LEASE_IT

Project Report Submitted by

Jubin M Varughese

Reg. No.: AJC18MCA-I039

In Partial fulfillment for the Award of the Degree of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2023

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**LEASE_IT"** is the bona-fide work of **JUBIN M VARUGHESE (Regno: AJC18MCA-I039)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

Mr. G S Ajith

Internal Guide

Ms. Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I here by declare that the project report **"LEASE_IT"** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

Date: 12–03-2023 **Jubin M Varughese**

Kanjirappally **Reg:. AJC18MCA-I039**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Mr. G S Ajith** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

Jubin M Varughese

ABSTRACT

Today, a great deal of land is left uncultivated, and many farmers are addressing issues like "where, why, and when" a specific crop is planted. The lease farming method can increase agricultural output while boosting lessors' and lessees' revenue. Farmers can find property that is available for lease in a variety of locations with a variety of climate conditions and soil kinds. One year is the typical length of a lease. The advance payment is often made at the start of the crop year. Farmers can examine the climatic factors and different soil types to gain insight into which crops would thrive in a certain region and produce more. Also, by clearing off uncultivated land or fields, it makes the land more productive and increases farmers' profits by twofold. The land can be chosen by farmers based on their ideas. If the specific type of land is not accessible at that moment, the data is recorded so that the specific farmer will be notified as soon as the specific type of land becomes available. Also, farmers can gain a thorough understanding of their cultivation process and determine whether their crops are profitable or not. They can also predict how much money they can generate during a specific time period.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	3
2.1	INTRODUCTION	4
2.2	EXISTING SYSTEM	4
2.3	DRAWBACKS OF EXISTING SYSTEM	4
2.4	PROPOSED SYSTEM	4
2.5	ADVANTAGES OF PROPOSED SYSTEM	4
3	REQUIREMENT ANALYSIS	5
3.1	FEASIBILITY STUDY	6
3.1.1	ECONOMICAL FEASIBILITY	6
3.1.2	TECHNICAL FEASIBILITY	6
3.1.3	BEHAVIORAL FEASIBILITY	6
3.2	SYSTEM SPECIFICATION	7
3.2.1	HARDWARE SPECIFICATION	7
3.2.2	SOFTWARE SPECIFICATION	7
3.3	SOFTWARE DESCRIPTION	7
3.3.1	VUE JS	7
3.3.2	MYSQL	7
3.3.3	NODE JS	8
4	SYSTEM DESIGN	9
4.1	INTRODUCTION	10
4.2	UML DIAGRAM	10
4.2.1	USE CASE DIAGRAM	10
4.2.2	SEQUENCE DIAGRAM	11-12
4.2.3	STATE CHART DIAGRAM	13-14
4.2.4	ACTIVITY DIAGRAM	15
4.2.5	CLASS DIAGRAM	16
4.2.6	OBJECT DIAGRAM	17
4.2.7	COMPONENT DIAGRAM	18

4.2.8	DEPLOYMENT DIAGRAM	19
4.2.9	COLLABORATION DIAGRAM	20
4.3	USER INTERFACE DESIGN USING FIGMA	21-22
4.4	DATA BASE DESIGN	23-29
5	SYSTEM TESTING	30
5.1	INTRODUCTION	31
5.2	TEST PLAN	31
5.2.1	UNIT TESTING	32
5.2.2	INTEGRATION TESTING	32
5.2.3	VALIDATION TESTING	33
5.2.4	USER ACCEPTANCE TESTING	33
5.2.5	AUTOMATION TESTING	33-34
5.2.6	SELENIUM TESTING	34-41
6	IMPLEMENTATION	42
6.1	INTRODUCTION	43
6.2	IMPLEMENTATION PROCEDURE	43
6.2.1	USER TRAINING	44
6.2.2	SYSTEM MAINTENANCE	44
6.2.3	TRAINING ON APPLICATION SOFTWARE	44
6.2.4	HOSTING	44
7	CONCLUSION & FUTURE SCOPE	45
7.1	CONCLUSION	46
7.2	FUTURE SCOPE	46
8	BIBLIOGRAPHY	47-48
9	APPENDIX	49
9.1	SAMPLE CODE	50-74
9.2	SCREEN SHOTS	75-77

List of Abbreviation

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

VUE - Virtual University Enterprises.

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Lease_It is a solution for people in order to make their agricultural lands more productive. It contains essential modules and features that systematically help farmers to manage their agricultural activities. It is a platform for farmers to get lands for lease for agricultural purposes and also by using different machine learning techniques, the system will recommends the most suitable crops for that particular lands. By cultivating the right crops in the fields farmers will get more profits. Landowners can post their agricultural lands in this platform and the interested farmer can lease that land. Agriculture is the world's largest industry. Pasture and cropland occupy around 50 percent of the Earth's habitable land and provide habitat and food for a multitude of species. Demand for agricultural commodities is rising rapidly as the world's population grows. Agriculture's deep connections to the world economy, human societies and biodiversity make it one of the most important frontiers for conservation around the globe. When agricultural operations are managed, we can preserve and restore critical habitats, help protect watersheds, and improve soil health and water quality. The general objective of this system is to make the agricultural lands more productive. Hence the project is developed proficiently to help both landowners and farmers to automate their operations. In proposed system, we provide facility to both landowners and farmers according to their convenience.

1.2 PROJECT SPECIFICATION

Lease_It is a solution to make the agricultural lands more productive. Agriculture is considered to be the backbone of economic system for developing countries. The major goal of this task is to enable the owner of the land to deal directly with the farmer. In addition, the farmers can place their choice of the land, so that when that particular land is available they will get notification and they can lease that land for agricultural purposes. This idea is discovered by keeping in mind the fact that there is no platform for farmers to lease agricultural lands. The farmers can check available lands at a particular place near to them and can lease the lands if available. The farmers can analysis the climatic conditions and soil types so that they get the ideas about which crop grow well in that particular area and it also increase more production. The farmers and landowners also given with the facility to view the details and can cancel the agreement if required.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

Lease_It is user friendly software which is fast and cost effective. It provides services like agricultural plots for leasing, predicts suitable crops for different lands and also make the environment greener. It also helps to eliminate land or fields from uncultivated and make the land more useful and also helps farmers to double their profit.

2.2 EXISTING SYSTEM

In the present system people are getting knowledge about the available agricultural plots by their mutual communication. So most of the people are not known by these available lands because of lack of platforms and communication facilities.

2.3 DRAWBACKS OF EXISTING SYSTEM

- There is no platform for people to post and get available agricultural plots.
- Less user friendly.

2.4 PROPOSED SYSTEM

Lease_It is a user friendly and cost effective platform where people can post and get agricultural plots.

The main function of the system is farmers can get land which are for lease in different places and of different climatic conditions and soil types. The farmers can analysis the climatic conditions and soil types so that they get the ideas about which crop grow well in that particular area and it also increase more production.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- User friendly
- Agricultural plots for leasing, predicts suitable crops for different lands and also make the environment greener.
- Helps farmers to double their profit.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Involves determining the sustainability of the project and putting up a business proposal that has a very detailed project plan as well as some cost forecasts. During system analysis, the suggested system's feasibility study must be finished. This is done to ensure that the system won't cost businesses money.

3.1.1 Economical Feasibility

Lease_It is enhanced to be economically feasible by:

- Reducing the overhead of farmers for finding the right agricultural land and exact crop for the particular land.
- The site will be user friendly and easy to use.
- For peoples it’s a right platform to make their agricultural land more productive.

3.1.2 Technical Feasibility

It considers the technical requirements for the system.

- NODE is used as the backend technology. Its’ a stable and one of the best language for web development.
- For frontend VUE JS, CSS is used to develop a well responsive site.

3.1.3 Behavioral Feasibility

It is a measure of how well a proper system solves the problem and opportunities identified during the scope of definition.

Lease_It is proposed to be operationally feasible by:

- It’s an open site for both farmers and landowners that provides throughput in less response time.
- It provides users with accurate and useful formatted information’s.
- It is flexible and expandable and also it is built in a way that all can adapt to the applying changes easily.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - intel Core i7 10th gen

RAM - 8 G B

Hard disk - 1 T B

3.2.2 Software Specification

Front End - VUE JS, CSS

Backend - MYSQL NODE

Client on PC - Windows 7 and above.

Technologies used - Captcha, Chatbot, Sentimental Analysis, Data visualization

3.3 SOFTWARE DESCRIPTION

3.3.1 VUE JS

Developers mostly use the Javascript framework Vue.js to create interactive user interfaces. This framework is more flexible and lighter than AngularJS, which is why many developers, from novices to specialists, choose it. It consists of a number of libraries, the most significant of which is the core library, which concentrates on the project's frontend's view part. It is generally used to develop engaging and aesthetically pleasing one-page applications and interactive web interfaces. With the help of HTML extensions and other helpful plugins, the benefits of Vue.js are increased. It is employed by developers for both desktop and mobile applications. Model View Controller architecture is one of its included features. Developers may see the user interface of their project using this architecture, whether it is a website, desktop app, or mobile app. It is because of this lightweight and useful feature that many developers like Vue.

3.3.2 MySQL

Information is organised into one or more data tables in a relational database. These data tables may be connected to one another, and these connections help give the data its structure. To create, modify, and retrieve data from relational databases as well as control user access to the databases, programmers use the SQL language. An RDBMS, like MySQL, collaborates with an operating system to build a database system in a computer's storage solution. It also handles users, provides network access, and makes it simpler to assess database integrity and produce backups.

3.3.3 NODE JS

An open source, cross-platform runtime environment called Node.js is used to create networking and server-side applications. Applications for Node.js can be created in JavaScript and run on Linux, OS X, and Microsoft Windows using the Node.js runtime. Also, it offers a comprehensive library of different JavaScript modules, greatly streamlining the creation of web applications using Node.js.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements.

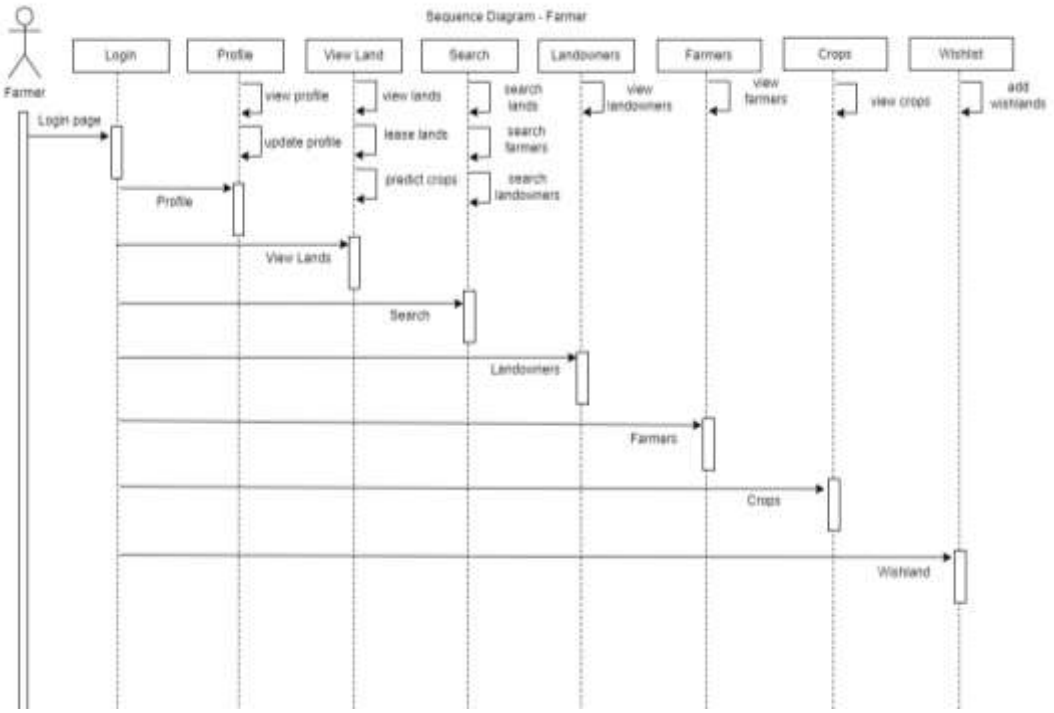
4.2 UML DIAGRAM

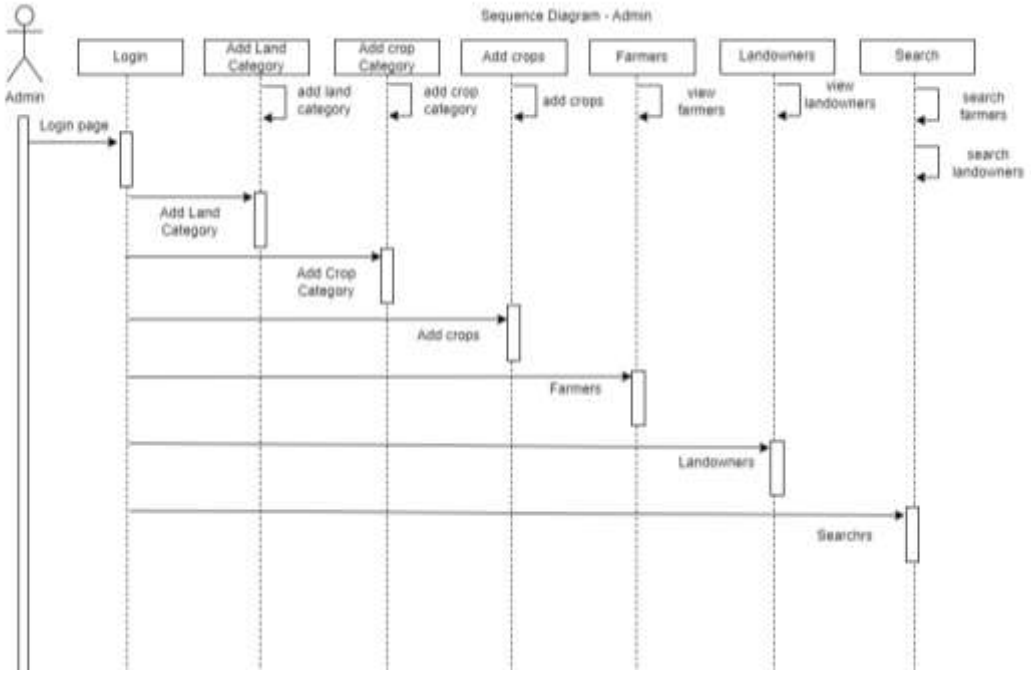
4.2.1 Use Case Diagram

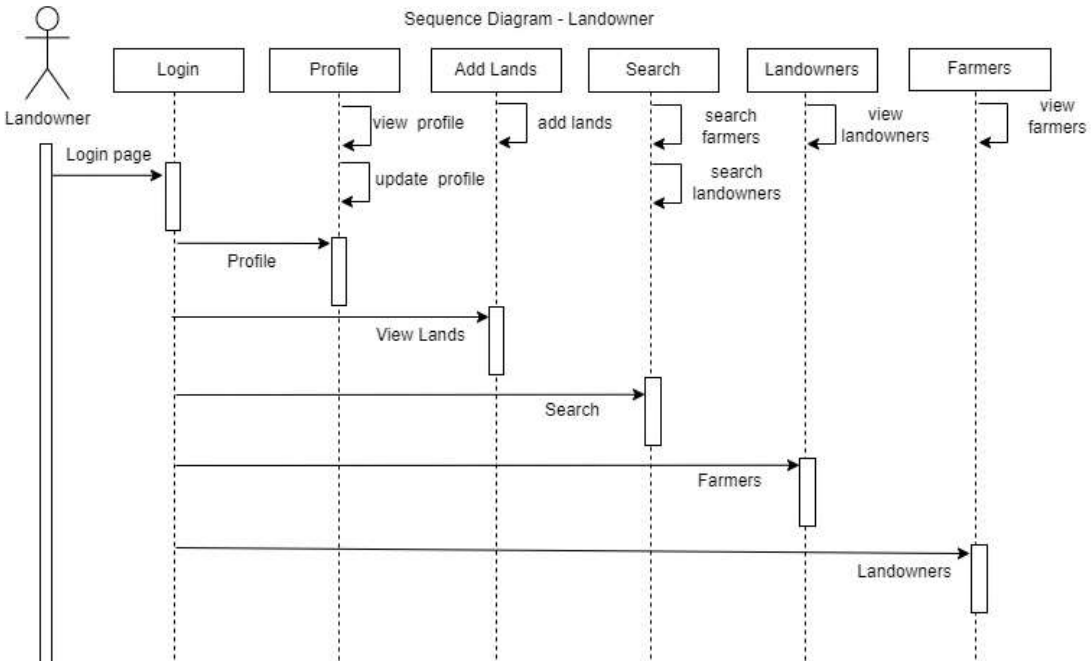
The use case diagram, also known as a behaviour diagram, is used to describe all user actions within a system. Each and every user described in the use case is an actor, and the functionality is a system action.

4.2.2 Sequence Diagram

Developers frequently use sequence diagrams to model the interactions between items in a single use case. They demonstrate the interactions that take place when a specific use case is executed and the order in which various system components interact with one another to perform a function.

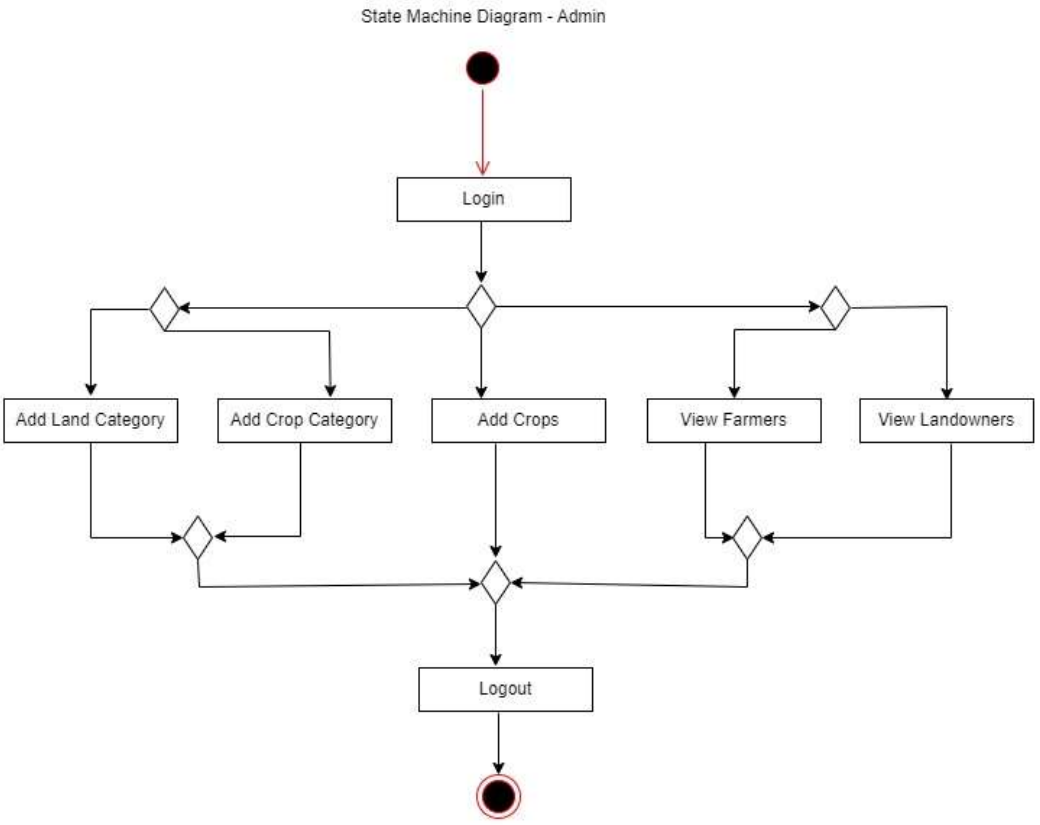




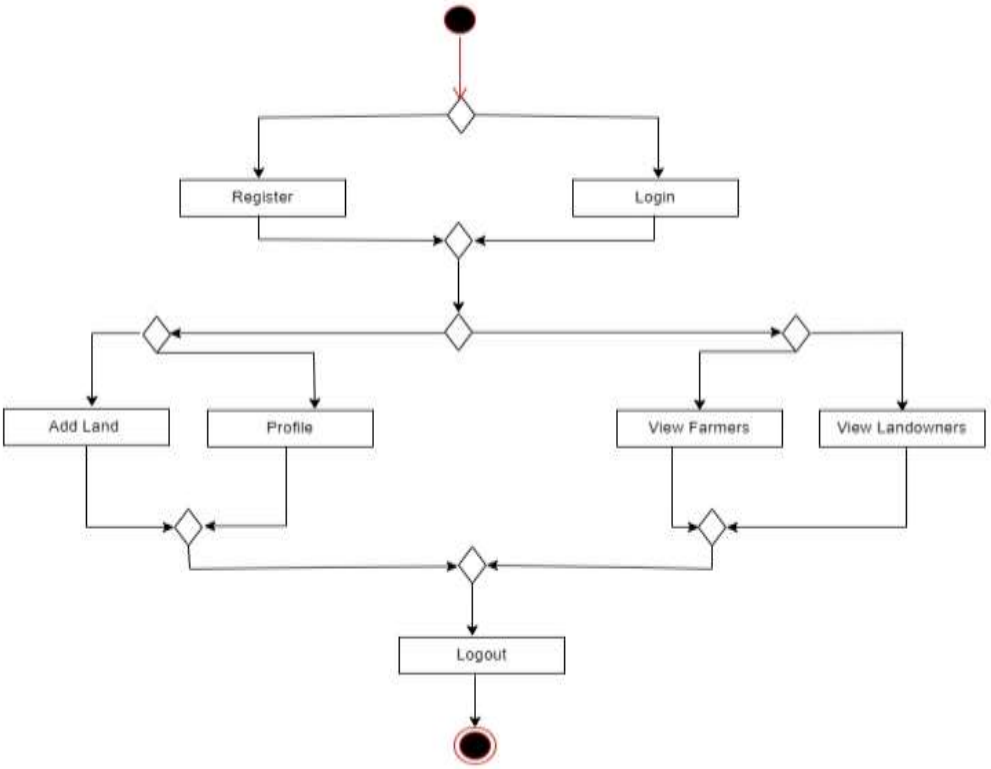

Sequence Diagram - Landowner
Landowner
Login
Profile
Add Lands
Search
Landowners
Farmers
view profile
add lands
search farmers
view landowners
view farmers
Login page
update profile
search landowners
Profile
View Lands
Search
Farmers
Landowners

4.2.3 State Chart Diagram

The states of various items during their life cycles are represented by state chart diagrams. The state changes that result from certain internal or external events are highlighted. These object states are crucial for accurate analysis and implementation. Diagrams of state charts are crucial for describing the states. States are defined as the state of an item at the time of an occurrence.

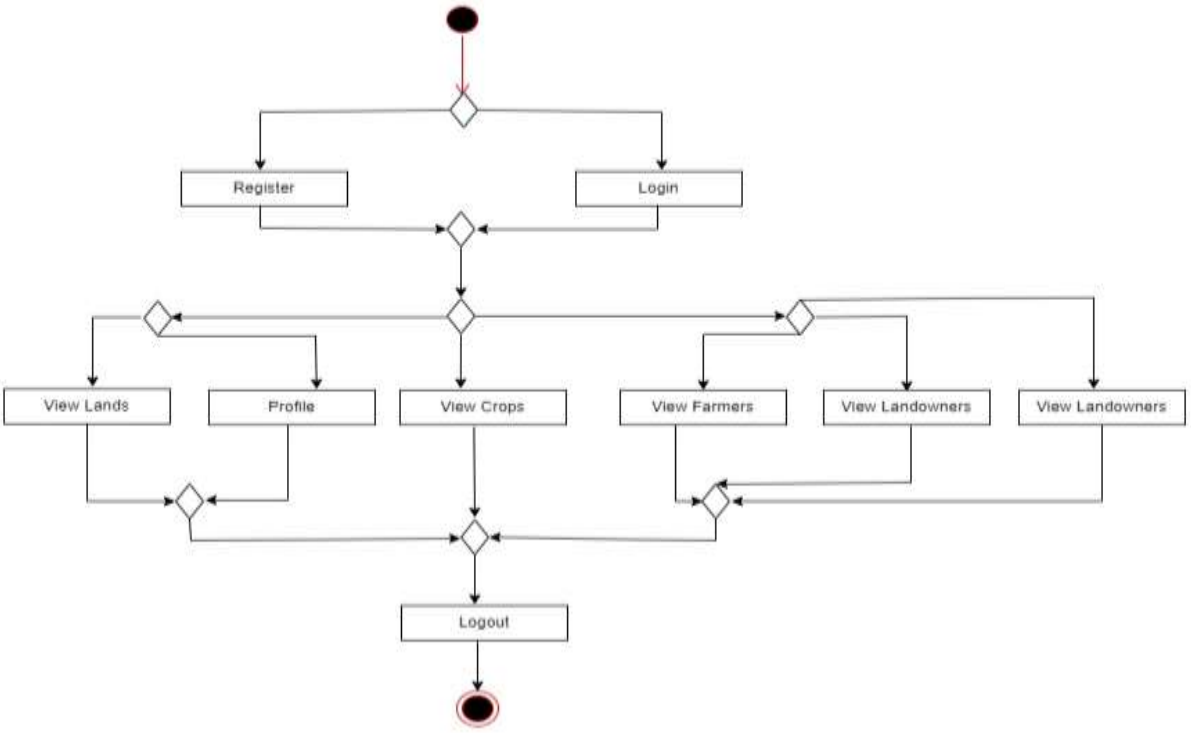


State Machine Diagram - Landowner



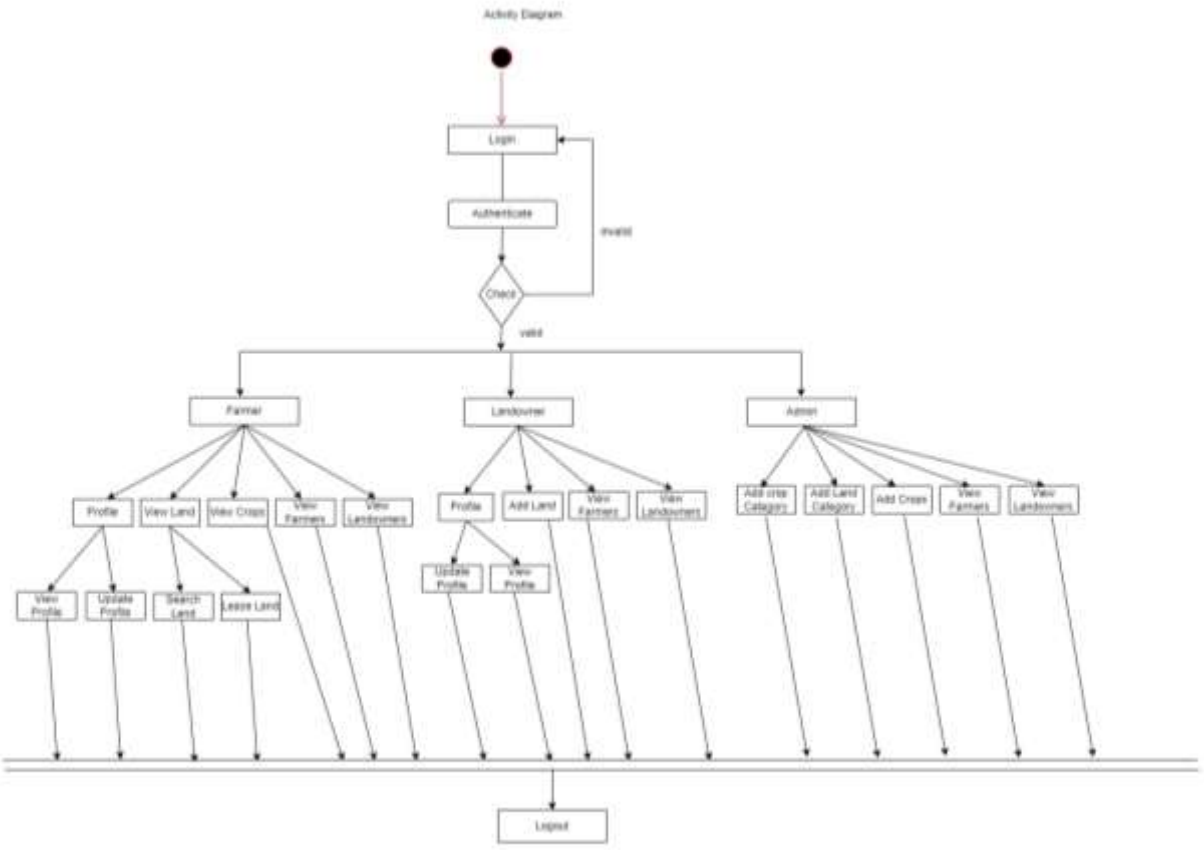
`

State Machine Diagram - Farmer



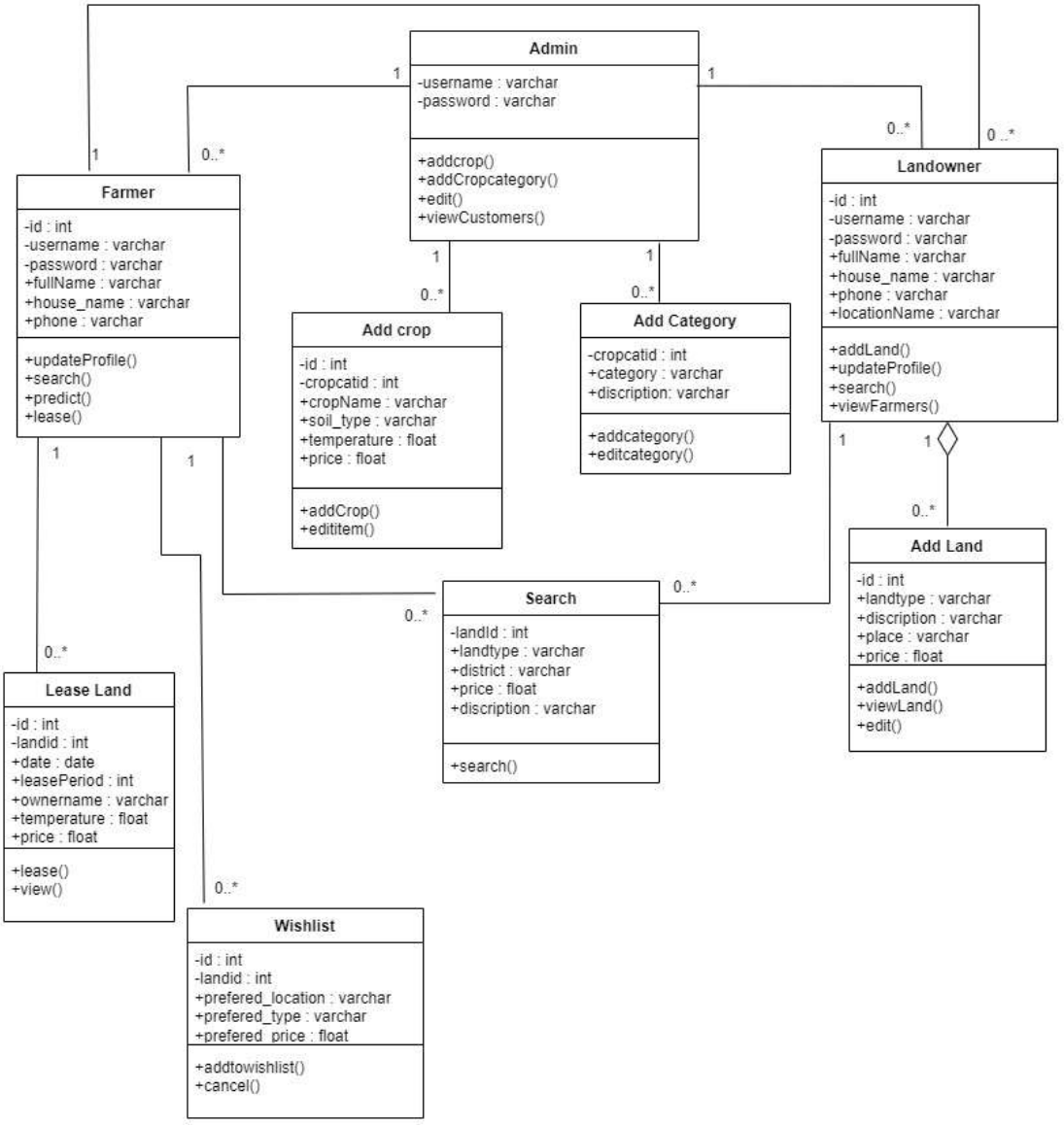
4.2.4 Activity Diagram

Similar to a flowchart or data flow diagram, an activity diagram visually displays a series of actions or the flow of control in a system. A common tool in business process modelling is the activity diagram.
Also, they can outline the procedures in a use case diagram.



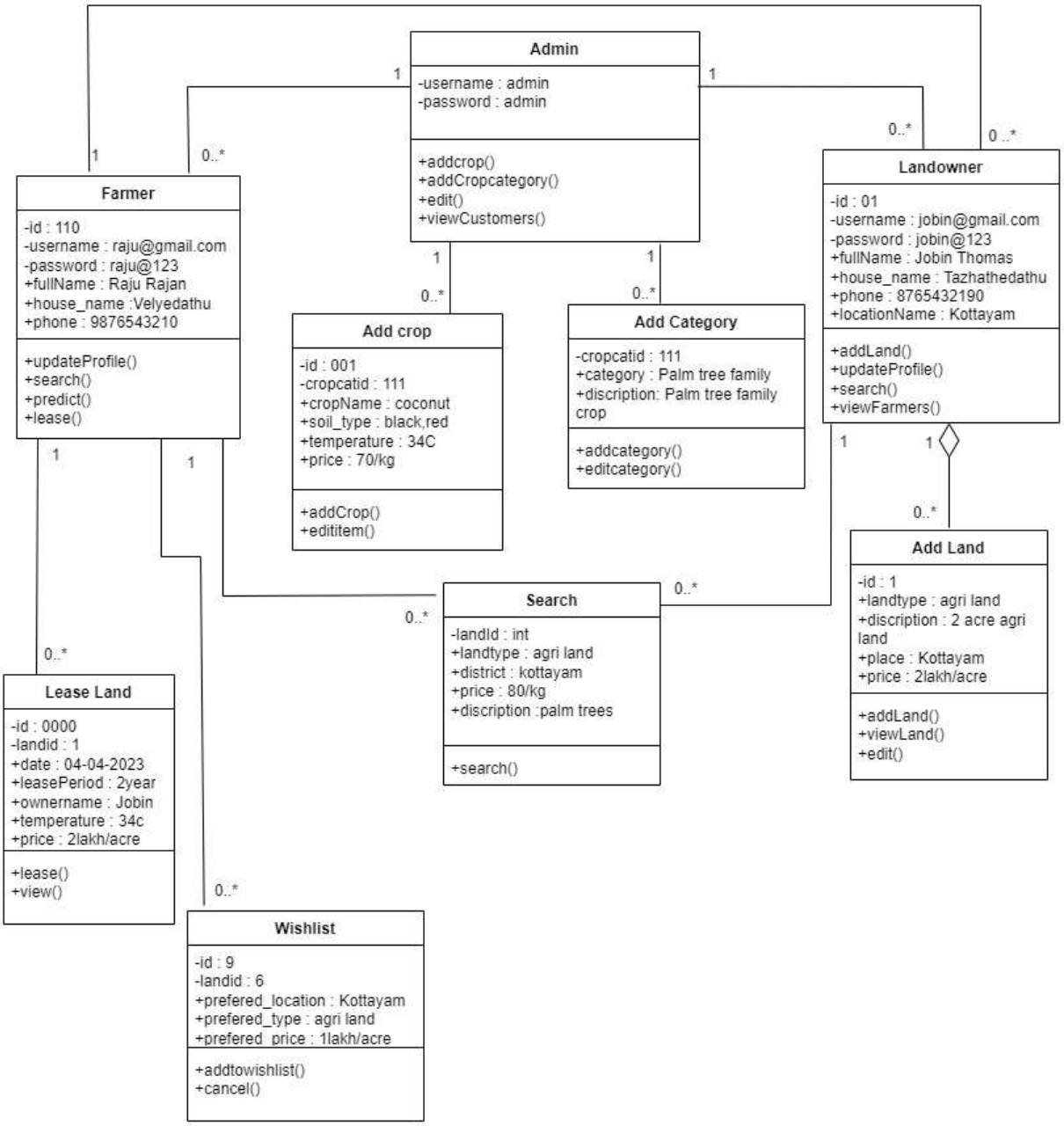
4.2.5 Class Diagram

The class diagrams are the system or subsystem's blueprints. Class diagrams are useful for modelling the system's constituent parts, showing the relationships among them, and describing the functions and services that each perform.



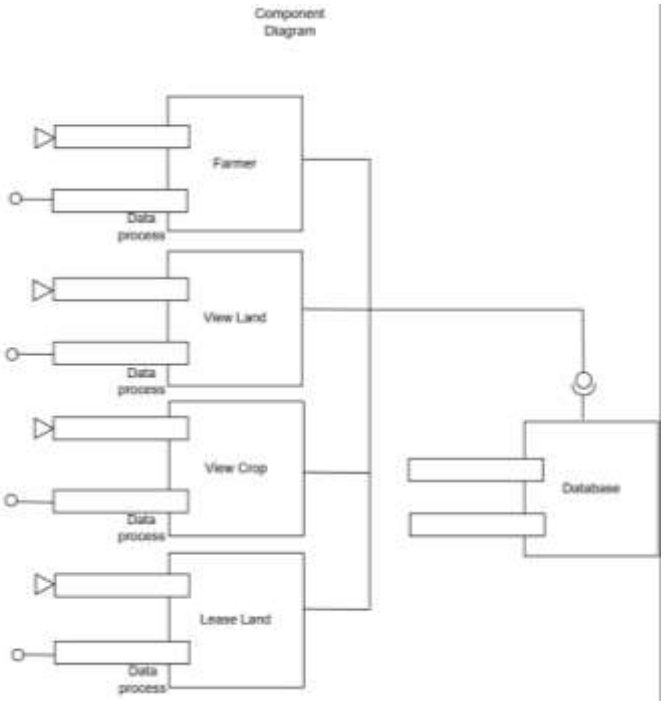
4.2.6 Object Diagram

A particular instance of a class diagram at a specific time is represented by a UML object diagram. You'll notice several graphic representations of this that resemble the class diagram. An object diagram focuses on the characteristics of a group of objects and their relationships with one another.



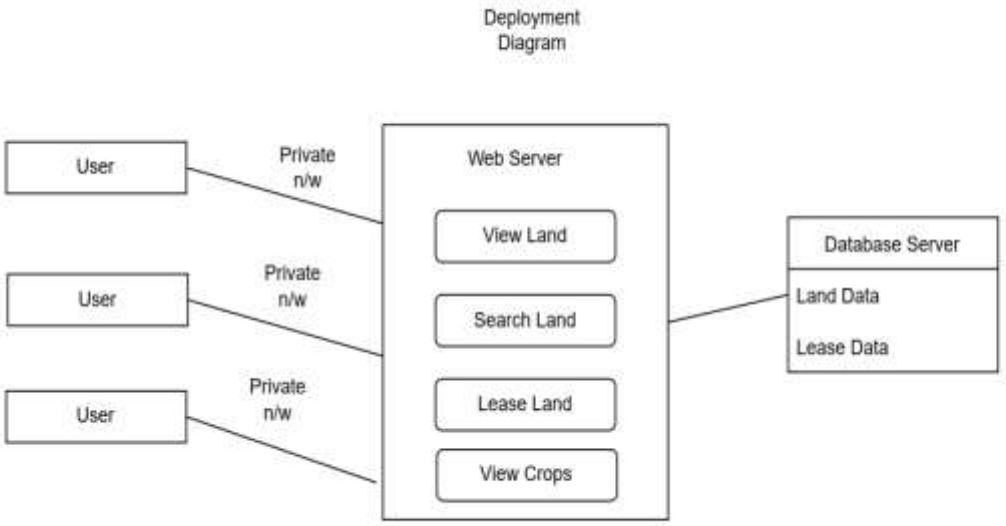
4.2.7 Component Diagram

A component diagram's objective is to depict the interrelationships between various system components. A module of classes that represent autonomous systems or subsystems with the capacity to interface with the rest of the system is referred to as a "component" for the purposes of UML 2.0.



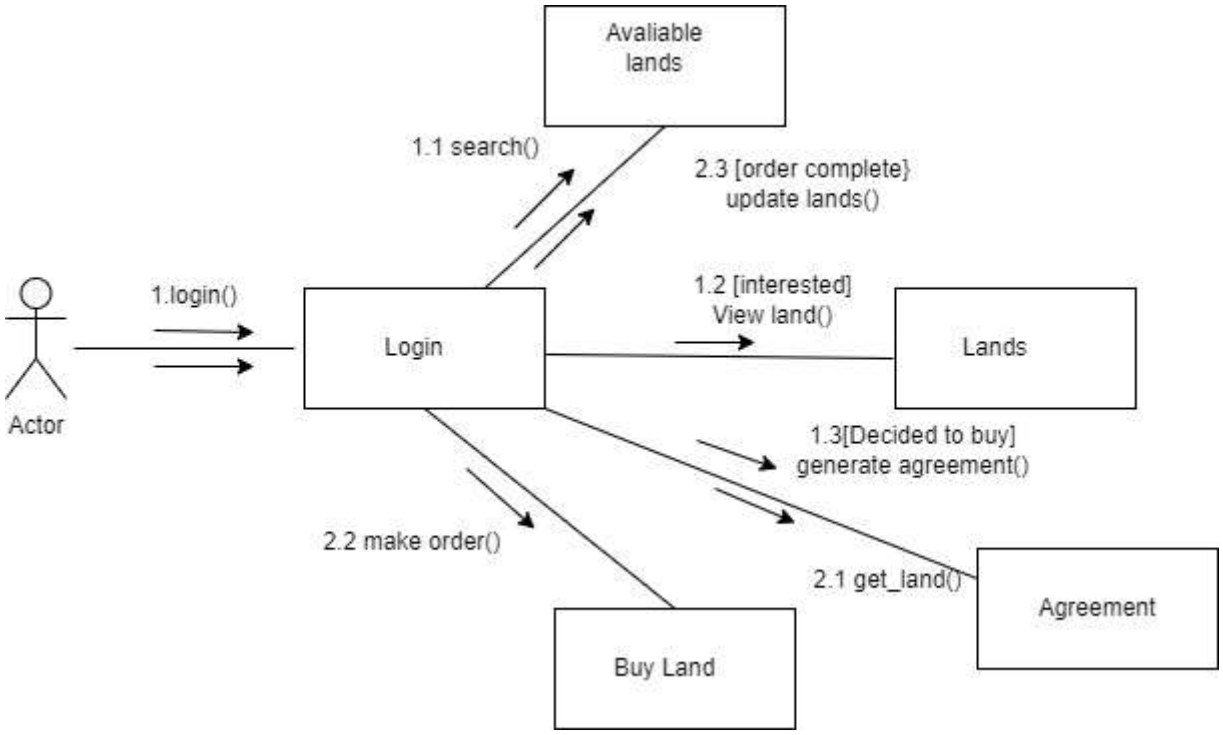
4.2.8 Deployment Diagram

Deployment diagrams in UML represent a system's physical architecture. The relationships between the system's hardware and software components as well as the physical distribution of processing are shown in the deployment diagram.



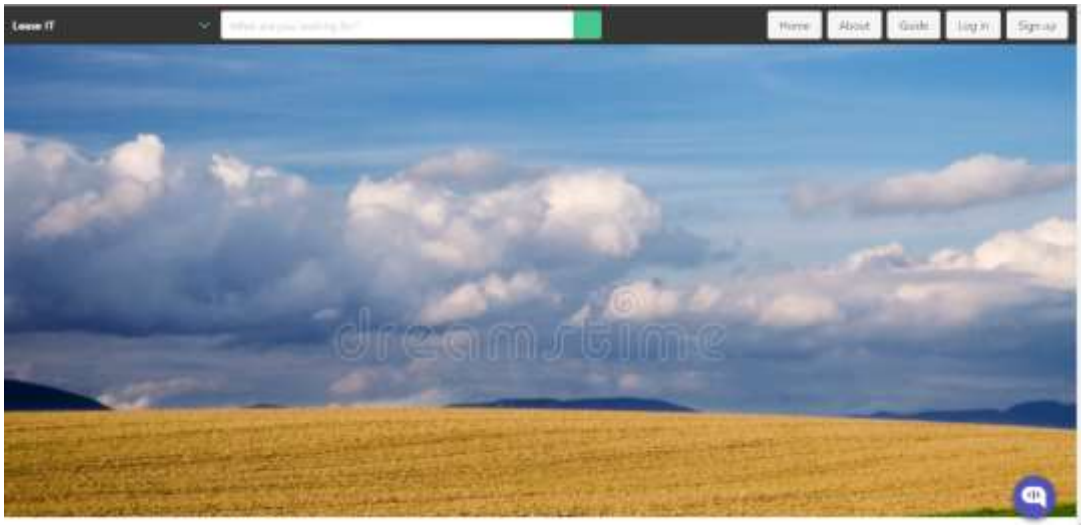
4.2.8 Collaboration Diagram

The relationship between the objects in a system is depicted using the cooperation diagram. The same data is shown differently in both the sequence and cooperation diagrams. As it is based on object-oriented programming, it represents the architecture of the object living in the system rather than the flow of messages. An object is made up of various features. The system's various objects are connected to one another. The system's object and architecture are shown using the collaboration diagram, sometimes referred to as a communication diagram. When it is crucial to show the interaction between the objects, collaborations are employed. The information is the same in both the sequence and cooperation diagrams, but they depict it very differently.

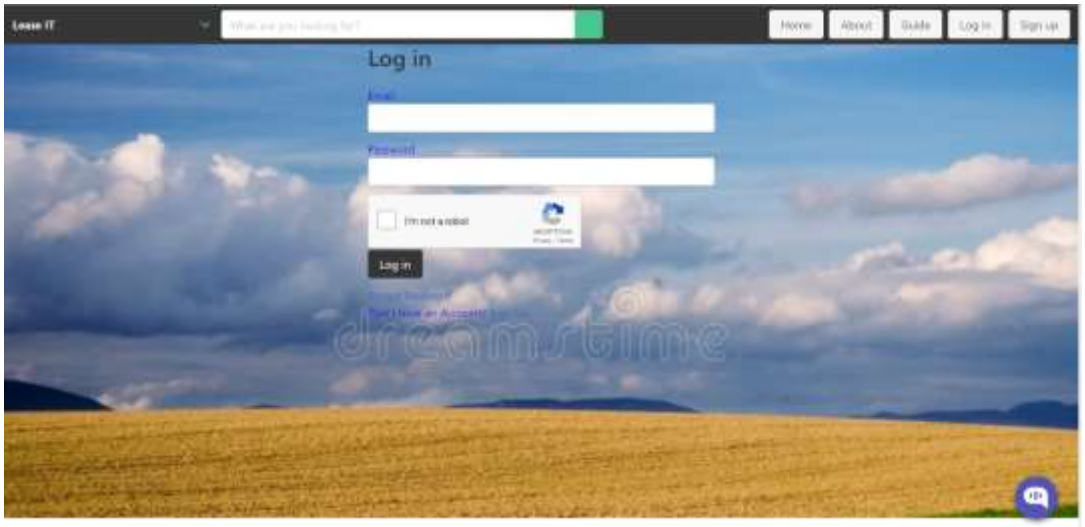


4.3 USER INTERFACE DESIGN USING FIGMA

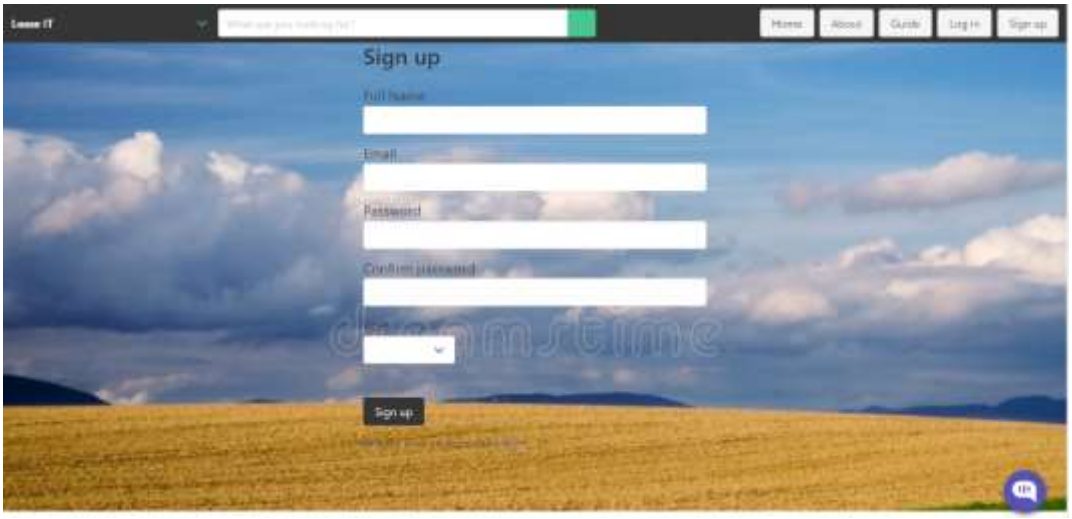
Form Name: Home page



Form Name: Login page

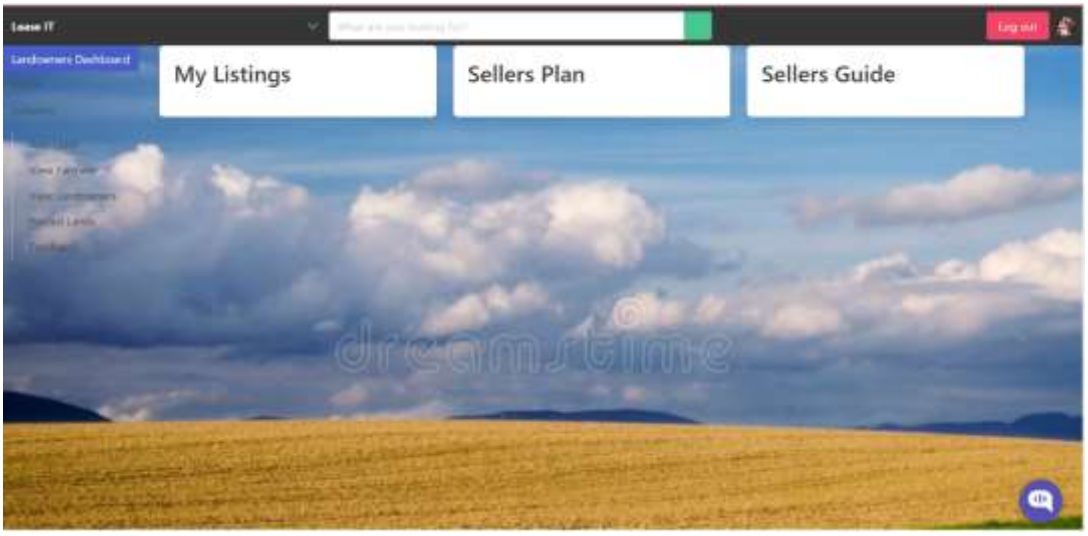


Form Name: Registration page



Form Name: Farmer home page

Form Name: Landowner home page



4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

Relational Database Management System is referred to as RDBMS. RDBMS is the foundation of every contemporary database management system, including SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access. Because it is based on the relational model created by E.F. Codd, it is known as Relational Database Management System (RDBMS). In RDBMS, data is displayed as rows of tuples. The most popular type of database is a relational database. It has a number of tables, each with a unique primary key. Data may be accessible quickly in RDBMS because of a collection of a well-organized set of tables.

4.4.2 Normalization

A database design method called normalisation avoids data duplication and gets rid of undesired traits like Insertion, Update, and Delete Anomalies. Using relationships, normalisation rules break up larger tables into smaller ones. To get rid of redundant (repetitive) data and make sure that data is stored correctly, SQL normalisation is used. With the introduction of the First Normal Form, the relational model's creator Edgar Codd put out the notion of data normalisation, and he later expanded it with the Second and Third Normal Forms. Subsequently, he collaborated with Raymond F. Boyce to create the Boyce-Codd Normal Form theory.

4.4.3 Sanitization

To ensure that no residual data can be recovered even after comprehensive forensic investigation, data sanitization entails the secure and irreversible erasure of sensitive material from datasets and media. Although data sanitization has several uses, it is typically employed to clean out outdated technology or to share and use huge datasets that contain private data. Physical destruction, cryptographic erasure, and data erasure are the primary methods for removing personal data from devices. Although some people would assume that data sanitization solely applies to data on electronic media, the phrase actually extensively refers to data on physical medium as well, such as paper copies. For computer files, these data kinds are referred to as soft, and for tangible medium like paper copies, as hard. Sensitive data is also cleaned using data sanitization techniques, such as heuristic-based techniques, machine-learning techniques, and k-source anonymity.

4.4.4 Indexing

Indexing is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. The index is a type of data structure. It is used to locate and access the data in a database table quickly.

- Primary Index − Primary index is defined on an ordered data file. The data file is ordered on a key field. The key field is generally the primary key of the relation.
- Secondary Index − Secondary index may be generated from a field which is a candidate key and has a unique value in every record, or a non-key with duplicate values.
- Clustering Index − Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.

4.5 TABLE DESIGN

1.Tbl_user

Primary key: **id**

No	Fieldname	Datatype	Size	Key constraints	Description
1	id	int	2	Primary Key	Primary key of the table
2	email	varchar	25	Not Null	To store login email of user
3	fullName	varchar	25	Not Null	To store user name
4	role	varchar	10	Not Null	To store login category of user
5	password	varchar	25	Not Null	To store login password of user
6	profile	varchar	25	Not Null	To store image
7	street	varchar	25	Not Null	To store street name
8	city	varchar	25	Not Null	To store city name
9	pincode	int	10	Not Null	To store pincode name
10	house_name	varchar	25	Not Null	To store house name
11	phone	varchar	15	Not Null	To store phone number

2. Tbl_landcategory

Primary Key: category_id

No	Fieldname	Datatype	Size	Key constraints	Description
1	lcatid	int	2	Primary Key	Primary key of the table
2	category	varchar	25	Not Null	To store name of the category
3	image	varchar	60	Not Null	To store image
4	description	varchar	50	Not Null	To store details of the category

3. Tbl_land

Primary Key: lid

Foreign Key: category references table **landcategory , userid** references table **user**

No	Fieldname	Datatype	Size	Key constraints	Description
1	lid	int	2	Primary Key	Primary key of the table
2	userid	int	2	Foreign Key	Foreign key
3	survey_number	int	10	Not Null	To store survey number
4	image	varchar	60	Not Null	To store image of the land
5	price	int	10	Not Null	To store price of the land
6	locationName	varchar	25	Not Null	To store location of the land
7	extend	int	10	Not Null	To store area of the land
8	status	int	2	Not Null	To store status of the land
9	price_per_acer	varchar	10	Not Null	To store posted date
10	leaseperiod	int	11	Not Null	To store lease time
11	category	varchar	25	Foreign Key	Foreign key of landcategory table
12	description	varchar	50	Not Null	To store description
13	ownername	varchar	25	Not Null	To store ownername value
14	ph	int	25	Not Null	To store ph value
15	pottasium	int	25	Not Null	To store Potassium value
16	phosphorous	int	25	Not Null	To store Phosphorous value
17	nitrogen	int	25	Not Null	To store nitrogen value
18	state	varchar	25	Not Null	To store state name
19	district	varchar	25	Not Null	To store district name

20	status	int	25	Not Null	To store status

4. Tbl_cropcategory

Primary Key: cropcatid

No	Fieldname	Datatype	Size	Key constraints	Description
1	cropcatid	int	2	Primary Key	Primary key of the table
2	category	varchar	25	Not Null	To store category name
3	image	varchar	60	Not Null	To store image
4	description	varchar	60	Not Null	To store details of category

5. Tbl_crops

Primary Key: cropid

Foreign Key: cropcatid references table **cropcategory**

No	Fieldname	Datatype	Size	Key constraints	Description
1	cropid	int	2	Primary Key	Primary key of the table
2	cropcatid	int	2	Foreign Key	Foreign key of the cropcategory table
3	cropname	varchar	25	Not Null	To store crop name
4	image	varchar	70	Not Null	To store crop image
5	price	int	10	Not Null	To store crop price
6	soil_type	varchar	15	Not Null	To store soil details
7	temperature	int	10	Not Null	To store temperature details
8	description	int	10	Not Null	To store humidity details
9	harvesting_time	varchar	20	Not Null	To store harvesting details

10	climatic	varchar	25	Not Null	To store climatic conditions

7. Tbl_wishland

Primary Key: wlid

Foreign Key: id references table **user**

No	Fieldname	Datatype	Size	Key constraints	Description
1	wlid	int	15	Primary Key	Primary key of the table wishland
2	userid	int	15	Foreign Key	Foreign Key
3	prefered _location	varchar	25	Not Null	To store prefered _location values
4	preferred_category	varchar	25	Not Null	To store preferred_category values
5	preferred_extension	varchar	10	Not Null	To store preferred_extension values
6	preferred_ price	varchar	25	Not Null	To store price
7	Description	varchar	50	Not Null	To store description
8	lease_period	int	15	Not Null	To store lease_period value

7. Tbl_payment

Primary Key: pid

Foreign Key: userid references table **user, landed** references table **land**

No	Fieldname	Datatype	Size	Key constraints	Description
1	pid	int	15	Primary Key	Primary key of the table payments
	userid	int	15	Foreign Key	Foreign Key
3	landid	int	15	Not Null	Foreign key
4	orderid	int	15	Not Null	To store order id
5	Payment status	varchar	10	Not Null	To store payment status

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Testing is the phase of implementation that aims to ensure that the system is functioning appropriately and effectively. The new system's flaws will be found during system testing, and they will be fixed. Performance criteria for system testing include things like turnaround speed, backup, file protection, and human aspects. The procedure of system testing has the objective of finding every flaw in our product. The programme was given a series of test inputs, and numerous observations were made. It will be determined whether the programme behaves as expected or not based on these observations.

There Are Two Types of Software Testing

- Black Box Testing
- White Box Testing

Software testing's black box testing subcategory includes system testing. White box testing refers to testing a software application's internal logic or code. The converse is true with black box or system testing. The exterior functionality of the software from the user's perspective is tested as part of the system.

5.2 TEST PLAN

A test plan is a thorough document that details the topics and actions involved in software testing. It describes the test strategy, goals, timetable, needed resources (people, software, and hardware), estimation for the test, and test deliverables. Every software testing process starts with a test plan. It is the most important task that guarantees the availability of all lists of scheduled activities in the proper order. The test plan serves as a guide for carrying out software testing tasks as a clear process that is completely under the testing manager's supervision and control. There are three different kinds of test plans.

o Master Test Plan

o Phase Test Plan

o Testing Type Specific Test Plans

Master Test Plan

Master Test Plan is a type of test plan that has multiple levels of testing. It includes a complete test strategy.

Phase Test Plan

A phase test plan is a type of test plan that addresses any one phase of the testing strategy. For example, a list of tools, a list of test cases, etc.

Specific Test Plans

Specific test plan designed for major types of testing like security testing, load testing, performance testing, etc. In other words, a specific test plan designed for non- functional testing.

5.2.1 Unit Testing

Each unit or individual component of the software application is tested as part of the unit testing process. It represents the initial stage of functional testing. The purpose of unit testing is to confirm the functionality of individual unit components. A unit is a single testable component that may be tested as part of the application software development process. Unit testing is used to ensure that isolated code is correct. An specific application function or piece of code is referred to as a unit component. Unit testing is typically conducted using the white box testing methodology by developers. When the programme is finished and submitted to the test engineer, the latter will begin individually or one-by-one testing each component of each module or section of the application. This procedure is referred to as unit testing..

5.2.2 Integration Testing

After unit testing, the software testing process moves on to integration testing. Units or individual software components are tested collectively during this testing. The goal of the integration testing level is to identify flaws when integrated components or units interact. Modules are used in unit testing for testing purposes, and integration testing combines and tests these modules. The Software is created using a variety of software modules that were created by various programmers or coders. Integrity testing is done to ensure that all of the modules are communicating properly.

5.2.3 Validation Testing or System Testing

Testing a fully integrated software system is part of system testing. Typically, software is integrated into the creation of a computer system (any software is only a single element of a computer system). To create a comprehensive computer system, the software is developed in modules and then interfaced with hardware and other applications. To put it another way, a computer system consists of a collection of software that can carry out a variety of functions, but only software can do so since it needs to communicate with appropriate hardware. System testing is a collection of several types of tests designed to put an integrated software computer system through its paces and check it against requirements. Testing that included both functional and non-functional testing is known as validation testing. Here, user acceptance testing is included under non-functional testing, while functional testing is comprised of unit testing, integration testing, and system testing (UAT).

5.2.4 Output Testing or User Acceptance Testing

User acceptability testing is a testing process where customers or end users test the product to ensure that it satisfies their needs. It takes place at the client's site or the developer's location. User acceptance testing also includes operational acceptance testing and contract and regulatory compliance testing for industries like medicine and aviation. UAT is context-specific, and the UAT plans are created based on the requirements rather than being required to carry out all types of user acceptance tests. The testing team may even coordinate and contribute to the UAT process. User acceptance testing (UAT), also known as application testing or end-user testing, is a stage of the software development process when the target user group tests the product in the real world.

5.2.5 Automation Testing

Software and other computer goods are tested automatically to make sure they abide by tight guidelines. In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that might occur throughout the creation of the product. Although some testing methods, such functional or regression testing, can be carried out manually, doing so has less advantages. Any time of day can be used to do automation testing. It looks at the software using scripted sequences. It then summarises what was discovered, and this data can be compared to results from earlier test runs. C#, JavaScript,

and Ruby are the most popular programming languages used by automation developers.

5.2.6 Selenium Testing

Every web application developers should be well-versed in Selenium, an open-source, automated testing tool. Selenium automated testing is the term used to describe testing carried out with Selenium. Selenium, however, is a group of tools that each serve a specific purpose for Selenium automated testing. You will learn everything there is to know about Selenium and the numerous Selenium automation testing tools.

Test Case 1 -Farmer

code

```
import org.openqa.selenium.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;


public class App {
    Run | Debug
    public static void main(String[] args) throws Exception {
        System.setProperty(key:"webdriver.chrome.driver", value:"D:/testingnew/chromedriver.exe");
        ChromeOptions co = new ChromeOptions();
        co.addArguments("--remote-allow-origins=*");
        WebDriver driver = new ChromeDriver(co);


        driver.get("http://localhost:3000/login");
        driver.findElement(By.id("emailLogin")).sendKeys("jubin@test.com");
        driver.findElement(By.id("passLogin")).sendKeys("jubin@CR7");
        driver.findElement(By.id("submitLogin")).click();



        System.out.println(x:"Test Passed");
```

Screenshot

Test report

Test Case 1

Project Name: Lease_It					
Landowner Login Test Case					
Test Case ID: Test_1			**Test Designed By: Jubin M Varughese**		
Test Priority(Low/Medium/High):			**Test Designed Date: 3/04/2023**		
Module Name: Login			**Test Executed By : Mr. G S Ajith**		
Test Title : Login test			**Test Execution Date: 3/04/2023**		
Description: Test the login module					
Pre-Condition : Landowner has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to login page		Landowner should be able to login.	Landowner navigated to landowner dashboard page after successful login	Pass
2	Provide valid email id	Email- “jubin@test.com”			
3	Provide valid password	Password-“ jubin@CR7”			
4	Click on login button				
Post-Condition: Landowner navigated to landowner dashboard page					

Test Case 2: Farmer –Update Profile

Code

```
import org.openqa.selenium.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;


public class App {
    Run|Debug
    public static void main(String[] args) throws Exception {
        System.setProperty(key:"webdriver.chrome.driver", value:"D:/testingnew/chromedriver.exe");
        ChromeOptions co = new ChromeOptions();
        co.addArguments("--remote-allow-origins=*");
        WebDriver driver = new ChromeDriver(co);


        driver.get("http://localhost:3000/login");
        driver.findElement(By.id("emaillogin")).sendKeys("raman@gmail.com");
        driver.findElement(By.id("passlogin")).sendKeys("raman@123");
        driver.findElement(By.id("submitlogin")).click();
        driver.get("http://localhost:3000/customer/profile");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div[1]/div/article/div/div/button")).click();
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div[1]/div/article/div/fieldset/div[2]/div[2]/div/div/input")).sendKeys("1111111111");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div[1]/div/article/div/fieldset/div[4]/div[2]/div/div/input")).sendKeys("testadd");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div[1]/div/article/div/fieldset/div[5]/div[2]/div/div/input")).sendKeys("test");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div[1]/div/article/div/fieldset/div[6]/div[2]/div/div/input")).sendKeys("test");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div[1]/div/article/div/fieldset/div[7]/div[2]/div/div/input")).sendKeys("12345");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div[1]/div/article/div/div/button")).click();


        System.out.println("Test Passed");
```

Screenshot

```
PS D:\testingnew\hi>  d; cd "d:\testingnew\hi" ; & "C:\Program Files\Java\jdk-16.0.1\bin\java.exe" @C:\Users\XBOOM-1\AppData\Local\Temp\cp_205\vrhvmqzwvlempiaii2.argfile
App
Starting ChromeDriver 111.0.5563.64 (c710e91d5b6c0b7095afe0c2c17df344000784394d-refs/branch-heads/5563@{#995}) on port 38841
Only local connections are allowed.
chromeDriver was started successfully.
Apr 04, 2023 3:43:37 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 111
Apr 04, 2023 3:43:37 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to 'org.seleniumhq.selenium:s
elenium-devtools-v86:4.5.3' where the version ("v86") matches the version of the chromium-based browser you're using and the version number of the artifact is the same as Sele
nium's.
Test Passed
```

Test report

Test Case 2					
Project Name: Lease_It					
Update Profile Test Case					
Test Case ID: Test_2			**Test Designed By: Jubin M Varughese**		
Test Priority(Low/Medium/High):			**Test Designed Date: 3/04/2023**		
Module Name: Update profile			**Test Executed By : Mr. G S Ajith**		
Test Title : Update profile			**Test Execution Date: 3/04/2023**		
Description: Update profile login module					
Pre-Condition : User should update his profile successfully.					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to login page		User should be able to update his profile successfully.	User should update his profile successfully.	Pass

2	Provide valid email id	Email- “raman@test.com”			
3	Provide valid password	Password- “raman@CR7”			
4	Click on login button				
5	Navigate to profile page				
6	Click on edit button				
7	Provide valid phone number	phone number- 9563258996			
8	Provide valid house name	house name- Vazhakala			
9	Provide valid street name	street name- ramapuram			
10	Provide valid city name	city name- Thiruvalla			
11	Provide valid pincode	Pincode- 689588			
Post-Condition: User update his profile successfully					

Test Case 3 - Farmer

Code

```
import org.openqa.selenium.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;

public class App {
    Run|Debug
    public static void main(String[] args) throws Exception {
        System.setProperty(key:"webdriver.chrome.driver", value:"D:/testingnew/chromedriver.exe");
        ChromeOptions co = new ChromeOptions();
        co.addArguments("--remote-allow-origins=*");
        WebDriver driver = new ChromeDriver(co);


        driver.get("http://localhost:3000/login");
        driver.findElement(By.id("emailLogin")).sendKeys("jubin@test.com");
        driver.findElement(By.id("passLogin")).sendKeys("jubin@123");
        driver.findElement(By.id("submitLogin")).click();
        driver.get("http://localhost:3000/shop/products/add");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[1]/div/input")).sendKeys("pathanamthitta");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[2]/div/div/select")).sendKeys("agriland");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[3]/div/label/span/span[2]")).sendKeys("pexels-math-21393.jpg");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[4]/div/input")).sendKeys("5");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[5]/div/input")).sendKeys("4000");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[6]/div/input")).sendKeys("Rubber");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[7]/div/input")).sendKeys("For plantations");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[8]/div/input")).sendKeys("110/12");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[9]/div/input")).sendKeys("400000");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[10]/div/input")).sendKeys("15000");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[11]/div/input")).sendKeys("2");

        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[12]/div/button")).click();


        System.out.println("Test Passed");
```

Screenshot

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\testingnew\hi> & "C:\Program Files\Java\jdk-19.0.1\bin\java.exe" "@C:\Users\XXXMM\AppData\Local\Temp\cp_mmf3rhmmg8m4lmmpxxcl.argfile" App
Starting ChromeDriver 111.0.5563.64 (c71b0343b8c3b7005afb8c2c12df3488007b43b4-refs/branch-heads/5563@{#995}) on port 9563
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Apr 04, 2023 7:49:00 PM org.openqa.selenium.devtools.CdpVersionFinder FindNearestMatch
WARNING: Unable to find CDP implementation matching 111
Apr 04, 2023 7:49:00 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to `org.seleniumhq.selenium:selenium-devtools-v86:4.5.1`
  where the version ("v86") matches the version of the chromium-based browser you're using and the version number of the artifact is the same as Selenium's.
Test Passed
PS D:\testingnew\hi>
```

Test report

Test Case 3	
Project Name: Lease_It	
Add New Land Test Case	
Test Case ID: Test_3	**Test Designed By: Jubin M Varughese**
Test Priority(Low/Medium/High):	**Test Designed Date: 3/04/2023**
Module Name: Add Land	**Test Executed By : Mr. G S Ajith**
Test Title : Add new land test	**Test Execution Date: 3/04/2023**
Description: Test the Add land module	

Pre-Condition : User should add new land					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to login page		User should add new land successfully	User should add new land successfully	Pass
2	Provide valid email id	Email-“ jubin@test.com”			
3	Provide valid password	Password-“ jubin@CR7”			
4	Click on login button				
5	Navigate to Add land page				
6	Provide valid Location name	pathanamathitta			
7	Select valid category	agriland			
8	Provide valid extension of land	5			
9	Provide valid land image	Pexels-math-21393.jpg			
10	Provide valid price per Acer	60000			
11	Provide valid owner name	rohan			
12	Provide valid description	For plantations			
13	Provide valid survey number	110/12			
14	Provide valid price	300000			
15	Provide advance amount	15000			

16	Provide lease period	2			
17	Click on submit button				
Post-Condition: User should add new land successfully.					

Test Case 4 – Admin- Add land category

Code

```
import org.openqa.selenium.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;


public class App {
    Run|Debug
    public static void main(String[] args) throws Exception {
        System.setProperty(key:"webdriver.chrome.driver", value:"D:/testingnew/chromedriver.exe");
        ChromeOptions co = new ChromeOptions();
        co.addArguments("--remote-allow-origins=*");
        WebDriver driver = new ChromeDriver(co);


        driver.get("http://localhost:3000/login");
        driver.findElement(By.id("emaillogin")).sendKeys("admin@test.com");
        driver.findElement(By.id("passlogin")).sendKeys("Password@123");
        driver.findElement(By.id("submitlogin")).click();
        driver.get("http://localhost:3000/admin/products/category");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[1]/div/input")).sendKeys("small farms");
        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[2]/div/input")).sendKeys("for agri activities");


        driver.findElement(By.xpath("//*[@id='wrapper']/section/div/section/div/div/div/form/div[3]/div/button")).click();


        System.out.println(x:"Test Passed");
```

Screenshot

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\testingnew\App> & "C:\Program Files\Java\jdk-20.0.1\bin\java.exe" @C:\Users\HP\AppData\Local\Temp\cp_7ff27e7ntg3wie0fqxxzi.argfile App
Starting ChromeDriver 111.0.5563.64 (c7b0412d8b1b7d05a7e0c2c13f3d0000f8134d-refs/branch-heads/5563@{#995}) on port 5088
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-consideration for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Apr 04, 2023 8:06:36 PM org.openqa.selenium.devtools.CdpVersionFinder findBestMatch
WARNING: Unable to find CDP implementation matching 111
Apr 04, 2023 8:06:36 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific version of the CDP using something similar to 'org.seleniumhq.selenium:selenium-devtools-v86:4.5.3'
  where the version ("v86") matches the version of the chromium-based browser you're using and the version number of the artifact is the same as Selenium's.
Test Passed
PS D:\testingnew\App>
```

Test Report

Test Case 4

Project Name: Lease_It					
Admin Add new land category Test Case					
Test Case ID: Test_4			**Test Designed By: Jubin M Varughese**		
Test Priority(Low/Medium/High):			**Test Designed Date: 3/04/2023**		
Module Name: Add new land category			**Test Executed By : Mr. G S Ajith-**		
Test Title : Add new land category test			**Test Execution Date: 3/04/2023**		
Description: Add new land category module					
Pre-Condition :Amin should add new land category					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to login page		Admin should add new land category successfully	Admin should add new land category successfully	Pass
2	Provide valid email id	Email- “admin@test.com”			
3	Provide valid password	Password- “Password@123”			
4	Click on login button				
5	Navigate to Add land category page				
6	Provide new category name	Small farms			
7	Provide description of the category	For agri activities			
8	Click on submit button				
Post-Condition: Admin should add new land category successfully.					

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The phase of a project where goals and plans are realised is called project implementation (or project execution). This is the logical outcome after the evaluation, decision-making, visioning, planning, funding application, and identification of a project's financial resources. The project's implementation phase is where the conceptual design is transformed into a functional system. It can be regarded as the most important stage in creating a successful new system since it gives users assurance that the system will operate as intended and be reliable and accurate. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result. Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organisational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. It is crucial to remember that regardless of the project's nature, implementation requires time, often more than is anticipated, and that several external limitations may manifest. These factors should be taken into account when starting the implementation process.

6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People have early reservations about the programme, but it's important to prevent resistance from growing because the active user needs to be aware of the advantages of utilising the new system. Their faith in the software is increased. The user is given the appropriate instruction so that he feels comfortable using the application. Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

6.2.2 System Maintenance

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Errors."

6.2.3 Training on the Application Software

The user will need to receive the essential basic training on computer awareness after which the new application software will need to be taught to them. This will explain the fundamental principles of how to use the new system, including how the screens work, what kind of help is displayed on them, what kinds of errors are made while entering data, how each entry is validated, and how to change the date that was entered. Then, while imparting the program's training on the application, it should cover the information required by the particular user or group to operate the system or a certain component of the system. Depending on the user group and hierarchical level, this training could be different.

6.2.4 Hosting

The online service of web hosting makes the content of your website available to internet users. When you buy a hosting package, you are renting space on a real server to keep all the files and information for the website. Web hosts offer the resources and hosting technology needed for your website to run efficiently and securely. They are in charge of maintaining the server's functionality, putting security measures in place, and making sure that information like texts, pictures, and other files are correctly delivered to the visitors' browsers.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

.

The project entitled “Lease_It” have been developed to help people in different parts of India to lease various kinds of agricultural lands in different states. It brings different farmers across the country to lease different types of agricultural land for agricultural purposes. The performance of the system is provided to be efficient and can meet all the requirements of the user. Thus providing a useful website to lease land directly from the landowners.

7.2 FUTURE SCOPE

The project will eventually have a very broad scope. The project is quite expandable, with features like,

- Farmers can lease lands near to them.
- Recommends suitable crops for the leased lands and so on.

So, it may be upgraded as and when the need for materializes in the near future.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- https://www.sciencedirect.com/journal/agricultural-systems
- https://www.frontiersin.org/articles/10.3389/fsufs.2022.777816/full
- https://www.99acres.com/agricultural-land-for-rent-in-kerala-ff
- https://landforgood.org/lesson/leasing-farmlan

WEBSITES:

- https://sfarmsindia.com
- https://www.sothebysrealty.com
- https://www.india.gov.in/topics/agriculture
- https://farmer.gov.in

CHAPTER 9

APPENDIX

9.1 Sample Code

Home page code:

```
<template>
 <div id="wrapper">
 <nav class="navbar is-dark">
 <div class="navbar-brand">
 <router-link to="/" class="navbar-item" id="logoText"><strong>Lease IT</strong></router-link >
       <a
         class="navbar-burger"
         aria-label="menu"
         aria-expanded="false"
         data-target="navbar-menu"
         @click="showMobileMenu = !showMobileMenu">
         <span aria-hidden="true"></span>
         <span aria-hidden="true"></span>
         <span aria-hidden="true"></span>
       </a>
       </div>
      <div
       class="navbar-menu"
       id="navbar-menu"
       v-bind:class="{ 'is-active': showMobileMenu }>
       <div class="navbar-end mx-auto">
    <div class="navbar-item">
       <div class="select is-success">
       </div>
           <form method="get" action="/search">
             <div class="field has-addons">
               <div class="control searchBox">
                 <input
                   type="text"
                   class="input"
                   placeholder="What are you looking for?"
                   name="query"   />
               </div>

               <div class="control">
                 <button class="button is-success">
                   <span class="icon">
                     <i class="fa fa-search" aria-hidden="true"></i>
                   </span>
                 </button>
               </div>
             </div>
           </form>
         </div>
       </div>
       <div class="navbar-end ml-0">
```

```
<div class="navbar-item">
 <div class="buttons">
   <template v-if="$store.state.isAuthenticated">
     <router-link
       to="/admin"
       class="button is-light"
       v-if="$store.state.userRole == 'ADMIN'"
       >My account</router-link
     >

     <router-link
       class="button is-light"
       v-if="$store.state.userRole == 'SHOP'"
       to="/shop"
       >My account</router-link
     >

     <router-link
       class="button is-light"
       v-if="$store.state.userRole == 'CUSTOMER'"
       to="/customer"
       >My account</router-link
     >

     <button @click="logout()" class="button is-danger" id="logoutButton">
       Log out
     </button>
   </template>

   <template v-else>
     <router-link to="/" class="button is-light"
       >Home </router-link>
       <router-link to="/About" class="button is-light"
       >About</router-link>
       <router-link to="/Guide" class="button is-light"
       >Guide</router-link>
     <router-link to="/login" class="button is-light"
       >Log in</router-link
     >
     <router-link to="/signup" class="button is-light"
       >Sign up</router-link
     >
   </template>

   <!-- <router-link to="/cart" class="button is-success" id="cartButton"
     v-if="$store.state.userRole !== 'SHOP' && $store.state.userRole !== 'ADMIN'"
   >
      <span class="icon"><i class="fas fa-shopping-cart"></i></span>
     <span>Cart ({{ cartTotalLength }})</span>
   </router-link>  -->

   <template v-if="$store.state.isAuthenticated">
     <div class="profile-dp-div">
```

```
  <img

 class="profile-dp"
    src="http://www.thegurughantal.com/uploads/7/5/8/2/75825867/delhinightclubs-5-
bwxyimsnzqm_orig.jpg"
                />
              </div>
            </template>
          </div>
        </div>
      </div>
    </div>
  </nav>
  <section class="gunda">
    <router-view />
  </section>

  <!-- <footer class="footer">
    <p class="has-text-centered">Lease IT</p>
  </footer> -->
  </div>
</template>

<script>
import axios from "axios";

/* eslint-disable no-undef */
// import { computed, ref, onMounted, onUnmounted, watch } from 'vue'
// import { useGeolocation } from './useGeoLocation'

export default {
  data() {
    return {
      showMobileMenu: false,
      cart: {
        items: [],
      },
      userLoc: {
        city: "",
        region: "",
        country: "",
        latitude: "",
        longitude: "",
      },
      categories: [],
      selectedCategory: "Category",
    };
  },

  beforeCreate() {
    this.$store.commit("initializeStore");

    const token = this.$store.state.token;
```

```
  if (token) {

 axios.defaults.headers.common["Authorization"] = "Bearer " + token;
    } else {
      axios.defaults.headers.common["Authorization"] = "";
    }
  },
  mounted() {
    this.cart = this.$store.state.cart;
    this.getGeolocationInformation();
    this.getCategories();

    (function(d, m){
        var kommunicateSettings =
{"appId":"a70a75e9b0024f5a8444826e9c43cc5c","popupWidget":true,"automaticChatOpenOnNavigation":t
rue};
        var s = document.createElement("script"); s.type = "text/javascript"; s.async = true;
        s.src = "https://widget.kommunicate.io/v2/kommunicate.app";
        var h = document.getElementsByTagName("head")[0]; h.appendChild(s);
        window.kommunicate = m; m._globals = kommunicateSettings;
      })(document, window.kommunicate || {});
    },
  computed: {
    cartTotalLength() {
      let totalLength = 0;

      for (let i = 0; i < this.cart.items.length; i++) {
        totalLength += this.cart.items[i].quantity;
      }

      return totalLength;
    },
  },

  methods: {

    getCategories() {
      axios.get(`/user/shop/product/category`).then((response) => {
        console.log(response);
        this.categories = response.data;
      });
    },
    getProductsByCategory(item) {
      this.$router.push(`/category?category=${item.category}`)
    },
    logout() {
      axios.defaults.headers.common["Authorization"] = "";

      localStorage.removeItem("token");
      localStorage.removeItem("username");
      localStorage.removeItem("userid");

     this.$store.commit("removeToken");
```

```
this.$store.commit("setUserRole", "");

      this.$router.push("/");
    },

    async getGeolocationInformation() {
      // const API_KEY = "ab5a99eb2c834bd5846f191401c2cfab";
      const API_KEY = "a081e277312e4671a6826d28a8496cb6";
      const API_URL =
        "https://ipgeolocation.abstractapi.com/v1/?api_key=" + API_KEY;
      const apiResponse = await fetch(API_URL);
      const data = await apiResponse.json();
      const { city, country, region, latitude, longitude } = data;
      this.city = city;
      this.region = region;
      this.country = country;
      this.latitude = latitude;
      this.longitude = longitude;
      console.log(data);
      console.log(
        "City: " +
          this.city +
          " Latitude: " +
          this.latitude +
          " Longitude: " +
          this.longitude
      );
    },
  },
};

</script>

<style lang="scss">
@import "../node_modules/bulma";

.profile-dp {
  width: max-width;
  height: max-width;
  border-radius: 50%;
}

.searchBox {
  width: 500px;
}
.gunda{
  display: flex;
  align-items: center;
   justify-content: center;

height: calc(100vh - 65px);
    background-image: url("dd5.jpg");
   background-position: 70% 30%;
    background-size: cover;
```

```
}
.footer-slot {
  background: #991c1c;
}
</style>
```

Login page code:

```
<template>
  <div class="page-log-in">
    <div class="columns">
      <div class="column is-4 is-offset-4">
        <h1 class="title">Log in</h1>

        <form @submit.prevent="submitForm" class="pakaran">
          <div class="field">
            <label>Email</label>
            <div class="control">
              <input type="text" class="input" v-model="state.email" id="emailLogin" />
              <span v-if="v$.email.$error" class="has-text-danger">
                {{ v$.email.$errors[0].$message }}
              </span>
            </div>
          </div>

          <div class="field">
            <label>Password</label>
            <div class="control">
              <input type="password" class="input" v-model="state.password" id="passLogin" />
              <span v-if="v$.password.$error" class="has-text-danger">
                {{ v$.password.$errors[0].$message }}
              </span>
            </div>
          </div>
          <vue-recaptcha sitekey="6LcTVn0kAAAAAPisr9_hgsq_wAYnAHXmZhPwgUjQ" ></vue-
recaptcha>
          <div class="field">
            <div class="control">
              <button class="button is-dark" id="submitLogin">Log in</button>
            </div>
          </div>
        <div>
         <router-link to="/ForgotPassword">  Forgot Password</router-link>
        </div>



   Don't have an Account? <router-link to="/signup">Sign Up</router-link>
```

```
  </form>
      </div>
    </div>
  </div>
</template>

<script>
import axios from "axios";

import { toast } from "bulma-toast";

import useValidate from "@vuelidate/core";
import {
  required,
  email,
  sameAs,
  minLength,
  maxLength,
  helpers,
} from "@vuelidate/validators";
import { reactive, computed } from "vue";
import { VueRecaptcha } from 'vue-recaptcha';
export default {
  name: "LogIn",
  components: { VueRecaptcha },
  setup() {
    const state = reactive({
      email: "",
      password: "",
    });
    const rules = computed(() => {
      return {
        email: { required, email },
        password: {
          required,
          minLength: minLength(8),
          maxLength: maxLength(131),
          containsPasswordRequirement: helpers.withMessage(
            () =>
              `The password requires an uppercase, lowercase, number and special character`,
            (value) =>
              /(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#\$%\^&\*])/.test(value)
          ),
        },
      };
    });
    const v$ = useValidate(rules, state);
```

```
 return {
     state,
     v$,
   };
 },

 mounted() {
   document.title = "Log In | Lease It";

   (function (d, m) {
     var s = document.createElement("script");
     s.type = "text/javascript";
     s.async = true;
     var h = document.getElementsByTagName("head")[0];
     h.appendChild(s);

   }) ()

 },
 methods: {
   async submitForm() {
     this.v$.$validate();

     if (!this.v$.$error) {
       axios.defaults.headers.common["Authorization"] = "";
       localStorage.removeItem("token");
       const formData = {
         email: this.state.email,
         password: this.state.password,
       };
       this.$store.commit("setIsLoading", true);
       await axios
         .post("http://localhost:8080/api/users/login", formData)
         .then((response) => {
           const token = response.data.token;
           this.$store.commit("setToken", token);
           axios.defaults.headers.common["Authorization"] = "Bearer " + token;
           localStorage.setItem("token", token);
           var base64Url = token.split(".")[1];
           var base64 = base64Url.replace(/-/g, "+").replace(/_/g, "/");
           var jsonPayload = decodeURIComponent(
             window
               .atob(base64)
               .split("")
               .map(function(c) {
                 return "%" + ("00" + c.charCodeAt(0).toString(16)).slice(-2);
               })
```

```
    .join("")
          );
          let decoded = JSON.parse(jsonPayload);
          console.log('decoded', decoded)
          let decodedRole = decoded.role;
          let decodedId = decoded.userId

          this.$store.commit("setUserRole", decodedRole);
          this.$store.commit("setUserId", decodedId)
          let toPath = this.$route.query.to || "/";
          if (decoded.role) {
            toPath = decoded.role === 'farmer'? '/customer': decoded.role === 'land_owner'? '/shop':
'/admin'
          } else {
            toPath = this.$route.query.to || "/";
          }


          this.$router.push(toPath);
        })
        .catch((error) => {
          toast({
            message: "Not Logged In, Try again",
            type: "is-success",
            dismissible: true,
            pauseOnHover: true,
            duration: 2000,
            position: "bottom-right",
          });
        });

        this.$store.commit("setIsLoading", false);
      }
    },
  },
};
</script>
<style>
.pakaran{
color: rgb(43, 13, 239);
}
</style>
```

Registration page code

```
<template>
  <div class="page-sign-up">
    <div class="columns">
      <div class="column is-4 is-offset-4">
```

```
<h1 class="title">Sign up</h1>

    <form @submit.prevent="submitForm">
      <div class="field">
        <label class="is-size-5">Full Name</label>
        <div class="control">
          <input type="text" class="input" v-model="state.fullName" />
          <span v-if="v$.fullName.$error" class="has-text-danger">
            {{ v$.fullName.$errors[0].$message }}
          </span>
        </div>
      </div>

      <div class="field">
        <label class="is-size-5">Email</label>
        <div class="control">
          <input type="text" class="input" v-model="state.email" />
          <span v-if="v$.email.$error" class="has-text-danger">
            {{ v$.email.$errors[0].$message }}
          </span>
        </div>
      </div>

      <div class="field">
        <label class="is-size-5">Password</label>
        <div class="control">
          <input type="password" class="input" v-model="state.password" />
          <span v-if="v$.password.$error" class="has-text-danger">
            {{ v$.password.$errors[0].$message }}
          </span>
        </div>
      </div>

      <div class="field">
        <label class="is-size-5">Confirm password</label>
        <div class="control">
          <input type="password" class="input" v-model="state.password2" />
          <span v-if="v$.password2.$error" class="has-text-danger">
            {{ v$.password2.$errors[0].$message }}
          </span>
        </div>
      </div>

      <div class="field">
        <label class="is-size-5">Role</label>
        <div class="control">
          <div class="select">
```

```
 <select class="is-hovered" v-model="state.role">
  <option selected value="farmer">Farmer</option>
 <option value="land_owner">Landowner</option>
              </select>
              <span v-if="v$.role.$error" class="has-text-danger">
                {{ v$.role.$errors[0].$message }}
              </span>
            </div>
          </div>
        </div>
        <div class="field mt-6">
          <div class="control">
            <button class="button is-dark">Sign up</button>
          </div>
        </div>
        Already have an Account? <router-link to="/login">Log in</router-link>
      </form>
    </div>
  </div>
 </div>
</template>

<script>
import axios from "axios";
import { toast } from "bulma-toast";
// import { toast } from "@import 'bulma/css/bulma.css'"

import useVuelidate from "@vuelidate/core";
import {
  required,
  email,
  sameAs,
  minLength,
  maxLength,
  helpers,

} from "@vuelidate/validators";
import { reactive, computed } from "vue";

export default {
  name: "SignUp",
  setup() {
    const state = reactive({
      fullName: "",
      email: "",
      password: "",
      password2: "",
```

```
role: "",
  });

const rules = computed(() => {
    return {
      fullName: { required },
      email: { required, email },
      password: {
        required,
        minLength: minLength(8),
        maxLength: maxLength(131),
        containsPasswordRequirement: helpers.withMessage(
          () =>
            `The password requires an uppercase, lowercase, number and special character`,
          (value) =>
            /(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#\$%\^&\*])/.test(value)
        ),
      },
      password2: { required, sameAs: sameAs(state.password) },
      role: { required },
    };
  });

  const v$ = useVuelidate(rules, state);

  return {
    state,
    v$,
  };
},

mounted() {
  document.title = "Sign Up | CloudStore";
},
methods: {
  async submitForm() {
    this.v$.$validate();

    if (!this.v$.$error) {
      const formData = {
        fullName: this.state.fullName,
        email: this.state.email,
        password: this.state.password,
        role: this.state.role,
      };

      this.$store.commit("setIsLoading", true);
```

```
 await axios({
            method: "post",
            url: "http://localhost:8080/api/users",
            data: formData,
          })
            .then((res) => {
              if (res.status == 200) {
                console.log("Success!!");
                this.userCreated = false;
                this.$router.push("/login");
              } else {
                throw res;
              }
            })
            .catch((err) => {
              this.userCreated = false;
              this.email_error = err.response.data.email;
            });

        this.$store.commit("setIsLoading", false);
      }
    },
  },
};
</script>
```

Post Land page code:

```
<template>
  <div class="page-log-in">
    <div class="columns">
      <div class="column is-4 is-offset-4">
        <h1 class="title">Add New Land</h1>

        <div class="modal" v-bind:class="{ 'is-active': modalActive }">
          <div class="modal-background"></div>
          <div class="modal-card">
            <header class="modal-card-head">
              <p class="modal-card-title">Similar Lands</p>
              <button
                class="delete"
                aria-label="close"
                @click="hideSimilarLands"
              ></button>
            </header>

            <section class="modal-card-body">
```

```
          <div class="columns is-multiline">
            <SimilarProductBox
              v-for="product in similarLandss"
              v-bind:key="product.id"
              v-bind:product="product"
            />
          </div>
        </section> -->
      </div>
    </div>
    <form @submit.prevent="submitForm">
      <div class="field has-addons">
        <div class="control is-expanded" >
          <input type="text" class="input" placeholder="Location Name" v-model="LandName"
required/>
        </div>
        <!-- <div class="control">
          <a class="button is-info" @click="showSimilarLandss"> Check </a>
        </div> -->
      </div>
      <div class="field">
        <div class="control">
          <div class="select" >
            <select v-model="category" Category required>
              <option value="" disabled selected hidden>Category</option>
              <option selected value="Agriland">agriland</option>
              <option value="Estatelandr">farmland</option>
            </select>
          </div>
         </div>
      </div>
      <div class="field">
        <div class="file">
          <label class="file-label">
            <input
              class="file-input"
              type="file"
              name="resume"
              @change="imageFileSelect"
              required/>
            <span class="file-cta">
              <span class="file-icon">
                <i class="fas fa-upload"></i>
              </span>
              <span class="file-label"> Upload Land Image </span>
            </span>
          </label>
```

```
        </div>
        </div>
        <div class="field">
          <div class="control">
            <input type="text" class="input" placeholder="Extension Of Land" v-model="mainUnit"
required/>
          </div>
        </div>
        <div class="field">
          <div class="control">
            <input type="text" class="input" placeholder="Price Per Acer" v-model="pprice" required
>
          </div>
        </div>
        <div class="field">
          <div class="control">
            <input type="text" class="input" placeholder="Owner Name" v-model="namess"
required/>
          </div>
        </div>
        <div class="field">
          <div class="control">
            <input type="text" class="input" placeholder="Description" v-model="saleUnit"
required/>
       </div>
        </div>
        <div class="field">
          <div class="control">
            <input type="text" class="input" placeholder="Survey Number" v-model="weight"
required/>
          </div>
        </div>

        <div class="field">

          <div class="control">
            <input type="text" class="input" placeholder="Total Price" v-model="price" required/>
          </div>
        </div>
        <div class="field">
          <div class="control">
            <input type="text" class="input" placeholder="Advance Payment Amount" v-
model="stock" required />
          </div>
        </div>
        <div class="field">
          <div class="control">
```

```
  <input type="text" class="input" placeholder="Lease Period" v-model="increment" required/>
          </div>
        </div>
         <div class="field">
          <label>User Residing Street</label>
          <div class="control">
            <input type="text" class="input" v-model="streetName" />
          </div>
        </div>

        <div class="field">
          <label>User Residing City</label>
          <div class="control">
            <input type="text" class="input" v-model="cityName" />
          </div>
        </div>
        <div class="field">
          <label>User Residing State</label>
          <div class="control">
            <input type="text" class="input" v-model="stateName" />
          </div>
        </div>
        <div class="notification is-danger" v-if="errors.length">
          <p v-for="error in errors" v-bind:key="error">{{ error }}</p>
        </div>
        <div class="field">
          <div class="control">
            <button class="button is-dark">Submit</button>
          </div>
        </div>
       </form>
     </div>
   </div>
  </div>
</template>
<script>
import axios from "axios";
import SimilarProductBox from "@/modules/Shop/components/SimilarProductBox.vue";
import { toast } from "bulma-toast";

export default {
  name: "AddShopDetails",
  data() {
    return {
      prodName: "",
      category: "",
      imageUrl: "",
```

```
  stock: "",
      namess: "",
      mainUnit: "",
      saleUnit: "",
      weight: "",
      price: "",
      increment: "",
pprice:"",
      productImage: "",
      errors: [],
      categories: [],

      modalActive: false,
      similarProducts: [],
    };

  },
  mounted() {
    this.getCategories();
  },

  components: {
    SimilarProductBox,
  },

  methods: {
    showSimilarProducts() {
      this.modalActive = true;
      axios
        .get(`/user/shop/product/similar?prodName=${this.prodName}`)
        .then((response) => {
          this.similarProducts = response.data;
        });
    },
    hideSimilarProducts() {
      this.modalActive = false;
    },

    getCategories() {
      axios.get("/user/shop/product/category").then((response) => {
        console.log(response);
        var tempData = response.data;
        // console.log(tempData[0].category)
        var responseData = response.data;
        for (let item in responseData) {
          console.log(responseData[item].category);
          this.categories.push(responseData[item]);
```

```
  }
// console.log(this.categories)
      });
    },

    imageFileSelect(event) {
      this.$store.commit("setIsLoading", true);
      console.log(event);
      var imageFile = event.target.files[0];
      this.createBase64Image(imageFile);
    },

    createBase64Image(fileObject) {
      var reader = new FileReader();
      reader.onload = (e) => {
        var imageFileData = e.target.result;
        this.productImage = imageFileData.slice(imageFileData.indexOf(",") + 1);

 console.log(this.productImage);
        this.uploadImage();
      };
      reader.readAsDataURL(fileObject);
    },

    async uploadImage() {
      var formData = new FormData();
      formData.append("image", this.productImage);

      const client = axios.create({
        transformRequest: [
          (data, headers) => {
            // add required "Content-Type" whenever body is defined
            delete headers.common.Authorization;
            return data;
          },
        ],
      });
      await client
        .post(
          "https://api.imgbb.com/1/upload?key=0f6650dbe5d582897945e5dd899204bd",
          formData
        )
        .then((response) => {
          console.log(response);
          var imageData = response.data.data;
          this.imageUrl = imageData.url;
        });
```

```
this.$store.commit("setIsLoading", false);
  },

  async submitForm() {
    this.$store.commit("setIsLoading", true);
    const formData = {
      locationName: this.LandName,
      category: this.category,
      image: this.imageUrl,
      ownername:this.namess,
      extend: this.mainUnit,
      description: this.saleUnit,
      survey_number: this.weight,
      price_per_acer: this.pprice,
      advance: this.stock,
      leaseperiod: this.increment,
    };


  await axios
      .post("http://localhost:8080/api/users/shop/product", formData)
      .then((response) => {
        toast({
          message: "New land successfully added",
          type: "is-success",
          dismissible: true,
          pauseOnHover: true,
          duration: 2000,
          position: "bottom-right",
        });

        //   const toPath = this.$route.query.to || "/cart";

        //   this.$router.push(toPath);
      })
      .catch((error) => {
        if (error.response) {
          for (const property in error.response.data) {
            this.errors.push(`${property}: ${error.response.data[property]}`);
          }
        } else {
          // this.errors.push("Something went wrong. Please try again");

          console.log(JSON.stringify(error));
        }
        this.$store.commit("setIsLoading", false);
      });
```

```
  },
  },
};
</script>
```

View Land page code:

```
<template>
   <div class="galle" v-for="(land, index) in farmlands" :key="index">

      <img :src="land.image">
      <div class="desco">Place: {{ land.locationName }}</div>
      <div class="desco">Description: {{ land.description }}</div>
      <div class="desco">Extend: {{ land.extend }} Acer</div>
      <div class="desco">Price: {{ land.price }}</div>
      <div class="desco">Survey Number: {{ land.survey_number }}</div>
      <div class="desco">Owner Name: {{ land.ownername }}</div>
      <div class="desco">Advance Amount: {{ land.advance }}</div>
      <button class="button is-dark" @click="createOrder()" style="margin-bottom: 5px; margin-left: 50px">Lease Now</button>


 <button class="button is-dark" @click="loadReactWebsite" style="margin-bottom: 5px; margin-left: 50px">Predict Crop</button>
   </div>
</template>
<script>
import axios from "axios";
import { toast } from "bulma-toast";
export default {
   name: "Addcrop",
   data() {
      return {
         farmlands: []
      };
   },

   mounted() {
      let razorPayScript = document.createElement('script')
      razorPayScript.setAttribute('src', 'https://checkout.razorpay.com/v1/checkout.js')
      document.head.appendChild(razorPayScript)
      this.fetchData()
   },
   methods: {
      loadReactWebsite() {
     window.open('http://localhost:3001/', '_blank') // Replace with your React website's URL
   },
      async createOrder() {
         try {
```

```
const response = await axios.post('http://localhost:8080/api/razor-pay/order', {
                amount: 50000,  // amount in the smallest currency unit
                currency: "INR",
                receipt: "order_rcptid_11"
            })
            this.payment(response.id)
            console.log('response', response)
        } catch (error) {

        }
    },
    async payment(orderId) {
        const a= this
        // await
        var options = {
            "key": "rzp_test_iJs9lNifWJnygM", // Enter the Key ID generated from the Dashboard
            "amount": "1000000",
            "currency": "INR",
            "name": "Acme Corp",
            "description": "Test Transaction",


"image": "https://example.com/your_logo",
            "order_id": orderId, //This is a sample Order ID. Pass the `id` obtained in the response of
            "handler": function () {
                // a.updateLandBuy()
                a.$router.push('Agreement')
                // alert(response.razorpay_payment_id);
                // alert(response.razorpay_order_id);
                // alert(response.razorpay_signature)
            },
            "prefill": {
                "name": "Gaurav Kumar",
                "email": "gaurav.kumar@example.com",
                "contact": "9000090000"
            },
            "notes": {
                "address": "Razorpay Corporate Office"
            },
            "theme": {
                "color": "#3399cc"
            }
        };
        var pay = new window.Razorpay(options);
        pay.open();

    },
```

```
 async updateLandBuy() {
          await axios
             .get("http://localhost:8080/api/users/lands/updatestatus")
             .then((response) => {
                 this.farmlands = response.data
                console.log('response', response);

                  const toPath = this.$route.query.to || "/cart";

                  this.$router.push(toPath);
            })
             .catch((error) => {
                 if (error.response) {
                     for (const property in error.response.data) {
                         this.errors.push(`${property}: ${error.response.data[property]}`);
                     }
                 } else {
                    // this.errors.push("Something went wrong. Please try again");

                     console.log(JSON.stringify(error));
                 }
             });


 },
      async fetchData() {

          await axios
             .get("http://localhost:8080/api/users/lands/get")
             .then((response) => {
                 this.farmlands = response.data
                 console.log('response', response);

                 //   const toPath = this.$route.query.to || "/cart";

                 //   this.$router.push(toPath);
             })
             .catch((error) => {
                 if (error.response) {
                     for (const property in error.response.data) {
                         this.errors.push(`${property}: ${error.response.data[property]}`);
                     }
                 } else {
                    // this.errors.push("Something went wrong. Please try again");

                     console.log(JSON.stringify(error));
                 }
```

```
},
   },

};

</script>
<style>
.galle {
   margin: 10px;
   border: 1px solid #ccc;
   float: left;
   width: 400px;
   height: 380;
}
div.desco {
   padding: 2px;
   text-align: left;
   font-weight: 600;
   color: aqua;
}
</style>
```

Agreement page code:

```
<template>
<div style="text-align: center;color:black; font-weight: 800;"> Lease Agreement<br></div>
<div id="divToPrint" class="descc" > This agreement is between _____ (landowner)
and _____,
(tenant), for the lease of certain parcels of land for the purpose of _____
_____ [describe agricultural purpose(s) and operation].<br>
1.The parcel(s) contained in this agreement are is/described as follows: [parcel location,
acreage, bounds, features, condition, etc.]<br>

2. The term of this lease shall be from _____ to _____
except as terminated earlier according to the provisions below.<br>
3. The tenant agrees to pay a lease fee to the landowner of rs_____ per acre or
rs_____ total, per year. The tenant agrees to pay such sum at the beginning of the
lease term and on the anniversary thereof unless otherwise mutually agreed. A late
penalty of up to [ ]%/month may be assessed on all late payments. This lease fee may
be renegotiated annually.<br>
4. Prohibited Uses: The tenant shall not, unless by mutual agreement to the contrary,
engage in any of the following activities on said parcel(s):<br>
5. The tenant agrees to prepare an annual management plan for review by the landlord,
complete annual soil testing, and apply amendments as indicated at his/her own
expense. The tenant agrees to proper disposal of trash and waste. The tenant further
agrees:<br>
```

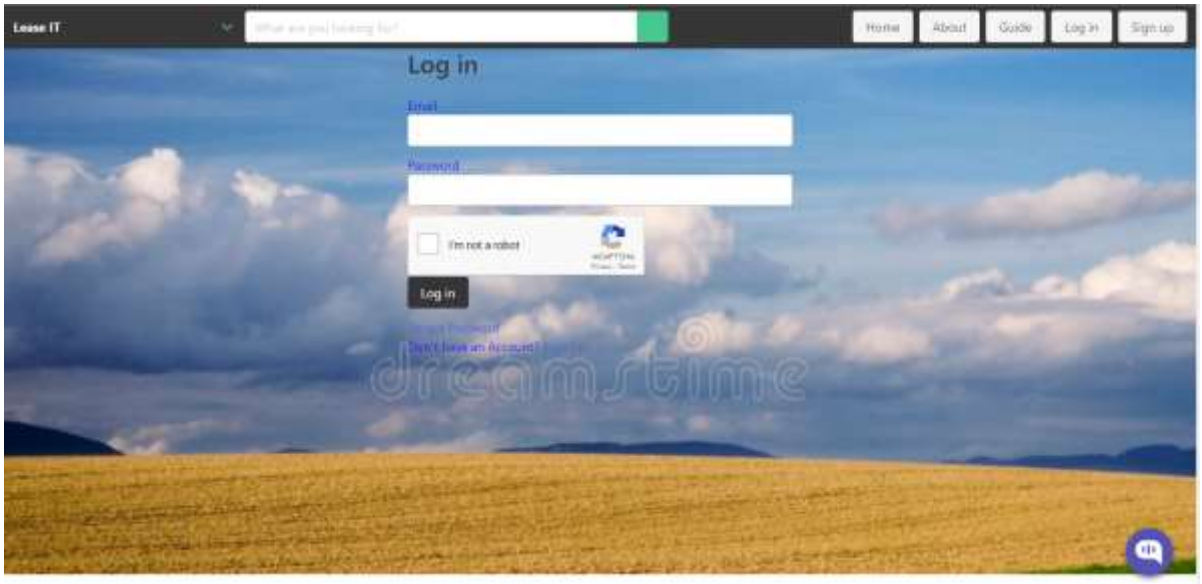
```
6. The [landowner/tenant] agrees to pay all taxes and assessments associated with this
parcel.<br>
7. The farmer agrees to provide the landowner with evidence of liability insurance
coverage.<br>
8. Either party may terminate this lease at any time with _____ month notice to the other
party. The tenant agrees not to assign or sublease his/her interest.<br>
9. The terms of this lease may be amended by mutual consent.<br>
10. A default in any of these provisions by either party may be cured upon written notice
by the other party within ____ days of receipt of such notice. Any disputes occurring
from this lease may be resolved by standard mediation practices, if necessary.<br>
11. Landowner retains his/her right to access the parcel(s) for the purposes of inspection
with prior notification to the tenant.<br>
signed:<br>
_____date_____<br>
_____ date_____ <br>
  </div>
  <br>
  <div style="float: right;"> <button   @click="printDocument()">Download</button></div>

  </template>

<script>
import pdfMake from 'pdfmake';
import pdfFonts from 'pdfmake/build/vfs_fonts';
import htmlToPdfmake from 'html-to-pdfmake';
export default {

    methods: {
      printDocument() {

          //get table html
          const pdfTable = document.getElementById('divToPrint');
          //html to pdf format
          var html = htmlToPdfmake(pdfTable.innerHTML);

          const documentDefinition = { content: html };
          pdfMake.vfs = pdfFonts.pdfMake.vfs;
          pdfMake.createPdf(documentDefinition).open();

    }
}
}
</script>
<style>
.descc {
    margin-top: 2%;
padding: 2px;
```

```
text-align: left;
color: white;
font-size: 100%;
}
 /* .button {

        display: inline-block;
        background-color: #14ee67;
        width: 100px;
        color: #ffffff;
        text-align: center;

 } */

</style>
```

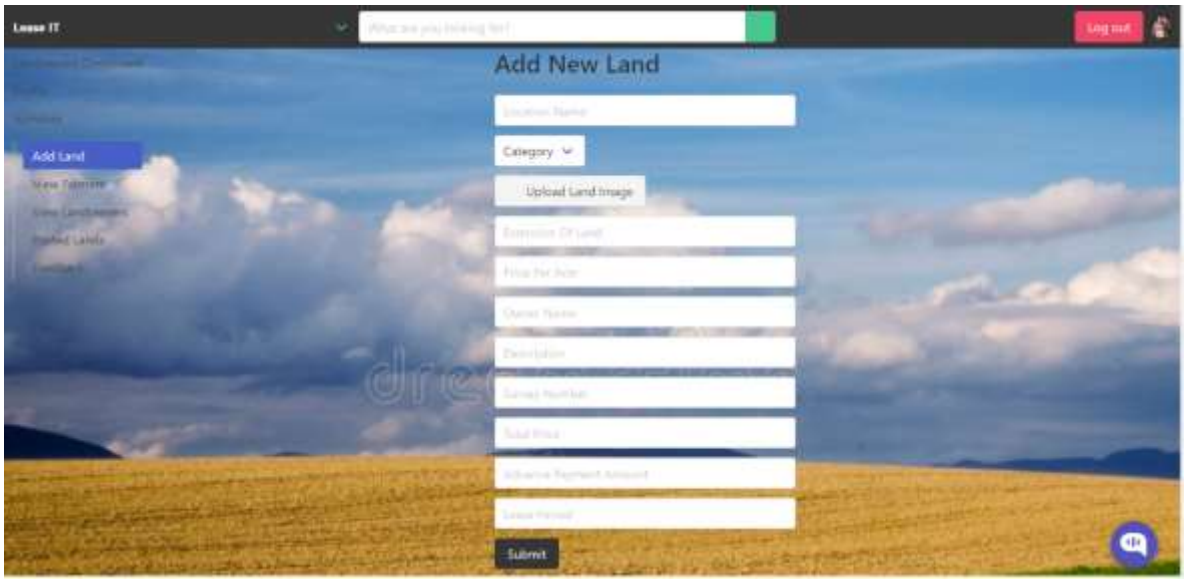
9.2 Screen Shots

Home page

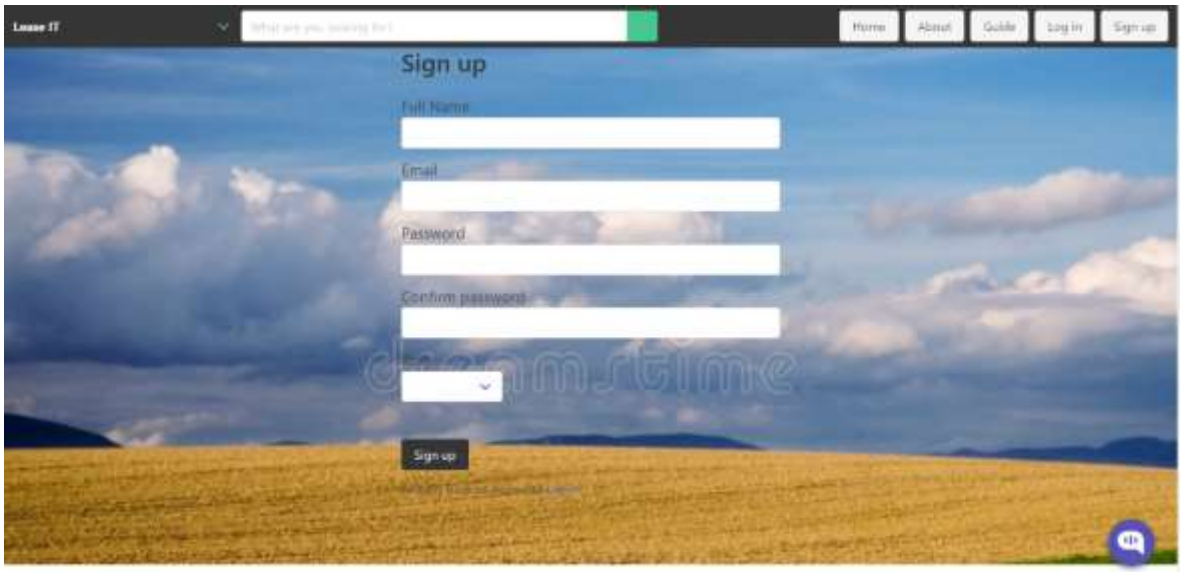
Login page



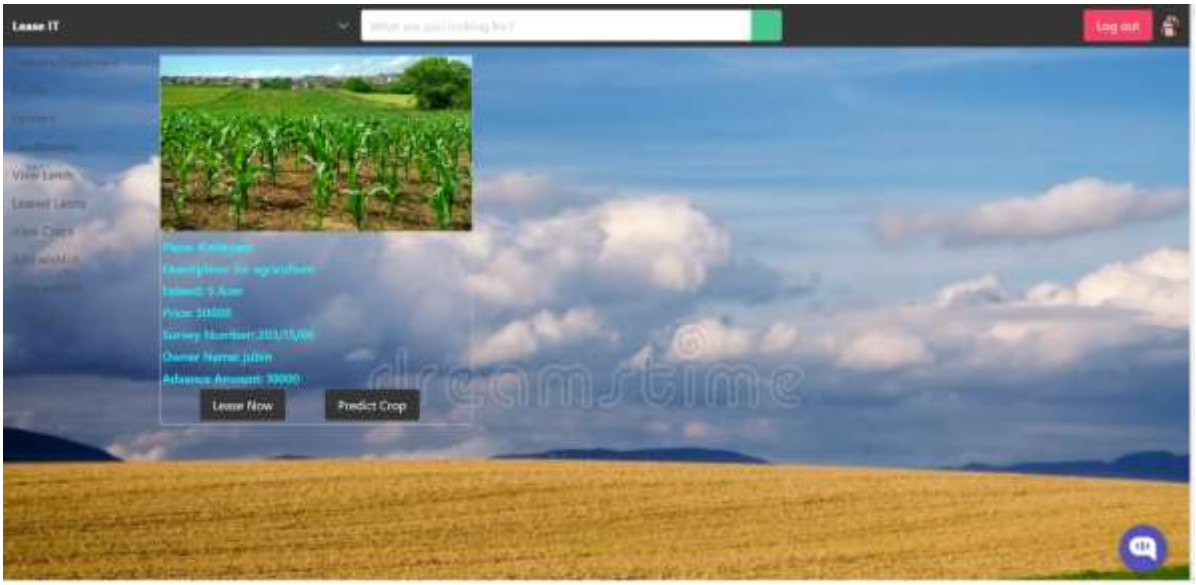
Post land page



Registration page



View Land page



Agreement page