API Contract for Pet Adoption App

GET /pets

- Description: Return the list of all pet data.
- Input: None
- Output:
  - Success: 200 OK
    - Returns list of all pets in json format

GET /pets/<int:pet_id>

- Description: Returns the data of one pet based on the pet_id
- Input:
  - pet_id (integer): In the query, the pet_id will be specified
    - Example: /pets/1
- Output:
  - Success: 200 OK
    - Returns information about the pet specific to the given pet_id.
  - Error: 404 Not Found
    - When there is no pet with the given pet_id, the error: "Pet not found" is returned.

User:

1. POST /login

- **Description**: Authenticates a user with a username and password.
- **Input**:
  - username (string): Username attempting to log in.
  - password (string): Associated password.
- **Output**:
  - **Success**: 200 OK
    - Returns JSON object: {"username": "user", "message": "Login successful", "token": "JWT-token"}.
      - **Token**: A JSON Web Token (JWT) for session validation.
  - **Error**: 401 Unauthorized
    - Returns JSON message: {"error": "Invalid username or password"}.

2. POST /register

- **Description**: Registers a new user with a unique username, password, and optional profile data.
- **Input**:
  - username (string): New username.
  - password (string): Password for the user.
  - **Optional**:
    - email (string): User's email address.
    - phone (string): User's phone number.
    - preferences (list): User's pet preferences (e.g., species, age, size).
- **Output**:
  - **Success**: 201 Created
    - Returns JSON message: {"message": "User registered successfully", "user_id": "unique_id"}.
      - **User ID**: A unique identifier for the registered user.
  - **Error**: 400 Bad Request
    - Returns JSON message if username is taken: {"error": "Username already exists"}.
  - **Error**: 400 Bad Request
    - Returns JSON message if required fields are missing: {"error": "Missing required fields"}.

Management

1. GET   /applications
   a. Input: none
   b. Output: all the applications including user's information and pet information(json)
2. GET   /applications/<int:app_id>
   a. Input: app_id(int)
   b. Output: the information associated with application id (json)
3. POST   /manage_login
   a. Input:
      Manager_email(string)
      Manager_password(string)
   b. Output:
      Message: login successful(json)
      Message: invalid credentials(json)