled "Rental Service for Rent Home, Sell Home" submitted **by Pranta Roy, ID No: 151-15-4979, Md. Mobassir Al Masud Basunia, ID No: 151-15-4899, and Hreday Sagar Chakraborty, ID No:151-15-5017** to the Department of Computer Science and Engineering, Daffodil International University has been accepted of 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 06 November 2018.

BOARD OF EXAMINERS

____

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Narayan Ranjan Chakraborty **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Md. Tarek Habib **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

____

Dr. Mohammad Shorif Uddin **External Examiner**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Moushumi Zaman Bonny, Senior Lecturer, Department of CSE** Daffodil International University. We also declare that neither this Project nor any part of this project has been submitted elsewhere for award of any degree or Diploma.

Supervised by:

Moushumi Zaman Bonny
Senior Lecturer
Department of CSE
Daffodil International University

Co- Supervised by:

Mr. Anup Majumder
Lecturer
Department of CSE
Daffodil International University

Submitted By:

Pranta Roy
ID: 151-15-4979
Department of CSE
Daffodil International University

Mobassir Al Masud Basunia
ID: 151-15-4899
Department of CSE
Daffodil International University

Hreday Sagar Chakraborty
ID: 151-15-5017
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

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

We really grateful and wish our profound our indebtedness to **Moushumi Zaman Bonny, Senior Lecturer**, Department of Computer Science and Engineering, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of web development influenced us to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain, Professor and Head, Department of Computer Science and Engineering**, for his kind help to finish our project and to other faculty member and the staff of Computer Science and Engineering department of Daffodil International University.

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

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

ABSTRACT

The main purpose our project is to get information about how rental houses are currently being managed. The name of our project is “Online Rental System”. It is a web based and android app. This system is very essential for whole country especially for Dhaka City Students, Family and Bachelors. Nowadays the people of Dhaka city faces problems while they try to find home for rent and sell. We have made this system to prevent this kind of problems. By using this application a user or person can easily get a home with a reasonable price and search the best result easily. This software application gives the functionality for buyers, allowing them to search for houses by features or address. Similarly, it provides functionality for the seller, authorize them to log into the system and add new advertisements or delete existing ones. In our system mainly we focused for prevent current analog system make our system fully digital. User can get their expected result shortly and very easily. We have already mentioned it that the main goal of our application is to save people time, cost and sufferings. This era is a full of modern technology and smart phone is available to almost everyone. We try to solve this problem by technology. So hopeful our project will be helpful for the city people.

TABLE OF CONTENTS

CONTENTS **PAGE**

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Introduction

Rental house service has become important service in modern society hence the need to have rental house system. However, providing houses affordable rent price has been a long-standing issue in our city. For this reason we create a system to help people. A house is a multidimensional good, which consists of a bundle of attributes that differs in quantity and quality and influence house rent price. This includes number of rooms, toilet, chicken, housing type and place of work. People search location in our site. It's very helpful in our service.

1.2 Motivation

Nowadays students, bachelors and family in Dhaka city suffers a lot when they want home. It is not possible to visit the whole city to get a good home. We have made this app to decrease the sufferings of the people. Actually we got the motivation from our personal experience, when we first came in Dhaka city we suffered a lot in this problem. We did not able to manage a single sit for us to stay. Then we thought we would made something that would help to prevent that kind of situation and that was the reason behind our motivation.

1.3 Objectives

Objective of this project are given below:

- We have to spend a lot of time to rent home & apartment
- Discrimination between students, bachelors and family for renting home
- Rent Cost varies from area to area
- In existing system, there is a lot of lacking of Data and information.
- Existing data's are not Store in online

1.4 Expected Outcome

The expected outcome of this project is to make a website and android apps that will enable to see the details about the house in their phone screen within the short time. People can choose houses as per their wants. The main goal is to save the time .People need not go to see the houses, they can see the houses with the help of this app. f this app is installed to someone's smart phone then they can easily get a good home on the basis of their wants. It will not only save people time but also cost too.

1.5 Report Layout

In the first chapter of this report, we have discussed what actually motivates us for this project and from where we get inspiration to work on this as well as the objectives of this project and the outcome we expect to get from this project. In the second chapter of this report, we have focused on the background of food and nutrition, discussed some similar works, compare the activities and challenges we faced through the project work. In the third chapter we have discussed the business process model, use case model, ER diagram, design requirement and requirement collection. In the fourth chapter we have focused on the front end, back end design specification and requirements for implementing the project. In the fifth chapter we have showed the project implementation phase, like database, front end, testing and test results implementations. In the sixth chapter we have conclude our discussion with further development ideas that we have in our minds.

CHAPTER 2

Background

2.1 Introduction

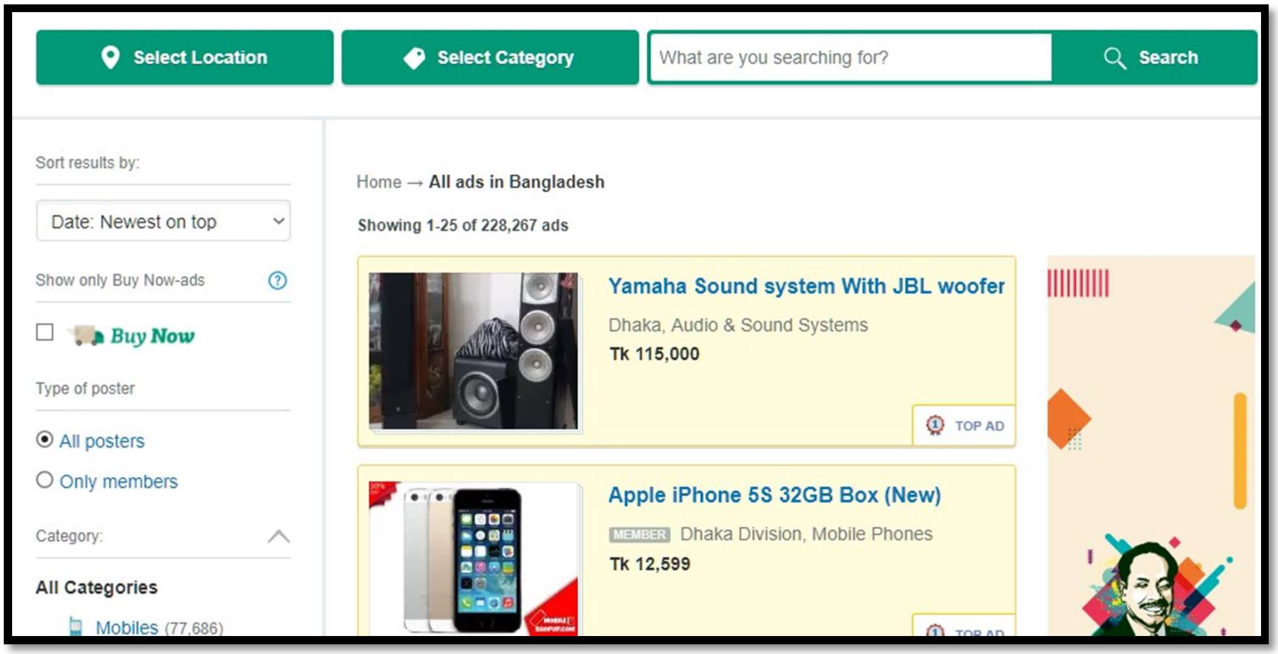
Rent home is a viral part in our bachelor and family life. Most of the people in our city is busy in different occupation. So they have not enough time to see perfect home. Our system help to find perfect home in front of Dhaka city. People use it desktop, laptop, mobile phone. In this paragraph below we will be talking about the similar kind of system that have been used for find home.

2.2 Related Works

Some Web Based Online system related to our project are given below with their main features

2.2.1 Bikroy.com

Bikroy.com is an open source web based online system. This system provides user account, profiles, user can create post, edit & modify properties, visitor can view add by visiting this site [1].



Sort results by: Date: Newest on top

□ Buy Now

Type of poster: ● All posters ○ Only members

Category:

Home → All ads in Bangladesh

Showing 1-25 of 228,267 ads

Yamaha Sound system With JBL woofer
Dhaka, Audio & Sound Systems
Tk 115,000

Apple iPhone 5S 32GB Box (New)
MEMBER Dhaka Division, Mobile Phones
Tk 12,599

Figure 2.1: Bikroy.com web site

2.2.2 Lamudi.com

Lamudi.com.bd helps people in the property market to find their ideal home, land or commercial property. Our goal is to empower consumers with the most thorough information in the market and get them in contact with the developers, agents and landlords. We aim to have our listings give you as much information about the property as possible [2].

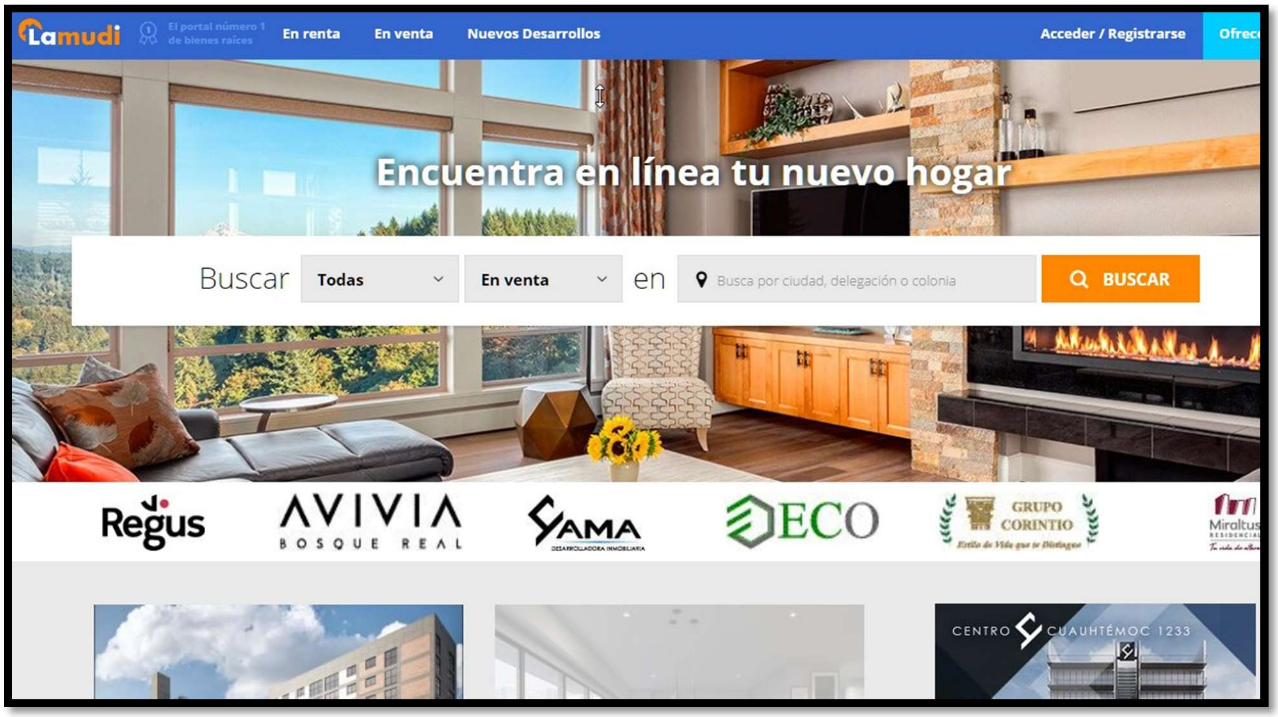


Figure 2.2: Lamudi.com web site

2.3 Comparative Studies

There are some work that have researched previously is in the context of on line rent application that we have mentioned in the above section. Those works some way related to our project. But they all are not same, some are different in technology, some are different in platform and some are limited in work and features. But those application helps us grasp the idea that how we should build our project. We have figured out which features we should use, so that we can make a different and an alternative application with respect to those previous applications.

2.3.1 Bikroy.com Vs Online Rent Application Similarity

- User can create profile
- Submit Properties
- Get contact information

Dissimilarity:

- Admin approval for accept user submitted properties
- Admin can add post as featured home page and user cannot access it
- If any user cannot share their personal phone number in public place for security then our system provide mailing service. This services helps user contact another user by sending mail fill up some required fill. ‘
- Our system provide android application for view and parse data from our central server.

2.3.2 Lamudi.com Vs Online Rent Application Similarity

Lamudi provides sale and rent services only flat and apartment. This system does not provide single room or single bed. Lamudi is not free to user they are provide paid services

Dissimilarity:

- Our system provide free service for all user, anyone can create profile and submit any properties without any conditions and fees.
- We provide android application for user efficiency and our system is fully user friendly any one can operate our system without any pre knowledge about our system.

2.4 Scope of the Problem

There are some problems that we have faced in the way of developing this application. The First thing is the resource of similar projects. Because every resource we come across to make these project are made in web and android based. During working on this project, we have faced some challenges and here we have listed out some of them:

- Integrating Json parsing from laravel in android.
- Integrating server and client at the same time in an app.
- Mailing system.
- Laravel routing.
- Establish database connection and relations.
- Make user pending post and accept or reject method that can access only admin.
- Make featured home page post after approve post.

2.5 Challenges

During working on this project, we have faced some challenges and here we have listed out some of them:

- Laravel plugin, Routing
- JSON parsing in android Apps
- Integrating programming in android.
- Parse JSON file using Laravel
- Android work in a single thread. But we have run different method that had to run alongside with the main thread. That's why we needed background process to run. That was a challenging part.
- Connection in database
- Integrating server and apps
- Making feature in homepage
- Finding resource

CHAPTER 3

Requirement Specification

3.1 Business Process Modeling

Business process modeling is the graphical representation of a company's business processes or workflows, as a means of identifying potential improvements. This is usually done through different graphing methods, such as the flowchart, data-flow diagram, etc [3].

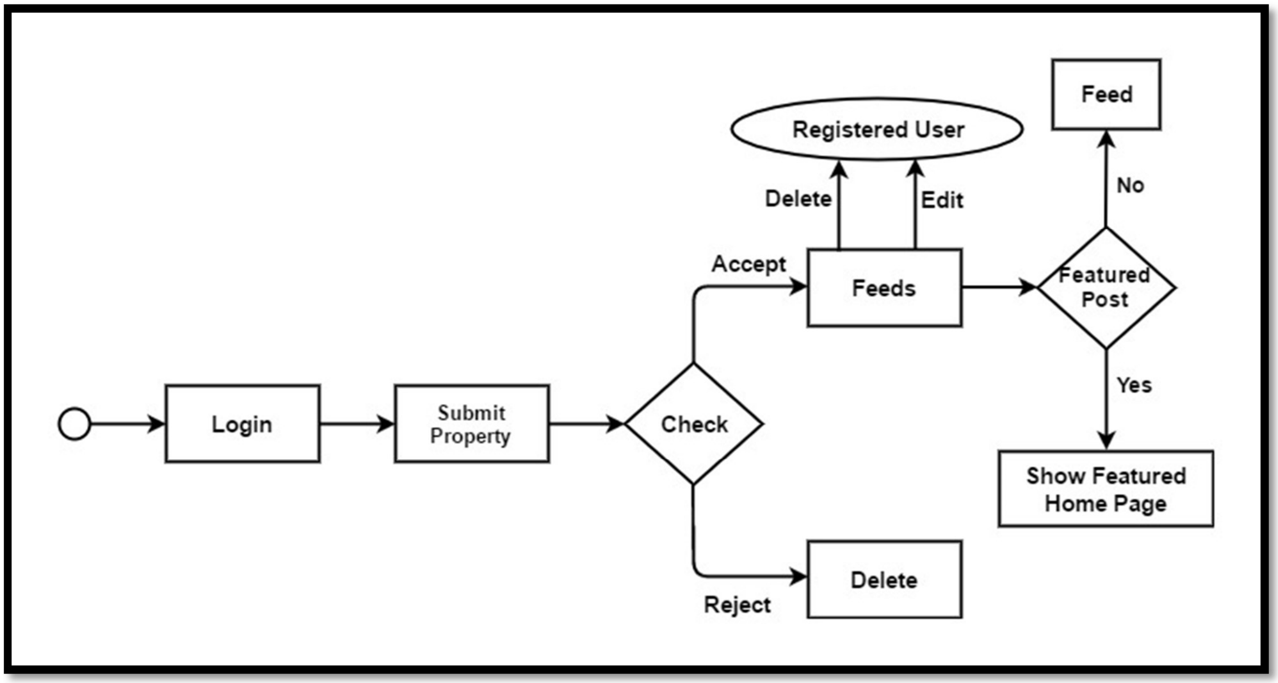


Figure 3.1: Business process model of our project

3.2 Requirement Collection and Analysis

Analysis is a detailed study of the various operation performed by a system and their relationships within and outside the system. This involves gathering information and using structured tools for analysis. Gathering information is also known as requirement collection.

To make analysis easier, consider grouping the requirements into four categories:

Functional Requirements:

- Administration can remove users, delete and modify user posts.
- Administration Can accept or reject users submit property.
- Administration Can view post as featured post by manually accepted post
- Registered User can create profile, Edit Profile.
- Registered User can submit Post and Modify and Delete Post.
- User can show post and all information's.
- User can send mail by fill up some specific fields without any preconditions.

Non Functional Requirements:

- Performance
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security

3.3 Use Case Modeling and Description

Use case modelling is a useful tool for requirements elicitation. It provides a graphical representation of the software system's requirements [4].

3.3.1 Use Case Model

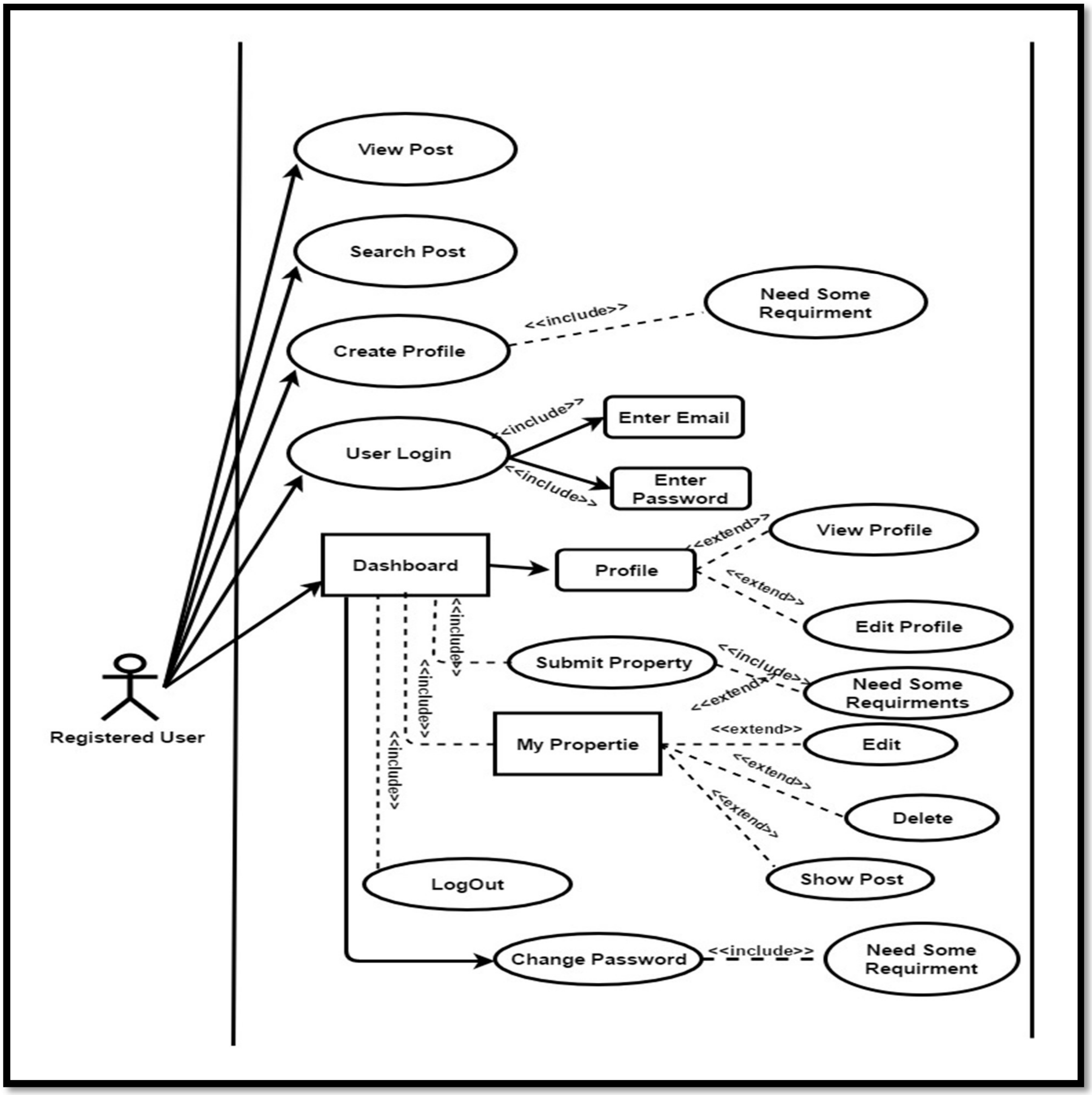


Figure 3.2: Use case model of our project

3.3.2 Use Case Model

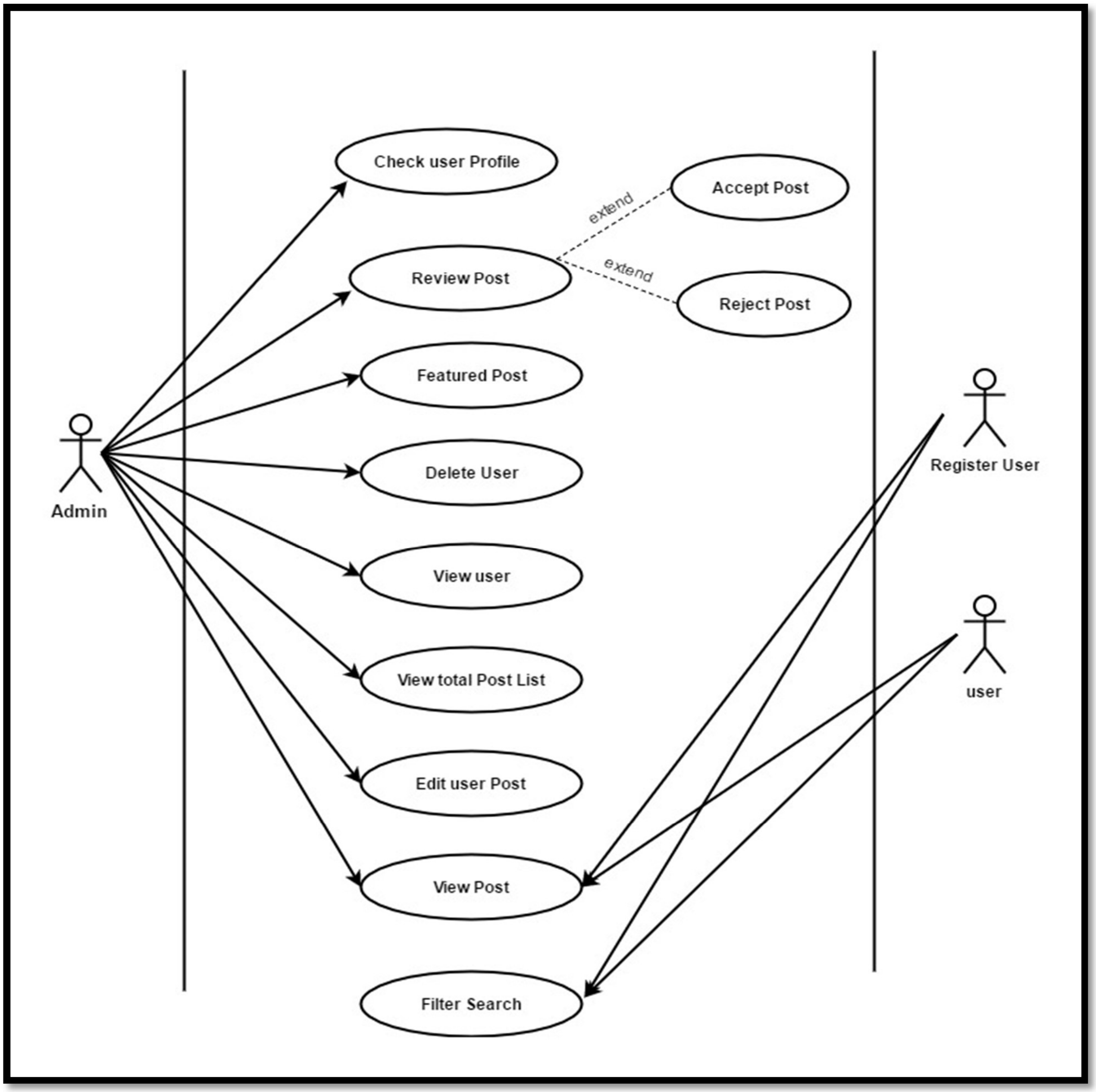


Figure 3.3: Use case model of our project

3.3.2 Use Case Description

Actor – who is using the system? Primary Actor – stakeholder who initiates an interaction with the system to achieve a goal Preconditions – what must be true or happen before and after the use case runs. Main success scenarios – use case in which nothing goes wrong [5].

Use Case 01: View Post

Primary Actor: User

Precondition: Open application

Secondary Actor: Null

Main Success Scenario: User can View post

Exception Scenario: User will continue without account

Use Case 02: Search Post

Primary Actor: User

Precondition: Open Application

Secondary Actor: Null

Main Success Scenario: user will see the most appropriate search result

Exception Scenario: Invalid Search Query

Use Case 03: Create Profile

Primary Actor: User

Precondition: Open Sign Up page

Secondary Actor: Null

Main Success Scenario: User will input his/her information

Exception Scenario: Invalid email or password

Use Case 04: User Login

Primary Actor: User

Precondition: Must have registered

Secondary Actor: Null

Main Success Scenario: Login successful

Exception Scenario: Wrong email or password

Use Case 05: Edit Profile

Primary Actor: User

Precondition: Successfully Login

Secondary Actor: Null

Main Success Scenario: Update your profile successfully

Exception Scenario: Invalid information

Use Case 06: Change Password

Primary Actor: User

Precondition: Successfully Login

Secondary Actor: Null

Main Success Scenario: password has been reset successfully

Exception Scenario: Password Mismatch

Use Case 07: Submit Property

Primary Actor: User

Precondition: Successfully Login

Secondary Actor: Null

Main Success Scenario: Your property has been submitted for review

Exception Scenario: invalid information

Use Case 08: Edit or Delete Post

Primary Actor: User

Precondition: Successfully Login

Secondary Actor: Null

Main Success Scenario: Your property has been successfully edit or delete

Exception Scenario: Invalid information

Use Case 09: Admin Login

Primary Actor: Admin

Precondition: Must register user

Secondary Actor: User

Main Success Scenario: Successfully login and admin dashboard

Exception Scenario: Invalid access

Use Case 10: Review Post

Primary Actor: Admin

Precondition: Login as admin

Secondary Actor: User

Main Success Scenario: Admin will accept or reject user pending post

Exception Scenario: Null

Use Case 11: Featured post

Primary Actor: Admin

Secondary Actor: User

Precondition: Users accepted post

Main Success Scenario: Post successfully added as featured post

Exception Scenario: Null

Use Case 12: Delete User

Primary Actor: Admin

Precondition: Must have an active user

Secondary Actor: user

Main Success Scenario: Successfully remove

Exception Scenario: There is no one to delete.

Use Case 13: Edit or delete post

Primary Actor: Admin

Precondition: Must have user submitted post

Secondary Actor: User

Main Success Scenario: post successfully removed or edited

Exception Scenario: Null

3.4 Logical Data Model of Online Rental Service

A logical data model or logical schema is a data model of a specific problem domain expressed independently of a particular database management product or storage but in terms of data structures such as relational tables and columns, object-oriented classes [6].

3.4.1 Logical Data Model: Level 0

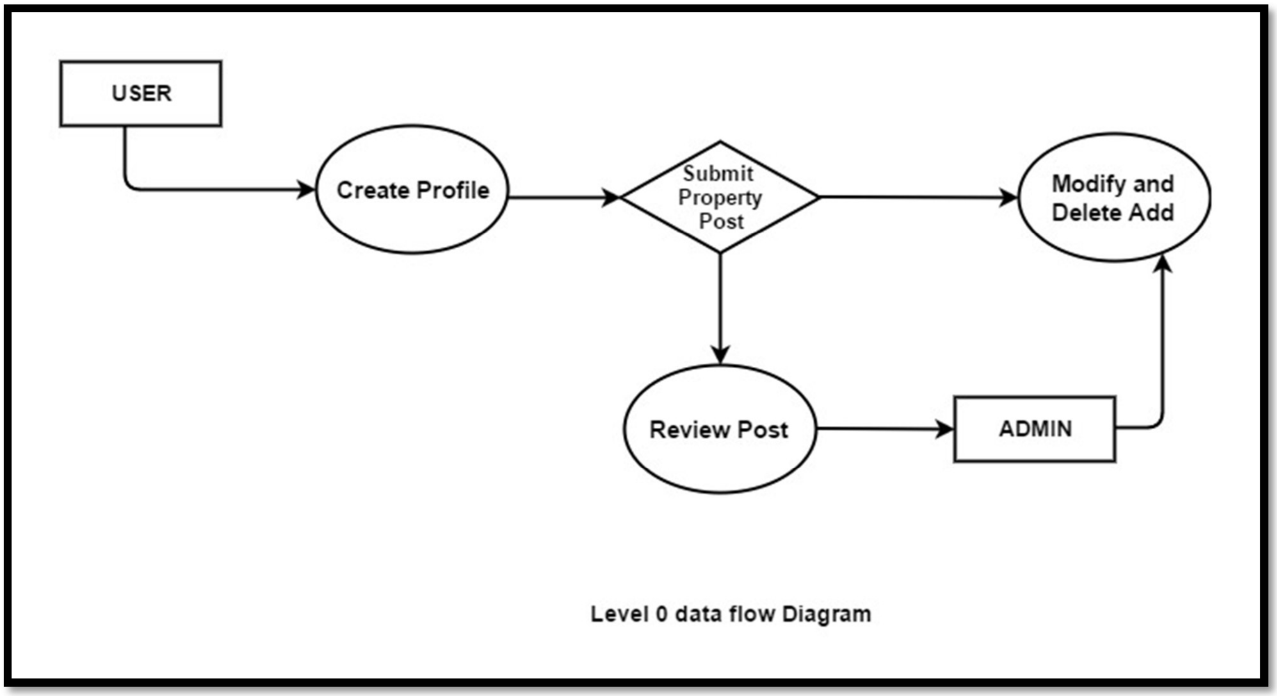


Figure 3.4: Logical data model of our project

3.4.2 Logical Data Model: Level 1

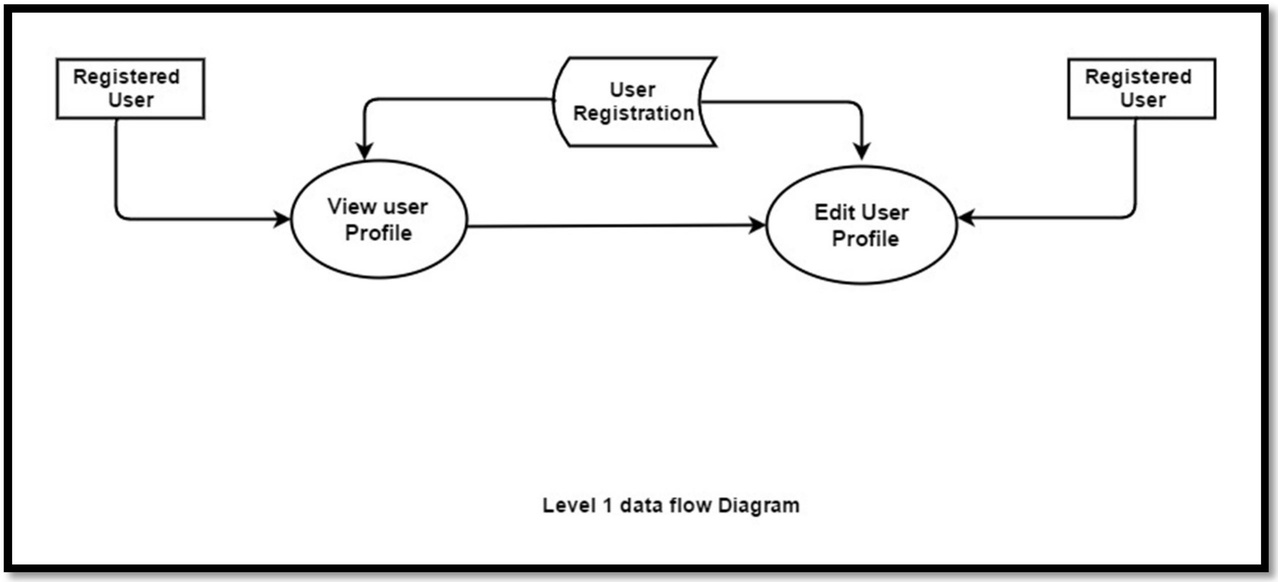


Figure 3.5: Logical data model of our project

3.4.3 Logical Data Model: Level 2

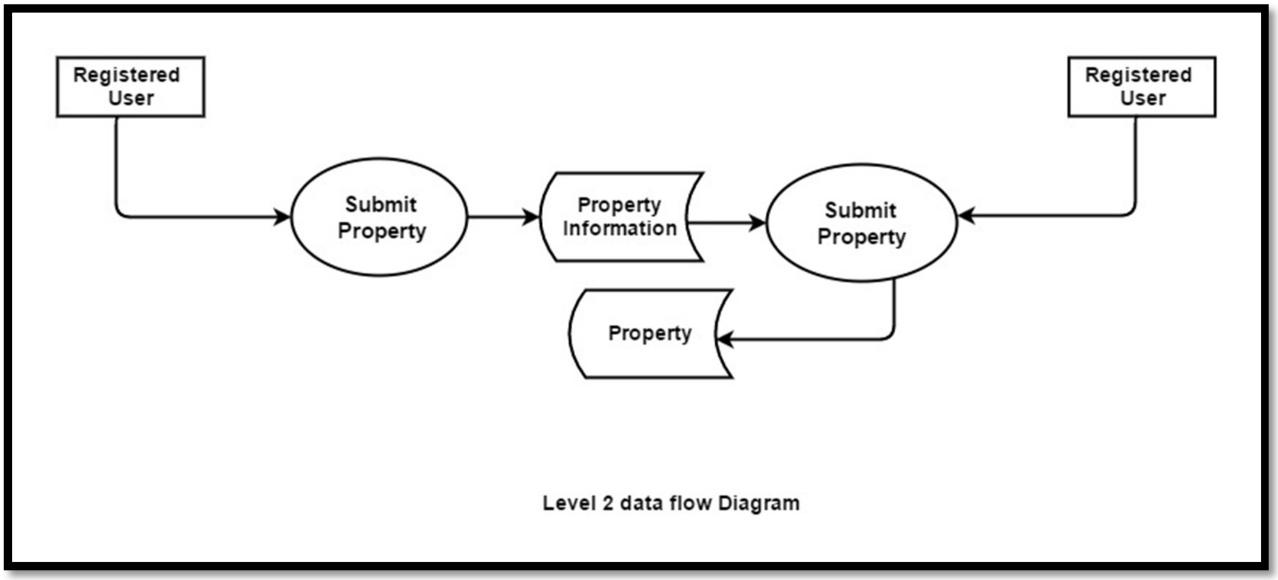


Figure 3.6: Logical data model of our project

3.4.4 Logical Data Model: Full Part

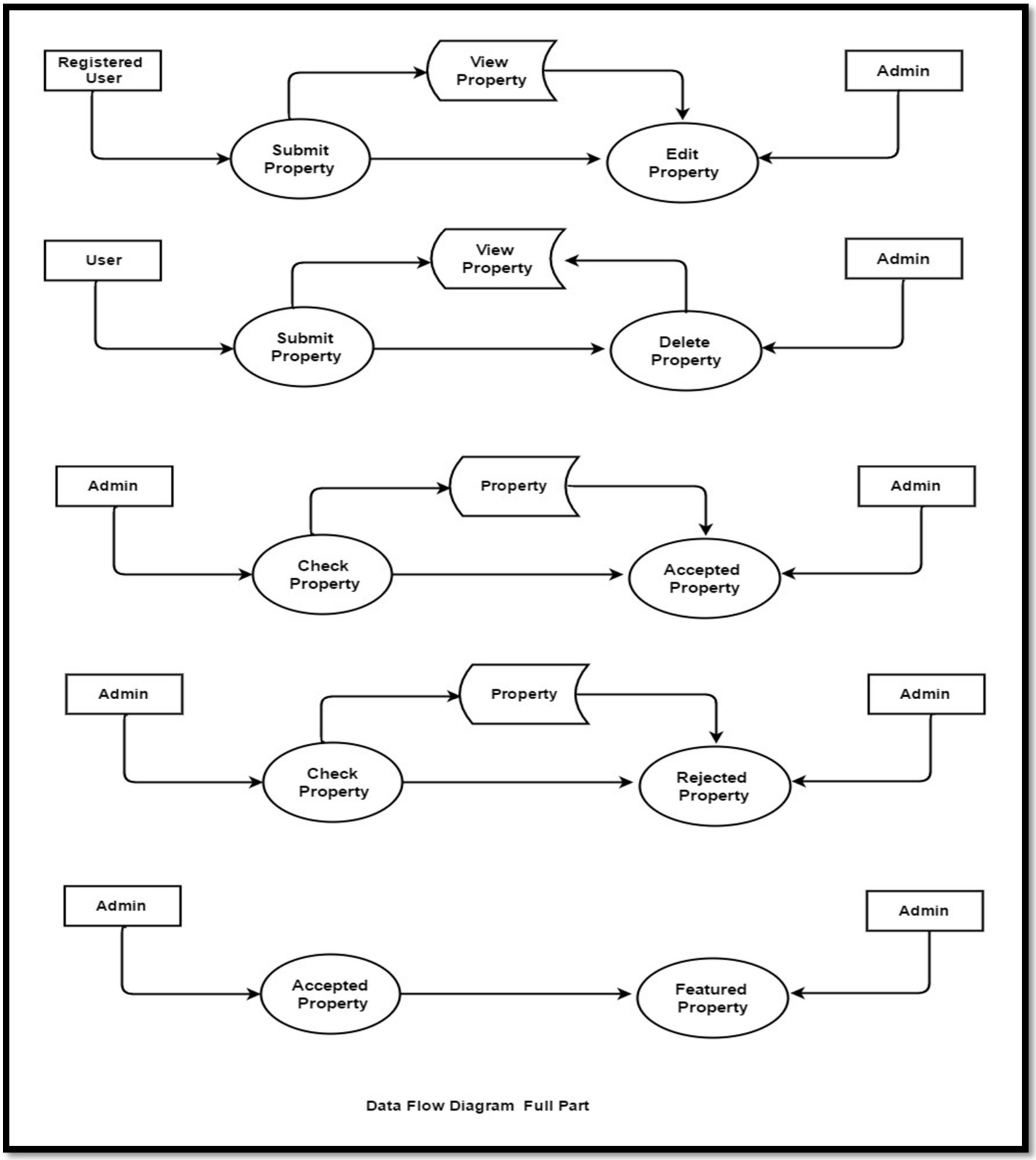


Figure 3.7: Logical data model of our project

3.5 Design Requirements

Design requirements state the important characteristics that your design must meet in order to be successful. Some design requirements we planned for our project are given below:

- system should be very user friendly
- There is one actor, its user and admin
- user can view post
- user can send mail
- user can search and filter any post
- registered user can create and update profile
- registered user can edit and delete post
- admin can approve and delete registered user post
- admin can delete any registered user and post
- admin can make feature post for show home page
- admin can search registered user profile and information

CHAPTER 4

Design Specification

4.1 Front-End Design

Front-end design is very important part of an application. The purpose of front-end design is make the experience of user better. By keeping this our purpose we have made our System is user friendly and easy to interact. Front end design of this project is done by HTML, CSS and BOOTSTRAP. For android Part Using JAVA, XML. The front-end design includes navbar, sidebar, forms, button, list group, dropdowns, carousel etc. At first dashboard page its control all off features in this project. The First page is home page it has search option to search property features and view submit property in this page. Those are the logo, home, feed, about us, sign in/sign up button and property views. When we accept user pending properties and make featured it then all details come to this page otherwise not. The second page is feed when we submit the property all details view to this page. This page control by dashboard. The third page is about us. To details our project member. Fourth page is admin page it's only for registered user details. User can update profile, change password, edit and modify post in this page submit property to view feed page [7].

4.2 Back-End Design

Back-end of any software, app or any website is where logical function happened. Back-end design for web application is completed by using PHP, JavaScript and Laravel. We make database connection to use SQL server xampp. We also use MySQL database store our email, message, property details and personal information user. For android application we used java android programming and views post from our central server by json parsing [7].

4.2.1 Laravel

Laravel is a PHP framework. In the back-end we use this framework for make our system interact with fond-end. We make all logic by Laravel. When we login, show featured, contact client and details to registered user for handle this operation done by PHP framework. We handle our mysql operation [8].

4.3 Interaction Design and UX

Interaction Design and UX is the part where user perform task. It is the visual Part where user gets to know the system and command the backend system to do work. In the below section we will discuss about our project visual side how it interact with user [9].

4.3.1 UX Design

User experience design focuses on the overall experience between a user and a product. It is not just concerned with the interactive elements but also the way that certain elements look, feels or contrive to deliver certain outputs. As we discuss UX stand for user experience. In our system we have provide buttons, logo, lists and actions of those list so that user can interact and feel easy and comfortable with the system. We have provided logo for our each tab so that use can identify them easily and tell which page is used for what. We have provided accept and reject button in admin page. When admin see user post he will accept or reject [9].

4.4 Implementation Requirements

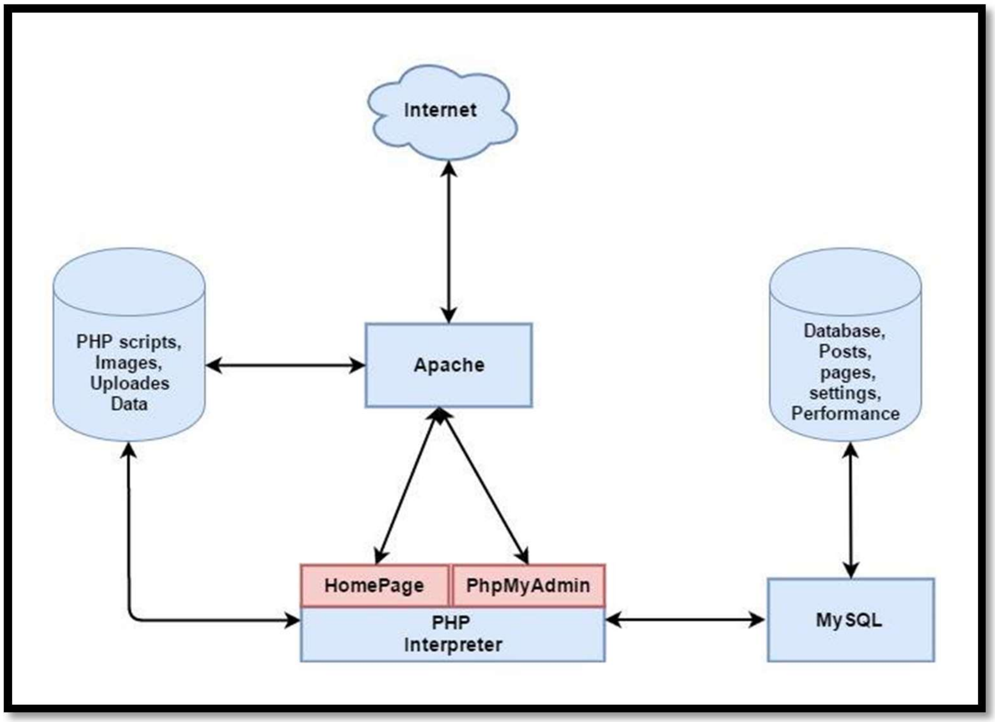
For implementing this project we need to use a mobile device to implement. We need Xampp server, Laravel install in working platform, and android studio ide used for java programming, json and xml code for app interface and notepad ide as a tool for writing HTML, CSS, BOOTSTRAP, Laravel php code, JAVA, XML. We need to first open xampp server to test our project. Then open PowerShell and run Laravel server. At last run the system in your browser. For live our application we need a server that support PHP 7.0.0 and MySQL database.

CHAPTER 5

Implementation and Testing

5.1 Implementation of Database

A database management system handles the request generated from the sql interface, producing or modifying data in response to these requests. This involves s multilevel processing system. Database server consisting with a computer program and it provides database services to computer; mainly depend on the client server model. To create the database for our Laravel project, we simply open the phpMyadmin panel in our browser of choice and we proceed to create the database by giving it a name and editing the security credentials. Before getting started, we need to be sure to configure the database connection in app/config/database.php file by editing the lines of PHP code containing the credentials to match our database's credentials. Laravel's default database, which ais MySQL will be kept as our database management system for our current project [10].



Internet
PHP scripts, Images, Uploades Data
Apache
Database, Posts, pages, settings, Performance
HomePage
PhpMyAdmin
PHP Interpreter
MySQL

Figure 5.1: Architecture used in the development of our system

5.1.1 All Database In this project:

In our project we are use all those database table

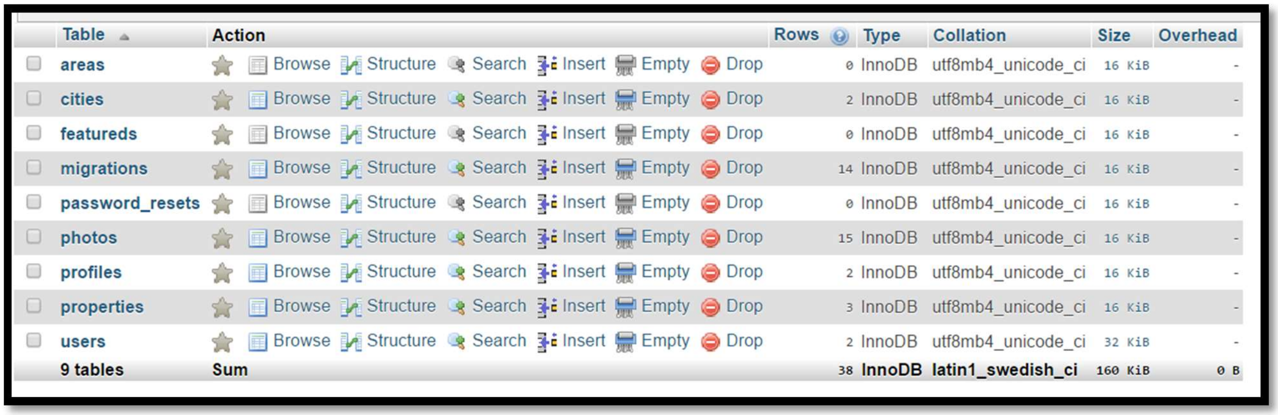
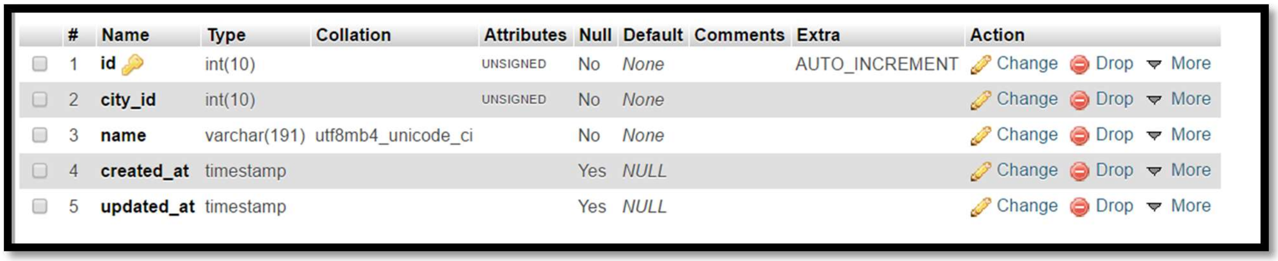


	Table	Action	Rows	Type	Collation	Size	Overhead
□	areas	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	cities	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	featureds	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	migrations	Browse Structure Search Insert Empty Drop	14	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	password_resets	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	photos	Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	profiles	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	properties	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	16 KiB	-
□	users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32 KiB	-
	9 tables	Sum	38	InnoDB	latin1_swedish_ci	160 KiB	0 B

Figure 5.2: All database in this project

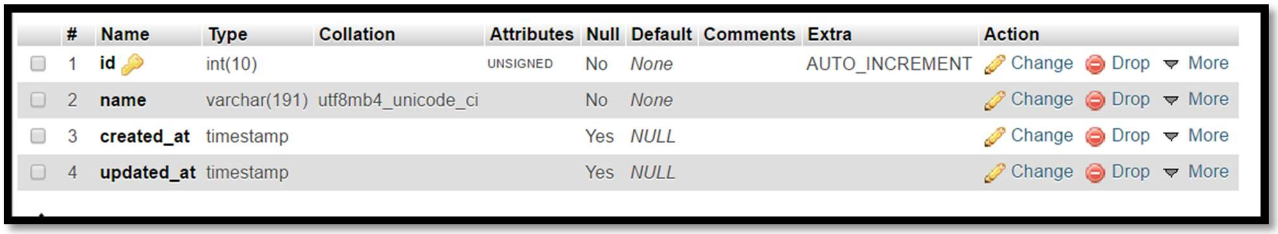
5.1.2 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
□	2	city_id	int(10)		UNSIGNED	No	*None*			Change Drop More
□	3	name	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	4	created_at	timestamp			Yes	*NULL*			Change Drop More
□	5	updated_at	timestamp			Yes	*NULL*			Change Drop More

Figure 5.3: Areas database implementation

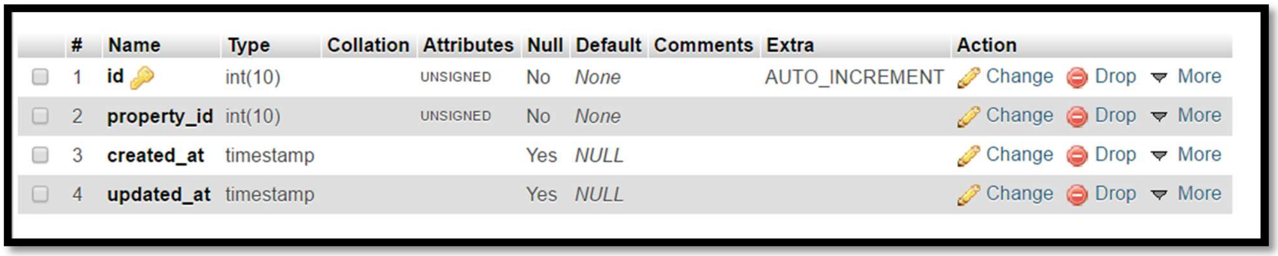
5.1.3 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id** 🔑	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**name**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	3	**created_at**	timestamp			Yes	*NULL*			Change Drop More
☐	4	**updated_at**	timestamp			Yes	*NULL*			Change Drop More

Figure 5.4: City’s database implementation

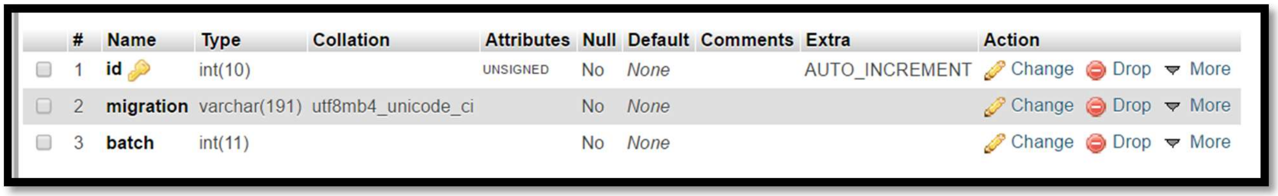
5.1.4 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id** 🔑	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**property_id**	int(10)		UNSIGNED	No	*None*			Change Drop More
☐	3	**created_at**	timestamp			Yes	*NULL*			Change Drop More
☐	4	**updated_at**	timestamp			Yes	*NULL*			Change Drop More

Figure 5.5: Featured database implementation

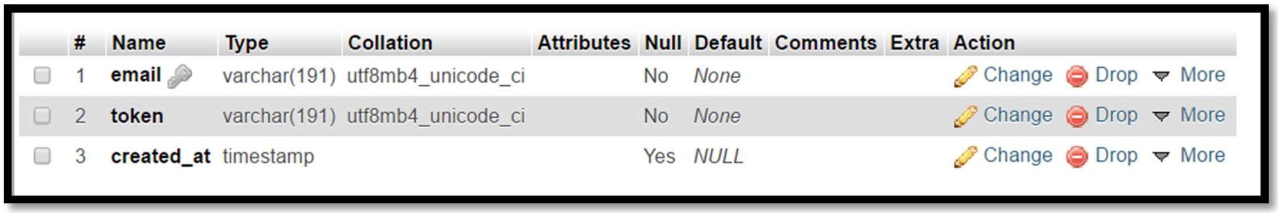
5.1.5 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id** 🔑	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**migration**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
☐	3	**batch**	int(11)			No	*None*			Change Drop More

Figure 5.6: Migration database implementation

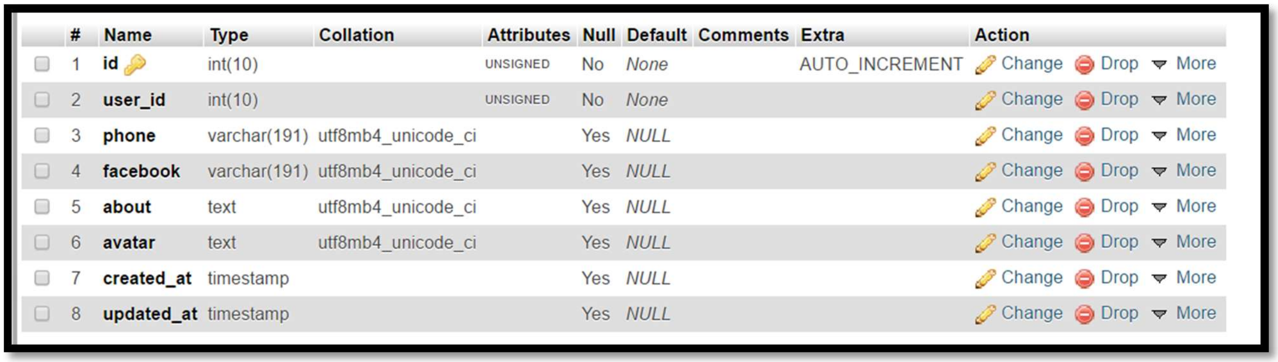
5.1.6 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	email	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop ▾ More
□	2	token	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop ▾ More
□	3	created_at	timestamp			Yes	*NULL*			Change Drop ▾ More

Figure 5.7: Reset password database implementation

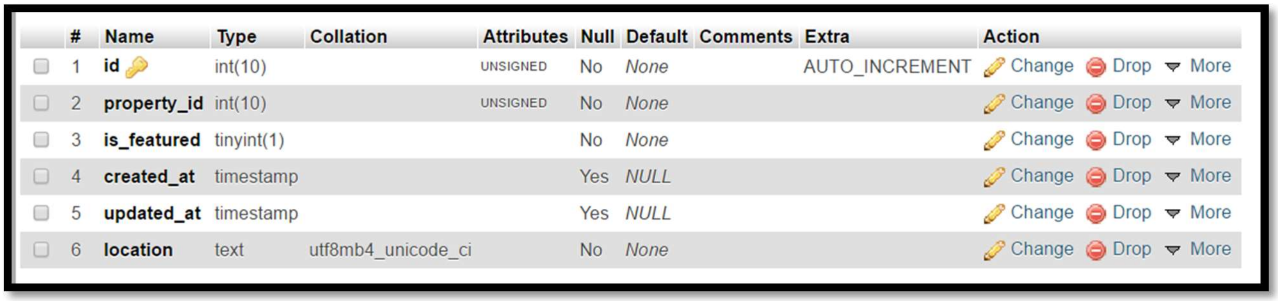
5.1.7 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop ▾ More
□	2	user_id	int(10)		UNSIGNED	No	*None*			Change Drop ▾ More
□	3	phone	varchar(191)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop ▾ More
□	4	facebook	varchar(191)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop ▾ More
□	5	about	text	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop ▾ More
□	6	avatar	text	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop ▾ More
□	7	created_at	timestamp			Yes	*NULL*			Change Drop ▾ More
□	8	updated_at	timestamp			Yes	*NULL*			Change Drop ▾ More

Figure 5.8: Profiles database implementation

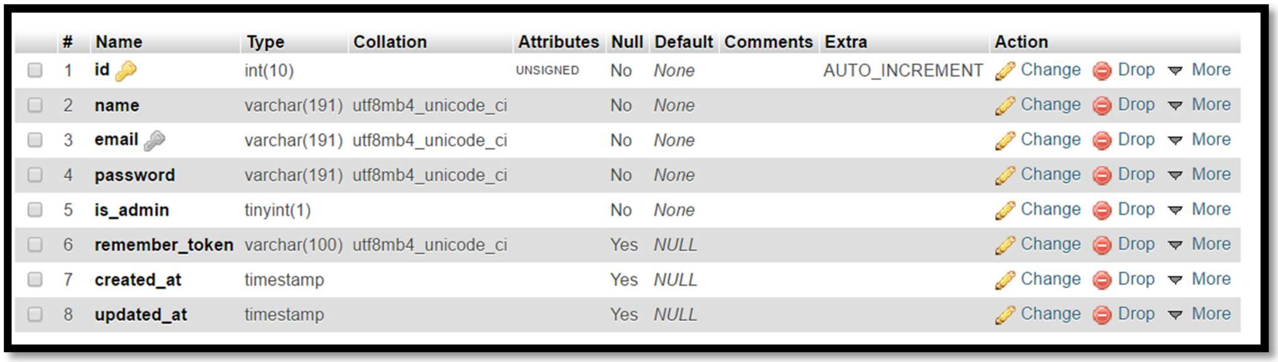
5.1.8 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop ▾ More
□	2	property_id	int(10)		UNSIGNED	No	*None*			Change Drop ▾ More
□	3	is_featured	tinyint(1)			No	*None*			Change Drop ▾ More
□	4	created_at	timestamp			Yes	*NULL*			Change Drop ▾ More
□	5	updated_at	timestamp			Yes	*NULL*			Change Drop ▾ More
□	6	location	text	utf8mb4_unicode_ci		No	*None*			Change Drop ▾ More

Figure 5.9: Photos database implementation

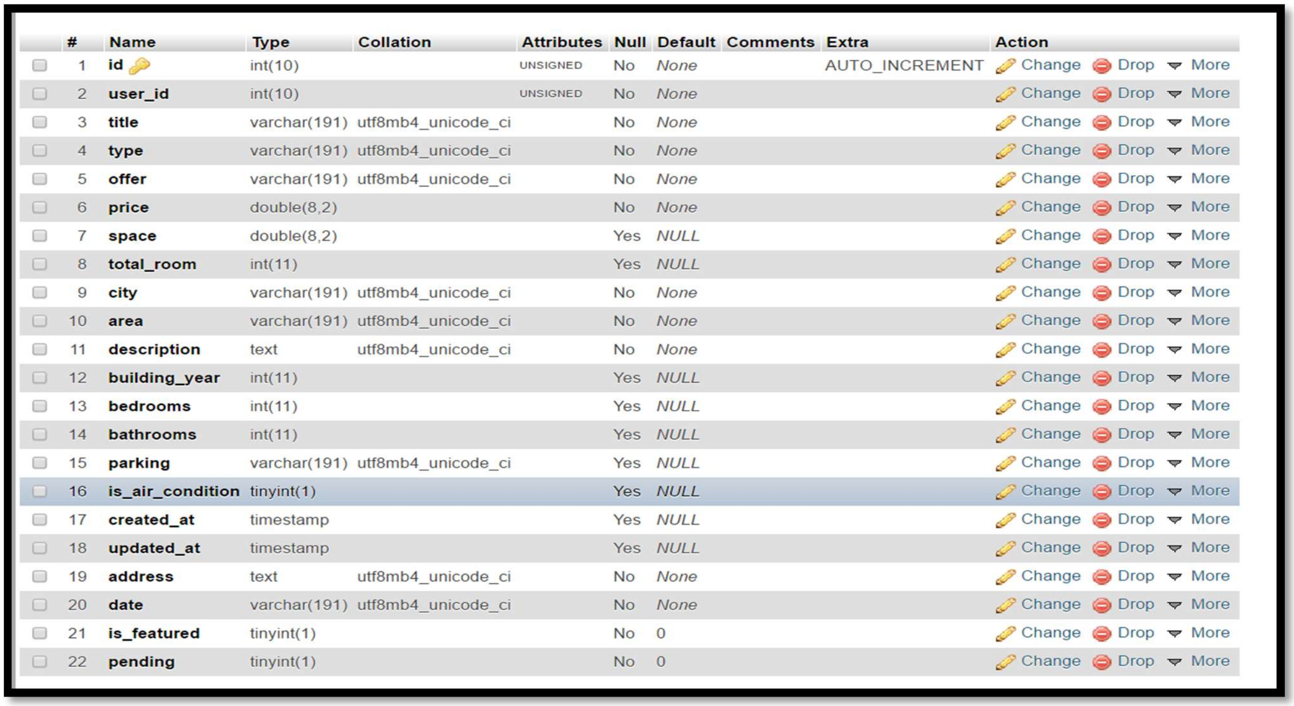
5.1.9 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**name**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	3	**email**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	4	**password**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	5	**is_admin**	tinyint(1)			No	*None*			Change Drop More
□	6	**remember_token**	varchar(100)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
□	7	**created_at**	timestamp			Yes	*NULL*			Change Drop More
□	8	**updated_at**	timestamp			Yes	*NULL*			Change Drop More

Figure 5.10: Users database implementation

5.1.10 Database Implementation:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(10)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**user_id**	int(10)		UNSIGNED	No	*None*			Change Drop More
□	3	**title**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	4	**type**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	5	**offer**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	6	**price**	double(8,2)			No	*None*			Change Drop More
□	7	**space**	double(8,2)			Yes	*NULL*			Change Drop More
□	8	**total_room**	int(11)			Yes	*NULL*			Change Drop More
□	9	**city**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	10	**area**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	11	**description**	text	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	12	**building_year**	int(11)			Yes	*NULL*			Change Drop More
□	13	**bedrooms**	int(11)			Yes	*NULL*			Change Drop More
□	14	**bathrooms**	int(11)			Yes	*NULL*			Change Drop More
□	15	**parking**	varchar(191)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
□	16	**is_air_condition**	tinyint(1)			Yes	*NULL*			Change Drop More
□	17	**created_at**	timestamp			Yes	*NULL*			Change Drop More
□	18	**updated_at**	timestamp			Yes	*NULL*			Change Drop More
□	19	**address**	text	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	20	**date**	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
□	21	**is_featured**	tinyint(1)			No	0			Change Drop More
□	22	**pending**	tinyint(1)			No	0			Change Drop More

Figure 5.11: Properties database implementation

5.1.11 Implementation tools and Software

The following software's and tools where selected and used while creating the system.

- MySQL
- APACHE
- XAMPP
- Android Studio

5.1.12 MySQL

MySQL is the world's most popular open-source relation database management system. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! And many more [10].

5.1.13 APACHE

Apache HTTP Server, colloquially called Apache is free and open-source platform web software, released under the terms of Apache License 2.0. Apache is development and maintained by an open community of developers under the auspices of the Apache Software Foundation [11].

5.1.14 XAMPP

XAMPP is the most popular PHP development environment. It's a free & open source cross-platform web server solution stack package.

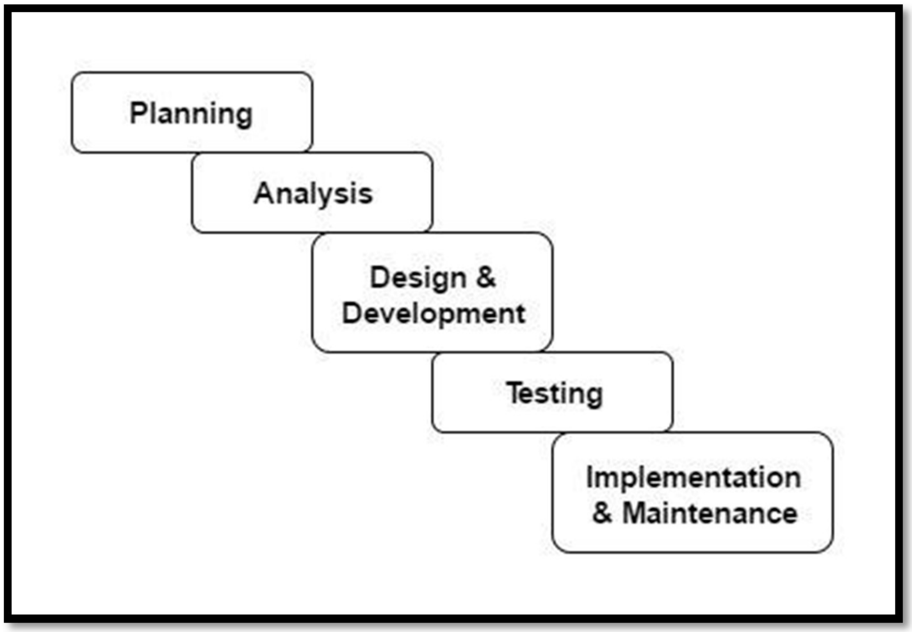
XAMPP includes:

- APACHE HTTP server.
- MySQL date base.
- Interpreter for scripts which is written in-PHP

XAMPP is a completely free, easy to install Apache distribution containing PHP, & Perl. Apache Friends is a non-profit project to promote the Apache web server & is home to the XAMPP project. Web developer & programmer can test their work in PC local host without the access of the internet [12].

5.2 Implementation of Front-end Design

The web development life cycle (WDLC) is very similar to systems development Life Cycle (SDLC) and must be iterative under taken from the client's perspective. Any software development project, a methodology should be followed to ensure project consistency and completeness. The Web development life cycle includes the following phases: planning, analysis, design and development, testing, and implementation and maintenance [13].



Figurer 5.12: Web development life cycle

5.2.1 Implementation of Admin Dashboard:

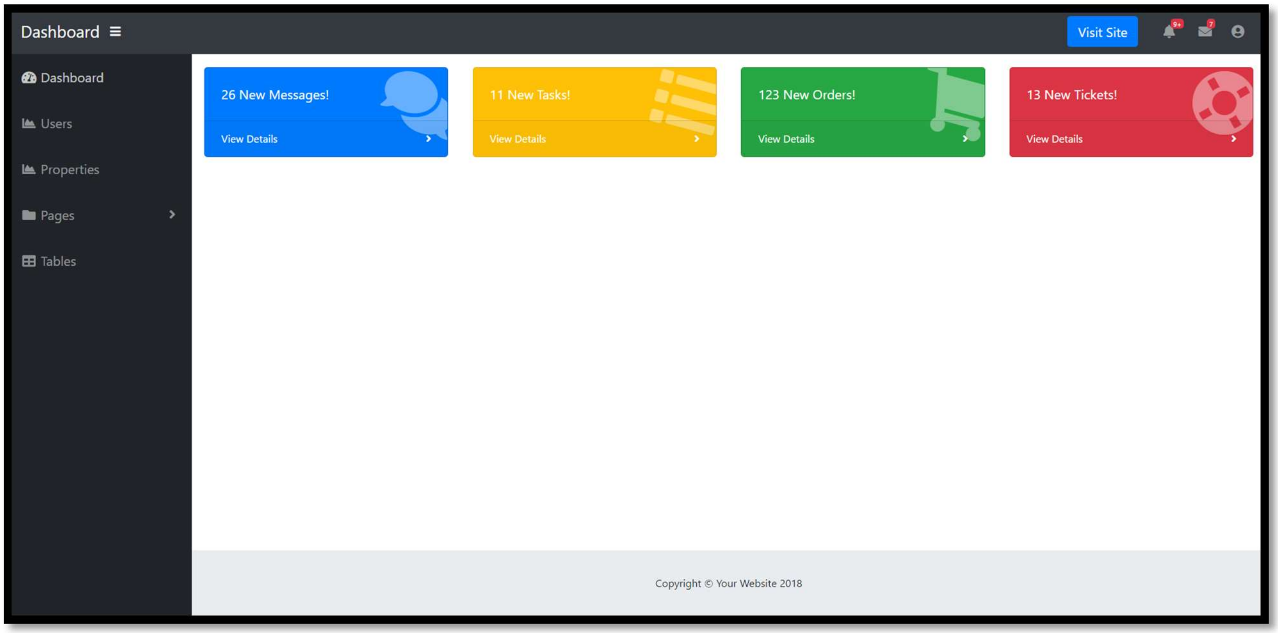
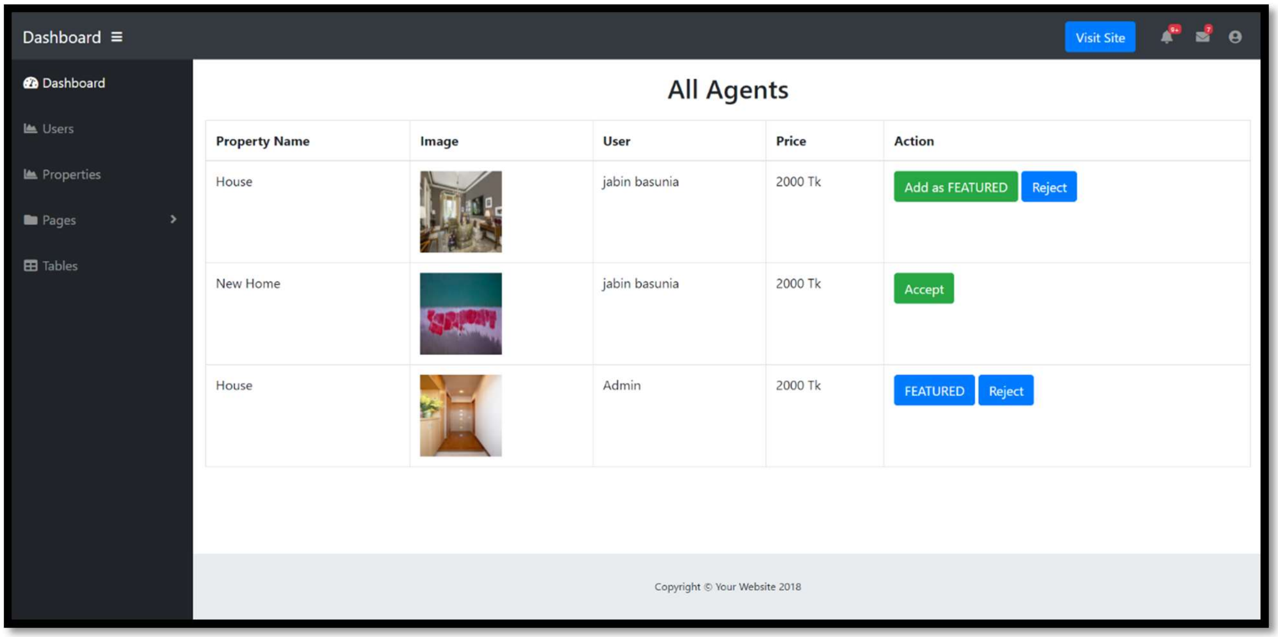


Figure 5.13: Admin dashboard

5.2.2 Implementation of Admin Control:



All Agents

Property Name	Image	User	Price	Action
House		jabin basunia	2000 Tk	Add as FEATURED Reject
New Home		jabin basunia	2000 Tk	Accept
House		Admin	2000 Tk	FEATURED Reject

Figure 5.14: Admin control

5.2.3 Implementation of Home Page:

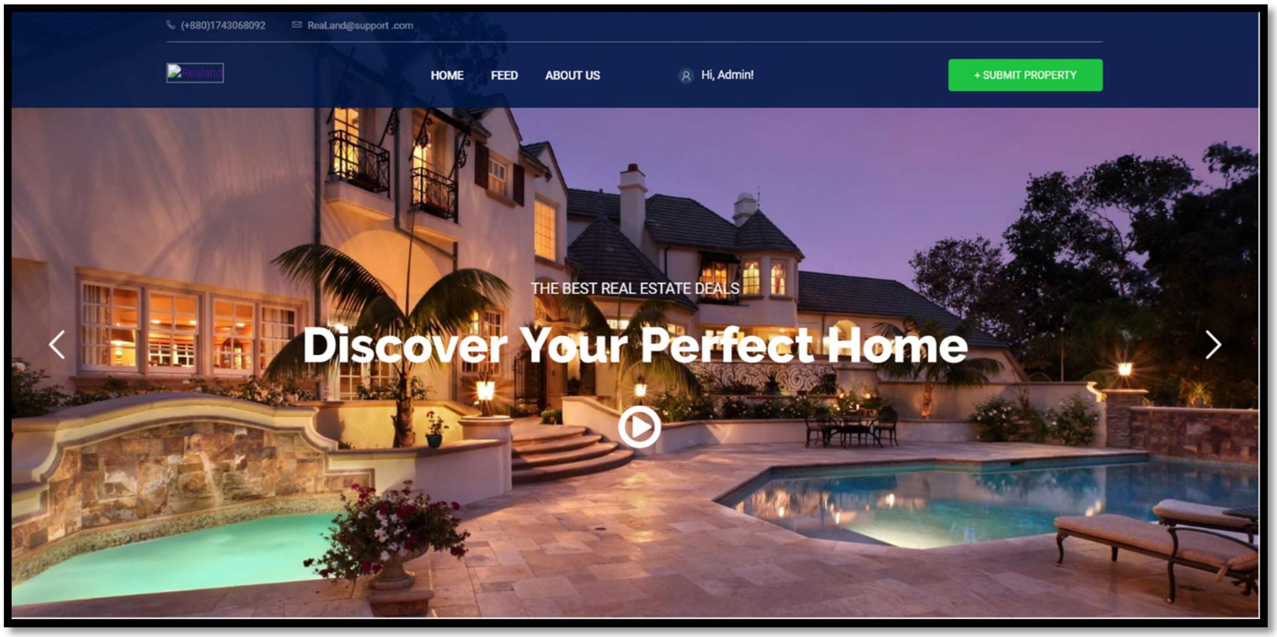
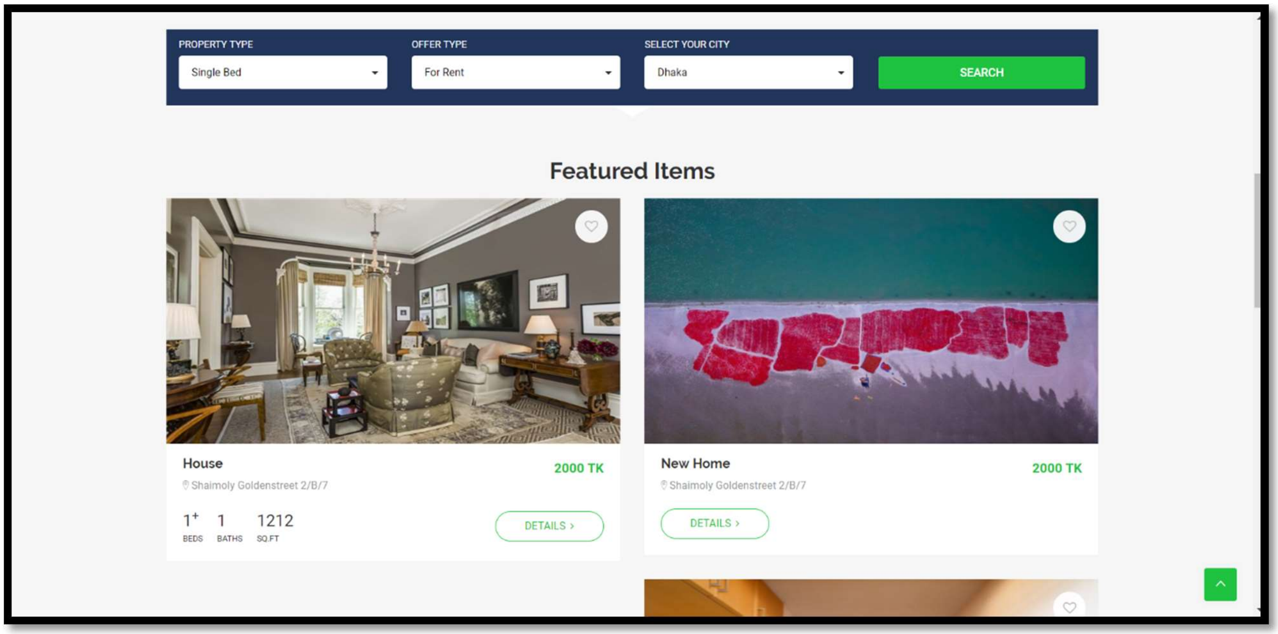


Figure 5.15: Home page

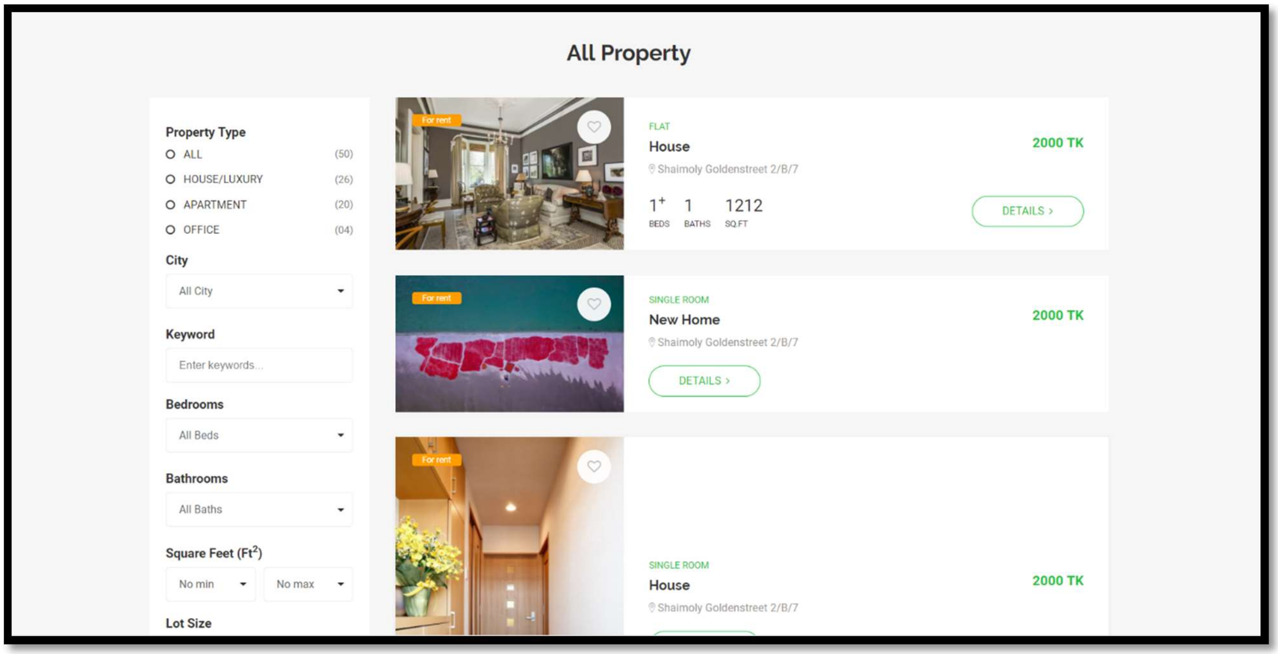
5.2.4 Implementation of Search Property:



PROPERTY TYPE: Single Bed
OFFER TYPE: For Rent
SELECT YOUR CITY: Dhaka

Figure 5.16: Search property

5.2.5 Implementation of Property Details:



All Property

Property Type
○ ALL (50)
○ HOUSE/LUXURY (26)
○ APARTMENT (20)
○ OFFICE (04)

City: All City

Keyword: Enter keywords...

Bedrooms: All Beds

Bathrooms: All Baths

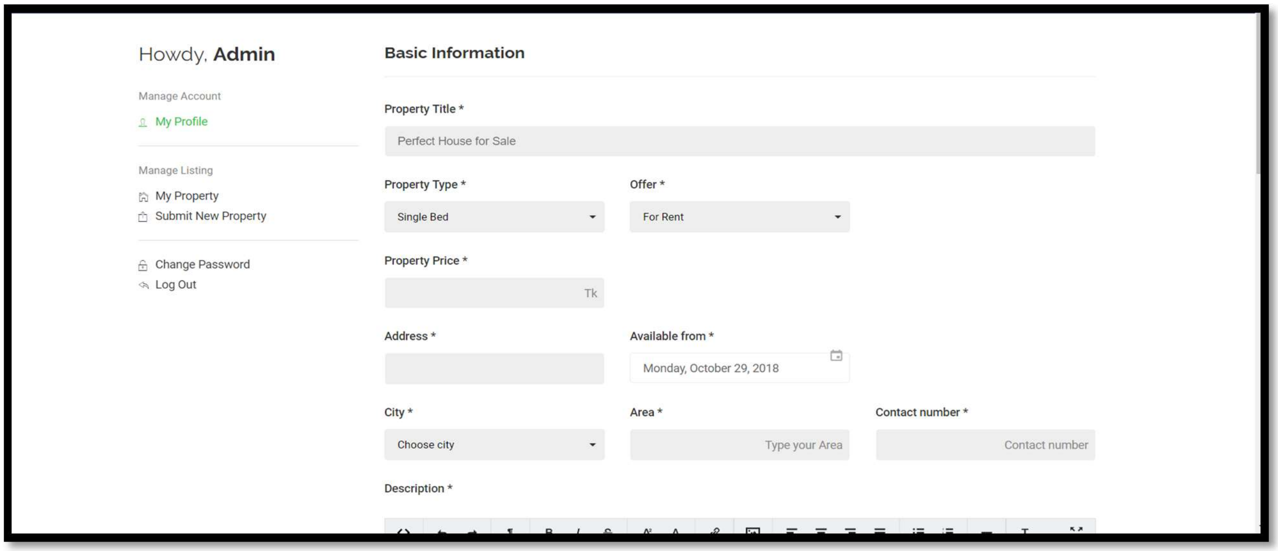
Square Feet (Ft^2): No min, No max

Lot Size

For rent	FLAT House Shaimoly Goldenstreet 2/B/7 1+ BEDS, 1 BATHS, 1212 SQ.FT	2000 TK DETAILS ›
For rent	SINGLE ROOM New Home Shaimoly Goldenstreet 2/B/7 DETAILS ›	2000 TK
For rent	SINGLE ROOM House Shaimoly Goldenstreet 2/B/7	2000 TK

Figure 5.17: Property details

5.2.6 Implementation of Submit Property:



Howdy, **Admin**

Manage Account
- My Profile

Manage Listing
- My Property
- Submit New Property

- Change Password
- Log Out

Basic Information

Property Title *: Perfect House for Sale

Property Type *: Single Bed

Offer *: For Rent

Property Price *: Tk

Address *:

Available from *: Monday, October 29, 2018

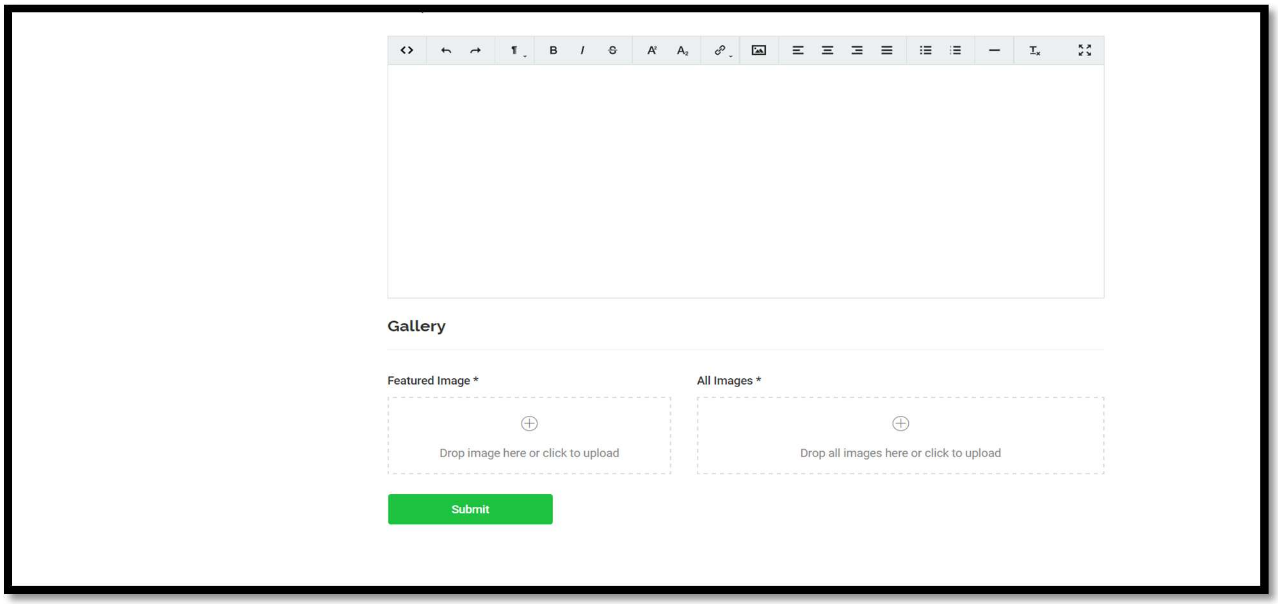
City *: Choose city

Area *: Type your Area

Contact number *: Contact number

Description *

(a)


Gallery
Featured Image *
Drop image here or click to upload
All Images *
Drop all images here or click to upload
Submit

(b)

Figure 5.18: (a) & (b) Submit property

5.2.7 Implementation of User Details:

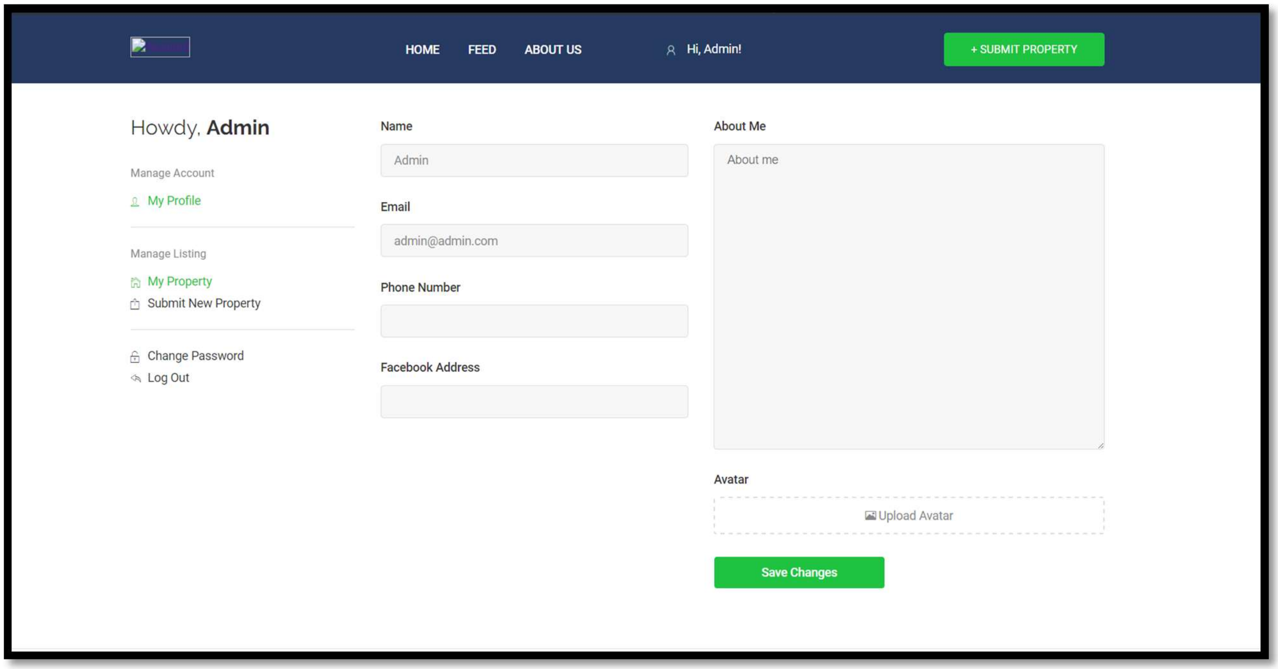


Figure 5.19: User details

5.2.8 Implementation of Android Splash Screen:

Figure 5.20: Splash screen

5.2.9 Implementation of Property View:

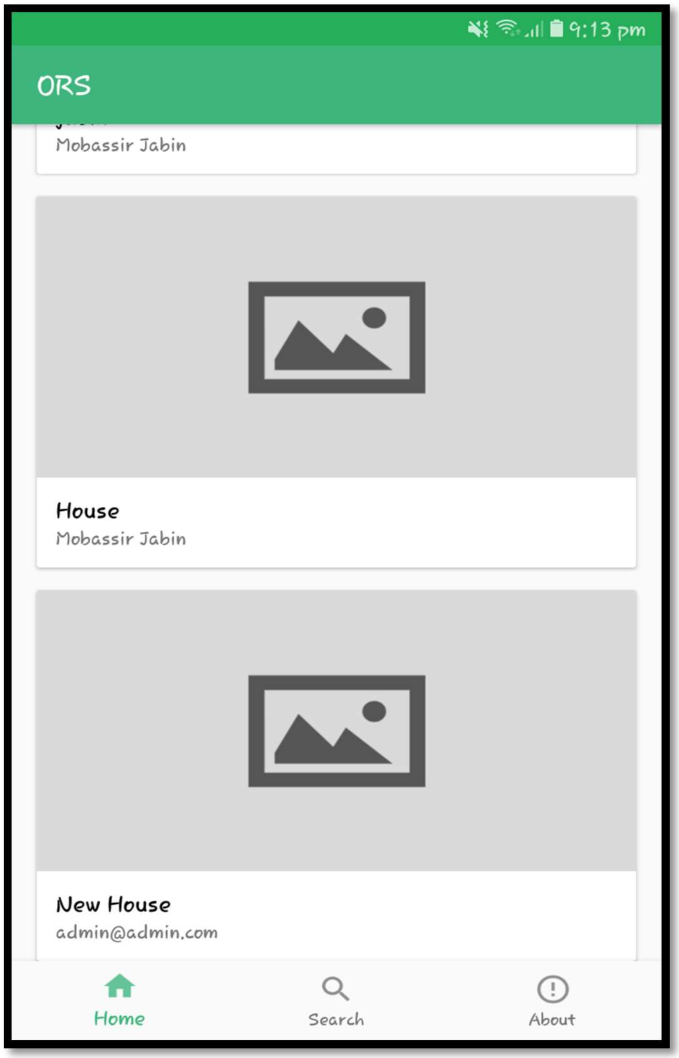
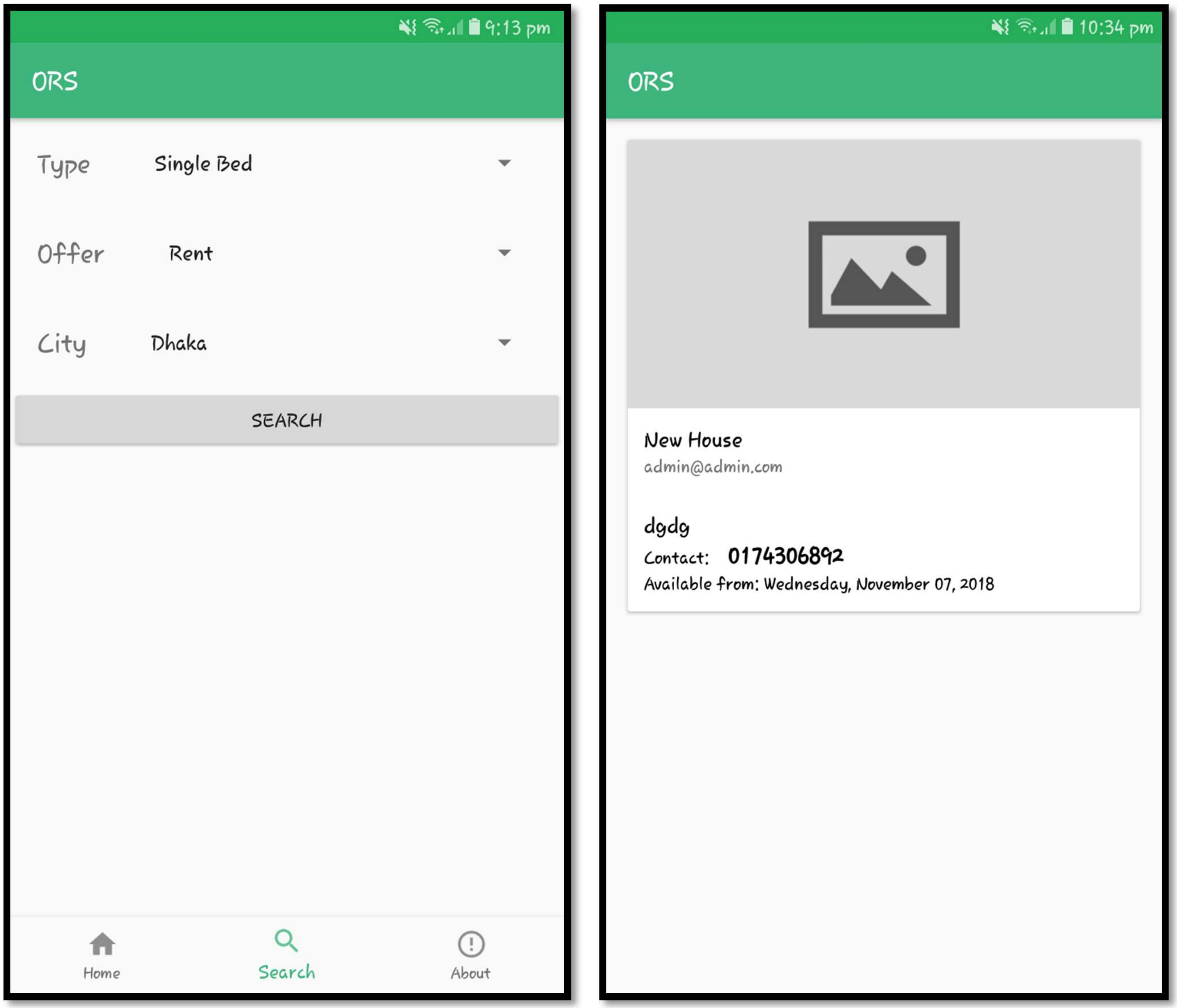


Figure 5.21: Property view

5.2.10 Implementation of Property Search:



Type Single Bed

Offer Rent

City Dhaka

New House

admin@admin.com

dgdg

Contact: 0174306892

Available from: Wednesday, November 07, 2018

(a)

(b)

Figure 5.22: (a) & (b) Property search

5.3 Testing Implementation

A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

5.3.1 Unit Testing

The Unit testing part of a testing methodology is the testing of individual software modules or components that make up an application of system. These tests are usually written by the developers of the module and in a test-driven-development methodology they are actually written before the module is created as part of the specification [14].

5.3.2 Integration Testing

Integration testing part of a testing methodology is the testing of the different modules that have been successfully unit tested when integrated together to perform specific tasks and activities [15].

5.3.3 System Testing

The system testing part of a testing methodology involves testing the entire system for errors and bugs. This test is carried out by interfacing the hardware and software components of the entire system and testing it as a whole [16].

5.3.4 Acceptance Testing

The acceptance testing part of a testing methodology is the final phase of functional software testing and involves making sure that all the product /project requirements have been met and that the end-users and customers have tested the system to make sure it operates as expected and meets all their defined requirements [17].

5.3.5 Performance Testing

There are several different types of performance testing in most testing methodologies, for example: performance testing is measuring how a system behaves under an increasing load testing is verifying that the system can operate at the required response times when subjected to its expected load [18].

5.3.6 Security Testing

Previously, security was something that was tested after-the-fact. With the rise in cyber-crime and the awareness of the risks associated with software vulnerabilities [19].

5.3.7 Compatibility Testing

The compatibility test part of a testing methodology test that the products or application is compatible with all the specified operating system, hardware platforms, web browser , android version , and other designed third party programs [20] .

5.4 Test Results and Reports

Here, we have made a test results and reports table. The name of the first column of the table is Serial Number, second column we have given the name as Type of Testing, similarly we we have added column three as Grade and column four as Remarks. In the first column we just make a serial number. From the second column we can do Unit Test, Link Test etc, in third column we have fixed our grade and the highest possible value of the grade is 10 and in the last column we have shown how many marks we have gotten.

Table 5.1: Result and report

SL#	Type of Testing	Grade	Remarks
1.	Unit Test	10	9
02	Link Test	10	9
03	System Test	10	9
04	Usability	10	8
05	Compatibility Testing	10	8
06	User Training	10	9
07	Security	10	8
08	Performance	10	9
09	Functionality	10	8

CHAPTER 6

Conclusion and Future Scope

6.1 Conclusion

The system will be effective for the people to see nearest houses information also it will help people who may need home. The design of this application is very user friendly, for this who has minimum knowledge about android smartphone and web application can easily use it. We have already mentioned it that the main goal of our application is to save people time, cost and sufferings. This era is a full of modern technology and smart phone is available to almost everyone. So we try to solve this problem by technology. Our application has been successfully tested and implemented. People who has internet access and smartphone can use this application. This application is easy to user. Hopefully people understand our system and take it with advantage.

6.2 Future Scopes

The project made here is just to ensure that this service could be valid in today real challenging world. Here all the facilities are made and tested. In near future the system will be extended for many types of features so that efficiency can be improved. We add secure payment method for secure transaction for booking and sell properties. Add live google maps for get the best result by tracking user live location, in first page user can get only nearby locations properties that was submitted by our system registered users. If we successfully added google maps and user can add location by maps then we can suggestion nearby restaurant, shopping mall, School and Hospital. This suggestion and hints will work automatically and get exact output by tracking location.

References

[1] Bikroy.com << https://bikroy.com/bn/help/about#help-content >> last accessed on 27.august.2018

[2] Lumidi.com << https://www.lamudi.com.bd/about-us/>> last accessed on 27.august.2018

[3] BPM model <<https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation >>>> last accessed on 27. august.2018

[4] Use case modeling << https://www.cs.uct.ac.za/mit_notes/software/htmls/ch03s05.html >> last accessed on 27.july.2018

[5] Use case description <<https://www.usability.gov/how-to-and-tools/methods/use-cases.html>> last accessed on 27.july.2018

[6] Logical data model << https://en.wikipedia.org/wiki/Logical_data_model >> last accessed on 26.july.2018

[7] Tutorial Point <<https://www.tutorialspoint.com/laravel/ >> last accessed on 15.september.2018

[8] User experience design process << https://uxplanet.org/user-experience-design-process-d91df1a45916>> last accessed on 20.september.2018

[9] Java Server << https://www.paginaswebs.com/analisis-del-java-server-pages/ >> last accessed on 20.september.2018

[10] W3 schools << https://www.w3schools.com/php/php_mysql_connect.asp >> last accessed on 10.july.2018

[11] Apache server<< https://www.apachefriends.org/index.html >> last accessed on 20.september.2018

[12]Codementor<<https://www.codementor.io/magarrent/how-to-install-laravel-5-xampp-windows-du107u9ji >>last accessed on 30.september.2018

[13] WDLC<< http://www.xtend2india.com/website-development-life-cycle.php >> last accessed on 30.september.2018

[14] Unit testing<< http://softwaretestingfundamentals.com/unit-testing/ >> last accessed on 30.september.2018

[15] Integration testing << https://www.guru99.com/integration-testing.html >> last accessed on 1.october.2018

[16] System testing << http://softwaretestingfundamentals.com/system-testing-life-cycle/ >> last accessed on 1.october.2018

[17] Acceptance testing<<https://www.tutorialspoint.com/software_testing_dictionary/acceptance_testing.htm>>last accessed on 1.october.2018

[18] Performance testing << https://searchsoftwarequality.techtarget.com/definition/performance-testing>> last accessed on 1.october.2018

[19] Security testing<< https://www.softwaretestinghelp.com/how-to-test-application-security-web-and-desktop-application-security-testing-techniques>> last accessed on 1.october.2018

[20] Forward and backward testing << https://www.guru99.com/compatibility-testing.html >> last accessed on 1.october.2018

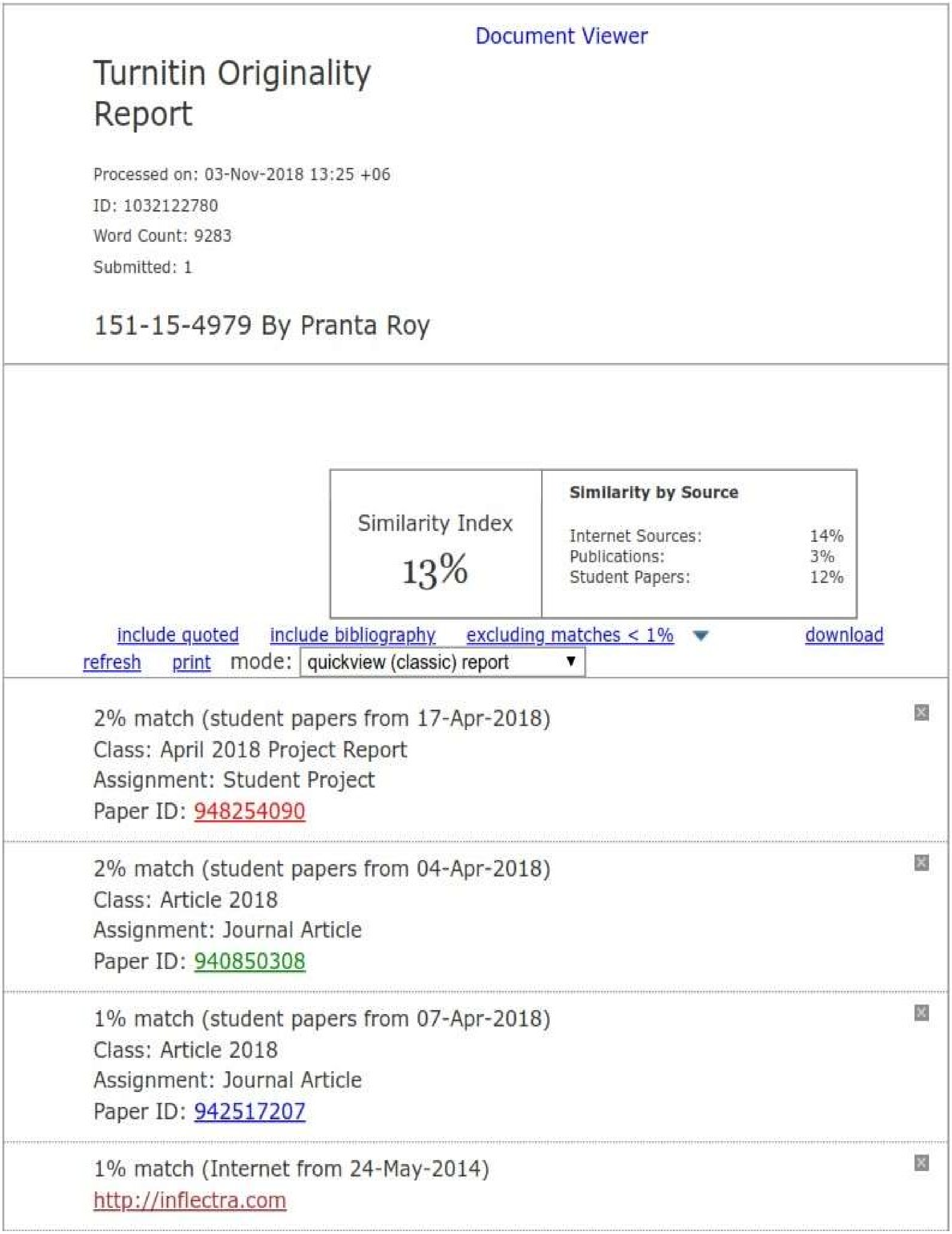
Appendices

Appendix A: Project Reflection

```
@include('admin.partials.navbar')
<div id="wrapper">
@include('admin.partials.sidebar')
<div id="content-wrapper">
<div class="container-fluid">
@yield('content')
</div>
<!-- /.container-fluid -->
<!-- Sticky Footer -->
<footer class="sticky-footer">
<div class="container my-auto">
<div class="copyright text-center my-auto">
<span>Copyright © Your Website 2018</span>
</div>
User
@extends('admin.app')

@section('content')
<div class="row">
<div class="col-md-12">
<h2 class="text-center" style="margin-bottom: 20px">All Agents</h2>
<table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
<th>Facebook</th>
<th>Total property</th>
</tr>
</thead>
<tbody>
@foreach($users as $user)
<tr>
<td>{{ $user->name }}</td>
<td>{{ $user->profile->phone ? $user->profile->phone : "Not Available" }}</td>
<td>{{ $user->profile->facebook ? $user->profile->facebook : "Not Available" }}</td>
<td>{{ count($user->properties) }}</td>
</tr>
@endforeach
</tbody>
</table>
```

```
</div>
</div>
@endsection
@foreach($properties as $property)
<li class="manage-list__item">
<div class="manage-list__item-container">
<div class="manage-list__item-img">
<a href="">
<img style="max-width: 50%"
src="{{ asset('/images/uploads/'.$property->get_featured_image()->location) }}"
alt="Weston Hightpointe Place" class="listing__img">
</a>
</div><!-- manage-list__item-img -->
<div class="manage-list__item-detail">
<h3 class="listing__title">
<a href="">{{ $property->title }}</a>
</h3>
@if(!$property->pending)
<a href="#" class="btn btn-success">Pending</a>
@else
<a href="#" class="btn btn-primary">Accepted</a>
@endif
<p class="listing__location"><span
class="ion-ios-location-outline listing__location-icon"></span>
{{ $property->address }}</p>
<p class="listing__price">{{ $property->price }} Tk</p>
</div>
<script>
$(document).ready(function () {
$('#property-space').attr('disabled', 'disabled');
$('#property-room').attr('disabled', 'disabled');
$('#property-bathrooms').attr('disabled', 'disabled');
$('#property-bedrooms').attr('disabled', 'disabled');
$('#property-parking').attr('disabled', 'disabled');
$('#property-year').attr('disabled', 'disabled');
$('#air-conditioning').attr('disabled', 'disabled');
$('#property_room').hide();
$('#property_space').hide();
$('#info_flat').hide();
$('#info_other').hide();
$('#dk0-listbox li').on('click', function (event) {
//console.log(event.target.innerText);
if (event.target.innerText === "Flat") {
$('#property_room').show();
$('#property_space').show();
$('#info_flat').show();
$('#info_other').show();
$('#property-space').removeAttr('disabled');
$('#property-room').removeAttr('disabled');
$('#property-bathrooms').removeAttr('disabled');
$('#property-bedrooms').removeAttr('disabled');
$('#property-parking').removeAttr('disabled');
$('#property-year').removeAttr('disabled');
$('#air-conditioning').removeAttr('disabled');
} else {
$('#property_room').hide();
$('#property_space').hide();
$('#info_flat').hide();
$('#info_other').hide();
$('#property-space').attr('disabled', 'disabled');
$('#property-room').attr('disabled', 'disabled');
$('#property-bathrooms').attr('disabled', 'disabled');
$('#property-bedrooms').attr('disabled', 'disabled');
$('#property-parking').attr('disabled', 'disabled');
$('#property-year').attr('disabled', 'disabled');
$('#air-conditioning').attr('disabled', 'disabled');
}
});
});
</script>
@endsection
@section('scripts')
<script>
$(document).ready(function () {
var li = $('#dk0-listbox').find('li.dk-option-selected');
console.log(li.text().trim());
if (li.text().trim() !== "Flat") {
$('#property-space').attr('disabled', 'disabled');
$('#property-room').attr('disabled', 'disabled');
$('#property-bathrooms').attr('disabled', 'disabled');
$('#property-bedrooms').attr('disabled', 'disabled');
$('#property-parking').attr('disabled', 'disabled');
$('#property-year').attr('disabled', 'disabled');
$('#air-conditioning').attr('disabled', 'disabled');
$('#property_room').hide();
$('#property_space').hide();
$('#info_flat').hide();
$('#info_other').hide();
}
$('#dk0-listbox li').on('click', function (event) {
```



Turnitin Originality Report

Processed on: 03-Nov-2018 13:25 +06

ID: 1032122780

Word Count: 9283

Submitted: 1

151-15-4979 By Pranta Roy

Similarity Index

13%

Similarity by Source

Internet Sources:	14%
Publications:	3%
Student Papers:	12%

mode: quickview (classic) report

2% match (student papers from 17-Apr-2018)
Class: April 2018 Project Report
Assignment: Student Project
Paper ID: 948254090

2% match (student papers from 04-Apr-2018)
Class: Article 2018
Assignment: Journal Article
Paper ID: 940850308

1% match (student papers from 07-Apr-2018)
Class: Article 2018
Assignment: Journal Article
Paper ID: 942517207

1% match (Internet from 24-May-2014)
http://inflectra.com