Project Plan & Overview
Prasanth_Late_2022

---

1. Project Overview

- This project aims to create a React Native application with token-based authentication and content retrieval from secure APIs.
- The application includes:
- A modern, login screen, home screen and detail screen.
- Secure token generation using the API: POST https://swsut62sse.execute-api.ap-south-1.amazonaws.com/prod/generateToken. (login)
- Token-based content fetching via: GET https://tzab40im77.execute-api.ap-south-1.amazonaws.com/prod/getContent. (detail)
- Centralized API client with robust error handling for all HTTP status codes.
- AsyncStorage-based token persistence for login sessions.
- Api=null for detail screen

2. Goals

- Implement a secure login flow (Token generated from email).
- Store and retrieve JWT or session tokens locally.
- Use fetch with centralized error handling and status code mapping.
- Fetch protected content with the saved token.
- Provide logout functionality to clear local tokens.
- Get data from backend using token
- Deliver a clean, reusable API architecture (apiClient.ts).

3. Key Features

- **Authentication Flow:**
❖ Login screen with email input.
❖ Generate and store token securely using AsyncStorage.
❖ Navigate users to content screen after login.
❖ Then details screen

- **API Integration:**
❖ generateToken API (POST).
❖ getContent API (GET) with token-based Authorization header.
❖ Common apiClient.ts to handle all requests (GET/POST).

- **Error Handling:** Centralized exception handling for HTTP status codes (400, 401, 403, 404, 500). .

- **Token Management:**
❖ Save token on login.
❖ Retrieve token for authenticated requests.
❖ Clear token on logout.

- **UI/UX:**
❖ Colorful, modern design (gradient backgrounds).
❖ Loading states for network calls.
❖ Alerts for success/error states.
❖ Custom html viewer for react native
❖ Unit test for Home screen

4. Technical Stack

Package and Plugins to install;

npx @react-native-community/cli@latest init AwesomeProject ==> Creating a new application

React native Navigation

npm install @react-navigation/native

npm install @react-navigation/native-stack

npm install react-native-screens react-native-safe-area-context

Storage Plugin for ROM

npm install @react-native-async-storage/async-storage

5. For standard Release

- generate a private signing key using keytool. must be run from

  C:\Program Files\Java\jdkx.x.x_x\bin

  keytool -genkeypair -v -storetype PKCS12 -keystore my-upload-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000

config files

- copy my-upload-key.keystoremy-upload-key.keystore from

  C:\Program Files\Java\jdkx.x.x_x\bin

  and paste it to android/app directory in your project folder.

- paste the below text in android/gradle.properties

  MYAPP_UPLOAD_STORE_FILE=my-upload-key.keystore
  MYAPP_UPLOAD_KEY_ALIAS=my-key-alias
  MYAPP_UPLOAD_STORE_PASSWORD=****
  MYAPP_UPLOAD_KEY_PASSWORD=*****

```
...
android {
    ...
    defaultConfig { ... }
    signingConfigs {
        release {
            if (project.hasProperty('MYAPP_UPLOAD_STORE_FILE')) {
                storeFile file(MYAPP_UPLOAD_STORE_FILE)
                storePassword MYAPP_UPLOAD_STORE_PASSWORD
                keyAlias MYAPP_UPLOAD_KEY_ALIAS
                keyPassword MYAPP_UPLOAD_KEY_PASSWORD
            }
        }
    }
    buildTypes {
        release {
            ...
            signingConfig signingConfigs.release
        }
    }
}
...
```

- paste this in android/app/build.gradle
- gradlew clean ==> to remove previus build cache
- gradlew assembleRelease ==> to generate standard build apk