Team name: Jeopardy
Members: Bermet Kalmakova, Tiffany Chen, David Doktorman
Title: Trivia Game

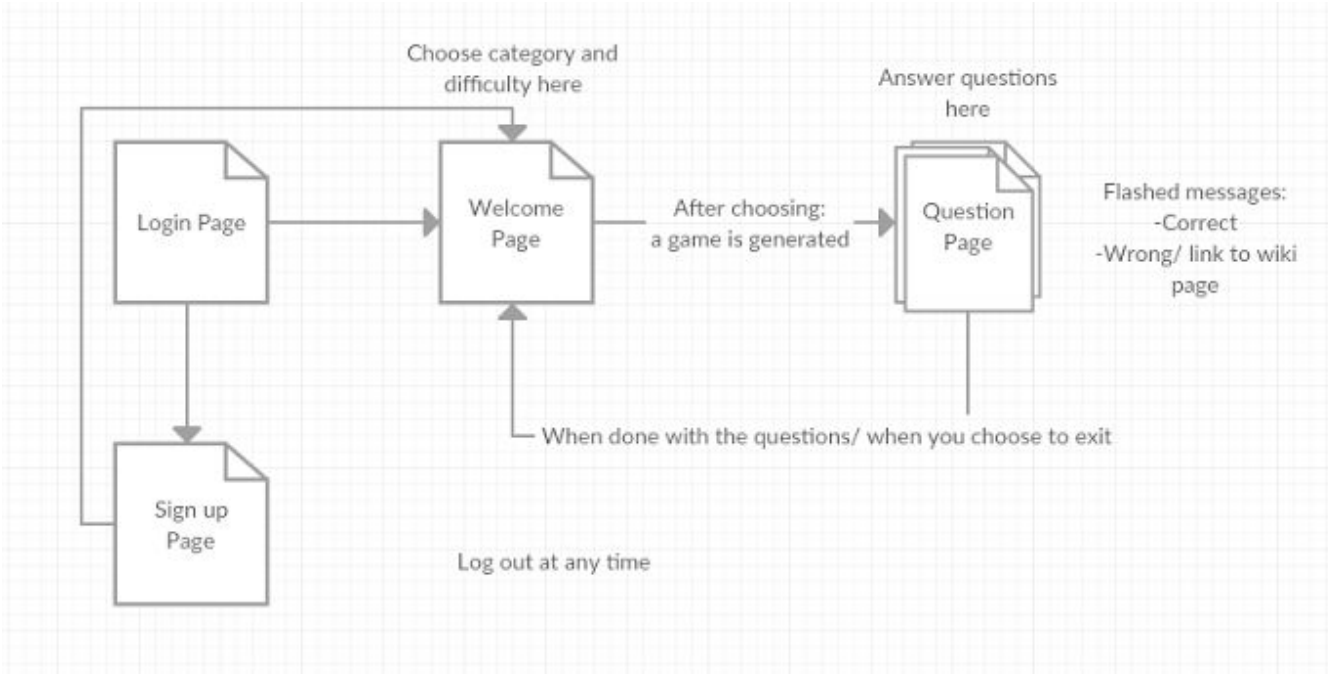
Description
This website is a trivia game that provides information about the topic of the question when answered incorrectly.

Components

- app.py
    - Flask app
    - Connections to both APIs
    - Functions:
        - Get functions (getUsername(), getScore(), getPlacement(), getPassword())
        - createTriviaGame(category, difficulty): calls Open Trivia API to retrieve questions
        - question(): returns question for question.html
        - findWikipedia(question): finds a Wikipedia page that has the answer to the trivia question
        - changePlacement(user): changes the placement of the user after every question answered correctly
        - changeScore(user, score): changes the score of the user after every question answered correctly
        - checkLogin(username, password): returns true if username and password match, returns false
        - createUser(username, password)
        - checkUsername(username): returns true if username doesn't exist (used in conjunction with checkLogin() and createUser())
- base.html
    - Basic template for html files
- welcome.html
    - Welcome page
    - Has a form with categories that user can choose from (or random!) and the difficulty of the questions given
    - Shows username, user ranking, and score
- style.css
    - Our additional styling for the website
- question.html
    - Page where the user is asked a question and can input the answer
    - Will receive the question from the Open Trivia API
    - If user answers the question correctly, this page will be rendered again with a flashed message that the user got it correct with a different question to answer

    - If user answers the question incorrectly, this page will be rendered again with a flashed message including a link to a Wikipedia article on the subject (with help from the MediaWiki API)
    - Has a back button to return to the welcome page at any time
    - Shows username, user ranking, and score
- login.html
    - Allows the user to log in to play the game
- leaderboard.html
    - Ranks users in a leaderboard
    - Shows user ranking and score; after that is large table with everyone's ranking and score
- database.db
    - Holds information about the user (username, password, placement on leaderboard, current score)

Site Map



Database Schema

- database.db has one table: peeps
    - peeps has username, password, score, and placement

Example:

username	password	score	placement
alice	aliceiscool	800	1
bob	bobthebuilder	700	2

…	…	…	…

APIs Used

- MediaWiki
    - Provides access to wikipedia pages and information
    - https://en.wikipedia.org/w/api.php
- Open Trivia
    - Provides thousands of different trivia questions depending on category and difficulty
    - https://opentdb.com/api_config.php

Stages of Development

1. Set up app.py
2. Set up Welcome page
    a. Dropdown choice for different categories and difficulties
3. Connections to APIs
    a. Obtain categories and difficulties from API
4. Set up Question page
    a. Obtain trivia questions from API
    b. Randomize order of choices
    c. Flash a message if answer is correct
    d. Display new question
5. Set up Answer page
    a. Link to wikipedia page about topic from API
6. Set up css framework/design stuff
    a. Bootstrap + separate .css file
7. If there's time: Login system and scoreboard
    a. Database work

Tasks

Bermet Kalmakova: Project Manager; css
Tiffany Chen: html; databases
David Doktorman: APIs