UniSpot

Kevin Buhler, Blake Hamilton, Tristan Hunt, Joanna Parker, Baxter Romero, Shuchi Shah

Description of the Project

- UniSpot is a website where you can can find out what your university's top songs
- UniSpot also lets you search songs based on keywords
- The trends page allows you to see when and how many times the songs was listened to
- Check out you profile page to see what you’ve been listening to

Tools

Project Tracker/VCS Repository

GitHub Project Board

- *Github* was our primary project management tool.
  - Track the progress of stories
  - Assign stories/tasks
  - Request feedback
- When it came to more miscellaneous items/project planning, we used a *Google Drive* Folder to share documents
  - (Google docs w/ ideas, database diagrams, etc…)

GitHub Repository: Version Control System

- We tracked all changes to code/meeting logs/milestones in git throughout the project…
- This made progress iterative and unified (each pull to main reviewed and discussed)

Rating: 5 (a necessity)

Database

PostgreSQL

- Purpose: a powerful, open source object-relational database system
- Rating: 3
- Methodologies: storing, maintaining and accessing our user data
  - We track: username, password, university for each user
  - When a user indicates a liked song/listen: We track that entry as a ‘transaction,’ saving data about the song, the user, and the time they liked that song.

IDE

Visual Studio Code

- Purpose: enables programmers to consolidate the different aspects of writing a computer program
- Rating: 4
- Methodologies: iteraverive, individual programming, pair programming

UI Tools

EJS

- Purpose: embed JavaScript code in a template language that is then used to generate HTML
- Rating: 2
- Methodologies: create website pages

Tailwind CSS

- Purpose: open source CSS framework
- Rating: 4.8
- Methodologies: design website pages

ChartJs

- Used to create customize our data_trends barchart.

Application Server

Node JS

- Purpose: a multi-purpose server-side processing engine
- Rating: 3
- Methodologies: back-end use

Deployment Environment

- Did not complete

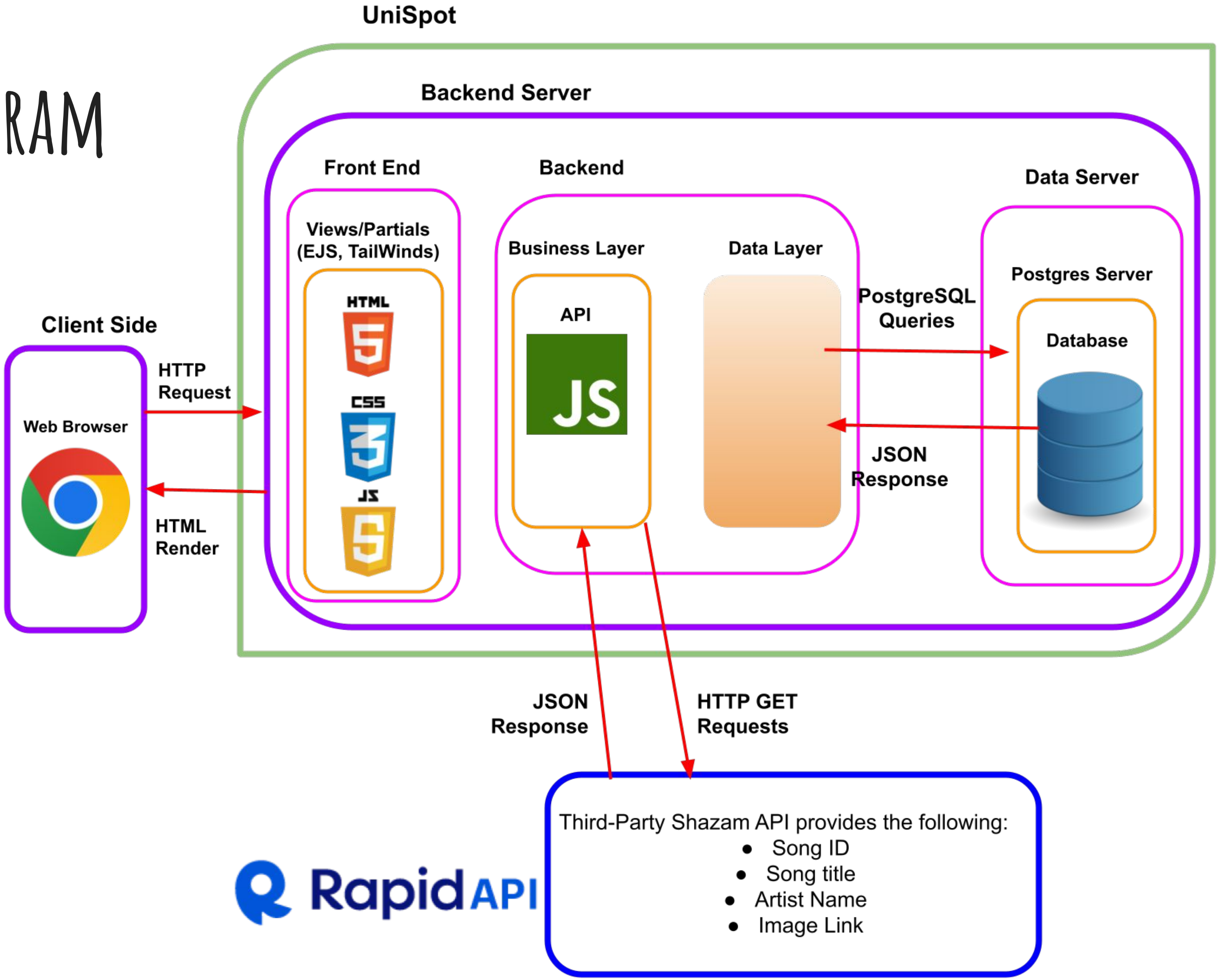
External APIs

Rapid

Rapid API

- Using third party Shazam API
- Purpose: consume any API using a unified format that is easy to understand and embed in your application
- Rating: 4
- Methodologies: understanding Shazam’s API

Architecture Diagram



Third-Party Shazam API provides the following:

- Song ID
- Song title
- Artist Name
- Image Link

Challenges

- Originally we planned on using Spotify's API, but due to it not being provided for common use we decided to use Shazam's public API.
- Password Hashing created a lot of problems for us.
- We also planned on having a page for artists to view their top songs by university, but due to lack of API calls and time, it was not a feasible addition.

Demo

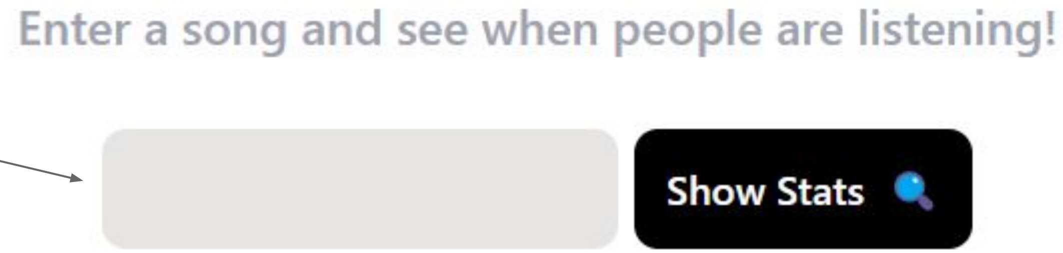
Demo

UniSpot Demo Video

Future Enhancements

- Have an artists page, where artists can find out what universities are listening to their songs

On data_trends page, searching for a song would generate a dropdown on the search bar and allow user to select the specific song they want to see data for.



Enter a song and see when people are listening!

Questions