

Python Promethazine

Justin Chen: HTML/Bootstrap

Alex Olteanu: Database/Python -- Setting up the pokemon/pages

Alex Thompson: API/Python -- Getting data from the API

David Wang: PM, works on whatever is needed, JS

Front End Framework: Bootstrap

Topic: Pokemon Showdown

Description (for now):

- A game that's similar to Showdown. Users can build their own team or use a preexisting one and fight the computer in Pokemon battles. We will build a bare version and add features as necessary. People will have to unlock Pokemon by collecting their cards. Wins will award loot boxes that contain cards.
- Limited to Gen 1 Pokemon
- No items/abilities

APIs:

- Pokemon API (For graphics and stats)
- Pokemon TCG (For pokemon cards and chances)

Front End:

1. base.html
 - a. Template page
2. landing.html
 - a. First page user sees when they enter the site
 - b. Options for log-in and register
3. login.html
4. register.html
 - a. Only if the user needs to register
5. home.html
 - a. User is brought to this page after the user logs in or registers
 - b. Options to battle or build teams
 - c. Displays all the cards/pokemon that you currently own
6. build.html
 - a. Team creator page
 - b. If a team is to be created, players must add 6 pokemon at a time
 - c. For each pokemon, they must pick 4 moves that the pokemon can learn
 - d. Go back to home button when done creating teams
7. battle.html (uses JS)
 - a. Local 2 player battle, players will alternate turns
 - b. Uses saved teams or randomly generates one depending on user selection (per player)
 - c. Each turn, the player has an option to select a move unless conditions do not allow it (ex if a pokemon is charging a move, it cannot attack)
 - d. First player to have their entire party knocked out loses
 - e. Go back to home button when battle completes

Back End:

1. Database

a. Table: credentials

Username (Text)	Password (Text)
example	example
username	password

b. Table: teams

- i. 1 table per user, named after username
- ii. Every consecutive 6 pokemon is 1 team

Pokemon (Text)	Move 1 (Text)	Move 2 (Text)	Move 3 (Text)	Move 4 (Text)
bulbasaur	tackle	growl	vine whip	growth

2. app.py

a. /

- i. renders home.html if logged in, else landing.html

b. /login

- i. renders login.html
- ii. redirect to /

c. /register

- i. renders register.html
- ii. redirect to /

d. /build

- i. renders build.html

e. /battle

- i. renders battle.html

f. /logout

- i. redirects to /
- ii. removes user from session

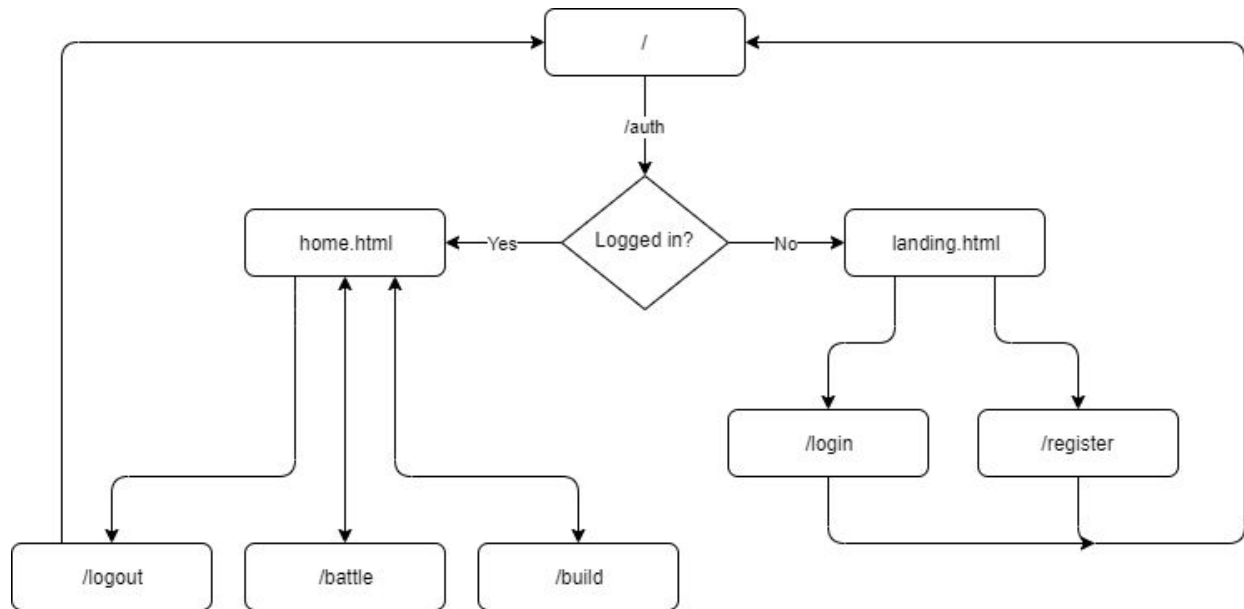
- g. /auth
 - i. checks if user is in session, unviewable to user

3. Functions

- a. login()
 - i. param: username
 - ii. param: password
- b. register()
 - i. param: username
 - ii. param: password
 - iii. All usernames must be unique
- c. login()
 - i. Renders home.html
- d. auth()
 - i. param: username
 - ii. Checks if user is in session
- e. battle()
 - i. param: team1
 - ii. param: team2
 - iii. Renders battle.html
- f. build()
 - i. Param: username
 - ii. Renders team.html
 - iii. Add 1 team comprised of 6 Pokemon with 4 moves each to the user's username database
- g. logout()
 - i. Remove user from session
 - ii. Render landing.html
- h. attack(player, target, move)

- i. Attacks with specific move stats and calculates the result using the pokeAPI

Site Map:



Component Map:

