Title: Barba - Barber & Salon Booking React Native Expo Ui Kit

Table of contents:

What is Barba?

Barba - Barber & Salon Booking React Native Expo Ui Kit, a comprehensive and feature-rich mobile UI kit designed to empower developers and designers in creating intuitive and efficient barber and salon booking appointment applications.

With Barba, developers can save valuable time and effort by leveraging its pre-built UI components, templates, and ready-to-use features. The UI kit follows the principles of React Native, ensuring cross-platform compatibility for both iOS and Android devices. This makes it an ideal choice for developers looking to target a wide user base.

Use cases for this template:
- Barber Shop Booking App
- Beauty Salon Appointment App
- Haircut & Fashion App
- Messenger App (Chatting & Call)

Key features of Barba:
- Built with React Native Expo
- Only Functional Components & React Hooks
- Well structured components
- Clean and Easy Code with Clear Navigation.
- This UI Provides Easy, Enjoyable, and Effective Interaction Between the User and the App
- Organized Layers and Groups.
- Responsive design to any device screen
- Multipurpose screens
- Search & Filter Salons, Barbers, Review Features
- Multiple Payments & Transactions Management
- Messenger Features (Chat & Calls Management)
- Onboarding, Sign up, Sign in, Reset Password, Account Setup, Notification, Bookmark, Search & Filter, Reviews, Help Center (FAQ & Contact Support), Profile & Settings

- ● Dark Mode Support

Technologies used

- ● React Native
- ● Expo
- ● React Native Gifted Chat

How to Start Barba?

To kickstart your Barba project, follow these step-by-step instructions:

1. Install Expo

- Open a terminal or command prompt.
- Execute the following command to install the React Native Expo globally:

```
npm install -g expo-cli
```

2. Navigate to your Project Directory

- Open a terminal or command prompt.
- Use the `cd` command to navigate to the directory where your Barba project is stored.

3. Install Project Dependencies:

- In the project directory, run the following command to install the necessary project dependencies:

```
npm install
```

4. Start the Development Server

- Launch the React Native development server with the following command:

```
npm run start
```

5. Choose How to Run Your App:

- Once the development server is running, it will display relevant information.
- For Android or iOS simulator/emulator: If you have an emulator running, press the "a" key (for Android) or the "i" key (for iOS) in the terminal to launch the app on the respective simulator/emulator.
- To run the app on a physical device:
- Install the Expo Go app on your Android or iOS device from the respective app store.
- Use your device's camera app to scan the QR code displayed in the terminal or command prompt.
- This will open the Expo Go app and load your app.

6. Test and Develop Your App:

- With your app running, you can make code changes and observe automatic updates in the simulator/emulator or on your physical device.
- Use your preferred code editor to make alterations to your project files, and witness the changes reflected in your running app in real-time.

Project folder structure

- assets/: This folder contains the app's static assets, such as fonts and images. fonts/: Contains the font files used in the app. icons/: Contains the icons files used in the app. images/: Contains image assets used in the app.
- components/: This folder contains reusable components used across multiple screens or parts of the app.
- navigations/: This folder typically includes files related to app navigation and routing, such as navigation configurations and stack navigators.
- screens/: This folder holds the individual screens or pages of the app. Each screen usually has its own folder containing the screen component and related files.
  - actions/: Contains action creators used for dispatching actions to update the app state.
  - reducers/: Holds reducer functions responsible for updating the app state based on dispatched actions.
- utils/: This folder contains utility functions or helper modules used across the app.
- App.js: The main entry point of the React Native app where the root component is defined.
- package.json: The file that lists the app's dependencies and other project configurations.
- Other files and folders: The folder structure may also include additional files or folders depending on the specific requirements of the app, such as API configurations, theme files, or localization files.

How To Use Barba

Integrating Barba into your React Native project is a straightforward process.

- Import the necessary components and screens seamlessly.
- Establish intuitive navigation flows to guide users through different sections Customize UI elements to align with your branding and design preferences.
- Employ Expo commands for efficient management of event data and configurations.
- Connect your own API

Support

Thank you for purchasing this product! Should you have any questions about it, please don't hesitate to reach out to us at lalacorp.contact@gmail.com. For customization or any custom web and mobile app development needs, feel free to contact us at hello@lalaweb.dev. We're here to assist you!