Hero Wars by Teem_sqrt4469
P01: ArRESTed Development
Yuhang Pan (PM), Matthew Ciu, Michelle Chen, Thomas Mackey
2025-12-02
TARGET SHIP DATE: 2025-12-19

Project Idea: A pokemon showdown type game but with superheroes vs. historical figures instead of pokemon. We will try to fetch real data from multiple API's and convert those info into in-game stats.

Program Components:

A. Flask App (Python)
    - **__init__.py** creates flask app, make routes, and run app
        - **/register** adds entered username & password into users table, checks that username has to be unique, stores username in session then redirects to /home
        - **/login** checks if the entered username & password is stored in users table, if so, stores username in sessions and redirects to /home
        - **/home** extracts the user's info (name, win/loss ratio, match history)
        - **/menu**
            - Makes API calls to fetch characters (superheroes and historical figures)
            - Randomly selects a list of characters to battle (maybe an option to reselect)
        - **/game**
            - Pulls data from APIs and convert certain stats to in-game stats
            - Stores match data in flask session
            - Allow players to interact with battle (attacks, etc.)
            - Can use evilinsults API to taunt the other player
        - **/gameover**
            - Displays winner
            - Optionally stores win/loss, characters used, time of match, etc. in user table
    - **build_db.py** creates the user table
    - **apis.py** handles all API calls & fetch and convert data
    - **game.py** handles game states (running/game over) and turn based logic, converts API stats to in-game stats.

B. Templates (HTML templates for each page w/ Jinja2)
    - **login.html** & **register.html** (form input boxes, handles authentication)
    - **home.html** (homepage showing profile info: name, profile picture, win/loss ratio)

- **menu.html** (pregame phase where player with buttons to randomly select characters and cards to display characters; game title with a start button)
- **game.html** (page that displays battle in real-time)
- **gameover.html** (page that shows that the game is over and displays the winner of the showdown)

C. Database (SQLite3)

- User table (stores user data when they register and updates it when they change profile picture or finish a game)

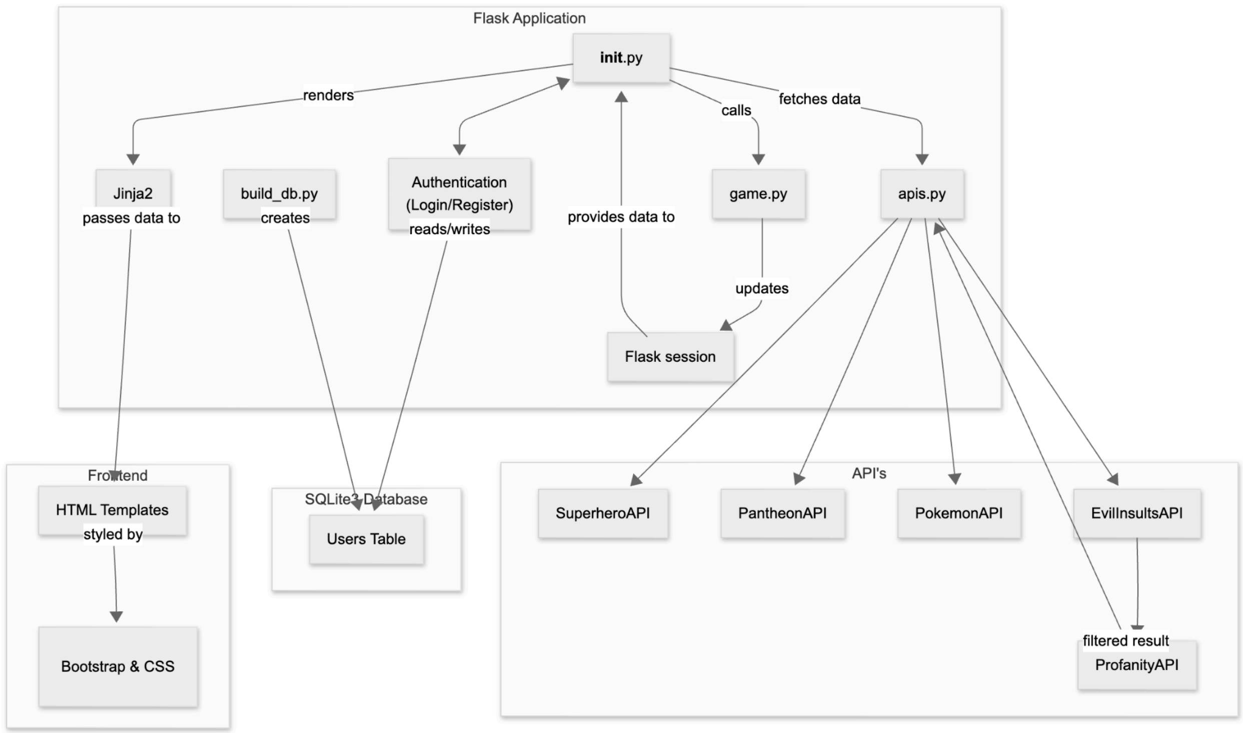
D. API's

- Superhero API - https://www.superheroapi.com/
- Pantheon (historical figures) API - https://pantheon.world/data/api
- Pokemon API - https://emiliebarnard.github.io/pokemon-in-python/#moves
- EvilInsults API - https://evilinsult.com/generate_insult.php?lang=en&type=json
- PurgoMalum API - https://www.purgomalum.com/ to filter EvilInsultsAPI

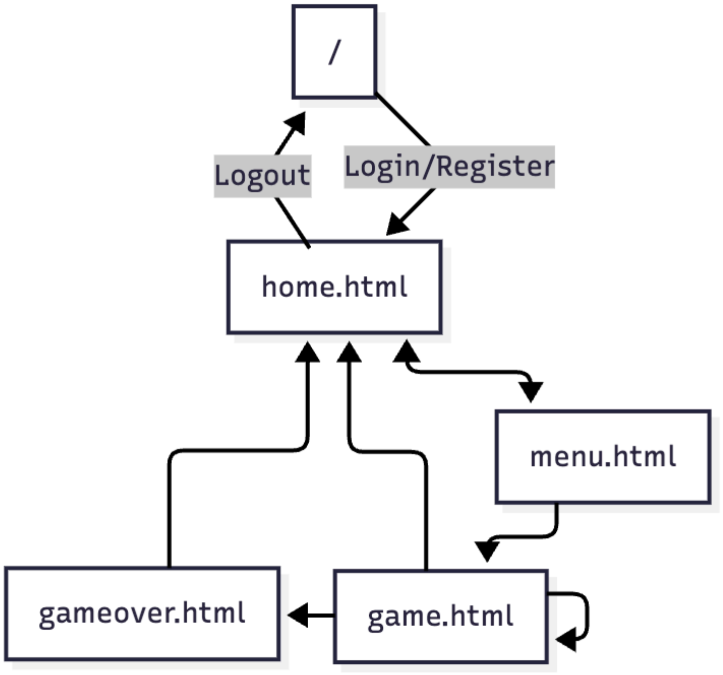
Front-end Framework (Bootstrap):

- Good documentation since the documentation for Foundation isn't great; Tailwind is confusing with all the utility classes in the HTML
- Bootstrap grid system is easy to work with which we can use to display the abilities of our characters; can also be used to divide the battle page to left and right for the two players
- Card container to display randomly selected characters

Component Map:


Flask Application
init.py
renders
fetches data
calls
Jinja2
passes data to
build_db.py
creates
Authentication (Login/Register)
reads/writes
provides data to
game.py
updates
apis.py
Flask session
Frontend
HTML Templates
styled by
Bootstrap & CSS
SQLite3 Database
Users Table
API's
SuperheroAPI
PantheonAPI
PokemonAPI
EvilInsultsAPI
filtered result
ProfanityAPI

Site Map:


/
Logout
Login/Register
home.html
menu.html
gameover.html
game.html

Database Organization:

USERS

TEXT	name	PK	Username is unique
TEXT	password		Used for authentication
INTEGER	wins		# of wins
INTEGER	losses		# of losses
TEXT	profile_pic		Customizable profile picture

Tasks:

- Yuhang Pan
    - game.py (game logic with turn-based gameplay, convert API info to in-game stats)
- Matthew Ciu
    - apis.py (all the apis: SuperheroAPI, PantheonAPI, PokemonAPI, EvilInsultsAPI & ProfanityAPI; get API keys, call API and fetch data)
- Michelle Chen
    - HTML Templates (menu, game, & gameover page)
- Thomas Mackey
    - Bootstrap, __init__.py & build_db.py