VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI-590018



## A MINI PROJECT REPORT ON

## FOOD ORDERING SYSTEM

Submitted in partial fulfilment of the requirements

For the award of degree of

Bachelor of Engineering

In

Computer Science and Engineering

By

MADHU M PANDURANGI

[1KS15CS052]

Under the guidance of



Department of Computer Science & Engineering

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakpura Main Road, Bengaluru-560109

2017-2018

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakpura Main Road, Bengaluru-560109

Department of Computer Science & Engineering



CERTIFICATE

This is to certify that mini project work entitled “FOOD ORDERING SYSTEM” carried out by Mr.Madhu M Pandurangi bearing USN 1KS15CS052 bonafide student of K.S.Institute of Technology in the partial fulfilment for the award of the Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi, during the year 2017. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini Project work prescribed for the said degree.

Dr. Rekha. B. Venkatapur Dr. T.V. Govindaraju

Prof & HOD ,CS & E Department Principal/Director,KSIT

Ms. Sougandhika Narayan Mr. K Venkata Rao Mr. Kumar K

Asst. Prof, Dept. Of CSE Assoc. Prof, Dept. Of CSE Asst. Prof, Dept. Of CSE

Name of the Examiners Signature

with date 1.

2.

ACKOWLEDGEMENT

I take this opportunity to thank one and all involved in building this project. Firstly I would like to thank the college for providing us an opportunity to work on the project.

First and foremost, I thank the management of K. S. Institute of Technology for

providing all the resources required for the project.

I wish to acknowledge my sincere gratitude to our beloved Principal, Dr. T.V. Govindaraju for his encouragement and providing all facilities for the accomplishment of this project.

This project would not have been possible without the support of our beloved Prof & HOD, Dr. Rekha.B.Venkatapur, Dept. of CSE.

I am also highly grateful to my project guide, Ms. Sougandhika Narayan, Mr. K Venkata Rao and Mr. Kumar K, Dept. of CSE who have been very generous in

assisting and supporting ,to do this Project “Real Estate Listing “, which formally

started as just a rough idea and now resulted in this.

I also would like to thank all other teaching and non teaching staff members who

had extended their support and co-operation while bringing up this project.

Madhu M Pandurangi

(1KS15CS052)

Abstract

The food ordering system provides convenience for the customers. It overcomes the

disadvantages of the traditional queuing system. This system increases the takeaway of

foods than visitors. Therefore, this system enhances the speed and standardization of taking the order from the customer.

The food ordering system set up menu and the customers easily places the order with a simple mouse click. This system allows the user to select the desired food items from the displayed menu. The use r’s details are maintained confidential because it maintains a separate account for each user. An id and password is provided for each user. Therefore it provides a more secured ordering.

INDEX

Page

1. Introduction

2

# Chapter 1

Introduction

An online food ordering system is a web-based application that stimulates the foodies (customers) to put food orders through internet by locating their favorite restaurant or nearest one. This application is based on the asp.net platform .

## 1.1 Project

In today’s age of fast food and take-out, many restaurants have chosen to focus on quick

preparation and speedy delivery of orders rather than offering a rich dining experience.

Until very recently, all of these delivery orders were placed over the phone, but there are

many disadvantages to this system, including the inconvenience of the customer needing to

have a physical copy of the menu, lack of a visual confirmation that the order was placed

correctly, and the necessity for the restaurant to have an employee answering the phone and taking orders.

The main advantage of my system is that it greatly simplifies the ordering process for both the customer and the restaurant. After making a selection, the item is then added to their order, which the customer can review the details of at any time before submitting out. This provides instant visual confirmation of what was selected and ensures that items in the order are, in fact, what was intended.

## 1.2 DBMS

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create,

retrieve, update and manage data.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains

easily accessible.

A DBMS provides concurrency, security, data integrity, consistency, controls redundancy and data independence.

In this project the Relational DBMS (RDBMS) used is MySQL. It is an open source software which uses SQL (Structured Query Language) which is a standard language for

storing, manipulating and retrieving data in databases.

## 1.3 Java connections

To connect the database with the front end we use a java connector JDBC (Java Database Connectivity). JDBC is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation.

To achieve connectivity we use JSPs (JavaServer Pages) and Servlets in this project. JavaServer Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. JSP is similar to

PHP and ASP, but it uses the Java programming language.

# Chapter 2

Requirements Specification

A computerized way of handling information about property and users details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a

database driven web application whose requirements are mentioned in this section.

## 2.1 Overall Description

A reliable and scalable database driven web application with security features, that is easy

to use and maintain is the requisite.

## 2.2 Specific Requirements

The specific requirements of the Food Ordering System are stated as follows:

### 2.2.1 Software Requirements

Technology used:

* Front end – JSP
* Controller – JSP/Servlets
* Backend – SQL

Software:

* IDE - Netbeans 8.2
* Database support - MySQL 5.7
* Operating system – Windows 8 and above
* Server deployment - Glassfish server

Technology:

* HTML is integrated in JSP. It provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and

hyperlinks.

* Javascript is a scripting language which supports the development of both client and server applications. It is preferred at client side to write programs that can be

executed by a web browser within the context of a web page.

* CSS(Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document,
* SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data

manipulation.

* Java Server pages is a simple yet powerful technology for creating and maintaining dynamic-content web pages. It is based on the Java programming language. It can be thought of as an extension to servlet because it provides more functionality than servlet A JSP page consists of HTML tags and JSP tags. The jsp pages are easier to maintain than servlet because we can separate designing and

development.

We require a JDBC connection between the front end and back end components to write

to the database and fetch required data.

### 2.2.2 Hardware Requirements

* Processor – Pentium IV or above
* RAM – 2 GB or more
* Hard disk – 3 GB or more

# Chapter 3

Detailed Design

## 3.1 System design

The web server needs a JSP engine, i.e, a container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages. A server(generally referred to as application or web server) supports the Java Server Pages. This server will act as a mediator between the client browser and a database. The following diagram

shows the JSP architecture.

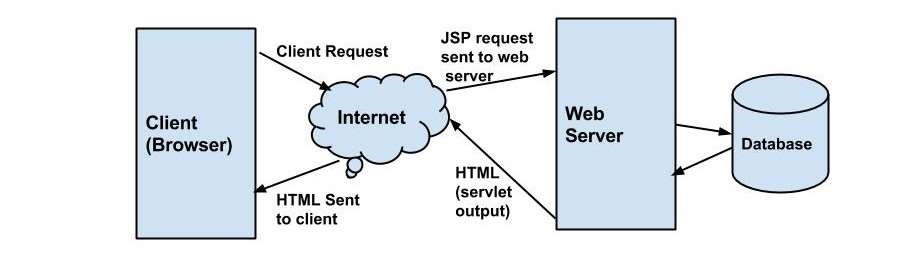


Fig 3.1 JSP architecture

Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic(constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending

partially processed data between the database server and the client.

Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses,

agencies and institutions.

A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database architects may use one of several languages to create databases, such as structured query language (SQL). SQL is a database computer language designed

for the retrieval and management of data in a relational database.

## 3.2 ER Diagram

An entity–relationship model is usually the result of systematic analysis to define and

describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies

between entities.

An ER model can also be expressed in a verbal form, for example: one building may be divided into zero or more apartments, but one apartment can only be located in one

building.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign

key" in the table of another entity

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of

specification, and is different from the three schema approach to software engineering.

While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-

existing information system.

Three main components of an ERD are the entities the relationship between those entities,

and the cardinality, which defines that relationship in terms of numbers.

Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional (for example, an employee rep could have no customers or could have many) or mandatory (for example, there must be at least one product listed in an order.)

The three main cardinal relationships are:

* One-to-one (1:1) - For example, if each customer in a database is associated with

one mailing address.

* One-to-many (1:M) - For example, a single customer might place an order for

multiple products. The customer is associated with multiple entities, but all those

entities have a single connection back to the same customer.

* Many-to-many (M:N). For example, at a company where all call center agents

work with multiple customers, each agent is associated with multiple customers,

and multiple customers might also be associated with multiple agents.

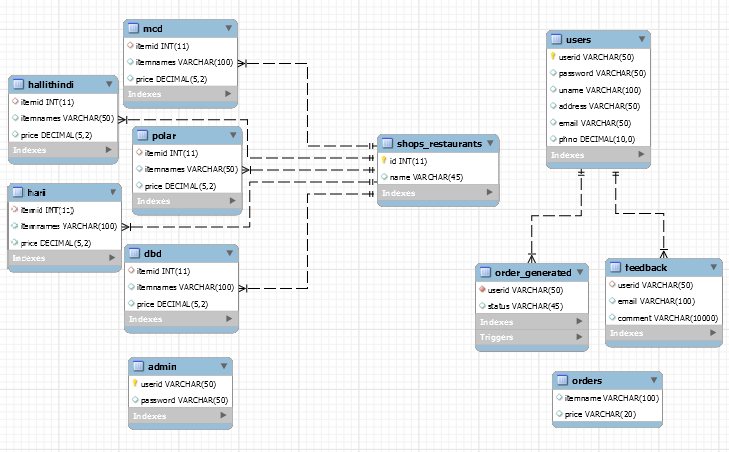


Fig 3.2 E-R diagram for real estate listing system with tables

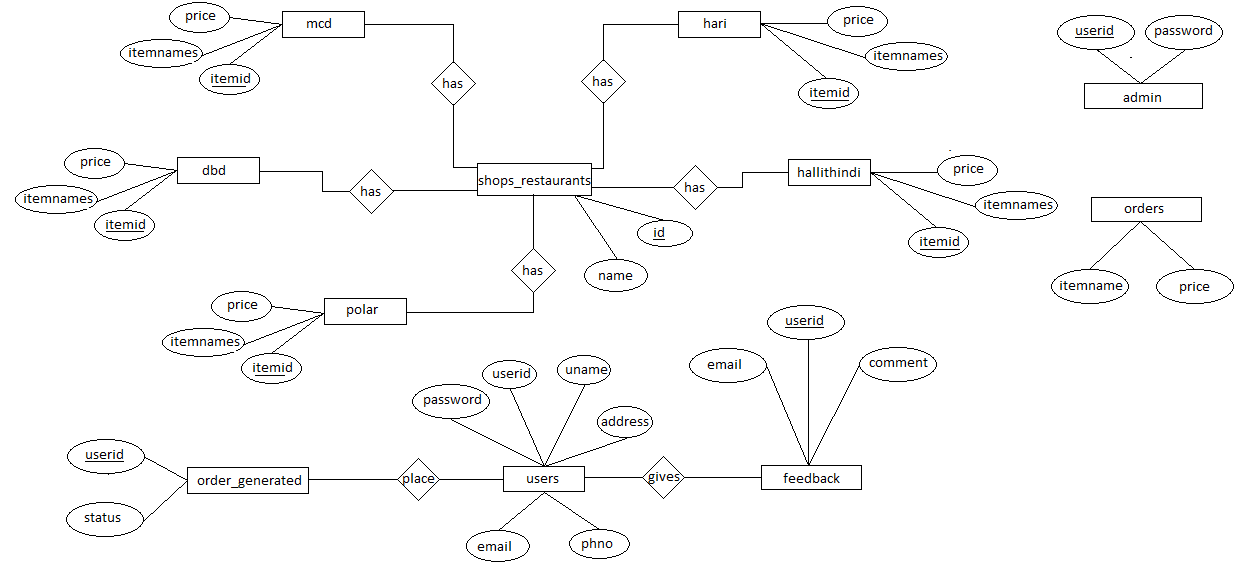


Fig 3.3 E-R diagram

## 3.3 Relational Schema

The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database.

A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted byan arrow with the head pointing at the referenced key attribute.

A schema diagram helps organize values in the database. It also gives an idea of what order the tables should be created in. The following diagram shows the schema diagram

for the database.

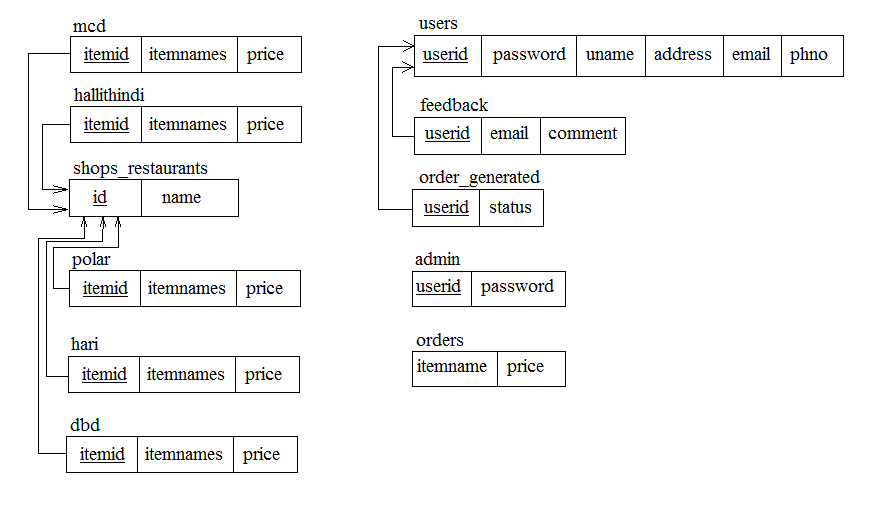


Fig 2.3 Schema diagram

## 3.4 Description of Tables

The database miniproject consists of eleven tables:

1. Users - storing user details
2. Admin - storing admin details.
3. Orders - contains order details
4. Order\_generated - contains order status.
5. Feedback - contains name, email and comment of the user.
6. Shops\_restaurants - contains id and shops and restaurants name.
7. Mcd - contains id, menu list and price.
8. Hallithindi - contains id, menu list and price.
9. Polar - contains id, menu list and price.
10. Hari - contains id, menu list and price.
11. Dbd - contains id, menu list and price.

USERS

* Userid – Primary key. Identifies user uniquely. This is used

during login.

* Uname – Username given by user during registration.
* Password – Used during login.
* Address – Address of the user.
* email – email id of the user.
* Phno – phone number of the user.

ADMIN

* Userid – Primary key. Identifies user uniquely. This is used

during login.

* Password – Used during login.

ORDERS

* Itemname – ordered food item name.
* Price – Price of food item.

ORDER\_GENERATED

* UserID – Foreign key. Referenced to userID in Users table.
* Status – Status of the order.

SHOPS\_RESTAURANTS

* ID – Primary key. Identifies shops and restaurants uniquely
* Name – Name of the shops and restaurants.

McD

* Itemid – Foreign key . Referenced to id in shops\_restaurants table.
* Itemnames – Food item names.
* Price – Price of food items.

POLAR

* Itemid – Foreign key . Referenced to id in shops\_restaurants table.
* Itemnames – Food item names.
* Price – Price of food items.

HALLITHINDI

* Itemid – Foreign key . Referenced to id in shops\_restaurants table.
* Itemnames – Food item names.
* Price – Price of food items.

HARI

* Itemid – Foreign key . Referenced to id in shops\_restaurants table.
* Itemnames – Food item names.
* Price – Price of food items.

DBD

* Itemid – Foreign key . Referenced to id in shops\_restaurants table.
* Itemnames – Food item names.
* Price – Price of food items.

# Chapter 4

Implementation

## 4.1 Modules and their requirements

1. User registration and login:

A new user can register by providing their full name and contact details with a unique username. A user ID is unique which identifies the user.

The user can login with their username and password.

1. Web Ordering system:

This module provides the functionality for customers to place their order and supply necessary details. Users of the system, namely restaurant customers, must be provided the following functionality:

• Log in to the system.

• Click the restaurant’s menu.

• Select an item from the menu.

• Add an item to their current order.

• Review their current order.

• Place an order.

1. Menu management module:

This module provides functionality for the power user-Administrator only. It will not be available to any other users of the system like Restaurant Employees or Customers. Using a graphical interface, it will allow an Admin to manage the menu that is displayed to users of the web ordering system: (1).Add food items to the menu. (2). Delete food item /from the menu.

Before customers can actually use this system, functionality provided by this component will have to be configured first. Once the initial configuration is done, this will be the least likely used component as menu updates are mostly seasonal and do not occur frequently.

1. View order.

A user can view a list of food items they have ordered.

Scope:

This system is flexible and efficient and allows easy access to owner information. Security, speed and accuracy should be focused on. It is a user friendly system and can overcome some user validation checks. A user cannot upload any property if they are not

registered in the database with suitable details.

All the property information is suitably maintained on the server and can be accessed when required. It identifies various sources of information and accordingly provides

access to the requested details.

The system maintains details such as type of property, dimensions, amenities available,

etc.

Triggers and stored procedures:

The project makes use of a trigger to delete the item list from the order table after submitting and getting a confirmation

A stored procedure is used as a

## 4.2 Result

The resulting system is able to:

* Authenticate user credentials during login
* Allows user to quickly and easily look for the property they wish to buy or rent
* The user can upload their property for sale or rent
* Gives accurate information as uploaded by the owner

# Chapter 5

Testing

## 5.1 Software testing

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been

defined and then implemented in testable programs.

## 5.2 Module testing and integration

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows to implement parallelism into the testing process by giving the

opportunity to test multiple modules simultaneously.

In this Real Estate Listing System, when a user logs in or registers, their user id for the session is maintained internally. When a user wants to upload property, their user ID is automatically saved into the database for the corresponding property. The system displays all property uploaded by every user when a user wants to search for property. Further details of property can be viewed. A unique property ID is maintained for each uploaded property which is referenced in the further details in the corresponding house, apartment or site tables. Hence all the modules are linked by identifying entities that are maintained

in the database.

Testing has been conducted using various test cases. The system sustains some errors such as invalid pincode. A user cannot upload property unless they are logged into the

system.

Limitations:

The Current project can only maintain details of the property. It does not allow interaction

between users. This could be turned into an e-commerce platform with suitable resources. Edit access isn't allowed to any user as it requires some kind of user authentication

process. This will be implemented in future versions of the project.

# Chapter 6

Snapshots

This chapter consists of working screenshots of the project with code snippets. 6.1 Registration and login

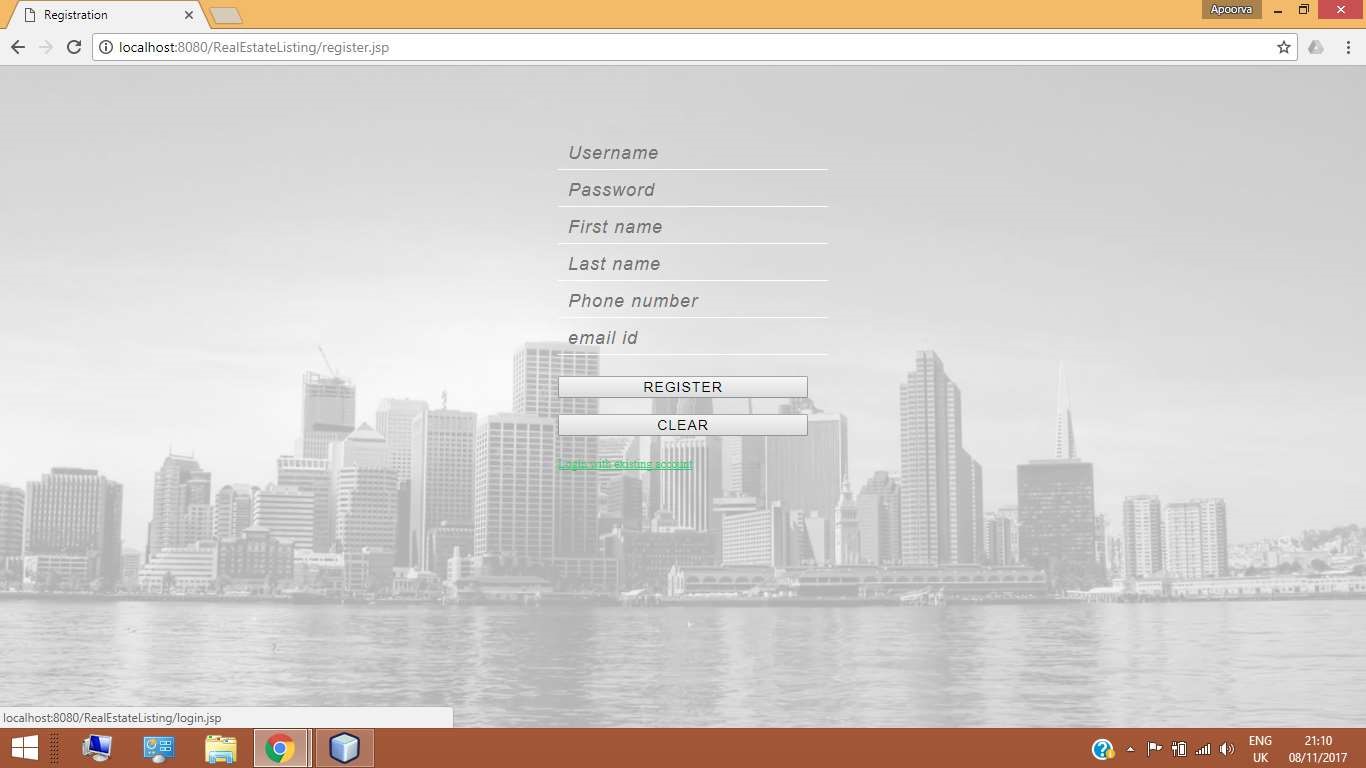


Fig 6.1 The register page

A new user can register with a unique username, password, full name, phone number and email id. A unique user id identifies each user in the back end. The user id is created in

the database on successful registration. The user is then redirected to the homepage.

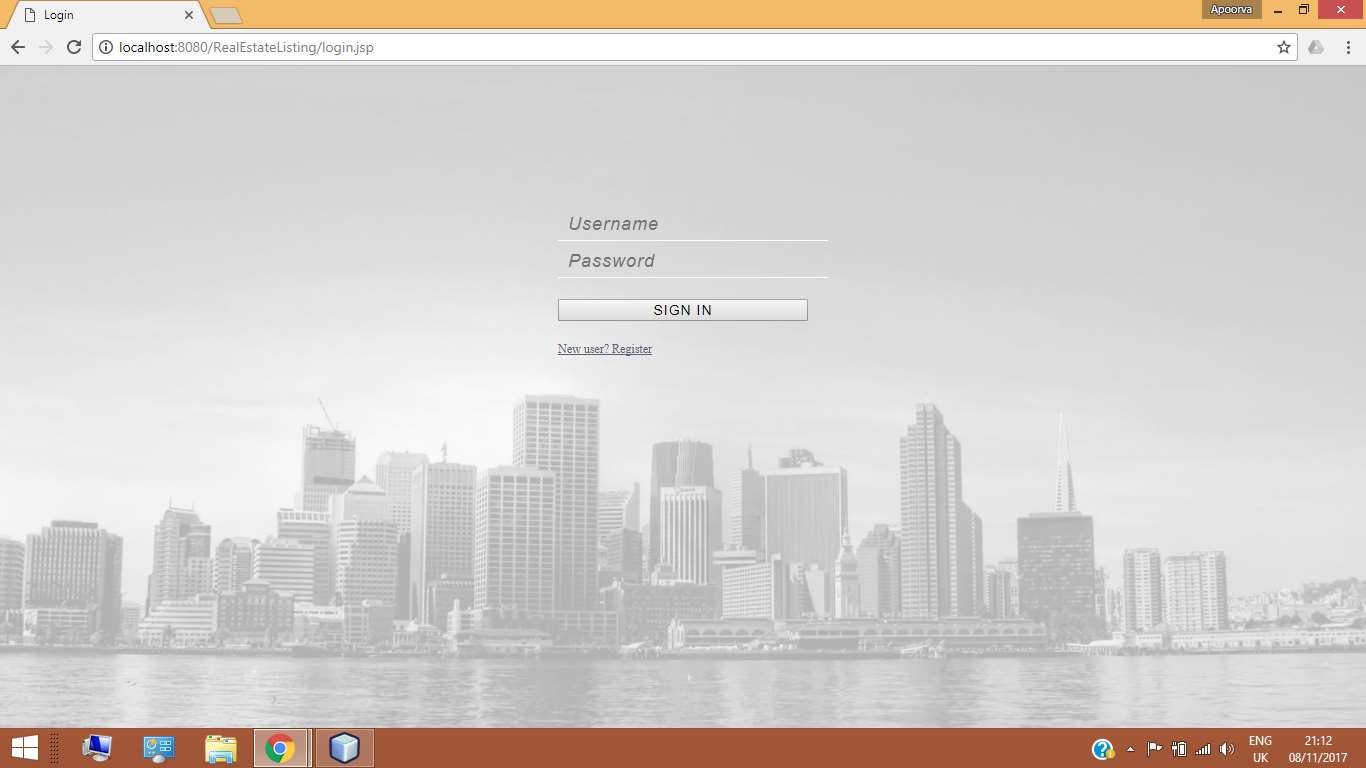


Fig 6.2 The login page

A previously registered user can login through this page with valid username and

password.

Class.forName("com.mysql.jdbc.Driver").newInstance(); java.sql.Connection con =

DriverManager.getConnection("jdbc:mysql://localhost:3306/realestate","root","root"); PreparedStatement pst = con.prepareStatement("Select username,password from users where username=? and password=?");

pst.setString(1, username); pst.setString(2, pwd);

ResultSet rs = pst.executeQuery();

CallableStatement cstmt = con.prepareCall("{call PR\_GET\_USER\_ID(?, ?)}");

cstmt.setString(1, username); cstmt.registerOutParameter(2, java.sql.Types.INTEGER);

cstmt.execute(); usid =cstmt.getInt(2); uid=""+usid; session.setAttribute("uid",uid); The above snippet makes use of stored procedure PR\_GET\_USER\_ID to check username

and return userid.

## 6.2 The Homepage

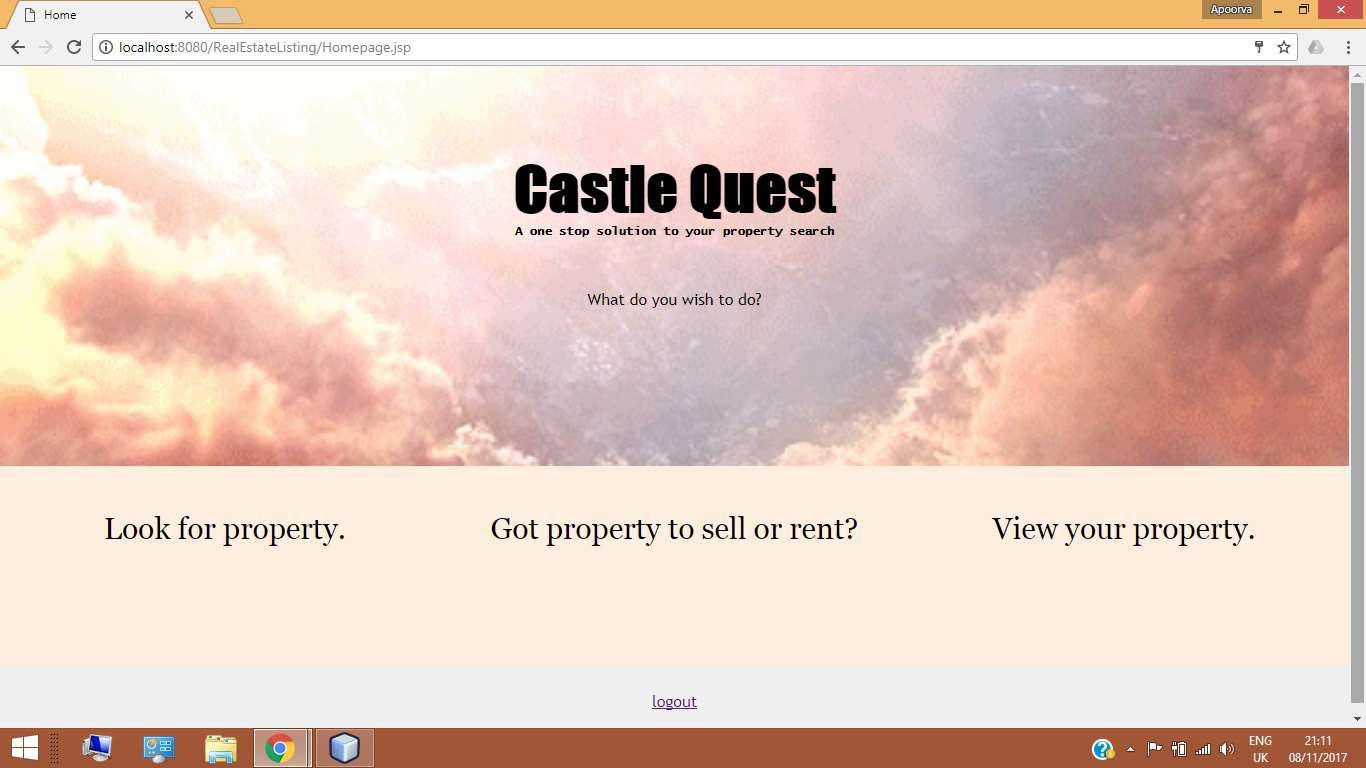


Fig 6.3 The homepage

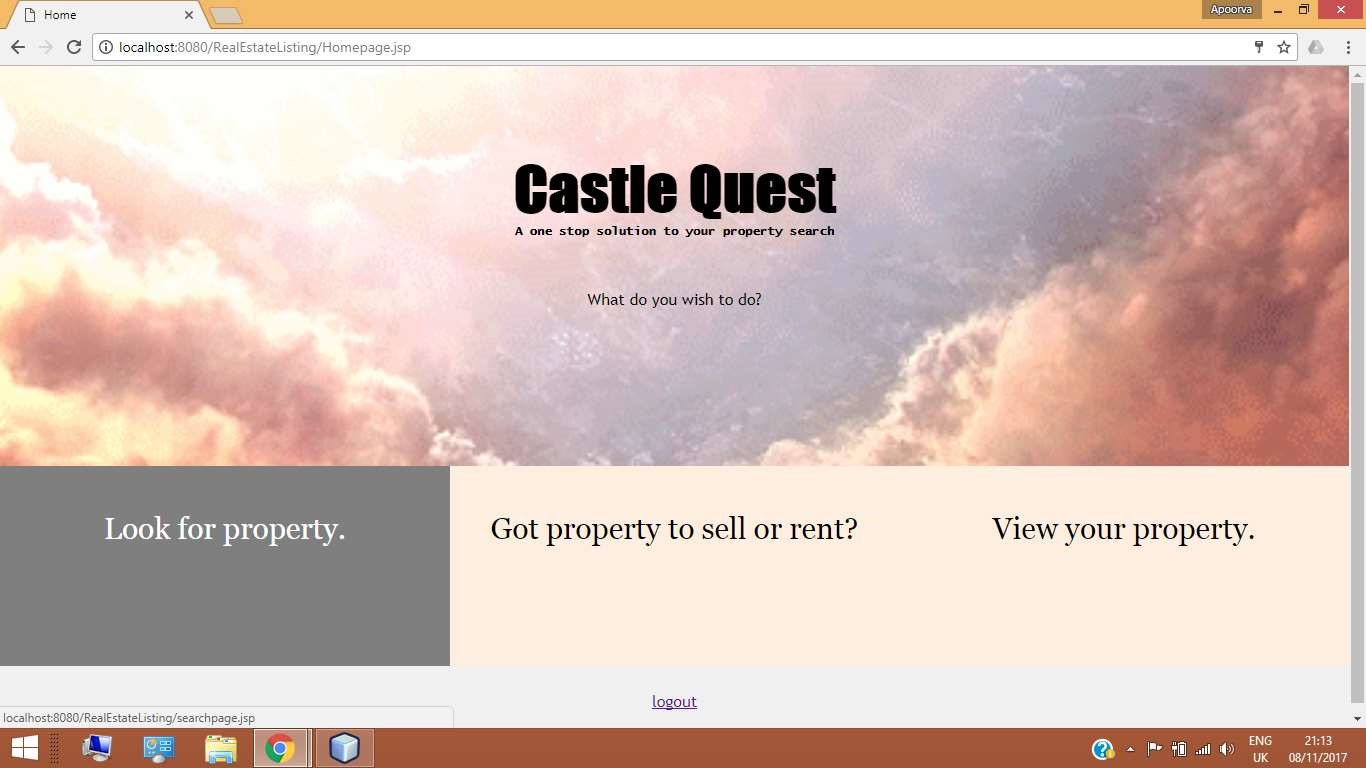


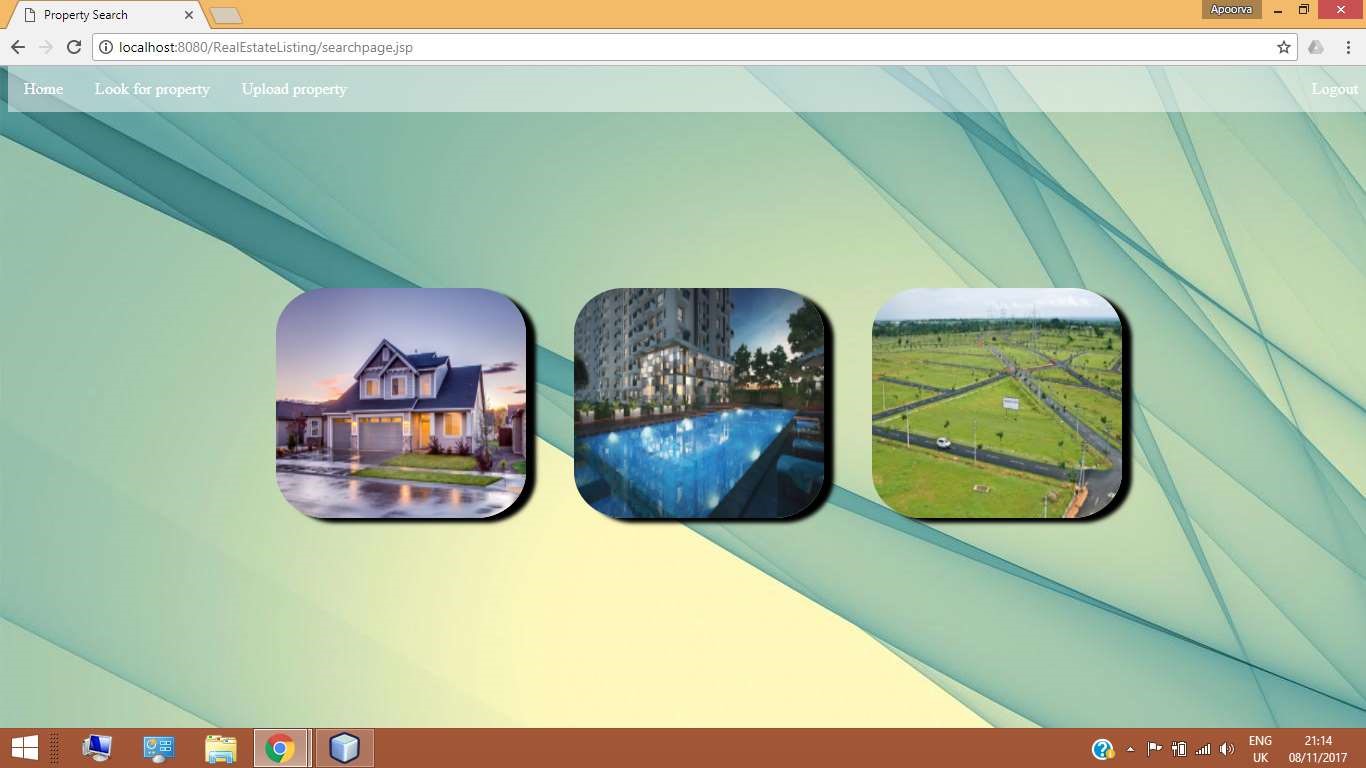
Fig 6.4 Homepage navigation

The user is redirected tot the homepage on successful registration or login. They will have

three options to navigate from here-

1. Look for property uploaded by different users.
2. Upload property to sell or rent.
3. View property they have uploaded.

## 6.3 Find property



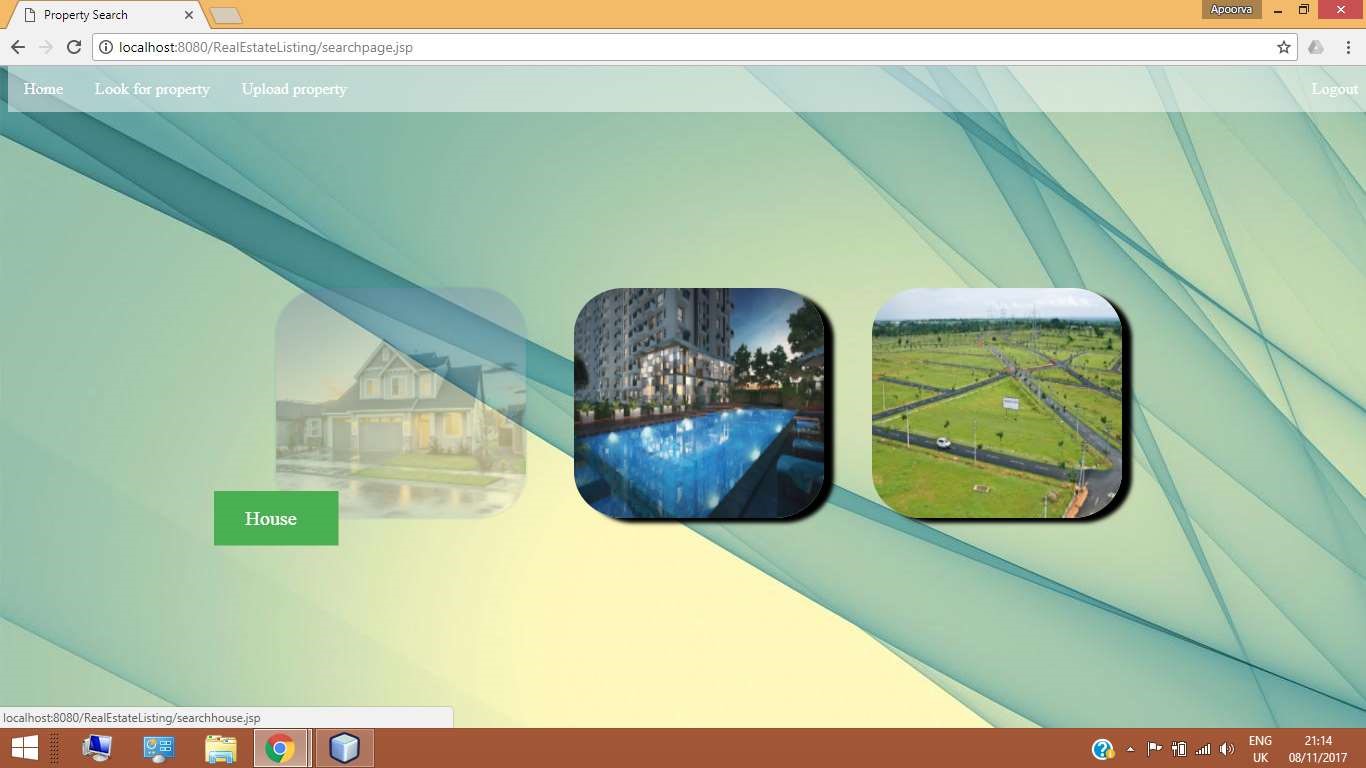


Fig 6.5 Find Property

The user can look for property based on whether they are looking for a House, Apartment or a Site by navigating from this page. The navigation bar allows the user to go anywhere

from this page.

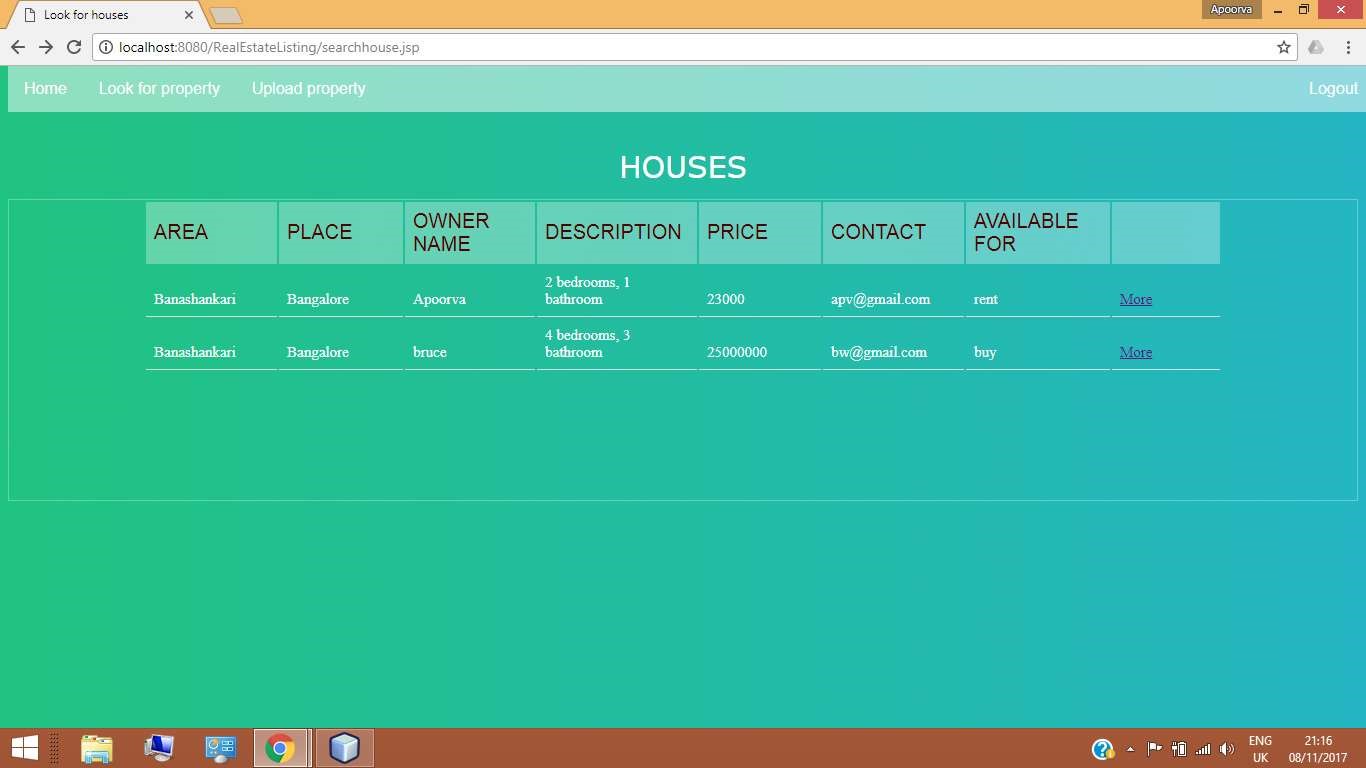


Fig 6.6 Houses available

Suppose a user wants to look for houses. All the houses details uploaded by different users with their details are available. They can view more details about a certain property

they are interested in.

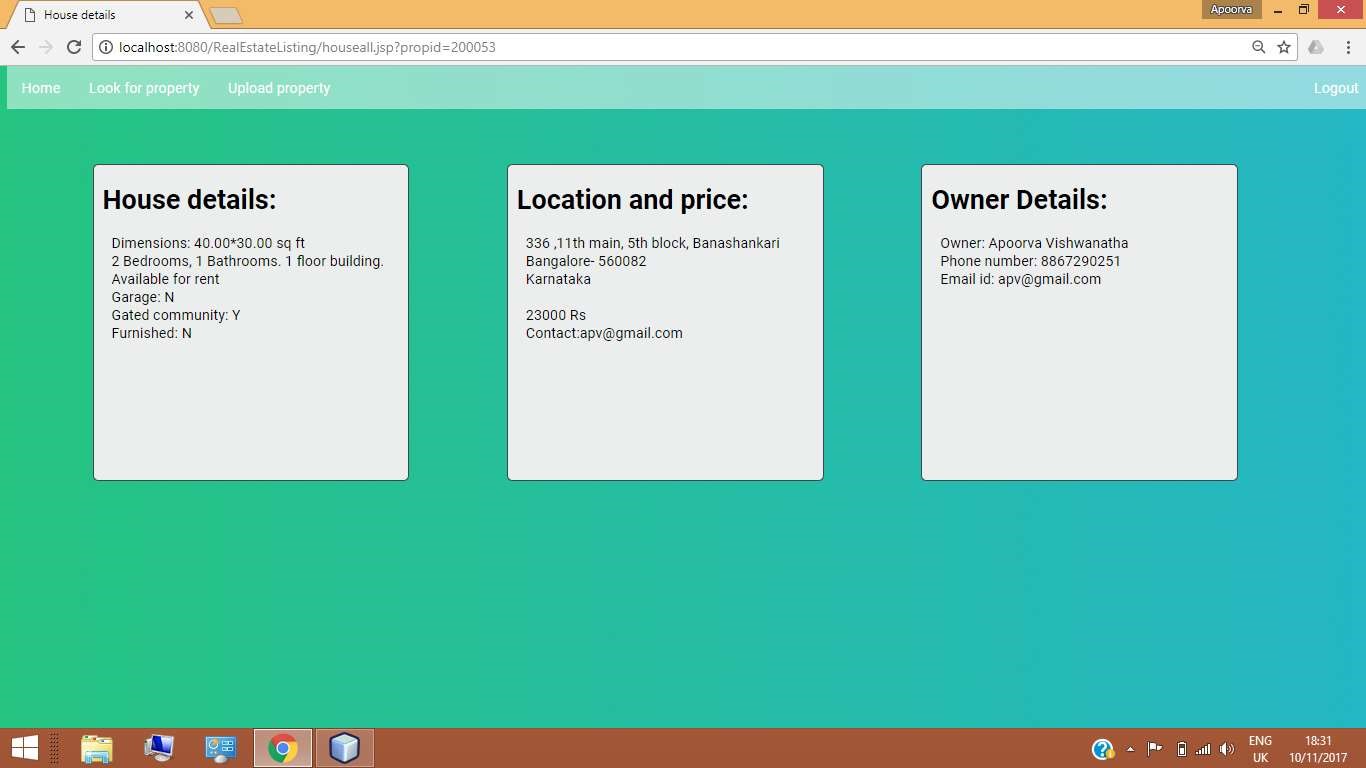


Fig 6.7 Details

The owner details, address and in depth description of the house can be viewed.

We get property id sent through the url with the following jsp code:

<%

String propid=request.getParameter("propid");

%>

This property id can be used for querying:

<% String QueryString1 = "SELECT \* from property where propertyID='"+propid+"'"; rs = statement.executeQuery(QueryString1); %>

## 6.4 Upload property

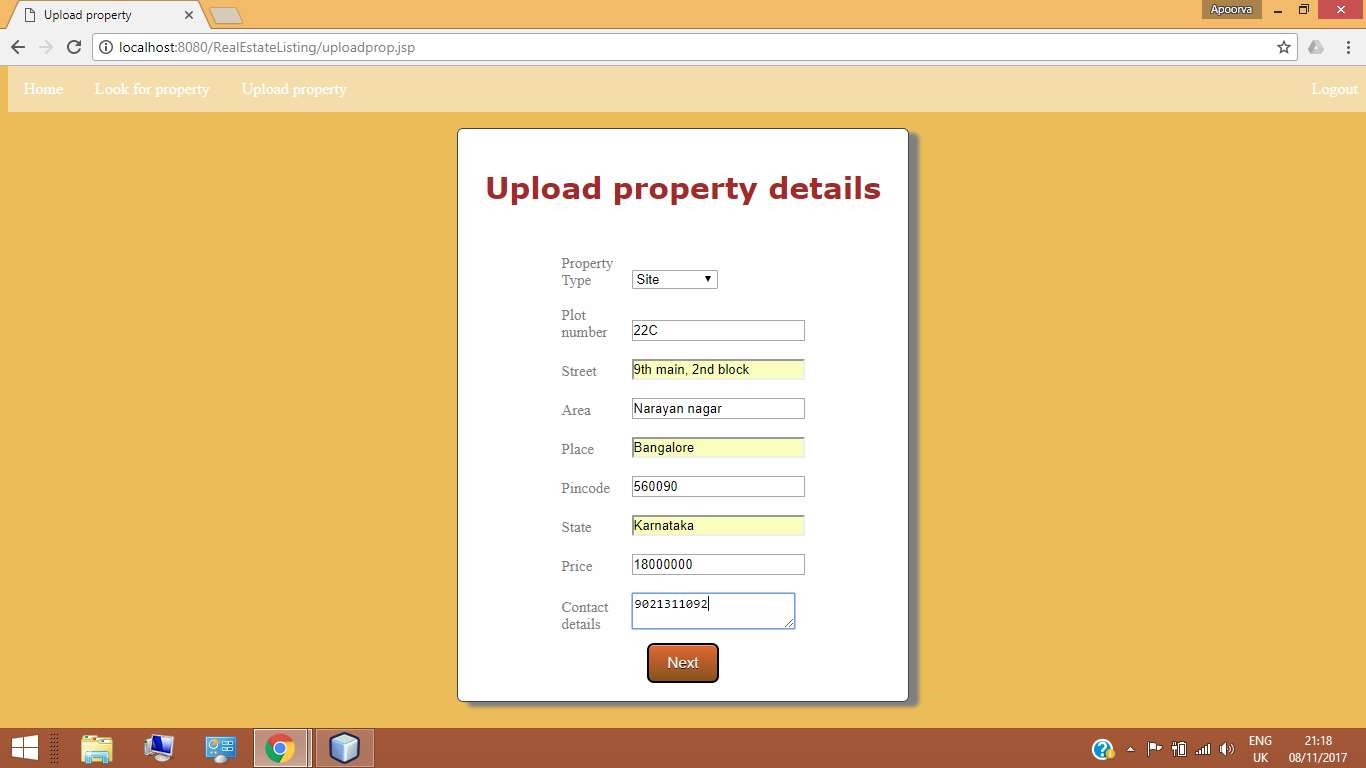


Fig 6.8 Upload property

Any user can upload details of property they want to advertise from their account. The logged in user will be automatically listed as the owner of the property. This allows the user to choose the type of property they want to upload and the location and price with contact details. Each type of property chosen has a corresponding form to enter

specifications about the property suitably.

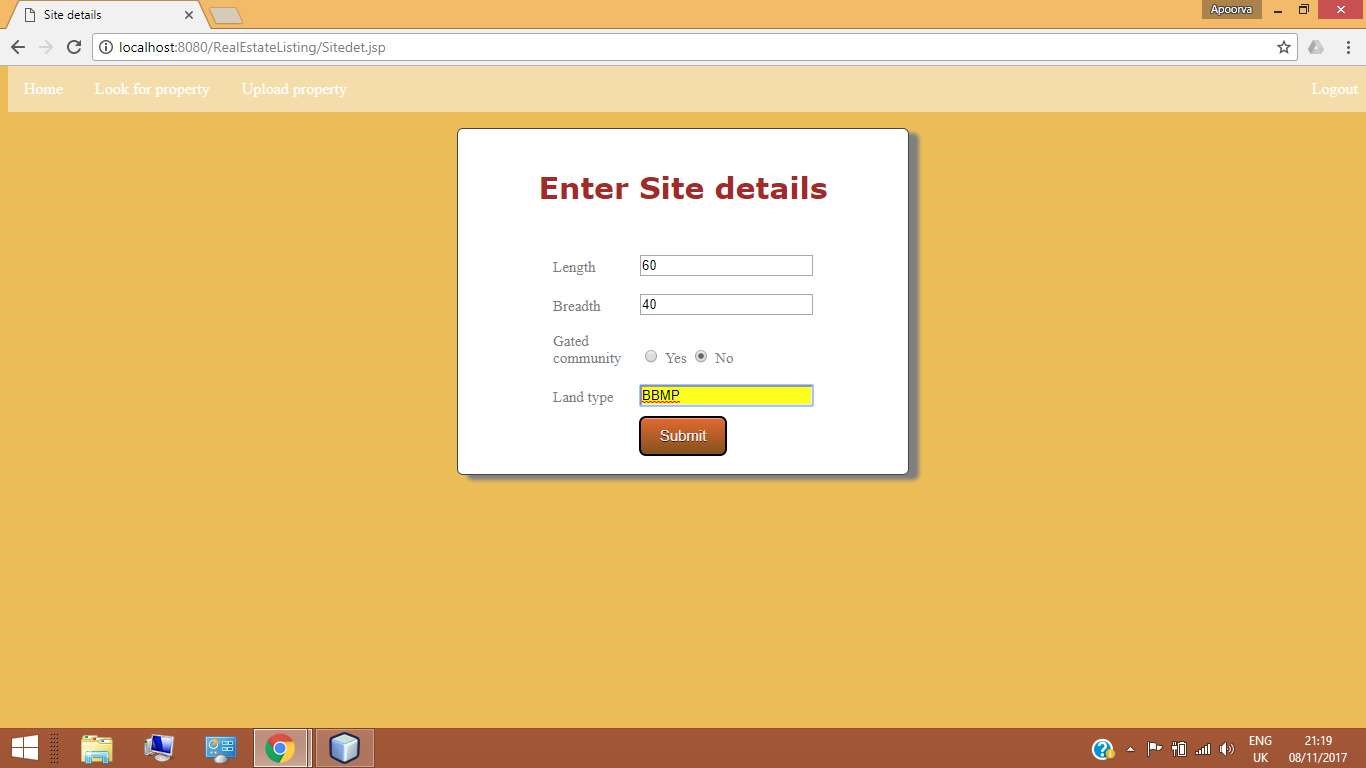


Fig 6.9 Upload site

When a user wants to put up a site for sale, they have to provide the dimensions and type

of land.

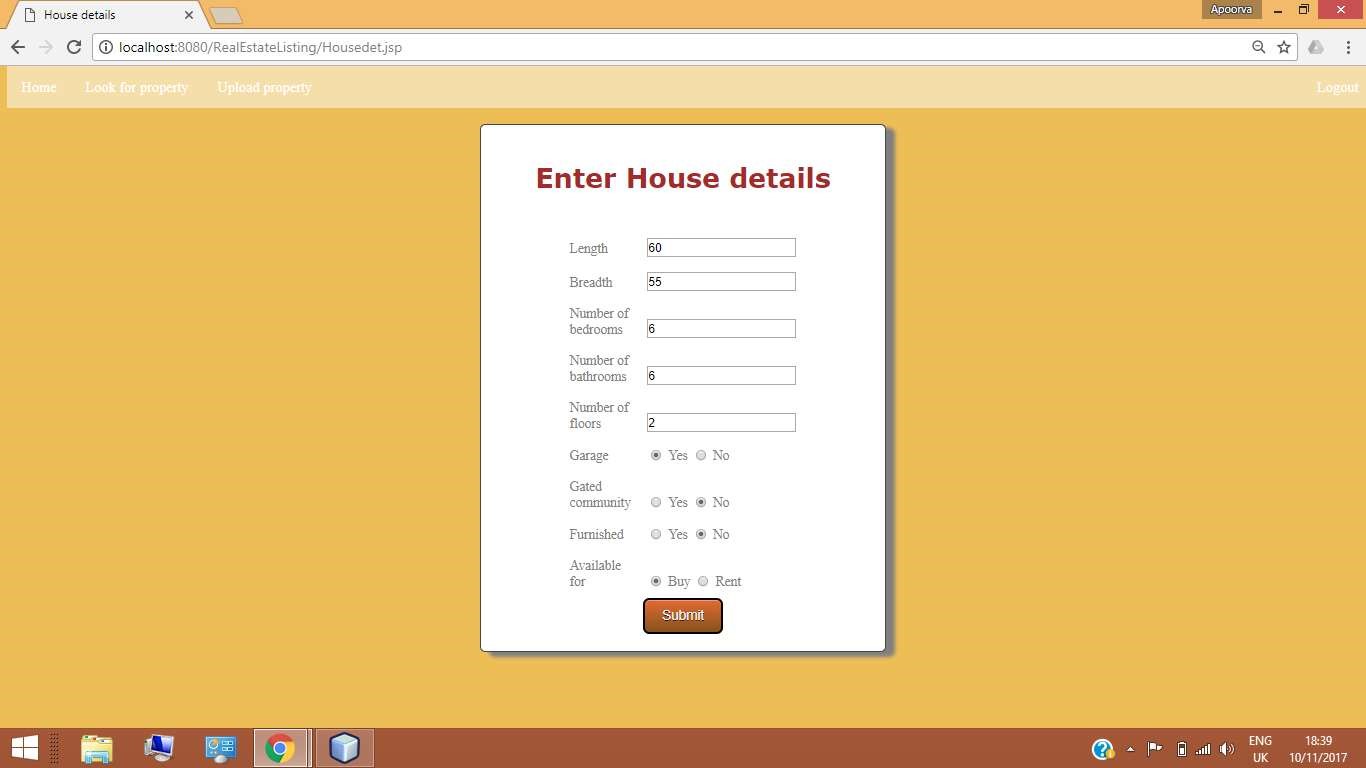


Fig 6.10 Upload house

When a user wants to put up a house for sale or rent, they have to provide the dimensions and the details such as number of rooms, bathrooms and floors, availability of garage, etc.

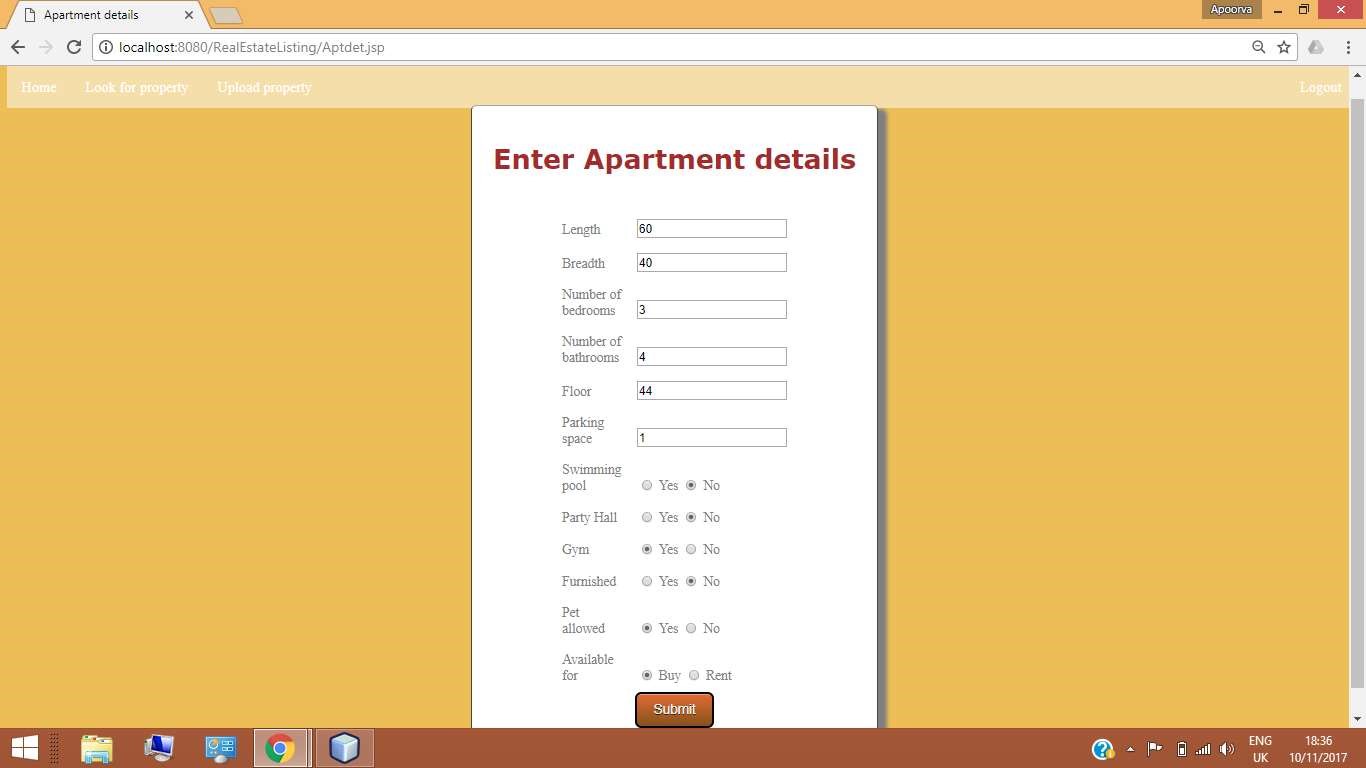


Fig 6.11 Upload Apartment

When a user wants to put up an apartment for sale or rent, they have to provide the dimensions and the details such as number of rooms, bathrooms, floor, availability of

parking space and other amenities such as swimming pool, gym, etc.

## 6.5 View uploaded property

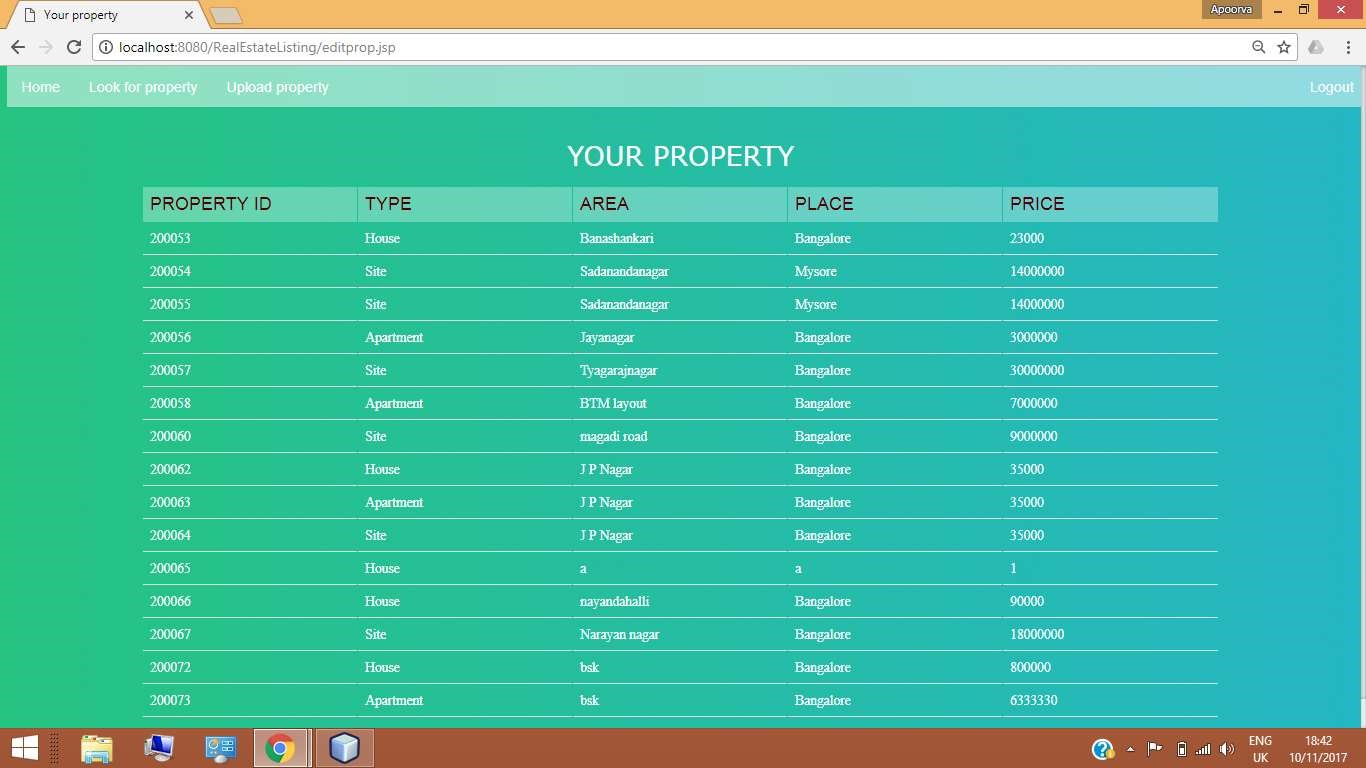


Fig 6.12 Uploaded property

This section allows users to view details of property they have uploaded. This is to keep track of the user's property. It contains the property ID, type of property, location and

price details.

## 6.6 Logout

When the user logs out, their session id is cleared and they are redirected to the login

page. A new user can log in after this.

## 6.7 Navigation

The user can navigate through pages from the homepage.

An additional navigation functionality is provided by a HTML 5 component called

navbar. This is present at the top of every webpage to allow easy navigation Sample HTML code:

<div class="topnav" id="myTopnav">

<ul>

<li> <a href="Homepage.jsp">Home</a></li>

<li><a href="searchpage.jsp">Look for property</a></li>

<li><a href="uploadprop.jsp">Upload property</a></li>

<li style="float:right"> <a href="login.jsp">Logout</a></li> </ul>

</div>

Sample CSS code:

ul { list-style-type: none; margin: 0; padding: 0; overflow: hidden; background-color: rgba(255,255,255,0.5);

position: fixed; top: 0;

width: 100%;

}

li { float: left;

}

li a { display: block; color: white; text-align: center;

padding: 14px 16px; text-decoration: none;

}

/\* Change the link color to #111 (black) on hover \*/ li a:hover { background-color: #111;

}

Conclusion

The Real Estate Listing System allows a user to easily upload and access details about property. It allows simplified operation and is a time saving platform. It offers a seamless

interface between property owners and seekers.

The application has been completed successfully and tested with suitable test cases. It is

user friendly and contains suitable options for users.

This is developed using Java as front end and MySQL as back end in Windows

environment. The goals achieved by this project are:

* Instant access
* User friendly environment
* Efficient management of records
* Simplification of operations

Future enhancements

Future upgrades to this project will implement:

* Filtering options – Property of a particular type can be filtered based on exact requirements, such has number of rooms, availability of amenities, location,

dimensions, etc.

* Modify uploaded property – This will allow owners to modify details of

previously uploaded property after verifying credentials over a secure database

system.

* E-commerce – The system currently allows only listing. Buying and selling

property dynamically will be implemented in the future.

References

1. Ramakrishnan, R., &amp; Gehrke, J. (2011). Database management systems.

Boston: McGraw-Hill.

1. Monson-Haefel, R. (2007). J2EE Web services. Boston, Mass.: Addison-Wesley.

Silberschatz, A., Korth, H. F., &amp; Sudarshan, S. (2011).

1. Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.
2. Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill

Education;

1. David F. (2011). JavaScript: The Definitive Guide Sixth edition. Shroff