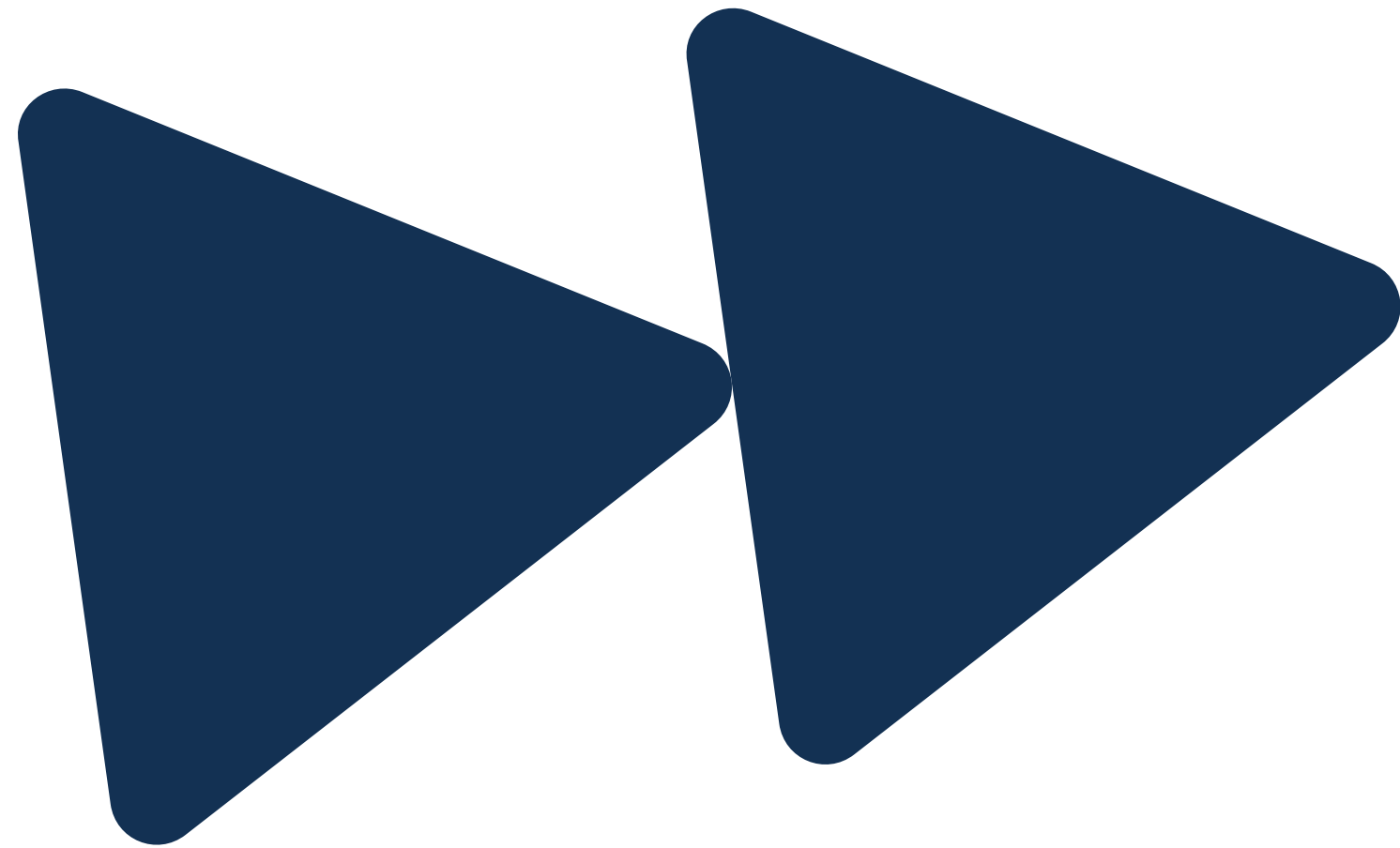


zoio

Driving African Tech Innovation



Property24 App Clone



zaio

Note

Please make sure you have been placed into your team and have good clarity on how group upskilling at Zaio works. Bear in mind, one developer will build the mobile app, one developer for the web app, and one developer for the backend (API + database).

Purpose Of Challenge

This challenge has been designed based on the feature requests Zaio has received in the past from clients. This challenge will be intense. However here are the things you will take away:



Joy of Creation

You will realize the joy of creation and truly enjoy this art.



Teamwork

You will be able to collaborate with others in industry style



Upskilled

Validate that you can code the skills you select.



Build Skills Profile

Unlock Paid Projects to prepare for full-time jobs ranging from 25-35k



Soft Skills

Improved communication, time management, performance under pressure, conflict resolution.

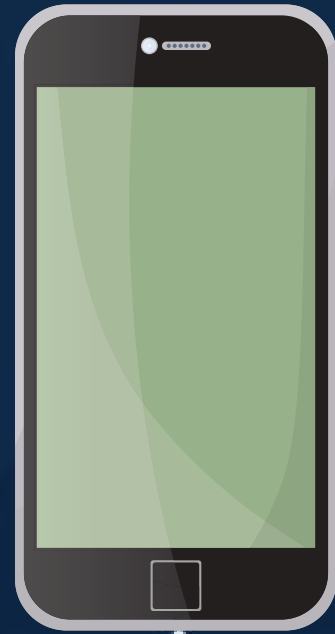


Adaptable

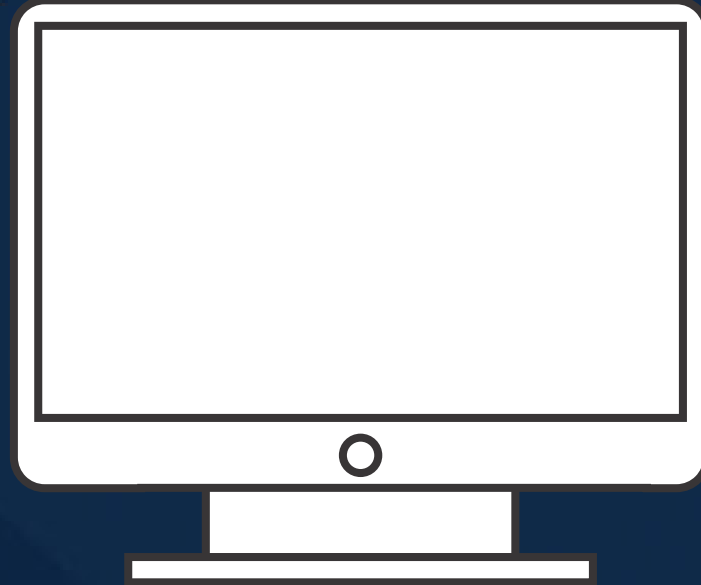
You will prove that you are adaptable and have learned how to learn ;)

How does this work?

Your team will be expected to build the following:



Mobile App



Web App



Backend API



Database



The above is a recommendation using MERN stack (Mongo, Express, React & Node)

However, please feel free to use different stacks such as MEAN (Mongo, Express, Angular & Node)

Feel free to consider using other stacks such as C# .NET, and Python Django if you already have those skills, otherwise please follow the recommended MERN stack.





Property Agent Mobile App



zaio

Agent Mobile App

- In this app, an agent representative of a property such as Rawson Properties will
- be able to create accounts and create properties that can be viewed by potential customers in future.

Do create some sort of state management system eg. Use Redux if coding in React Native. A lot of react tutorials also go through Redux.

- Login & Register
- View created properties
- Edit past created properties
- Create a new property

Login

Feature	Pt
Email/Password inputs	1
Login button submits the email and password to the API	1
Registration button navigates to the Registration page	1
Error alert if the incorrect email and/or password is entered	1
Navigates to the List of Properties page on successful login	1
Stores the JWT token/user ID for access within app	1
Going the extra mile (super important to leverage your profile)	
Attractive UI	1
Splash Screen upon app open	1

Registration

Feature	Pt
Firstname/Lastname/Email/Password inputs	1
Registration button submits the inputs to the API	1
Back button (or equivalent button e.g. a button that says “I already have an account”) navigates to the Login Page	1
Error alert if the email already exists	1
Navigates to the List of Properties page on successful registration (registration does an automatic login)	1
Stores the JWT token/user ID for access within app	1

User Profile

Feature	Pt
Agent logo (with dummy profile pic)/Firstname/Lastname/Email should be visible on the profile page	1
The data should come from the JWT token GET /users API endpoint, managed by a user service class	1
Going the extra mile (super important to leverage your profile)	
Attractive UI	1

List of Properties

Feature	Pt
List of properties created by that agent should render	1
The property service class should use an HTTP GET method to retrieve the data from the API using User ID stored accessible	1
On click of a property item, navigate to the property details page to render that specifically pressed property data	1
Going the extra mile (super important to leverage your profile)	
Attractive UI	1

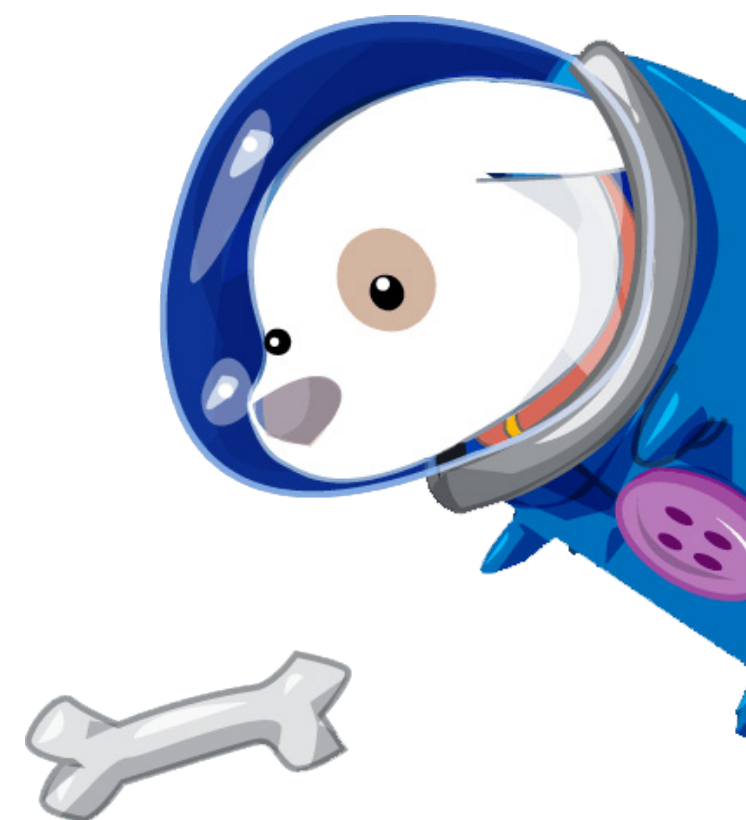
Create Property Page

Feature	Pt
Name (string), Location (string), Price (number) required Inputs (View property24 for other data points and create form to collect data)	1
Create button should POST this data to the API	1
Image Upload for property	1
Going the extra mile (super important to leverage your profile)	
Gallery of images upload	3
Attractive UI	1

Edit Property Page

Feature	Pt
Name (string), Location (string), Price (number) required Inputs (View property24 for other data points and create form to edit data) - these inputs need to be preloaded and user should be to change these values	1
Delete Button should do a DELETE request to the API	1
Upload new picture dynamic loading of changing from old pic to new pic	2
Save changes button to redirect to list of properties page showing updated properties	1
Going the extra mile (super important to leverage your profile)	
Gallery of images upload - add/remove pictures	3
Attractive UI	1

Customer Web App



zaio

Customer Web App

In this app, a normal user such as yourself or I should be able to view properties posted by all agents & view all properties after signing up/logging in. There is a caveat though, a user can only view 4 properties per hour. A user should not be able to click into a card and view details of a profile if viewed 4 properties already, the card should be greyed out and show that its locked. If a user has already opened 4 properties and viewed its details, the user cannot open a 5th one in the same hour. They have to wait for 1 hour from the time they viewed the fourth one. (the algorithm to handle if 4 properties are viewed should be handled in the backend (NOT FRONTEND)).

Do create some sort of state management system eg. Use Redux if coding in React. A lot of react tutorials also go through Redux.

- Login & Register
- View all properties by cards & on a map (side by side -airbnb style)
- Click into a property and view details of a property
- Search properties by name of agent and name of property

Login

Feature	Pt
Email/Password inputs	1
Login button submits the email and password to the API	1
Registration button navigates to the Registration page	1
Error alert if the incorrect email and/or password is entered	1
Navigates to the List of Properties page on successful login	1
Stores the JWT token/user ID for access within app	1
Going the extra mile (super important to leverage your profile)	
Attractive UI	1

Register

Feature	Pt
Firstname/Lastname/Email/Password inputs	1
Registration button submits the inputs to the API	1
Back button (or equivalent button e.g. a button that says "I already have an account") navigates to the Login Page	1
Error alert if the email already exists	1
Navigates to the List of Properties page on successful registration (registration does an automatic login)	1
Stores the JWT token/user ID for access within app	1

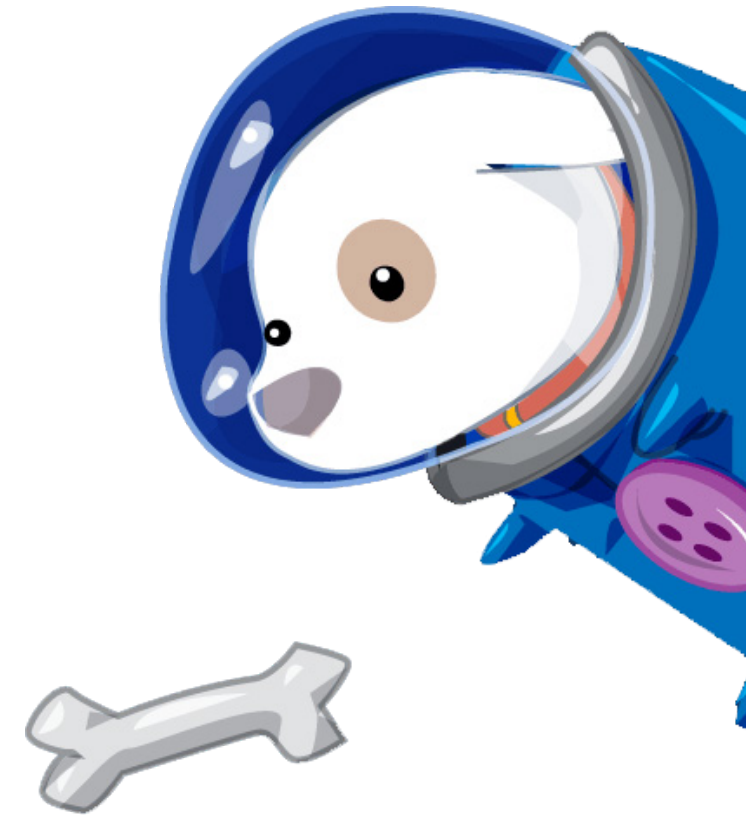
List of Properties page

Feature	Pt
List of all agent properties should render	1
Use HTTP GET method to retrieve the data from the API	1
On click of a property item, navigate to the property details page to render that specifically pressed property data	1
Search by Property Name & Agent Name	3
View properties on Map (like airbnb - using Google maps or your choice of rendering a map with properties) and allow clicking on property icon on map to redirect to details page	3
Going the extra mile (super important to leverage your profile)	
Attractive UI	1

Property Details Page

Feature	Pt
Display at least the following data points of a property: <ul style="list-style-type: none">- name (string)- location (string)- price (number)	1
The details page should expect a URL Query Parameter to identify which property to search for and which property data to display	1
Display Images gallery/image	1
The back button should navigate back to the list of properties	1
Add Book Viewing button that gives an alert saying "Coming Soon."	1
Going the extra mile (super important to leverage your profile)	
Attractive UI	1

Backend API & Database



zaio

Backend API

This API will have all the necessary routes to handle any requests to create, read, update or delete any data from the database. The backend will also be responsible for creating logic for tracking if data can be viewed or not (remember the caveat - only 4 views per hour).

The designed API endpoints can be created at the backend developers discretion, however, the following are fundamentally important from a grading perspective and feel free to add any other api end points as you feel to ensure the app works.

- Handle Authentication for Agents & Customers
- Handle Property Data (CRUD)
- Caveat logic
- Any other API calls to help with front end development (might have to add search functionality).

Users (Customers)

Feature	Pt
POST /customers/authentication - should login the user Body: <ul style="list-style-type: none">- email- password Response: <ul style="list-style-type: none">- JWT Token	1
POST /customers - should create the user Body: <ul style="list-style-type: none">- firstname- <u>lastname</u>- email- password Response: <ul style="list-style-type: none">- JWT Token	1

Users (Agents)

Feature	Pt
POST /agents/authentication - should login the user Body: <ul style="list-style-type: none">- email- password Response: <ul style="list-style-type: none">- JWT Token	1
POST /agents - should create the user Body: <ul style="list-style-type: none">- firstname- <u>lastname</u>- email- password Response: <ul style="list-style-type: none">- JWT Token	1

Properties I

Feature	Pt
<p>POST /properties - should create a new Property</p> <p>Body:</p> <ul style="list-style-type: none">- name (string)- location (string)- imageUrl (string)- price (string) <p>Response:</p> <ul style="list-style-type: none">- id (number)- name (string)- location (string)- imageUrl (string)- price (string)	1

Properties II

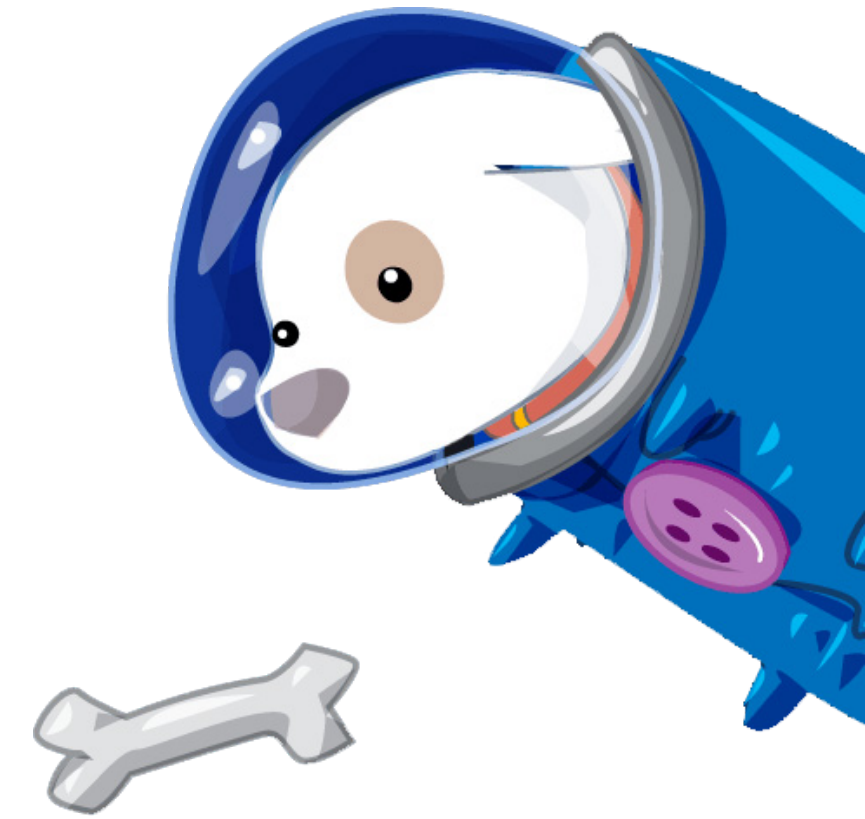
Feature	Pt
<p>UPDATE /properties - should partially update new Property</p> <p>Body:</p> <ul style="list-style-type: none">- id (number)- name (string)- location (string)- imageUrl (string)- price (string) <p>Response:</p> <ul style="list-style-type: none">- id (number)- name (string)- location (string)- imageUrl (string)- price (string)	1

Properties III

Feature	Pt
DELETE /properties/:id - delete a Property by ID	1
GET /properties/:id - find a Property by ID	1
GET /properties - return all properties Here is a twist: <ul style="list-style-type: none">- <i>If a user has already opened 4 properties and viewed its details, the user cannot open a 5th one in the same hour. They have to wait for 1 hour from the time they viewed the fourth one.</i>	4

Any questions, suggestions or appeals to build differently?

Whatsapp Or Call on
+27 (64) 613-7022



Cape Town



Johannesburg

✉ recruitment@zaio.io

zaio



www.zaio.io