

PROJECT 1

Pd 9 | Team Monotunes

Kevin Li, Adeebur Rahman, Michael Cheng, James Smith

WHAT DOES IT DO?

Our program is a lyrics library where you can search up a song by its name and artist or just search through an artist's albums to find a song. Once a song is selected, it brings you to a page displaying its lyrics and an audio file can be played through the page that reads you the lyrics. You may also have an account if you want to save your favorite songs somewhere.

OUR APIs

Musixmatch - a music database API to supply our search feature and lyrics

IBM Watson Text to Speech - Takes text you send it and returns an .wav file

TEXT TO SPEECH LYRICS LIBRARY - COMPONENTS

HTML files

See 'Files' section for details

Stylesheets

Bootstrap's as well as our own

db.py

Handles the database, including login information and saving music preferences.

main.py

Flask app, handles front-end navigation and operations

api.py

Handles API calls and extracting information

profiles.db

Contains user information including username, password, saved music

FILES

template.html

login/logout button, profile button, and homepage button on a toolbar

home.html

Presents a set of suggested songs(top 10 trending songs)

an intro/how to use the website

Logout/Login button

Profile Page button

Create Account button

create.html

Asks for profile information for first time setup

login.html

Form for username and password

Has a link to the create account page

profile.html

Contains list of links to your favorite songs

Option to remove songs from favorites

song.html

Shows song title, artist, and lyrics

Includes link to '/artist' route

Takes query strings "title"

Option to the song to your profile's favorites

artist.html

Shows songs of the artist by album, which are links to the '/song' route

Takes query string "artist"

search.html

Form to search by artist

Form to search by song

error.html

Displays error message

style.css

To make the website pretty

bootstrap.min.css and bootstrap.min.js

Incorporating Bootstrap

main.py

- **add_session(username, password)** - creates a session cookie for the current user with valid credentials or returns error message
- **logout()** - Deletes a session key of a user, if one is logged in
- **root()** - provides functionality for the / route
- **login()** - provides functionality for the /login route
- **create()** - provides functionality for the /create route
- **profile()** - provides functionality for the /profile route
- **song()** - provides functionality for the /track route
- **artist()** - provides functionality for searching for a song by artist
- **search()** - provides functionality for the /search route
- **is_null(username, password, confpw)** - checks to see if any of the username or password submissions were null

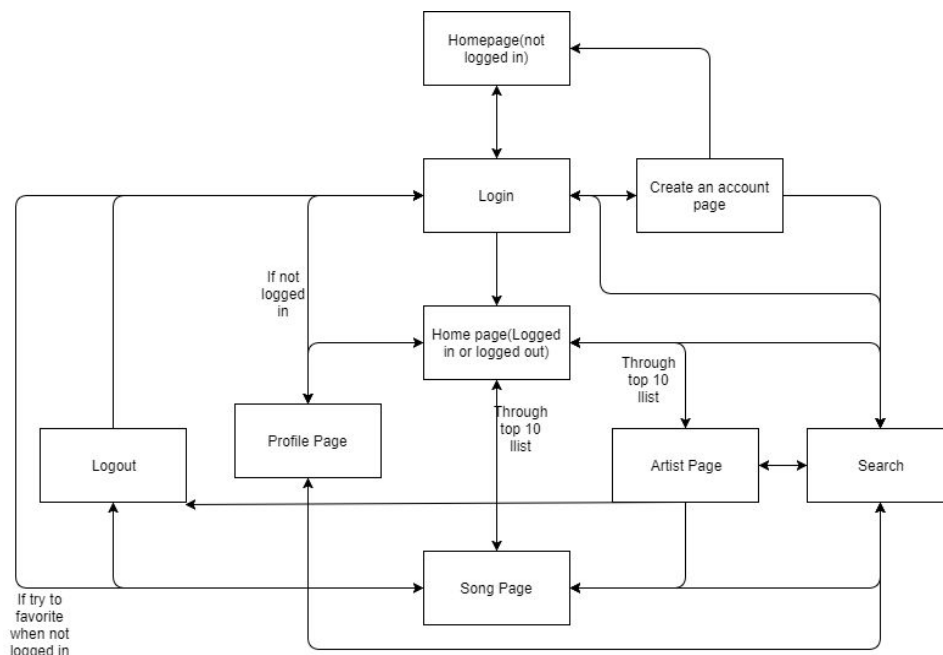
api.py

- **get_song_id(track, artist="")** - returns the songid for the desired song
- **get_track(trackid)** - returns the track information given its trackid
- **get_lyrics(trackid)** - returns the lyrics of a song given its trackid
- **get_artistid(artist)** - returns the artistid of an artist given their name
- **get_albums(artistid)** - returns a dictionary of the albums
- **get_album_tracks(albumid)** - returns a dictionary of tracks on the album
- **get_top_songs()** - returns a dictionary of the top songs
- **get_wav(text, filename, voice = "en-US_AllisonVoice")** - downloads a wav file of speech generated from text and returns the file name.

db.py

- **login(username, password)** - Logs user in if username and password are correct
- **encrypt_password(password)** - Returns SHA-256 version of password
- **create_account(username, password)** - Creates a new entry in .profiles table
- **get_favorites(username)** - Returns a list of all the username's favorites
- **does_username_exist(username)** - Returns true if username exists, false otherwise
- **add_favorite(username, trackid)** - Appends trackid to current list of favorites using eval(idList) and repr(idList)
- **is_favorite(username, trackid)** - Returns true if user has this song already in their favorites
- **remove_favorite(username, trackid)** - Removes trackid from username's list of favorites

SITEMAP



DATABASE SCHEMA

Profile Table

The profile table contains 3 fields: username, password, and favorites. Username and password are used to log the user in. Username is a primary key so that you can't have a duplicate account. The last field, favorites, is a string representation of a list of track IDs (musixmatch API Song ID) that the user has selected as favorites. To edit the favorites list, `eval(trackList)` can be used to get a working list, then the new ID can be appended, then `repr(trackList)` can be used to turn it back into a list to update that field.

Username (Primary Key)	Password *encrypted	Favorites (String of a list)
watson	ibm135	'[234678435, 238956384, 128593759, 285749384]'
sherlock	shrek	'[234678435, 573956382]'
moriarty	p455w0rd3	'[]'

TASK ASSIGNMENTS

Kevin: Project Manager and CSS

Adeeb: Backend and watson API

Michael: Frontend HTML files (helps other members when finished)

James: flask app and musixmatch API