SoftDev P0I: ArRESTed Development
TNPG: Gluten Free Pizza
Roster: Michelle Zhu, Tahmim Hassan, Anastasia Lee, Ivan Gontchar
TARGET SHIP DATE: 2024-12-17

DESIGN DOCUMENT

I. Description

This project is a country guessing game where users must identify a country based on progressively more revealing hints. The website uses a database to store user information, APIs to generate the hints, and Bootstrap to provide a clean frontend view.

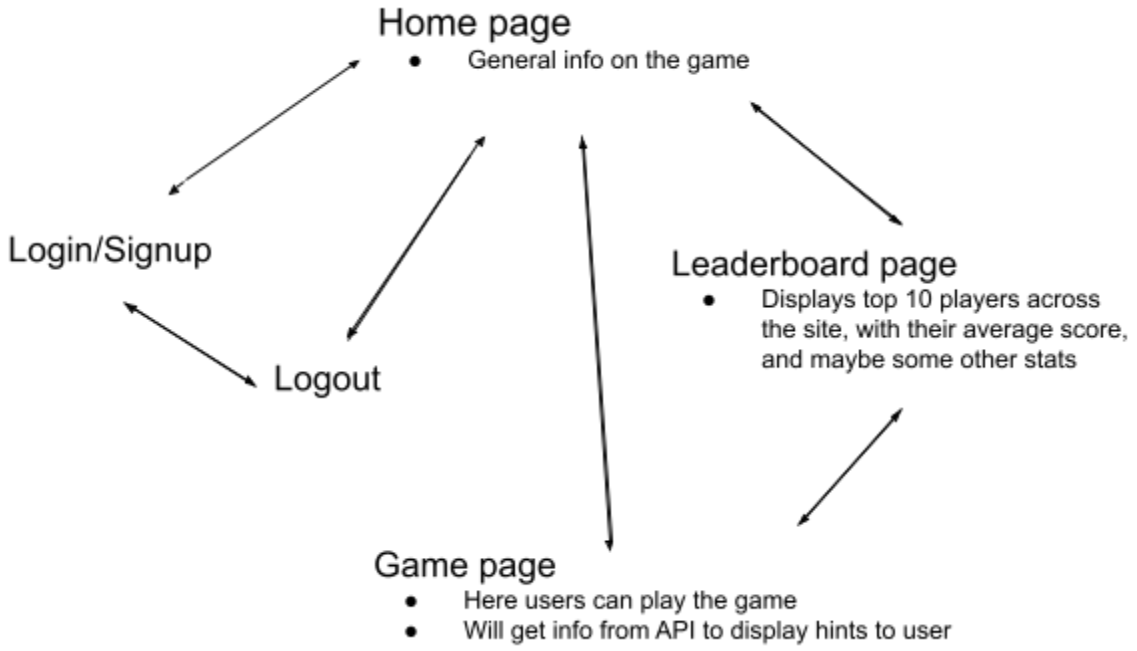
A. Program Components

- Login/register/logout:
  - First-time users must register
  - Users stay logged in until they log out (Flask session)
  - Users can play without an account, but their results are not stored unless they are logged in
  - Account information stored in database
- Game:
  - A country is randomly chosen from a list of all possible countries (generated by API call), and the first hint is automatically given
  - Hints are generated by API calls
  - After each incorrect guess, a more helpful hint is given
    - A more “helpful” hint is one with generally more detail and less ambiguity. For instance, first giving weather data on the country, then its population, then the flag, and other such identifying information given progressively.
  - If all hints (weather, population, flag, subregion, currency, capital) are given and the user has not correctly guessed the country, they lose and the answer is revealed
- Leaderboard:
  - Displays players with the lowest average number of guesses per country
- HTML/CSS/FEF:
  - Neatly display all information
  - Provide coherent structure

B. Component Map

```
Frontend (HTML/CSS/Bootstrap)
   |
   v
Backend (Flask) ──> login/register/logout
   |
   +-- SQLite Database
   |      - User table
   |      - Guesses table ──> leaderboard
   |
   +-- APIs ──> Game
          - OpenWeatherMap (Fetches temperature)
          - REST Countries (Provides country details)
          - FlagsAPI (Fetches country flags)
```

C. Site Map



D. Database Organization

- User table
  - Username (Primary Key) - must be unique
  - Password

username	password
user1	*********
user2	**********

- Guesses table
  - Username (Primary Key)
  - Total number of guesses
  - Number of countries completed
  - Average number of guesses per country
  - Current Country
    - This stores the current country the player is guessing, so they can leave the game page, and return to the same country.
  - Current Game Hints
    - This stores the current number of hints a player has used on their current game, so they can leave the game page and return to the same stage in their game.

username	g_total	c_num	g_avg	c_curr	hint_num
user1	57	16	3.5625	Mongolia	3
user2	12	5	2.4	Grenada	1

E. API Documentation

- OpenWeatherMap API
  - https://openweathermap.org/api
  - Usage: Fetch the current temperature and weather data for the location to be used as hints.
- REST Countries API
  - https://restcountries.com/
  - Usage: Provide a list of independent countries as well as country details like the capital city, population, and currency to be used as hints.

- FlagsAPI
    - https://flagsapi.com/
    - Usage: Retrieve the country flag as a visual hint.

F. Front-end Framework: Bootstrap

Why

Bootstrap is versatile and easy to use and comes with many predefined elements. Its designs automatically adjust to different screen sizes and are highly customizable, and do not depend on CSS and JavaScript as much as the other FEFs do.

How

We plan to use the following Bootstrap elements:

- Top navbar to navigate the site
- Forms for login, registration, and submitting guesses for countries
- List groups to display hints
- Table to display leaderboard

G. Task Assignments

1. Frontend Development - Anastasia
    a. Design responsive HTML pages with Bootstrap styling.
    b. Implement display of game hints and leaderboard.
2. Backend Development - Ivan
    a. Set up Flask routes for handling game logic and interactions.
    b. Implement API calls for OpenWeatherMap API, REST Countries API, and FlagsAPI.
3. Database Setup - Michelle
    a. Use SQLite to design and populate tables for locations and game history.
4. Integration - Tahmim
    a. Connect frontend with Flask backend.
    b. Integrate SQLite database with Flask for storing and retrieving data.
5. Testing - Everyone!