# **Assignment 4: "Coffee Corner" - Enhanced Web and Mobile Application**

**Contributors:**

* Likhita Sai Matta (ID: 101293458)
* Jaywant Vijaykumar Patel (ID: 101321483)

**GitHub Repository:** [LikhitaSaiMatta/Web-and-Mobile-Development-Assignments](https://github.com/LikhitaSaiMatta/Web-and-Mobile-Development-Assignments)

## **Overview**

Coffee Corner has evolved from a simple web application to a full-fledged web and mobile platform using React for the web frontend and Ionic for the mobile application. This report outlines the key features and improvements made to the project.

## **What We Did**

## Web Application (React)

1. Component-Based Architecture
   * Converted HTML structure to React components
   * Created separate components for Header, Footer, Home, About, CafeList, CafeDetails, Login, and Signup
2. State Management
   * Implemented React Context API for global state management
   * Created AuthContext for handling user authentication state
3. Routing
   * Utilized React Router for navigation between different pages
   * Implemented protected routes for role-based access control
4. User Authentication
   * Developed login and signup functionality using JWT
   * Integrated with backend API for user authentication
5. Role-Based Access Control
   * Implemented different views and permissions for customers and store managers
   * Restricted certain actions (e.g., adding new cafes) to manager accounts only
6. Cafe Management
   * Created CafeList component to display all cafes
   * Implemented CafeDetails component for individual cafe information
   * Added functionality for managers to add new cafes
7. Review System
   * Developed a review submission form for authenticated users
   * Displayed cafe ratings and reviews on cafe detail pages

## Mobile Application (Ionic)

1. Cross-Platform Development
   * Utilized Ionic framework to create a mobile version of Coffee Corner
   * Ensured consistent user experience across iOS and Android platforms
2. Native Features Integration
   * Leveraged Ionic's Capacitor for accessing device features (e.g., camera for uploading cafe images)
3. Responsive Design
   * Adapted the web application's UI to fit mobile screens
   * Implemented touch-friendly interfaces for better mobile user experience
4. Offline Functionality
   * Implemented data caching for offline access to cafe information
   * Synchronized data with the server when the device comes back online
5. Push Notifications
   * Integrated push notification service for user engagement
   * Sent notifications for new cafe additions or review responses

## Backend Integration

1. API Consumption
   * Connected React and Ionic frontends to the NestJS backend
   * Utilized Axios for making HTTP requests to the server
2. Data Persistence
   * Integrated with MongoDB for storing user data, cafe information, and reviews
   * Ensured data consistency across web and mobile platforms

## **Testing and Validation**

1. Component Testing
   * Implemented unit tests for React components using Jest and React Testing Library
   * Ensured proper rendering and functionality of key components
2. End-to-End Testing
   * Conducted comprehensive testing of user flows on both web and mobile platforms
   * Verified data persistence and synchronization between frontend and backend

## **Conclusion**

The addition of React for the web frontend and Ionic for the mobile application has significantly enhanced the Coffee Corner project. These technologies have enabled the creation of a responsive, user-friendly, and feature-rich application accessible on both web and mobile platforms. The implementation of state management, authentication, and role-based access control has improved the overall security and functionality of the application.

### **Contributions**

**Note:** The commits for this assignment were made after completing all tasks, as we worked collaboratively in the library.

### **Likhita Sai Matta**

* Converted HTML structure into React components for modularity.
* Created components: Header, Footer, Home, About, CafeList, CafeDetails, Login, Signup.
* Used React Context API for global state and AuthContext for authentication.
* Implemented React Router with protected routes for role-based access.
* Integrated JWT-based login/signup with backend API.
* Developed customer and store manager views with permissions.
* Built CafeList and CafeDetails components with review and rating functionality.

### **Jaywant Vijaykumar Patel**

* Built a mobile version using Ionic for iOS and Android.
* Used Capacitor for device features (e.g., camera for image uploads).
* Designed a touch-friendly, responsive UI for mobile screens.
* Enabled offline access and data syncing with server.
* Added push notifications for new cafes and review updates.