

Roster: Sasha Murokh, Chloe Wong, Claire Song, Moyo Fagbuyi

TNPG: GarfieldsCatHair

p02-RoH: TuneTown - Social Media for Music Lovers

2025-01-08

Time Spent: 2 hours

TARGET SHIP DATE: 2025-01-17

Brief Overview

TuneTown is a social media web app for music lovers, structured similarly to Myspace and its parody SpaceHey. Users will be able to create profiles, review songs, and, if time permits, directly communicate with one another. The Spotify API will allow users to search up musical artists and learn about their discography.

Components

- Flask app: Host web page, connect frontend and backend
- Databases: Store users and blogs
 - User accounts: Profiles including username, display name, profile image as image link, background image as image link, and an about me section
 - Blog posts about music: Allow users to review music, with songs embedded into posts
- Spotify API integration: Allow users to search up desired information about musical artists and their works
- HTML templates: Provide layout for web pages
- CSS: Provide style for HTML templates
- Tailwind: Frontend framework

Templates Folder

- login.html
 - Allows the user to input a username and password
- register.html
 - Allows the user to register an unused username and password to be added to the database
 - Profile fields (profile picture/background image/about me) is filled in with placeholders until user modifies
- error.html
 - It will be rendered when an error occurs (e.g., a user tries to create an account with the same username as another user).
- index.html

- Visible to all
- Explains the purpose of the site and shows other's profiles
- profile.html
 - Shows a signed-in user's profile
- settings.html
 - Allows users to change their password (by confirming their current one)
 - Allows users to change their "about me"
 - Allows users to change their profile pictures and background pictures by pasting in an image link
- search_music.html
 - Parameters for search (artist/track/album) and search bar
 - Search artist for artist's top songs/albums/etc
 - Search song for track length/release date/etc
 - Search album for track list/featured artists/etc
 - Displays results
- blogs.html
 - List of posts made by users
- create_blog.html
 - Make new post
- Stretch goal: Message page: read received messages
 - Same idea as blog page except only messages directed towards a certain user
- Stretch goal: Write message page
 - Same idea as write post page except with section to enter the recipient username

APP Folder

- build_db.py
 - Use SQLite3 to create a database of user information (username and password).
- spotify.py
 - All spotify functions
- __init__.py
 - Initializes the app.
- config.py
 - Checks that API keys exist and are stored in the right place.
- keys folder
 - key_spotify_client_id.txt

- key_spotify_client_secret.txt
- readme
 - Explains how to get the API keys
- Templates Folder
 - login.html
 - homepage.html
 - register.html

Frontend Framework

- Tailwind
 - We decided to use Tailwind because it's easy to understand and consistent. We plan to make use of its button/box customization. If able, we'd like to use its dark mode feature now that we're a little more comfortable with using JavaScript.

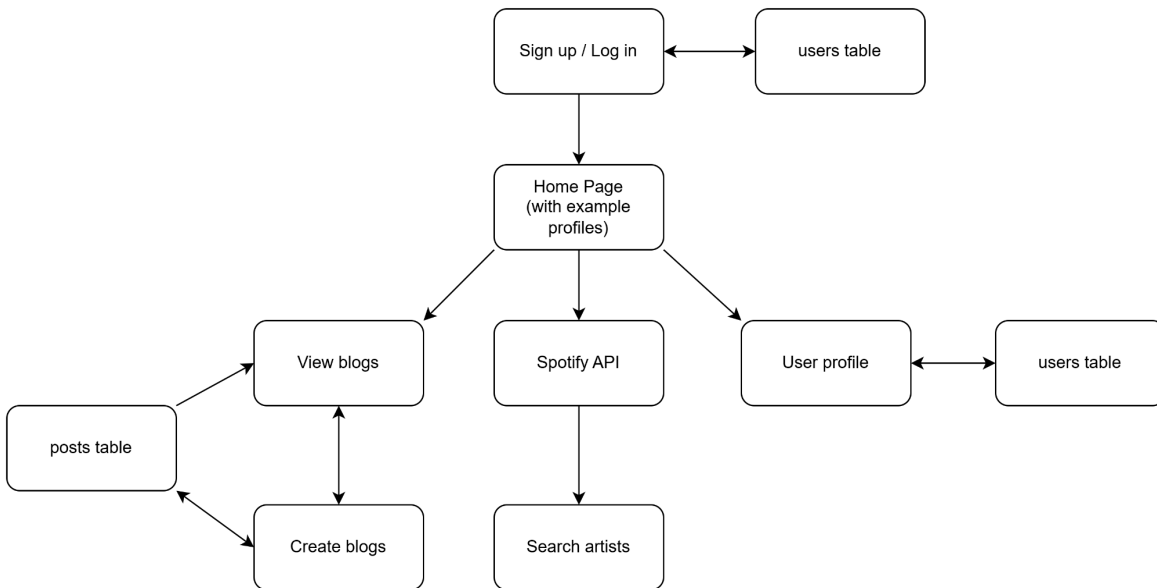
APIs

- Spotify API
 - Allows users to search for Spotify content (e.g., artists, albums)
 - Allows users to find information about searched content
 - Utilizing this, Spotify links can be easily embedded into posts
 - Requests are rate limited within a rolling 30-second time frame (the exact number of not calls allowed is not shared)

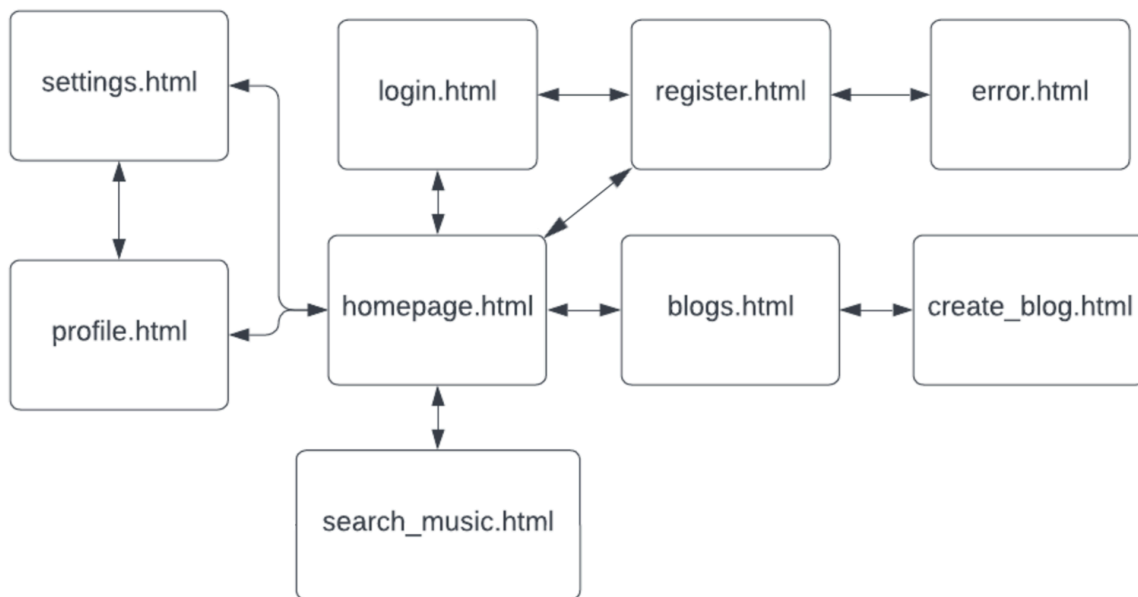
Documentation and Assets

- design.pdf
 - This document. Devos should refer to and edit it often.
- flag.jpg
 - Our team's flag. Proudly portending greatness.
- README.md
 - Roles, description of website, how to, etc. Please read it.
- devlog.txt
 - Updated by each group member after they push to GitHub and make changes. Maintained by PM. The most recent entry goes on the bottom detailing changes and bugs.
- requirements.txt
 - Lists flask and pip installs needed to run the application

Component Map



Site Map



Database map

users

index	username	password	pfp	background img	aboutMe
INTEGER	TEXT	TEXT	TEXT	TEXT	TEXT

posts

index	username	post
INTEGER	TEXT	TEXT

Stretch goal: Messages

index	sender username	recipient username	post
INTEGER	TEXT	TEXT	TEXT

Assignments

<u>Task</u>	<u>Assigned To</u>
Flask App Development	Chloe, Claire
Database Engineering and API Implementation	Sasha (PM)
HTML and CSS/Tailwind	Moyo
Design Doc, Devlog, Testing, and Debugging	All members