



ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ  
VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

A MINI PROJECT REPORT ON

# **"AGRICULTURAL LAND MANAGEMENT SYSTEM"**

*Submitted to Visvesvaraya Technological University in partial fulfillment of the  
requirement for the award of degree of*

***Bachelor of Engineering***

***in***

***Computer Science and Engineering.***

**Submitted by:**

Mr. SANATH GOUTHAM

4JN19CS089

Mr. SHREYAS R ADIGA

4JN19CS091

Mr. SUHAS S A

4JN19CS108

Mr. NIRANJAN GOWDA

4JN19CS064

**Under the Guidance of:**

**Dr. Jalesh Kumar** Ph.D.

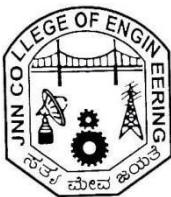
Professor, CSE

**Dr. Chetan K .R** Ph.D.

Associate Prof., CSE

**Mrs. Ganavi M** M.Tech., (Ph.D.)

Assistant Prof., CSE



Department of Computer Science & Engineering  
**Jawaharlal Nehru New College of Engineering**

Shivamogga - 577 204

January 2022

National Education Society ®



Jawaharlal Nehru New College of Engineering

Department of Computer Science & Engineering

## CERTIFICATE

*This is to certify that the Mini-project Report entitled*

# *Agricultural Land Management System*

**Submitted by:**

SANATH GOUTHAM 4JN19CS089

SHREYAS R ADIGA 4JN19CS091

SUHAS S A 4JN19CS108

NIRANJAN GOWDA 4JN19CS064

*Students of 5<sup>th</sup> semester B.E under the supervision and guidance towards the partial fulfillment of the requirement for award of degree of the Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2021 – 2022.*

**Signature of Guides and HOD:**

\_\_\_\_\_  
**Dr. Jalesh Kumar** Ph.D.

Professor, CSE

\_\_\_\_\_  
**Dr. Chetan K R** Ph.D.

Associate Prof., CSE

\_\_\_\_\_  
**Mrs. Ganavi M** M.Tech., Ph.D.

Associate Prof., CSE

\_\_\_\_\_  
**Dr. Poornima K M** Ph.D.,

Professor & Head, CSE

**Name of Examiners**

**Signature with Date**

1.

2.

## **ABSTRACT**

With growing purchasing power, investors in India are now searching for innovative ideas to earn returns from their investments. One such way is investing in agricultural land. While some investors keep such land as an asset, there is a section of investors who leverage the growing market of organic fruits and vegetables, to supplement their income. A number of experts have agreed that farm investment is a safe option for parking one's funds as the return on investment is usually higher than other investments and also lends safety to investors money. Also termed as agro-realty, the market for such investors is growing. An agricultural land plot can guarantee long-term returns, if it is in an area where the government has planned some infrastructure project in the near future. Urban investors are now looking at the returns potential of agricultural land in the suburban or peripheral areas of big cities and state capitals. There is an increasing demand for such land parcels, owing to the scarcity and high price of land in cities, with urban investors buying it to earn profit on resale or use it for cultivation. However, buying agricultural land can be tricky. The "agricultural land management system" developed in this mini-project helps the buyers to search different land, and the sellers to advertise or market their land to the buyers.

# ACKNOWLEDGEMENT

On presenting the Database Management Systems Mini - Project report on "**AGRICULTURAL LAND MANAGEMENT SYSTEM**", I feel great to express my humble feelings of thanks to all those who have helped me directly or indirectly in the successful completion of the project work.

I would like to thank our respected guides **Dr. JALESH KUMAR**, Professor, Dept. of CS & E, **Dr. CHETAN K R**, Associate Professor, Dept. of CS & E and **Mrs. Ganavi M** Assistant Professor, Dept. of CS&E, who has helped me a lot in completing this task, for their continuous encouragement and guidance throughout the project work.

I would like to thank **Dr. Poornima K M**, Professor and Head, Dept of CSE, JNNCE, Shivamogga and **Dr. K Nagendra Prasad**, The Principal, INNCE, Shivamogga for all their support and encouragement.

I am grateful to **Department of Computer Science and Engineering** and our institution **Jawaharlal Nehru New College of Engineering** and for imparting me the knowledge with which I can do my best.

Finally, I am thankful to the whole teaching and nonteaching staff of Computer Science and Engineering Dept. I also would like to thank to my parents and friends, who helped me in one way or the other throughout my project work.

SANATH GOUTHAM  
SHREYAS R ADIGA  
SUHAS S A  
NIRANJAN GOWDA

# Table of Contents

<b>Abstract.....</b>	<b>i</b>
<b>Table of contents... ..</b>	<b>iii</b>
<b>List of Tables... ..</b>	<b>v</b>
<b>List of Figures.....</b>	<b>vi</b>
<b>Chapter- 1: Introduction .....</b>	<b>1-4</b>
<b>1.1 Overview</b>	<b>1</b>
<b>1.2 Applications of agricultural land management system</b>	<b>1</b>
<b>1.3 Problem statement</b>	<b>1</b>
<b>1.4 Objectives</b>	<b>2</b>
<b>1.5 Feature of the system</b>	<b>2</b>
<b>1.6 Overview of Django</b>	<b>2</b>
<b>1.6.1 Features of Django</b>	<b>3</b>
<b>1.6.2 Characteristics of Django</b>	<b>3</b>
<b>1.7 Overview of MySQL</b>	<b>4</b>
<b>1.7.1 Features of MySQL</b>	<b>4</b>
<b>Chapter-2: Design and Implementation .....</b>	<b>5-20</b>
<b>2.1 Functional requirements</b>	<b>5</b>
<b>2.1.1 Functionalities of Agricultural land management system</b>	<b>5</b>
<b>2.1.2 ER Diagram</b>	<b>6</b>
<b>2.2 Design</b>	<b>8</b>
<b>2.2.1 Schema diagram</b>	<b>8</b>
<b>2.2.2 Front end mock-up</b>	<b>9</b>
<b>2.3 HTML</b>	<b>13</b>
<b>2.3.1 HTML tags</b>	<b>13</b>
<b>2.3.2 CSS</b>	<b>14</b>
<b>2.4 Django MVT system</b>	<b>15</b>
<b>2.5 Django admin interfaces</b>	<b>15</b>

<b>2.6 Implementation</b>	<b>17</b>
<b>2.6.1 Flowchart for login</b>	<b>18</b>
<b>2.6.2 Flowchart for registration</b>	<b>19</b>
<b>2.7 APIs used</b>	<b>20</b>
<b>Chapter-3: Results .....</b>	<b>21-35</b>
<b>3.1 Front End Results</b>	<b>21</b>
<b>3.2 Admin interface</b>	<b>27</b>
<b>3.3 Working of send email and forgot password</b>	<b>28</b>
<b>3.3.1 Email received</b>	<b>28</b>
<b>3.3.2 Working of Forgot password</b>	<b>30</b>
<b>3.4 Backend (MySQL) snapshots</b>	<b>32</b>
<b>Chapter-4: Conclusions and future scope.....</b>	<b>36</b>
<b>4.1 Future scope</b>	<b>36</b>

## **LIST OF TABLES**

Table 2.1 Entities and their attributes	<b>7</b>
Table 2.2 Relationships between entities	<b>7</b>

## LIST OF FIGURES

Fig. 2.1 ER diagram of agriculture land management system	6
Fig. 2.2 Schema Diagram of agriculture land management system	8
Fig. 2.3 Mock up of Index page	9
Fig. 2.4 Mock up of signup page	9
Fig. 2.5 Mock up buyer signup page	10
Fig. 2.6 Mock up buyer signup page	10
Fig. 2.7 Mock up broker signup page	11
Fig. 2.8 Mock up broker signup page	11
Fig. 2.9 Mock up of add new land page	12
Fig. 2.10 Mock up of land details	12
Fig. 2.11 Mock up of land details	13
Fig. 2.12 Control flow of MVT	15
Fig. 2.13 Django Admin login Screen	17
Fig. 2.14 Django admin index page	17
Fig 2.15 Flowchart for login	18
Fig. 2.16 Flowchart for registration	19
Fig. 3.1 Homepage	21
Fig. 3.2 Seller loginpage	22
Fig. 3.3 Broker loginpage	22
Fig. 3.4 Buyer loginpage	22
Fig. 3.5 Seller signupdate	23
Fig. 3.6 Broker signupdate	23
Fig. 3.7 Buyer signupdate	23
Fig. 3.8 About us page	24
Fig. 3.9 Contact us page	24
Fig. 3.10 Add land details page	25
Fig. 3.11 Lands available page	25
Fig. 3.12 Search page	26
Fig. 3.13 Brokers available page	26
Fig. 3.14 Broker page	27



Fig. 3.15 Admin interface	27
Fig. 3.16 Email received by broker for registration	28
Fig. 3.17 Email received by seller for registration	28
Fig. 3.18 Email received by buyer for registration	28
Fig. 3.19 Email received by seller for adding a land for sale	28
Fig. 3.20 Email received by buyer for buying a land	29
Fig. 3.21 Email received by buyer for booking a broker	29
Fig. 3.22 OTP received by user for changing the password	29
Fig. 3.23 Forgot password option	30
Fig. 3.24 Password reset window	30
Fig. 3.25 Notification of OTP sent	31
Fig. 3.26 Notification of password changed	31
Fig. 3.27 Notification of invalid OTP	31
Fig. 3.28 Tables created in the database	32
Fig. 3.29 Data in agri_seller table	32
Fig. 3.30 Data in agri_buyer table	33
Fig. 3.31 Data in agri_broker table	33
Fig. 3.32 Data in agri_amenities table	34
Fig. 3.33 Data in agri_broker_helps table	34
Fig. 3.34 Data in agri_amenities_containss table	35

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

“Agricultural land management system” is designed to help and assist farmers and agricultural enthusiasts in the marketing of their agricultural land and its selling. It enables the land owners to upload or advertise their land to the market on our system. Clients can view different lands hosted on our system. Brokers can post the ad on our system. Interested clients can contact the brokers or directly approach the owner of the land, and move further in registration process. The important feature of our system is, client can buy the land directly from the owner without the intervention of the middleman. This approach helps in bringing transparency in the business and in removing the middleman.

### 1.2 Applications of agricultural land management system

This system is a free, online land marketing platform. The applications are as follows:

- Sellers can upload the details of their land like address, land area, soil type and its amenities for marketing purpose.
- Seller can even upload the images of the land.
- Buyer can view different lands available for sale, they can contact the seller for buying the land.
- Buyers can even take the assistance of the agent/broker. This is optional, it is left to the buyer and the seller, whether to consult a broker or not.
- The main objective of this project is to remove the middleman in the business of selling the land.
- This system brings transparency between the seller and the buyer.

### 1.3 Problem statement

In recent days to find a land to sell or buy is a difficult process. People have to travel land distance to visit the land, examine it, speak with the seller, make some discussions and come to a point where both the seller and buyer agree on a rate for the land. This whole process involves lot of work. It is difficult to find the people who are willing to sell their land. And even to market their

land for sale and to advertise that information. As a solution for these problems agriculture land management system helps the sellers to put their land for sale and to advertise it. The buyers can find information of the lands available for sale, they can contact the seller and buy the land. This system saves lot of time of both buyer and seller, it quicken the process of selling the land.

## **1.4 Objectives**

The objective of this system is to provide a platform for those who want to sell their land and those who want to buy some land. Following are the other key objectives:

- This system can be used by seller to advertise their land, the people who are interested in buying some land can search lands by using this system according to their requirements.
- The buyer can directly contact the seller with the contact information provided and move forward with buying the land. The buyer can even take the advice from the broker.
- The broker receives some percentage as commission. But tacking advice from the broker is left to the buyer or seller as it is left optional.
- The system leads to the elimination of the middleman or agent. Elimination of the middleman brings transparency in the business between seller and the buyer.

## **1.5 Features of the System**

- Seller, buyer and broker can create their account in the system. Seller can add their address, contact details, land details in the system. Seller can add details of the land like land area, soil type, address and its amenities. The buyer can also create their account and search for the land. The broker can also create their account. Seller can add the images of their land in their account.

## **1.6 Overview of Django**

Django is a web development framework for Python which offers standard methods for fast and effective website development. The primary goal of this high-level web framework is to create complex database-driven websites. It helps you to build and maintain quality web applications. It enables you to make the development process smooth and time-saving for rapid development.

### 1.6.1 Features of Django

Features of Django are enlisted as follows:

- Helps you to define patterns for the URLs in your application.
- Simple but powerful URL system.
- Built-in authentication system.
- Object-oriented programming language database which offers best in class data storage and retrieval.
- Automatic admin interface feature allows the functionality of adding, editing and deleting items. You can customize the admin panel as per your need.
- Cache framework comes with multiple cache mechanisms.

### 1.6.2 Characteristics of Django

Here are the main characteristics of Django:

- Loosely Coupled- Django helps you to make each element of its stack independent of the others.
- Less code- Ensures effective development
- Not repeated- Everything should be developed in precisely one place instead of repeating it again
- Fast development- Django's offers fast and reliable application development.
- Consistent design- Django maintains a clean design and makes it easy to follow the best web development practices.

## 1.7 Overview of MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

- **MySQL is a database management system.** - A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.
- **MySQL databases are relational-** A relational database stores data in separate tables rather

than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

- **MySQL software is Open Source** - Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.
- **MySQL Server works in client/server or embedded systems** - The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multithreaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

### 1.7.1 Features of MySQL

#### ➤ **Open-Source**

MySQL is open-source, which means this software can be downloaded, used and modified by anyone. It is free-to-use and easy-to-understand. The source code of MySQL can be studied, and changed based on the requirements. It uses GPL, i.e. GNU General Public license which defines rules and regulations regarding what can and can't be done using the application.

#### ➤ **Quick and Reliable**

MySQL stores data efficiently in the memory ensuring that data is consistent, and not redundant. Hence, data access and manipulation using MySQL is quick.

#### ➤ **Scalable**

Scalability refers to the ability of systems to work easily with small amounts of data, large amounts of data, clusters of machines, and so on. MySQL server was developed to work with large databases.

## **CHAPTER-2:**

# **DESIGN AND IMPLEMENTATION**

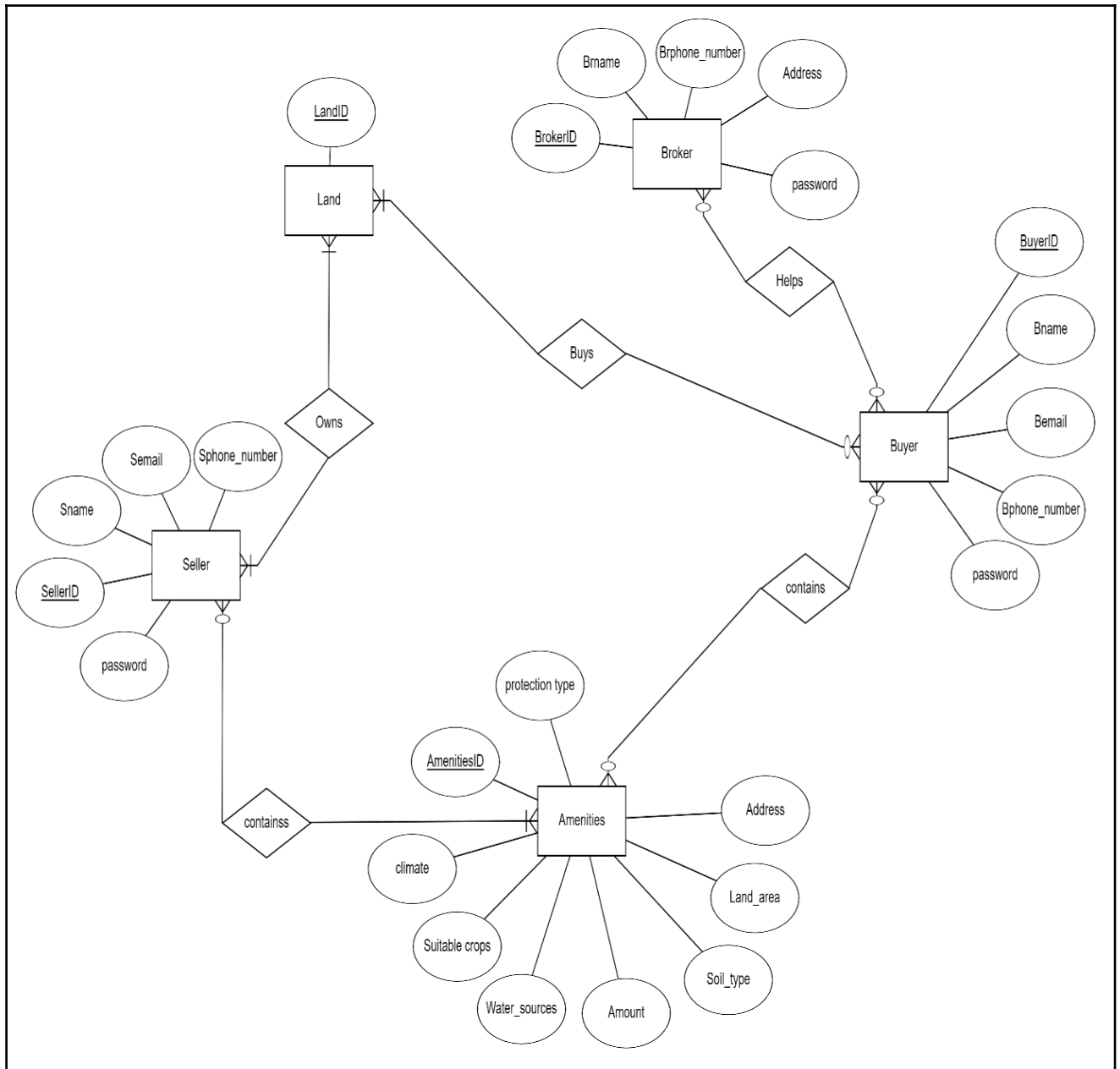
## **2.1 Functional Requirements**

### **2.1.1 Functionalities of Agricultural land management system**

Agricultural land Management system is proposed to be an automate database management. The system stores the information of the seller, buyer, broker, land details and the amenities of the land. It also provides the facility for searching the lands. This system provides data storing with graphical user interface (GUI). Some of the main services provided by the system are:

- Storing information of seller and buyer according to their id.
- This system saves time for recording details of each land details and its amenities.
- The proposed system is highly secured, because for login the system requires the username and password which is different for each user.
- It provides wide range of land options to the client who is searching for better and quicker solution.
- It maintains report of all land and its documents.
- Stores information about land and its amenities.
- This system can run on any operating system

## 2.1.2 ER Diagram



**Fig. 2.1 ER diagram of agriculture land management system**

Fig. 2.1 represents the ER Diagram of agricultural land management system. There are 5 entities along with their respective attributes and relationship between entities.

The entities and attributes in the ER Diagram are as follows:

**Table 2.1 Entities and their attributes:**

Entity name	Attributes
Seller	SellerID, Sname, Semail, password, Sphone_number.
Buyer	BuyerID, Bname, Bemail, password, Bphone_number.
Land	LandID.
Amenities	AmenitiesID, Address, Land_area, Soil_type, Amount, Watersources, Suitable_crop, climate, Protection_type.
Broker	BrokerID, Brname, Brphone_number, password, Address.

The relationship between the entities are as follows:

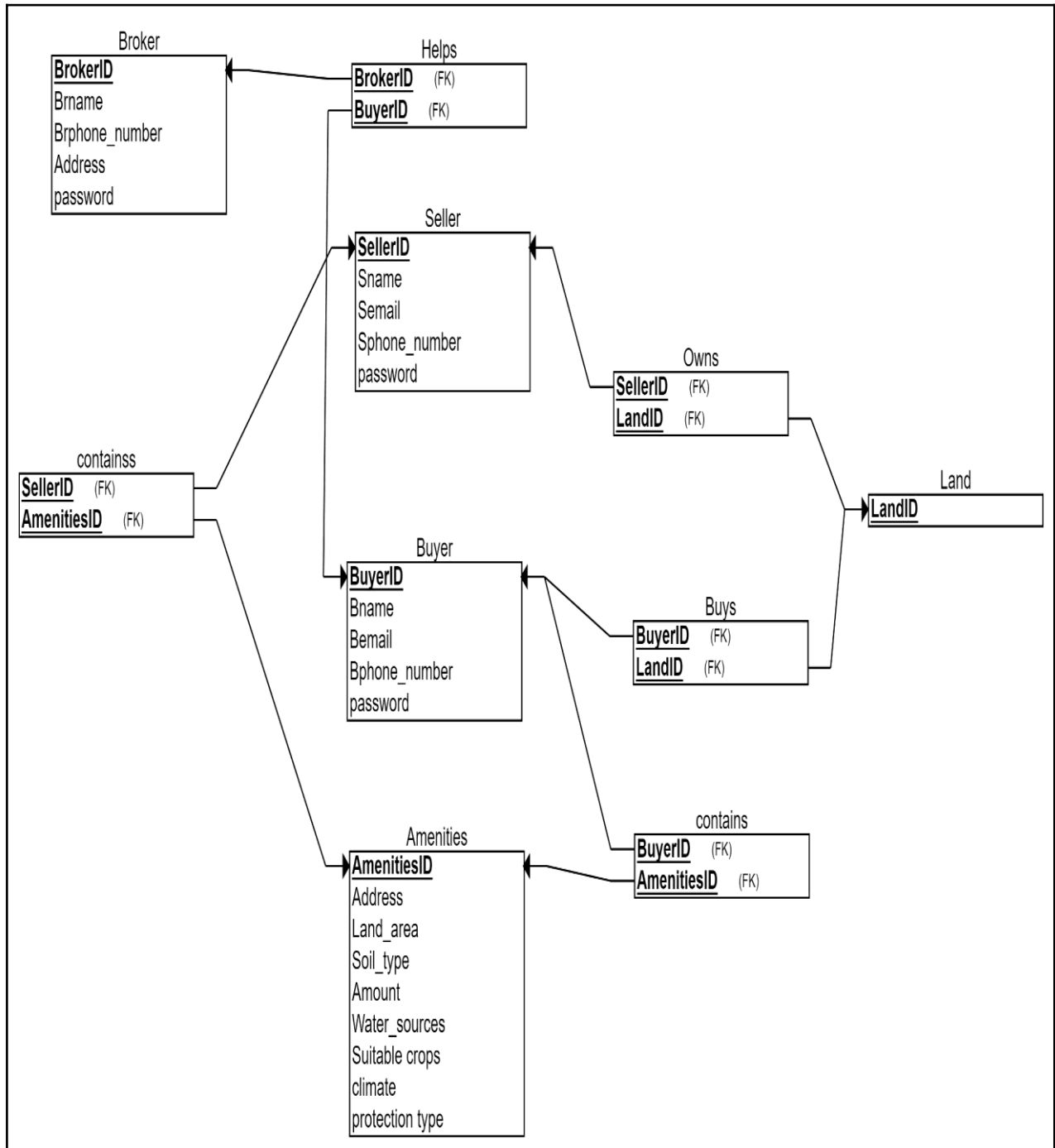
**Table 2.2 Relationships between entities:**

Entities involved	Relationship	Participation constraints	Cardinality ratio
Seller-Land	OWNS	Seller – mandatory Land – mandatory	Seller – Many Land – One
Buyer-Land	BUYS	Buyer – mandatory Land – optional	Buyer – Many Land – One
Broker-Buyer	HELPS	Broker – optional Buyer – optional	Broker – Many Buyer – Many
Seller-Amenities	CONTAINS	Seller - mandatory Amenities - optional	Seller – Many Amenities – Many
Buyer-Amenities	CONTAINS	Buyer-Optional Amenities-Optional	Buyer-Many Amenities-Many



## 2.2 Design

### 2.2.1 Schema diagram



**Fig. 2.2 Schema Diagram of agriculture land management system**

## 2.2.2 Front end mock-up



Fig. 2.3 Mock up of Index page

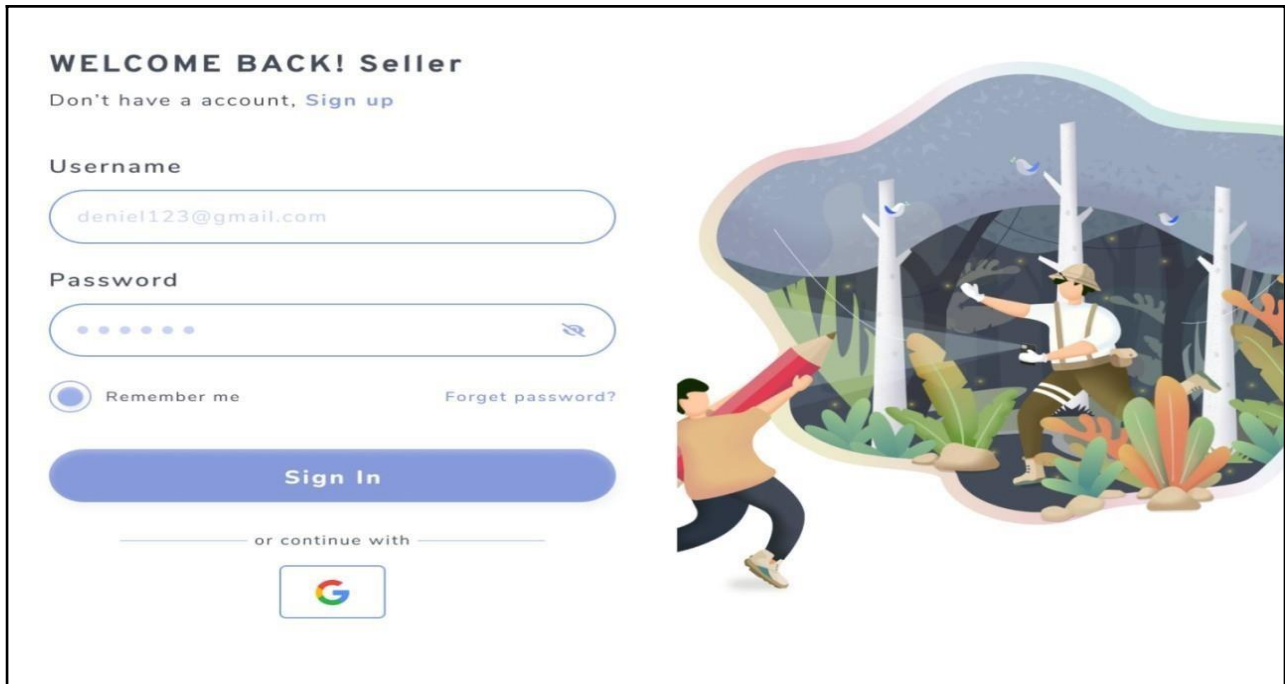
Index page consists the following: Button to go to seller, buyer and broker signup page.

The mockup for the Sign Up page includes the heading "WELCOME BACK!" followed by the text "Have an account, [Sign in](#)". Below this are four input fields: "Fullname" (containing "deniel123@gmail.com"), "Password" (with masked dots and a toggle icon), "Address" (containing "Thunga Hostel,Shimoga,India"), and "Phone Number" (containing "deniel123@gmail.com"). There is a "Remember me" checkbox and a "Forget password?" link. A large blue "Sign Up" button is positioned below the form fields. At the bottom, there is a link "or continue with" followed by a Google logo icon. The right side of the page features the same colorful illustration as the Index page, showing a person in a field with another person running towards them.

Fig. 2.4 Mock up of signup page

In the signup page user should enter the username, password, address and phone number and click

on the signup button.



**WELCOME BACK! Seller**  
Don't have a account, [Sign up](#)


**Username**

**Password**

☒ Remember me [Forget password?](#)

**Sign In**

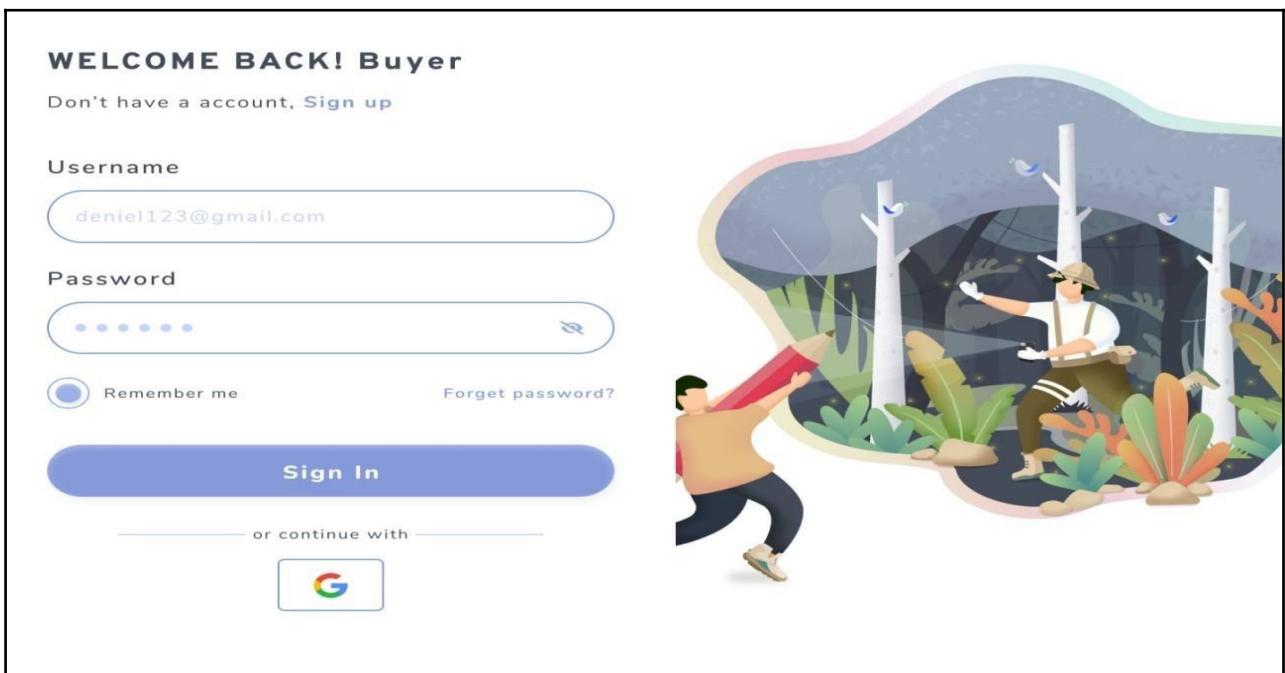
or continue with



The mockup features a clean, modern design with a white background. On the left, the login form is organized vertically, starting with a heading, a link for new users, input fields for username and password, a 'Remember me' checkbox, a 'Sign In' button, and a 'Continue with Google' option. On the right, a vibrant illustration depicts a person in a hat and boots standing in a forest, holding a large red pencil, symbolizing the design process.

**Fig. 2.5 Mock up buyer signup page**

In the seller signup page seller should enter the username, password, address and phone number and click on the signup button.



**WELCOME BACK! Buyer**  
Don't have a account, [Sign up](#)


**Username**

**Password**

☒ Remember me [Forget password?](#)

**Sign In**

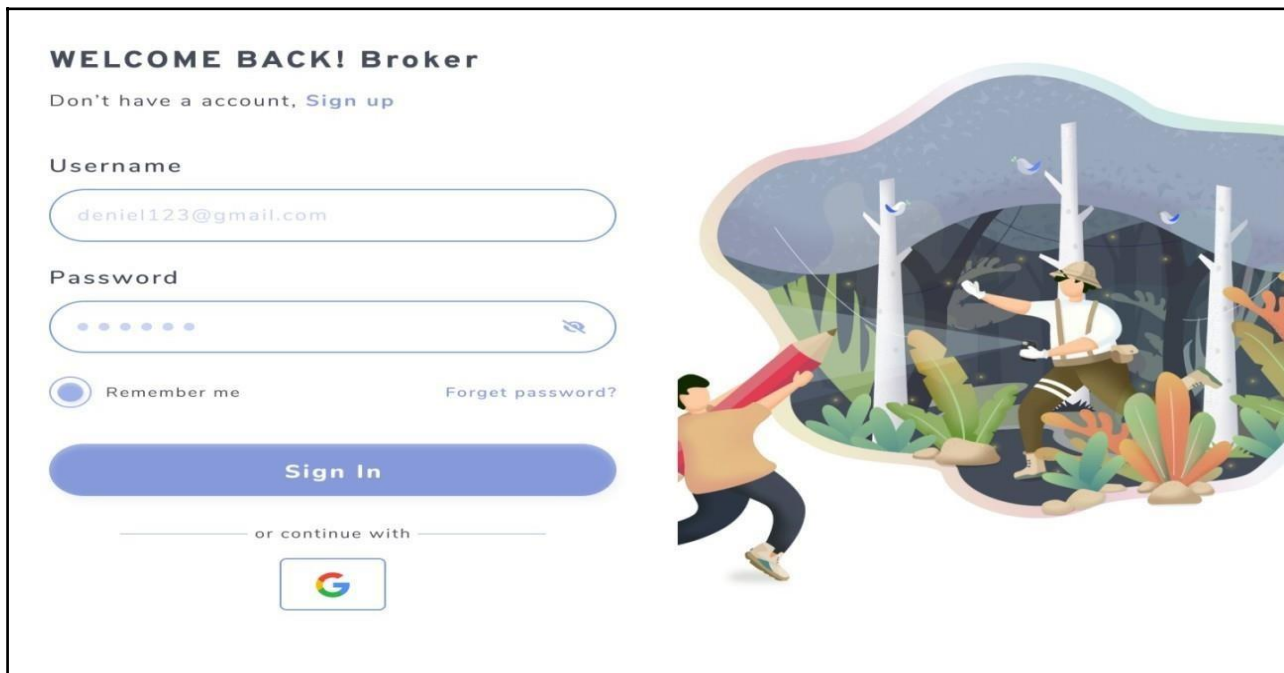
or continue with



This mockup is identical in layout and content to the one in Fig. 2.5, but it is tailored for a buyer. It includes the same login form elements and the same illustrative graphic of a person in a forest with a large red pencil.

**Fig. 2.6 Mock up buyer signup page**

In the buyer signup page buyer should enter the username, password, address and phone number and click on the signup button.



**WELCOME BACK! Broker**  
Don't have a account, [Sign up](#)


**Username**

**Password**

☒ Remember me [Forget password?](#)

**Sign In**

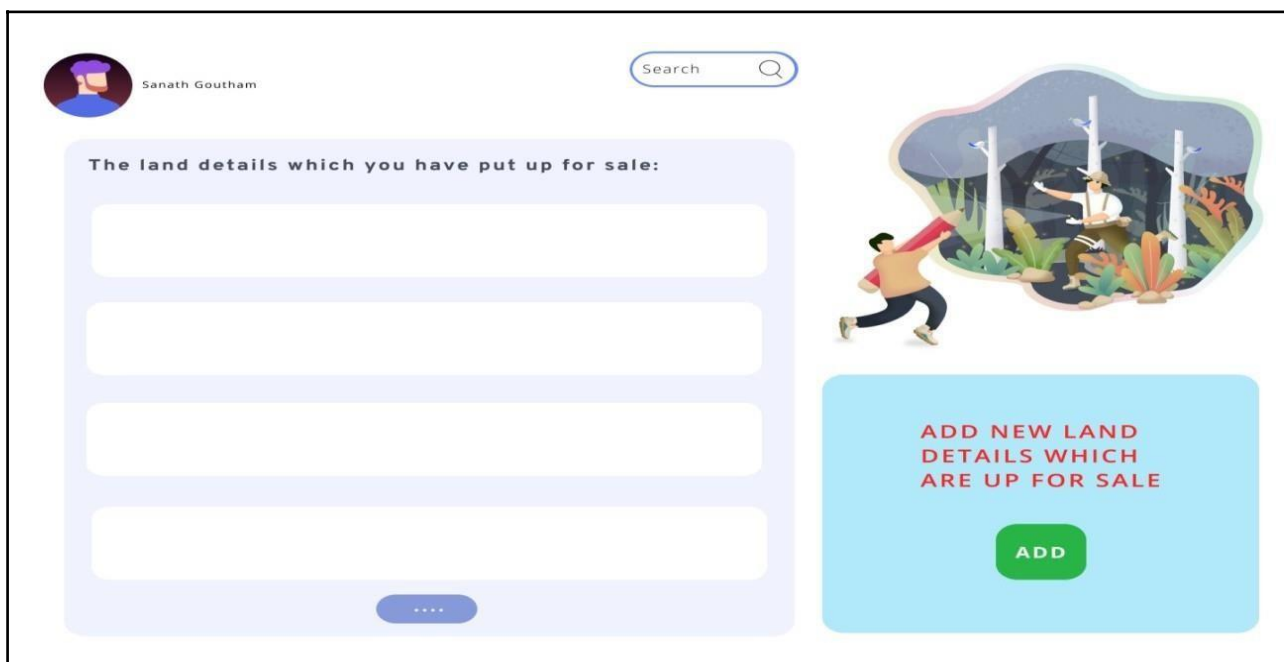
or continue with




The mockup shows a login interface for a broker. It includes a welcome message, a link to sign up for new accounts, fields for username and password, a 'Remember me' checkbox, a 'Forgot password?' link, a 'Sign In' button, and an option to continue with Google. On the right, there is a colorful illustration of a person in a hat and boots running through a forest, carrying a large red pencil.

**Fig. 2.7 Mock up broker signup page**

In the broker signup page broker should enter the username, password, address and phone number and click on the signup button.



 Sanath Goutham

**The land details which you have put up for sale:**

**ADD NEW LAND DETAILS WHICH ARE UP FOR SALE**

**ADD**

The mockup displays a user profile for 'Sanath Goutham' with a search bar. Below this is a section titled 'The land details which you have put up for sale:' containing four empty text input fields. To the right, there is a blue box with the text 'ADD NEW LAND DETAILS WHICH ARE UP FOR SALE' and a green 'ADD' button. The same forest illustration from the previous mockup is present on the right side.

**Fig. 2.8 Mock up broker signup page**

In the broker signup page broker should enter the username, password, address and phone number and click on the signup button.

Sanath Goutham

### ADD NEW LAND DETAILS:

Address:

Land Area:

Soil Type:

Water Sources:

Suitable Crop:

Weather:

Protection\_Type:

Distance\_From\_Mainroad:

**SAVE** **CLEAR**

**Fig. 2.9 Mock up of add new land page**

In the add new land page seller can add land details like address, land area, soil type, water sources, suitable crop, weather, protection type, distance from main road and click on the save button.

Sanath Goutham

Search

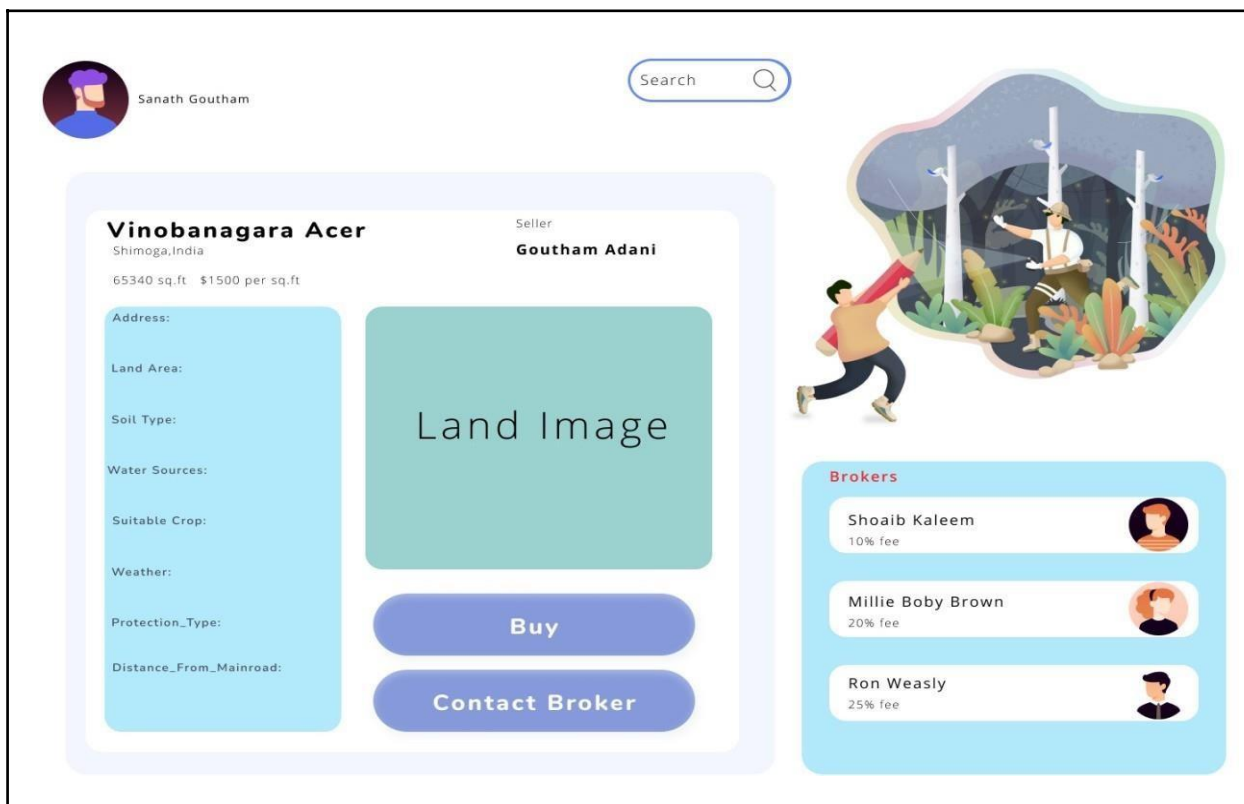
Land Name	Location	Area	Price	Seller
Vinobanagara Acer	Shimoga, India	65340 sq.ft	\$1500 per sq.ft	Goutham Adani
Gadikoppa Acer	Jnnce, India	75340 sq.ft	\$2000 per sq.ft	Suhas Adani
Tank Mohala Acer	Kachinkatte, India	85340 sq.ft	\$15000 per sq.ft	Samrudh Adani
Durgigudi Acer	Washington, USA	65340 sq.ft	\$20000 per sq.ft	Goutham Adani

**Brokers**

Broker Name	Fee
Shoaib Kaleem	10% fee
Millie Boby Brown	20% fee
Ron Weasly	25% fee

**Fig. 2.10 Mock up of land details**

In this page buyer can see the available lands and also the available brokers.



**Fig. 2.11 Mock up of land details**

In this page buyer can see the available land details and also the available brokers.

## 2.3 HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

### 2.3.1 HTML tags

HTML contains lots of predefined tag. Most commonly used tags in HTML are:

**<!DOCTYPE>:** It defines the document type or it instruct the browser about the version of HTML.

**<html> :** This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <!DOCTYPE>.



Syntax: <html> Statements... </html>

**<head>:** It should be the first element inside the <html> element, which contains the metadata (information about the document). It must be closed before the body tag opens.

Syntax: <head> Statements... </head>

**<title>:** As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately.

Syntax: <title> Statements... </title>

**<body> :** Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

Syntax: <body> Statements... </body>

Heading tag: It is used to define the heading of html document.

Syntax: <h1> Statements... </h1>

          <h2> Statements... </h2>

          <h3> Statements... </h3>

## 2.3.2 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice, and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is

accessed on a mobile device.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

## 2.4 Django MVT system

MVT architecture is the software design pattern used by the Django web framework. Although Django at its core is based on MVC architecture, it actually is implementing a variation of MVC, called MTV architecture. There are some drawbacks of MVC architecture and it has certain areas where Django is offering a better-quality feature when using the MTV architecture. In this architecture, the model remains the same, it provides the interface for storing the data in the database. Just like View in the MVC model, Django replaces it with a Template in its framework, the controller part in the MVC model is taken care of by View in the MVT architecture.

MVT stands for **Model – View – Template**.

- **Model:** Just like the Model in MVC, here as well it has the same functionality of providing the interface for the data stored in the database. The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database generally relational databases such as MySQL, Postgres.
- **Template:** Just like view in MVC, Django uses templates in its framework. Templates are responsible for the entire User Interface completely. It handles all the static parts of the webpage along with the HTML, which the users visiting the webpage will perceive. A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.
- **Views:** In Django, Views act as a link between the Model data and the Templates. The View is the user interface is what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript

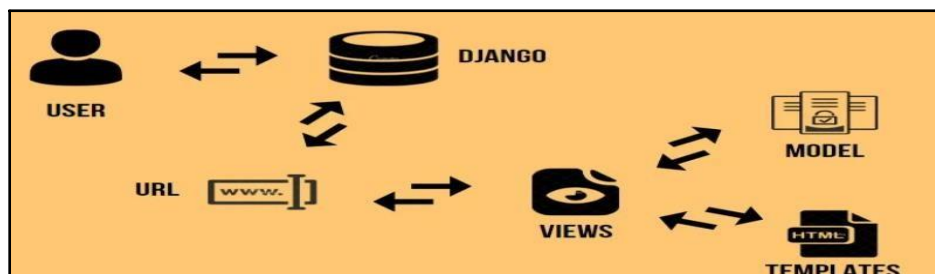


Fig. 2.12 Control flow of MVT

As shown in the Fig. 2.12:



1. The user sends a URL request for a resource to Django.
2. Django framework then searches for the URL resource.
3. If the URL path links up to a View, then that particular View is called.
4. The View will then interact with the Model and retrieve the appropriate data from the database.
5. The View then renders back an appropriate template along with the retrieved data to the user.

## 2.5 Django Admin Interfaces

Django provides a ready-to-use user interface for administrative activities. We all know how an admin interface is important for a web project. Django automatically generates admin UI based on your project models. That interface will let you administrate Django groups and users, and all registered models in your app. Django provides a default admin interface which can be used to perform create, read, update and delete operations on the model directly. It reads set of data that explain and gives information about data from the model, to provide an instant interface where the user can adjust contents of the application. This is an in-built module and design to execute admin related work to the user.

When you run startproject, Django created and configured the default admin site for you. All that you need to do now is create an admin user (superuser) to log into the admin site. To create an admin user, run the following command from inside your virtual environment:

```
python manage.py createsuperuser
```

Enter your desired username and press enter.

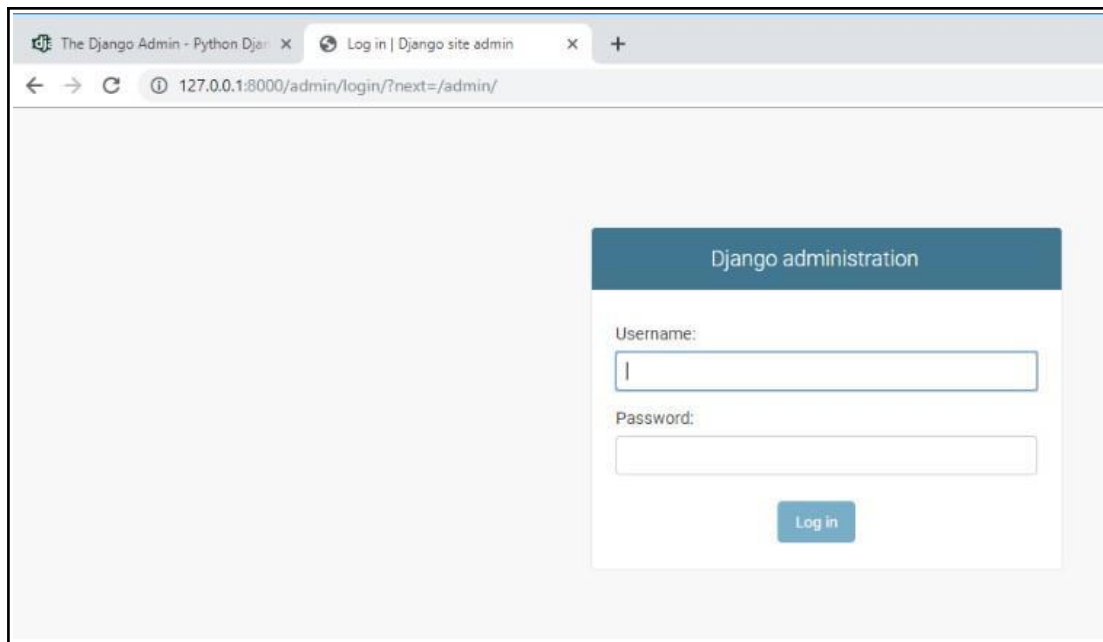
```
Username: admin
```

You will then be prompted for your email address:

```
Email address: admin@example.com
```

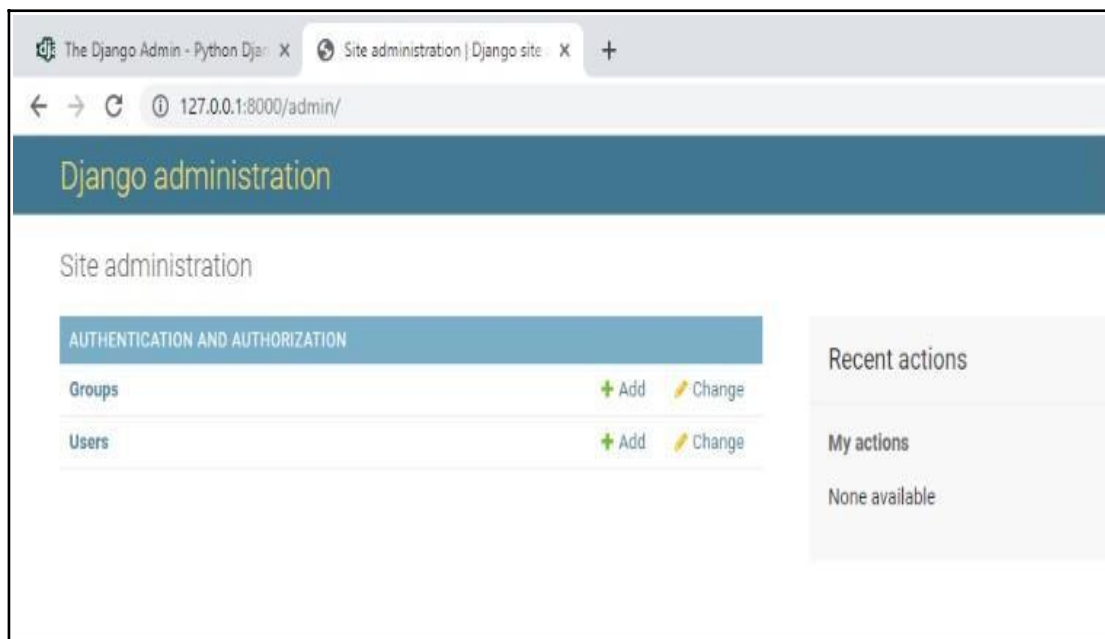
The final step is to enter your password. You will be asked to enter your password twice, the second time as a confirmation of the first.

First, make sure the development server is running, then open a web browser to <http://127.0.0.1:8000/admin/>. You should see the admins login screen.



**Fig. 2.13 Django Admin login Screen**

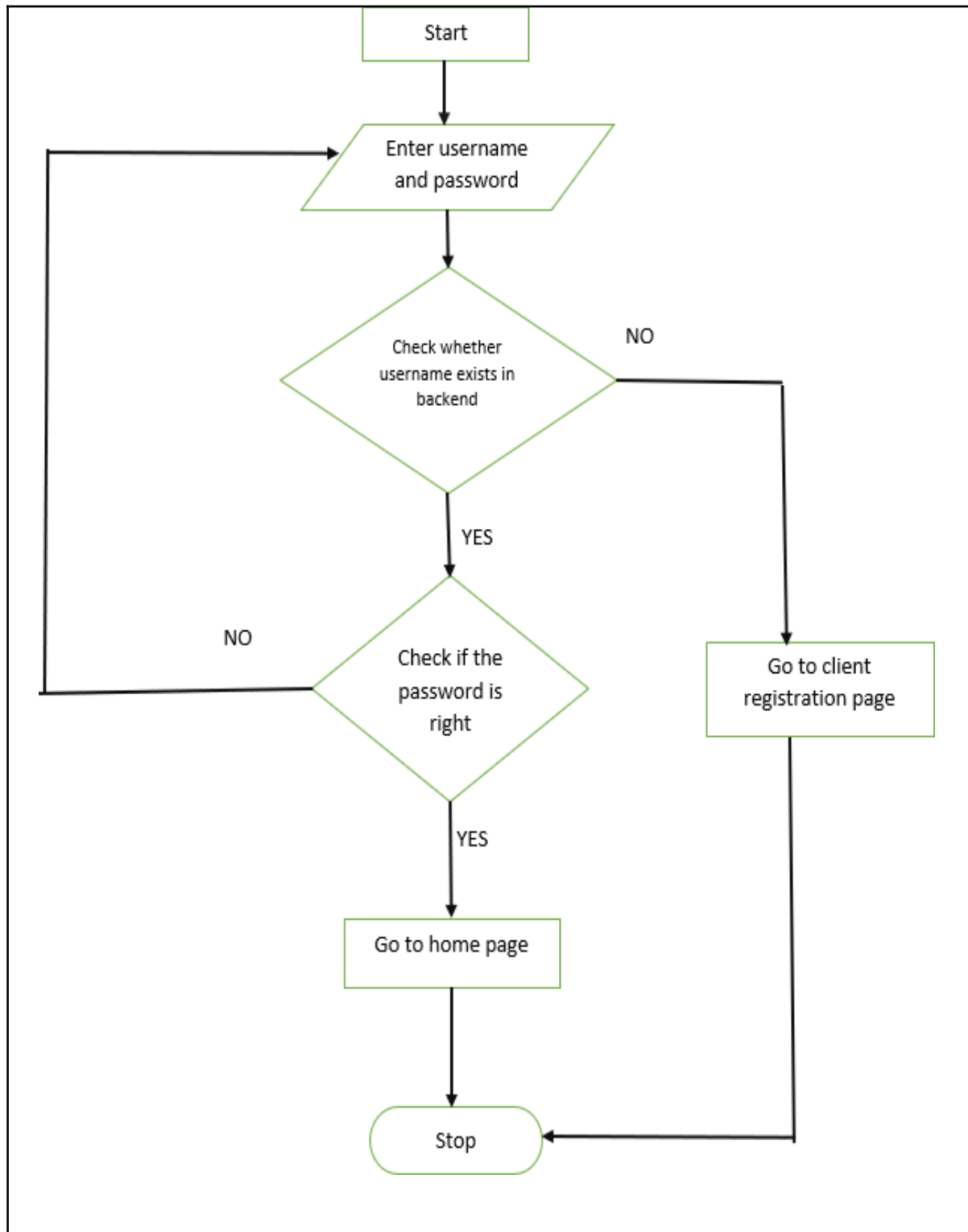
After logging in, the following screen is obtained:



**Fig. 2.14 Django admin index page**

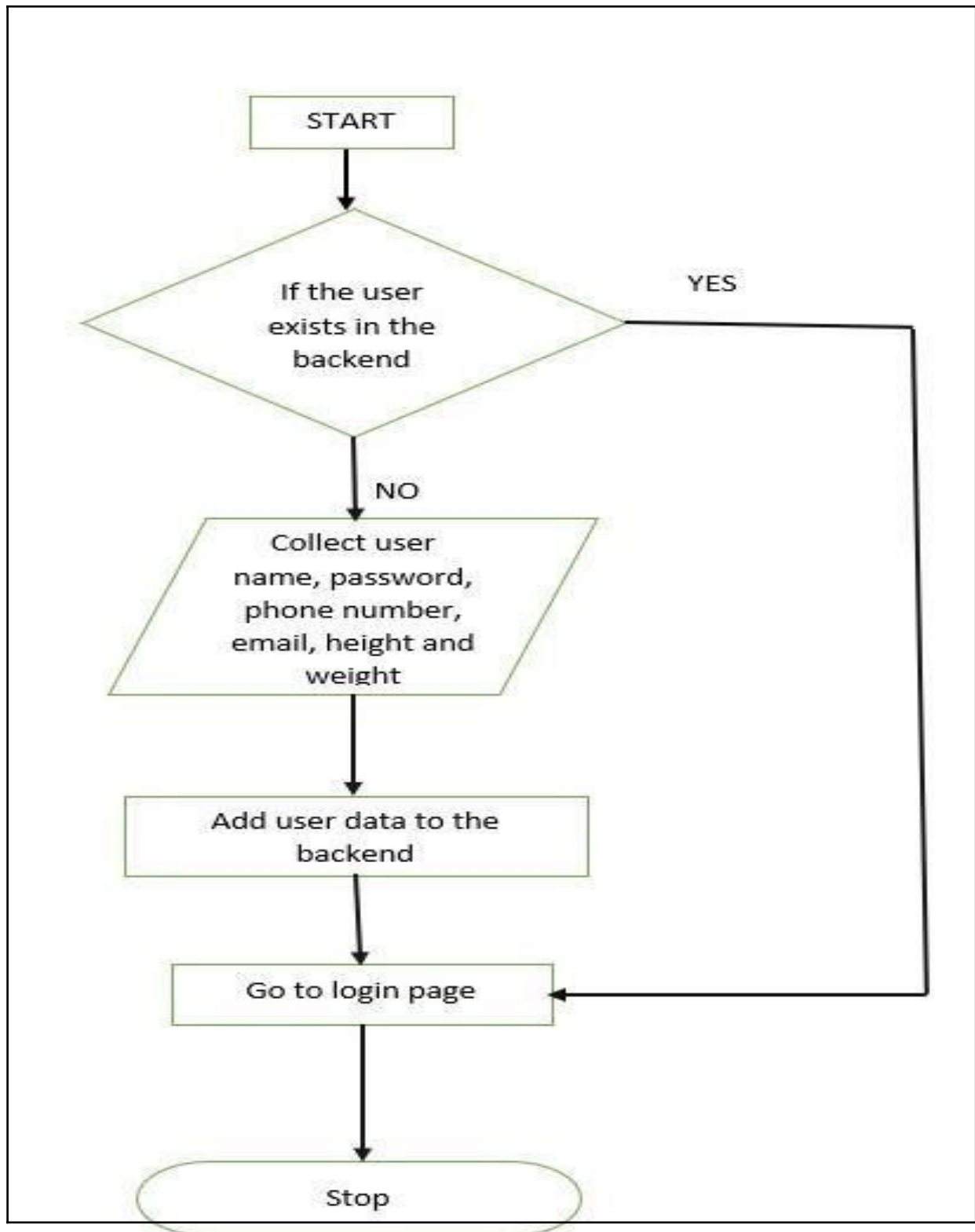
## 2.6 Implementation

### 2.6.1 Flowchart for login



**Fig 2.15 Flowchart for login**

### 2.6.2 Flowchart for registration



**Fig. 2.16** Flowchart for registration

## 2.7 APIs used

1. Request.GET.get: Request.GET contains the **GET variables**. These are what you see in your browser's address bar. The get() method is a method used for dictionaries.
2. objects.filter: Objects.filter is to get a matching result from the database, return a list of objects.
3. objects.get: It is a method of accessing objects in python. It returns the value associated with that object.
4. response.set\_cookie: The set\_cookie() method in DjangoHttpResponse has a name: Name of the cookie.value: Value you want to store int or string but it will return string.
5. objects.all: Return a copy of the current QuerySet.
6. request.COOKIES.get: Using request.COOKIES.get() in Django also provides a method to get the desired value from the cookie. You can directly access that value using get method over request object.

## Chapter-3:

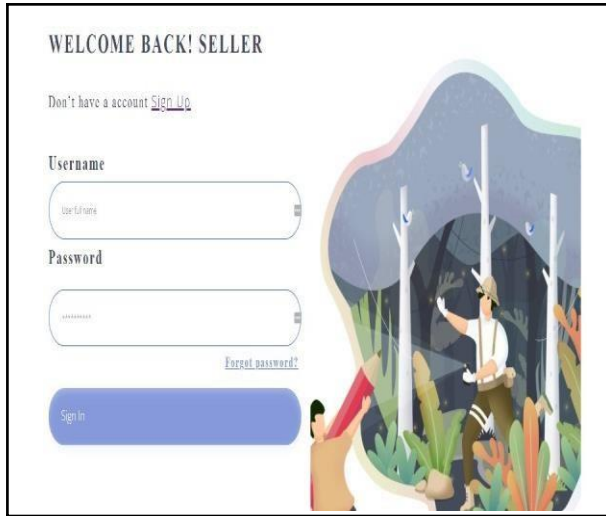
## Results

### 3.1 Front End Results



**Fig. 3.1 Homepage**

Figure 3.1 shows the home page. User can select seller, broker or broker option to create new account, if the user is already registered the username and password is stored in the database and if you click on the login button it will take you to the login page. If the user is not registered or if you have entered a wrong password, it will refresh and remain in the same page. User can send feedback by clicking the contact us button and see some details about the project by clicking about us.



WELCOME BACK! SELLER

Don't have a account, [Sign Up](#)

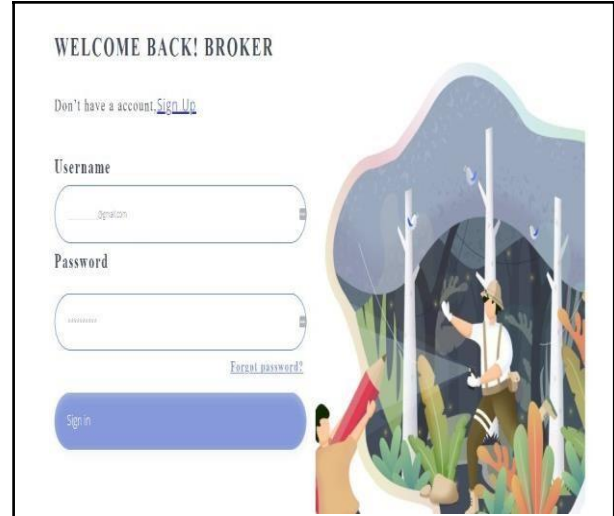
Username

Password

[Forgot password?](#)

Sign in

**Fig. 3.2 Seller loginpage**



WELCOME BACK! BROKER

Don't have a account, [Sign Up](#)

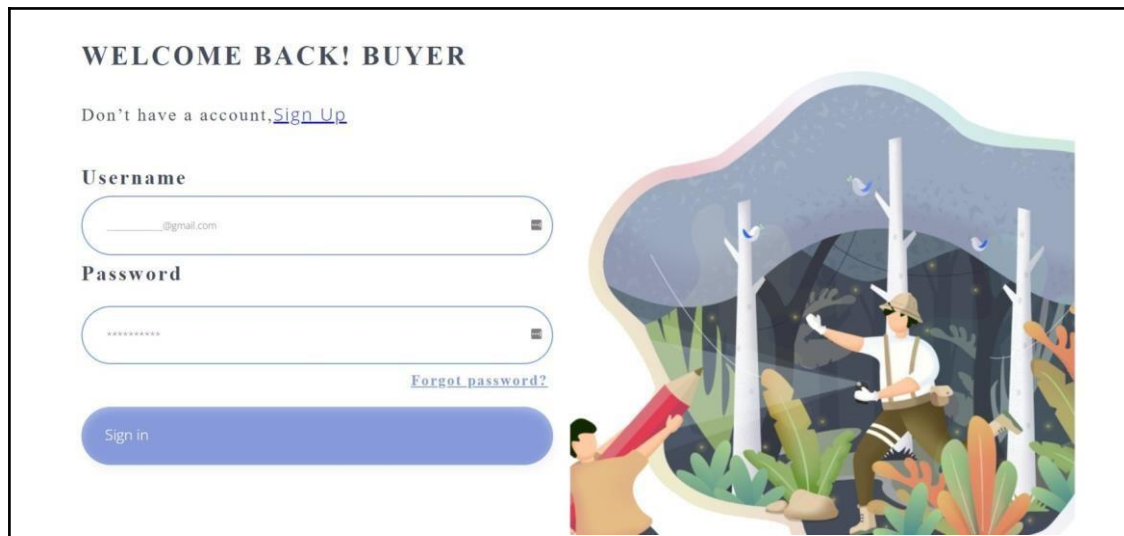
Username

Password

[Forgot password?](#)

Sign in

**Fig. 3.3 Broker loginpage**



WELCOME BACK! BUYER

Don't have a account, [Sign Up](#)

Username

Password

[Forgot password?](#)

Sign in

**Fig. 3.4 Buyer loginpage**

Figures 3.2, 3.3 and 3.4 shows the seller loginpage, broker loginpage and buyer loginpage respectively. User can login to their account by entering username and password and clicking signin button. If the user don't have any account they can create one by pressing signup option. After clicking signup option respective buyer, broker and seller signup page opens and user can create their new account by entering the required details.





**SELLER**  
Have an account, [Sign In](#)

**Full Name**

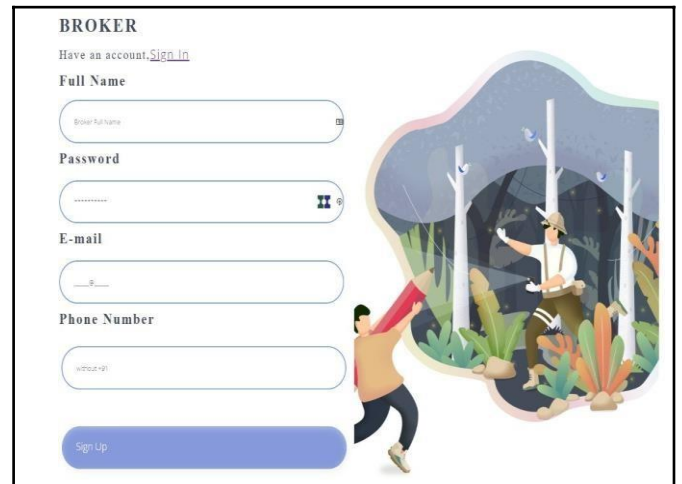
**Password**

**E-mail**

**Phone Number**

[Sign Up](#)

**Fig. 3.5 Seller signuppge**



**BROKER**  
Have an account, [Sign In](#)

**Full Name**

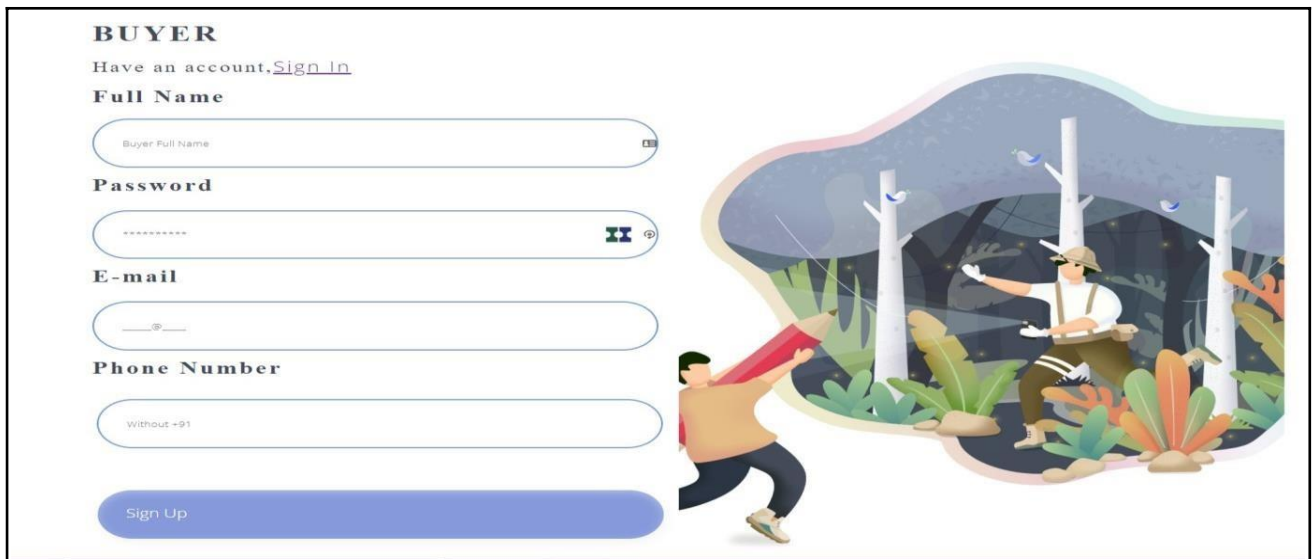
**Password**

**E-mail**

**Phone Number**

[Sign Up](#)

**Fig. 3.6 Broker signuppge**



**BUYER**  
Have an account, [Sign In](#)

**Full Name**

**Password**

**E-mail**

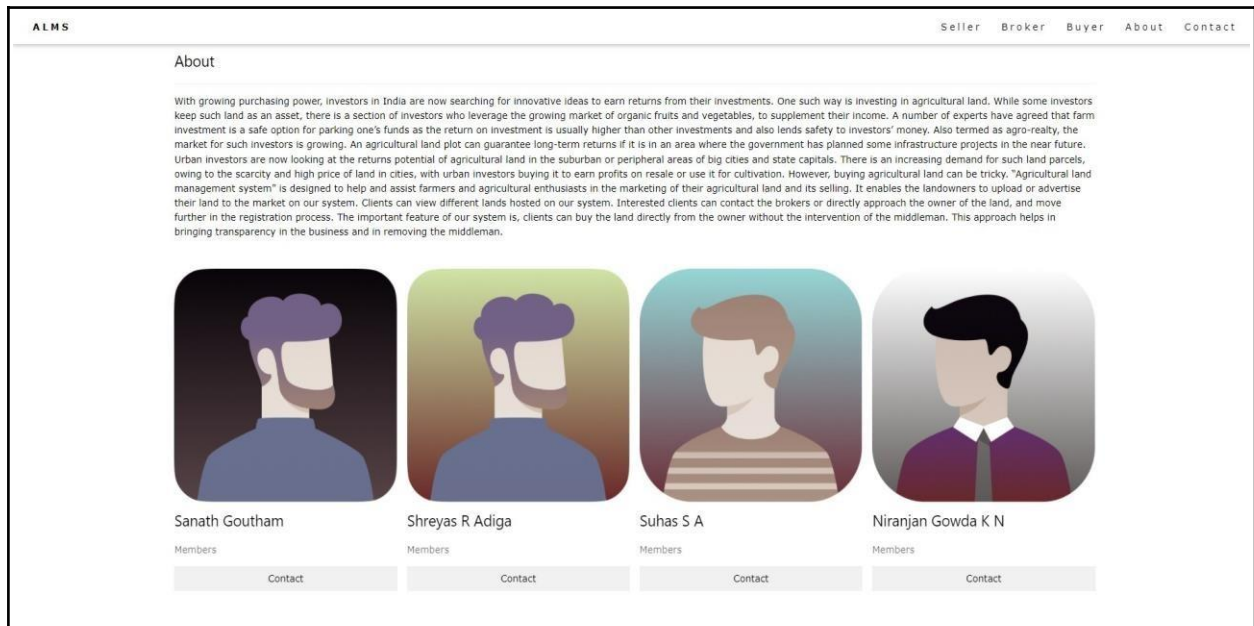
**Phone Number**

[Sign Up](#)

**Fig. 3.7 Buyer signuppge**

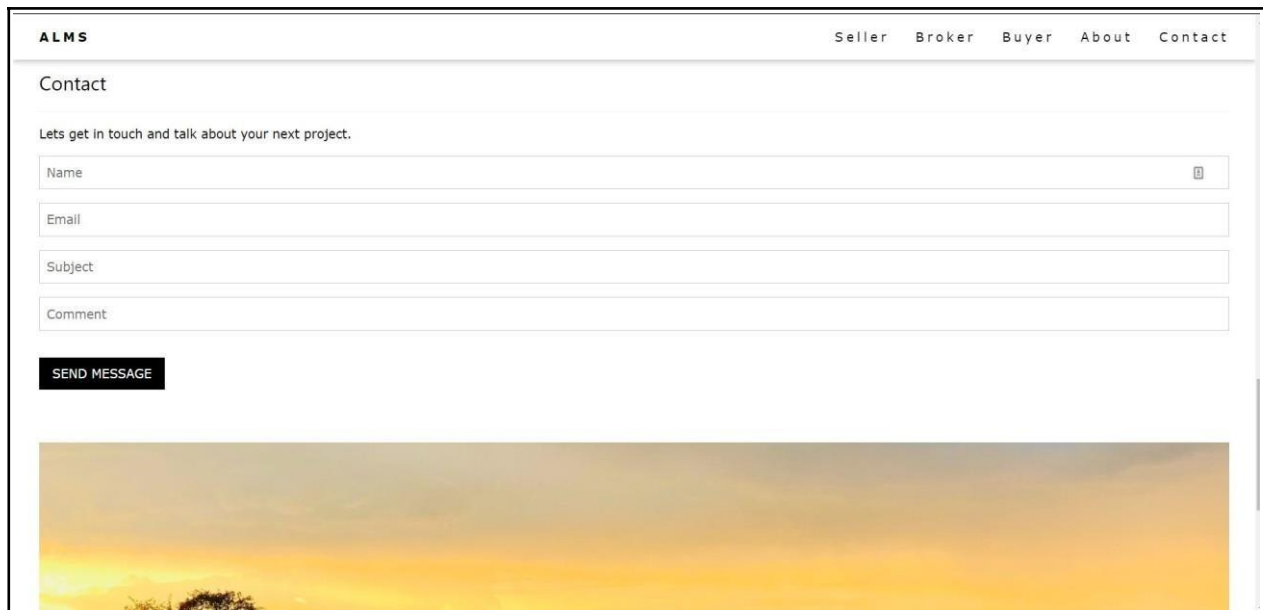
Figures 3.5, 3.6 and 3.7 shows the seller signuppge, broker signuppge and buyer signuppge respectively. If user don't have any account they can create one by pressing signup option. After clicking signup option respective buyer, broker and seller signup page opens and user can create their new account by entering the required details. User is asked to enter their name, set a password, enter their mail id and phone number.





**Fig. 3.8 About us page**

Figure 3.8 shows the about us page which gives some information about the project and its developers.



**Fig. 3.9 Contact us page**

Figure 3.9 shows the contact us page where users can give their feedback to our team.

## ADD NEW LAND DETAILS:

Address:

town,pincode

Land Area:

Sq.ft

Soil Type:

Ex: Sandy,Clay

Water Sources:

Ex: River,pond,Tubewell

Suitable Crop:

Ex: Paddy,Wheat,Sugarcane

Climatic\_Zone:

Ex: Dry,Humid,Cold,Cloudy,Mixed


Protection Type:

Price of land:

₹

SAVE

CLEAR



**Fig. 3.10 Add land details page**

In the add land details page seller can add land details. Seller should add address of land, land area, soil type, water resources, climate, suitable crop, protection type and price of the land. Clicking the save button will add the land details to the database and the clear button will clear the entered details in the boxes to put the details again.

## LANDS AVAILABLE:

SPECIFIC ADDRESS SEARCH

BROKERS BOOKING

shimoga

Area: 350.0

Soil: clay

water\_source: pond

suitable\_crops: Bean

weather: humid

protection\_type:

Amount: 500

BUY Now

Ablegere

Area: 250.0

Soil: sandy

water\_source: river

suitable\_crops: sugarcane

weather: humid

protection\_type:

Amount: 600

BUY Now

sagara

Area: 300.0

Soil: clay

water\_source: pond

suitable\_crops: wheat

weather: dry

protection\_type:

Amount: 450

BUY Now

**Fig. 3.11 Lands available page**

Fig. 3.11 shows the page where the lands added by the seller will be displayed to the buyer. By clicking the specific address search the buyer can search the lands that are in the vicinity of the required area of the buyer. The buyer can book broker by clicking the broker booking button.

Address	Soil_type	water_sources	Land_area	suitable_crop	weather	protection_type	Amount
shimoga	clay	pond	350.0	Bean	humid		500
shimoga	sandy	river	350.0	maize	dry		450

**Fig. 3.12 Search page**

Figure 3.12 displays the lands which have the land area matching to the word entered by the buyer in the search box.

ID: 2

Name: sharma

Phone: 998855661122

Email: sharma@gmail.com

Book Now

ID: 3

Name: prakash

Phone: 9896543217

Email: prakash

Book Now

ID: 4

Name: babu rao

Phone: 9845216985

Email: baburao@gmail.com

Book Now

ID: 5

Name: Raju

Phone: 9978664533125

Email: raju@gmail.com

Book Now

ID: 6

Name: suhas s a

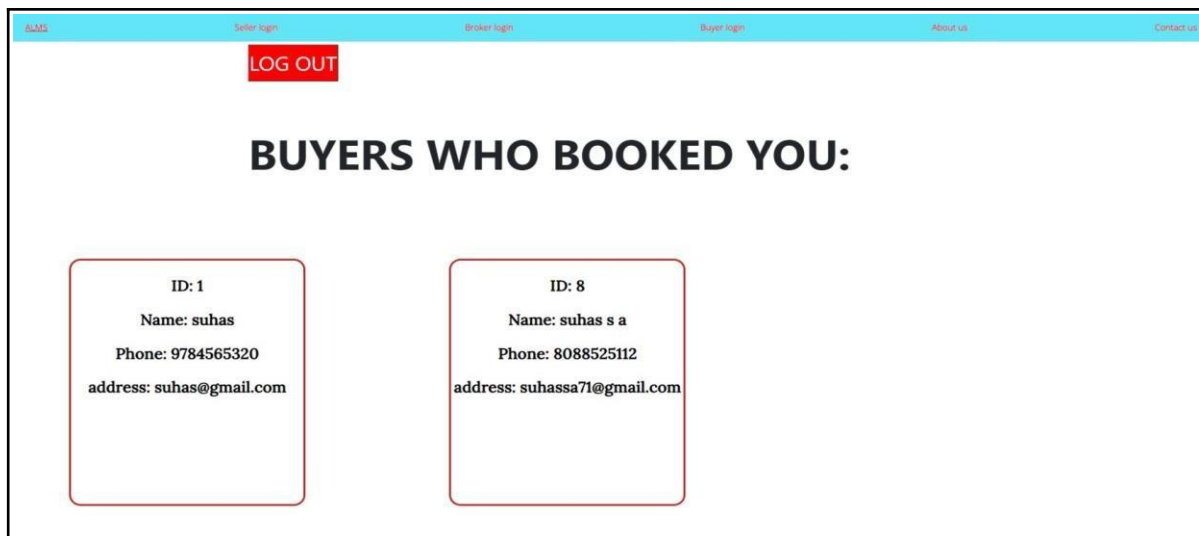
Phone: 8088525112

Email: suhasa71@gmail.com

Book Now

**Fig. 3.13 Brokers available page**

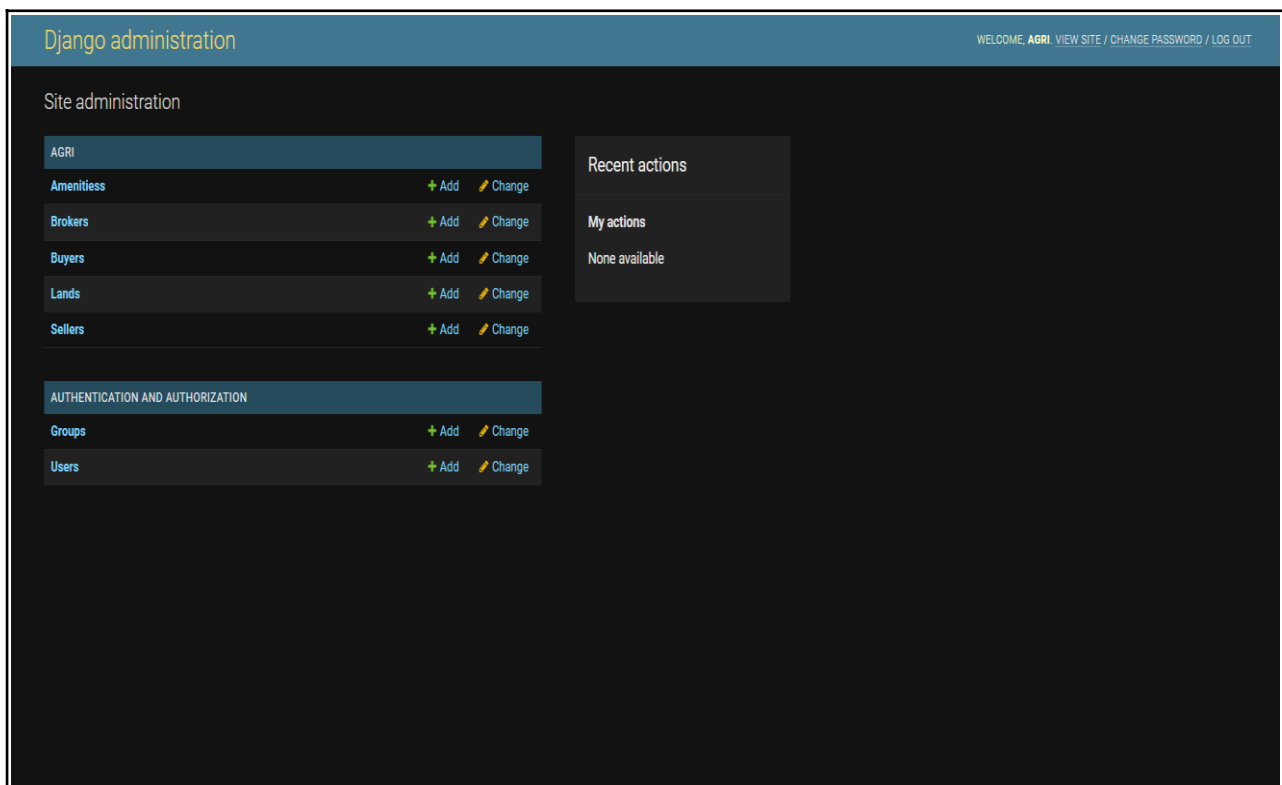
Broker page display the available brokers. The buyers can book the broker they want by clicking the book now button, booked broker will get a message with which buyer booked them for service.



**Fig. 3.14 Broker page**

This page will show the list of buyers who booked a particular broker. It shows the name of the buyer, their id, phone number and email id.

## 3.2 Admin interface



**Fig. 3.15 Admin interface**

Fig. 3.15 shows the outlook of admin interface

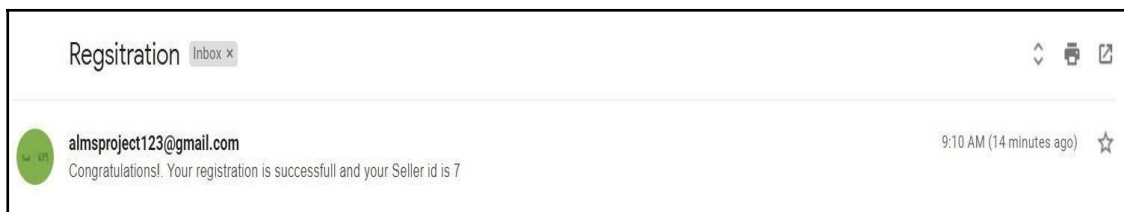
## 3.3 Working of send email and forgot password

### 3.3.1 Email received

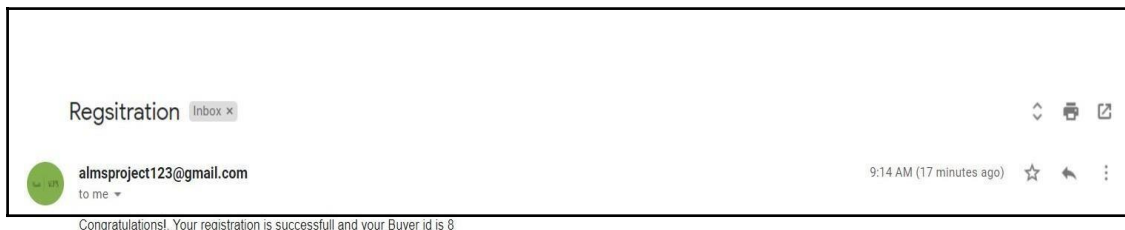
When a user creates a seller account, buyer account or a broker account they receive an email to their mail id regarding the creation of their account. The mail contains the information of their seller id, buyer id and broker id.



**Fig. 3.16 Email received by broker for registration**

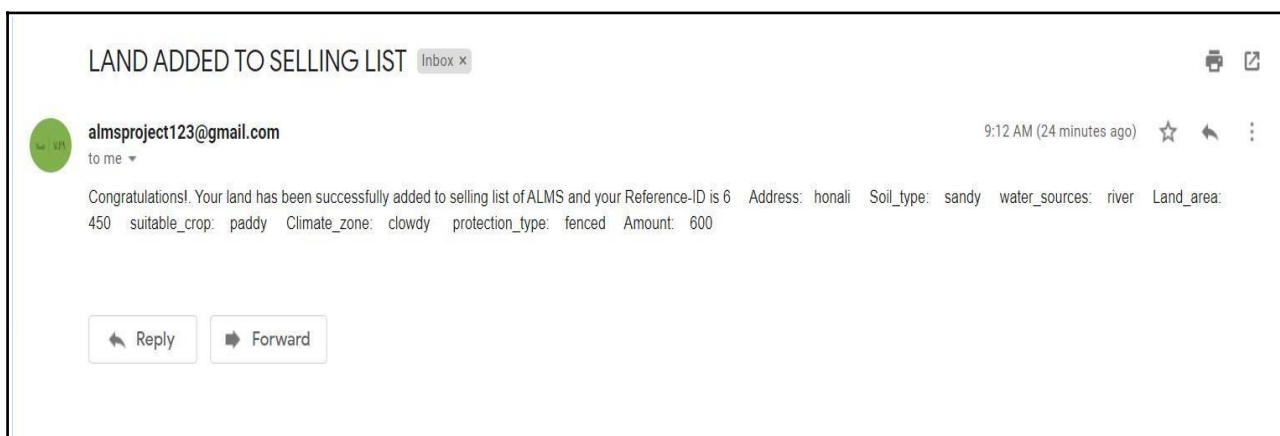


**Fig. 3.17 Email received by seller for registration**



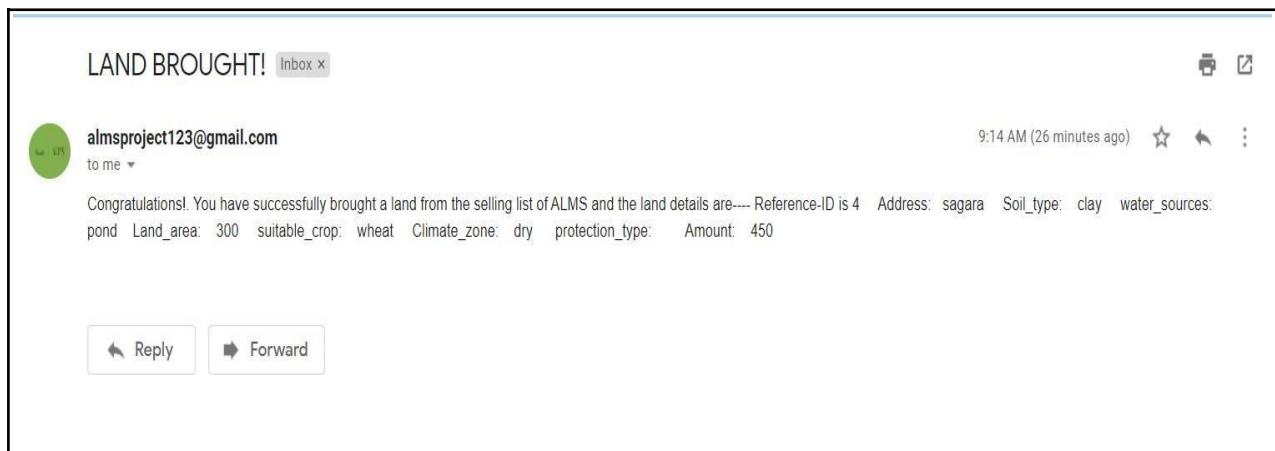
**Fig. 3.18 Email received by buyer for registration**

Figures 3.16, 3.17 and 3.18 shows the mail received by broker, seller and buyer for creating an account. The mail contains their id.



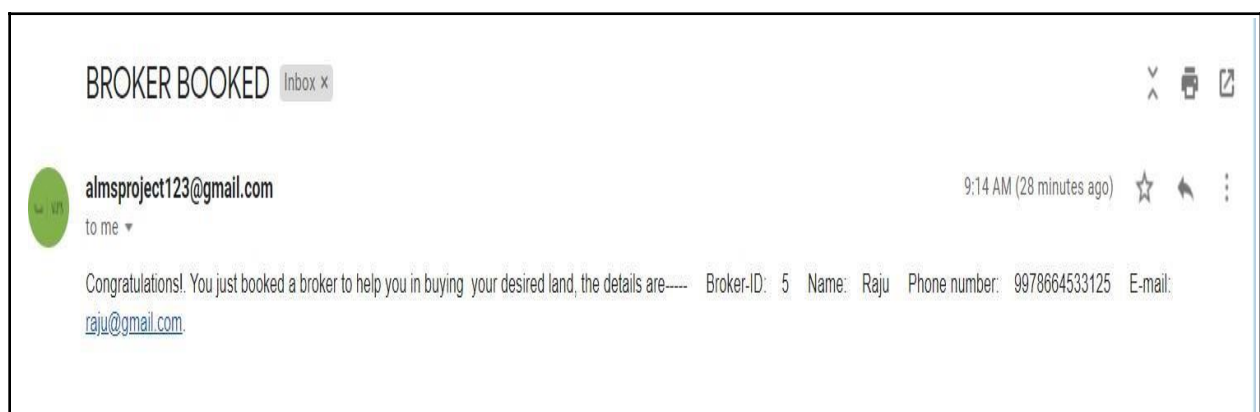
**Fig. 3.19 Email received by seller for adding a land for sale**

Figure 3.19 shows the mail received by seller for adding a land for sale. The mail contains the reference id and the details about the land like address, soil type, water sources, rate etc.



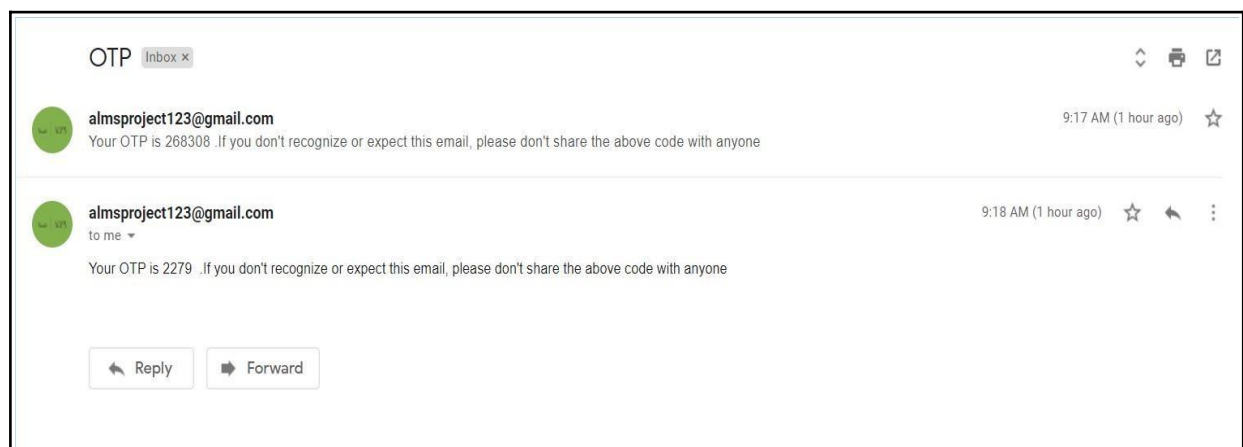
**Fig. 3.20 Email received by buyer for buying a land**

Figure 3.20 shows the mail received by buyer for buying a land. The mail contains the reference id and the details about the land like address, soil type, water sources, rate etc.



**Fig. 3.21 Email received by buyer for booking a broker**

Figure 3.21 shows the mail received by buyer when a buyer for booking broker for their service. The mail contains the broker id, broker name ,brokers phone number and their email id.



**Fig. 3.22 OTP received by user for changing the password**

Figure 3.22 shows the mail containing the otp received by the user to change the account

password.

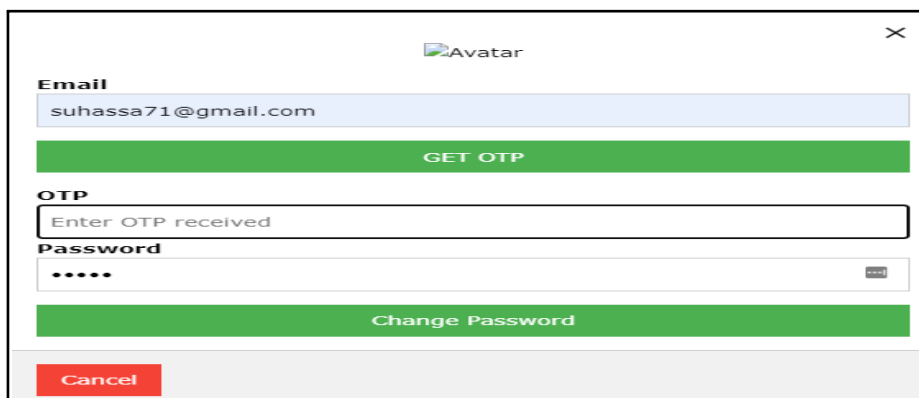
### 3.3.2 Working of Forgot password

When the buyer, seller or broker forgets the password of their account, they can reset the password of their account with the 'forgot password' option.



**Fig. 3.23 Forgot password option**

When the user clicks the forgot password option a new window will open. The user is asked to enter the registered email id to which otp (one time password) will be sent. User should enter the received otp then enter the new password they want to set and then press change password. User accounts password is now changed, the user can login to their account with the new password.



**Fig. 3.24 Password reset window**



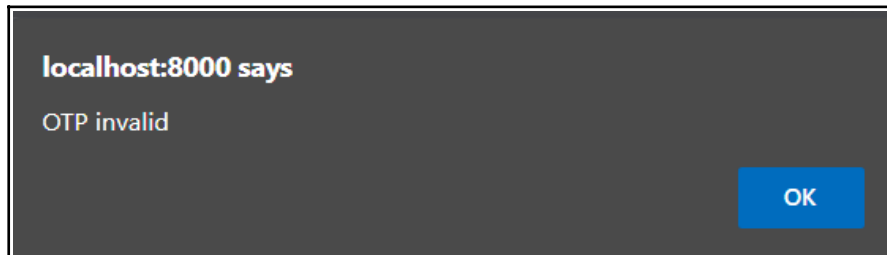
**Fig. 3.25 Notification of OTP sent**

User will get to see this notification when they click 'get otp'.



**Fig. 3.26 Notification of password changed**

User will see the notification when they enter the received otp, enter the new password and change the password.



**Fig. 3.27 Notification of invalid OTP**

If the user enters invalid OTP then the above notification is displayed.



### 3.4 Backend (MySQL) snapshots

Table	Action	Rows	Type	Collation	Size	Overhead
agri_amenities	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
agri_amenities_contains	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
agri_amenities_containss	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
agri_broker	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
agri_broker_helps	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
agri_buyer	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
agri_land	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
agri_seller	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
auth_group	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
auth_group_permissions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
auth_permission	Browse Structure Search Insert Empty Drop	44	InnoDB	utf8mb4_general_ci	32.0 KiB	-
auth_user	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
auth_user_groups	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
auth_user_user_permissions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
django_admin_log	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
django_content_type	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	48.0 KiB	-
django_migrations	Browse Structure Search Insert Empty Drop	19	InnoDB	utf8mb4_general_ci	16.0 KiB	-
django_session	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<b>18 tables</b>	<b>Sum</b>	<b>76</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>640.0 KiB</b>	<b>0 B</b>

**Fig. 3.28 Tables created in the database**

Figure 3.28 shows the required tables created in the database.

	SellerID	Sname	password	Address	Sphone_number
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	ravi	202cb962ac59075b964b07152d234b70	ravi@gmail.com	9764512823
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	suhas	202cb962ac59075b964b07152d234b70	suhas@gmail.com	9876543210
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	shreyas	202cb962ac59075b964b07152d234b70	shreyas@gmail.com	9876543250
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	sanath	202cb962ac59075b964b07152d234b70	sanath@gmail.com	9635642180
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	niranjan	202cb962ac59075b964b07152d234b70	niranjan@gmail.com	9517536540
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	ramesh	6fc42c4388ed6f0c5a91257f096fef3c	suhassa71@gmail.com	8088525112

**Fig. 3.29 Data in agri\_seller table**

Figure 3.29 shows the data of the seller stored in the database when user creates a seller account.

<

**Fig. 3.30 Data in agri\_buyer table**

Figure 3.30 shows the data of the buyer stored in the database when user creates a buyer account.

agri\_amenities\_containss

agri\_broker

agri\_broker\_helps

agri\_buyer

agri\_land

agri\_seller

auth\_group

auth\_group\_permissions

auth\_permission

auth\_user

auth\_user\_groups

auth\_user\_user\_permissions

+ Options

▼

BrokerID

Bname

Brphone\_number

password

address

Edit

Copy

Delete

2

sharma

998855661122

202cb962ac59075b964b07152d234b70

sharma@gmail.com

Edit

Copy

Delete

3

prakash

9896543217

202cb962ac59075b964b07152d234b70

prakash

Edit

Copy

Delete

4

babu rao

9845216985

202cb962ac59075b964b07152d234b70

baburao@gmail.com

Edit

Copy

Delete

5

Raju

9978664533125

202cb962ac59075b964b07152d234b70

raju@gmail.com

Edit

Copy

Delete

6

suhas s a

8088525112

caf1a3dfb505ffed0d024130f58c5cfa

suhassa71@gmail.com

Edit

Copy

Delete

7

prashanth

8088525112

900150983cd24fb0d6963f7d28e177f2

suhassa71@gmail.com

Edit

Copy

Delete

8

srivatsa

9874563215

fc292bd7df071858c2d0f955545673c1

suhassa71@gmail.com

Edit

Copy

Delete

9

shreyas

7892415414

202cb962ac59075b964b07152d234b70

sradigu@gmail.com

**Fig. 3.31 Data in agri\_broker table**

Figure 3.31 shows the data of the broker stored in the database when user creates a broker account.

agri\_amenities\_contains

+

agri\_broker

+

agri\_broker\_helps

+

agri\_buyer

+

agri\_land

+

agri\_seller

+

auth\_group

+

auth\_group\_permissions

+

auth\_permission

+

auth\_user

+ Options

←T→

▼ AmenitiesID

Address

Land\_area

Soil\_type

Amount

water\_sources

suitable\_crop

weather

protection\_type

☐

 Edit

 Copy

 Delete

2

shimoga

350

clay

500

pond

Bean

humid

☐

 Edit

 Copy

 Delete

3

Ablegere

250

sandy

600

river

sugarcane

humid

☐

 Edit

 Copy

 Delete

5

shimoga

350

sandy

450

river

maize

dry

☐

 Edit

 Copy

 Delete

7

shimoga

800

sandy

650

pond

wheat

mixed

barbed

☐

 Edit

 Copy

 Delete

8

sagara

500

clay

400

river

paddy

dry

fenced

☐

 Edit

 Copy

 Delete

9

shikaripura

250

clay

500

tubewell

paddy

cloudy

barbed

**Fig. 3.32 Data in agri\_amenities table**

Figure 3.32 shows the data of the sellers land stored in the database when seller uploads theland details.

<

**Fig. 3.33 Data in agriBrokerHelps table**

Figure 3.33 shows the data of the broker booked buy buyers stored in the database when buyer books a broker. It contains broker id and buyer id.





## **Chapter 4:**

### **Conclusions and Future Scope**

The “AGRICULTURAL LAND MANAGEMENT SYSTEM” is successfully designed and developed. The new computerized system was found to be much faster and reliable and user friendly. The system has been designed and developed step by step and tested successfully. It eliminates the human error that are likely to creep in the kind of working in which a bulk quantity of data and calculations has to be processed. The system results in quick retrieval of information that is very vital for the progress of any organization.

#### **4.1 Future Scope**

The software has been developed in such a way that it can accept modifications and further changes. The software is very user friendly and changes can be done easily. Every system should allow scope for further development and enhancement. The system can be adapted for any further development. The system is so flexible to allow any modification need for the further functioning of programs. Since the objectives may be brought broad in future, the system can be easily modified accordingly, as the system has been modularized. The future expansion can be done in a concise manner in order to improve the efficiency of the system.

## References

- [1] Antonio Mele, Django3 by Example, 3<sup>rd</sup> Edition, Pack Publishers, 2020.
- [2] Daniel Rubio, Beginning Django: Web Application Development and Deployment with Python, APress, 1stEdition, 2017.
- [3] Arun Ravindran, Django Design Patterns and Best Practices, 2<sup>nd</sup> Edition, Pack Publishers, 2020.
- [4] Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Publications, 2016
- [5] <https://erdplus.com/>
- [6] <https://www.w3schools.com/css/>
- [7] Vikram Vaswani, MYSQL The Complete Reference, Tata McgrawHill, 1<sup>st</sup> Edition, 2017