### PROJECT TITLE

**A PROJECT REPORT**

***Submitted by***

### VELAVAN M 727821TUEE150

***in partial fulfillment for the award of the degree of***

### BACHELOR OF ENGINEERING IN

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

### June 2023

**BONAFIDE CERTIFICATE**

Certified that this project report **“PROJECT TITLE”** is the bonafide work of

### “VELAVAN M” who carried out the project work under my supervision.

#### SIGNATURE SIGNATURE

**Dr.R.VIDHYA**

#### (Industry Mentor Name) HEAD OF THE DEPARTMENT

Designation Associate Professor,

Department of Computer Science

and Engineering

Sri Krishna College of Technology,

Coimbatore-641042.

Certified that the candidates were examined by us in the Project Work Viva Voce examination held on at Sri Krishna College of Technology, Kovaipudur, Coimbatore -641042

### INTERNAL EXAMINER EXTERNAL EXAMINER

***ACKNOWLEDGEMENT***

### ACKNOWLEDGEMENT

First and foremost, we thank the **Almighty** for being our light and for showering his gracious blessings throughout the course of this project.

We are grateful to our beloved Principal **Dr.M.G. Sumithra M.E., Ph.D**. for her tireless and relentless support.

We extend our sincere thanks to our Head of the Department **Dr.R.Vidhya M.Tech., Ph.D.** for her encouragement and inspiration.

We are greatly indebted to our Industry Mentor **Mr.Manoj Kumar**  for his valuable guidance and suggestions in all aspects that aided us to ameliorate our skills.

We are thankful to all those who have directly and indirectly extended their help to us in completing this project work successfully.

# ABSTRACT



iv

## ABSTRACT

The fertilizer selling app is a revolutionary digital platform designed to facilitate the selling of fertilizers in the agricultural industry. With the increasing demand for agricultural products and the need for improved crop productivity, this app serves as a convenient and efficient solution for farmers, suppliers, and distributors. The app features a comprehensive product catalog, allowing users to access detailed information on fertilizers, including composition, application instructions, and recommended usage. Advanced search and filtering options enable users to find the most suitable fertilizers based on crop type, nutrient requirements, and brand. Seamless ordering and secure payment processes ensure a smooth transaction experience. With features such as delivery tracking, reviews and ratings, and agricultural insights, the app enhances transparency, quality assurance, and productivity in the agricultural sector. The app's user-friendly interface, scalable architecture, and robust security measures make it a valuable tool in the evolving digital agriculture landscape.

# TABLE OF CONTENTS



**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER.NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | **iv** |
|  | **LIST OF FIGURES** | **vi** |
|  | **LIST OF ABBREVIATIONS** | **vii** |
| 1 | 1.2 OVERVIEW | 2 |
|  | 1.3 OBJECTIVE | 2 |
| 2 | LITERATURE SURVEY | 3 |
| 3 | SYSTEM SPECIFICATIONS | 5 |
| 4 | PROPOSED SYSTEM | 8 |
|  | 4.1 PROPOSED SYSTEM | 8 |
|  | 4.2 ADVANTAGES | 9 |
| 5 | METHODOLOGIES | 10 |
|  | 5.1 LOGIN PAGE | 11 |
|  | 5.2 SIGN UP PAGE | 12 |
|  | 5.3 FORGET PASSWORD | 13 |
|  | 5.4 PRODUCT PAGE | 13 |
|  | 5.5 CART PAGE | 14 |
| 6 | IMPLEMENTATION AND RESULT | 15 |
| 7 | CONCLUSION AND FUTURE SCOPE | 36 |
|  | 7.1 CONCLUSION | 36 |
|  | 7.2 FUTURE SCOPE | 37 |
| 8 | REFERENCES | 38 |

# LIST OF FIGURES



vi

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **TITLE** | **Page No** |
| 5.1 | Process flow diagram | 10 |
| 5.2 | Login page flowchart | 11 |
| 5.3 | Signup page flowchart | 12 |
| 5.4 | Products page flowchart | 13 |
| 5.5 | Cartpage flowchart | 14 |
| 6.1 | Login page | 15 |
| 6.2 | Signup page | 16 |
| 6.3 | Recovery page | 17 |
| 6.4 | Otp verification | 17 |
| 6.5 | Home page | 18 |
| 6.6 | Product page | 19 |

# LIST OF ABBREVIATIONS



vii

### LIST OF ABBREVIATIONS

#### ABBREVIATIONS

**ACRONYMS**

**HTML** HYPERTEXT MARKUP LANGUAGE

**CSS** CASCADING STYLESHEET

**JS** JAVASCRIPT

**PC** PERSONAL COMPUTER

# INTRODUCTION



1

## CHAPTER 1 INTRODUCTION

The fertilizer selling app is a transformative digital platform that aims to revolutionize the agricultural industry by providing a convenient and efficient solution for selling fertilizers. With the global demand for agricultural products on the rise and the need to improve crop productivity, this app serves as a crucial tool for farmers, suppliers, and distributors. The fertilizer selling app aims to address these challenges by leveraging the power of digital technology and mobile connectivity. By connecting farmers, suppliers, and distributors on a single platform, the app streamlines the entire fertilizer procurement process, making it more accessible, transparent, and convenient.

One of the key features of the app is its comprehensive product catalog. It provides users with a wide range of fertilizers to choose from, along with detailed information on each product. Users can access information such as composition, application instructions, recommended usage, and pricing. Once users have identified their desired fertilizers, the app simplifies the ordering process. It enables users to add fertilizers to their cart, review their order, and proceed to a secure payment gateway for seamless transactions. This eliminates the need for time-consuming and manual order placement, making the entire buying process more efficient and convenient. Another notable aspect of the fertilizer selling app is its focus on providing agricultural insights and recommendations.

Users can access valuable information such as fertilization schedules, best practices, and crop-specific recommendations based on their location and soil conditions. By offering these insights, the app aims to empower farmers to optimize their fertilizer usage, improve crop yields, and adopt sustainable farming practices. With its potential to enhance transparency, convenience, and productivity, the app represents a significant step forward in the digital transformation of the agricultural sector.

2

### PROBLEM STATEMENT

The traditional process of buying and selling fertilizers in the agricultural industry is plagued by inefficiencies, limited product availability, lack of transparency, and a fragmented supply chain. Farmers, suppliers, and distributors face numerous challenges in accessing and procuring the right fertilizers for their specific needs. These challenges include limited access to a comprehensive catalog of fertilizers, lack of detailed product information, difficulties in finding suitable fertilizers based on crop type and nutrient requirements, and a cumbersome ordering and delivery process. Moreover, the absence of a centralized platform hampers transparency and makes it challenging for users to make informed decisions. The need of the hour is a fertilizer app that can streamline the entire fertilizer procurement process, provide a wide range of fertilizers with detailed information, offer advanced search and filtering options, simplify the ordering and payment process, and provide real-time updates on delivery status. Addressing these challenges and providing a user-friendly, efficient, and transparent solution would greatly enhance the fertilizer buying and selling experience for all stakeholders in the agricultural industry.

### OVERVIEW

The Fertilizer Selling App is a digital platform designed to streamline the buying and selling of fertilizers in the agricultural industry. It offers a comprehensive catalog of fertilizers with detailed information, allowing users to find the most suitable products based on crop type and nutrient requirements. The app simplifies the ordering process and provides secure payment transactions. It also includes a delivery management system for real-time tracking. With a focus on transparency and agricultural insights, the app aims to enhance efficiency and productivity in the fertilizer procurement process.

3

### OBJECTIVE

Enhance Accessibility: The primary objective of the app is to make fertilizers easily accessible to farmers, suppliers, and distributors. By providing a comprehensive catalog and detailed product information, the app aims to overcome the limitations of traditional purchasing methods and ensure a wide range of fertilizers are readily available.

Simplify Procurement Process: The app aims to simplify the fertilizer procurement process by offering a user-friendly interface and streamlined features. Users can easily search for fertilizers, add them to their cart, and proceed with secure payment transactions, eliminating the complexities and time-consuming nature of traditional methods.

Provide Agricultural Insights: The app aims to go beyond being a mere marketplace by offering valuable agricultural insights. Users can access fertilization schedules, best practices, and crop-specific recommendations based on location and soil conditions. This objective empowers farmers with knowledge to enhance their fertilizer usage and improve overall productivity.

Enhance User Experience: User experience is a central objective for the app. By providing a user-friendly interface, seamless ordering process, real-time delivery tracking, and secure transactions, the app aims to create a positive and hassle-free experience for users, ensuring their satisfaction and loyalty.

By aligning with these objectives, the Fertilizer Selling App aims to transform the way fertilizers are bought and sold, revolutionizing the agricultural industry and supporting the needs of farmers, suppliers, and distributors.

4

# LITERATURE SURVEY



## CHAPTER 2 LITERATURE SURVEY

### 2.1 RELATED WORKS

1. "Mobile Applications in Agriculture: A Review of Literature and Trends" by E. Geze, A. Ozcelik, and O. Kucuk published in Computers and Electronics in Agriculture (2016).

This paper provides an overview of mobile applications in agriculture, highlighting their benefits and challenges. It discusses the use of mobile apps for various agricultural purposes, including fertilizer management and procurement.

2. "Mobile Applications in Agriculture and Rural Development: A Systematic Literature Review" by M. Akter, R. H. Rayhan, and G. S. D. Paul published in Information Processing in Agriculture (2019).

This systematic literature review explores the use of mobile applications in agriculture and rural development. It provides insights into the different functionalities and features offered by mobile apps in the context of agriculture, including their potential for facilitating fertilizer management.

3. "Smart Farming Technologies for Sustainable Agricultural Development: A Systematic Review of Literature" by R. Ali, S. N. Sarwar, and S. A. Chaudhry published in Computers and Electronics in Agriculture (2017).

This paper presents a systematic review of smart farming technologies, including mobile applications, for sustainable agricultural development. It discusses the role of mobile apps in optimizing fertilizer usage, improving crop yields, and enhancing overall agricultural productivity.

4. "Farmers' Use of Mobile Devices and the Impact on Modern Agricultural Practices" by N. V. Reyes, M. T. Flottemesch, and D. P. Trechter published in Journal of Agricultural Education (2014).

This study examines the adoption and use of mobile devices by farmers and their impact on modern agricultural practices. It discusses the potential benefits of mobile apps in the agricultural sector, including the ability to access fertilizer information, make informed decisions, and improve farm management.

5. "A Review on Precision Agriculture Monitoring Systems: Sensing Technologies, Data Acquisition Strategies, and Applications" by H. Wang, S. Li, and D. Wang published in Journal of Sensors (2019).

This review article focuses on precision agriculture monitoring systems, including mobile applications, for improved agricultural practices. It highlights the use of mobile apps for fertilizer management, data acquisition, and decision support, emphasizing their role in optimizing fertilization strategies.

6. "ICT Applications in Agriculture: Prospects and Challenges" by D. Ghosh and S. Ray published in Indian Journal of Science and Technology (2013).

This article discusses the prospects and challenges of using information and communication technology (ICT) applications in agriculture. It explores the potential of mobile apps in addressing challenges related to fertilizer procurement, optimizing fertilizer usage, and enhancing agricultural productivity.

These literature sources provide valuable insights into the role of mobile applications in agriculture, including their potential in the context of fertilizer selling and management. They highlight the benefits, challenges, and trends associated with such apps, emphasizing the importance of user-friendly interfaces, real-time data, and sustainable farming practices.

# SYSTEM SPECIFICATION



## CHAPTER 3 SYSTEM SPECIFICATION

In this chapter, we are gonna see the softwares that we have used to build the website. This chapter gives you a small description about the softwares used in the project.

### 3.1 VS CODE

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux, and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences.

VS Code is an excellent code editor for React projects. It is lightweight, customizable, and has a wide range of features that make it ideal for React development. It has built-in support for JavaScript, JSX, and TypeScript, and enables developers to quickly move between files and view detailed type definitions. It also has a built-in terminal for running tasks. Additionally, VS Code has an extensive library of extensions that allow developers to quickly add features like code snippets, debugging tools, and linting support to their projects.

### 3.2. REACT

React is a JavaScript library created by Facebook for building user interfaces. It is a component-based, declarative, and highly efficient library that is used to develop interactive UIs (user interfaces) for single page web applications. React uses a virtual DOM (Document Object Model) that makes it faster and easier to manipulate the DOM

elements. It also provides declarative components that allow developers to write code that is easy to read and maintain. React also offers an extensive library of tools and components that make it easier to develop complex user interfaces.

### ROUTERS IN REACT

Routers are important components in React applications. They provide the ability to navigate between different views or components of the application. React Router is the most popular library to handle routing in React applications. It provides the ability to define routes, set up links, and render components based on the current route. It also provides features like data fetching, code-splitting, and server-side rendering.

### LOCAL STORAGE

Local storage is a type of web storage for storing data on the client side of a web browser. It allows websites to store data on a user’s computer, which can then be accessed by the website again when the user returns. Local storage is a more secure alternative to cookies because it allows websites to store data without having to send it back and forth with each request. Local storage is a key-value pair storage mechanism, meaning it stores data in the form of a key and corresponding value. It is similar to a database table in that it stores data in columns and rows, except that local storage stores the data in the browser rather than in a database. Local storage is often used to store user information such as preferences and settings, or to store data that is not meant to be shared with other websites. It is also used to cache data to improve the performance of a website. Local storage is supported by all modern web browsers, including Chrome,

Firefox, Safari, and Edge. It is accessible through the browser’s JavaScript API. Local storage is a powerful tool for websites to store data on the client side. It is secure, efficient, and can be used to store data that does not need to be shared with other websites.

Local Storage is a great way to improve the performance of a website by caching data. Local storage in web browsers allows website data to be stored locally on the user’s computer. It is a way of persistently storing data on the client side, which is not sent to the server with each request. This allows users to store data such as preferences, login information, and form data without needing to send it to a server. It is typically stored in a browser’s cookie file, but it can also be stored in other locations such as HTML5 Local Storage and IndexedDB. The data stored in local storage is persistent and can be accessed by the website even if the user closes the browser or navigates to another page. It is a great way for websites to store user-specific data, as it is secure, reliable, and fast. It is also a great way for developers to store data that does not need to be sent to the server with each request.

One of the key benefits of using local storage is its reliability. Unlike server-side storage, which can be affected by network outages or other server issues, local storage is stored locally on the user’s machine, and so is not affected by these issues. Another advantage of local storage is its speed. Because the data is stored locally, it is accessed quickly, as there is no need to send requests to a server. This makes it ideal for storing data that needs to be accessed quickly, such as user preferences or session data. Local storage is also secure, as the data is stored on the user’s machine and not on a server. This means that the data is not accessible by anyone other than the user, making it a good choice for storing sensitive information.

# PROPOSED SYSTEM



## CHAPTER 4 PROPOSED SYSTEM

This chapter gives a small description about the proposed idea behind the development of our website

### PROPOSED SYSTEM

The proposed system for the fertilizer selling app aims to create a comprehensive and user-friendly platform that facilitates the buying and selling of fertilizers in the agricultural industry. The system incorporates various features and functionalities to enhance the overall user experience and streamline the fertilizer procurement process.

The system allows users, including farmers, suppliers, and distributors, to register and create profiles. Users can manage their personal information, preferences, and communication settings within their profiles. The app features a comprehensive catalog of fertilizers, including various brands and types. Each fertilizer listing provides detailed information such as composition, nutrient content, application instructions, recommended usage, and pricing. Users can easily search for fertilizers based on specific criteria such as crop type, nutrient requirements, brand, and price range. Advanced filtering options enable users to narrow down their search and find the most suitable fertilizers

The proposed system aims to provide a user-friendly and efficient platform for buying and selling fertilizers. By incorporating features such as a comprehensive catalog, advanced search options, secure transactions, delivery tracking, agricultural insights, and customer support, the system streamlines the fertilizer procurement process, enhances transparency, and improves the overall user experience.

9

### ADVANTAGES

Convenience: The fertilizer selling app offers convenience to users by providing a centralized platform for buying and selling fertilizers. Users can access the app anytime, anywhere, using their mobile devices, eliminating the need to physically visit stores or suppliers.

Wide Product Selection: The app provides a comprehensive catalog of fertilizers, offering users a wide range of options to choose from. This ensures that users can find fertilizers that meet their specific crop type, nutrient requirements, and budget, resulting in better agricultural outcomes.

Detailed Product Information: The app offers detailed information about each fertilizer, including composition, application instructions, recommended usage, and pricing. This empowers users to make informed decisions based on the specific needs of their crops, ensuring optimal fertilizer selection.

Advanced Search and Filtering: The app incorporates advanced search and filtering options, allowing users to refine their fertilizer choices based on specific criteria. Users can search for fertilizers based on crop type, nutrient requirements, brand, and price range, making it easier to find the most suitable products.

Transparent Reviews and Ratings: The app includes a review and rating system where users can provide feedback on fertilizers they have purchased. This transparency enables users to make informed decisions based on the experiences of others, promoting reliable product choices.

Secure Transactions: The app ensures secure payment transactions, protecting users' financial information. Integration with trusted payment gateways adds an extra layer of security and confidence for users when making purchases.

Real-time Delivery Tracking: Users can track the status of their orders in real-time through the app. This feature provides updates on the progress of the delivery and estimated delivery times, allowing users to plan their farming activities accordingly.

Agricultural Insights and Recommendations: The app offers valuable agricultural insights and recommendations, including fertilization schedules, best practices, and crop-specific guidance.

# METHODOLOGIES



## CHAPTER 5 METHODOLOGIES

This chapter gives a small description about how our system works.

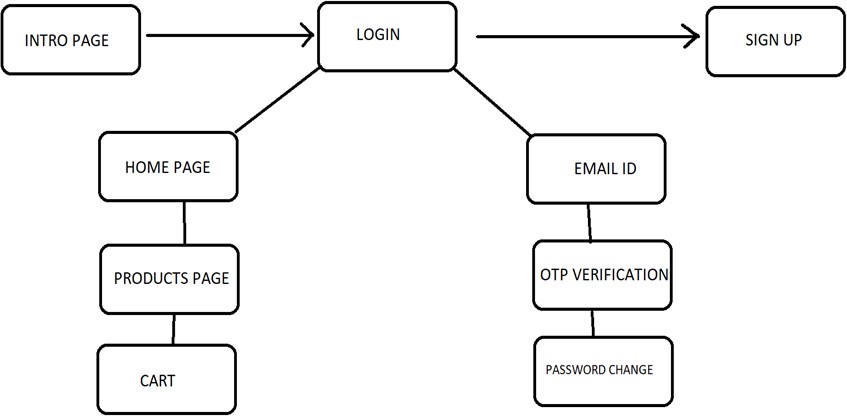


Fig 5.1 PROCESS FLOW DIAGRAM

### 5.1.LOGIN

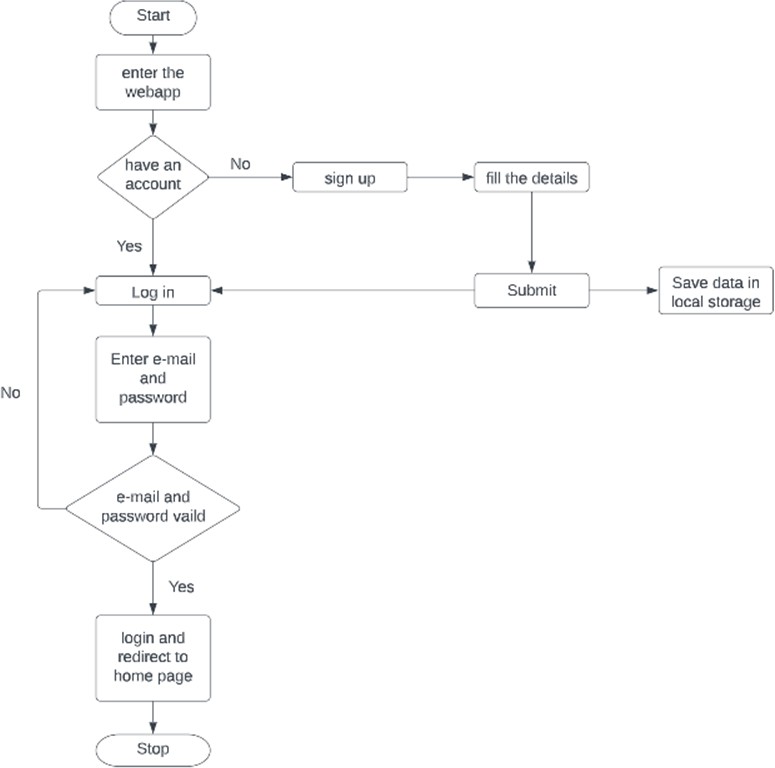


Fig 5.2. LOGIN PAGE FLOWCHART

In this page we will be asking about the username and password of the user. Firstly the website validates the user inputs. It verifies the username and password by checking it with the usernames and passwords stored in the local storage when the user creates an account in the website.

### 5.2 SIGNUP PAGE

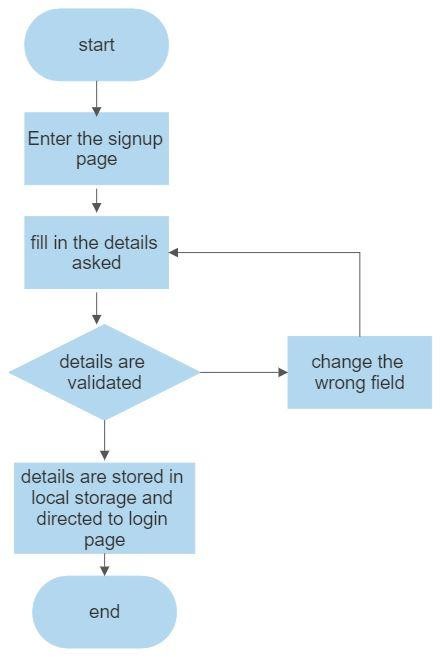


Fig 5.3 SIGNUP PAGE FLOWCHART

This page asks users about the basic details of the user to create an account. This page asks for details like username, password,email id, phone number. After the user enters the details , these details are then validated by our code . If all details are correct then the user is then directed to the login page

### PRODUCTS PAGE

This page displays the list of products available for sale. These details have been displayed from the list of products available in the javascript file named data.js. So that if we want to add some products to a particular product we can easily do it by adding the details to the products js file.

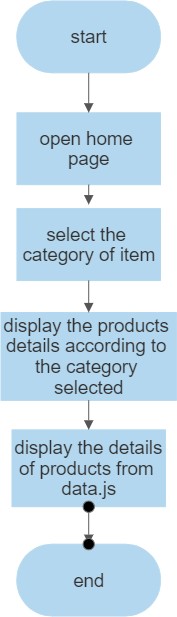


Fig 5.4 PRODUCTS PAGE FLOWCHART

### 5.4 ORDER PAGE

This page displays the set of products that the user selected by clicking the button “Add to cart” available on the products page. It also displays the products with their names and its price with quantity. This also helps you to increase the quantity of the products you have added using a plus button which is available on the page after every product's name. You can also remove or reduce the quantity by using the minus symbol button which is also available below every product's name. If the cart doesn’t contain any item it will display that the cart is empty.

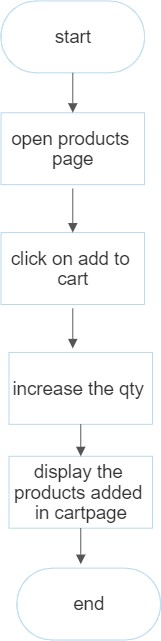


Fig 5.5 CART PAGE FLOWCHART

# IMPLEMENTATION AND RESULT



**CHAPTER 6**

**IMPLEMENTATION AND RESULT**

This chapter gives a description about the output that we produced by developing the website of our idea.

### 6.1 LOGIN

When User enters our website he will be asked about his login details like email id and password. The login details will be verified with the details given while the user creates an account.



Fig 6.1 LOGIN PAGE

### SIGN UP

If a user doesn’t have an account on the website, User can use a component named create new account available in the login page. When the user clicks on that he will be redirected to the signup page. In sign up he should fill up his email id, password and phone number. These inputs will be validated.



Fg 6.2 SIGNUP PAGE

### HOME PAGE

If an existing user forgets the password he can access a component named forgot password where he will be directed for a page that’s requesting their email id, after entering the existing email id the user will be redirected to the email OTP verification, then the user can change their password.

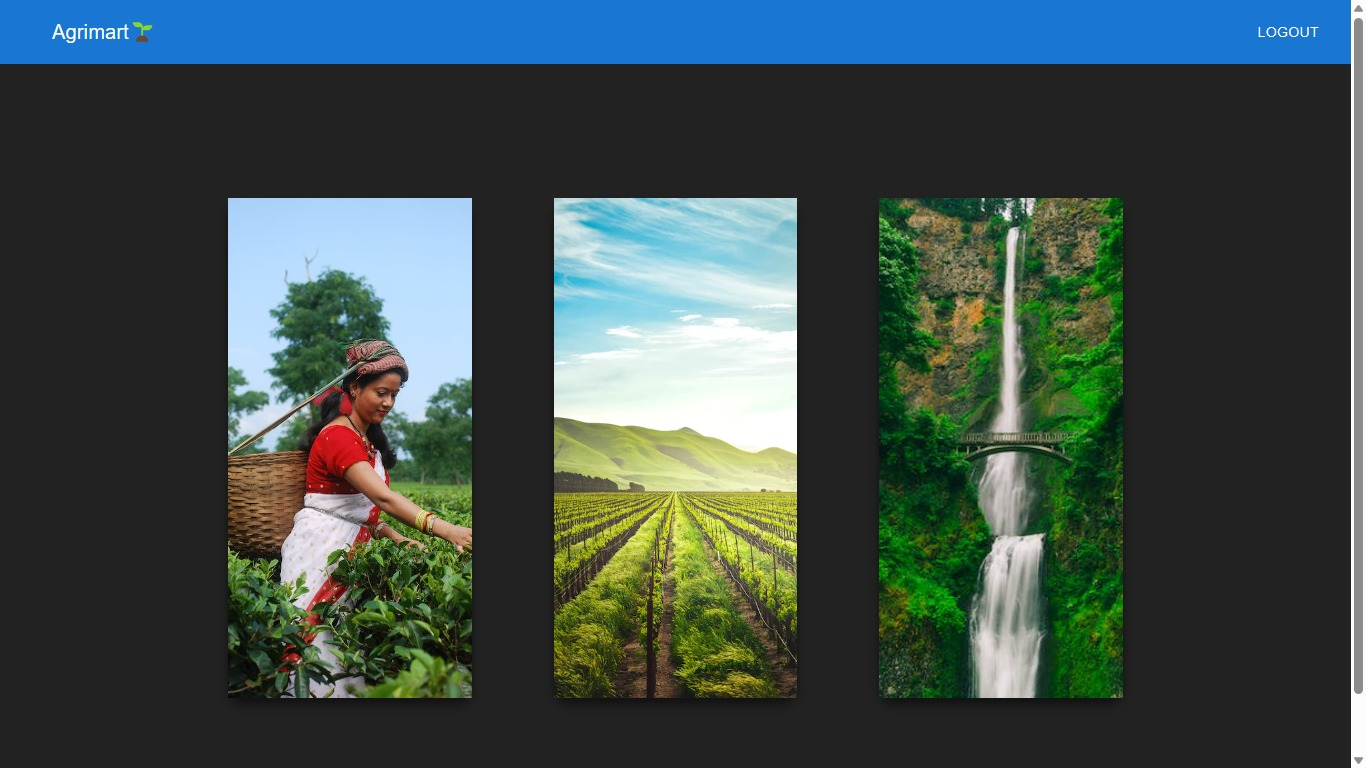


Fig 6.3 RECOVERY PAGE

### LIST OF FERTILIZER

The webpage has different categories of products available for sale and users can redirect to the products description page by selecting the respective products. This page also contains the main taglines of the whole website idea.



Fig 6.4 LIST OF FERTILIZER

### ORDER

This page displays the list of products available for sale. These details have been displayed from the list of products available in the javascript file named data.js[1]. So that if we want to add some products to the particular product we can easily do it by adding the details to the products js file.

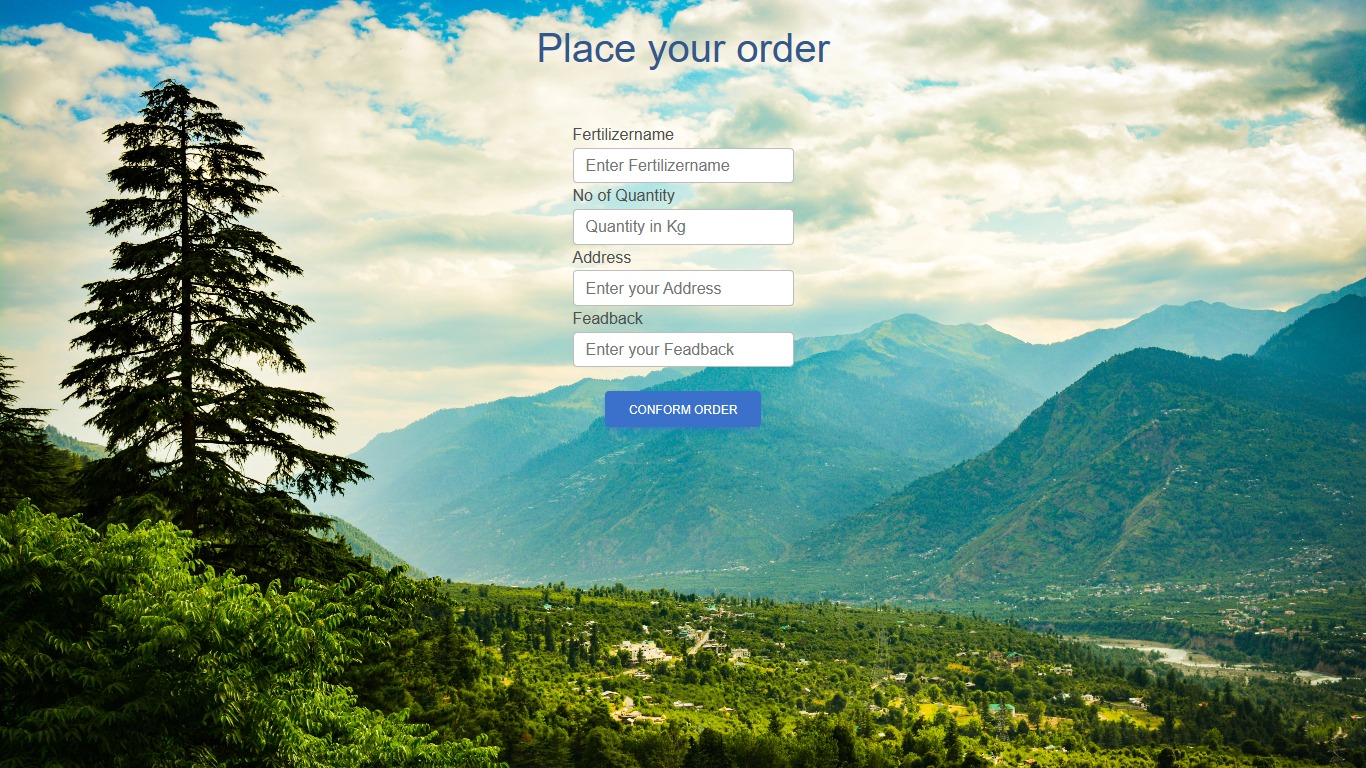


Fig 6.5 PRODUCT PAGE

### 6.6 CODING

#### Login:

#### import { useState } from "react";

#### import { useNavigate } from 'react-router-dom';

#### import axios from "axios";

#### import {Link} from 'react-router-dom';

#### import "../components/login.css";

#### function Login()

#### {

#### const [username, setUsername] = useState("");

#### const [password, setPassword] = useState("");

#### const navigate = useNavigate();

#### async function login(event) {

#### event.preventDefault();

#### try {

#### await axios.post("http://localhost:8080/newlogin", {

#### username: username,

#### password: password,

#### }).then((res) =>

#### {

#### console.log(res.data);

#### if (res.data.message === "Username not exits")

#### {

#### alert("Username not exits");

#### }

#### else if(res.data.message === "Login Success")

#### {

#### navigate('/card');

#### }

#### else

#### {

#### alert("Incorrect Username and Password not match");

#### }

#### }, fail => {

#### console.error(fail); // Error!

#### });

#### }

#### catch (err) {

#### alert(err);

#### }

#### }

#### return (

#### <div className="loginbg">

#### <div className="bgimg">

#### <div className="app" >

#### <div className="login-form">

#### <div className="title">Login</div>

#### <div className="form">

#### <form id="forl">

#### <div className="input-container">

#### <input type="text" placeholder="Username" id='fan' name="uname" value={username}

#### onChange={(event) => {

#### setUsername(event.target.value);

#### }} required />

#### </div><br></br>

#### <div className="input-container">

#### <input type="password" placeholder="Password" id='fan' name="pass" value={password}

#### onChange={(event) => {

#### setPassword(event.target.value);

#### }} required />

#### </div>

#### <div>

#### <button id="sam" class="btn btn-primary" onClick={login} >Login</button>

#### </div><br></br>

#### <div></div><br></br>

#### <div>To create new account -

#### <Link to="/signup">Signup</Link>

#### </div>

#### <div>For Admin login -

#### <Link to="/adminlogin">Admin Login</Link>

#### </div>

#### </form>

#### </div>

#### </div>

#### </div>

#### </div>

#### </div>

#### );

#### }

#### export default Login;

#### Sign up:

import React, { Component } from 'react';

import "../components/signup.css";

import { useState } from "react";

import { useNavigate } from 'react-router-dom';

import axios from "axios";

function Signup() {

const [username, setUsername] = useState("");

const [email, setEmail] = useState("");

const [mobilenumber, setMobilenumber] = useState("");

const [password, setPassword] = useState("");

const [userroll, setUserroll] = useState("");

const navigate = useNavigate();

async function save(event) {

event.preventDefault();

if(username===""||email===""||mobilenumber===""||password===""||userroll==="")

{

alert("Please fill all the details")

}

else{

try {

await axios.post("http://localhost:8080/usersave", {

username: username,

email: email,

mobilenumber:mobilenumber,

password: password,

userroll:userroll

});

alert("User Registation Successfull");

navigate("/")

} catch (err) {

alert(err);

}

}

}

return (

<div className='signbg'>

<h1 id='urg'>Signup</h1>

<form id='form' action="/">

<table>

<tr>

<td><b id='cc'>Username</b></td>

<td><input type="text" placeholder='Enter your Username' id='uno' value={username}

onChange={(event) => {

setUsername(event.target.value);

}}

required/></td>

</tr>

<tr>

<td><b id='cc'>Email</b></td>

<td><input type="email" placeholder='Enter your name' id='uno'

value={email}

onChange={(event) => {

setEmail(event.target.value);

}}

required/></td>

</tr>

<tr>

<td><b id='cc'>Password</b></td>

<td><input type="password" placeholder='Enter your password' id='uno'

value={password}

onChange={(event) => {

setPassword(event.target.value);

}}

required/></td>

</tr>

<tr>

<td><b id='cc'>Mobile No</b></td>

<td>

<select id='sa'>

<option type="">+91</option>

<option type="">+81</option>

<option type="">+71</option>

<option type="">+61</option><br></br>

</select>

<input type="tel" placeholder='Enter your number' id='sab'

value={mobilenumber}

onChange={(event) => {

setMobilenumber(event.target.value);

}}

required/></td>

</tr>

<tr>

<td><b id='cc'>Userroll</b></td>

<td><input type="text" placeholder='Enter Userroll' id='uno'

value={userroll}

onChange={(event) => {

setUserroll(event.target.value);

}}

required/></td>

</tr>

<br></br>

</table>

<button id='ros' onClick={save}>Submit</button>

</form>

</div>

);

}

export default Signup;

#### Home page:

import React, { Component } from 'react';

import '../components/cards.css';

import Appbar from './ferthome';

class Card extends Component {

render() {

return (

<div className='cardhome'>

<Appbar/>

{/\* <div class="card">

<div class="imgBox">

<img src="https://images.unsplash.com/photo-1473167052083-3d31fa1f6776?crop=entropy&cs=srgb&fm=jpg&ixid=M3wzMjM4NDZ8MHwxfHJhbmRvbXx8fHx8fHx8fDE2ODQwNzU4ODl8&ixlib=rb-4.0.3&q=85" alt="New York Photo"/>

<div class="img-blur">

<a href="#">Explore </a>

</div>

</div>

<h2 class="title">

<a href="#">New York</a>

</h2>

<p class="text">

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Dolor, repudiandae. Lorem.

</p>

</div>

<div class="card">

<div class="imgBox">

<img src="https://images.unsplash.com/photo-1604604994333-f1b0e9471186?crop=entropy&cs=srgb&fm=jpg&ixid=M3wzMjM4NDZ8MHwxfHJhbmRvbXx8fHx8fHx8fDE2ODQwNzU5NzR8&ixlib=rb-4.0.3&q=85" alt="New York Photo"/>

<div class="img-blur">

<a href="#">Explore </a>

</div>

</div>

<h2 class="title">

<a href="#">Tokyo</a>

</h2>

<p class="text">

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Dolor, repudiandae. Lorem.

</p>

</div> \*/}

<div id="body">

<div id="main">

<div class="card">

<img src="https://images.pexels.com/photos/2835599/pexels-photo-2835599.jpeg?auto=compress&cs=tinysrgb&w=600" alt=""/>

<div class="info">

<h1>Heading</h1>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloribus minus incidunt id!</p>

<a href="/home" class="btn">View Fertilizers</a>

</div>

</div>

<div class="card">

<img src="https://images.unsplash.com/photo-1523741543316-beb7fc7023d8?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8Nnx8ZmFybWluZ3xlbnwwfHwwfHx8MA%3D%3D&auto=format&fit=crop&w=500&q=60" alt=""/>

<div class="info">

<h1>Heading</h1>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloribus minus incidunt id!</p>

<a href="/home" class="btn">View Fertilizers</a>

</div>

</div>

<div class="card">

<img src="https://images.pexels.com/photos/358457/pexels-photo-358457.jpeg?auto=compress&cs=tinysrgb&w=300" alt=""/>

<div class="info">

<h1>Heading</h1>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloribus minus incidunt id!</p>

<a href="/home" class="btn">View Fertilizers</a>

</div>

</div>

</div>

</div>

</div>

);

}

}

export default Card;

#### Fertilizer page:

import React from "react";

import { Link } from "react-router-dom";

import axios from "axios";

import '../components/home.css';

import ArrowBackIcon from '@mui/icons-material/ArrowBack'

export default class Deleteagri extends React.Component {

state = {

posts: [],

filterText: '',

};

componentDidMount() {

axios.get(`http://localhost:8080/agri`).then((res) => {

const posts = res.data;

this.setState({ posts });

});

}

handleFilterTextChange = (event) => {

this.setState({ filterText: event.target.value });

}

render() {

const { posts, filterText } = this.state;

// Filter the posts based on the filterText

const filteredPosts = posts.filter(post =>

post.name.toLowerCase().includes(filterText.toLowerCase())

);

return (

<div className="homebg">

<Link to="/card" id='kr'><ArrowBackIcon /></Link>

<div>

<h1> Fertilizer Details </h1>

<div class="input-group">

<input type="search" class="form-control rounded" placeholder="Search" aria-label="Search" aria-describedby="search-addon"

value={filterText} onChange={this.handleFilterTextChange}/>

<button type="button" class="btn btn-outline-primary">search</button>

</div>

<table className="table table-striped">

<thead style={{ backgroundColor: 'green', color: 'white', borderRadius: '10px' }}>

<tr>

<th>Id</th>

<th>Name</th>

<th>Cost</th>

<th>Manufactured</th>

<th>Expired</th>

<th>Order</th>

</tr>

</thead>

<tbody>

{filteredPosts.map((post, i) => (

<tr key={i}>

<td>{post.id}</td>

<td>{post.name}</td>

<td>{post.cost}</td>

<td>{post.manufactured}</td>

<td>{post.expired}</td>

<td><button id="orb"><Link to="/order" style={{ color: 'white' }}>Order</Link></button></td>

</tr>

))}

</tbody>

</table>

</div>

</div>

);

}

}

#### Order page

import { useState } from "react";

import axios from "axios";

import '../components/order.css';

import { useNavigate } from 'react-router-dom';

function Order() {

const [fertilizername, setFertilizername] = useState("");

const [quantity, setQuantity] = useState("");

const [address, setAddress] = useState("");

const [feadback, setFeadback] = useState("");

const navigate=useNavigate();

async function save(event) {

event.preventDefault();

if(fertilizername===""||quantity===""||address===""||feadback==="")

{

alert("Please fill all the details")

}

else{

try {

await axios.post("http://localhost:8080/osave", {

fertilizername: fertilizername,

quantity: quantity,

addess:address,

feadback: feadback

});

alert("Ordered Successfull");

navigate("/card")

} catch (err) {

alert(err);

}

}

}

return (

<div className="orderbg">

<div class="container mt-4" >

<div class="">

<h1>Place your order</h1>

<form>

<div class="form-group">

<label>Fertilizername</label>

<input type="text" class="form-control" id="fertilizername" placeholder="Enter Fertilizername"

value={fertilizername}

onChange={(event) => {

setFertilizername(event.target.value);

}}

/>

</div>

<div class="form-group">

<label>No of Quantity</label>

<input type="number" class="form-control" id="Quantity" placeholder="Quantity in Kg"

value={quantity}

onChange={(event) => {

setQuantity(event.target.value);

}}

/>

</div>

<div class="form-group">

<label>Address</label>

<input type="text" class="form-control" id="address" placeholder="Enter your Address"

value={address}

onChange={(event) => {

setAddress(event.target.value);

}}

/>

</div>

<div class="form-group">

<label>Feadback</label>

<input type="text" class="form-control" id="feadback" placeholder="Enter your Feadback"

value={feadback}

onChange={(event) => {

setFeadback(event.target.value);

}}

/>

</div>

<button type="submit" class="btn btn-primary mt-4" onClick={save} >conform order</button>

</form>

</div>

</div>

</div>

);

}

export default Order;

# CONCLUSION



## CHAPTER 7 CONCLUSION

This chapter tells about the conclusion that anyone can drive from the project and the learning we learnt by taking over this project.

### CONCLUSION

In conclusion, the fertilizer selling app offers numerous benefits and opportunities for both farmers and fertilizer suppliers. By leveraging the power of technology, the app streamlines the process of buying and selling fertilizers, making it more efficient, convenient, and accessible for users.

Firstly, the app provides farmers with a wide range of fertilizer options, allowing them to choose the most suitable products for their specific needs. With detailed product information, including ingredients, application rates, and customer reviews, farmers can make informed decisions and maximize their crop yields.

For fertilizer suppliers, the app opens up new avenues for reaching and connecting with potential customers. They can showcase their products to a broader audience and establish a strong online presence. The app's features, such as order management and inventory tracking, streamline their operations, making it easier to handle customer requests and fulfill orders efficiently.

Overall, the fertilizer selling app revolutionizes the way fertilizers are bought and sold, benefiting both farmers and suppliers. It improves accessibility, efficiency, and convenience, empowering farmers to make informed choices and facilitating suppliers in expanding their customer base. By leveraging technology, this app contributes to the advancement of agriculture, ultimately leading to improved productivity and sustainable farming practices.

### FUTURE SCOPE

Integration of IoT and Sensor Technology: By integrating Internet of Things (IoT) and sensor technology, the app can provide real-time data on soil conditions, weather patterns, and crop health. This information can be used to offer customized fertilizer recommendations and precise application guidelines, ensuring optimal nutrient supply to crops.

Collaboration with Agricultural Experts: The app can collaborate with agricultural experts, agronomists, and researchers to provide farmers with expert advice, best practices, and educational content. This partnership can enhance the app's credibility and support farmers in implementing sustainable and efficient fertilizer practices.

Integration with Precision Farming Technologies: The app can integrate with precision farming technologies such as GPS-enabled machinery, drones, and satellite imagery. This integration would enable farmers to precisely apply fertilizers, taking into account field variability, reducing waste, and optimizing resource utilization.

Expansion into International Markets: As the app gains traction and establishes a strong user base, it can explore expansion into international markets. This would involve incorporating region-specific fertilizers, local suppliers, and customized features to cater to the unique needs of farmers in different countries.

Sustainability and Organic Fertilizers: With the increasing emphasis on sustainable farming practices and organic agriculture, the app can introduce a dedicated section for organic fertilizers. This would cater to the growing demand for environmentally friendly and chemical-free options, allowing farmers to make sustainable choices.

Farmer Community and Knowledge Sharing: Creating a community within the app where farmers can connect, share experiences, and exchange knowledge would foster collaboration and learning. Features such as discussion forums, farmer success stories, and expert-led webinars can encourage engagement and empower farmers with valuable insights.

# REFERENCES



### REFERENCES

1. "Fertilizer management app helps farmers reduce environmental impacts" - Article by American Society of Agronomy
2. "Precision agriculture: A new fertilizer app" - Article by Farmers Weekly
3. "Digital revolution in agriculture: The rise of farming apps" - Article by Agriland
4. "The Future of AgriTech: How mobile apps are revolutionizing farming" - Blog post by Arkenea
5. "Digital Agriculture: How Apps Can Help Farmers" - Article by International Finance Corporation
6. "The Power of Apps in Agriculture" - Blog post by The Wall Street Journal
7. "Farmers are using apps to decide which fertilizers to use" - Article by AgFunder News
8. "How Technology and Apps Are Changing Agriculture" - Article by Forbes