Cure My Boredom by Squash
P01: ArRESTed Development
Jason Chan (PM), Aoanul Hoque, Isabel Zheng, Maya Berchin
2025-12-02
TARGET SHIP DATE: 2025-12-23

Program Components:
  A. Flask App (Python)
      a. data.py
          ● Connects to SQLite3 database and creates this table:
              ○ **Userdata** (stores personal info and score in trivia for competition purposes)
      b. __init__.py
          ● Routes:
              ○ **/register** adds a username and password to the **users** table. Checks if username is unique, stores username in session then redirects to **/home**
              ○ **/login** checks if username is in **users** table and if password matches. If so, stores username in session then redirects to **/home**
              ○ **/home** displays user's stats and paths to unboredom activities
                  ■ **/jokes** uses JokeAPI Generator to entertain user with jokes
                  ■ **/trivia** uses Open Trivia API to entertain users with a fun trivia game
                  ■ **/activities** uses Bored API to entertain user with some memes
              ○ **/leaderboard** displays user's stats and displays stats of top 10 users
              ○ **/logout** takes username out of session and redirects to /login
  B. Templates
      - Consistent navbar on all pages that redirects to different pages with a logout button
      - **/login** & **/register** (form input boxes, handles authentication)
      - **/home** (user stats, buttons towards following three)

- **/jokes**
- **/trivia**
- **/activities**

C. Database (SQLite3) (stored in data.db)
- **userdata** table stores all usernames and matching passwords plus stats of users

D. RESTful API's
These APIs were chosen because each part is connected to a "fun" feature: trivia questions, random jokes, and activities (we assume that the user is already bored when visiting).
- [Open Trivia API](#) (no sign-up needed; generated based on request & the user could decide the difficulty, question type, number of questions, and category)
- [JokeAPI](#) (no sign-up needed; flag categories to avoid inappropriate questions)
- [Bored API](#) (no sign-up needed; similar to Trivia API, can filter activities)

E. Frontend Frameworks
- Tailwind (buttons, filters, columns, background layering): Tailwind makes buttons, css fonts, wraps, and columns easy to categorize and style. Additionally, it allows you to leave a separate section specifically for aesthetics.
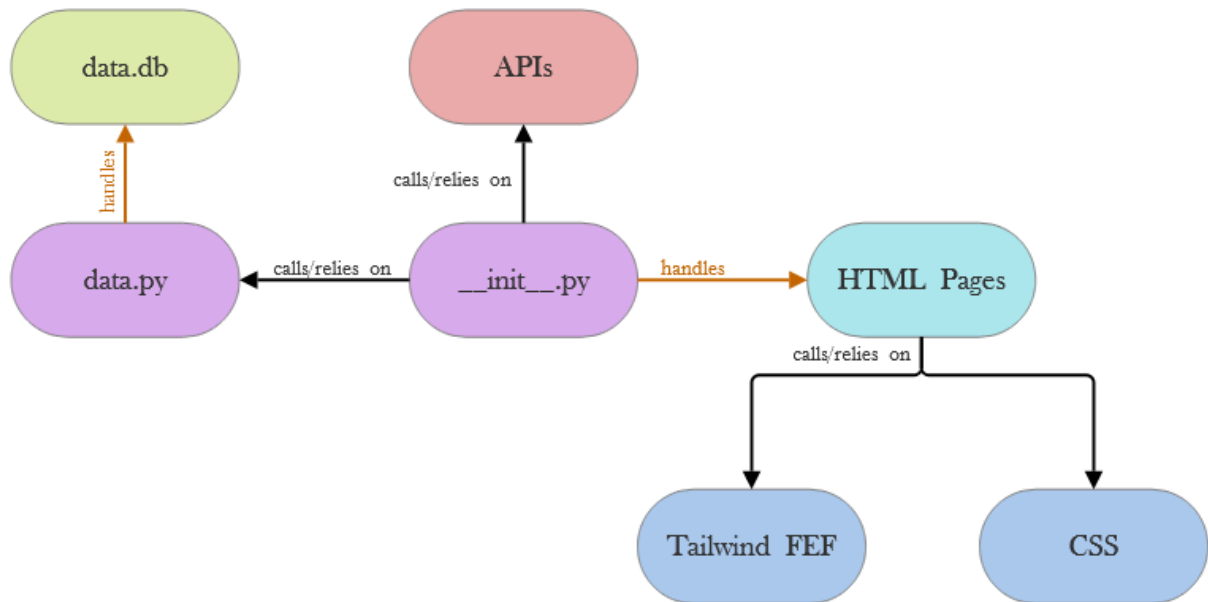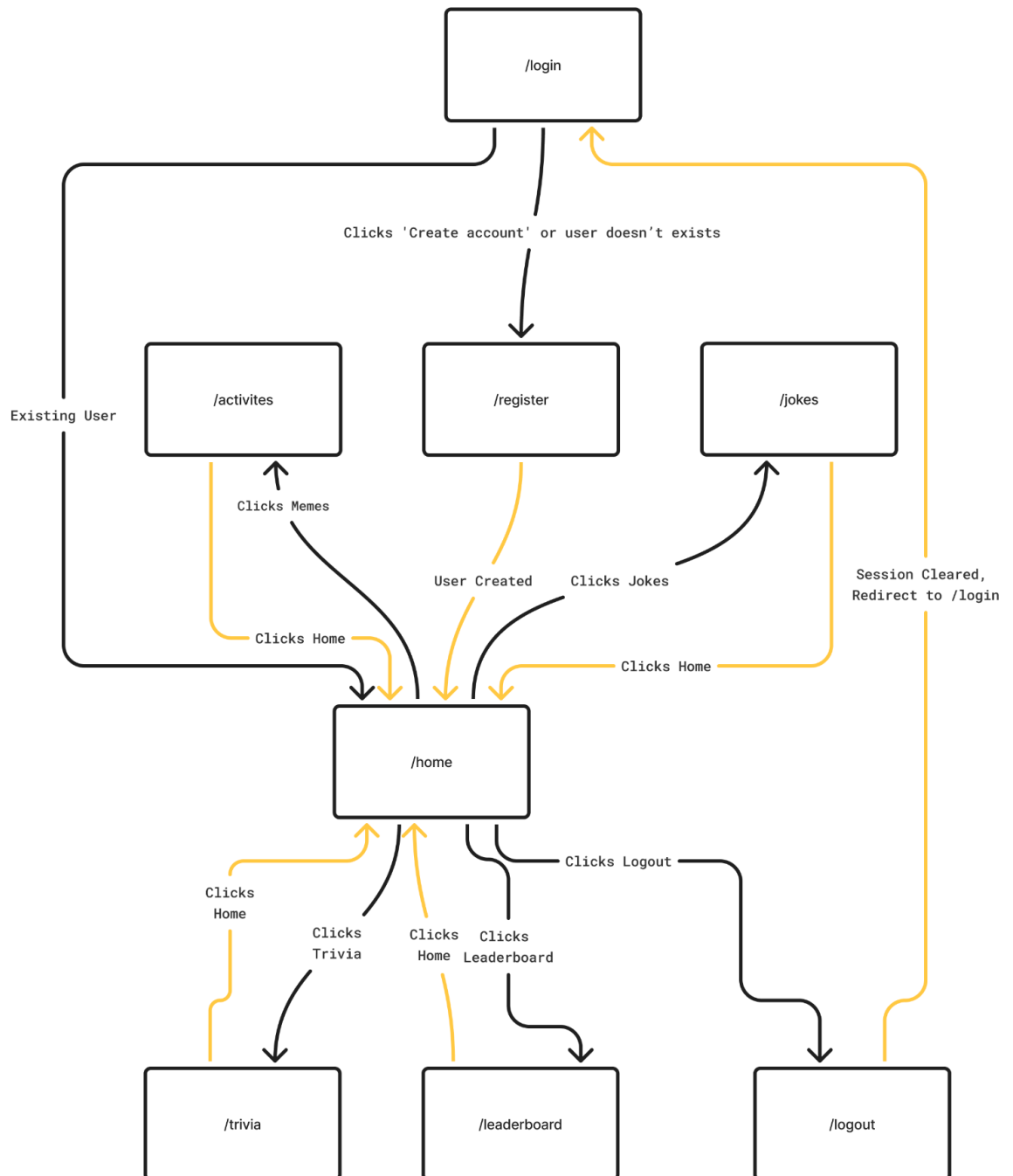
F. CSS as necessary

Database organization:

USERDATA

| string | username | PK NOT NULL |
|--------|----------|-------------|
| string | password | NOT NULL |
| string | bio | |
| date | sign_up_date | NOT NULL |
| int | trivia_score | |

Component Map:

# SQUASH COMPONENT DIAGRAM

Site Map:

## DESIGN DOC WHO DOES WHAT

| Task | Devo(s) |
| --- | --- |
| Program components & explanation | Jason Chan |
| Component map | Maya Berchin |
| DB Organization | Maya Berchin, Isabel Zheng |
| Site map | Aoanul Hoque |
| API section | Isabel Zheng |
| FEF section | Isabel Zheng |
| Task Breakdown | Everyone overseen by Jason |

## PROJECT WHO DOES WHAT

| Task | Devo(s) |
| --- | --- |
| Handle middleware | Maya Berchin |
| Login page | Jason Chan |
| Register page | Jason Chan |
| Home page | Maya Berchin |
| Jokes page | Isabel Zheng |
| Activities page | Maya Berchin |
| Trivia page | Aoanul Hoque |
| Leaderboard page | Aoanul Hoque |
| FEF and CSS implementation | Everyone overseen by Isabel Zheng |