**ECO-FRIENDLY PRODUCT MARKETPLACE - EcoMart**

# **A PROJECT REPORT**

***Submitted by***

**JAYASHREE T (920422205042)**

**SANDHIYA K (920422205090)**

**SRIDEEPALAKSHMI S (920422205106)**

***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 “JAYASHREE T (920422205042), SANDHIYA K (920422205090), SRIDEEPALAKSHMI S (920422205106)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

# **Dr. E. VAKAIMALAR** **Dr. R. ARTHY**

**Head of the Department,**  **SUPERVISOR,**

Associate Professor, Assistant Professor,

Dept. of Information Technology, Dept. of Information Technology,

Kamaraj College of Engg & Tech, Kamaraj College of Engg &Tech, K.Vellakulam, K.Vellakulam,

Virudhunagar - 625 701. Virudhunagar - 625701.

# **INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**ABSTRACT:**

The "Eco-Friendly Market System" is a comprehensive MERN stack project designed to promote sustainable commerce by connecting environmentally conscious buyers with sellers offering eco-friendly products. Built using MongoDB, Express.js, React.js, and Node.js, this platform facilitates seamless, user-friendly transactions while encouraging responsible consumption. It offers features such as product filtering based on sustainability criteria, secure payment options, and real-time order tracking. This system benefits the environment by fostering green practices, reduces the carbon footprint, and empowers consumers to make informed choices, thus contributing to a more sustainable future.

**ACKNOWLEDGEMENT:**

I would like to express my sincere gratitude to Dr. E. Vakailamar, Head of the Department of Information Technology, for their continuous encouragement and valuable insights throughout the development of the "Eco-Friendly Market System" project. Special thanks to Dr. R. Arthy, my supervisor, whose guidance and expertise were instrumental in shaping the direction of this project. Their support helped me gain in-depth knowledge of the MERN stack and its application in creating sustainable, eco-friendly solutions for modern marketplaces. This project has been an invaluable learning experience, enhancing both my technical and problem-solving skills.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | [**ABSTRACT**](bookmark://_Abstract)  **LIST OF FIGURES** | iii  vi |
| **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 | **7**  7  7  7  8 |
| **3** | **RESULTS AND DISCUSSION** | **11** |
| **4** | **CONCLUSION** | **16** |
| **5** | **REFERENCES** | **17** |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIGURE NO.** | **TITLE** |  | **PAGE NO.** |
| 2.1 | Block Diagram |  | 7 |
| 3.1 | Home page |  | 12 |
| 3.2 | Products page |  | 12 |
| 3.3 | Cart page |  | 13 |
| 3.4 | About page |  | 13 |
| 3.5 | Getting details and placing order |  | 14 |
| 3.6 | Order Confirmation |  | 14 |
| 3.7 | Storing data in MongoDB |  | 15 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 HTML:**

**HTML** (HyperText Markup Language) is the standard markup language used to create the structure of web pages. It defines the content of a web page by using elements like headings, paragraphs, images, links, and forms. HTML provides the skeleton or foundation for all web content, allowing browsers to interpret and display the structure to users.

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:**

**CSS** (Cascading Style Sheets) is a design language used to enhance the visual presentation of HTML elements. In this eco-friendly marketplace project, **CSS** was crucial for implementing the site's green, sustainable theme. By applying a combination of global styles, reusable class-based styles, and component-specific customizations, the project achieves a cohesive look and feel aligned with the eco-conscious branding.

#### **CSS Implementation**

**1. Theme Customization:** The overall theme of the website reflects environmental sustainability. A green color palette is consistently applied to backgrounds, buttons, and headers, symbolizing nature and eco-friendliness.

**2. Layout and Responsiveness:** CSS Flexbox and Grid layout techniques were extensively used to build responsive and flexible designs, ensuring the marketplace works well on all screen sizes.

**3. Typography:** Typography plays a crucial role in establishing a clean, eco-friendly aesthetic:

* **Font Choices**: Simple and modern sans-serif fonts (such as Arial or Roboto) were chosen to keep the design minimalistic.

**4. Buttons and Interactions:** CSS provides styling for interactive elements like buttons. The buttons are designed to be visually appealing and easy to interact with Hover Effects and Rounded Corners.

**1.3 JAVASCRIPT:**

**JavaScript** is a high-level, versatile programming language that enables interactive and dynamic features on web pages. It plays a crucial role in making web applications responsive, providing a seamless user experience by dynamically updating content and handling user inputs without requiring a full page reload.

In this eco-marketplace project, **JavaScript** is primarily used through **React.js**, a popular JavaScript library for building user interfaces. React.js efficiently manages the state of the application, handling real-time updates to the shopping cart, user interactions, and form submissions. Some of the key JavaScript features in this project include:

* **Shopping Cart Functionality**: JavaScript enables users to add or remove items from the cart dynamically.
* **Page Navigation**: React Router is used to handle navigation between pages like "Products," "Cart," and "Checkout." This provides a smooth, single-page application (SPA) experience, where users can move between different parts of the website without page refreshes.
* **Checkout Process**: JavaScript manages the entire flow of form handling during checkout, including input validation for fields such as name, address, and email. It ensures that the order is correctly submitted to the server and provides feedback to the user about the success or failure of their order.
* **Event Handling**: The "Buy Now" and "Add to Cart" buttons are powered by JavaScript, allowing real-time responses to user clicks. When a user interacts with these buttons, JavaScript functions are triggered to execute the appropriate actions, such as redirecting the user or adding products to their cart.

**1.4 MERN STACK:**

The **MERN stack** is a popular web development framework that uses **MongoDB**, **Express.js**, **React.js**, and **Node.js** to build full-stack applications. It allows developers to use JavaScript throughout the entire development process, covering the client-side, server-side, and database management.

#### **Components of the MERN Stack**

1. **MongoDB**:
   * A NoSQL database that stores data in flexible, JSON-like formats.
   * It supports horizontal scaling and handles large volumes of data efficiently.
   * MongoDB's rich query language and indexing enable fast data retrieval.
2. **Express.js**:
   * Express simplifies server-side development by managing routing, middleware, and request handling.
   * Developers can set up middleware to serve static files, handle HTTP requests, and manage errors.
3. **React.js**:
   * A JavaScript library for building user interfaces, especially single-page applications (SPAs).It uses a virtual DOM to optimize UI rendering, ensuring that only the necessary parts of the page are updated.
4. **Node.js**:
   * A runtime environment for executing JavaScript on the server side.
   * It is event-driven and non-blocking, making it ideal for handling multiple concurrent connections.
   * Node.js has a rich ecosystem of libraries and tools through npm (Node Package Manager), which speeds up development.

### **How the MERN Stack Works**

1. **Client-Side (React)**:
   * Users interact with the React front end, which renders the UI and handles user input.
   * When actions like form submissions occur, React captures these events and triggers functions to manage the logic.
2. **Communication (Axios/Fetch API)**:
   * React sends and receives data using HTTP requests via tools like Axios or Fetch.
   * For example, when data is needed (like loading products), React makes API calls to the Express server.
3. **Server-Side (Express and Node.js)**:
   * The Express server handles incoming HTTP requests and processes business logic.
   * Express routes manage various application functionalities like user authentication and data management.
4. **Database Interaction (MongoDB)**:
   * The server interacts with MongoDB via Mongoose to fetch or store data.
   * MongoDB organizes data in collections, and Mongoose enforces schema validation.

**CHAPTER 2**

**METHODOLOGY**

**2.1 OBJECTIVE:**

The objective of this project is to develop a dynamic, scalable eco-friendly product marketplace using the MERN stack to facilitate user interactions, product listings, and secure transactions while promoting sustainable consumer practices.

**2.2 PROBLEM STATEMENT:**

The current online marketplaces often lack a dedicated platform for eco-friendly products, making it difficult for consumers to find and support sustainable brands, while also failing to provide sellers with the visibility and tools needed to reach environmentally conscious customers effectively.

**2.3 BLOCK DIAGRAM:**

**Figure 2.1 Block Diagram**

The figure 2.1 shows the block diagram of the proposed system.

**2.4 MODULE EXPLANATION:**

1. **Frontend - React(Public Folder)**:
   * index.html: The main HTML file for your React application. This is the entry point, and React components will be injected into this file.
   * manifest.json: This file provides metadata about your web app, crucial for things like Progressive Web Apps (PWA).
   * images/: Contains image assets such as steel-straw.jpg, water-bottle.jpg. These might represent eco-friendly products.
   * favicon.ico: The website's favicon.
2. **Src Folder**:

assets/eco-friendly.jpg: This might be an image representing eco-friendly products or branding.

* + **Components**:
    - **Footer.js**: A reusable component for the website footer, likely containing navigation links or legal information. Given your eco-friendly theme, this might have a design in line with sustainability (dark background, white text).
    - **navbar.js**: A reusable component for the website's navigation bar, typically containing links to different sections (home, products, cart, etc.).
    - **ProductCard.js**: This component likely displays individual products with details such as images, names, prices, etc., and may include an "add to cart" button.
  + **Pages**:
    - **About.js**: This page probably provides information about the marketplace, its mission, and commitment to eco-friendly products.
    - **Cart.js**: This handles the shopping cart logic, displaying items that users have added, along with prices and totals.
    - **Checkout.js**: Responsible for managing the checkout process, this would include input fields for shipping details and order confirmation.
    - **Home.js**: Likely the homepage, this might feature highlighted eco-friendly products and promotional content.
    - **Products.js**: This page would display a list of available eco-friendly pr­­­­oducts, probably using the ProductCard component.
* **Context**:
  + - **CartContext.js**: This file seems to handle global state management for the shopping cart. It will provide a context that other components can access to update the cart across the application.
  + **Styles**:
    - **App.css**: Contains global styles for the React app. Since you're using a dark background with white fonts, the CSS here will reflect that.

1. **Backend (Express/Node.js)**
   * **Models**:
     + **CustomerOrder.js**: This file likely defines the structure of the customer orders in the MongoDB database.
   * **Index.js**: The main entry point for the Express.js backend. This file likely sets up the server and routes for API endpoints.
   * **Order.js**: This module probably handles order-related operations, such as creating new orders, retrieving past orders, etc.
   * **server.js**: This is where the Express server is initialized, and it listens for incoming requests. It also connects to the MongoDB database.
2. **Server Setup** (server.js):
   * **Purpose**: Handles incoming API requests and interacts with the database.
   * **Key Components**:
     + **Express Middleware**: express.json() Parses incoming request bodies in JSON format. cors() Enables Cross-Origin Resource Sharing, allowing frontend to interact with the backend.
     + **MongoDB Connection**: Connects to a MongoDB database named eco\_marketplace.
     + **Order Schema**: Defines an Order model with fields for name, address, email, and createdAt (timestamp of order creation).

**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:**

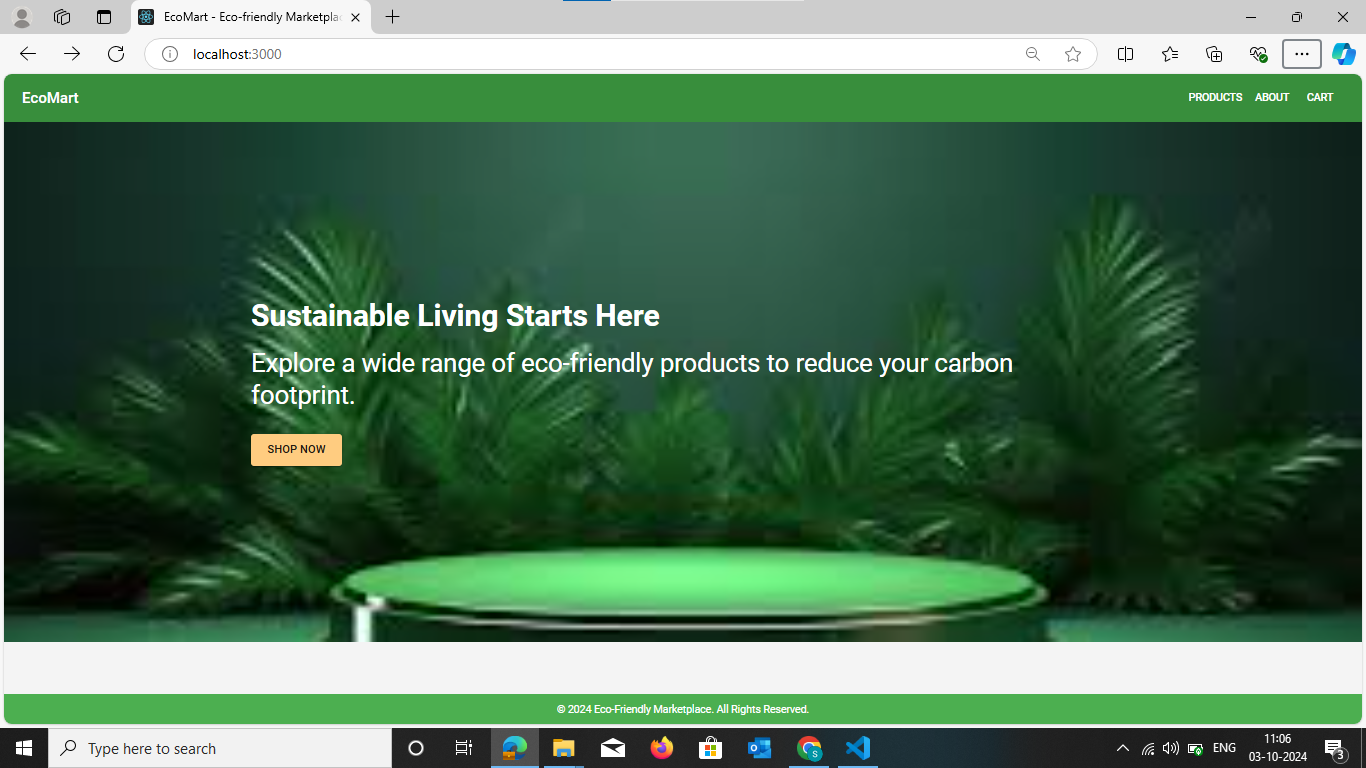
****

Figure 3.1 Home page

This figure 3.1 shows the EcoMart homepage, introducing users to sustainable living with a shopping option.

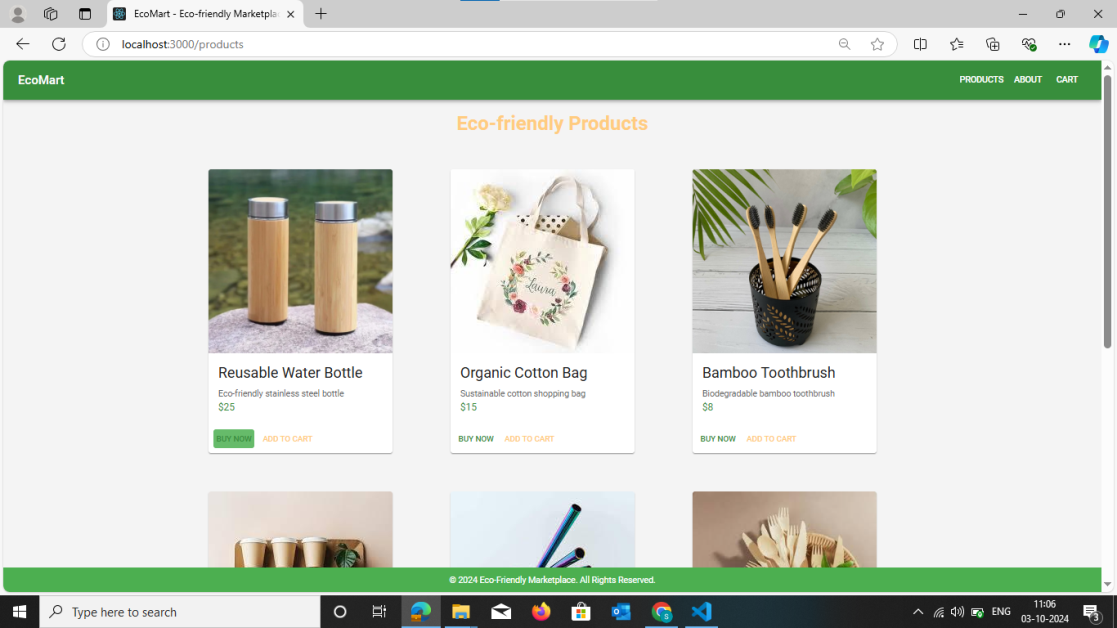
****­

Figure 3.2 Products page

The figure 3.2 shows products page showcases eco-friendly items like reusable bottles and bamboo toothbrushes, with options to add to cart or buy now.

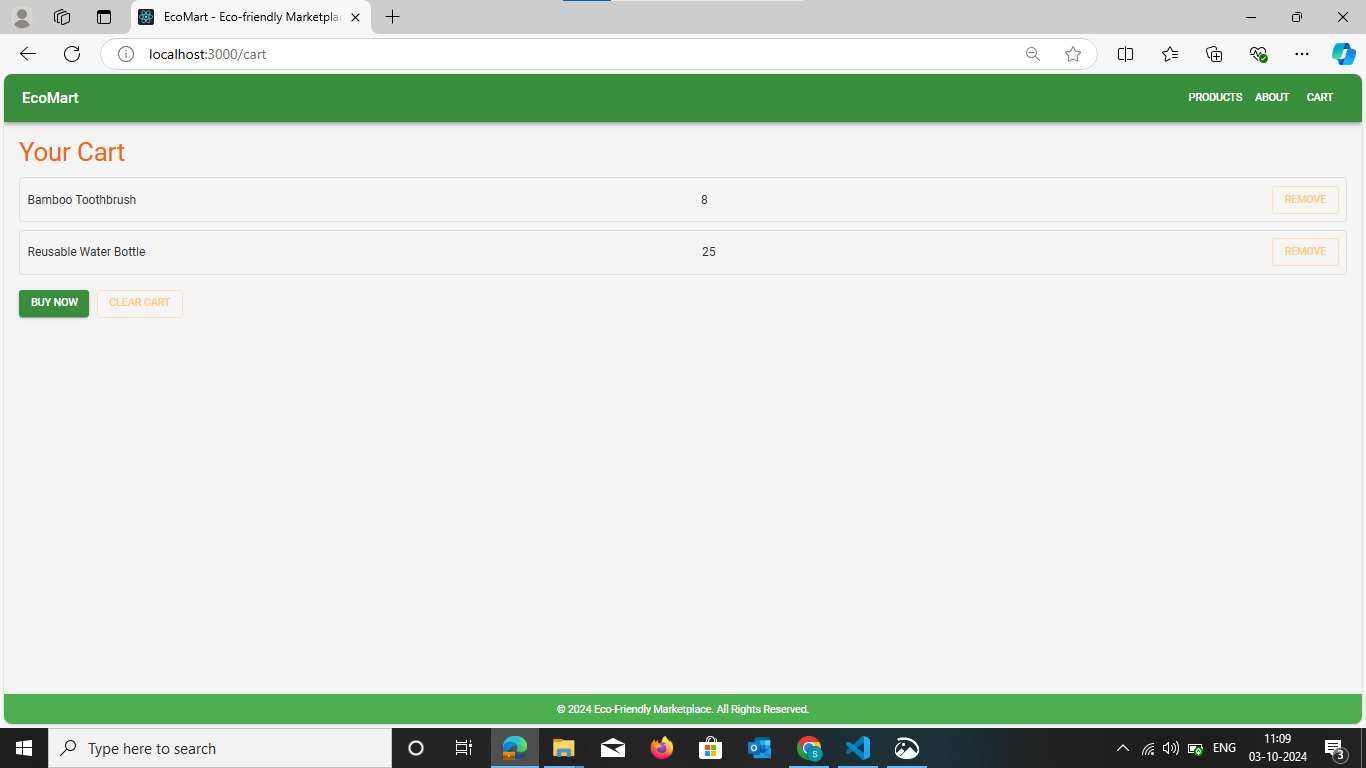
****

Figure 3.3 Cart page

The figure 3.3 shows cart page summarizes the user's selected items, with the option to remove or proceed with purchasing the eco-friendly products in the cart.

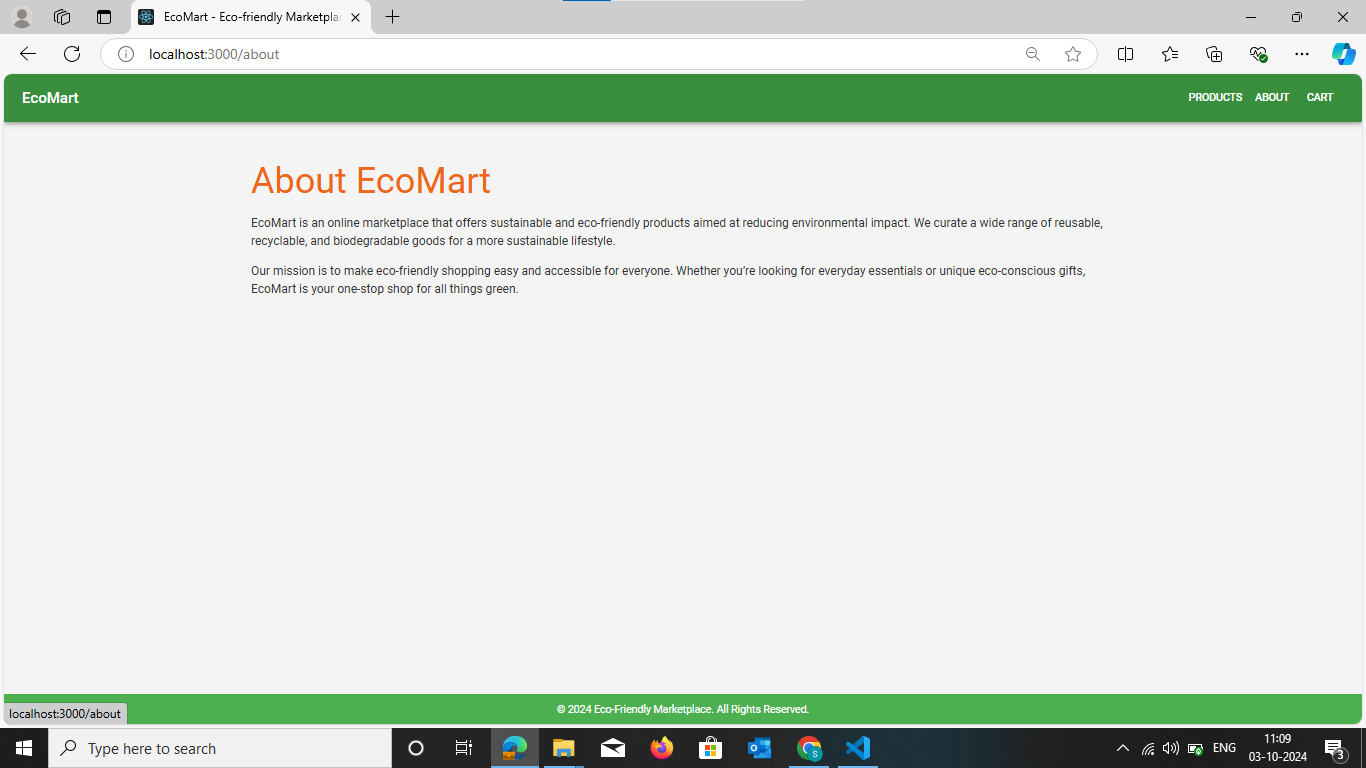
****

Figure 3.4 About page

The figure 3.4 shows the "About" page introduces EcoMart as a platform for sustainable and eco-friendly products.

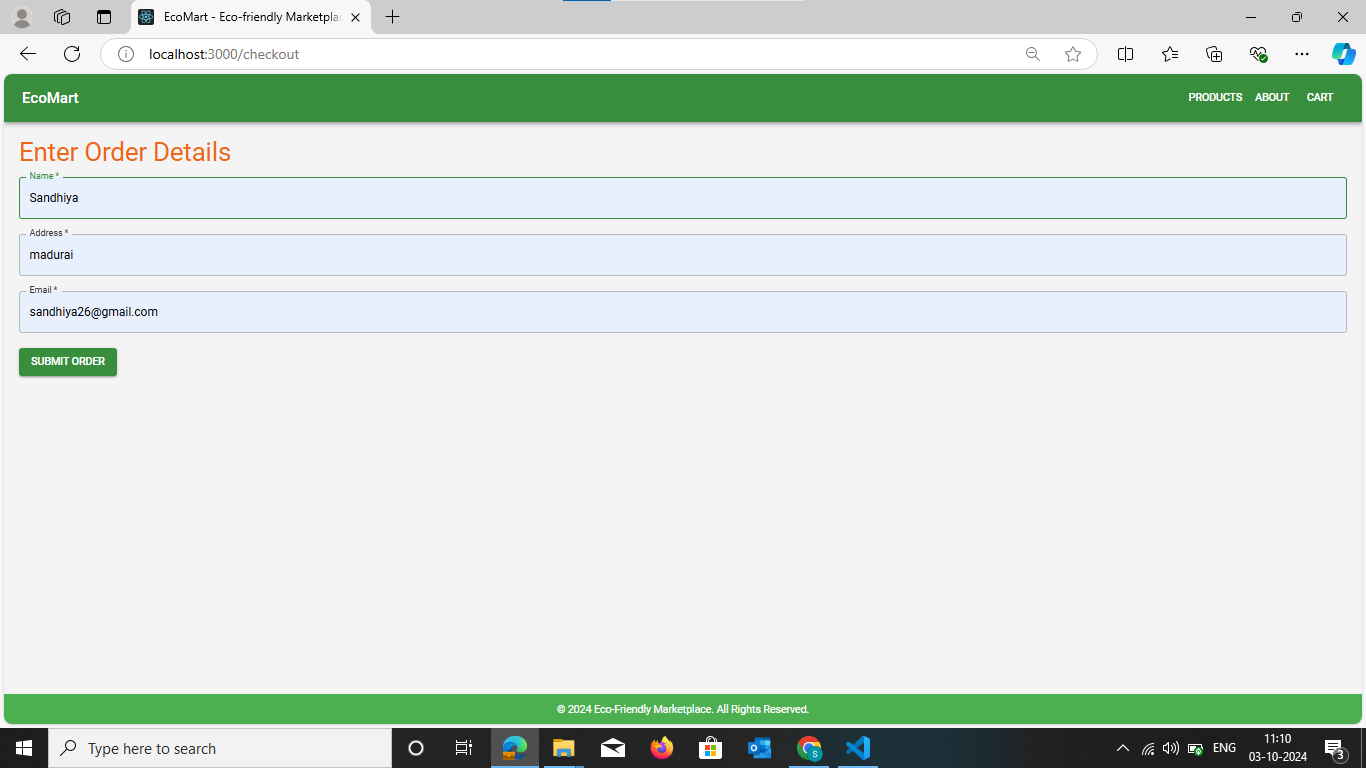
****

Figure 3.5 Getting details and placing order

The figure 3.5 shows the page allows the user to enter order details such as name, address, and email.

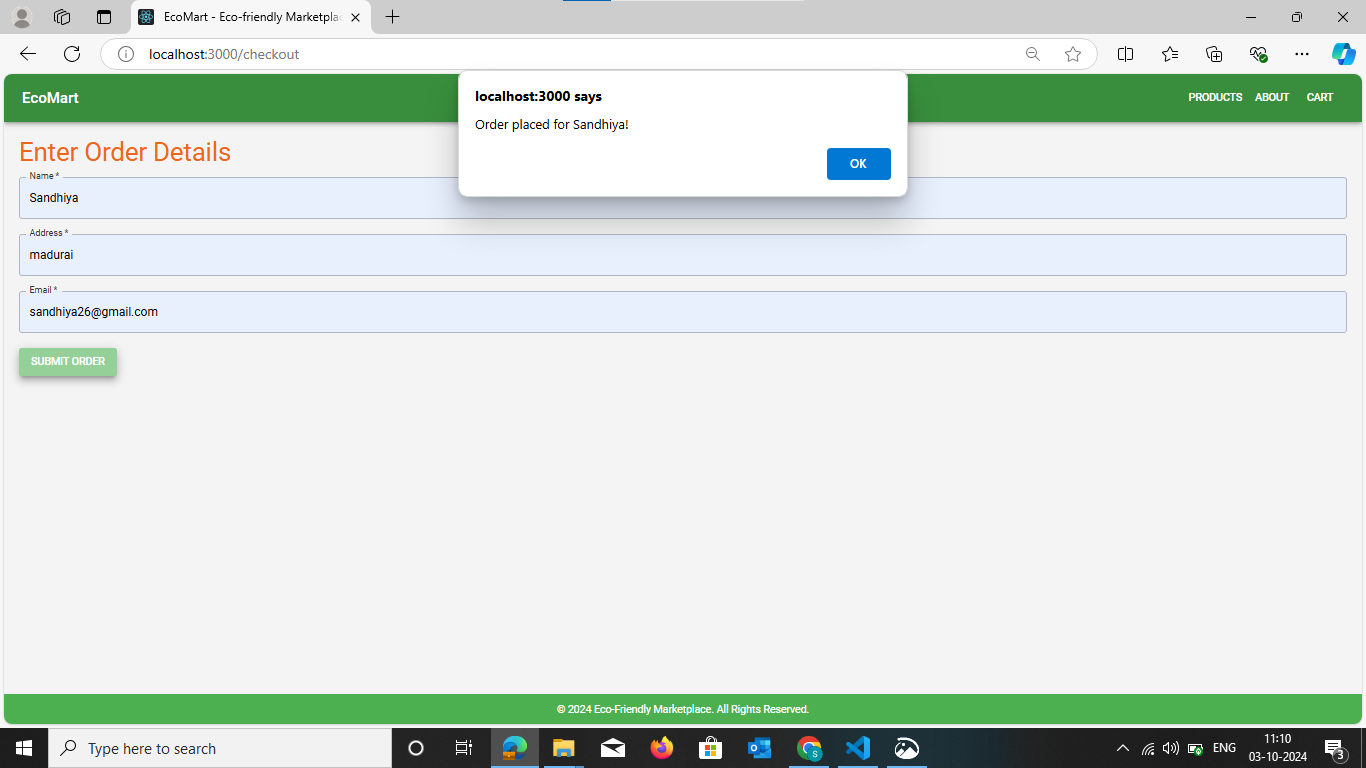
****

Figure 3.6 Order Confirmation

The figure 3.6 shows confirmation pop-up, order has been successfully placed.

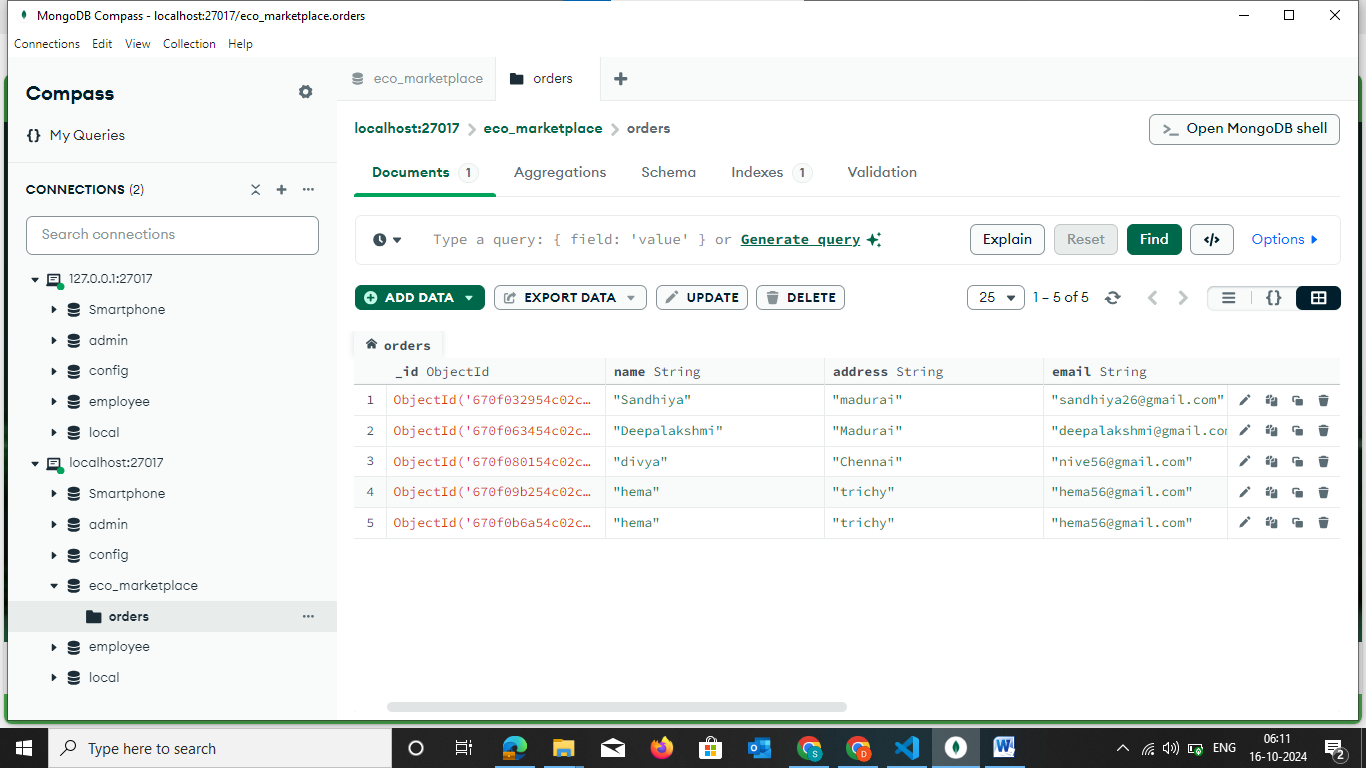
****

Figure 3.7 Storing data in MongoDB

The figure 3.7 shows user order details, including name, address, email, and the timestamp for when the order was created.

**CHAPTER 4**

**CONCLUSION**

In conclusion, the development of the Eco-Friendly Product Marketplace- EcoMart using the MERN stack has successfully created a dynamic platform that not only facilitates the buying and selling of sustainable products but also raises awareness about eco-conscious consumerism. The project highlights the importance of user engagement, security, and performance, ensuring a seamless experience for both buyers and sellers. By leveraging modern web technologies, this marketplace has positioned itself as a leader in promoting environmentally friendly practices and has laid the groundwork for future enhancements, such as personalized recommendations and expanded product offerings, further solidifying its role in the growing market for eco-friendly goods.

**REFERENCES:**

1. [Uncovering Sustainability Insights from Amazon’s Eco-Friendly Product Reviews for Design Optimization](https://www.mdpi.com/2071-1050/16/1/172) - MDPI
2. The Origins, Evolution, Current State, and Future of Green Products and Consumer Research: A Bibliometric Analysis - MDPI
3. [The Impact of Eco-Friendly Product Labels on Consumer Purchase Behavior](https://www.researchgate.net/publication/335166987_The_Impact_of_Eco-Friendly_Product_Labels_on_Consumer_Purchase_Behavior) - ResearchGate
4. Green Marketing Strategies for Eco-Friendly Products - Academia.edu
5. Eco-Friendly Consumer Behavior: Analyzing Consumer Attitudes and Purchase Intentions – Kaggle.