

Team TreeHouse3

Team Lead: Prabhleen Bagri

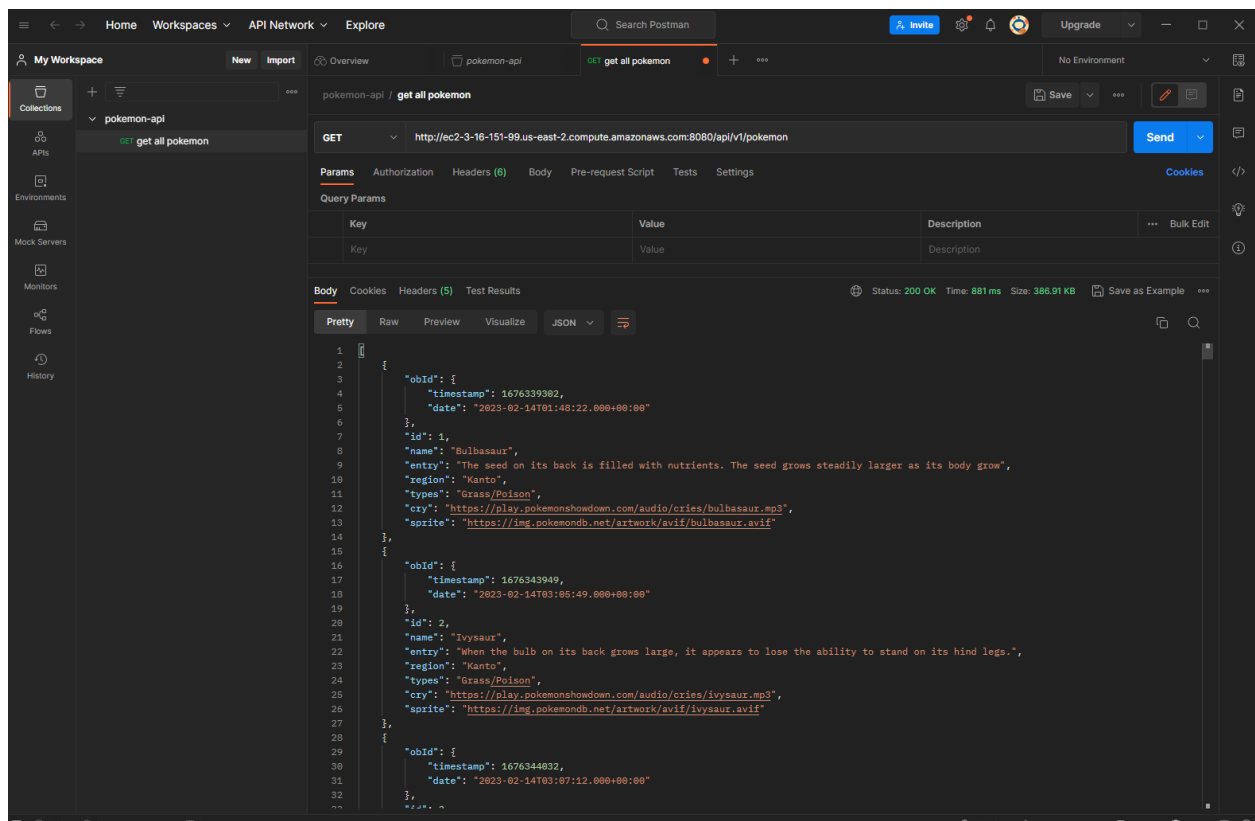
Members: Abhinav Selvaraj, Ken Ho, Breanna Chi

Who's That Pokemon Successful Test Completion

Backend

Our backend is built using Java and the Spring framework. The backend requests data from our database and returns information. We have multiple possible queries. To test that our backend is functioning, we utilized Postman. Below are the results of our tests. A Pokemon's Pokedex number is their unique ID number.

Function 1: Return all Pokemon Documents in the DB



Function 2: Return a Pokemon based on their Pokedex number

The image shows the Postman API client interface. On the left, the 'My Workspace' sidebar displays a collection named 'pokemon-api' containing a single request 'GET get all pokemon'. The main panel shows the details of this request:

- Method:** GET
- URL:** `http://ec2-3-16-151-99.us-east-2.compute.amazonaws.com:8080/api/v1/pokemon/326`
- Params:** None
- Headers:** 6 headers are listed, including 'Content-Type: application/json'.
- Body:** The response is displayed in the 'Pretty' tab as a JSON object for the Pokemon 'Grumpig'.

The JSON response body is as follows:

```
{
  "obId": {
    "timestamp": 1681368350,
    "date": "2023-04-13T04:32:30.000+00:00"
  },
  "id": 326,
  "name": "Grumpig",
  "entry": "It can perform odd dance steps to influence foes. Its style of dancing became hugely popular overseas.",
  "region": "Hoenn",
  "types": "Psychic",
  "cry": "https://play.pokemonshowdown.com/audio/cries/grumpig.mp3",
  "sprite": "https://img.pokemondb.net/artwork/avif/grumpig.avif"
}
```

The status bar at the top right indicates a successful response: Status: 200 OK, Time: 178 ms, Size: 538 B.

Function 3: Return a Pokemon based on a random Pokedex number

The screenshot shows the Postman interface with a GET request to the endpoint `http://ec2-3-16-151-99.us-east-2.compute.amazonaws.com:8080/api/v1/pokemon/random`. The request is part of a collection named "pokemon-api" with the name "get all pokemon". The response is a JSON object for the Pokemon Nidoran-f.

Request Details:

- Method: GET
- URL: `http://ec2-3-16-151-99.us-east-2.compute.amazonaws.com:8080/api/v1/pokemon/random`
- Headers (6): Key, Value, Description
- Body: Params, Authorization, Headers (6), Body, Pre-request Script, Tests, Settings

Response Details:

- Status: 200 OK
- Time: 96 ms
- Size: 594 B
- Save as Example

Response Body (JSON):

```
1 {
2   "obId": {
3     "timestamp": 1677649867,
4     "date": "2023-03-01T06:21:27.000+00:00"
5   },
6   "id": 29,
7   "name": "Nidoran(female)",
8   "entry": "Females are more sensitive to smells than males. While foraging, they'll use their whiskers to check wind direction and stay downwind of predators.",
9   "region": "Kanto",
10  "types": "Poison",
11  "cry": "https://play.pokemonshowdown.com/audio/cries/nidoranf.mp3",
12  "sprite": "https://img.pokemondb.net/artwork/large/nidoran-f.jpg"
13 }
```

Frontend

Our frontend is built using create-react-app with heavy use of MUI v5 UI libraries. Utilizing their modern styling syntax, we were able to quickly get a responsive layout up and running. As per the product_design_spec test plan, we have passed all the outlined tests. We will add more tests as time progresses. Here are the checklists:

- ☒ SinglePlayer - Complete a gameplay loop as a single player
- ☒ Home Page - able to load homepage
- ☒ Start Game Page - Gameplay starts once this is clicked
- ☒ Game Over page - Complete gameplay as a single player, and observe a players score
- ☒ Scoring System - Complete a gameplay loop in a certain manner so that the observed score is expected to match the score we calculate. Will use various hints in various rounds to include variation
- ☒ Gameplay page - Test to see if we can play a random pokemon's audio, present 4 random options with one correct one, and present 4 hints for the pokemon (information retrieved from the database). Testing will also include if we can hide the information from the players until they ask for the hint.

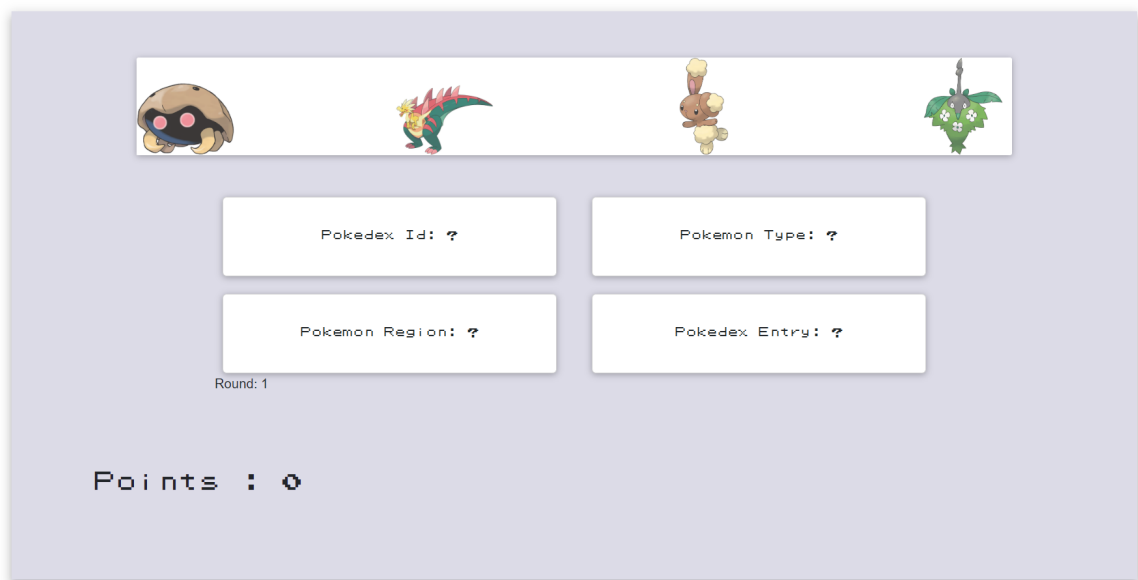
Home Page: This is the first page users encounter. They can select two different game modes



Hint/Start Game Page: A loading page to give the players some instructions on how to play



Gameplay page: This is where most players spend their time



Gameover page: After the round ends, their scores will be displayed here and users will be redirected to the homepage after 5 seconds.

Game Over!

Total points:

Great score! Another game?

[Home](#)[Play Again](#)

Multiplayer: After prompting the user for the join code and username, they will be added to the realtime database for multiplayer.

https://pokemon-f40f8-default-rtdb.firebaseio.com

https://pokemon-f40f8-default-rtdb.firebaseio.com/

```
0: "Test"
└─ example
  └─ game
    └─ -NUYqco13tIo04wf7BeF
      ├── gameCode: "123Test"
      └─ player: "puli"
```

Join Code: 123Test

Username: Join Code: