# Introduction

For this challenge, you are requested to create a simple product renting and buying/selling products application called Teebay (You can provide an alternate name if you like for creativity purposes). This solution will have Mobile App (FE), Back End (BE) and a DataBase (DB).

## PART 1: Preliminary features

For this part, please create the following preliminary features

- ○ Login
- ○ User registration

**IMPORTANT:** You need to implement login functionality using Biometrics and FaceID. Feel free to decide their placement based on your best UX understanding.

Assumptions for this part:

- You **do not** need to implement secure data transfer or password encryption for the Login feature. Simple string matching will suffice.

## PART 2: Implementation documentation

For this part, please create the following features -

- As a user, you can -
    - ○ Add your product (the design HAS to match as per wireframe. I.e. It has to be a multi page form where the user can go back and forth and edit (go back) as you are filling up the form)
    - ○ Edit your product
    - ○ Delete your product

More about data modelling -

- Teebay will have the concept of categories. A product can be **under one or more categories**. The categories are -
    - ○ ELECTRONICS

    - ○ FURNITURE
    - ○ HOME APPLIANCES
    - ○ SPORTING GOODS
    - ○ OUTDOOR
    - ○ TOYS

## PART 3: Rent and buy/sell

For the final part, please create the following features -

- List all products created by all users

  **NOTE:** We have written a script to mass add products given a user_id. Please use it to test the app's performance under high data render. You will most likely have to demo this in your coming interview rounds.
- Ability to buy a product. You can assume once you accept buying a product, then the product has been bought.
- Ability to rent a product
- Display all the products bought/sold/borrowed/lent by the user

**Note:** The backend server you'll clone to your machine will send push notifications whenever a product is sold or rented by a user.

Your app must be set up to receive FCM notifications (Android implementation is sufficient) and navigate the user to the product page upon clicking the notification.

The push notification will include the product_id to identify the relevant product. You will need to configure Firebase using the provided JSON files along with firebase configuration instructions for android: Teebay Mobile Assesment Firebase JSON

## PART 4: Implementation documentation [MANDATORY]

Please attach a **Part_4_documentation.md** with your submission and **in brief** provide an explanation of how you solved each part of the problem. Think of this part as a "technical documentation" you are providing to the engineering
team who would want to know about this application. Feel free to discuss some

of the corner cases that are worth documenting and how you went ahead in solving it.

## WIREFRAME DESIGN

A rough wireframe has been provided to you. But you can get creative and create your own designs as long as the functionalities in each of the 3 parts exist. **But the priority is to complete the challenge within the specified time as opposed to better design**.

## TIPS FOR THE CANDIDATE

- The mobile app must be developed in **React Native**.
- A BE Django template is provided with instructions how to run it and the necessary code for the features has already been written. You do not need to work on the backend; instead, you will focus on integrating your mobile application.
    - **Important:** Here's the Firebase admin SDK credentials to complete your BE setup locally.
- Think of this challenge as not just a coding exercise but software you will be deploying to production. So there are various nuances to think about that **are intentionally left open ended**. **These need to be reflected either in code or in documentation**. For example -

    - Testing
    - User experience (which includes input validation and proper user feedback)
    - Component architecture and reusability
    - Routing
    - Database modelling
    - Readability and software best practices
    - Handling practical application corner cases. (For example, what happens when there is a rent time overlap?)
    - Handling error cases
    - Be aware of battery life and app performance when you ingest a lot of data

## SUBMISSION GUIDELINE

- Please **do not** submit your entire submission in one commit. We want to see how you organize and push your commits in your submission (HINT: With meaningful commit messages)
- Finish **as much as possible** in the provided time slot and if you are not able to complete the entire challenge, we would highly encourage you to still submit and we can discuss how you would solve the pending items.
- Please share the challenge via a github link. The challenge needs to run on localhost only. The challenge must have a README that lists out all the steps required to run the project. The challenge can be reviewed in any type of machine. NOTE: You can assume the reviewer will have Docker installed
- Avoid Code duplication like copy-pasting the same thing (multiple same input fields, cards, etc). Focus on code reusability.
- Please have a small video (Loom recording) that demos all parts. The video does not have to be comprehensive, just a quick summary of all the parts. If the file is too big please share a drive link.

## ASSESSMENT EVALUATION POINTS

- **Correctness**: Is your solution complete?
- **Code organization and readability**: Is your code easy to read and maintainable and easily testable?
- **Code design**: Choice of component architecture, data structures and efficiency
- **Framework knowledge:** How well the tools and libraries were used
- **Communication:** How professionally was the challenge written in terms of code comments or any other documentation.

Finally, if you have any questions/concerns or if you think there is any error in any of the parts, please email at: ehsanur.rahman@sazim.io