Gotta Catch ‘em All

Task:
This project explores your abilities in full-stack development and cloud deployment.
The expected time to complete the task is 24 hours

Your assignment is to create a React frontend and Django backend web application that:

1. Has user authentication (login/sign-up) mechanisms
2. Allows users to view the current portfolio of pokemon they own and their stats
3. Allows users to add newly captured pokemon to their portfolio, and remove pokemon from their portfolio
4. Allows users to view the names of pokemons they do not own yet (only names, not the stats)

The Dataset provided will contain a number of Pokemons. You may assume this dataset represents the full Pokemon universe.

Dataset:
Download pokemon.csv from the google drive link provided.
This dataset contains pokemon instances that contain the following information respectively

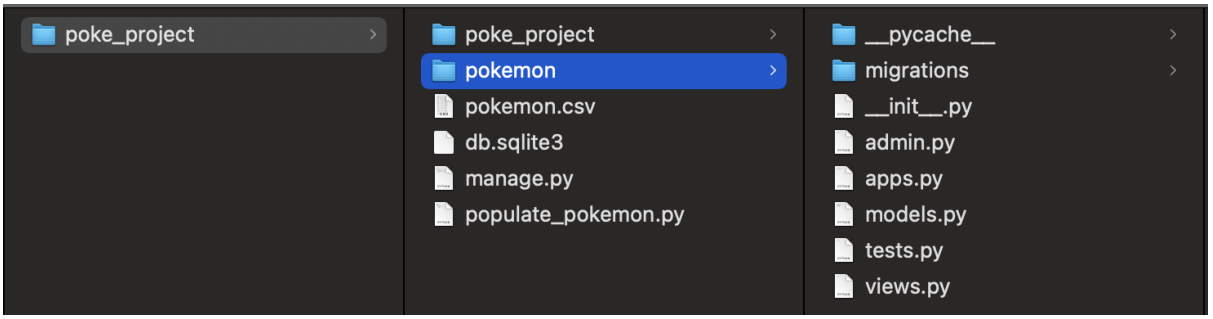
1. name (string)
2. hp (int)
3. attack (int)
4. defense (int)
5. type (string)

Getting Started:

Start a new django project, ‘poke_project’

Create the ‘pokemon’ app, and import the data in the csv into your db.sqlite3 database

Your project should have a file structure similar to the one below:



User Authentication

- Using Djoser and JWTAuthentication, allow authentication of users, where each user has to login with a username and password
- Forget/ Reset password function is not required

Core functionality

- Create a “Catch Pokemon” tab, accessible via the Navbar
- Show a randomly generated pokemon from the dataset(from the list of all pokemons), available for capture
- To capture, we implement a simple “guess the number” game (see https://www.funbrain.com/games/guess-the-number)
- You may decide the difficulty of the game
- If the user correctly guesses the number within 3 tries, the pokemon is captured, and added to the user’s portfolio
- If the user is not able to guess the number within 3 tries, the pokemon is returned to the wild, the page is refreshed, and a new game instance is generated (with new randomly generated pokemon)

Additional Requirements

- Each pokemon may only be owned by a single user and identified by its unique ID
- Each owner may own any number of pokemon
- Each pokemon has a (int) level 1-100 that is randomly generated upon capture

Create a simple front-end that integrates with your back-end database of Users and Pokemon using APIs from Django Rest Framework.

API endpoints required:

1. pokemon/unownedpokemon/ - a GET request here should return a serialised list of all the pokemon that the user does not currently own
2. pokemon/mypokemon/ - a GET request here should return a serialised list of the pokemon owned by the user
3. pokemon/allpokemon/ - a GET request here should return a serialised list of all pokemon in the dataset
4. pokemon/addpokemon/ - a POST request here should add a pokemon to the user’s collection
5. pokemon/releasepokemon/ - a POST request here should allow a user to discard one of his pokemons in his collection

Front End

- Create at least three pages, one for the user to log in, another for the user to view the pokemon in their collection, and another one for catching pokemons

Deployment

For deployment of your project, please use the following deployment architectures:

- Front end -> AWS EC2 (T2.micro)
- Back end -> AWS EC2 (T2.micro)
- Database -> AWS Relational Database Service (PostgreSQL)

Help/resources

React basics: https://www.youtube.com/watch?v=Ke90Tje7VS0
Django basics: https://www.youtube.com/watch?v=rHux0gMZ3Eg
Deployment basics: https://www.youtube.com/watch?v=-psX4Kr5B-U