Dharmsinh Desai University

Department of MCA

Semester - III

Advanced Web Development

Termwork Report on

Product Management System

Submitted by

Roll No	**Name**	**Student Id**
MCA005	**Nikhil R. Lathiya**	**23MAPBG026**

INDEX

Sr-No	Title	Page No.
1	Project Definition	1
2	Code Structure	2
3	Source Code	3
4	Project Output	43

PROJECT DEFINITION

The Product Management System is a web-based application designed to streamline the process of managing products in an online store. The system provides an admin dashboard for administrators to perform essential product management tasks, such as adding, updating, deleting, and viewing product details. The admin can manage product information, including name, price, quantity, description, and images.

On the user side, the application offers a shopping experience where users can browse the available products, view detailed information, add items to their cart, and place orders. The system also features a cart functionality, where users can increase or decrease product quantities, view the total cost, and confirm orders. Once an order is confirmed, an invoice is generated, and users have the option to download the invoice in PDF format.

Tools & Technologies :

- Node.js
- Express.js
- Prisma (ORM)
- PostgreSQL (Database)
- EJS (Embedded JavaScript for templating)
- Bootstrap (CSS framework)
- Puppeteer (for generating PDFs)

CODE STRUCTURE

```
/product_management_system
 ├── node_modules/
 ├── prisma/
|    ├── migrations/
|    ├── schema.prisma
|   └── seed.js
 ├── public/
|    ├── css/
|    ├── images/
|   └── bills/
 ├── src/
|    ├── controllers/
|   |    ├── adminController.js
|   |   └── userController.js
|    ├── models/
|   |    ├── prisma.js
|   |   └── productModel.js
|   └── routes/
|        ├── adminRoutes.js
|       └── userRoutes.js
 ├── views/
|    ├── admin/
|   |    ├── addProduct.ejs
|   |    ├── adminDashboard.ejs
|   |    ├── adminLogin.ejs
|   |   └── editProduct.ejs
|   └── user/
|        ├── cart.ejs
|        ├── invoice.ejs
|        ├── order.ejs
|        ├── productDetails.ejs
|       └── shop.ejs
 ├── .env
 ├── app.js
 ├── package.json
└── package-lock.json
```

CODE

.env

```
DATABASE_URL="postgresql://postgres:123456@localhost:5432/product_Management"
```

DATABASE

schema.prisma:

```
datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

generator client {
  provider = "prisma-client-js"
}

model Product {
  id          Int      @id @default(autoincrement())
  name        String
  price       Float
  quantity    Int
  description String
  imageUrl    String

}
```

Prisma/seed.js:

```
import { PrismaClient } from "@prisma/client";

const prisma = new PrismaClient();

async function main() {
  // Inserting initial product data
  await prisma.product.createMany({
    data: [
    {
      name: "Wireless Headphones",
      price: 99.99,
      quantity: 150,
      description:
        "High-quality wireless headphones with noise cancellation.",
      imageUrl:
        "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQYgMEjpvHkOchM16UZBjU1S5uLtxBdMEojwA&s",
    },
    {
      name: "Gaming Mouse",
      price: 49.99,
      quantity: 300,
      description: "Ergonomic gaming mouse with customizable buttons.",
      imageUrl:
        "https://assetsio.gnwcdn.com/g502x_f9QuuM8.jpeg?width=1200&height=1200&fit=bounds&quality=70&format=jpg&auto=webp",
    },
    {
      name: "Smart Watch",
      price: 199.99,
      quantity: 120,
      description:
        "Feature-rich smartwatch with health tracking and notifications.",
      imageUrl:
        "https://www.corseca.in/cdn/shop/files/2_6b32baf1-0bdf-4e4e-8fde-d4946d1e94fd.jpg?v=1700823367",
    },
    {
      name: "Bluetooth Speaker",
      price: 79.99,
      quantity: 200,
      description:
        "Portable Bluetooth speaker with deep bass and long battery life.",
      imageUrl:
        "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTbppGSX3HdN4bXy-NFVMfATbcQQg3Yg7bbHQ&s",
    },
      {
        name: "Laptop Stand",
        price: 29.99,
        quantity: 500,
        description:
          "Adjustable laptop stand for better ergonomics and cooling.",
        imageUrl:
          "https://symplify.in/cdn/shop/products/Wooden-Laptop-Stand-Opt3-
3_1024x1024.jpg?v=1658253933",
      },
      {
        name: "Wireless Charger",
        price: 39.99,
        quantity: 250,
        description: "Fast wireless charger for smartphones and other devices.",
        imageUrl:
          "https://www.brookstone.com/cdn/shop/files/BK15-4QL1F-
WT_White_2_750x.jpg?v=1708116047",
      },
      {
        name: "Action Camera",
        price: 149.99,
        quantity: 100,
        description:
          "Compact action camera with 4K recording and waterproof design.",
        imageUrl: "https://eavf3cou74b.exactdn.com/wp-
content/uploads/2023/08/28104711/Action-Camera-4.jpg",
      },
    ],
  });

  console.log("Seed data for products created successfully.");
}

main()
  .catch((e) => {
    console.error("Error during seed: ", e);
    process.exit(1);
  })
  .finally(async () => {
    await prisma.$disconnect();
  });
```

MODELS

prisma.js

```
const { PrismaClient } = require('@prisma/client');

// Create and export the Prisma client instance
const prisma = new PrismaClient();

module.exports = prisma;
```

productModel.js

```
const prisma = require("./prisma");

exports.getAllProducts = async () => {
  try {
    return await prisma.product.findMany();
  } catch (error) {
    console.error("Error fetching products:", error);
    throw new Error("Failed to fetch products");
  }
};

exports.getParticularProduct = async (id) => {
  try {
    return await prisma.product.findUnique({
      where: { id: Number(id) },
    });
  } catch (error) {
    console.error(`Error fetching product with ID ${id}:`, error);
    throw new Error("Failed to fetch product");
  }
};

exports.addProduct = async (productData) => {
  try {
    return await prisma.product.create({
      data: productData,
    });
  } catch (error) {
    console.error("Error adding product:", error);
    throw new Error("Failed to add product");
  }
};

exports.editProduct = async (id, updatedData) => {
  try {
    return await prisma.product.update({
      where: { id: Number(id) },
      data: updatedData,
    });
  } catch (error) {
    console.error(`Error updating product with ID ${id}:`, error);
    throw new Error("Failed to update product");
  }
};

exports.deleteProduct = async (id) => {
  try {
    return await prisma.product.delete({
      where: { id: Number(id) },
    });
  } catch (error) {
    console.error(`Error deleting product with ID ${id}:`, error);
    throw new Error("Failed to delete product");
  }
};

exports.updateProductQuantity = async (id, quantity) => {
  try {
    return await prisma.product.update({
      where: { id: Number(id) },
      data: { quantity: quantity },
    });
  } catch (error) {
    console.error(`Error updating product quantity:`, error);
    throw new Error("Failed to update product quantity");
  }
};
```

CONTROLLERS

adminController.js

```
const { PrismaClient } = require("@prisma/client");
const prisma = new PrismaClient();

// Display the login page
exports.getLoginPage = (req, res) => {
  res.render("admin/adminLogin", { errorMessage: null });
};

// Handle login attempt
exports.handleLogin = async (req, res) => {
  const { username, password } = req.body;

  try {
    // Hardcoded credentials
    const hardcodedUsername = "admin";
    const hardcodedPassword = "admin";


    if (username !== hardcodedUsername || password !== hardcodedPassword) {
      return res.render("admin/adminLogin", {
        errorMessage: "Invalid username or password",
      });
    }

    // Redirect to the admin dashboard upon successful login
    res.redirect("/admin/dashboard");
  } catch (error) {
    console.error("Login error:", error);
    res.render("admin/adminLogin", {
      errorMessage: "An error occurred, please try again.",
    });
  }
};

exports.getDashboard = async (req, res) => {
  try {
    // Fetch all products from the database
    const products = await prisma.product.findMany();

    // Render the admin dashboard and pass the products
    res.render("admin/adminDashboard", { products });
  } catch (error) {
    console.error("Error fetching products:", error);
    res.status(500).send("Internal Server Error");
  }
};

exports.logout = (req, res) => {
  // Redirect back to the login page
  res.redirect("/shop");
};

exports.getAddProductPage = (req, res) => {
  res.render("admin/addProduct");
};

exports.addProduct = async (req, res) => {
  const { name, price, quantity, imageUrl, description } = req.body;

  try {
    // Save the new product to the database
    await prisma.product.create({
      data: {
        name,
        price: parseFloat(price),
        quantity: parseInt(quantity),
        imageUrl,
        description,
      },
    });
    // Redirect back to admin dashboard to show the new product
    res.redirect("/admin/dashboard");
  } catch (error) {
    console.error("Error adding product:", error);
    res.status(500).send("Error adding product.");
  }
};

exports.getEditProductPage = async (req, res) => {
  const productId = req.params.id;
  try {
    // Fetch the product by ID
    const product = await prisma.product.findUnique({
      where: { id: Number(productId) },
    });

    //  product doesn't exist then return 404
    if (!product) {
      return res.status(404).send("Product not found");
    }

    res.render("admin/editProduct", { product });
  } catch (error) {
    console.error("Error fetching product:", error);
    res.status(500).send("Error fetching product.");
  }
};

// Handle Edit Product form submission
exports.editProduct = async (req, res) => {
  const productId = req.params.id;
  const { name, price, quantity, imageUrl, description } = req.body;

  try {
    // Update the product in the database
    await prisma.product.update({
      where: { id: Number(productId) },
      data: {
        name,
        price: parseFloat(price),
        quantity: parseInt(quantity),
        imageUrl,
        description,
      },
    });

    // Redirect back to the admin dashboard
    res.redirect("/admin/dashboard");
  } catch (error) {
    console.error("Error updating product:", error);
    res.status(500).send("Error updating product.");
  }
};

exports.deleteProduct = async (req, res) => {
  const productId = req.params.id;

  try {
    // Delete the product from the database
    await prisma.product.delete({
      where: { id: Number(productId) },
    });

    res.redirect("/admin/dashboard");
  } catch (error) {
    console.error("Error deleting product:", error);
    res.status(500).send("Error deleting product.");
  }
};
```

userController.js

```
const puppeteer = require("puppeteer");
const fs = require("fs");
const path = require("path");
const ProductModel = require("../models/productModel");

exports.getShop = async (req, res) => {
  const products = await ProductModel.getAllProducts();
  res.render("user/shop", { products });
};

exports.getProductDetails = async (req, res) => {
  const productId = req.params.id;

  try {
    const product = await ProductModel.getParticularProduct(productId);

    if (!product) {
      return res.status(404).send("Product not found");
    }

    res.render("user/productDetails", { product });
  } catch (error) {
    console.error("Error fetching product details:", error);
    res.status(500).send("Internal Server Error");
  }
};

// Add product to cart
exports.addToCart = async (req, res) => {
  const productId = req.body.productId;

  try {
    const product = await ProductModel.getParticularProduct(productId);

    if (!product) {
      return res.status(404).send("Product not found");
    }

    // Initialize cart if it doesn't exist
    if (!req.session.cart) {
      req.session.cart = [];
    }

    // Check if product already exists in the cart
    const existingProductIndex = req.session.cart.findIndex(
      (item) => item.product.id === product.id
    );

    if (existingProductIndex > -1) {
      // Increase the quantity if the product already exists in the cart
      req.session.cart[existingProductIndex].quantity += 1;
    } else {
      // Add the product to the cart
      req.session.cart.push({
        product: product,
        quantity: 1,
      });
    }

    req.session.save(() => {
      res.redirect("/cart"); // Redirect to cart page
    });
  } catch (error) {
    console.error("Error adding to cart:", error);
    res.status(500).send("Internal Server Error");
  }
};

// View the cart
exports.viewCart = (req, res) => {
  const cart = req.session.cart || [];

  const totalPrice = cart.reduce((total, item) => {
    return total + item.product.price * item.quantity;
  }, 0);

  res.render("user/cart", { cart, totalPrice });
};

exports.decreaseQuantity = (req, res) => {
  const productId = parseInt(req.body.productId);

  if (!req.session.cart) {
    return res.redirect("/cart");
  }
  const productIndex = req.session.cart.findIndex(
    (item) => item.product.id === productId
  );

  if (productIndex > -1) {
    if (req.session.cart[productIndex].quantity > 1) {
      req.session.cart[productIndex].quantity -= 1;
    } else {
      req.session.cart.splice(productIndex, 1); // Remove item from cart if quantity reaches 0
    }
  }

  req.session.save(() => {
    res.redirect("/cart");
  });
};

// Increase quantity of product in cart
exports.increaseQuantity = (req, res) => {
  const productId = parseInt(req.body.productId);

  if (!req.session.cart) {
    return res.redirect("/cart");
  }

  const productIndex = req.session.cart.findIndex(
    (item) => item.product.id === productId
  );

  if (productIndex > -1) {
    req.session.cart[productIndex].quantity += 1; // Increase quantity
  }

  req.session.save(() => {
    res.redirect("/cart");
  });
};

exports.viewOrder = (req, res) => {
  const cart = req.session.cart || [];
  const totalPrice = cart.reduce((total, item) => {
    return total + item.product.price * item.quantity;
  }, 0);

  res.render("user/order", { cart, totalPrice });
};

exports.confirmOrder = async (req, res) => {
  const cart = req.session.cart || [];

  if (cart.length === 0) {
    return res.redirect("/cart");
  }

  // Decrease product quantities
  for (let item of cart) {
    const product = await ProductModel.getParticularProduct(item.product.id);
    if (product.quantity >= item.quantity) {
      await ProductModel.updateProductQuantity(
        product.id,
        product.quantity - item.quantity
      );
    } else {
      return res
        .status(400)
        .send("Not enough quantity in stock for " + product.name);
    }
  }

  const billsDir = path.resolve(__dirname, "../public/bills");
  if (!fs.existsSync(billsDir)) {
    fs.mkdirSync(billsDir, { recursive: true });
  }
  const totalPrice = cart.reduce(
    (total, item) => total + item.product.price * item.quantity,
    0
  );
  const orderDate = new Date().toLocaleDateString();

  res.render(
    "user/invoice",
    { cart, totalPrice, orderDate },
    async (err, html) => {
      if (err) {
        console.error("Error rendering EJS:", err);
        return res.status(500).send("Error generating invoice");
      }

      try {
        // Launch Puppeteer to convert HTML to PDF
        const browser = await puppeteer.launch();
        const page = await browser.newPage();
        await page.setContent(html);
        const pdfPath = path.join(billsDir, `bill_${Date.now()}.pdf`);

        await page.pdf({ path: pdfPath, format: "A4" });

        await browser.close();

        // Clear the cart
        req.session.cart = [];

        // Redirect to download the generated PDF
        res.redirect(`/order/download?file=${pdfPath}`);
      } catch (error) {
        console.error("Error generating PDF:", error);
        res.status(500).send("Error generating PDF");
      }
    }
  );
};

// Route to download the generated PDF
exports.downloadBill = (req, res) => {
  const filePath = req.query.file;
  res.download(filePath, "invoice.pdf", (err) => {
    if (err) {
      res.status(500).send("Error downloading file.");
    }
  });};
```

ROUTES

adminRoutes.js

```
const express = require("express");
const router = express.Router();
const adminController = require("../controllers/adminController");

router.get("/", (req, res) => {
  res.redirect("/admin/login");
});
// Display the login page when /admin is accessed
router.get("/login", adminController.getLoginPage);

// Handle admin login logic
router.post("/login", adminController.handleLogin);
router.get("/dashboard", adminController.getDashboard);
router.get("/add-product", adminController.getAddProductPage);
router.post("/products/add", adminController.addProduct);

// Route to show the Edit Product form
router.get("/products/edit/:id", adminController.getEditProductPage);

// Route to handle Edit Product form submission
router.post("/products/edit/:id", adminController.editProduct);

router.post("/products/delete/:id", adminController.deleteProduct);

router.get("/logout", adminController.logout);

module.exports = router;
```

userRoutes.js

```
const express = require("express");
const router = express.Router();
const userController = require("../controllers/userController");

router.get("/", userController.getShop);
router.get("/shop", userController.getShop);
router.get("/product/:id", userController.getProductDetails);
router.post("/cart", userController.addToCart);
router.get("/cart", userController.viewCart);
router.post("/cart/increase", userController.increaseQuantity);
router.post("/cart/decrease", userController.decreaseQuantity);

// Routes for order confirmation and downloading the bill
router.get("/order", userController.viewOrder);
router.post("/order/confirm", userController.confirmOrder);
router.get("/order/download", userController.downloadBill);

module.exports = router;
```

App.js

```
const express = require("express");
const app = express();
const bodyParser = require("body-parser");
const adminRoutes = require("./src/routes/adminRoutes");
const userRoutes = require("./src/routes/userRoutes");
const session = require("express-session");


app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use(express.static(__dirname + "/public"));


app.use(
  session({
    secret: "mySecretKey",
    resave: false,
    saveUninitialized: true,
    cookie: { secure: false }
  })
);
app.set("view engine", "ejs");
app.set("views", "views");
app.use("/admin", adminRoutes);
app.use("/", userRoutes);

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

VIEWS

Admin/addProduct.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Add Product</title>
  <link rel="stylesheet" href="/css/adminDashboard.css">
  <link rel="stylesheet" href="/css/addProduct.css">
  <link rel="stylesheet" href="
            https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">
</head>

<body class="bg-light">

  <!-- Navbar -->
  <header>
    <nav class="navbar bg-dark px-4">
      <div class="logo">
        <img src="/images/1.webp" alt="ShopEase Logo" class="logo-image">
        <h1 class="project-name">Admin Dashboard</h1>
      </div>
      <ul class="nav-links" style="padding-top:15px">
        <li><a href="/admin/dashboard">Products</a></li>
        <li><a href="/admin/add-product">Add Products</a></li>
        <li><a href="/admin/logout" class="logout-btn">Logout</a></li>
      </ul>
    </nav>
  </header>

  <!-- Add Product Form Section -->
  <section class="form-section">
    <h2 class="form-title">Add New Product</h2>
    <form action="/admin/products/add" method="POST" class="add-product-form">
      <div class="form-group">
        <label for="name">Product Name</label>
        <input type="text" id="name" name="name" placeholder="Enter product name" required>
      </div>
      <div class="form-group">
        <label for="price">Price</label>
        <input type="number" step="0.01" id="price" name="price" placeholder="Enter product
price" required>
      </div>
      <div class="form-group">
        <label for="quantity">Quantity</label>
        <input type="number" id="quantity" name="quantity" placeholder="Enter product
quantity" required>
      </div>
      <div class="form-group">
        <label for="imageUrl">Image URL</label>
        <input type="text" id="imageUrl" name="imageUrl" placeholder="Enter image URL"
required>
      </div>
      <div class="form-group">
        <label for="description">Description</label>
        <textarea id="description" name="description" placeholder="Enter product description"
required></textarea>
      </div>
      <center>
        <button type="submit" class="btn w-25 save-btn btn-dark">Save Product</button>
      </center>
    </form>
  </section>
```

```
</body>

</html>
```

adminDashboard.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Admin Dashboard</title>
  <link rel="stylesheet" href="/css/style.css">
  <link rel="stylesheet" href="/css/adminDashboard.css">
  <link rel="stylesheet" href="
            https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">
</head>

<body class="bg-light">

  <!-- Navbar -->
  <header>
    <nav class="navbar bg-dark px-4">
      <div class="logo">
        <img src="/images/1.webp" alt="ShopEase Logo" class="logo-image">
        <h1 class="project-name">Admin Dashboard</h1>
      </div>
      <ul class="nav-links" style="padding-top:15px">
        <li><a href="/admin/dashboard">Products</a></li>
        <li><a href="/admin/add-product">Add Products</a></li>
        <li><a href="/admin/logout" class="logout-btn">Logout</a></li>
      </ul>
    </nav>
  </header>

  <div class="container-fluid w-75 my-5">
  <h2 class="text-center">Manage Products</h2>
  <div class="row">
   <% products.forEach(product => { %>
   <div class="col-sm-3 my-3">
    <div class="card mx-2">
     <div class="card-body">
      <img src="<%= product.imageUrl %>" alt="<%= product.name %>" class="product-
image">
      <div class="container text-center">
       <h3 class="product-name"><%= product.name %></h3>
       <hr>
       <p class="product-price text-danger"><b>$<%= product.price %></b></p>
       <p class="product-quantity text-primary"><b>In Stock: <%=
product.quantity %></b></p>
       <div class="row">
        <div class="col p-1">
         <a href="/admin/products/edit/<%= product.id %>" class="btn w-100 btn-
primary"><b>Edit</b></a>
        </div>
        <div class="col p-1">
         <form action="/admin/products/delete/<%= product.id %>" method="post">

          <button type="submit" class="btn btn-danger w-100"><b>Delete</b></button>
         </form>
        </div>
       </div>

      </div>
     </div>
    </div>
   </div>
   <% }) %>
  </div>
 </div>


</body>

</html>
```

adminLogin.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Admin Login</title>
 <link rel="stylesheet" href="/css/adminLogin.css">

 <link rel="stylesheet" href="
       https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
 <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">
</head>

<body class="bg-light">

 <div class="container">
  <center>
   <div class="card w-50">
    <div class="card-body">
     <center>
      <img src="/images/1.webp" alt="Company Logo" class="logo">
      <h1 class="login-title">Admin Login</h1>
      <form action="/admin/login" method="POST">
       <div class="form-group">
        <input type="text" id="username" name="username" placeholder="Enter Username"
required>
       </div>
       <div class="form-group">
        <input type="password" id="password" name="password" placeholder="Enter
Password" required>
       </div>
       <% if (errorMessage) { %>
       <p class="error-message"><%= errorMessage %></p>
       <% } %>
```

```
              <button type="submit" class="btn btn-danger w-25">Login</button>
            </form>
          </center>
        </div>
      </div>
    </center>

  </div>

</body>

</html>
```

editProduct.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Edit Product</title>
  <link rel="stylesheet" href="/css/adminDashboard.css">
  <link rel="stylesheet" href="/css/editProduct.css">
</head>

<body>

  <!-- Navbar -->
  <header>
    <nav class="navbar">
      <div class="logo">
        <h1></h1>
      </div>
      <ul class="nav-links">
        <li><a href="/admin/dashboard">Products</a></li>
        <li><a href="/admin/add-product">Add Products</a></li>
        <li><a href="/admin/logout" class="logout-btn">Logout</a></li>
      </ul>
    </nav>
  </header>

  <!-- Edit Product Form -->
  <section class="form-section">
    <h2 class="form-title">Edit Product</h2>
    <form action="/admin/products/edit/<%= product.id %>" method="POST" class="edit-product-form">
      <div class="form-group">
        <label for="name">Product Name</label>
        <input type="text" id="name" name="name" value="<%= product.name %>" required>
      </div>
      <div class="form-group">
        <label for="price">Price</label>
        <input type="number" step="0.01" id="price" name="price" value="<%= product.price %>" required>
      </div>
      <div class="form-group">
        <label for="quantity">Quantity</label>
        <input type="number" id="quantity" name="quantity" value="<%= product.quantity %>" required>
      </div>
      <div class="form-group">
        <label for="imageUrl">Image URL</label>
        <input type="text" id="imageUrl" name="imageUrl" value="<%= product.imageUrl %>" required>
      </div>
      <div class="form-group">
        <label for="description">Description</label>
        <textarea id="description" name="description" required><%= product.description %></textarea>
      </div>
      <div class="form-actions">
        <button type="submit" class="btn save-btn">Save</button>
        <a href="/admin/dashboard" class="btn cancel-btn">Cancel</a>
      </div>
    </form>
  </section>

  <!-- Footer -->
```

```
 <footer>
  <p>&copy; 2024 Admin Dashboard. All Rights Reserved.</p>
 </footer>

</body>

</html>
```

Views/user

cart.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Your Cart</title>
 <link rel="stylesheet" href="/css/cart.css">
 <link rel="stylesheet" href="
      https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
 <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">

</head>

<body>

 <header>
  <nav class="navbar bg-dark navbar-expand-lg">
   <div class="container-fluid text-white">
    <a class="navbar-brand text-white" href="/shop" style="font-
size:30px"><b>ShopEase</b></a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
     <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse text-white" id="navbarSupportedContent">
     <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
```

```
          <li class="nav-item">
            <a class="nav-link text-white active" aria-current="page" href="/shop">Shop</a>
          </li>
          <li class="nav-item">
            <a class="nav-link text-white" href="/cart">My Cart</a>
          </li>
          <li class="nav-item">
            <a class="nav-link text-white" href="/order">Order</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</header>

<style>
  .navbar-nav .nav-link {
    font-size: 20px;
    margin-right: 20px;
    transition: color 0.3s ease, background-color 0.3s ease;
  }

  .navbar-nav .nav-link:hover {
    color: #ff5722;
    background-color: #ffffff20;
    border-radius: 5px;
  }

  .navbar-nav .nav-link:last-child {
    margin-right: 20px;
  }
</style>

<!-- Cart Section -->
<section class="cart">
  <h1 class="cart-title text-dark"><b>Your Cart</b></h1>

  <% if (cart.length === 0) { %>
  <p class="empty-cart-message">Your cart is empty.</p>
  <% } else { %>
```

```
    <table class="cart-table">
      <thead class="table-dark">
        <tr>
          <th>Product</th>
          <th>Price</th>
          <th>Quantity</th>
          <th>Total</th>
          <th>Actions</th>
        </tr>
      </thead>
      <tbody>
        <% cart.forEach(item => { %>
        <tr>
          <td>
            <div class="product-info">
              <img src="<%= item.product.imageUrl %>" alt="<%= item.product.name %>" class="cart-product-image">
              <span class="product-name"><%= item.product.name %></span>
            </div>
          </td>
          <td>$<%= item.product.price.toFixed(2) %></td>
          <td><%= item.quantity %></td>
          <td>$<%= (item.product.price * item.quantity).toFixed(2) %></td>
          <td>
            <form action="/cart/decrease" method="post" class="quantity-form">
              <input type="hidden" name="productId" value="<%= item.product.id %>">
              <button type="submit" class="btn quantity-btn decrease-btn btn-danger">-</button>
            </form>
            <form action="/cart/increase" method="post" class="quantity-form">
              <input type="hidden" name="productId" value="<%= item.product.id %>">
              <button type="submit" class="btn btn-success quantity-btn increase-btn">+</button>
            </form>
          </td>
        </tr>
        <% }) %>
      </tbody>
    </table>

    <h2 class="total-price">Total: $<%= totalPrice.toFixed(2) %></h2>
    <form action="/order" method="get">
      <center>
        <button type="submit" class="btn order-btn btn-primary" style="font-size:20px"><b>Order</b></button>
      </center>
    </form>

    <% } %>

  </section>


</body>

</html>
```

invoice.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Invoice</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
    }

    h1 {
      text-align: center;
    }

    .header,
    .footer {
      text-align: center;
      margin-bottom: 20px;
    }

    .invoice-table {
      width: 100%;
      border-collapse: collapse;
    }

    .invoice-table th,
    .invoice-table td {
      border: 1px solid #ddd;
      padding: 10px;
      text-align: left;
    }

    .invoice-table th {
      background-color: #f4f4f4;
    }

    .invoice-summary {
      margin-top: 20px;
      text-align: right;
      font-size: 1.2rem;
    }
  </style>
</head>

<body>
  <div class="header">
    <h1>Invoice</h1>
    <p>MyShop</p>
    <p>1234 Main Street</p>
    <p>City, State, ZIP</p>
    <p>Phone: (123) 456-7890</p>
    <p>Order Date: <%= orderDate %></p>
  </div>

  <table class="invoice-table">
    <thead>
      <tr>
        <th>Product</th>
        <th>Quantity</th>
        <th>Price</th>
        <th>Total</th>
      </tr>
    </thead>
    <tbody>
      <% cart.forEach(item => { %>
      <tr>
        <td><%= item.product.name %></td>
        <td><%= item.quantity %></td>
        <td>$<%= item.product.price.toFixed(2) %></td>
        <td>$<%= (item.product.price * item.quantity).toFixed(2) %></td>
      </tr>
      <% }) %>
    </tbody>
  </table>

  <div class="invoice-summary">
    <p>Total Price: $<%= totalPrice.toFixed(2) %></p>
  </div>

  <div class="footer">
    <p>Thank you for your purchase!</p>
  </div>
</body>

</html>
```

order.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Order Confirmation</title>
  <link rel="stylesheet" href="/css/order.css">
  <link rel="stylesheet" href="
           https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">
</head>

<body>

  <!-- Sticky Navbar -->
  <header>
    <nav class="navbar bg-dark navbar-expand-lg">
      <div class="container-fluid text-white">
        <a class="navbar-brand text-white" href="/shop" style="font-
size:30px"><b>ShopEase</b></a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse text-white" id="navbarSupportedContent">
          <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link text-white active" aria-current="page" href="/shop">Shop</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="/cart">My Cart</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="/order">Order</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>

  <style>
    .navbar-nav .nav-link {
      font-size: 20px;
      margin-right: 20px;
      transition: color 0.3s ease, background-color 0.3s ease;
    }

    .navbar-nav .nav-link:hover {
      color: #ff5722;
      background-color: #ffffff20;
      border-radius: 5px;
    }

    .navbar-nav .nav-link:last-child {
      margin-right: 20px;
    }
  </style>

  <!-- Order Confirmation Section -->
  <section class="order-confirmation">
    <h1 class="order-title">Order Confirmation</h1>

    <table class="order-table">
      <thead>
        <tr>
          <th>Product</th>
          <th>Price</th>
          <th>Quantity</th>
          <th>Total</th>
        </tr>
      </thead>
      <tbody>
        <% cart.forEach(item => { %>
        <tr>
          <td><%= item.product.name %></td>
          <td>$<%= item.product.price.toFixed(2) %></td>
          <td><%= item.quantity %></td>
          <td>$<%= (item.product.price * item.quantity).toFixed(2) %></td>
        </tr>
        <% }) %>
      </tbody>
      </table>

      <h2 class="total-price">Total Price: $<%= totalPrice.toFixed(2) %></h2>

      <form action="/order/confirm" method="post">
        <button type="submit" class="btn btn-success">Confirm Order</button>
      </form>

    </section>

  </body>

  </html>
```

productDetails.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Product Details</title>
  <link rel="stylesheet" href="/css/productDetails.css">
  <link href="https://fonts.googleapis.com/css2?family=Lobster&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="
      https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">
</head>

<body class="bg-light">

  <header>
    <nav class="navbar bg-dark navbar-expand-lg">
      <div class="container-fluid text-white">
        <a class="navbar-brand text-white" href="/shop" style="font-
size:30px"><b>ShopEase</b></a>
```

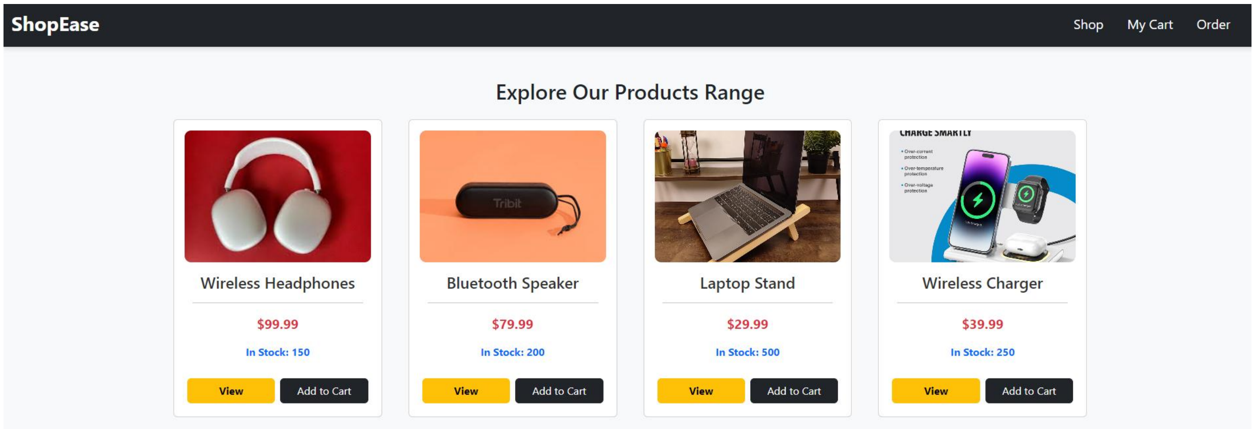
```
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse text-white" id="navbarSupportedContent">
          <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link text-white active" aria-current="page" href="/shop">Shop</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="/cart">My Cart</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="/order">Order</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>

  <style>
    .navbar-nav .nav-link {
      font-size: 20px;
      margin-right: 20px;
      transition: color 0.3s ease, background-color 0.3s ease;
    }

    .navbar-nav .nav-link:hover {
      color: #ff5722;
      background-color: #ffffff20;
      border-radius: 5px;
    }

    .navbar-nav .nav-link:last-child {
      margin-right: 20px;
    }
  </style>
  <div class="container d-flex align-items-center justify-content-center" style="margin-top:175px">

    <div class="card w-75">
      <div class="card-body">
        <div class="row">
          <div class="col-sm-4">
            <img src="<%= product.imageUrl %>" alt="<%= product.name %>" class="product-details-image">
          </div>
          <div class="col-sm-8 px-2">
            <hr>
            <h2 class="product-name"><%= product.name %></h2>
            <hr>
            <div class="my-3">
              <p class="product-price text-dark"><b>Price: <span class="text-danger">$<%= product.price %></span></b></p>
              <p class="product-quantity text-dark"><b>Available Quantity: <span class="text-primary"><%= product.quantity %></span></b></p>
              <b>
                <p class="product-description"><%= product.description %></p>
              </b>
              <form action="/cart" class="my-4" method="post">
                <input type="hidden" name="productId" value="<%= product.id %>">
                <button type="submit" class="btn btn-warning  "><b>Add to Cart</b></button>
              </form>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>



</body>

</html>
```

shop.ejs

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ShopEase</title>
  <link rel="stylesheet" href="/css/style.css">
  <link rel="stylesheet" href="
            https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js">
</head>

<body class="bg-light">

  <header>
    <nav class="navbar bg-dark navbar-expand-lg">
      <div class="container-fluid text-white">
        <a class="navbar-brand text-white" href="/shop" style="font-
size:30px"><b>ShopEase</b></a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse text-white" id="navbarSupportedContent">
          <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link text-white active" aria-current="page" href="/shop">Shop</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="/cart">My Cart</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-white" href="/order">Order</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </header>

  <style>
    .navbar-nav .nav-link {
      font-size: 20px;
      margin-right: 20px;
      transition: color 0.3s ease, background-color 0.3s ease;
    }

    .navbar-nav .nav-link:hover {
      color: #ff5722;
      background-color: #ffffff20;
      border-radius: 5px;
    }

    .navbar-nav .nav-link:last-child {
      margin-right: 20px;
    }
  </style>
  <div class="container-fluid w-75 my-5">
    <h2 class="text-center">Explore Our Products Range</h2>
    <div class="row">
      <% products.forEach(product => { %>
      <div class="col-sm-3 my-3">
        <div class="card mx-2">
          <div class="card-body">
            <img src="<%= product.imageUrl %>" alt="<%= product.name %>" class="product-image">
            <div class="container text-center">
              <h3 class="product-name"><%= product.name %></h3>
              <hr>
              <p class="product-price text-danger"><b>$<%= product.price %></b></p>
              <p class="product-quantity text-primary"><b>In Stock: <%= product.quantity %></b></p>
              <div class="row">
                <div class="col p-1">
                  <a href="/product/<%= product.id %>" class="btn w-100 btn-warning"><b>View</b></a>
        </div>
        <div class="col p-1">
         <form action="/cart" method="post">
          <input type="hidden" name="productId" value="<%= product.id %>">
          <button type="submit" class="btn btn-dark w-100">Add to Cart</button>
         </form>
        </div>
       </div>

      </div>
     </div>
    </div>
   </div>
   <% }) %>
  </div>
 </div>




</body>

</html>
```

OUTPUTS

User Dashboard



Explore Our Products Range

Wireless Headphones

$99.99

In Stock: 150

Bluetooth Speaker

$79.99

In Stock: 200

Laptop Stand

$29.99

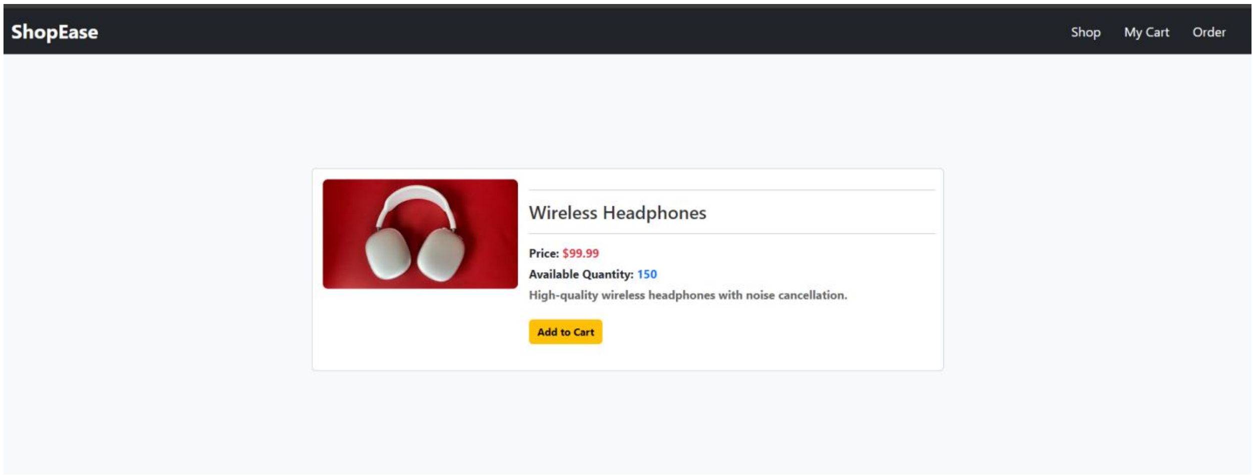
In Stock: 500

Wireless Charger

$39.99

In Stock: 250

View Details



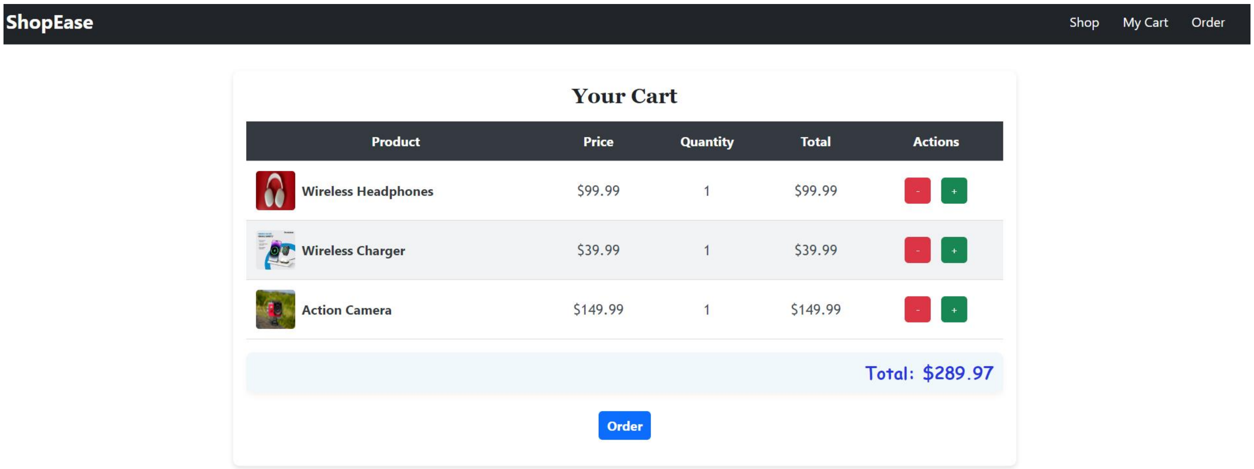
Wireless Headphones

Price: $99.99

Available Quantity: 150

High-quality wireless headphones with noise cancellation.

My Cart

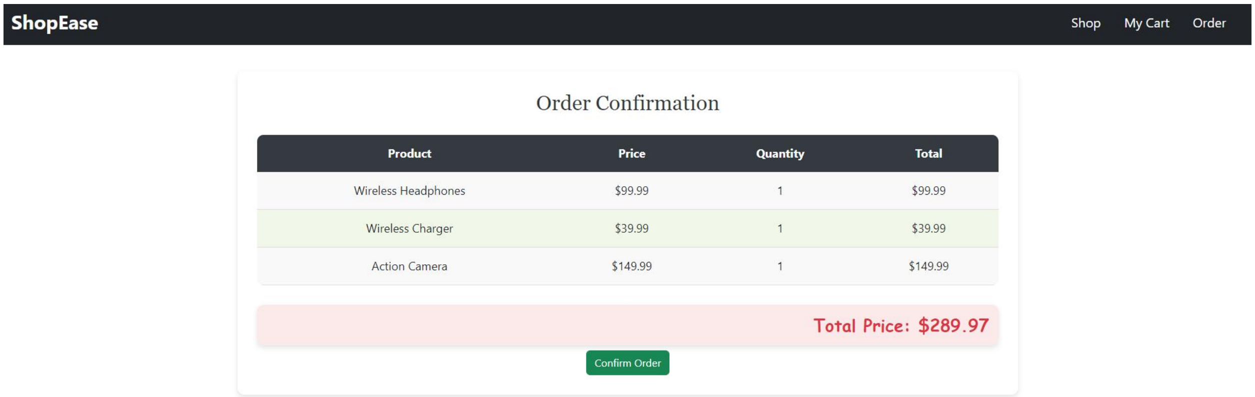


Your Cart

Product	Price	Quantity	Total	Actions
Wireless Headphones	$99.99	1	$99.99	- +
Wireless Charger	$39.99	1	$39.99	- +
Action Camera	$149.99	1	$149.99	- +

Total: $289.97

Order



Order Confirmation

Product	Price	Quantity	Total
Wireless Headphones	$99.99	1	$99.99
Wireless Charger	$39.99	1	$39.99
Action Camera	$149.99	1	$149.99

Total Price: $289.97

Invoice

MyShop

1234 Main Street

City, State, ZIP

Phone: (123) 456-7890

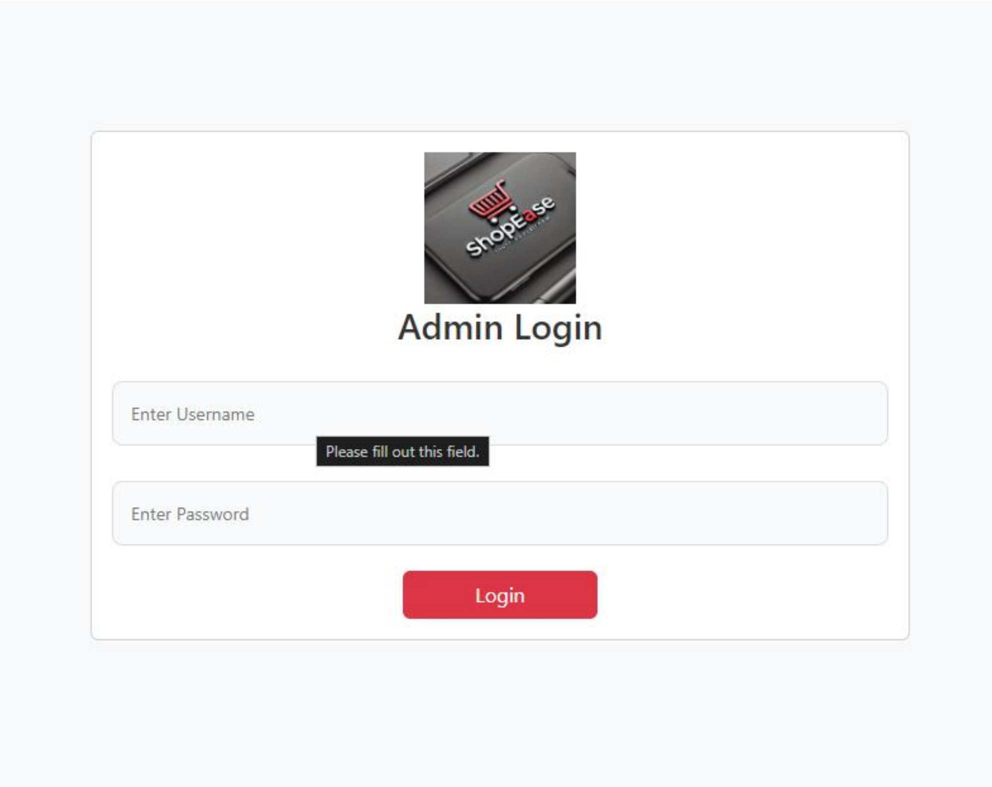
Order Date: 10/20/2024

Product	Quantity	Price	Total
Wireless Headphones	1	$99.99	$99.99
Wireless Charger	1	$39.99	$39.99
Action Camera	1	$149.99	$149.99

Total Price: $289.97

Thank you for your purchase!

Admin Login Form



Admin Login

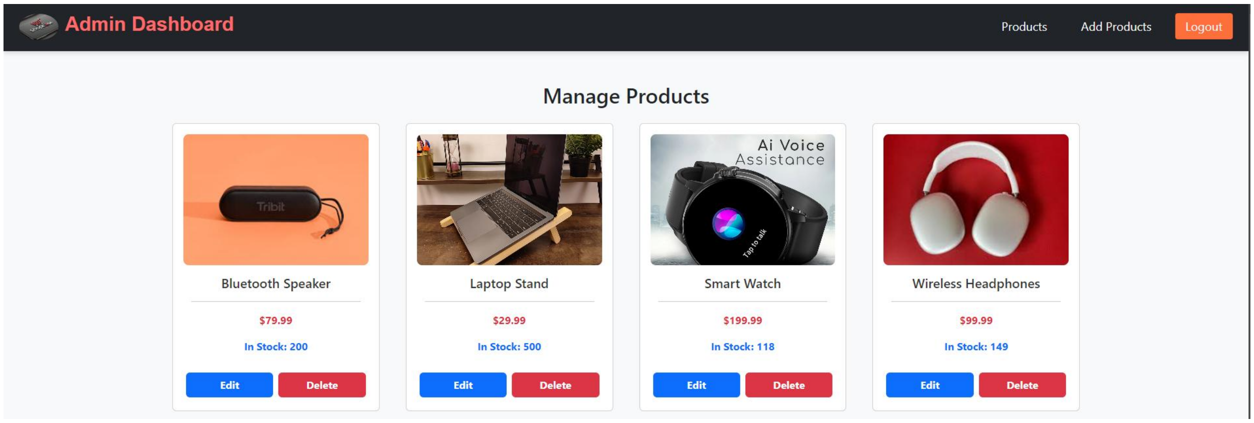
Enter Username

Please fill out this field.

Enter Password

Login

Admin Dashboard



Admin Dashboard

Products Add Products Logout

Manage Products

Bluetooth Speaker

$79.99

In Stock: 200

Edit Delete

Laptop Stand

$29.99

In Stock: 500

Edit Delete

Smart Watch

$199.99

In Stock: 118

Edit Delete

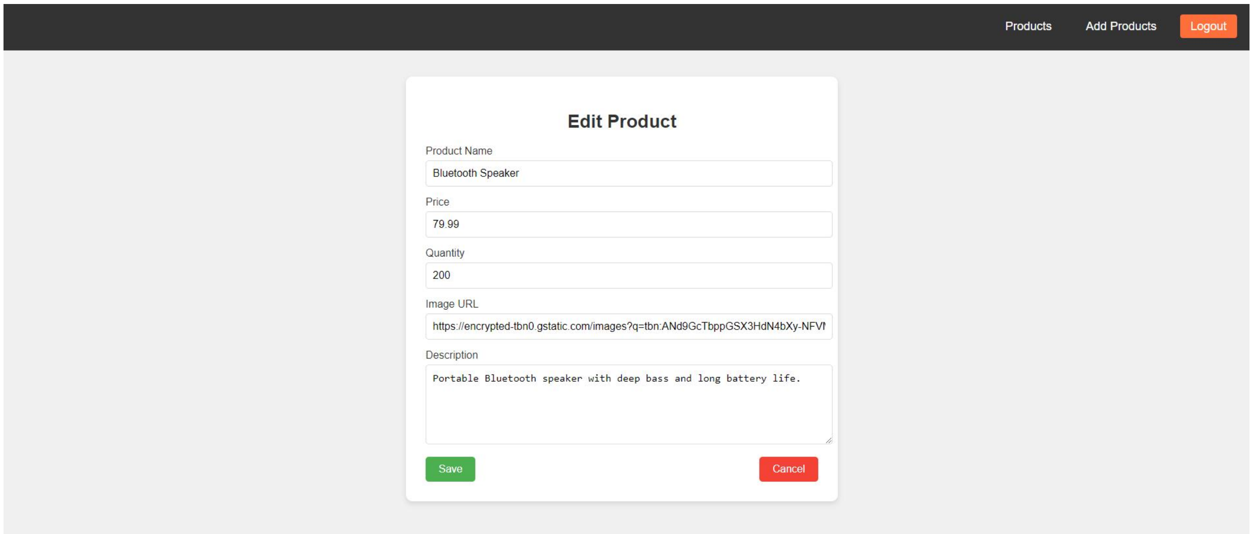
Wireless Headphones

$99.99

In Stock: 149

Edit Delete

Edit Product



Edit Product

Product Name

Bluetooth Speaker

Price

79.99

Quantity

200

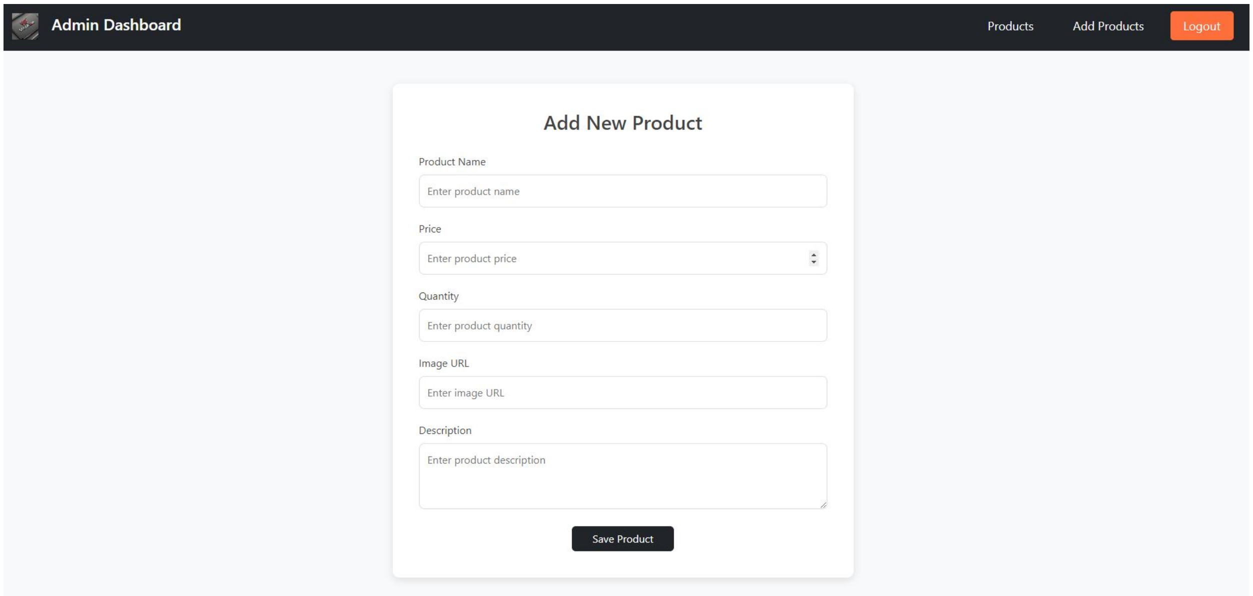
Image URL

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTbppGSX3HdN4bXy-NFVM

Description

Portable Bluetooth speaker with deep bass and long battery life.

Add Product



Add New Product

Product Name

Enter product name

Price

Enter product price

Quantity

Enter product quantity

Image URL

Enter image URL

Description

Enter product description