TEE_BEE_DEE by JAMEs
P02: Makers Makin' It, Act I
Ethan (PM), Amy, Jason, Matthew
2026-01-09
TARGET SHIP DATE: *2026-01-16*

Project Description: Our project is a website that creates the soon-to-be most popular song in the world. This is done by taking the most popular songs' lyrics and randomly picking lines and putting them in an assigned template. There will also be functionality to create and share, share and save these songs, as well as your own original songs.

Program Components:

A. Flask App (Python)
    a. data.py
        - Connects to SQLite3 database and creates this table:
            - **users** (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 various
                - **/TSG** "top song generator" uses songs from top billboard songs and uses their lines to create the next platinum-selling song
                    - Displays song lyrics using Lyrics API
                - **/Speech-Text** uses Apple's search API to get 30-sec audio of the song for billboard
                - **/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)

C. Database (SQLite3) (stored in data.db)

- **userdata** table stores all usernames and matching passwords plus stats of users

D. RESTful API's

- Billboard API
    - https://rapidapi.com/sharmadhirajnp2/api/billboard-charts-api
- Lyrics API
    - https://docs.genius.com/
    - https://docs.musixmatch.com/lyrics-api/introduction
- Speech-Text API
    - https://performance-partners.apple.com/search-api
        - 30-second preview
    - https://cloud.google.com/text-to-speech (AI)
- Bored API
- Chord API
    - https://api.uberchord.com/
        - Gathers guitar chords

E. Frontend Frameworks

- Bootstrap
    - Grid for organizing leaderboard and profile stats
    - Buttons and cards for music that the user has created
    - Form styling, for use in login/register
    - Navbar functionality
    - Badges to indicate top/trending songs (in top 10 songs)
    - Spinners for loading page
    - Accordion for showing/hiding song lyrics

F. CSS as necessary

Database organization:

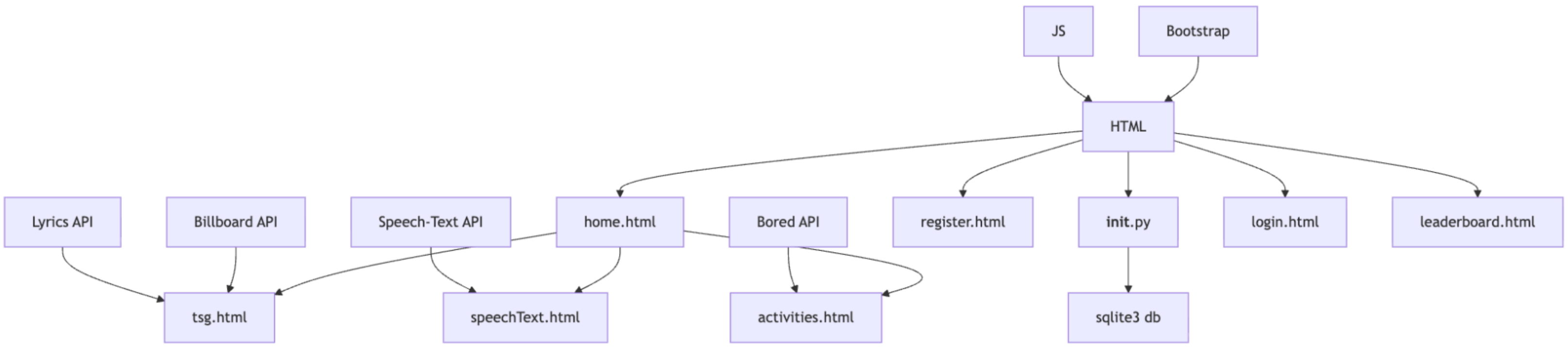
USERDATA

INTEGER	user_id	PK
TEXT UNIQUE	username	
TEXT	password	
TEXT	saved_songs	

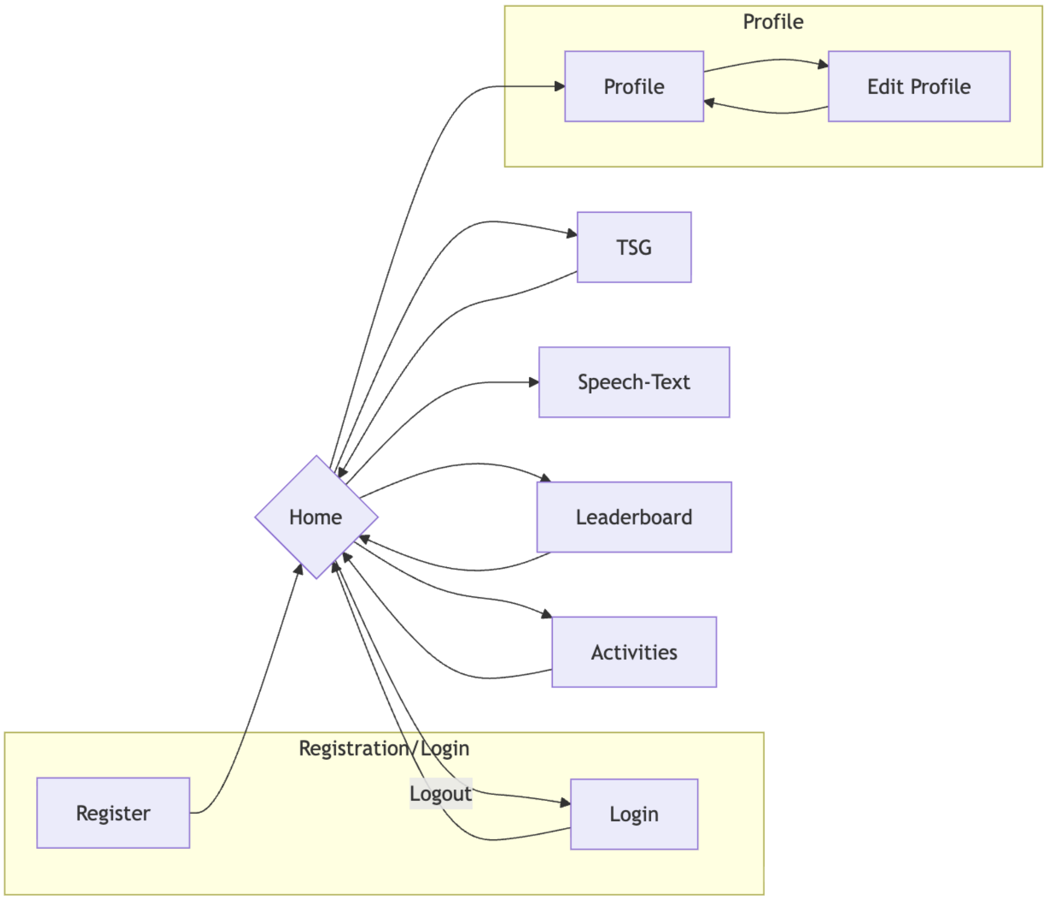
CHORDDATA

TEXT	chord_name	
TEXT	string_pattern	

Component Map:



Site Map:


Profile
Profile
Edit Profile
TSG
Speech-Text
Home
Leaderboard
Activities
Registration/Login
Register
Logout
Login

DESIGN DOC WHO DOES WHAT

Task	**Devo(s)**
Program components & explanation	Amy
Component map	Jason
DB Organization	Ethan
Site map	Matthew
API section	Amy
FEF section	Jason
Task Breakdown	Ethan

PROJECT WHO DOES WHAT

Task	**Devo(s)**
Handle middleware	Amy
Login page	Ethan
Register page	Ethan
Home page	Amy
Text-Speech page	Matthew
Speech-Text page	Jason
Activities page	Matthew
Leaderboard page	Jason
FEF and CSS implementation	All