Vanier College

Faculty of Science and Technology

System Development

420-951-VA sect. 01222

FINAL REPORT

UNBOX HOLIDAYS E-COMMERCE WEBSITE

Team Members:

Ana Pechonkin (0972547)

Belal Mohsen (2395009)

Diana Majolli (2328346)

Shunzi Yao (2395028)

TRANSACTIONAL WEB APPLICATIONS

Instructor: Mohammad Ali Hasheminezhad

January 28th, 2024

UNBOX HOLIDAYS

Project Aim

"Unbox Holidays" aims to revolutionize holiday decoration by offering uniquely curated blind boxes, bringing joy and surprise to seasonal decorating. Our goal is to deliver holiday cheer to every home through personalized and delightful decorative items.

Project Description

The website is a user-friendly platform designed to provide customers with an engaging and seamless experience. It allows users to subscribe or make single purchases of holiday boxes, tailored to their decorating preferences. The site includes innovative features like an interactive chatbot, a customization questionnaire, and an AI-powered image generator to preview decorations.

Functional Requirements:

I. **User Interface and Experience:**

- A user-friendly and intuitive interface for easy navigation and interaction.
- Mobile-responsive design to ensure accessibility across various devices.

II. **Account Management:**

- Secure user registration and login process, including social media authentication options.
- Profile management features allowing users to update personal information, preferences, and password.

III. **Subscription and Purchasing:**

- Facility for users to subscribe to regular holiday boxes or make single purchases.
- Customization options for subscribers through a detailed questionnaire.

IV. **Shopping Features:**

- An interactive shopping cart for reviewing and modifying selected items before purchase.
- Secure and versatile payment gateway integration, supporting multiple payment methods.

V. **Order and Subscription Management:**

- User dashboard to track and manage orders, subscriptions, and delivery status.
- Automated email notifications for order confirmation, shipping updates, and subscription renewals.

VI. **Customer Support:**

- An AI-powered chatbot for real-time customer assistance and query resolution.
- A feedback system allowing users to rate and review products and services.

VII. **AI and Data Analytics:**

- AI image generator to provide a preview of holiday decorations in the subscription boxes.
- Data analytics integration for analyzing customer behavior, preferences, and improving service.

Non-functional Requirements:

I. **Performance and Scalability:**

- High-performance website with quick load times and efficient handling of high traffic volumes.
- Scalability to accommodate growing user numbers and data volumes.

II. **Security:**

- Robust security measures to protect user data, including secure database practices and data encryption.
- Compliance with data protection regulations (like GDPR).

III. **Reliability and Availability:**

- High system availability with minimal downtime.
- Reliable and consistent performance across all features.

IV. **Maintainability and Support:**

- Codebase and architecture designed for ease of maintenance and updates.
- Regular updates and patches to improve functionality and security.

V. **Integration:**

- Seamless integration of third-party APIs for address validation, payment processing, and social media authentication.
- Compatibility with various browsers and devices.

User Stories

Subscription and Single Purchase Options
As a user, I want to have the option to either subscribe for regular holiday boxes or make a single purchase, So that I can choose the best option that suits my needs and preferences for holiday decorations.

Interactive Chatbot for Queries
As a user, I want to interact with a chatbot on the website, So that I can get quick answers to my queries and assistance with my shopping decisions.

Customization Questionnaire for Personalization
As a potential subscriber, I want to fill out a customization questionnaire before subscribing, So that the holiday boxes I receive are tailored to my personal style and preferences.

AI-Generated Preview of Decorations
As a user interested in holiday decorations, I want to use an AI image generator to preview decorations, So that I can get a general idea of what to expect in my holiday box before making a purchase.

Secure Account Creation and Login
As a new user, I want an easy and secure process to create and log into my account, So that I can manage my subscriptions and orders with confidence and ease.

Subscription Management in User Profile
As a returning user, I want to easily manage my subscription preferences in my user profile, So that I can customize, update, or cancel my subscriptions as per my changing needs.

Responsive and Mobile-Friendly Website
As a user, I want the website to be responsive and easy to navigate on any device, So that I can browse, order, and manage my subscriptions conveniently, whether I'm on a desktop or using a mobile device.

Test Cases -- ADD_TO_CART test

This report covers four unit tests for the addItemToCart Redux action. The tests are divided into two categories: The first two tests verify the correct formation and dispatch of actions, while the latter two tests confirm the functionality of adding items to the cart and updating the state accordingly.

1. **Payload Dispatch Test for New Item:**

- Objective: To verify that the correct action and payload are dispatched when adding a new item to an empty cart.
- Method: Utilizes configureMockStore to create a mock store with an empty initial cart state, and then dispatches the addItemToCart action with a new item.
- Expected Outcome: The action type ADD_TO_CART is dispatched with the payload containing the new item's details.
- Result: Passed

2. **Payload Dispatch Test for Existing Item:**

- **Objective**: To ensure that the correct action and payload are dispatched when adding an item that already exists in the cart.
- **Method**: A mock store is set up with an existing item in the cart. The addItemToCart action is dispatched with the same item.
- **Expected Outcome**: The action type ADD_TO_CART is dispatched, indicating an increment in the item's quantity.
- **Result**: Passed

3. **State Update Test for New Item:**

- **Objective**: To confirm that the Redux store's state updates correctly when a new item is added to the cart.
- **Method**: A real Redux store is created with the cart reducer. The addItemToCart action is dispatched with a new item.
- **Expected Outcome**: The state of the store updates to include the new item with a quantity of 1.

- **Result**: Passed

4. **State Update Test for Existing Item:**

- **Objective**: To verify that the Redux store's state updates accurately when the quantity of an existing item is incremented.
- **Method**: A store with preloaded state containing an existing item is used. The addItemToCart action is dispatched with the same item.
- **Expected Outcome**: The state of the store reflects the incremented quantity of the existing item.
- **Result**: Passed

Conclusion:
The combination of these tests provides a comprehensive validation of the addItemToCart action in the "Unbox Holidays" e-commerce platform. The first two tests confirm that the action correctly dispatches the expected payloads, while the latter two tests ensure that these actions effectively update the shopping cart's state in the Redux store. This dual approach to testing ensures both the correctness of actions dispatched and the integrity of the state management in the application.

Individual Role and Responsibilities

- **UI Interface (material design, activity flow)**
  The team created the following pages:
  - Diana: FAQ Page, Questionnaire Page, Image Generation Page
  - Chelsea: Home Page, Sign Up Page, Chatbot
  - Belal: Shopping Cart Page, User Profile Page
  - Ana: Log In Page, Forgot Password Page, Calendar Page, Contact Us Page, Admin Page

- **Database Connection and CRUD operations:**

  - Belal: User, Subscription, Payment Collections and Routes and Database connection. Admin Page back-end functionality
  - Ana: Box and Order Collections and Routes, Contact Us functionality (Email.js)

- **API calls (JSON, Async, Threading)**
  - Diana: Mock API for image generation
  - Belal: Stripe API for the payment process
  - Ana: Mapbox API for the address autofill

- **CI/CD**
  - Chelsea: Configured a CI/CD pipeline to automatically deploy to AWS EC2

- **Hosting (on an AWS server:**
  IP Address: http://ec2-3-99-217-162.ca-central-1.compute.amazonaws.com/)
  - Chelsea and Diana

- **Authentication: Firebase (email/password, Google, Facebook)**
  - Ana

- **Internalization and Localization**
  - Diana

- **Issue Tracking Software: JIRA**
  Everyone worked on it, so we each did ¼ of the work.

- **Final Report**
  - Chelsea