Mini Project

on

Kisaan Suvidha

Submitted in partial fulfilment of the requirements

of the degree of

Master of Computer Applications

By

Prasad Prabodh Gade

Utsav Uttam Mandal

Prashantkumar Gupta

Ravi Kumar Rai

under the guidance of

Supervisor (s):

Name of Guide

Dr. Dhanamma Jagli

Department of Master of Computer Applications

Vivekanand Education Society's Institute of Technology

2023-2024

Since 1962

Vivekanand Education Society's

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Master of Computer Applications

CERTIFICATE

This is to certify that **Mr. Prasad Gade, Mr. Utsav Mandal, Mr. Prashantkumar Gupta, Mr. Ravi Rai** of First Year Master Of Computer Applications studying under the University of Mumbai have satisfactorily presented the Mini Project entitled Kisaan Suvidha as a part of the MINI-PROJECT for Semester-II under the guidance of **Dr. Dhanamma Jagli** in the year 2023-2024.

Date:

(Name and sign)
Head of Department

(Name and sign)
Supervisor/Guide

Since 1962

Vivekanand Education Society's

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Mini Project Approval

This Mini Project entitled " Kisaan Suvidha**"** by **Mr. Prasad Gade (14), Mr. Utsav Mandal (30), Mr. Prashantkumar Gupta(17), Mr. Ravi Rai(42)** Of the Class **FY MCA** is approved for the degree of **Master of Computer Applications.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place

Since 1962

Vivekanand Education Society's

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Master of Computer Applications

DECLARATION

We, **Mr. Prasad Gade, Mr. Utsav Mandal, Mr. Prashantkumar Gupta, Mr. Ravi Rai** from **class FYMCA DIV A** declare that this project represents our ideas in our own words without plagiarism and wherever others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our project work.

We declare that we have maintained a minimum 75% attendance, as per the college norms.

We understand that any violation of the above will be cause for disciplinary action by the Institute.

Yours Faithfully

(Name & Signature of Students with Date)

Acknowledgement

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Dr. Dhanamma Jagli** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Master of Computer Applications Department **Dr. Shivkumar Goel** and our Principal **Dr. (Mrs.) J.M. Nair** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Master of Computer Applications.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Table of Contents

Abstract

List of Tables

List of Figures

1. Introduction

1.1. Introduction

1.2. Problem Statement

1.3. Objectives of the project

1.4 Functionalities

1.5. Scope

2. Literature Survey

2.1 Survey of Existing System

2.2 Limitation Existing System or Research Gap

2.3 Mini Project Contribution

3. Analysis and Design

3.1 Analysis of the system

3.2 Proposed Solutions (ER Diagram)

3.3

3.3.1 Architecture/ FrameworkDesign of the proposed system

3.3.2 Algorithm and Process Design

3.3.3 Details of Hardware & Software

3.3.4 Experiment and Results (Code and GUI)

4. Conclusion and Future Work

5. References

Kisaan Suvidha

1. Introduction:

1.1 Introduction:

Welcome to Kisaan Suvidha, an innovative online platform designed to empower farmers and streamline agricultural processes. Our mission is to bridge the gap between farmers and the resources they need by providing a comprehensive suite of modules catering to various aspects of agriculture. In this era of technological advancement, we believe in leveraging digital solutions to enhance the efficiency and effectiveness of farming practices.

At Kisaan Suvidha, we are committed to creating a digital ecosystem that empowers farmers, promotes sustainable practices, and contributes to the overall growth of the agricultural sector. Join us on this journey towards a more connected, informed, and efficient farming community. Together, let's cultivate success and harvest a brighter future for agriculture.

1.2 Problem statement :

Agriculture, the backbone of our society, grapples with issues that hinder the growth of individual farmers and the sector as a whole. The lack of awareness about government schemes leaves farmers missing out on vital support. Additionally, the hefty price tag on modern equipment restricts adoption, and navigating the complexities of land rentals proves challenging for both landowners and farmers. Kisaan Suvidha steps in as a comprehensive solution, offering a user-friendly platform. It not only enlightens farmers about government initiatives but also enables them to share and rent equipment among themselves. Simultaneously, it simplifies the process of land rentals, fostering collaboration and efficiency. With these interventions, we aim to empower farmers, boost productivity, and contribute to the resilience of our agricultural community.

1.3 Objectives :

1. Government Scheme Promotion:
- Raise awareness about available government schemes and subsidies for farmers.

2. Land Rental Facilitation:
- Develop a platform connecting landowners with farmers for transparent and fair land rental transactions.

3. Equipment Access Improvement:
- Establish a system for farmers to easily rent modern agricultural equipment.

4. Data Collection and Analysis:
- Gather and analyze data on land rental, equipment usage, and overall impact on farmers.

5. Environmental Sustainability:
- Promote eco-friendly farming practices and equipment.

6. Measurable Impact:
- Define KPIs to measure the success of the project in terms of farmer benefits and productivity.

7. Technology Integration:

- Integrate technology solutions for accessibility and data analytics.

8. Social Impact Assessment:

- Assess the project's social impact on local communities.

9. Adaptability and Scalability:

- Design for scalability and adapt strategies based on evolving needs.

1.4 Functionalities:

1. User Registration and Authentication:

- Allow farmers, landowners, and other stakeholders to register and authenticate their accounts.

2. Dashboard:

- Provide a user-friendly dashboard displaying relevant information, updates, and notifications.

3. Government Scheme Information:

- Feature a dedicated section with detailed information on various government schemes and subsidies.

4. Land Rental Platform:

- Create a platform for landowners to list available land for rent.

- Enable farmers to search and express interest in renting specific pieces of land.

5. Equipment Rental Marketplace:

- Develop an online marketplace for farmers to browse and rent agricultural equipment.

- Include detailed descriptions, rental rates, and user reviews for each piece of equipment.

6. Transaction Management:

- Facilitate transparent and secure transactions for land and equipment rentals.

- Include features for contract generation, e-signatures, and payment processing.

7. Communication Channels:

- Establish communication channels, including messaging and notifications, to keep users informed.

- Enable farmers and landowners to communicate regarding rental agreements and equipment transactions.

8. Mobile Accessibility:

- Develop a mobile-responsive platform or a dedicated mobile app for easy access on smartphones.

9. Community Forum:

- Implement a community forum for knowledge-sharing and discussions among farmers.

10. Document Management:

- Enable users to upload and manage essential documents, such as land lease agreements and equipment rental contracts.

11. Social Media Integration:

- Integrate social media sharing features to enhance project visibility and community engagement.

12. Feedback and Rating System:

- Implement a feedback and rating system for users to share their experiences with land rental and equipment transactions.

13. Localization and Multilingual Support:

- Offer localization options and multilingual support to cater to diverse user groups.

1.5 Scope:

1. Government Scheme Awareness:

- Provide detailed information about various government schemes related to agriculture, including eligibility criteria, application processes, and benefits.

- Regularly update the content to reflect any changes in government policies or the introduction of new schemes.

- Offer resources such as application forms, contact details for relevant government offices, and guidance on how farmers can benefit from these schemes.

2. Land Rental:

- Create a platform where landowners can list their available land for rent, and farmers can search for suitable rental opportunities.

- Include filters such as location, size of the land, and rental terms to help users find the most relevant options.

- Provide a secure and transparent platform for negotiations and agreements between landowners and farmers.

- Include information on legal aspects, such as lease agreements and documentation requirements.

3. Farmers Equipment Rental:

- Build a marketplace where farmers can rent agricultural equipment from each other or from equipment owners.

- Include a diverse range of equipment, from small tools to larger machinery, with details on specifications, usage guidelines, and rental rates.

- Implement a rating and review system to help users make informed decisions about the reliability and quality of equipment.

- Provide resources on proper equipment usage, maintenance, and safety guidelines.

4. Community Building:

- Create forums or discussion boards where farmers, landowners, and other stakeholders can share experiences, ask questions, and offer advice.

- Foster a sense of community by organizing local events, meetups, or online networking opportunities.

5. Technology Integration:

- Implement technologies such as mobile apps to make it easier for users to access information, list their services, or find what they need.

- Explore the integration of data analytics to provide insights into agricultural trends, weather forecasts, and market prices.

2.Literature Survey

2.1 Survey of Existing systems:

Government Scheme Awareness

Department of Agriculture and Farmers Welfare (DA&FW): https://agricoop.gov.in/

This is the official website of the DA&FW, which is responsible for formulating and implementing policies and programs for the development of agriculture and farmers' welfare in India. The website has a dedicated section on government schemes, which provides information on various schemes for farmers, including subsidies, loans, insurance, and training.

One Stop Window-Farmers Portal (e-Krishi): https://mKisaan.gov.in/

This is a government portal that provides a single-window platform for farmers to access information and services related to agriculture. The portal includes information on government schemes, as well as other services such as weather forecasts, market prices, and agricultural advisories.

mKisaan Portal: https://mKisaan.gov.in/

This is a mobile phone-based portal that provides farmers with information on various agricultural topics, including government schemes. The portal also allows farmers to register for various government schemes and receive SMS alerts on important agricultural updates.

Land Rental

Agriland: https://www.agriland.ie/

Agriland is a private website that provides a platform for farmers to rent or lease land. The website also provides information on various government schemes related to land rental.

Farmland for Lease: https://lawprofessors.typepad.com/agriculturallaw/2020/11/farmland-lease-income-proper-tax-reporting.html

Farmland for Lease is another private website that provides a platform for farmers to rent or lease land. The website also provides information on various government schemes related to land rental.

Krishi Yantra Seva Portal: https://dbt.mpdage.org/

This is a government portal that provides farmers with information on farmers equipment rental services. The portal also allows farmers to register for rental services and receive SMS alerts on equipment availability.

Hire Tractor: https://hellotractor.com/

Hire Tractor is a private website that provides a platform for farmers to rent tractors and other agricultural equipment. The website also provides information on various government schemes related to agricultural equipment rental.

Renting Equipment for Agriculture: https://dir.indiamart.com/impcat/tractor-rental.html

This is a website that provides information on various options for renting agricultural equipment, including government-sponsored rental schemes and private rental services.

2.2 Limitations of Existing systems:

Here are some of the limitations of agriculture-related websites that have government scheme awareness, land rental, and farmer equipment rental:

Limited Accessibility:

- **Digital Divide:** The digital divide between rural and urban areas can hinder farmers' access to these websites and their ability to utilize the information provided.
- **Language Barriers:** The lack of multilingual support on some websites can exclude farmers who are not proficient in the primary language used.
- **Technological Literacy:** Some farmers may not have the necessary technological literacy to effectively navigate the websites and find the information they need.

Accuracy and Timeliness of Information:

- **Outdated Information:** Websites may not be updated regularly, leading to outdated or inaccurate information about government schemes, land rental, and farmer equipment rental.
- **Inconsistent Information:** Discrepancies in information across different websites can create confusion and uncertainty for farmers.

User Interface and Navigation:

- **Complex Navigation:** The structure and organization of some websites can be complex and difficult for farmers to navigate, making it challenging to find the specific information they need.
- **Limited Search Functionality:** The search functions on some websites may not be robust enough to effectively locate specific information or keywords related to government schemes, land rental, or farmer equipment rental.

Content and Presentation:

- **Technical Jargon:** The use of technical jargon or overly complex language can make it difficult for farmers to understand the information provided.
- **Lack of Visual Aids:** The absence of visual aids, such as infographics, diagrams, or videos, can limit the effectiveness of information dissemination and comprehension.

Sustainability and Maintenance:

- **Funding Constraints:** Sustainable funding mechanisms are crucial for ensuring that these websites are maintained, updated, and expanded to meet the evolving needs of farmers.
- **Technical Expertise:** Maintaining these websites requires ongoing technical expertise, which may not always be readily available in rural areas.

These limitations can potentially hinder farmers' ability to access and utilize the information provided on these websites, thereby affecting their awareness of government schemes and their ability to participate in land rental and farmer equipment rental programs. Addressing these limitations is essential for ensuring that these websites serve as effective tools for empowering farmers and improving agricultural practices.

2.3 Mini project contribution:

1. Increased Awareness of Government Schemes

This website will help farmers with easy access to information about various government schemes that can provide them with financial assistance, access to resources, and training opportunities. This increased awareness can lead to:

- **Increased participation in government schemes:** When farmers are aware of the schemes available to them, they are more likely to take advantage of them. This can lead to increased financial security, improved agricultural practices, and higher productivity.
- **Reduced reliance on traditional sources of information:** Traditionally, farmers have relied on word-of-mouth and local extension officers for information about government schemes. These websites provide a more reliable and up-to-date source of information, which can help to reduce the spread of misinformation.

2. Improved Access to Land and Equipment

This website will help farmers to find land for rent and equipment to rent or purchase. This can be especially beneficial for small and marginal farmers who may not have the resources to own their own land or equipment. Access to land and equipment can lead to:

- **Increased agricultural production:** When farmers have access to land and equipment, they can expand their farms and produce more crops. This can lead to increased income and food security.
- **Reduced costs:** Renting land and equipment can be a more cost-effective option than purchasing them outright. This can free up farmers' financial resources so that they can invest in other areas of their farms.

3. Analysis and Design

3.1 Analysis of the system

1. User Interface (UI) and User Experience (UX):

- A clean and intuitive interface enhances user experience.

- Ensure easy navigation between different sections like government schemes, land rental, and equipment rental.

2. Content Quality:

- Evaluate the information provided on government schemes. It should be comprehensive, up-to-date, and easily understandable for farmers.

- Assess the details on land rental, including the available lands, rental rates, and terms.

- Examined information on equipment rental, covering types of equipment available, rental costs, and any associated terms.

3. Search Functionality:

- A robust search feature can help users find relevant information quickly.

4. Government Scheme Information:

- The details of government schemes are accurate and provide eligibility criteria, application procedures, and deadlines.
- New updates or news related to agricultural policies and schemes.

5. Land Rental Section:

- Easy process for farmers to find available land for rent.
- Information on land quality, location, and any restrictions or requirements for renting.

6. Equipment Rental Section:

- Assess the availability of different types of agricultural equipment for rent.
- Look for details such as rental costs, maintenance, and any support provided.

7. Security and Privacy:

- Given the sensitive nature of farming data, ensure that the website has secure protocols in place to protect user information.

8. Mobile Responsiveness:

- Confirm that the website is mobile-friendly, considering that farmers may access it from various devices.

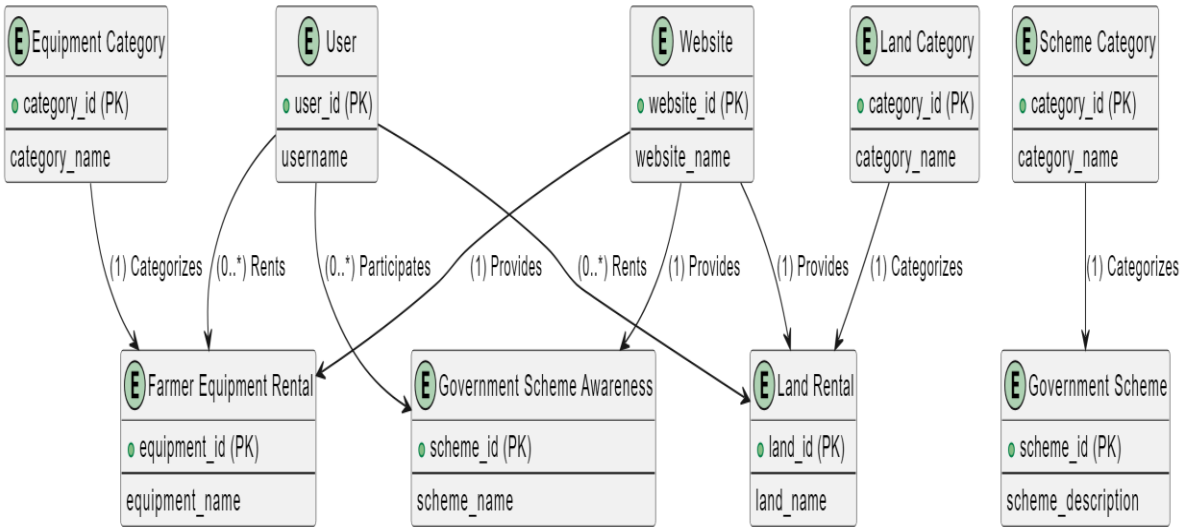
9. Contact and Support:

- Check for contact information and support options in case users have questions or issues. Responsive customer support is essential.

10. Up-to-Date Information:

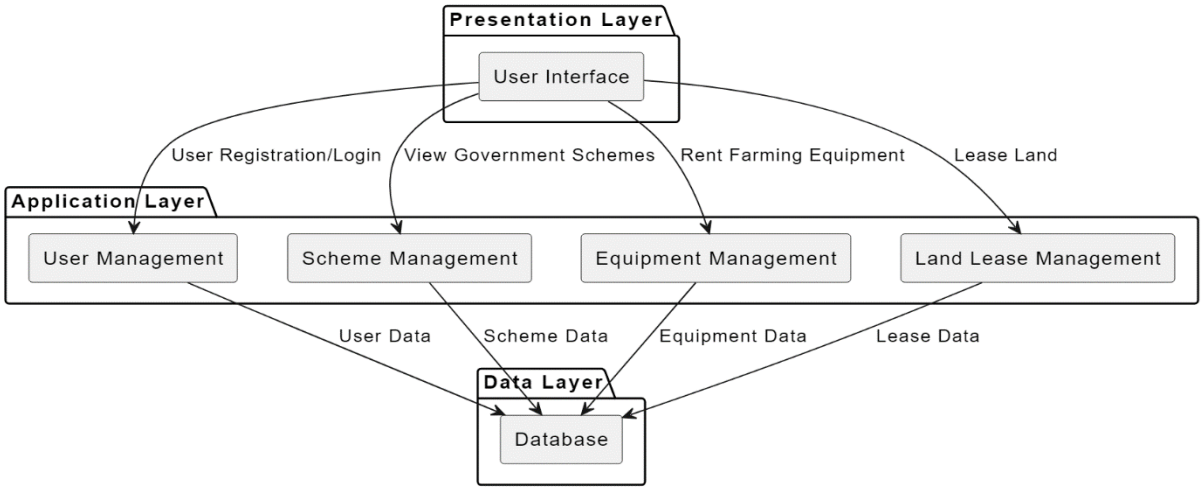
- Regularly update the content, especially regarding government schemes, to reflect any changes or new policies.

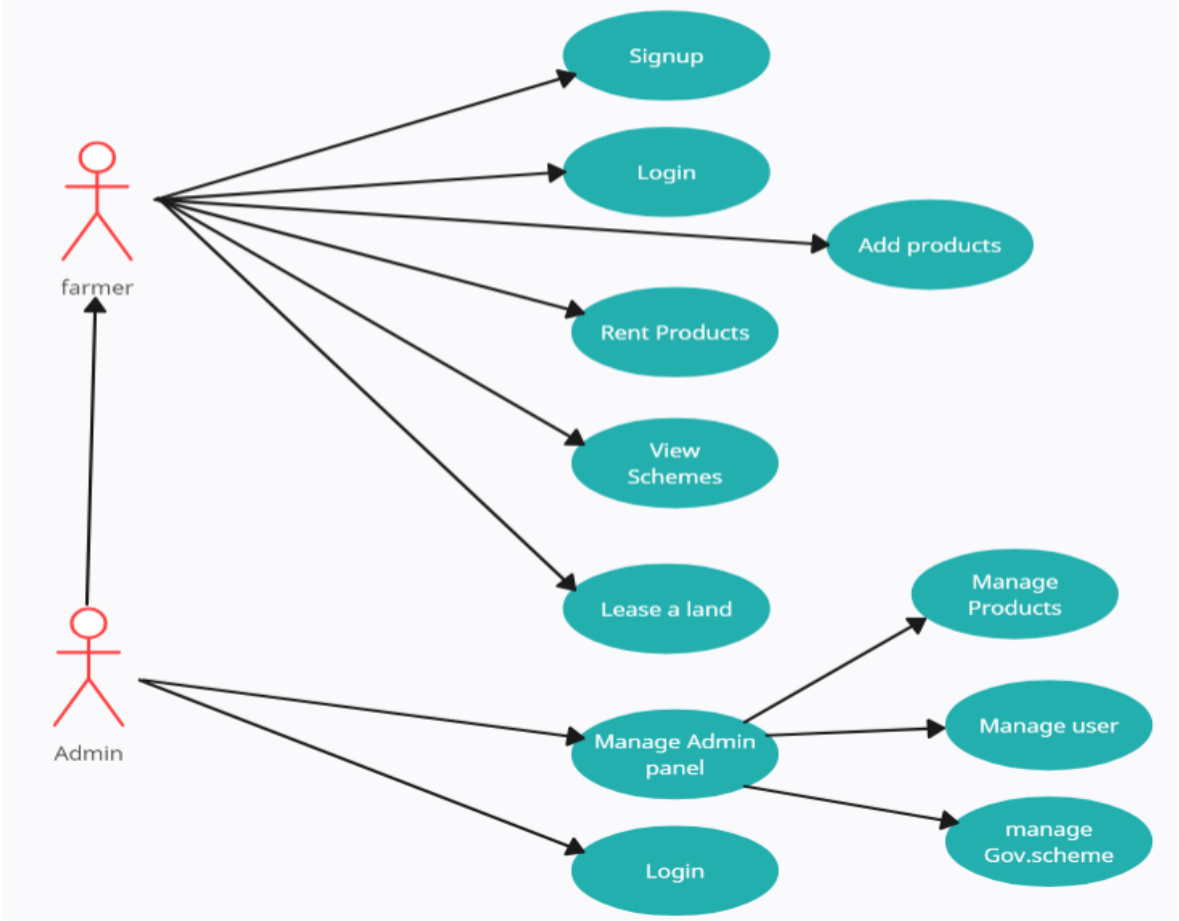
3.2 Proposed Solutions (ER Diagram)



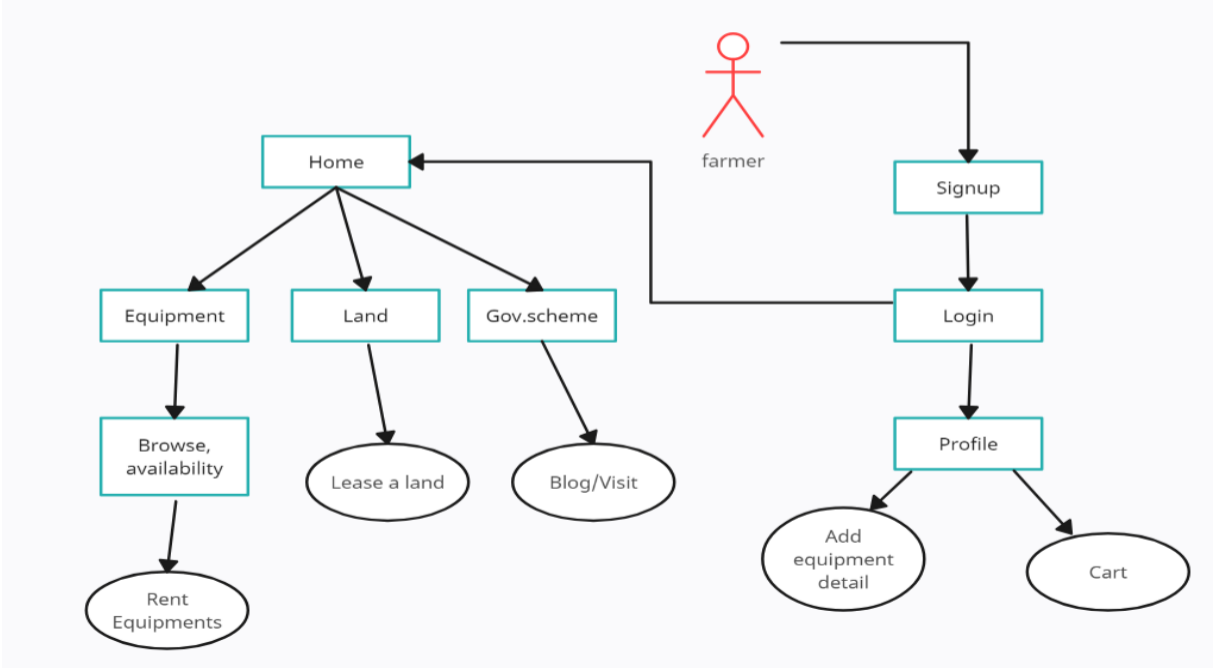
3.3

3.3.1 Architecture/ FrameworkDesign of the proposed system





3.3.2 Algorithm and Process Design



3.3.3 Details of Hardware & Software

Hardware:

CPU: X86_64 architecture with a clock speed of 1.6 GHz.

Memory: 4GB. RAM

Disk Space: 2GB.

Software:

Frontend: Html, CSS, ReactJs, JavaScript , Bootstrap

Backend: NodeJs (Sequelize ORM)

Database: MySql

Framework: ExpressJs

3.3.4 Experiment and Results (Code and GUI)

Code: Frontend

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="public/apple-touch-icon.png" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <script src="https://unpkg.co/gsap@3/dist/gsap.min.js"></script>
    <script src="https://unpkg.com/gsap@3/dist/ScrollTrigger.min.js"></script>
    <script src="https://unpkg.com/gsap@3/dist/MotionPathPlugin.min.js"></script>
    <script src=" https://unpkg.com/gsap@3/dist/MotionPathPlugin.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <link rel="apple-touch-icon" sizes="180x180" href="public/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="public/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="public/favicon-16x16.png">
    <link rel="manifest" href="public/site.webmanifest">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min
    <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
    <title>Farming</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

App.jsx

```
import "./App.css";
import "./styles/styles.css";
import { Route, Routes } from "react-router";
import CategoryScreen from "./Screens/Category/CategoryScreen";
import Home from "./Screens/Home/Home";
import Navbar from "./Components/Navbar/Navbar";
import Itemsbar from "./Components/Itemsbar/Itemsbar";
import Footer from "./Components/Footer/Footer";
import Intro from "./Components/Intro/Intro";
import HowItWorks from "./Components/HowItWorks/HowItWorks";
import Testimonials from "./Sections/Testimonials/Testimonials";
import Faq from "./Sections/Faq/Faq";
import Trending from "./Sections/Trending/Trending";
import TractorAnimation from "./Components/TractorAnimation/TractorAnimation";
import Features from "./Components/Features/Features";
import { useEffect, useState } from "react";
import LoadingAnimation from "./Components/LoadingAnimation/LoadingAnimation";
import ChatbotButton from "./Components/ChatBot/ChatBot";
import Login from "./Components/Login/Login";
import Signup from "./Components/Signup/Signup";
import CategorySection from "./Components/CategorySection/CategorySection";
import { ToastContainer, toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
import { useNavigate } from "react-router";
import Product from "./Screens/Product/Product";
import Cart from "../src/Screens/Cart/Cart";
import Profile from "./SubComponents/Profile/Profile";
import LandTools from "./Screens/LandTool/LandTool";
import Land from "./Screens/LandTool/Land";
import LandMap from "./Components/LandMap/LandMap";
```

index.html App.jsx

src > App.jsx > ...

```jsx
function App() {
  const [isLoading, setIsLoading] = useState(true);
  const [authenticated, setAuthenticated] = useState(false);
  const [about, setAbout] = useState({});
  const navigate = useNavigate('');
  const handleAuthentication = (status) => {
    if (status == false) {
      localStorage.clear();
      localStorage.removeItem("RLog");
      localStorage.removeItem("RName");
    }

    setAuthenticated(status);

    navigate("/");

  };
  useEffect(() => {
    let val = localStorage.getItem("RLog");
    let username = localStorage.getItem("RName");
    let useremail = localStorage.getItem("email");

    if (val == "yes") {
      setIsLoading(false);
      setAuthenticated(true);
      setAbout({
        "name": username,
        "email": useremail
      })
    } else if (!authenticated) {
```

index.html App.jsx main.jsx

src > main.jsx

```jsx
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App";
import "./index.css";
import { BrowserRouter as Router } from "react-router-dom";
import State from "./Context/State";
import { store } from "./store/reducer";
import { Provider } from "react-redux";

ReactDOM.createRoot(document.getElementById("root")).render(
  <Router>
    <Provider store = {store}>
      <State>
        <App />
      </State>
    </Provider>
  </Router>
);

```

Code: Admin

index.html

public > index.html > html > head

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1, shrink-to-fit=no"
    />
    <meta name="theme-color" content="#000000" />
    <!--
      manifest.json provides metadata used when your web app is added to the
      homescreen on Android. See https://developers.google.com/web/fundamentals/engage-and-retain/web-app-m
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico" />
    <link
      rel="apple-touch-icon"
      sizes="76x76"
      href="%PUBLIC_URL%/apple-touch-icon.png"
    />
    <link
      rel="stylesheet"
      href="//cdn.jsdelivr.net/chartist.js/latest/chartist.min.css"
    />

    <script src="//cdn.jsdelivr.net/chartist.js/latest/chartist.min.js"></script>
    <link
      rel="stylesheet"
      href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700|Material+Icons"
```

index.html | index.js

src > index.js

```
import React from "react";
import ReactDOM from "react-dom";
import { BrowserRouter, Route, Switch, Redirect } from "react-router-dom";
// import "primeicons/primeicons.css";
import "primereact/resources/themes/lara-light-indigo/theme.css";
import "primereact/resources/primereact.css";

// core components
import Admin from "layouts/Admin.js";
import RTL from "layouts/RTL.js";

import "assets/css/material-dashboard-react.css?v=1.10.0";

ReactDOM.render(
  <BrowserRouter>
    <Switch>
      <Route path="/admin" component={Admin} />
      <Route path="/rtl" component={RTL} />
      <Redirect from="/" to="/admin/login" />
    </Switch>
  </BrowserRouter>,
  document.getElementById("root")
);

```

index.html | index.js | ProductList.js

src > views > TableList > ProductList.js > ...

```
import React from "react";
import GridContainer from "components/Grid/GridContainer.js";
import { useState, useEffect } from "react";
import { DataTable } from "primereact/datatable";
import { Column } from "primereact/column";
import DeleteIcon from "@material-ui/icons/Delete";
import axios from "axios";
import { useHistory } from "react-router-dom";

export default function ProductList() {
  const [products, setProducts] = useState([]);
  const [path, setPath] = useState("");
  let history = useHistory();

  useEffect(() => {
    if (!sessionStorage.getItem("id")) {
      history.push("/home");
    }
  }, [history]);

  useEffect(() => {
    async function getAllProduct() {
      const resData = await axios.post(
        `http://localhost:5007/api/products/getAllProducts`,
        {
          headers: {
            "Content-Type": "application/json",
          },
        }
```

index.html | index.js | ProductList.js | Login.js

src > views > TableList > Login.js > ...

```
import React from "react";
import "./style.css";
import { useHistory } from "react-router-dom";

export default function Login() {
  const [email, setEmail] = React.useState("");
  const [password, setPassword] = React.useState("");
  const history = useHistory();

  const handleSubmit = () => {
    if (
      (email === "admin" || email === "admin@gmail.com") &&
      password === "Admin@123"
    ) {
      sessionStorage.setItem("id", 1);
      history.push("/admin/user-list");
    } else {
      alert("Username or Password is Wrong!");
    }
  };
  return (
    <>
      <div className="container">
        <div className="row justify-content-center mt-5">
          <div className="col-lg-4 col-md-6 col-sm-6">
            <div className="card shadow">
              <div className="card-title text-center border-bottom">
                <h2 className="p-3">Login</h2>
              </div>
              <div classNam="card-body">
```

Code: API

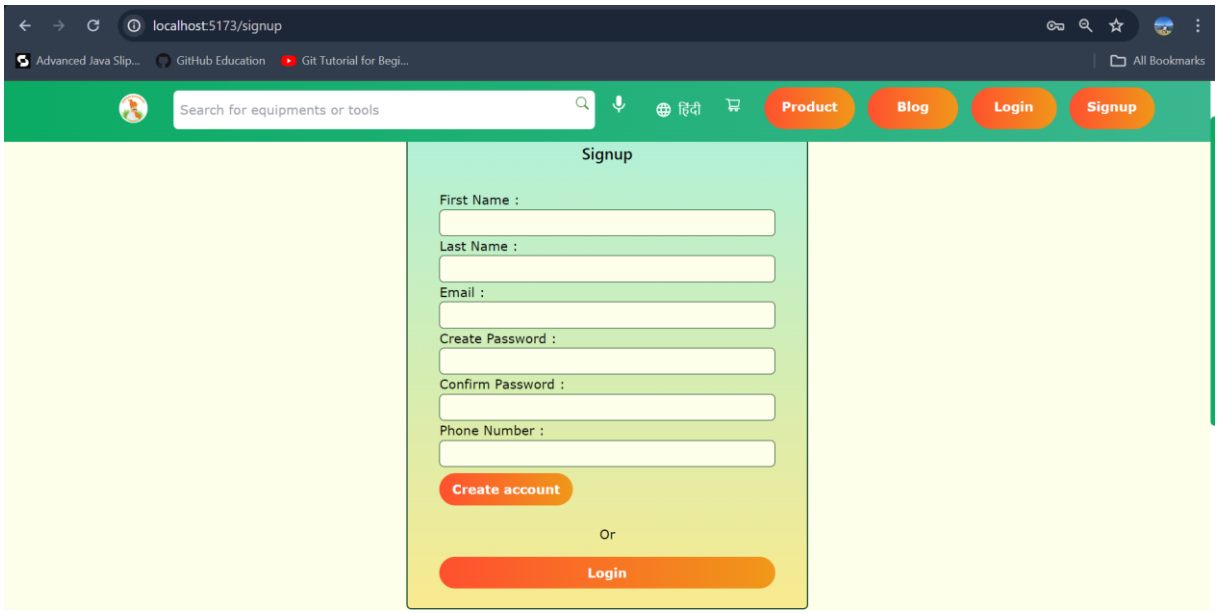
server.js

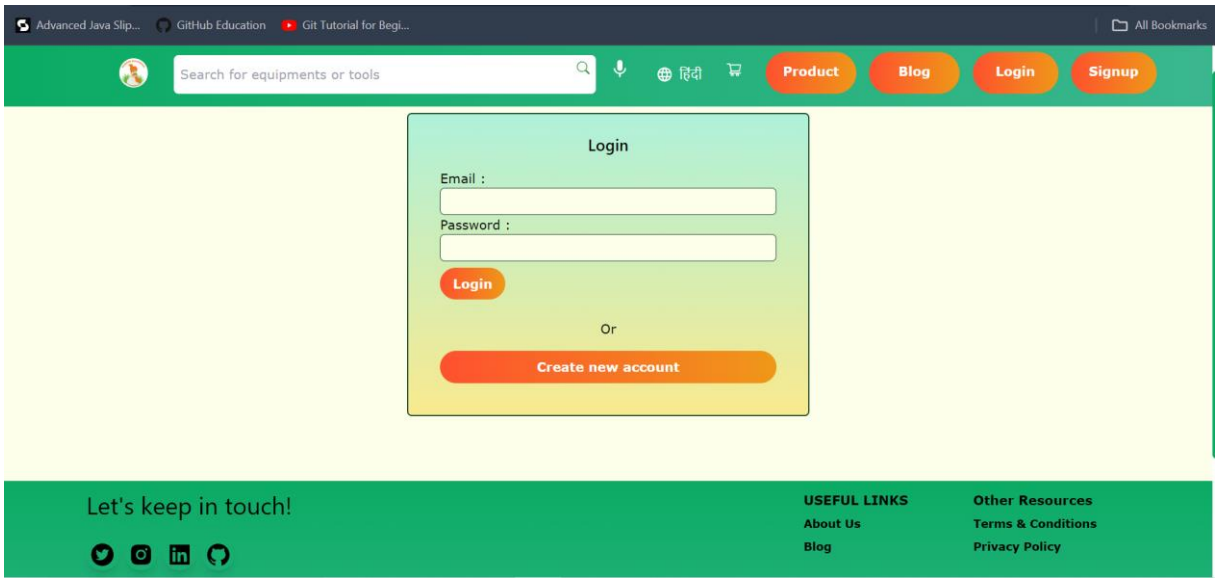
```
server.js > corsOptions > origin
1   const express = require("express");
2   const cors = require("cors");
3   const app = express();
4   require("dotenv").config();
5   const fileUpload = require("express-fileupload");
6   // import sequelize from "./app/model/connection.js";
7   const db = require("./app/model/index.js");
8   const AppError = require("./app/utils/appError.js");
9   const errorController = require("./app/utils/errorController.js");
10
11  var corsOptions = {
12      origin: [
13          "*",
14          "http://localhost:5173",
15          "http://localhost:3000",
16          "localhost:5007",
17          "localhost:3000"
18      ],
19      methods: ["GET", "POST", "OPTION"],
20      allowedHeaders: [
21          "Access-Control-Allow-Headers",
22          "Access-Control-Allow-Headers, Origin, X-Requested-With, Content-Type, Accept, authorization, isLogin",
23      ],
24  };
25
26  app.use(cors(corsOptions));
27
28  app.use(
29      express.json({
```

server.js

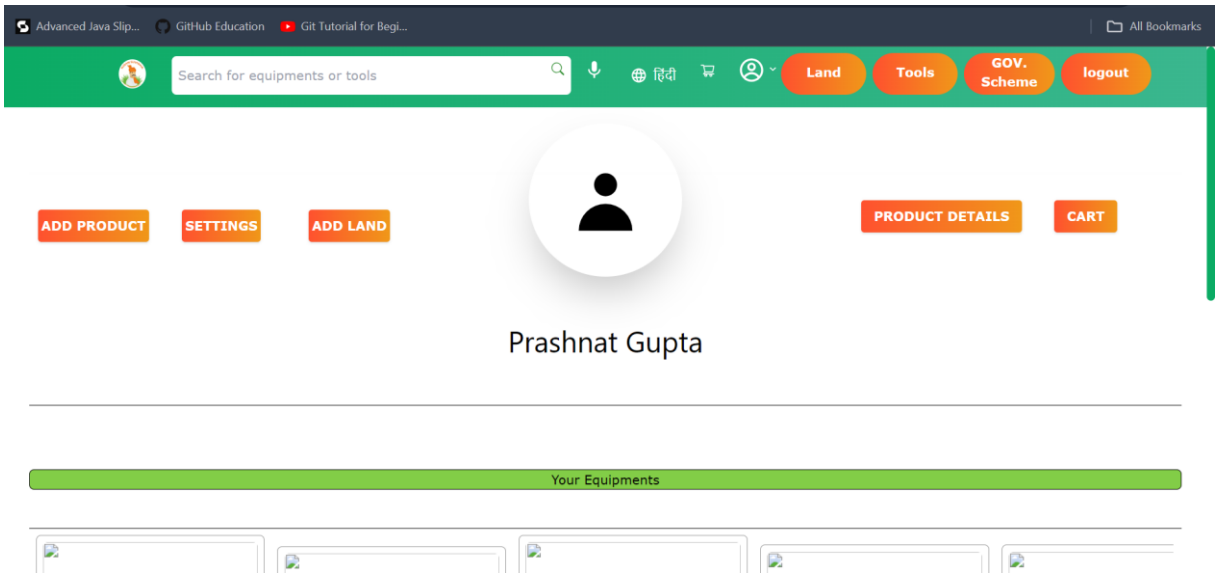
```
server.js > corsOptions > origin
28  app.use(
29      express.json({
30          limit: "1000mb",
31          extended: true,
32          parameterLimit: 500000,
33      })
34  );
35  app.use(
36      express.urlencoded({
37          extended: true,
38          limit: "1000mb",
39          parameterLimit: 500000,
40      })
41  );
42
43
44  app.use(fileUpload());
45
46  app.get("/api", (req, res, next) => {
47      try {
48          db.authenticate().then(() => {
49              return res.send({
50                  status: "success",
51                  code: "200",
52                  message: "Welcome to Farmer api",
53              });
54          });
55      } catch (error) {
56          next(error);
```

GUI: login & sugnup

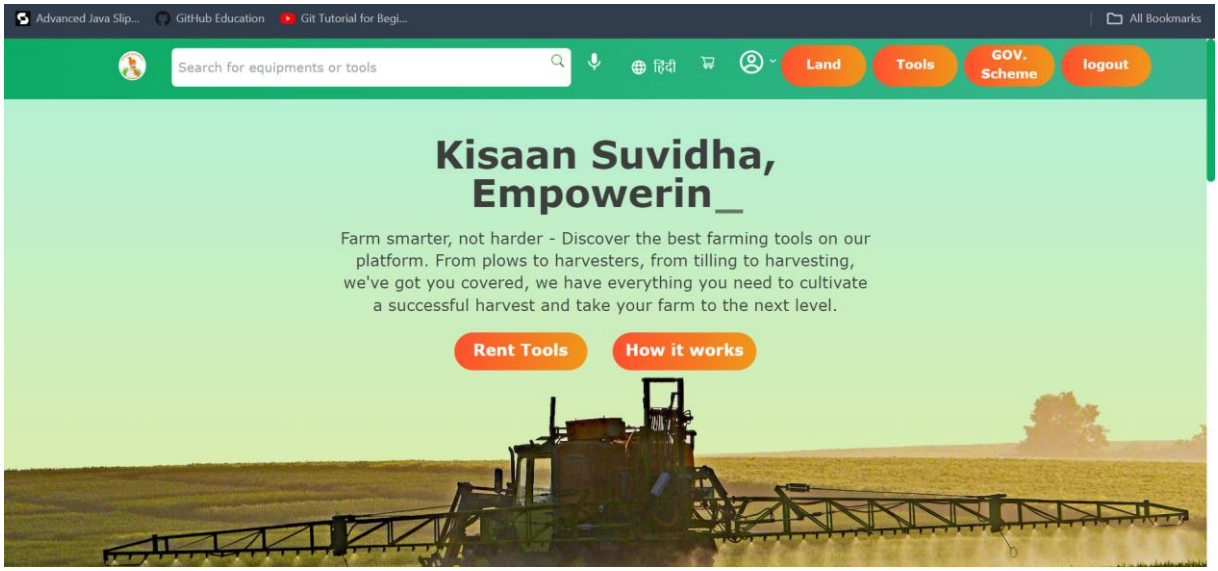




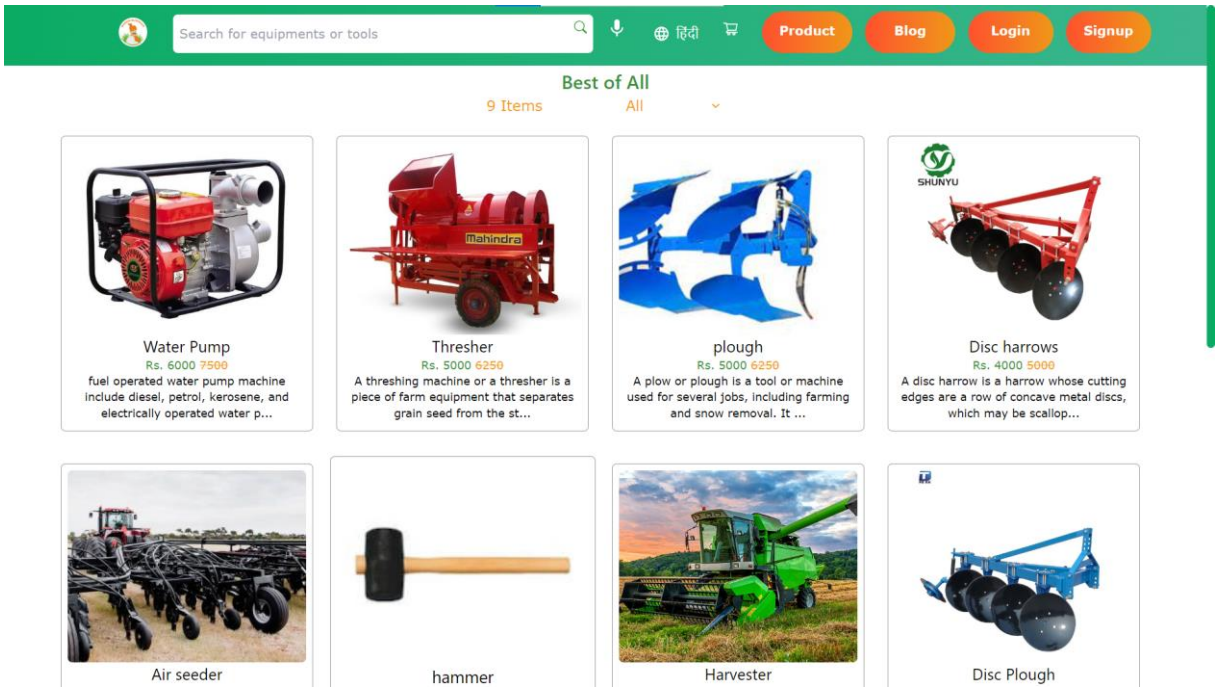
GUI: profile page



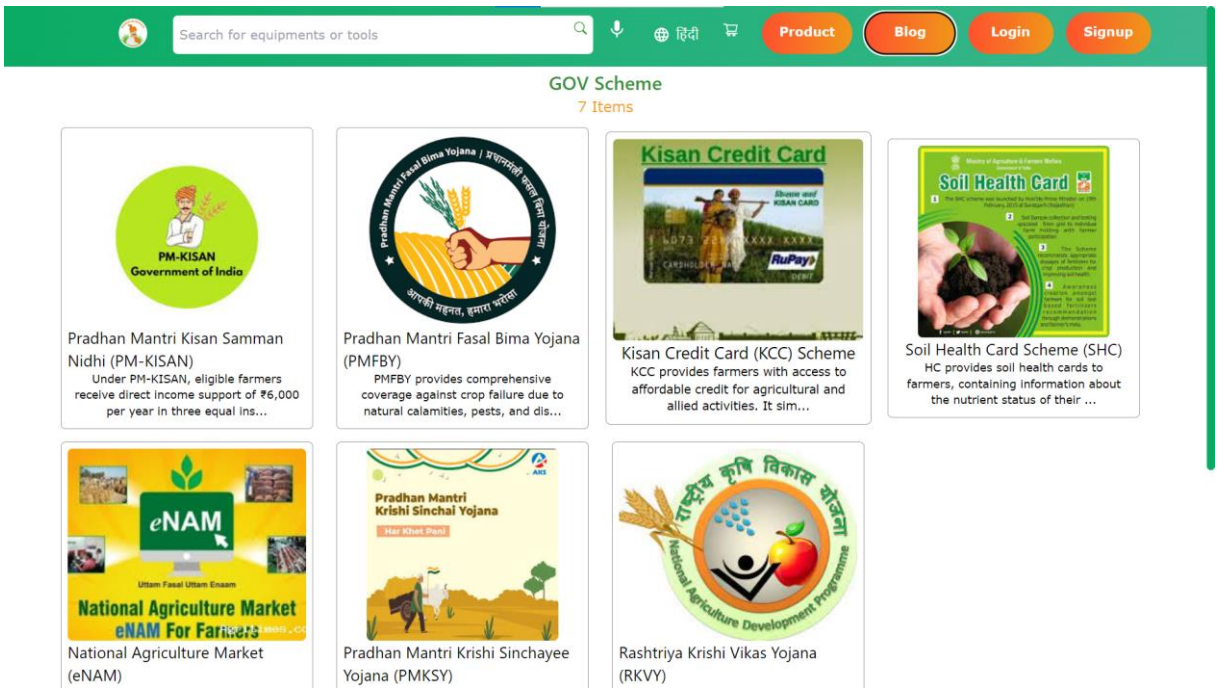
Homepage:



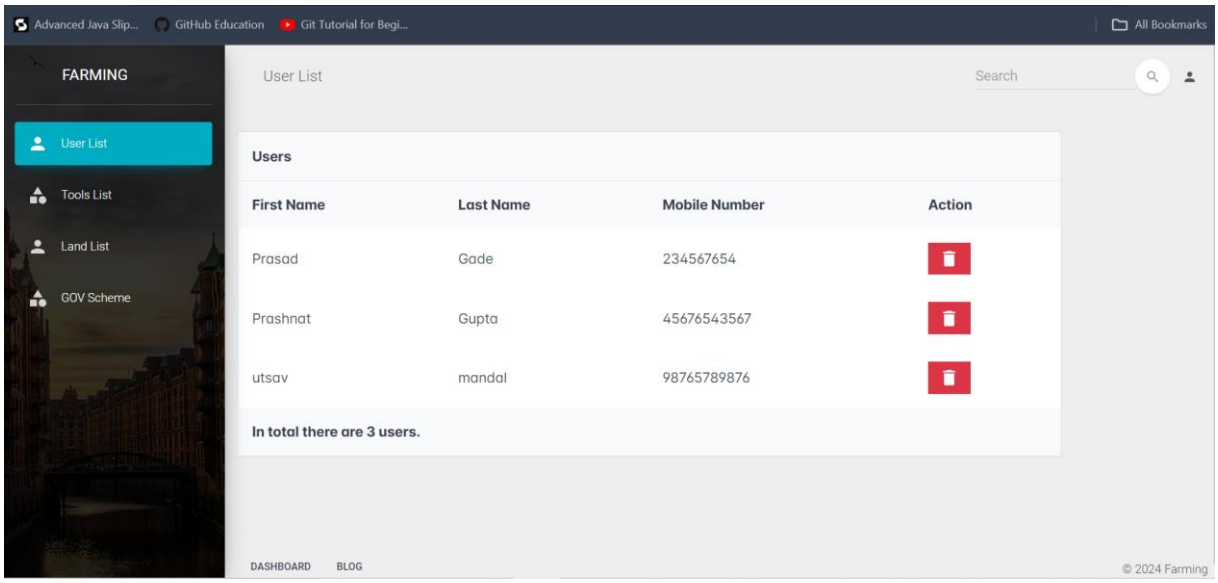
Product page:



Government schemes page:



Admin panel:



MySql: