**Written Task**

1. Provide screenshots of example Firestore documents to show the data structure of your app, please explain how and why your solution above is secure.

A screenshot of a computer

Description automatically generated

**Secure Authentication**: Ensure that only authenticated users can access their own data through Firestore security rules.

**Data Validation**: Validate the structure and size of likedImages and selectedBreeds to prevent malformed data from being stored.

**Data Access Rules**: Apply strict Firestore rules to ensure that users can only access or modify their own documents, not others'.

1. Provide a written explanation of how you would refactor the solution to include a Node.js Firebase function that connects to the dog.ceo https://dog.ceo/api/breeds/list/all API endpoint and fattens the data into a single array of strings.
   1. What type of things you would think about in this task and why.
   2. You don’t need to provide the code/or completed task, we are looking to understand if you have a low/med/adv level of understanding working with NodeJS.

Steps to Implement the Firebase Cloud Function:

1. **Set Up Firebase Cloud Function**:

Install and configure Firebase Functions in your Firebase project.

1. **Make a GET Request to the Dog CEO API**:

The API at https://dog.ceo/api/breeds/list/all returns a nested object where the breed names are keys and their sub-breeds are values. The goal is to flatten this object into a single array of strings (including both breeds and sub-breeds).

1. **Flatten the Data**:

Process the response data to extract the breeds and sub-breeds and create a unified array of breed strings. If a breed has sub-breeds, each sub-breed should be concatenated with its parent breed

1. **Return the Flattened Array**:

The Firebase Cloud Function will then return the resulting array to the caller

1. Write an explanation of your unit test approach, including an example of one unit test you would write, what parts of the code you would test, and why.

I want to test if the Viewfeed component renders a list of breeds with their images, we can write a unit test that simulates the fetching of breed data and checks if the rendered elements correspond to the expected breeds.