

## FlatFinder Client Side (React)

### Introduction

Application Name: FlatFinder

### Case Study

The previous project was challenging to maintain due to repetitive functionality and the limitations of JavaScript. With React's ability to divide tasks into components and distribute work among developers efficiently, it was decided to transition to React for easier code maintenance and future scalability. You are tasked with developing the next system in React, in pairs

### Goals

The project is supposed to make the student gain confidence in the following skills:

- Developing client-side in React.
- Consume REST API.
- Work as a part of a team.

### Environments

- Browser.
- Responsive: Desktop, Tablets, and Smartphones.

### Technologies

- UI: HTML, CSS, JavaScript, React
- Data Base Server: Firebase

### Soft Skills

The journey of this project will include practicing and evolving the following skills:

1. Code Review and Peer Review Skills: Practicing the ability to review someone else's code constructively, focusing on identifying areas for improvement while acknowledging strengths.
2. Flag Raising Abilities: Learning to raise flags effectively, addressing concerns about technical issues or project integrity in a clear and concise manner.
3. Communication with Team Leaders (your trainer): Developing skills in communicating with team leaders, including the ability to convey technical complexities in understandable terms and facilitate alignment and decision-making.

4. Diplomacy and Feedback: Cultivating diplomacy in providing feedback during code reviews, fostering a culture of constructive criticism and mutual growth within the team.
5. Transparency and Collaboration: Understanding the importance of transparency and collaboration within the team, ensuring that concerns are addressed openly and contributing to a cohesive project environment.
6. Individual Problem-Solving: Enhancing problem-solving abilities by independently tackling challenges encountered during the project, fostering resilience and creativity in finding solutions.
7. Self-Learning Skills: Emphasizing the importance of continuous self-learning to stay updated with new technologies and methodologies, enabling personal and professional growth in the ever-evolving field of full-stack development.

## Teamwork Flow

A team has two members, and the division of labor is specified in the [division of labor section](#). In the final phase, the team will integrate their work to form a full working web application.

## Data Model

### Entities

#### User

	Data Type
Name	
Email	String
Password	String
First Name	String
Last Name	String
Birth Date	Date

#### Flat

a flat owned by a single user.

	Data Type
Name	
City	String
Street name	String
Street number	Number
Area size	Number
Has AC	Boolean

Name	Data Type
Year built	Number
Rent price	Number
Date available	Date

### Message

a message was sent from a single user in the context of a single flat.

Name	Data Type
Creation Time	Date
Content	String

### Validation

#### User

- All fields are required.
- Data type check.
- Email must be in email format.
- Both names (first and last) must be at least 2 characters long.
- The derived age must be in the range of 18-120.
- Password must be at least 6 characters.
- Password must contain
  - Letters
  - Numbers
  - A character that is neither a letter nor a number.

#### Flat

- All fields are required.
- Data type check.

#### Message

- Message content can't be an empty string.

### Pages

#### New Flat

- Inputs for all flat's properties as specified in the entities section.
- Save button.

If all inputs are valid then the data will be saved. After a successful update, the user will be redirected to his home page.

## Flat View

This page shows all the properties of a given flat that are specified in the entities section.

In case the user is the owner of the flat entry then he will have a button that redirects him to the editing page.

## Messages

If the user doesn't own the flat entry then he can send a message to the owner about the given flat. Someone who doesn't own the flat will see in this page only his own messages. The owner will see all the messages that others sent him about the given flat. For the sake of simplicity, the owner doesn't have the option to send back a message. Every message will contain the following:

1. Creation time stamp.
2. Sender's full name and email.
3. Message content.

## Edit Flat

- Inputs for all flat's properties as specified in the entities section.
- Update button.

If all inputs are valid then the data will be saved. After a successful update, the user will be redirected to his home page.

## Search (Home)

### Body

This page contains a table of all the flats that are stored in the system. Each row in the table will hold the following:

- All of the flat's properties as specified in the entities section.
- Link/Button to open the flat view page.
- Toggle button to mark/unmark as favourite.
- Owner's full name and email.

In addition, the table will have filtering and sorting capabilities.

Filter options fields:

1. City
2. Price Range
3. Area Size Range

Sort categories:

1. City

2. Price
3. Area Size

### Header

- Company Logo
- Greetings: Hello - User Full Name
- The following links/buttons:
  - Logout
  - Home (Search Flats)
  - My Profile
  - My Flats
  - Favourites
  - All Users (shown only to admin)
  - Delete Account

### My Flats

This page allows a user to manage the entries of flats that he publishes in the system. The page contains the following:

- Insert new flat button.
- Table of all the users's flats. Each row will hold the following:
  - All of the flat's properties as specified in the entities section.
  - Delete button.
  - Link/Button to open flat view.
  - Edit button/link.

### Favourites

This page contains a table of all the users's favourite flats. The table structure will be the same as the one in the home page besides a single distinction: instead of a toggle button, each row will have a remove button. After pressing it the flat will be removed from the table and won't be considered to be one of the renter's favourite.

### Login

- Email input
- Password input
- Login button
- Register button

The first page that the user faces after opening the app. After a successful login the user will be redirected to the home page. Another login process will be required after 60 minutes. If the user is not registered yet he can click the "Register" button that will redirect him to the registration page.

## Register

- Inputs for all user's entity properties as specified in the entities section.
- Password Confirmation Input.
- Register button.

If all inputs are valid then the data will be saved. A successful registration will redirect the user to the home page.

## Profile

show all the given user properties specified in the entities section except for the password. Only the same user and the admin will have an edit button for the given user that will redirect to the profile update page.

## Profile Update

- Inputs for all user's entity properties as specified in the entities section.
- Password Confirmation Input.
- Update button.

Regular user can edit only his own profile and admin can edit every user in the system. If all inputs are valid then the data is saved. After a successful update, the user will be redirected to the home page if he edited his own profile. If the admin edited someone else profile then he will be redirected back to the "All Users" page.

## All Users

### View

This page can be viewed only by admin contains a table of all the registered users. Each row in the table will hold the following:

- All of the user's properties as specified in the entities section aside from the password.
- Published flats counter.
- Is Admin.
- Button to open user profile.

In addition, the table will have filtering and sorting capabilities.

Filter options fields:

1. User Type
2. Age Range
3. Flats Counter Range
4. Is Admin

Sort categories:

1. First Name
2. Last Name
3. Flats Counter

### Operations

This page will allow the admin to perform the following operations per user:

- Grant admin permissions.
- Remove user.

### Division Of Labor

Each of the two developers has 5 days total.

#### Developer A

Page	Days
New Flat (Without Messages)	1
View Flat + Edit Flat	1
My Flats	1
Favourites	1
Search (Home)	1

#### Developer B

Page	Days
Login	0.5
Register	1
Profile + Update Profile	1
All Users	1
Flat View Messages	1.5

Good luck!