**Home Assignment: Cat Collector Application**

**Objective**

Develop a web-based Cat Collector application that interacts with [TheCatAPI](https://thecatapi.com/" \t "_blank) to retrieve information about 100 random cats, including their images, and allows users to perform create, read, update, and delete (CRUD) operations on this data within a local application. This assignment tests knowledge of JavaScript, Python, and PostgreSQL.

**Requirements**

**Front-end**

* Implement the front-end using JavaScript and any modern framework/library of your choice (React, Vue.js, etc.), or vanilla JavaScript with HTML/CSS for simplicity.
* The front-end should provide a user-friendly interface that displays cat images and information in a visually appealing layout.
* Include functionality to add a cat to favorites, remove a cat from favorites, and update cat information (e.g., name, description).
* Implement pagination or infinite scroll to manage the display of cat images and data efficiently.
* Implement the ability for the user to specify a cat breed, and only retrieve cats of that breed

**Back-end**

* Develop the back-end service using Python with a framework like Flask or Django to handle HTTP requests from the front-end.
* Use Flask or Django to create RESTful API endpoints that enable CRUD operations on the cat data. These endpoints should include:
  + GET /cats to retrieve all cat data.
  + POST /cats to add a new cat to favorites.
  + GET /cats/:id to retrieve a specific cat's details.
  + PUT /cats/:id to update a specific cat's details.
  + DELETE /cats/:id to remove a cat from favorites.
* The back-end should also include an endpoint or function to fetch 100 random cats from TheCatAPI and store their data in the PostgreSQL database upon initial application setup.

**Database**

* Use PostgreSQL to store information about cats, including their API data and any additional information users may add through the application (e.g., favorite status, custom names, descriptions).
* Design the database schema with at least one table for storing cat data. This table should include columns for storing data relevant to the application's functionality (e.g., API ID, image URL, name, description).
* Provide SQL scripts for creating the database schema and initializing any required data.

**Integration with TheCatAPI**

* Register for an API key with TheCatAPI.
* Use the API key to authenticate requests to TheCatAPI for fetching cat data.
* Implement logic to retrieve 100 random cats from TheCatAPI, including their images and relevant information, and store this data in the PostgreSQL database.

**Submission Guidelines**

* Submit the source code for both the front-end and back-end components of the application, including any necessary configuration files and environment setup instructions.
* Include a [README.md](http://readme.md/) file with detailed instructions on how to set up and run the application, including steps for setting up the PostgreSQL database, and starting the front-end and back-end servers.
* Document the API endpoints and provide examples of requests and responses.
* Bonus: Include unit tests and/or integration tests demonstrating the reliability and functionality of the application.
* The entire source code, documentation, configuration should be submitted in Github or in Gitlab.

**Evaluation Criteria**

* Functionality: The application must work as described, fulfilling all the specified requirements.
* Code Quality: The code should be clean, well-organized, and follow best practices for JavaScript and Python development.
* Database Design: Effective use of PostgreSQL to store and manage cat data.
* User Interface: The front-end should be intuitive and visually appealing.
* Documentation: Clear, concise documentation on how to set up and use the application.
* Creativity and Problem-Solving: Innovative solutions and effective problem-solving demonstrated throughout the application's development.

This assignment offers you the opportunity to showcase your skills in full-stack development, API integration, and database management in a fun and engaging way. It assesses your ability to create a complete web application from start to finish, highlighting your technical abilities and attention to detail.