**PROJECT POC**

**Table of Contents**

* Project Title
* Introduction
* Technologies Used
* Workflow
* Key Features
* Future Scope
* Conclusion

**Project Title:** Online Pizza Ordering

**Introduction:**

This is an online pizza ordering system that allows users to browse a variety of pizzas, add them to their cart, and place orders conveniently from the comfort of their homes. The platform caters to both administrators, who manage the pizza inventory and orders, and end-users, who browse and order pizzas.

**Technologies Used:**

**React.js:** Frontend development framework for building dynamic and interactive user interfaces.

**CSS:** Styling language used to design and layout the user interface elements.

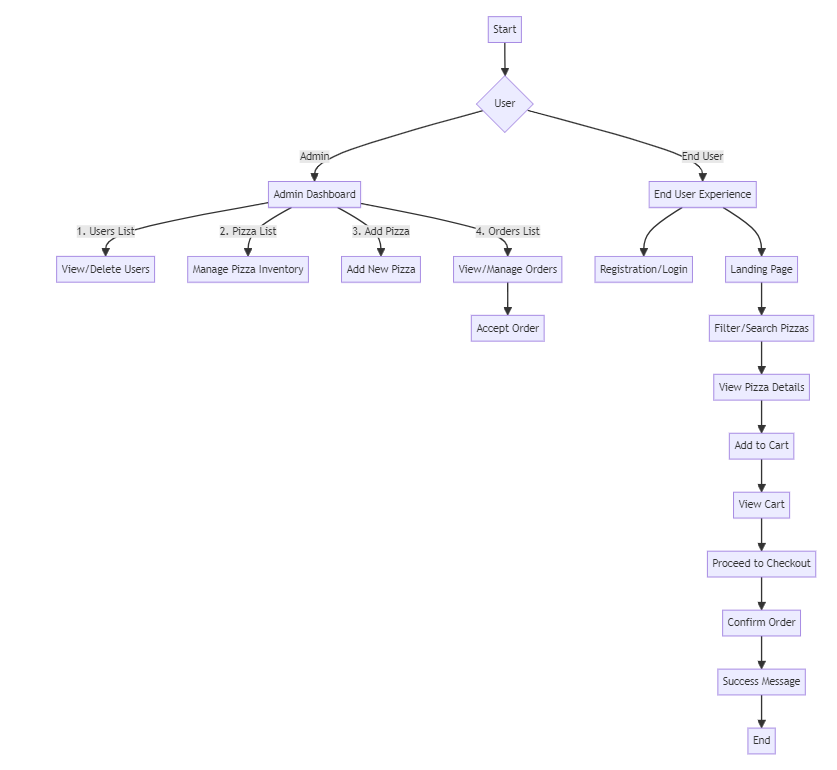
**HTML:** Markup language for structuring web pages.

**React-Redux:** State management library for managing application state and data flow.

**Node.js:** A runtime environment for executing JavaScript code server-side, Node.js powers the backend of the Online Cart website. It enables efficient handling of server-side logic and facilitates communication with the database.

**Express.js:** A web application framework for Node.js, Express.js simplifies the process of building robust and scalable web applications. It provides features for routing, middleware integration, and handling HTTP requests and responses.

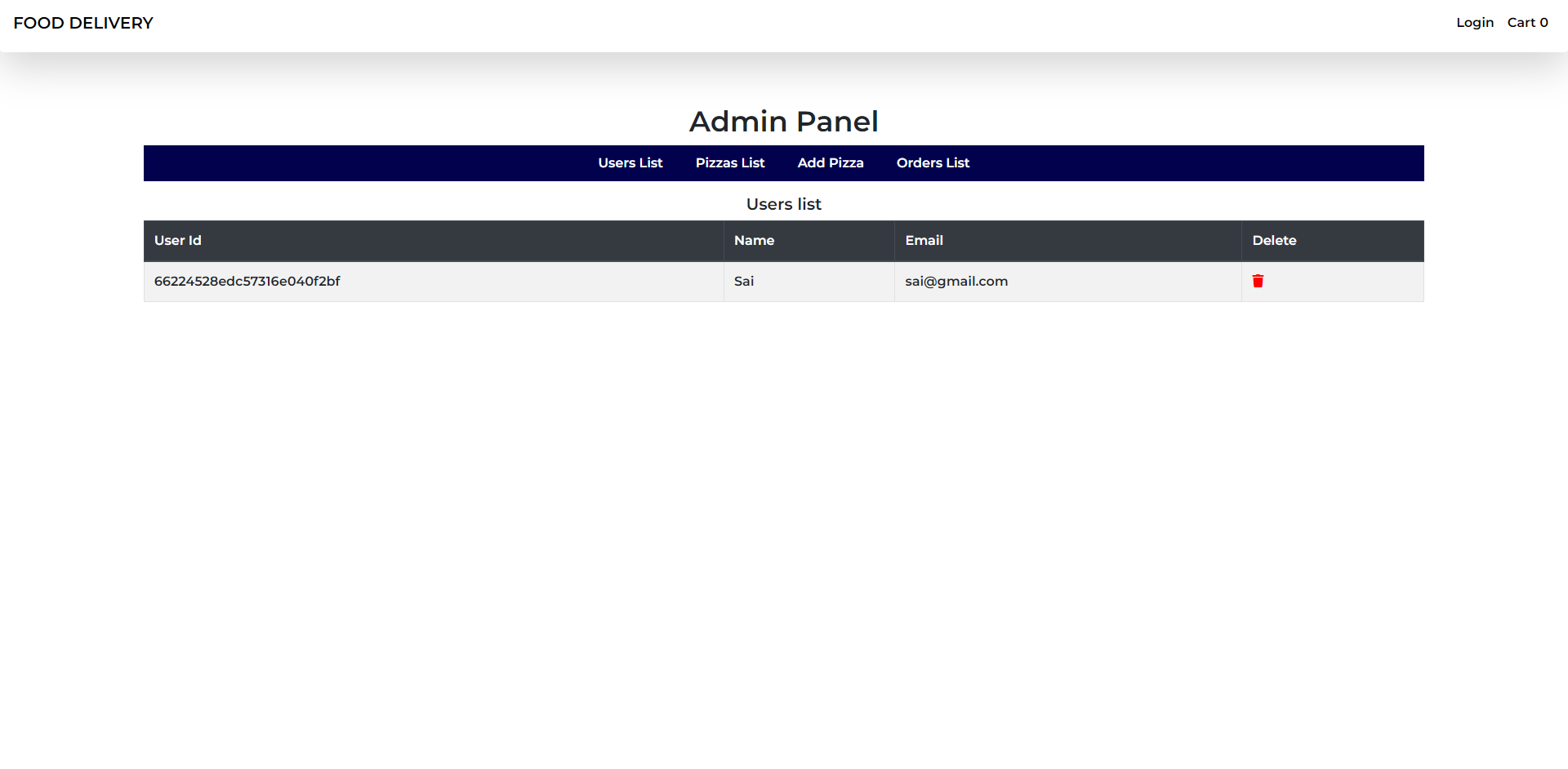
**Workflow:**

****

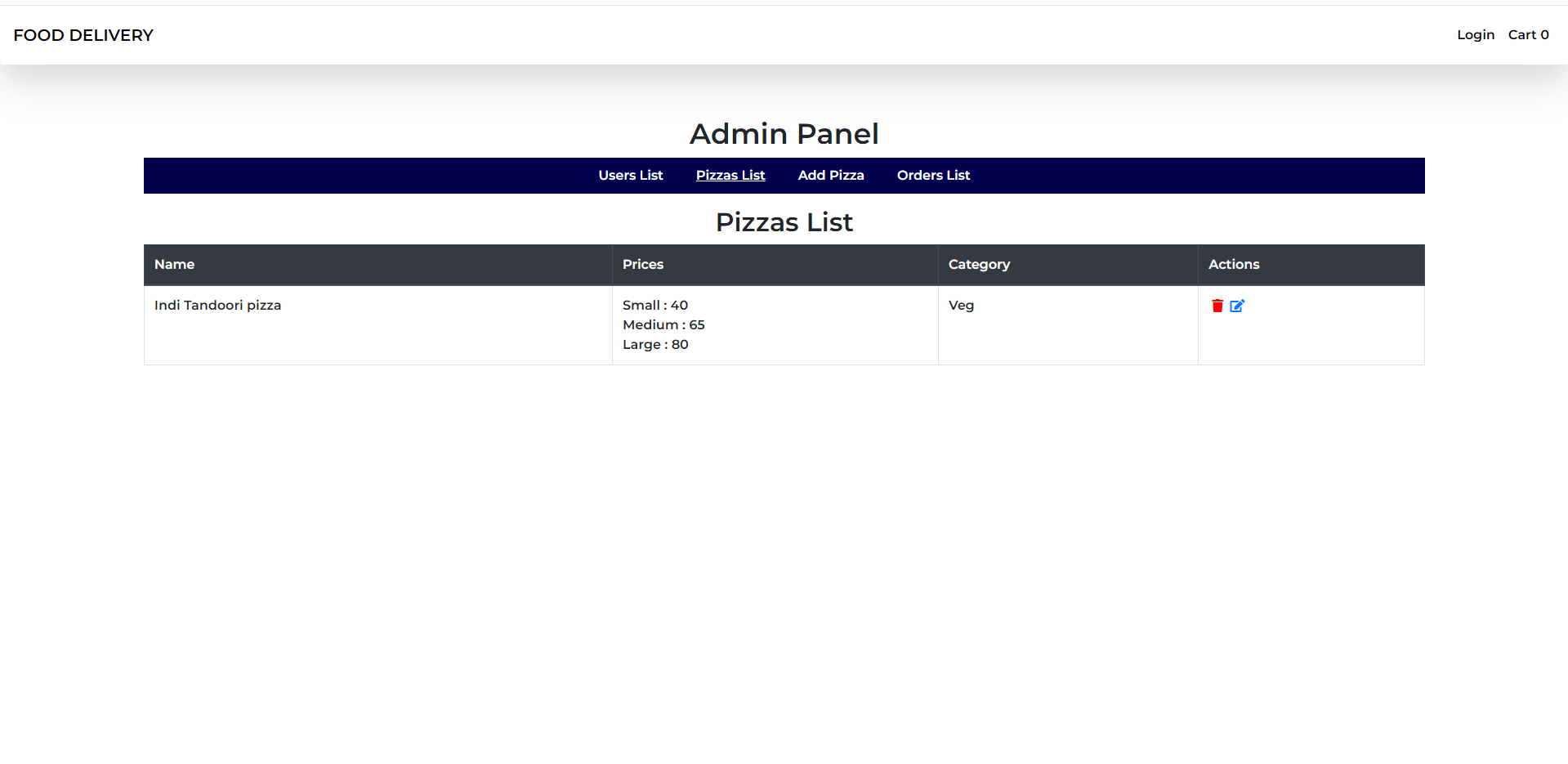
**Key features:**

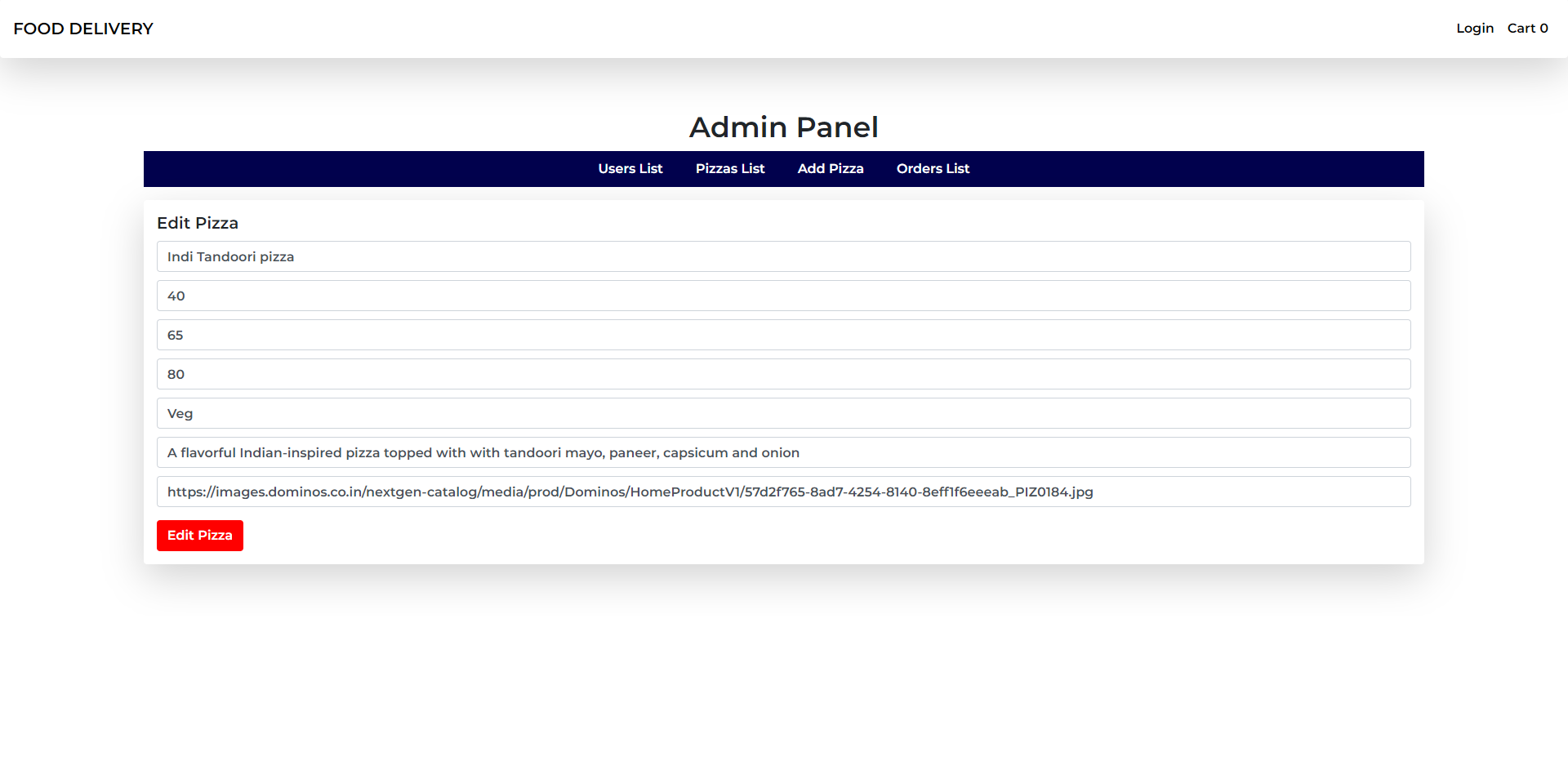
**Admin:**

**Users List:** View registered users' details and delete users if necessary.



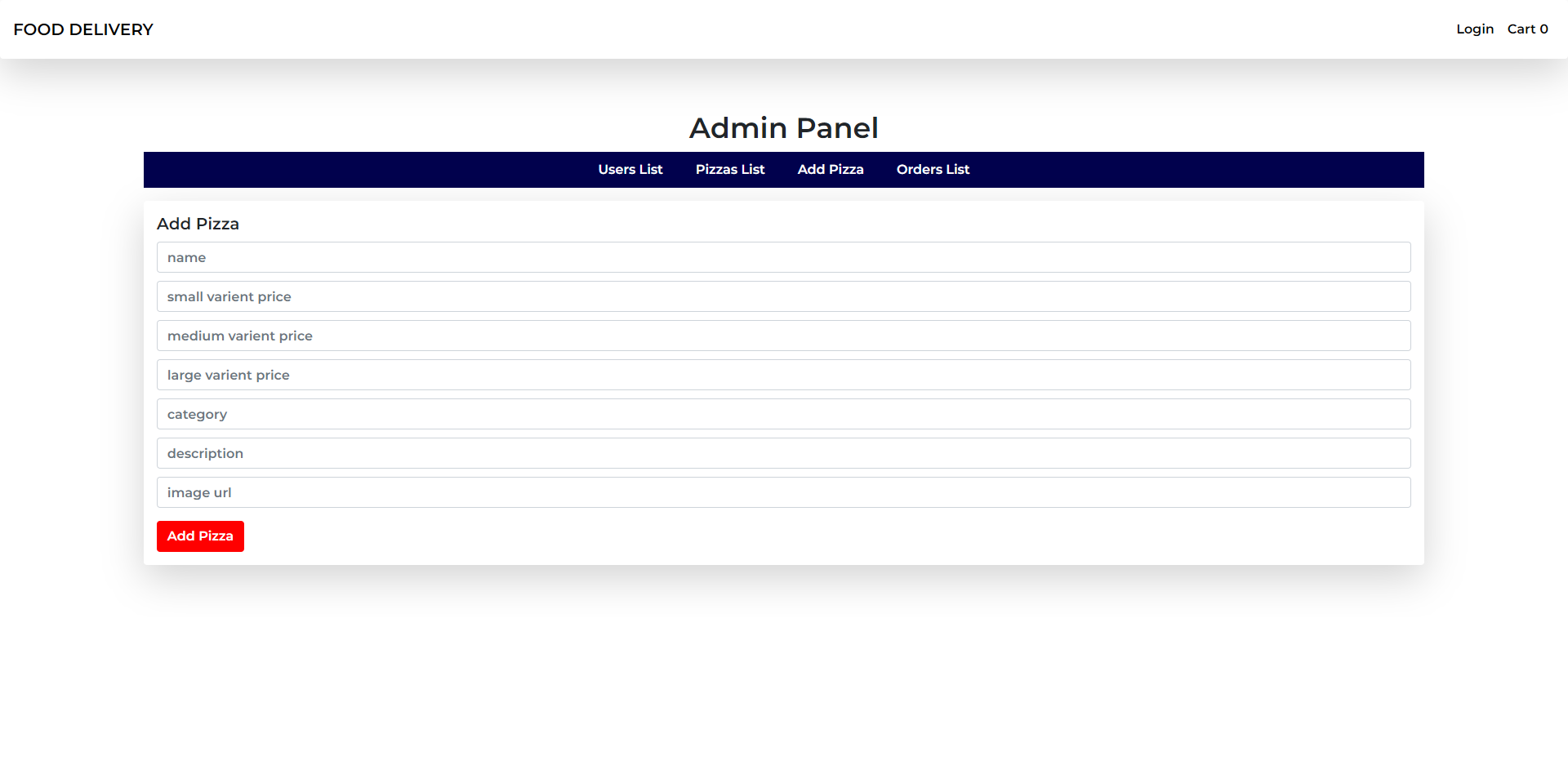
**Pizza List:** Manage pizza inventory by adding, editing, or deleting pizzas.





**Add Pizza:** Add new pizzas to the inventory with details such as name, price, category, description, and image URL.

Here all the fields are mandatory before click on Add pizza button to get proper details on end user login.



**Orders List:** View and manage orders, including changing order status to "delivered”. Here once the user order any product, it will be reflected in admin > Orders list with Deliver status button.

Once we click on Deliver button, the status will be changed to delivered with a success message. Thus we managed the data interaction between 2 roles.

We used Dynamic data with react-redux for the state management.

A screenshot of a computer

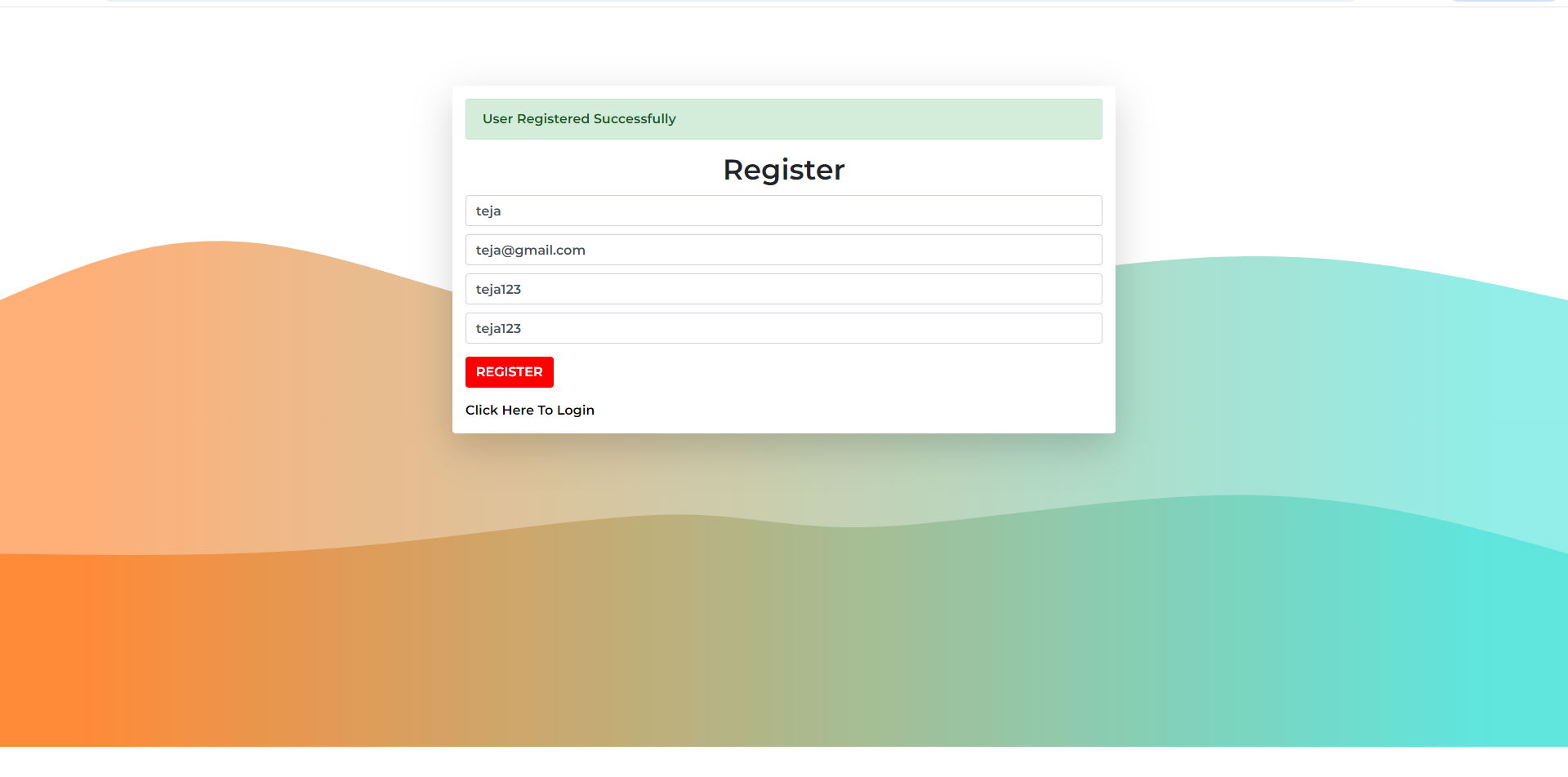
Description automatically generated

**User:**

**Registration/Login:** End users register with their name, email, and password and can subsequently log in using their credentials.

A screenshot of a computer

Description automatically generated



Once successfully registered, click on ‘Click here to login’. It will be redirected to login page.

Now, login with the registered credentials

A screenshot of a login screen

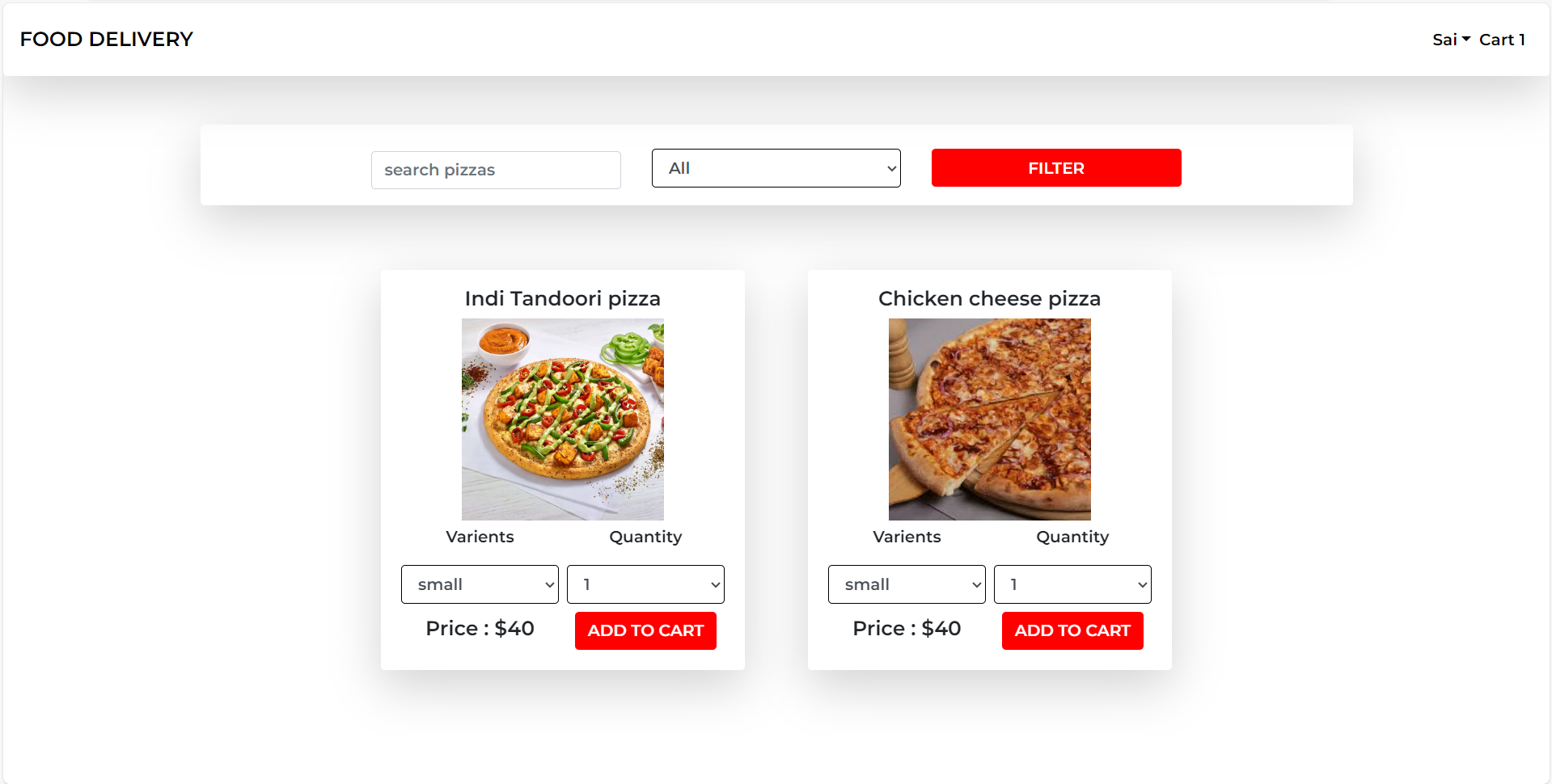
Description automatically generated

**Landing Page:** Displays a list of available pizzas with options to filter by category or search for specific items.

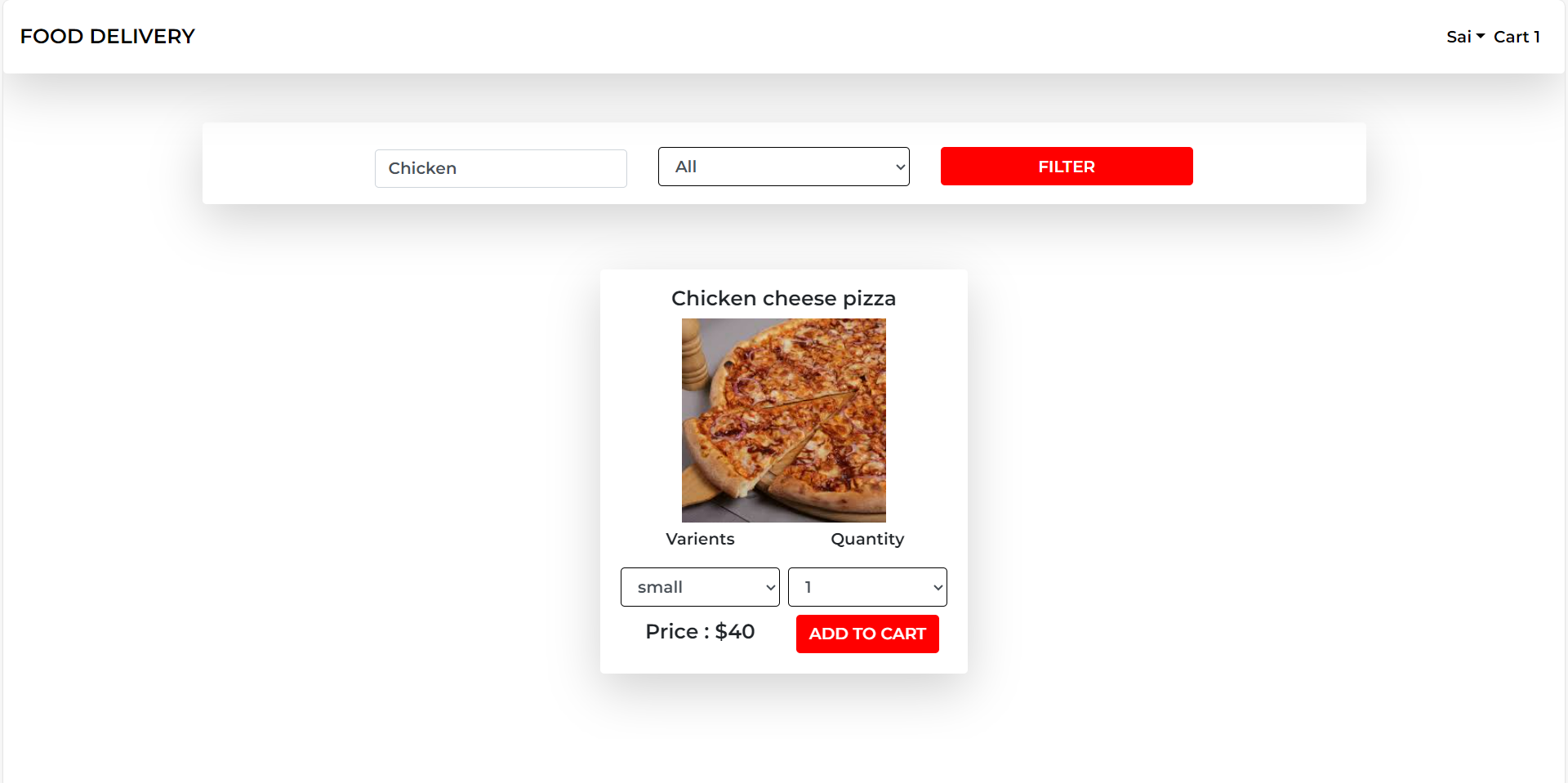
The application has header and content.

In header, we have Username with orders and logout options, cart and Application name.

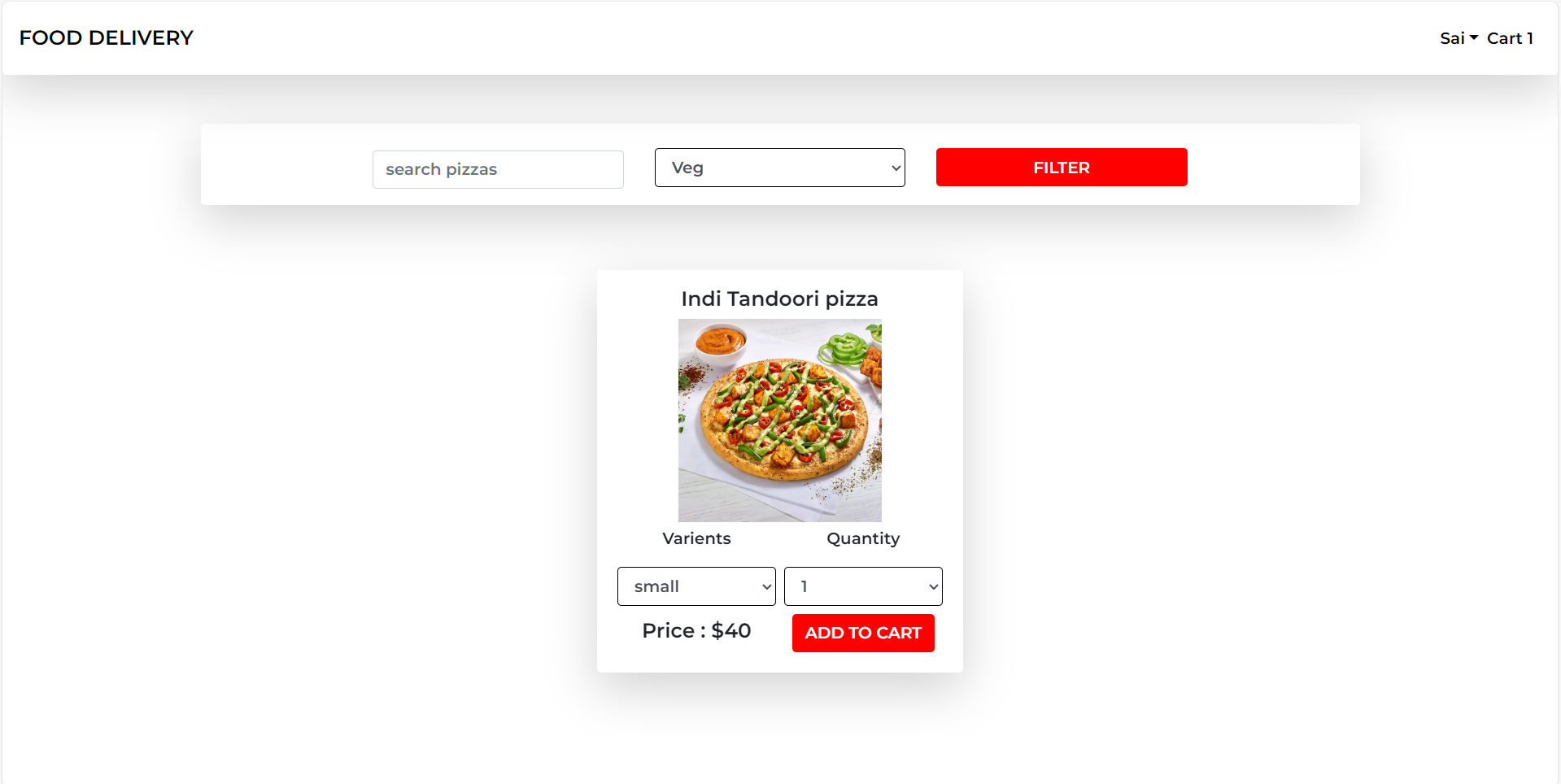
If we click on Food Delivery, it will be redirected to landing page.



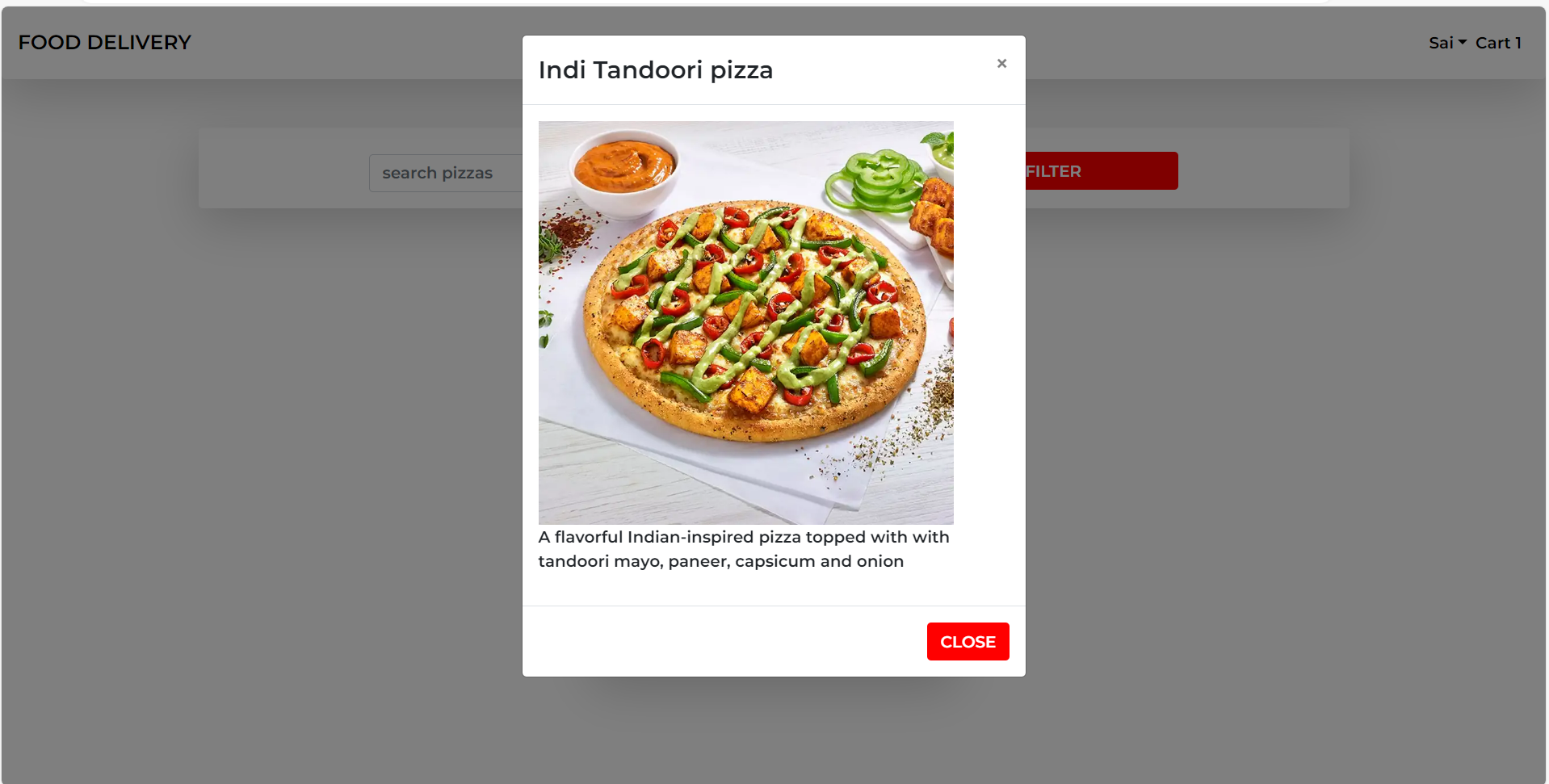
Here, we can search for the required product or can filter by category.



Here, the filter is for category selection. Once we select any category and click on filter the list will filter as per the selection.



We can get more details about the product by selecting. Once we select any product, a Modal popup will be opened with description.



Also, we have provision to select the quantity of the product from the Quantity dropdown for the list page.

A screenshot of a menu

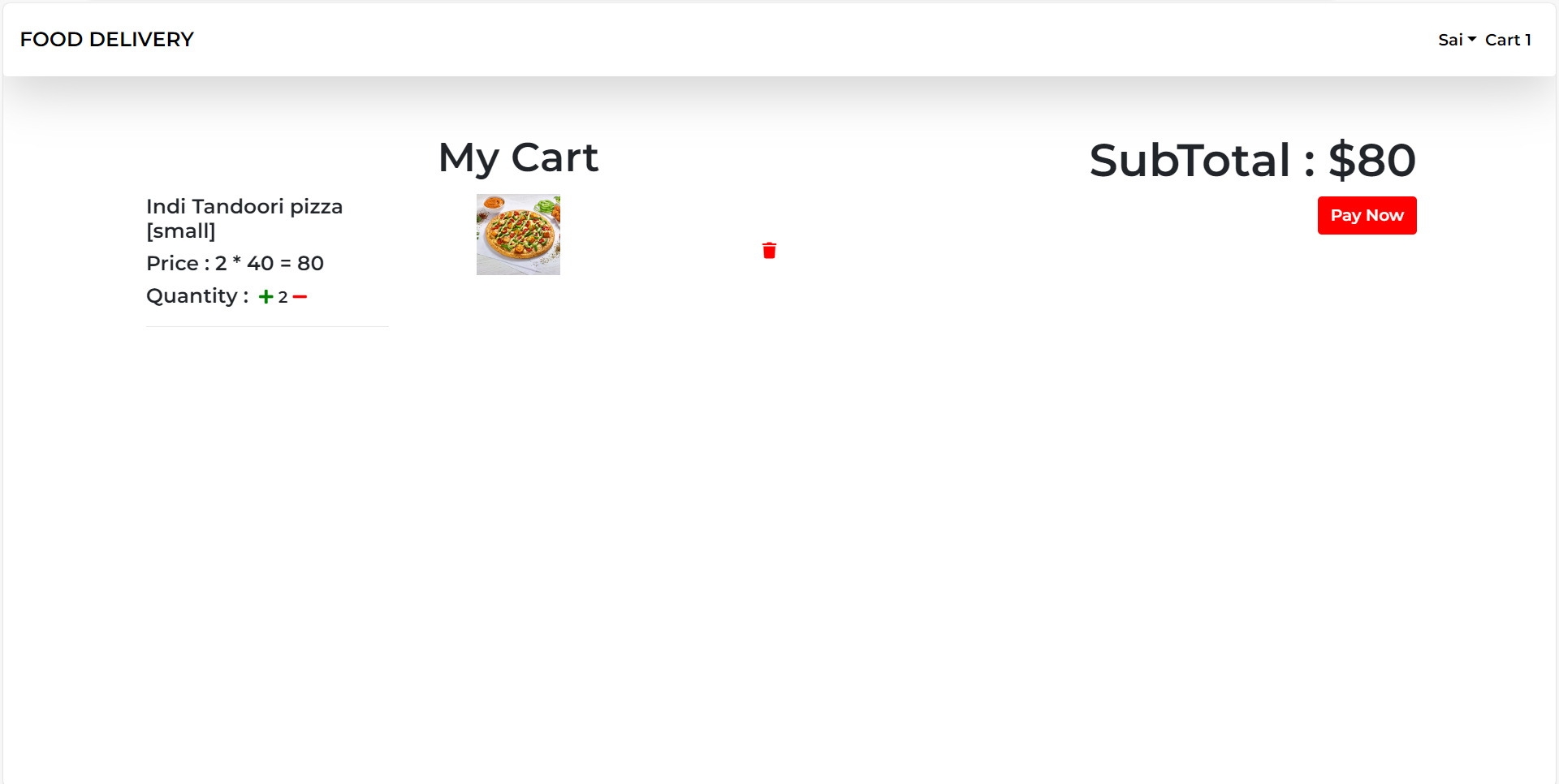
Description automatically generated

Once we select the quantity and click on add to cart button, the product will be added to the cart and cart count will be incremented on the header.

For this data update/management we used react-redux. React-Redux is state management concept and one of the main advantages of React.

**Cart Functionality:** The cart page Allows users to add pizzas to their cart, view cart contents, and proceed to checkout.

Clicking on the cart from header will be redirected to cart page.



It has all the selected product details and total price of the items. And we can continue to next by clicking on Pay Now button from cart.

Once we click on Pay now, the selected item will be reflected in Admin > orders list page where Admin can check the details and change the status of the order.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

This is how the basic interaction between the customer and admin/providers will be developed.

**Future Scope:**

**Integration with Payment Gateways:** Enable online payment options for a seamless checkout experience.

**Real-Time Order Tracking:** Implement real-time order tracking functionality for end users.

**User Reviews and Ratings:** Allow users to leave reviews and ratings for pizzas they have ordered.

**Loyalty Program:** Introduce a loyalty program to reward frequent customers with discounts or special offers.

**Mobile Application Development:** Develop native mobile applications for iOS and Android platforms to extend accessibility.

**Internationalization:** Support multiple languages to cater to a global audience.

**Conclusion:**

Online food ordering application offers a convenient and user-friendly platform for ordering delicious pizzas online. With its intuitive user interface, seamless navigation, and robust features for administrators and end users alike, the system aims to enhance the pizza ordering experience and delight customers. With continuous updates and enhancements, Pizza Delight is poised to become a go-to destination for pizza lovers everywhere.