Al-Imam University

Computer Science Department

College of Computer and Information Sciences

Project: Mobile App for the Health and Wellness Sector

[CS447]

أزهلني

Member Names	Member IDs
Raghad AlShanar	444008577

Instructor: Dr.Faten aljaloud

Section: 371

Table of Contents

1. App Description
2. Target Problem
3. Key Features
4. Advanced React Native Features Implemented
5. Technology Stack
   5.1 Frontend (React Native + Expo)
   5.2 Backend (Node.js + Firebase)
6. Challenges & Solutions
7. Refrences

1.App Description

إزهلني is a mobile nutrition app designed to simplify daily meal planning for users with specific dietary goals. The name is derived from the Najdi phrase:

"ازهل الأمر" أو "ازهله" - meaning "leave it to me.I'll take care of it."

That's the spirit and main idea of the app. It takes care of meal planning, calorie counting, and measurement calculation for the user who needs to stick to a calorie goal. Not only that, but it also takes care of the macros and reminds the users so that they don't have to stress over what to eat.

Just set the goals and إزهلني does the rest for you, considering all of your needs.

2. Target Problem

Many people, especially students, struggle with maintaining a calorie and macros goal due to a lack of time, knowledge, personalized guidance, or what is most importantly lack of nutrition apps that account for cultural restrictions. إزهلني solves this problem by:

- Generating meals with halal filtering
- Offering clear macro tracking and progress visualization
- Providing the meal recipe and the macros of it, calculated and prepared according to the user's goals
- Sending meal reminders and encouraging user engagement using streak

3. Key Features

1. **User Authentication:** Login, registration, and persistent sessions using Firebase Auth and AsyncStorage.
2. **Set Nutrition Goals:** Users can set and edit daily calorie, protein, carb, and fat goals.
3. **Meal Generation:** Meals are generated based on user goals using the Spoonacular API, with backend filtering for halal meals.
4. **Meal Details:** Each meal card shows total macros, and the details page displays ingredients and full recipe instructions.
5. **Meal History:** Firestore-stored history of meals eaten, displayed in the app.
6. **Progress Tracking:** Circular progress bars show daily macro progress.
7. **Notifications:** Local notifications remind users of meal times (breakfast, lunch, dinner) using Expo Notifications.
8. **Custom UI:** Consistent design with gradients, icons, and responsive layouts.
9. **Streak (User Engagement):** maintain a streak of days that the user has reached their goals to encourage healthy eating habits

4. Advanced React Native Features Implemented

- REST API Integration (Spoonacular + custom backend)
- Local push notifications with expo-notifications
- Persistent local data using AsyncStorage
- Circular progress visualization for daily goals
- Firebase Authentication with Firestore backend
- RTL support for a full Arabic experience

5. Technology Stack

5.1 Frontend (React Native + Expo)

- **React Native:** Main framework for building the app.
- **Expo:** for development, building, and running the app. Access to device APIs, etc.
- **Expo-notifications:** For local scheduling of meal reminders and handling notification permissions.
- **Expo-linear-gradient:** For visually appealing gradient backgrounds.
- **Expo-vector-icons:** For icons in the app.
- **Expo-constants:** For accessing device and app constants and changing them.
- **react-native-animatable:** For smooth UI animations (used in logo fade-in).
- **React-native-progress:** For displaying circular progress bars for macros.
- **React-navigation:** For stack and tab navigation between screens.
  - @react-navigation/native
  - @react-navigation/native-stack
  - @react-navigation/bottom-tabs
- **react async-storage:** For local storage of user tokens and info.
- **Axios:** For making HTTP requests to the backend and external APIs.
- **date-fns:** For date formatting.
- **Firebase:** For authentication and Firestore database integration.
- **React-native-safe-area-context:** For handling safe area insets.

5.2 Backend (Node.js + Express)

- Express.js - RESTful API server
- Firebase Admin SDK - To interact with Firestore(store goals, meals,etc)
- dotenv - Secure management of API keys and config
- axios - To fetch data from the Spoonacular API

- Spoonacular API

6. Challenges & Solutions

During the development of the project, many technical and design challenges occurred, mostly related to logic, functionality, and connecting the frontend with the backend. Not much with the front-end, it was quite enjoyable

Challenge	Solution / Experience
Filtering culturally mismatched meals from Spoonacular	Implemented backend filtering logic to exclude non-halal ingredients like pork and alcohol-based recipes from API results.
Couldn't test Cross-platform push notifications in Expo Go	Switched to an Expo development build
Tried to use Expo permissions	Used the built-in permission handling by expo-notifications, which worked without needing the separate package.
Designing the needed UI using react native, as it doesn't offer much	Had to research and use a lot of third-party libraries
Routing between the frontend and the backend is correctly	Took time to align HTTP requests, test endpoints, and ensure data flowed smoothly between Firebase, the backend server, and frontend components.
Managing full-stack logic alone as a solo developer	Required extra effort in structuring code, maintaining API logic, syncing data, and testing features end-to-end across the stack.
Frontend development overall	Surprisingly smooth and enjoyable, working with React Native made interface building intuitive once the logic was handled. Discovered a new appreciation for the framework.

7. References

1. **AsyncStorage**
   https://react-native-async-storage.github.io/async-storage/
2. **Expo Notifications**
   https://docs.expo.dev/versions/latest/sdk/notifications/
   https://www.geeksforgeeks.org/how-to-schedule-local-push-notification-in-react-native-using-expo/
3. **react-native-progress (Circular Bars)**
   https://github.com/oblador/react-native-progress
   https://medium.com/@muratcanyuksel/progress-circle-using-react-native-progress-78f193cd53ab
4. **react-native-animatable (Animations)**
   https://github.com/oblador/react-native-animatable
   https://www.geeksforgeeks.org/how-to-add-animations-in-react-native-using-react-native-animatable/
5. **expo-linear-gradient**
   https://docs.expo.dev/versions/latest/sdk/linear-gradient/
   https://blog.logrocket.com/create-gradient-backgrounds-react-native-expo-linear-gradient/
6. **Firebase Auth & Firestore**
   https://firebase.google.com/docs/auth
   https://www.geeksforgeeks.org/react-native-firebase-authentication-using-email-and-password/
7. **Spoonacular API (Halal Filtering)**
   https://spoonacular.com/food-api
   https://stackoverflow.com/questions/66636550/how-to-filter-out-pork-alcohol-from-meal-apis-results
8. **React Navigation**
   https://reactnavigation.org/docs/getting-started
   https://www.geeksforgeeks.org/react-native-navigation-stack-navigator/
9. **Firebase Admin SDK + Express.js**
   https://firebase.google.com/docs/admin/setup
   https://medium.com/firebase-developers/using-the-firebase-admin-sdk-with-node-js-express-8fcb0bded3c5
10. **Axios**
    https://github.com/axios/axios
    https://www.geeksforgeeks.org/how-to-make-api-calls-in-react-native-using-axios/