

Practical Test

Duration: 3 days

This assignment will test your competencies on **React native**, **REST API**, and **Local Storage** Technologies. Go through the following real-world scenario and develop an application as expected. You can use React Native and any third-party libraries as you wish. You need to use **GitHub** as a version control platform and create a **Public repository**. Once you complete the test attach a repository URL with the email. Make sure you **perform commits** with each Screen or module.

Imagine there is a requirement for a Mobile application that allows users to buy products through the app. After the first installation users should be able to log in using the username and a password (with credentials provided). You can use the **Login REST API** for authentication. Once a user successfully logs in the following tabs should be available. (Please have a look at the wireframe given.)

1. **Home Tab**

All available products should list down here with the product title and thumbnail. You can use the **Products REST API** to fetch the data. Once a user presses on a particular product app should navigate into a Product information screen. You can get product information by using **Product Information REST API** to fetch the data. In product details, screen users should be able to add products with a quantity. These products added to the cart should be stored in **local storage**.

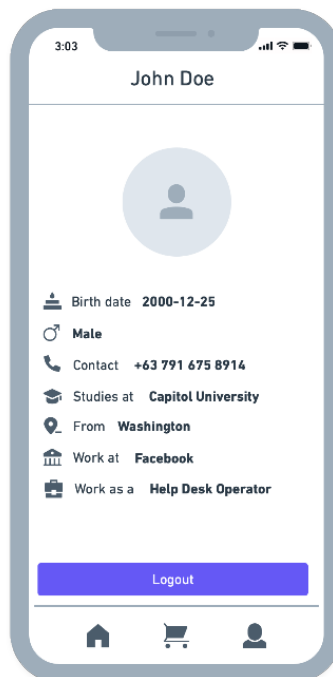
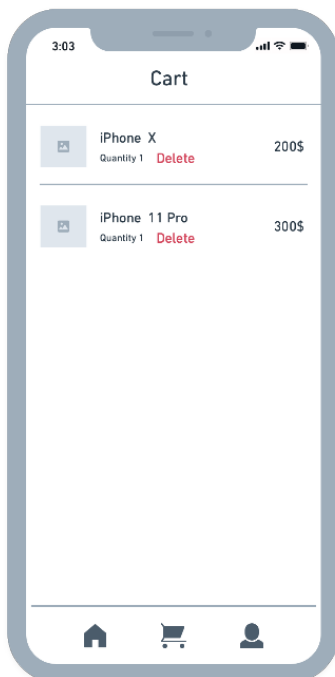
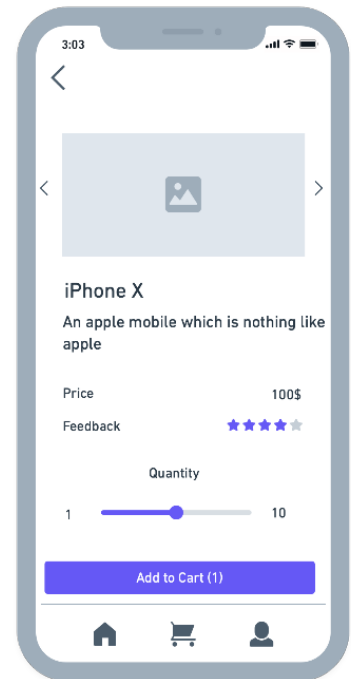
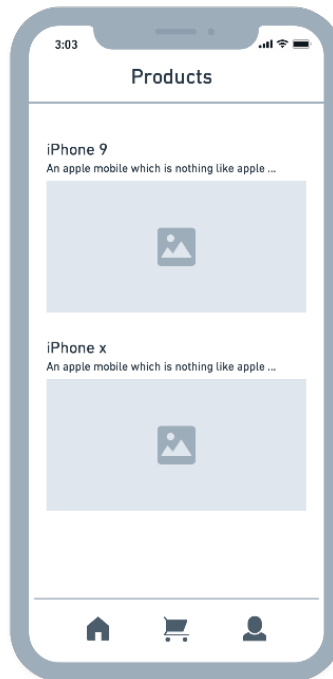
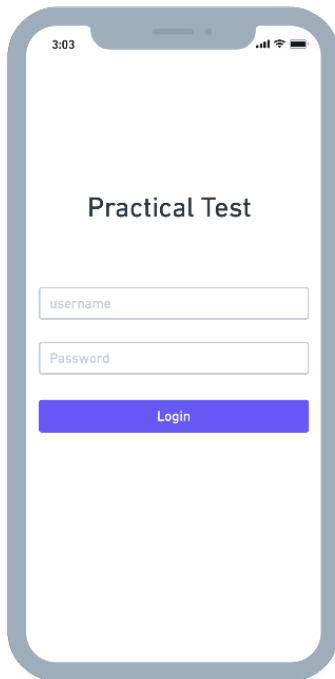
2. **Cart Tab**

As per the previous description, the products the user added to the cart should be able viewed in this tab as a list. And users should be able to remove items from the cart if they want.

3. **Profile Tab**

In this tab, users should be able to see their information. You can get all user information by using Single **User Rest API** And users should be able to log out from the application. Once a user clicks on logout, all user information should erase and navigate back to the login screen.

Please follow this wireframe



API Endpoints

Login REST API

POST <https://dummyjson.com/auth/login>

Content-Type: application/json

```
{  
  "username": "kminchelle",  
  "password": "0le1plR"  
}
```

Products REST API

GET <https://dummyjson.com/products>

Product Information REST API

GET <https://dummyjson.com/products/1>

User Rest API

GET <https://dummyjson.com/users/1>

NOTE: Keep in mind the following points when creating the Application.

1. Use OOP concepts along with Global state management.
2. Use proper validations where required.
3. Make sure your code is Decoupled, Testable, and Reusable.
4. Use Proper design patterns and always make sure you are following them.
5. You will not be allowed to use your previous codebase to develop this application.
6. Try to provide complete solutions within the given time to meet this requirement.
7. Use best practices and standards for Javascript/Java/Swift coding.