E-Commerce Application using Generative-AI

By,  
Pushpraj Singh

Email: pushpraj\_singh@persistent.com

Tools Used: ChatGPT, Nodejs, Angular CLI, VsCode

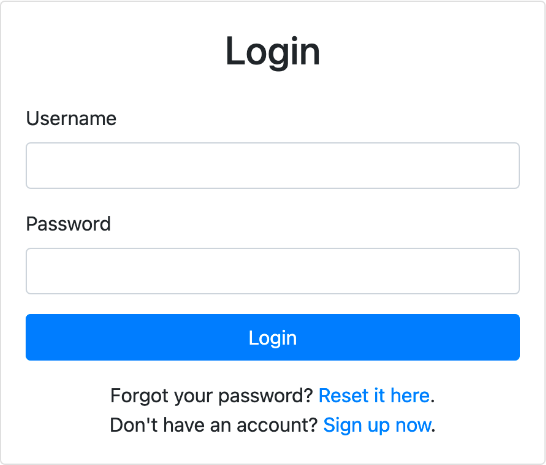
Chat Link : <https://chat.openai.com/share/478e0e1c-93d5-4e0e-b4a9-ff61546d8098>

Index :

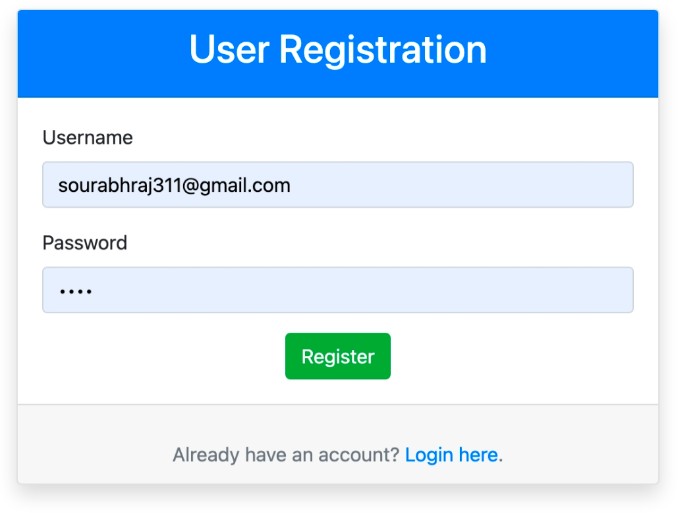
1. Design Document
2. Screenshots

Screenshots :

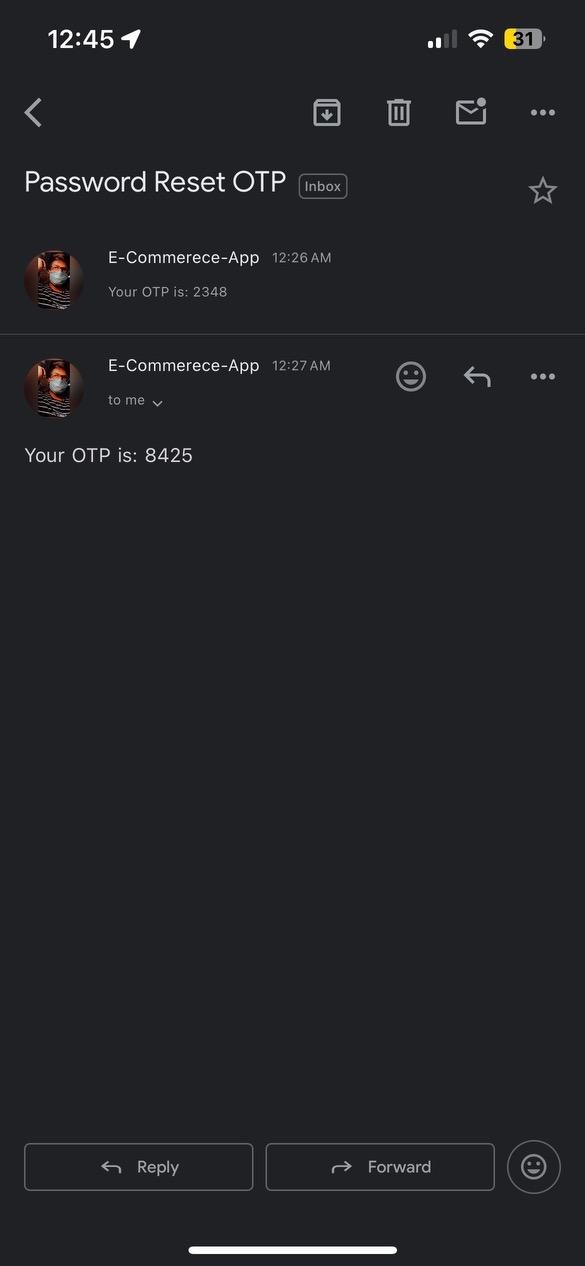
Login Module :

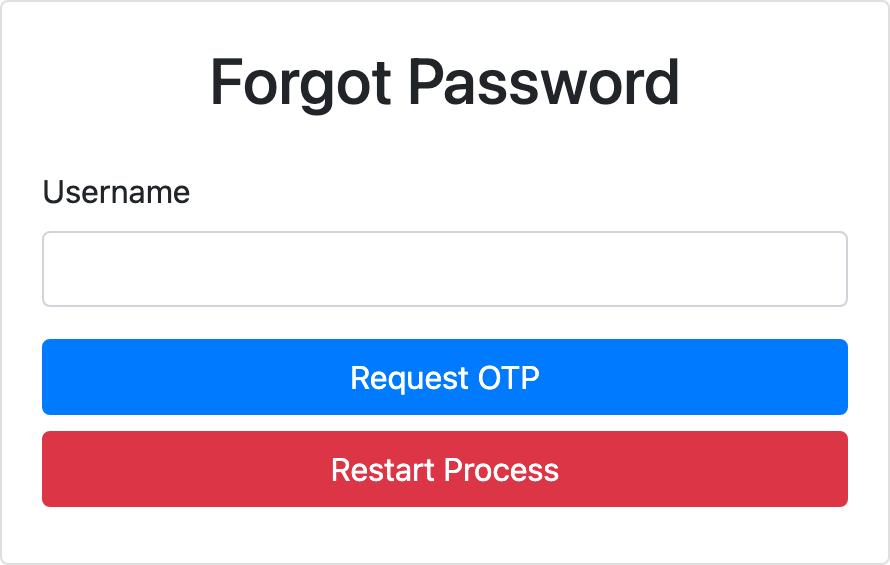


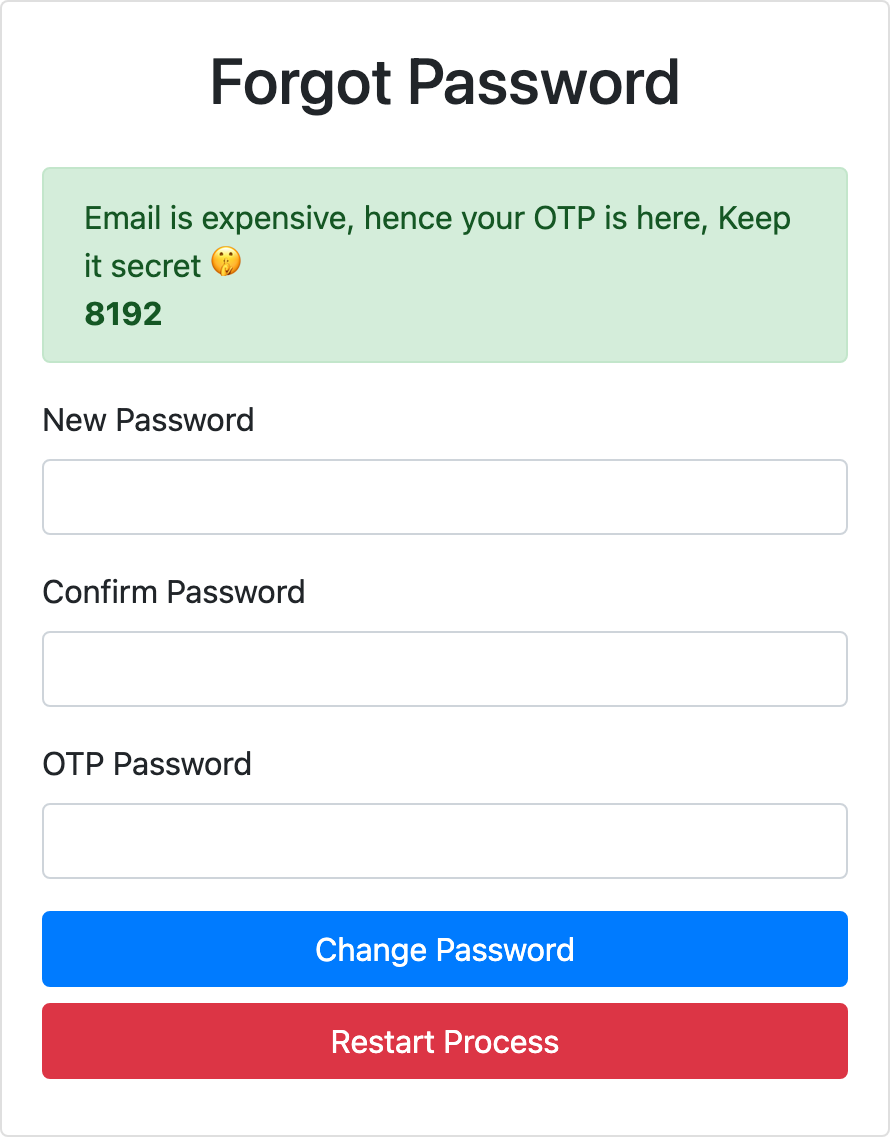
Registration Module :

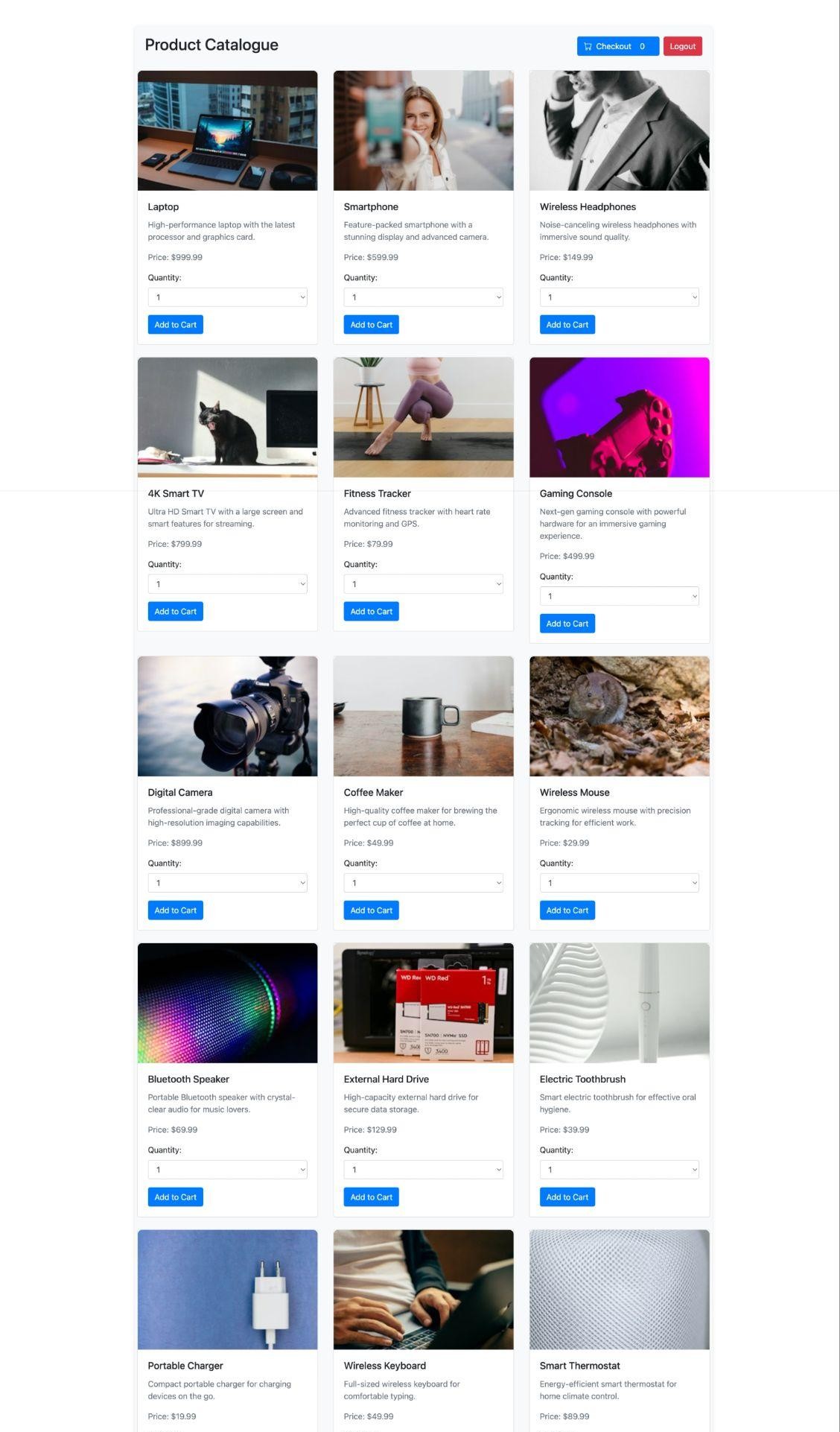


Forgot/Change Password email

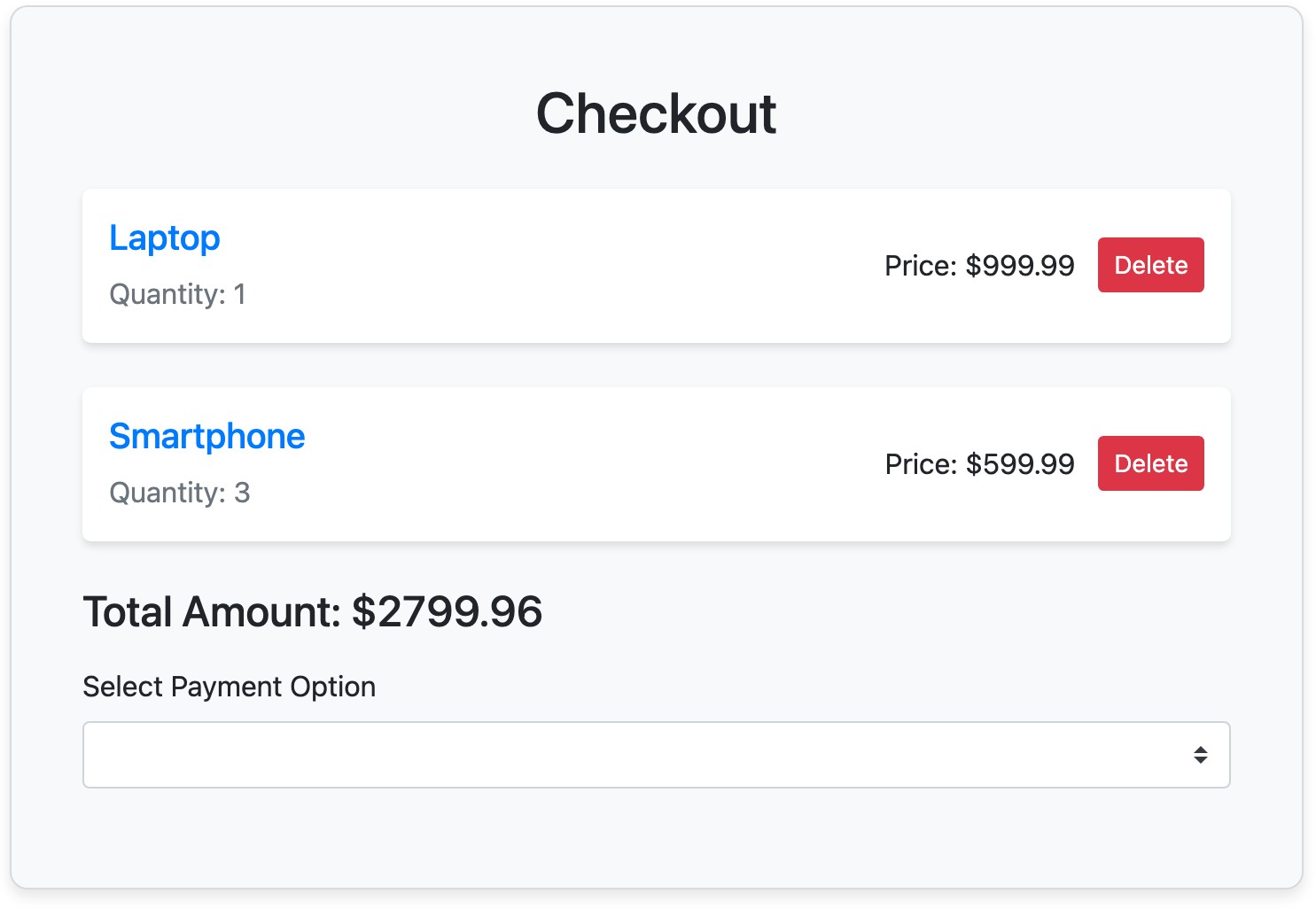


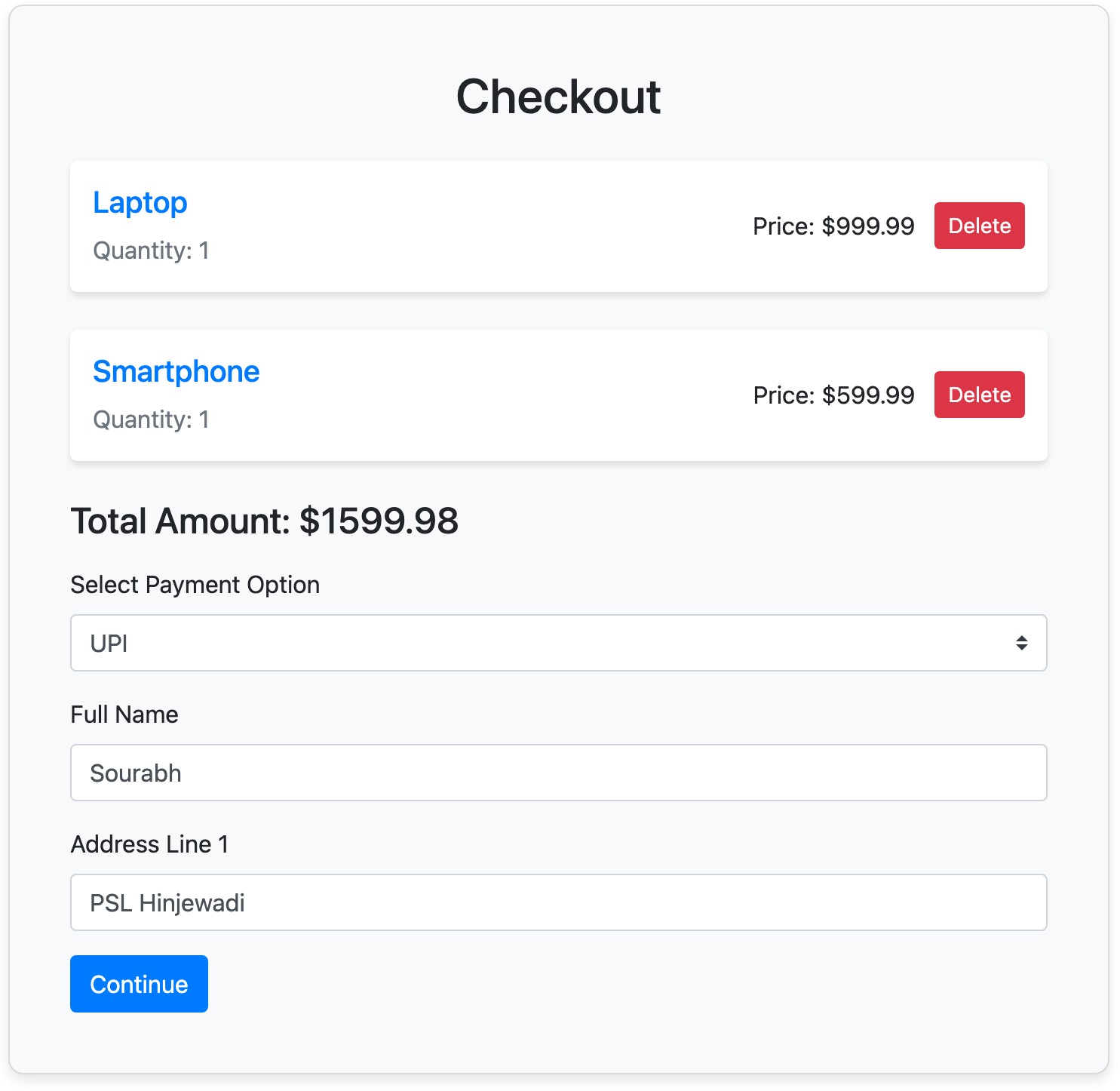


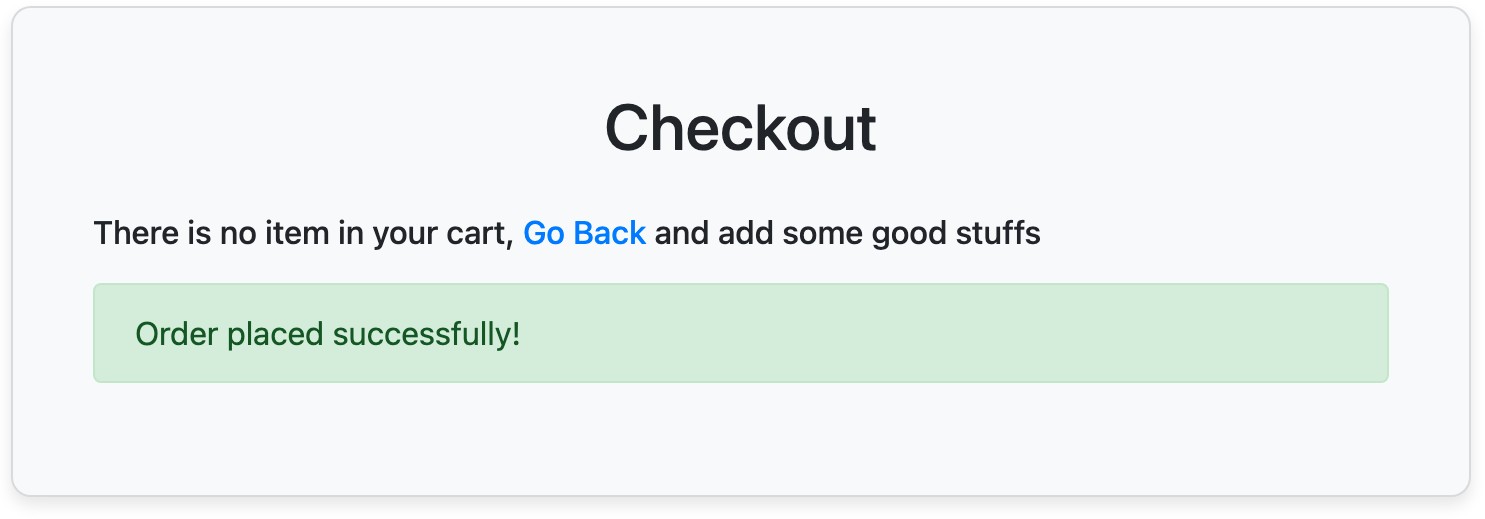
Products Catalogue :



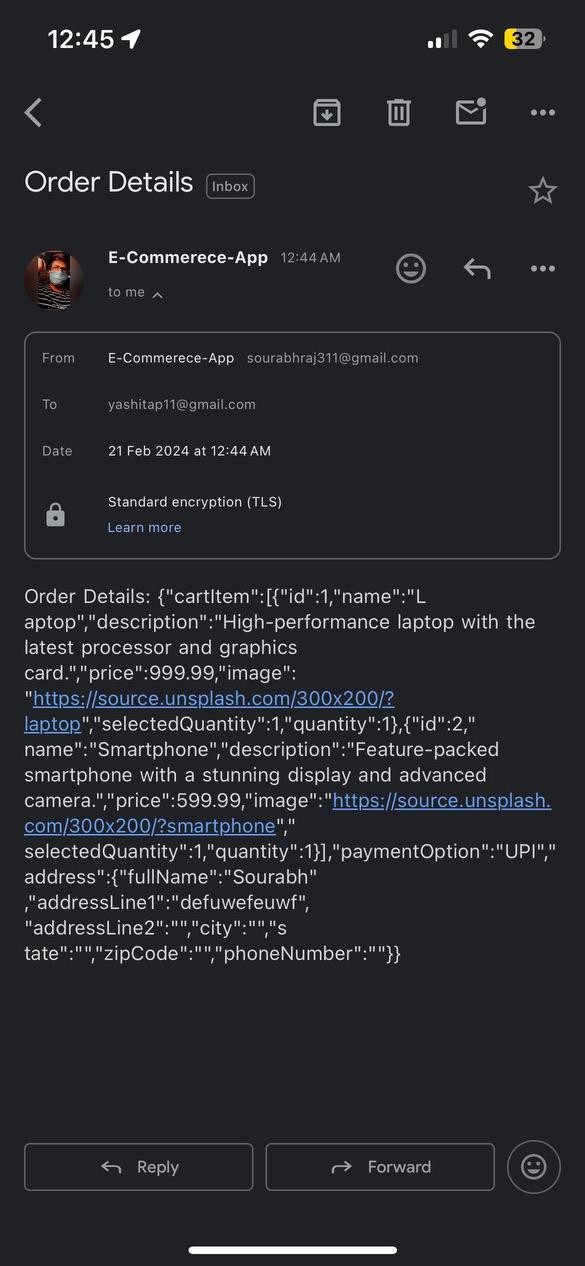
Checkout Page :







Email on order complete



**Project Design Document**

**1. Introduction**

This design document outlines the steps and code snippets discussed during the development of the project. The project involves creating a web application with features such as user registration, login, product catalog,

shopping cart, and checkout.

1. **Backend Implementation**
   1. **User Registration**

* Endpoint: /register
* Method: POST
* Description: Handles user registration by checking for existing usernames and saving new user data to MongoDB.

# User Login

* + - Endpoint: /login
    - Method: POST
    - Description: Validates user credentials during login and returns a success message upon successful login.

# Change Password

* + - Endpoint: /change-password
    - Method: POST
    - Description: Allows users to change their password by verifying the old password.

# Forgot Password

* + - Endpoint: /forgot-password
    - Method: POST
    - Description: Initiates the process of resetting the password by sending an OTP to the user’s email.

# OTP Validation during Login

* + - Endpoint: /login-otp
    - Method: POST
    - Description: Validates the OTP entered during login.

# Product Catalog

* + - Endpoint: /products
    - Method: GET
    - Description: Fetches product catalog data from a data.json file.

# Add to Cart

* + - Endpoint: /add-to-cart
    - Method: POST
    - Description: Adds a product to the user’s cart based on the provided parameters.

# Checkout and Payment

* + - Endpoint: /checkout
    - Method: POST
    - Description: Handles the checkout process, creating an order and clearing the user’s cart.

1. **Frontend Implementation**
   1. **User Interface**
      1. **User Registration Page**
      * Component: RegistrationComponent
      * Features:
        + User registration form with validation.
        + API call to the backend for registration.

## User Login Page

* Component: LoginComponent
* Features:
  + User login form with validation.
  + API call to the backend for login.

## Forgot Password Page

* Component: ForgotPasswordComponent
* Features:
  + Request OTP button.
  + Form to enter and submit the new password.

## Product Catalogue Page

* Component: ProductCatalogComponent
* Features:
  + Fetches and displays the product catalog.
  + Add to cart functionality.

## Checkout Page

* Component: CheckoutComponent
* Features:
  + Displays cart items.
  + Select payment option and enter address.
  + Place order functionality.

**3.2 Code Snippets**

## Backend

* MongoDB models: User, Product, Cart, Order.
* Express.js routes for user authentication, product catalog, and checkout.

## Frontend

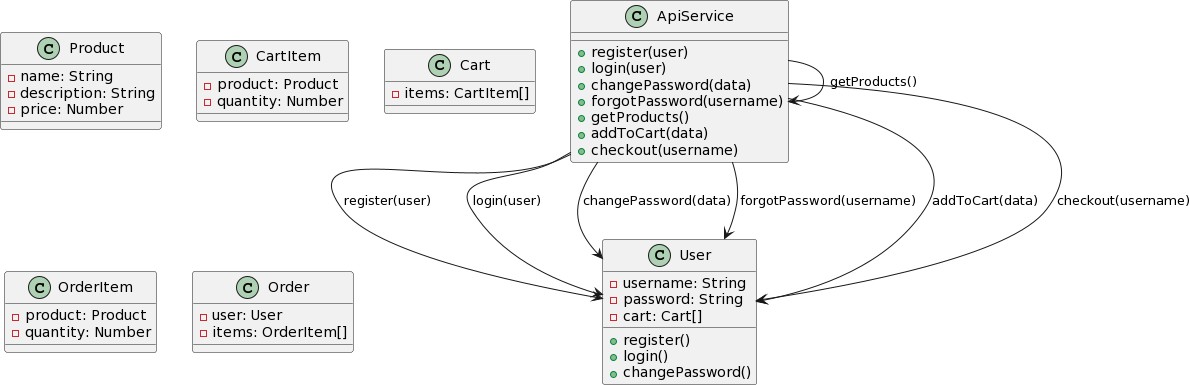
* Angular components for user registration, login, forgot password, product catalog, and checkout.
* Service for API communication (ApiService).
* State management for cart items.

**4. Conclusion**

This design document summarizes the backend and frontend

implementation steps, including API endpoints, components, and code snippets. Further enhancements and additional features can be integrated based on project requirements.

Diagrams



**Frontend Design Document**

## Angular Components:

* + - 1. Registration Component:
         * Responsible for user registration.
         * Renders a form with fields for username and password.
         * Invokes the register method of ApiService for user registration.
      2. Login Component:
         * Handles user login functionality.
         * Displays a form with fields for username and password.
         * Calls the login method of ApiService for user authentication.
         * Navigates to the product catalog page upon successful login.
      3. Product Catalog Component:
         * Fetches and displays a list of products from the server using

getProducts method of ApiService.

* + - * + Allows users to view product details and add them to the cart.
        + Uses the addToCart method of ApiService for adding products to the cart.
      1. Checkout Component:
         * Manages the checkout process.
         * Displays the user’s cart items and the total amount.
         * Provides a dropdown for selecting payment options.
         * Collects user address details through a form.
         * Invokes the checkout method of ApiService to place an order.
      2. User Service:
         * Stores user-related information and cart items.
         * Provides methods for user authentication and managing the cart.
         * Communicates with ApiService for server interactions.

