PA0 - Project Proposal

Justatea-F&B E-Commerce Website

1. Introduction

The proposed project - **Justatea** aims to develop a comprehensive e-commerce website for the food and beverage (F&B) industry. The website will serve as an online platform for customers to browse, customize, and purchase a wide range of F&B products from various vendors or branches. By leveraging modern web technologies and incorporating user-friendly features, the website will provide a seamless and convenient shopping experience for customers while streamlining operations for vendors.

2. Target Users and Environment

The primary target users for this e-commerce website are:

1. **Customers**: Individuals seeking to purchase F&B products online, including those with specific dietary preferences or requirements.
2. **Vendors/Restaurants**: F&B businesses that want to expand their reach and offer their products through an online platform.

The website will be accessible via various devices, including desktops, laptops, tablets, and smartphones, ensuring a consistent and responsive user experience across multiple platforms.

3. Key Features

3.1. User Authentication

- **Login**: Allow users to log in with their email and password, enabling access to personalized features and account information.
- **Sign Out**: Provide users with the ability to securely sign out of their current session.
- **Register**: Enable new users to create an account by collecting necessary information (name, email, password, etc.) and implementing email verification for account activation.
- **Password Reset**: Implement a password reset functionality for users to securely reset their passwords.

3.2. Shopping Cart

- **Add to Cart**: Allow users to add products to their shopping cart, displaying the number of items in the cart.
- **Change Item Amount**: Enable users to increase or decrease the quantity of items in their cart.
- **Remove Item**: Provide users with the option to remove items from their cart.

3.3. Product Details

- **Product Information**: Display detailed information about each product, including name, description, price, ingredients, and high-quality images or videos.
- **Customize Order**: Allow users to customize their orders by adding or removing ingredients, choosing portion sizes, or specifying dietary preferences.

3.4. Checkout Process

- **Contact Information**: Collect the user's contact information (name, email, phone number, etc.) during the checkout process.
- **Payment Method**: Integrate with a secure payment gateway (e.g., PayPal, Stripe) to process payments.
- **Default Address**: Enable users to save their default shipping address for future orders.
- **Vouchers**: Implement a system for applying voucher codes or discounts during checkout.

3.5. Search and Filtering

- **Search**: Allow users to search for products by name, category, or other relevant criteria.
- **Sort**: Enable sorting of products based on price, popularity, ratings, or other factors.
- **Choose Branch**: Allow users to select the branch or location they want to pick up or have their order delivered from.

3.6. Reviews and Recommendations

- **Customer Reviews**: Enable users to leave reviews and ratings for products they've purchased, and display these reviews on product pages.
- **New Product**: Highlight newly added products on the homepage or a dedicated section.
- **Best Seller**: Feature the best-selling or most popular products on the homepage or a dedicated section.

3.7. User Dashboard

- **Order History**: Allow users to view their past orders and order details.
- **Saved Addresses**: Enable users to save and manage multiple shipping addresses.
- **Account Settings**: Provide users with the ability to update their account information (password, email, etc.).

3.8. Chat Bot

- Implement a chatbot using a service like Gemini and integrate it into the website for easy access, enabling users to get instant assistance and address common queries.

4. Project Timeline and Milestones

Week 1 (June 17 - June 23):

- Project kickoff and planning

- Define project scope and requirements
- Set up development environment (code editors, Firebase project, etc.)
- Design wireframes and user interface mockups
- Deadline: PA0 & PA1 (June 23, 2024, 22:00)

Week 2 (June 24 - June 30):

- Finalize UI/UX design
- Develop frontend components (React.js)
- Implement user authentication (Firebase Authentication)
- Set up Firebase Realtime Database or Cloud Firestore

Week 3 (July 1 - July 7):

- Develop shopping cart functionality
- Implement product details and customization features
- Integrate payment gateway (e.g., PayPal, Stripe)
- Set up Firebase Cloud Functions for server-side logic
- Deadline: PA2 (July 7, 2024, 22:00)

Week 4 (July 8 - July 14):

- Implement search, filtering, and sorting features
- Develop user dashboard (order history, saved addresses, account settings)
- Integrate chatbot (Gemini) into the website
- Set up Firebase Cloud Storage for user file storage

Week 5 (July 15 - July 21):

- Implement reviews and recommendations (customer reviews, new products, best sellers)
- Conduct thorough testing (unit tests, integration tests, end-to-end tests)
- Address any bugs or issues identified during testing
- Set up Firebase Hosting for static content delivery
- Deadline: PA3 (July 21, 2024, 22:00)

Week 6 (July 22 - July 28):

- Final testing and debugging
- Deployment and go-live preparation
- Set up Firebase Analytics, Crashlytics, and Remote Config
- Conduct user acceptance testing
- Launch the website and monitor performance
- Deadline: PA4 (August 4, 2024, 22:00)
- Deadline: PA5 (August 11, 2024, 22:00)

5. Technical Requirements and Resources

Programming Languages

- JavaScript (React.js frontend, Node.js backend)

Frameworks & Libraries

- React.js (18.3.1)
- React Bootstrap (2.10.2) - UI component library
- React Hook Form (7.51.5) - Form handling library
- Axios (1.7.2) - HTTP client

Databases

- Firebase Realtime Database or Firebase Cloud Firestore (real-time data sync)

Backend Services

- Firebase Cloud Functions (server-side logic)
- Firebase Authentication (user auth)
- Firebase Cloud Storage (user file storage)
- Firebase Analytics (usage analytics)
- Firebase Crashlytics (error reporting)
- Firebase Remote Config (remote config management)

Development Tools

- Node.js (20.14.0)
- npm or yarn (package management)
- Create React App
- Firebase CLI
- Git (version control)
- React Testing Library (React component testing)
- Firebase Emulators (local service emulation)

Third-Party Integration

- Social media login APIs

Required Resources

Team

- 2 x Frontend Developers
- 1 x Backend Developer
- 1 x UI/UX Designer
- 1 x Project Manager

Development Environment

- Code editors (Visual Studio Code, etc)
- Firebase project with services setup

Hosting

- Firebase Hosting (static content)
- Cloud hosting for backend/server-side components

Additional Tools

- Project management (Jira, Trello, etc)
- Documentation (Google Docs, Confluence, etc)

6. Budget and Cost Estimation

Development:

- Development Environment: Free (using open-source tools like Visual Studio Code, Git, and Firebase Emulators)
- Firebase Services:
    - Firebase Authentication: Free (Spark plan)
    - Firebase Realtime Database: Free (Spark plan, up to 1GB storage)
    - Firebase Cloud Firestore: Free (Spark plan, up to 1GB storage)
    - Firebase Cloud Functions: Free (Spark plan, up to 2 million invocations per month)
    - Firebase Cloud Storage: Free (Spark plan, up to 5GB storage)
    - Firebase Hosting: Free (Spark plan)

Hosting:

- Firebase Hosting: Free (Spark plan, for static content delivery)
- Cloud Hosting for Backend/Server-side Components: $0 (using Firebase Cloud Functions)

Third-Party Services:

- Gemini Chatbot Integration: Free (basic plan)
- Payment Gateway Integration:
    - PayPal: No setup fee, transaction fees apply (2.9% + $0.30 per transaction)

Additional Costs:

- Maintenance and Support: $0 (assuming minimal maintenance and support requirements during the project duration)
- Future Updates: $0 (assuming no major updates or enhancements beyond the project scope)

Total Estimated Cost: $0 (excluding potential transaction fees for payment gateways)

7. Risks and Mitigation Strategies

Technical Challenges:

Risk: Lack of experience with the chosen technologies (React.js, Firebase, etc.) may lead to delays and technical difficulties during development.

Mitigation Strategy:

- Allocate time for team members to familiarize themselves with the technologies and attend relevant training or tutorials.
- Seek guidance from experienced developers or mentors when facing technical obstacles.
- Leverage online resources, documentation, and community forums for troubleshooting and problem-solving.

Timeline Delays:

Risk: Underestimating the complexity of certain features or tasks may result in timeline delays and missed deadlines.

Mitigation Strategy:

- Conduct thorough planning and break down tasks into smaller, manageable chunks.
- Regularly review progress and adjust timelines as needed.
- Implement agile methodologies (e.g., Scrum) to ensure flexibility and adaptability.
- Prioritize critical features and functionalities to ensure timely delivery of the minimum viable product (MVP).

Security Concerns:

Risk: Inadequate security measures may lead to data breaches, unauthorized access, or other security vulnerabilities.

Mitigation Strategy:

- Implement industry-standard security practices, such as encryption, secure authentication, and secure communication protocols.
- Regularly update dependencies and libraries to address known security vulnerabilities.
- Conduct security testing and penetration testing to identify and address potential vulnerabilities.
- Implement secure coding practices and follow best practices for handling sensitive data (e.g., payment information).

Compliance Issues:

Risk: Failure to comply with relevant regulations, such as data privacy laws or payment processing regulations, may lead to legal issues or penalties.

Mitigation Strategy:

- Research and understand applicable regulations and compliance requirements (e.g., GDPR, PCI-DSS).
- Implement appropriate data privacy and security measures to ensure compliance.
- Consult legal experts or compliance professionals for guidance on regulatory requirements.

- Regularly review and update policies and procedures to maintain compliance as regulations evolve.

Resource Constraints:

Risk: Limited resources, such as budget constraints or team availability, may impact the project's progress and quality.

Mitigation Strategy:

- Prioritize and focus on essential features and functionalities.
- Explore cost-effective solutions and leverage free or low-cost tools and services.
- Implement resource management strategies, such as task delegation and time management.
- Consider seeking additional resources or adjusting the project scope if necessary.

Integration Challenges:

Risk: Difficulties in integrating third-party services (e.g., payment gateways, chatbot) may cause delays or functionality issues.

Mitigation Strategy:

- Thoroughly research and test third-party integrations during the planning and development phases.
- Maintain open communication with third-party service providers for support and issue resolution.
- Have contingency plans or alternative solutions in case of integration failures.

8. Conclusion

The proposed F&B e-commerce website project aims to provide a user-friendly and comprehensive online platform for customers to browse, customize, and purchase F&B products from various vendors or branches. By incorporating modern web technologies and implementing essential e-commerce features, the website will enhance the shopping experience for customers while streamlining operations for vendors. With a robust feature set, secure payment integration, and a responsive design, the website has the potential to become a leading online destination for F&B enthusiasts.