**E-COMMERCE PLATFORM FOR FRESH FRUITS AND VEGETABLES**

**A PROJECT REPORT**

***Submitted by***

**MALATHI M (920422205058)**

**POOJA R (920422205077)**

**PRIYAJAYAM K (920422205080)**

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

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECNOLOGY**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution - Affiliated to Anna University, Chennai)**

**K.VELLAKULAM, VIRUDHUNAGAR - 625 701**

**NOVEMBER 2024**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution- Affiliated to Anna University, Chennai)**

**K.VELLAKULAM, VIRUDHUNAGAR - 625 701**

**BONAFIDE CERTIFICATE**

Certified that the project report **“ECO-FRIENDLY PRODUCT MARKETPLACES”** is the bonafide work **of “MALATHI M (920422205058), POOJA R (920422205077), PRIYAJAYAM K (920422205080)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.E.VAKAIMALAR Mrs. P.MAHALAKSHMI**

**Head of the Department, SUPERVISOR,**

Associate Professor , Assistant Professor,

Dept of Information Technology, Dept of Information Technology

Kamaraj college of Engg and Tech, Kamaraj college of Engg and Tech

K.vellakulam, K.vellakulam,

Virudhunagar-625 701 Virudhunagar-625 701

# INTERNAL EXAMINER EXTERNAL EXAMINER

**ABSTRACT**

FreshFarm is a user-friendly e-commerce platform built using the MERN stack (MongoDB, Express.js, React.js, Node.js), designed to make purchasing fresh fruits and vegetables online simple and convenient. It connects consumers directly with local farmers and suppliers, offering a wide selection of seasonal and organic produce. The platform includes features like an intuitive shopping cart, detailed product descriptions, personalized recommendations, and secure checkout, ensuring a seamless shopping experience.

With FreshFarm, users can easily browse products, save favorites, and receive recommendations based on their preferences. The platform emphasizes sustainability by supporting local farmers and reducing environmental impact, while also providing detailed information about the origin of each product. FreshFarm’s secure payment process and responsive design make it accessible and safe for users on any device, whether shopping from a phone, tablet, or desktop. By promoting fresh, local produce and supporting sustainable agriculture, FreshFarm makes healthy and eco-conscious food choices easily accessible to all consumers.

**ACKNOWLEDGEMENT**

I would like to take a moment to sincerely thank everyone who helped make my project, FreshFarm: E-Commerce Platform for Fresh Fruits and Vegetables, a success. First, I express my deepest thanks to Mrs. P. Mahalakshmi, Assistant Professor, in the Department of Information Technology, for her constant support and guidance during the development of this project. Her valuable feedback and encouragement helped me understand the complexities of the MERN stack and improve the platform. Her mentorship inspired me to overcome challenges and aim for the best.

I also want to extend my heartfelt gratitude to Dr. R. Arthy, Assistant Professor, in the Department of Information Technology for her invaluable assistance throughout this journey. Her insights and encouragement were instrumental in my growth and learning.

Additionally, I want to thank the many online resources, tutorials, and the open-source community that provided crucial knowledge on React.js, Node.js, and MongoDB. Their examples and best practices greatly helped me in building this project.

Lastly, I am grateful to my family for their continuous support and motivation throughout the journey.

Thank you all for your guidance and support in bringing FreshFarm to life.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | ABSTRACT  ACKNOWLEDGEMENT  LIST OF FIGURE | iii  iv  v |
| **1** | **INTRODUCTION**  1.1 HTML  1.2 CSS  1.3 JavaScript  1.4 MERN Stack | **1**  1  2  3  4 |
| **2** | **METHODOLOGY**  2.1 Objective  2.2 Problem Statement  2.3 Block Diagram  2.4 Module Explanation | **6**  6  6  7  8 |
| **3** | **RESULTS AND DISCUSSION** | 11 |
| **4** | **CONCLUSION** | 16 |
| **5** | **REFERENCES** | 17 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO** |
| 2.1 | Process Flow | **7** |
| 3.1 | Explore page | 12 |
| 3.2 | Home page | 12 |
| 3.3 | Vegetables detail page | 13 |
| 3.4 | Fruits detail page | 13 |
| 3.5 | Shop page | 14 |
| 3.6 | Favorite List Page | 14 |
| 3.7 | Order confirmation page | 15 |
| 3.8 | Getting Details from the user | 15 |
| 3.9 | Getting details and placing the order | 15 |
| 3.10 | Order confirmation page | 16 |
| 3.11 | User orders stored in the MongoDB | 16 |
| 3.12 | Order details stored in MongoDB | 16 |

**CHAPTER 1**

**INTRODUCTION**

**1.1. HTML**

HyperText Markup Language, is the standard language used to create web pages and web applications. It structures the content on the page using a system of tags, where each tag defines different types of content, such as text, images, links, and forms. HTML allows you to create headings, paragraphs, lists, and tables, making it easy to organize and display information on the web. It forms the foundation of all web pages, working alongside CSS for styling and JavaScript for interactivity.

In the context of this project, HTML (via JSX in React) plays a crucial role in structuring the eco-friendly marketplace. By using various semantic HTML tags like <header>, <nav>, <section>, and <footer>, we were able to organize the content in a way that is both accessible and meaningful. Additionally, interactive elements like forms (<form>, <input>, <button>) were used in the checkout process to capture customer information efficiently.

Key HTML concepts employed in this project include

* **Semantic HTML**: Helps to make the webpage more accessible by using meaningful tags, such as <header>, <main>, and <footer>.
* **Forms and Input Fields**: Used to gather user input on checkout pages, where fields like name, address, and email are captured via <input> and <form> elements.
* **Links and Navigation**: The project uses HTML anchors (<a>) and navigation elements (<nav>) for routing between different sections, ensuring smooth user transitions across pages.

**1.2 CSS (Cascading Style Sheets)**

CSS is a style sheet language used to describe the presentation of a document written in HTML. It defines how elements on a web page are displayed, including their layout, colors, fonts, spacing, and more.

**Key Features of CSS**

* **Layout Control**: CSS allows for precise control over the positioning and alignment of elements on a web page. You can create complex layouts that adapt to different screen sizes, making websites responsive.
* **Styling and Design**: CSS enables you to customize the appearance of HTML elements by applying styles such as background colors, borders, text alignment, shadows, and much more.
* **Font Management**: Easily control font families, sizes, weights, and other typography aspects to ensure readability and aesthetics.
* **Animations and Transitions**: CSS can be used to create animations, transitions, and dynamic effects, enhancing user experience and interactivity.
* **Consistency**: CSS ensures design consistency across all pages of a website. By applying global styles, you can maintain uniformity in appearance and branding.

**CSS Benefits**

* Separation of Concerns: CSS separates content (HTML) from presentation, making websites easier to maintain and update.
* Reusability: Styles can be applied across multiple web pages, reducing redundancy and effort in code.
* Accessibility: With CSS, you can improve website accessibility by providing visual cues and creating layouts that are easy to navigate**.**

**1.3 JAVASCRIPT**

**JavaScript**

JavaScript is a powerful programming language that enables web pages to become interactive and dynamic. It adds functionality to websites, allowing them to respond to user actions and update content in real time without needing to reload the page.

**Key Features of JavaScript**

* **Interactivity**: JavaScript allows you to create interactive elements on a web page, such as buttons, sliders, and forms that respond to user input like clicks or keystrokes.
* **Dynamic Content**: With JavaScript, you can modify the content of a web page dynamically. For instance, you can update text, images, or entire sections of a page based on user interaction without reloading the page.
* **Form Validation**: JavaScript can validate user input in forms before they are submitted, ensuring that the data entered is correct and complete.
* **Animations**: JavaScript is used to create smooth animations and effects like image sliders, pop-ups, and scrolling elements, improving the user experience.
* **Event Handling**: JavaScript responds to events triggered by user actions, such as clicks, hovers, or keyboard inputs, to make web pages interactive and responsive.

**1.4 MERN STACK**

MERN is a powerful full-stack JavaScript framework designed for developing web applications. The acronym MERN stands **for MongoDB, Express.js, React.js, and Node.js,** which are the four core technologies that make up this stack. Together, they provide a robust and efficient environment for building modern, dynamic web applications.

**Components of the MERN Stack**

1. MONGODB

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents rather than traditional table structures. This document-oriented approach allows for the storage of complex data types and relationships easily.

* It is schema-less, meaning you can store documents with different structures in the same collection, providing flexibility in data modeling.
* MongoDB supports horizontal scaling, making it easy to handle large volumes of data and traffic by distributing data across multiple servers.

2.EXPRESS.JS

Express.js is a lightweight web application framework for Node.js, designed to simplify the process of building web applications and APIs.

* It provides a set of features for web and mobile applications, including routing, middleware support, and template engines.
* With Express, developers can handle HTTP requests, set up middleware to process requests, and manage routes easily, allowing for a streamlined development process.

3. REACT.JS

React.js is a front-end JavaScript library developed by Facebook, used for building user interfaces, particularly single-page applications (SPAs).

* It allows developers to create reusable UI components, making it easier to manage and update complex user interfaces.
* React employs a virtual DOM, which optimizes rendering and enhances application performance by minimizing direct manipulations of the actual DOM.
* It supports one-way data binding, which helps maintain a unidirectional data flow, making the application more predictable and easier to debug.

4.NODE.JS

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It enables developers to execute JavaScript on the server side, allowing for the use of a single language (JavaScript) throughout the entire application stack.

* Node.js is known for its non-blocking, event-driven architecture, making it highly efficient for handling concurrent connections. This is particularly beneficial for real-time applications, such as chat applications or live updates.
* It has a rich ecosystem of libraries and frameworks available through npm (Node Package Manager), which simplifies the process of adding new features and functionalities.

**CHAPTER 2**

**METHODOLOGY**

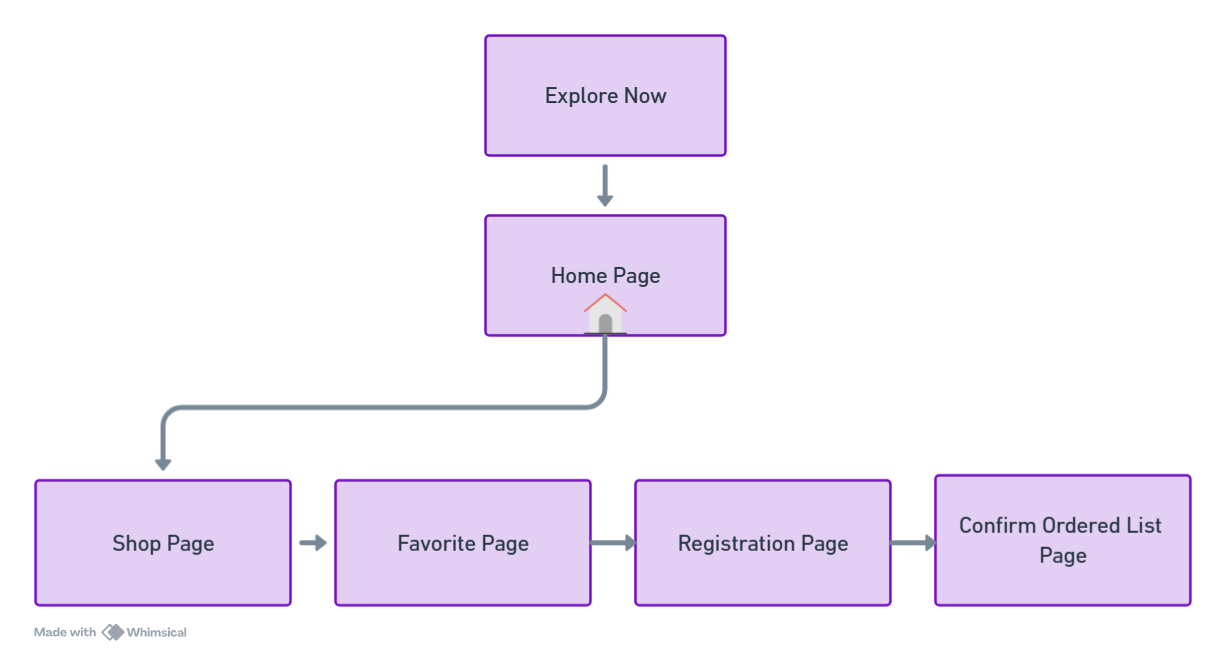
**2.1 OBJECTIVE**

The goal of the **Fruits and Vegetables Online Shopping Platform** is to make it easy for people to buy fresh produce online. The platform connects consumers with local farmers and sellers, allowing them to conveniently browse, select, and order a wide range of fruits and vegetables. This supports local growers while providing customers with quick access to fresh and healthy food.

**2.2 PROBLEM STATEMENT**

The fruits and vegetables shopping sector faces challenges in connecting consumers with fresh produce directly from farmers and local vendors. Traditional markets can be limited in availability, leading to inconveniences for consumers who may struggle to find the variety or quality they need. Current platforms often lack user-friendly features and a broad selection of fresh produce, making it difficult for users to complete their purchases efficiently. Both consumers and sellers need a streamlined, convenient platform that offers a wide range of fruits and vegetables with ease of access and delivery. The **Fruits and Vegetables Online Shopping Platform** addresses these issues by creating a central marketplace, enhancing convenience and supporting local growers, while ensuring consumers have access to fresh, high-quality produce.

**2.3 BLOCK DIAGRAM**

****



**Figure 2.1 Process Flow**

**2.4 MODULE EXPLANATION**

**1. Frontend - React (Public Folder & Src Folder)**

**Public Folder**

* **Index.html:** The main entry point for the React application where all components are injected.
* **Images:** Stores assets like product images (e.g., gram.jpg) used throughout the application.
* **favicon.ico:** The website's icon.

**Src Folder**

* **App.js:** The main file that sets up routes and renders components across different pages.
* **App.css & Index.css:** Global stylesheets for the React application, defining layouts and themes.

**Components Folder:**

* + - **navbar.js & navbar.css:** Responsible for rendering the navigation bar across the application with styles defined in the accompanying CSS file.

**Pages Folder:**

* + - **Home.js:** Displays the homepage content, showcasing fresh produce and promoting eco-friendly products.
    - **Explore.js:** The initial landing page where users are introduced to the marketplace, with engaging animations and a call-to-action.
    - **Favorite.js:** Manages and displays users' favorite items, allowing them to view or remove selections.
    - **Index.js:** Serves as an entry point for routing or setting up the structure of the various pages.
    - **OfferDetails.js:** Displays special offers and discounts on select fruits and vegetables.
    - **Shop.js:** The main shopping page, listing all available products with the ability to add items to the cart.
    - **Account.js:** Manages user account information and registrationdetails.

**Backend (Express/Node.js)**

**Models Folder**

* **Order.js:** Defines the schema for customer orders in MongoDB.
* **Product.js:** Represents the schema for products such as fruits and vegetables in the marketplace.
* **User.js:** Manages user data, including registration and login information.

**Routes Folder**

* **Orderitems.js:** Handles routes related to order management, including creating and retrieving order details.
* **Productroutes.js:** Manages routes related to fetching and storing product data.
* **UserRegistration.js:** Manages user registration and login routes, handling the logic for new users and authentication.
* **Server.js:** The main server file that initializes the Express application, sets up middleware like express.json() and cors(), and connects to MongoDB. It handles incoming API requests and interacts with the database.

**Server Setup (server.js)**

**Purpose:** Manages API requests from the frontend and interacts with the MongoDB database to process orders, products, and user data.

**Key Components**

* **Express Middleware:**
  + - express.json() Parses incoming JSON data.
    - cors() Enables cross-origin requests from the frontend to the backend.
* **MongoDB Connection:** Connects to the agrifresh\_marketplace database.
* **Order, Product, and User Schemas:** Defines models for managing user data, products, and customer orders.

**CHAPTER 3**

**RESULTS AND DISCUSSION**

The eco-friendly product marketplace project offers a clean and user-friendly design. Users can easily browse products, add them to their cart, and complete their purchase. The product details, including images and prices, are dynamically displayed, and the cart system works smoothly for managing orders. On the backend, the system effectively processes orders and stores data, ensuring it can handle more users and products in the future.

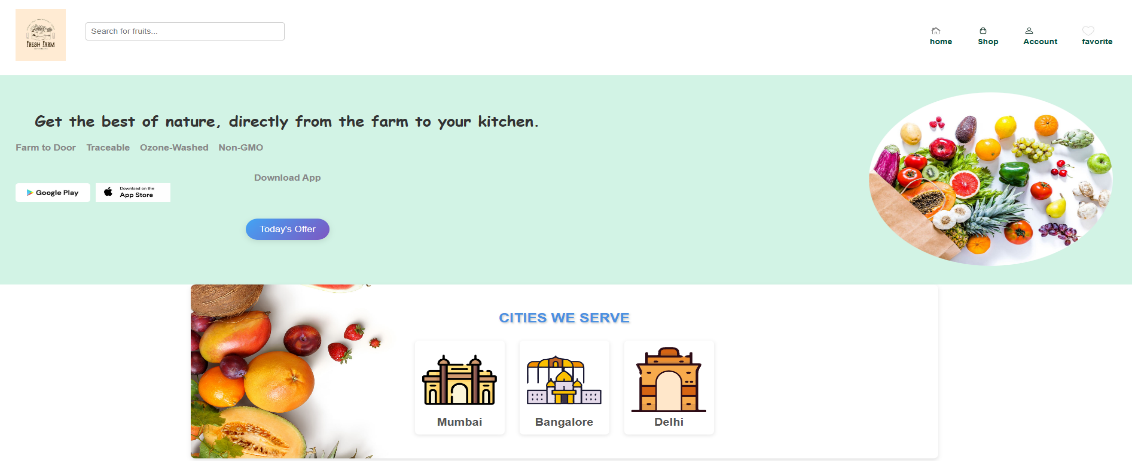
While the platform is functional, it could benefit from added features like product reviews, filtering, and personalized recommendations. Performance improvements, such as faster image loading, would enhance the user experience. Security for payment processing is a key challenge. Future updates could include rewards for sustainable purchases, AI recommendations, and community engagement to further promote eco-friendly choices.

**SCREENSHOTS**



Figure 3.1 EXPLORE PAGE

The figure 3.1 shows that on clicking the "Explore" button takes you from the Explore Page to the Home Page.



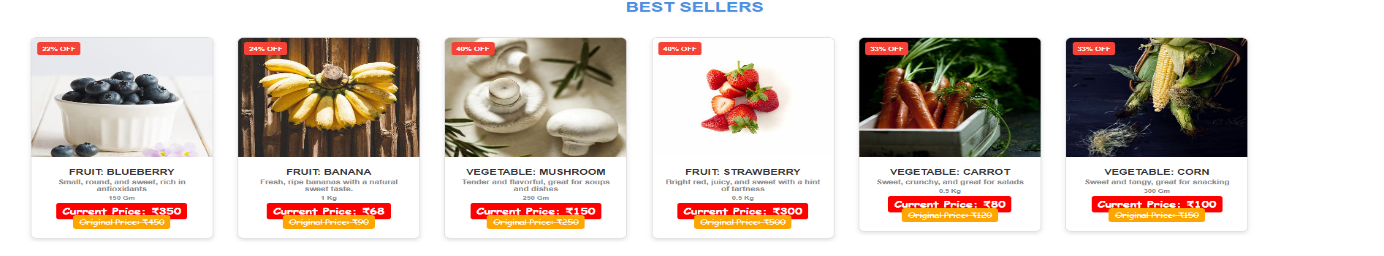




Figure 3.2 HOME PAGE

The Figure 3.2 shows the Home Page featured products and navigation options after clicking the "Explore" button.

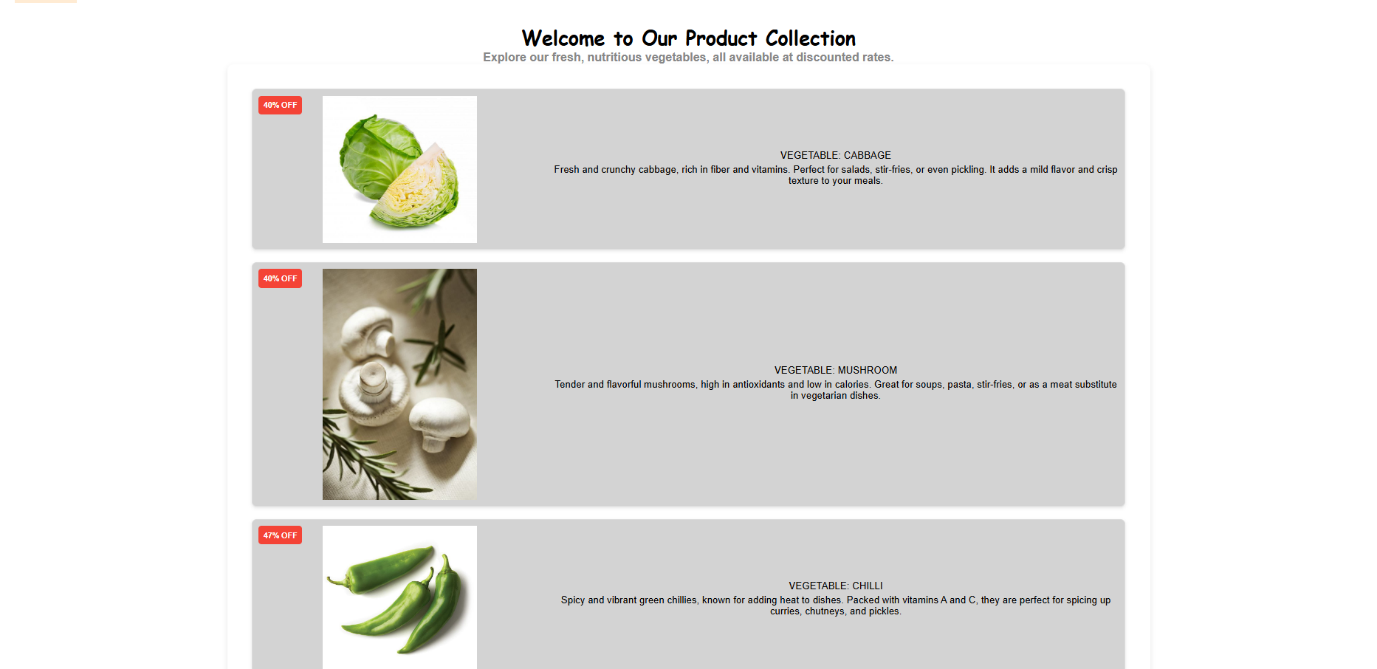


Figure 3.3 VEGETABLES DETAIL PAGE

The Figure 3.3 shows the vegetables Description and Discount.

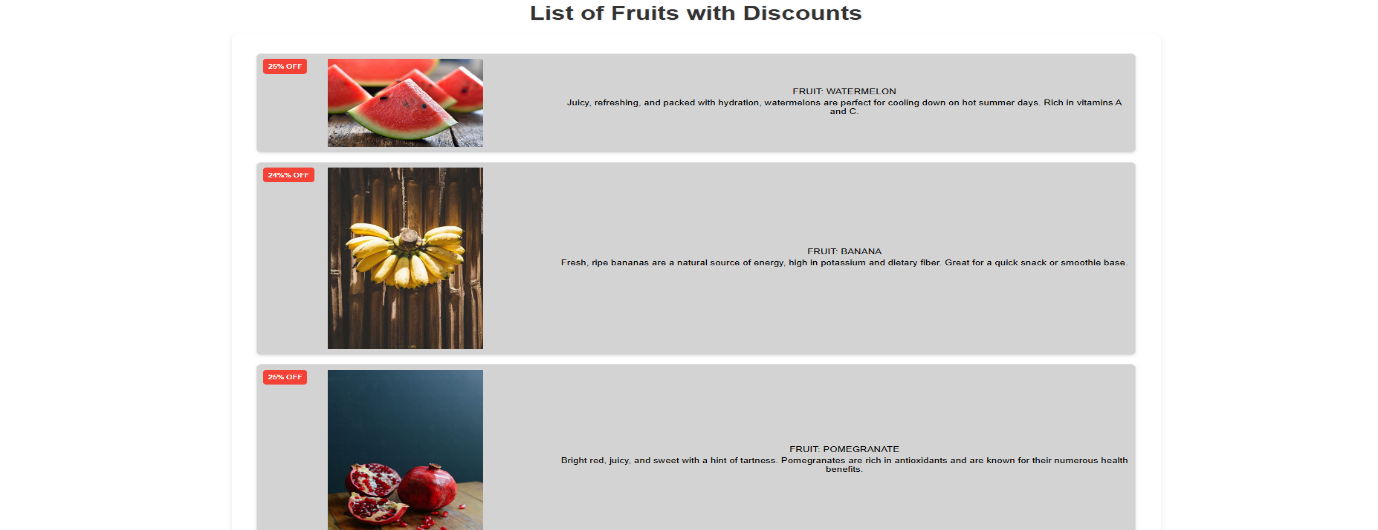
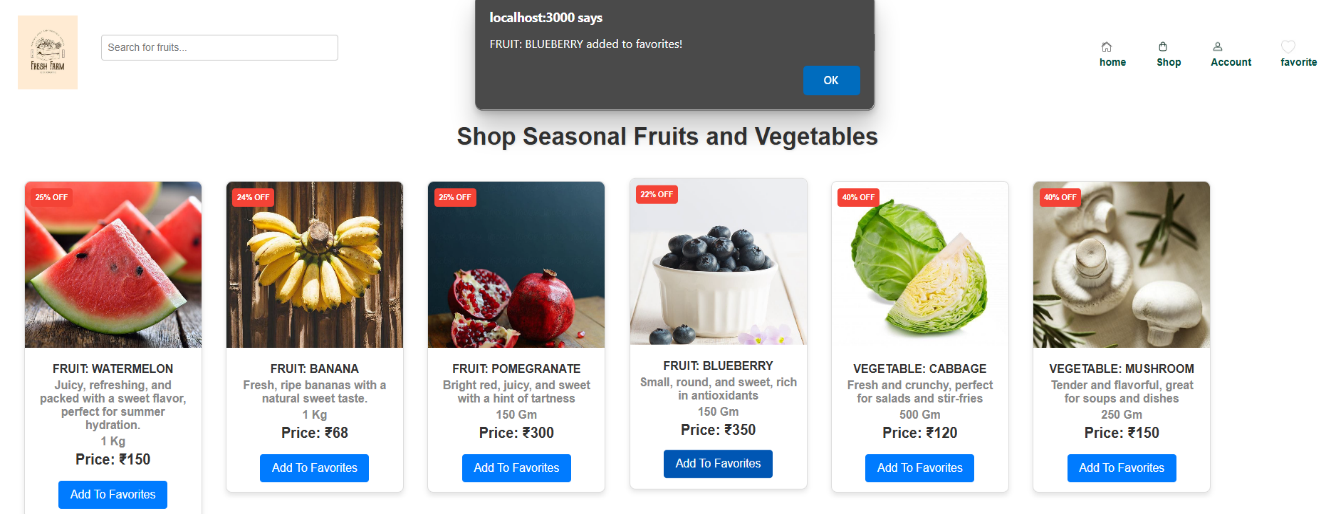


Figure 3.4 FRUITS DETAIL PAGE

The Figure 3.4 shows the Fruits Description and Discount.



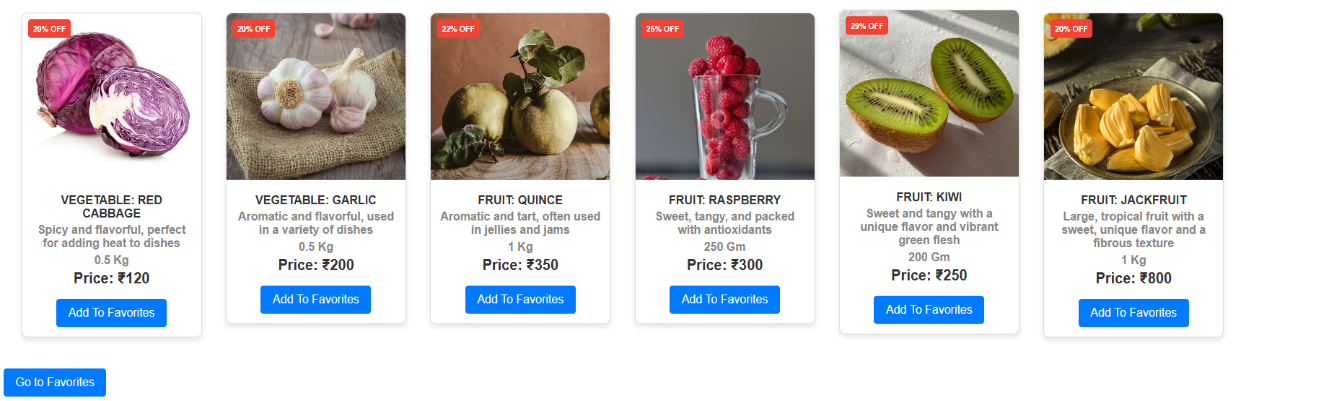


Figure 3.5 SHOP PAGE

The Figure 3.5 shows the Shop Page allows users to browse and select fruits and vegetables for their orders.

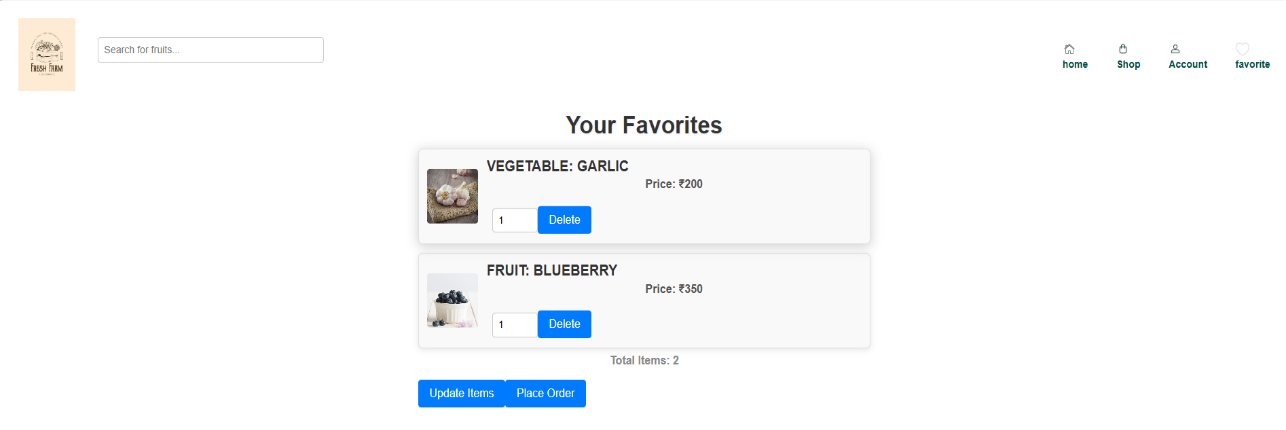


Figure 3.6 FAVORITE LIST PAGE

The Figure 3.6 shows the selected items from the shop page

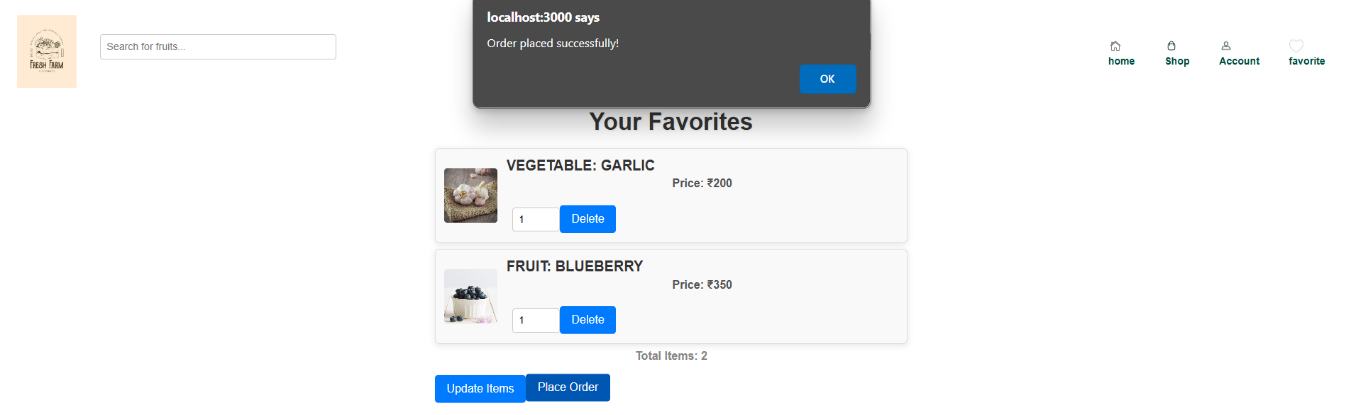


Figure 3.7 ORDER CONFIRMATION PAGE

The Figure 3.7 shows the confirmation message to the user.

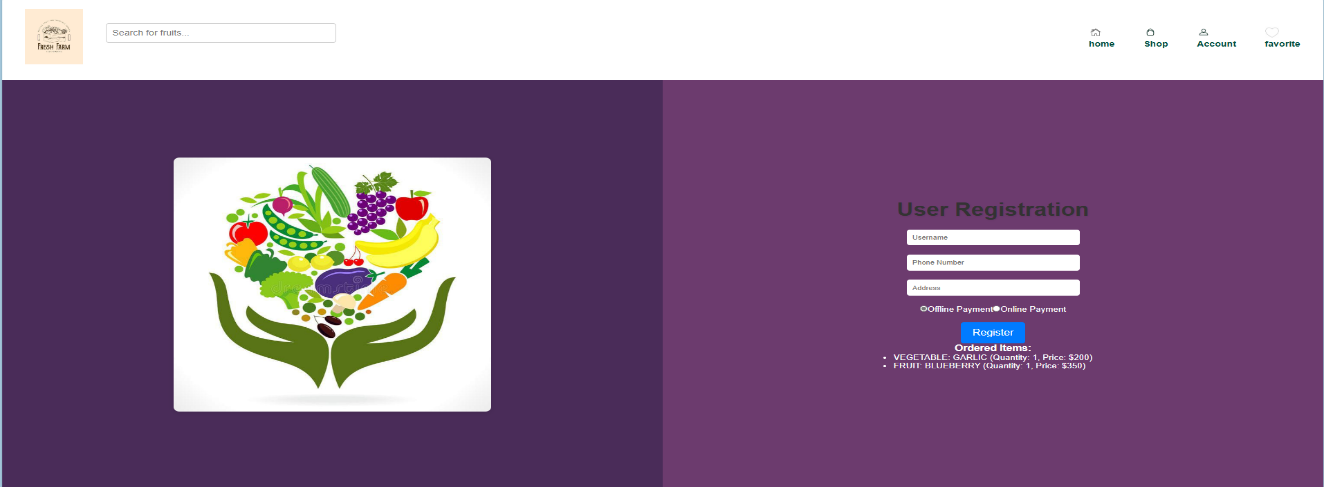


Figure 3.8 GETTING DETAILS FROM THE USER

The Figure 3.8 shows the order details page allows users to enter their deatils

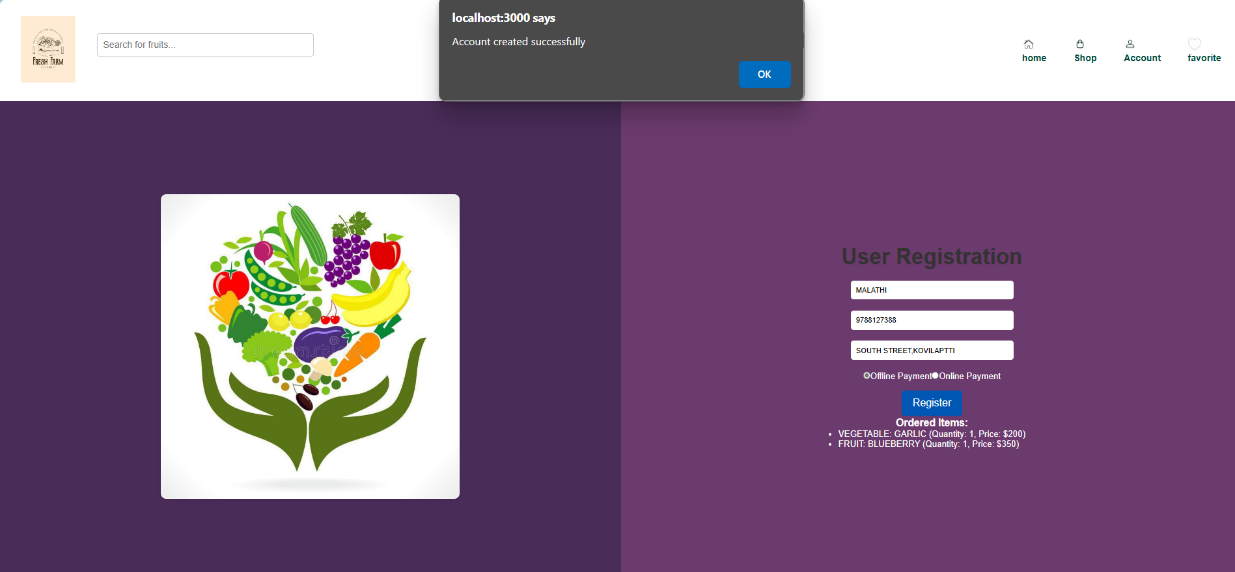


Figure 3.9 GETTING DETAILS AND PLACING THE ORDER

The Figure 3.9 shows the order details page allows users to review their selected items and complete their purchase.

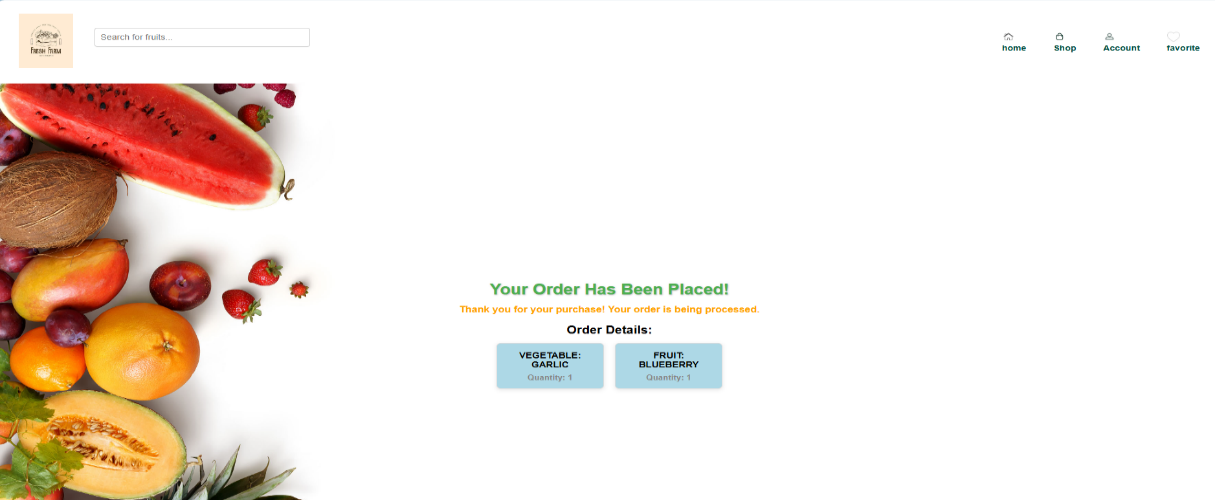


Figure 3.10 ORDER CONFIRM PAGE

The Figure 3.10 shows the Order Confirmation Page and summary of the completed order and thanks the user for their purchase.

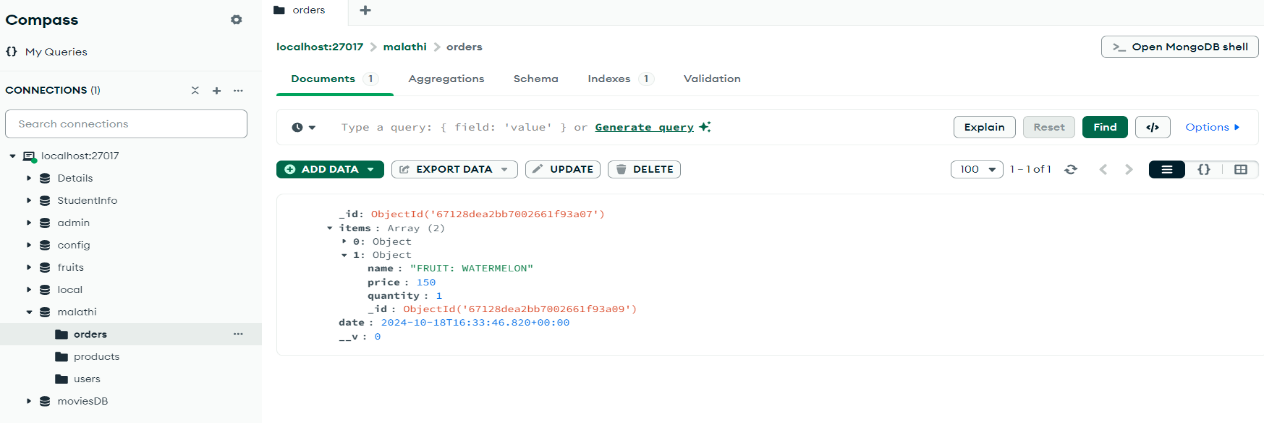


Figure 3.11 USER ORDERS STORED IN MONGODB

The Figure 3.11 shows the orders Data is stored in MongoDB to manage user orders

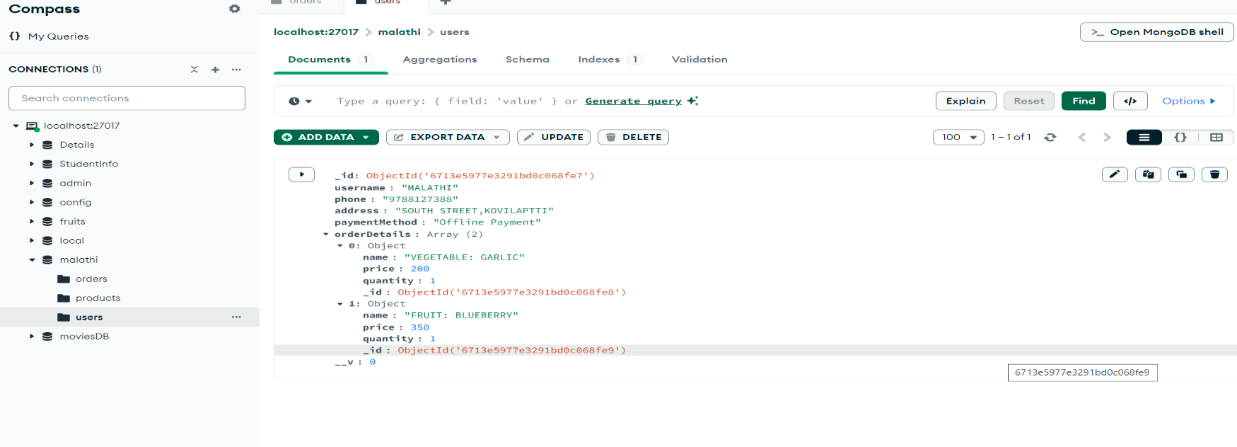


Figure 3.12 USER DETAILS STORED IN MONGODB

The Figure 3.12 shows the user details is stored in MongoDB to manage users details

**CHAPTER 4**

**CONCLUSION**

The fruits and vegetables online shopping platform makes it easy for people to buy fresh fruits and vegetables online. With a wide variety of products, personalized recommendations, and a simple ordering system, the platform helps customers find what they need quickly. It’s designed to support farmers and local businesses by making it easier for them to sell their produce.

Built using modern technologies like react.js, node.js, and mongodb, the platform is fast and can grow to meet future needs. This means it can handle more users and orders as it becomes more popular.

In summary, the fruits and vegetables online shopping platform is a helpful tool for shoppers and local farmers, promoting fresh, local produce and making online shopping easy and efficient.

**REFERENCES**

1. MongoDB Documentation

URL: [How To Use the MongoDB Shell - GeeksforGeeks](https://www.geeksforgeeks.org/how-to-use-the-mongodb-shell/)

2. Express.js Documentation

URL: [What is Express.js? | Codecademy](https://www.codecademy.com/article/what-is-express-js)

3. React Documentation

URL: [Refs and the DOM – React (reactjs.org)](https://legacy.reactjs.org/docs/refs-and-the-dom.html)

4.Node.js Documentation

URL: [About this documentation | Node.js v23.0.0 Documentation (nodejs.org)](https://nodejs.org/api/documentation.html)

5. MERN Stack Tutorial - FreeCodeCamp

URL [The MERN Stack Tutorial: The complete guide with examples - DEV Community](https://dev.to/alakkadshaw/the-mern-stack-tutorial-the-complete-guide-with-examples-2nf7)