**Personalized News Website**

**Abstract**

This project report provides a detailed version of the design and implementation of a personalized news website. The website aims to enhance the efficiency and quality of information retrieval by delivering tailored news content based on user preferences and interests. The project achieves dynamic content loading and interactive user interfaces by employing contemporary web development technologies, including the React.js framework and REST API interactions.

**Project Background**

In the digital age, the accessibility of news and information has never been easier, yet this has also led to the issue of information overload. Users often feel overwhelmed when searching for relevant news. A personalized news website leverages user behavior data and preference settings to filter out the most relevant news, thereby improving the efficiency of information acquisition and the reading experience.

**Project Objectives**

1. Develop a responsive, user-friendly news website.
2. Implement a personalized news recommendation algorithm to display news based on user preferences.
3. Provide a comprehensive user preference setting system, including interests, age, and gender.
4. Implement user profile management functionality, allowing users to save and manage multiple profiles.
5. Ensure the scalability and maintainability of the website for future development.

**Technology Stack**

* **Frontend Framework**: React.js
* **Styling**: CSS / Bootstrap
* **API Interaction**: JavaScript Fetch API
* **State Management**: React Context API
* **Storage**: local storage
* **Testing**: Jest / React Testing Library

**System Design**

**Architecture**

The system adopts the MVC (Model-View-Controller) architecture, separating the frontend interface, business logic, and data models. React components are responsible for rendering views, JavaScript files handle business logic, and CSS files manage styling.

**Main Components**

1. **API.js**: Handles interactions with the news API, including fetching top headlines and searching for news.
2. **App.js**: The main React component defining the overall layout and user interface of the application.
3. **ProfileContext.js**: A React context for managing the state and operations related to user profiles.
4. **display.js**: Contains functions for rendering news articles and attaching "like" and "save" button event listeners.
5. **interactions.js**: Defines functions for user interactions with news articles, such as favoriting and saving articles.
6. **preferences.js**: Functions for saving, retrieving, and updating user preference settings.

**User Interface**

* **index.html**: The onboarding page for collecting user preferences.
* **dashboard.html**: The personalized news dashboard displaying news filtered by user preferences.
* **profile.html**: The profile management page allows users to create and manage profiles.
* **settings.html**: The settings page for updating preference settings and adjusting accessibility options.

**Functional Implementation**

**News Acquisition and Caching**

Interact with the news API through API.js to fetch news data and cache it in localStorage for improved performance. This process involves asynchronous data requests and error handling to ensure data reliability and a smooth user experience.

**Personalized Recommendations**

Dynamically display relevant news based on user preferences set on the onboarding page. This feature analyzes user-inputted preference data in conjunction with news content categorization and tagging to personalized news recommendations.

**User Interaction**

Users can like and save news articles, with these actions implemented through interactions.js and stored in localStorage. This feature enhances user engagement and allows users to quickly access articles of interest on subsequent visits.

**Profile Management**

Users can create and manage profiles on the profile.html page, with these profiles stored in localStorage, allowing users to sync preferences across different devices. This feature offers flexibility and convenience for a personalized user experience.

**Testing and Validation**

Unit and integration tests are conducted on React components using Jest and React Testing Library to ensure the correctness and stability of the functions. Testing covers user interface interactions, API calls, and state management.

**Conclusion**

The project successfully implemented a personalized news website that provides customized news content through user preference settings. The website boasts a good user experience and scalability, laying the foundation for further functional expansion and optimization in the future.