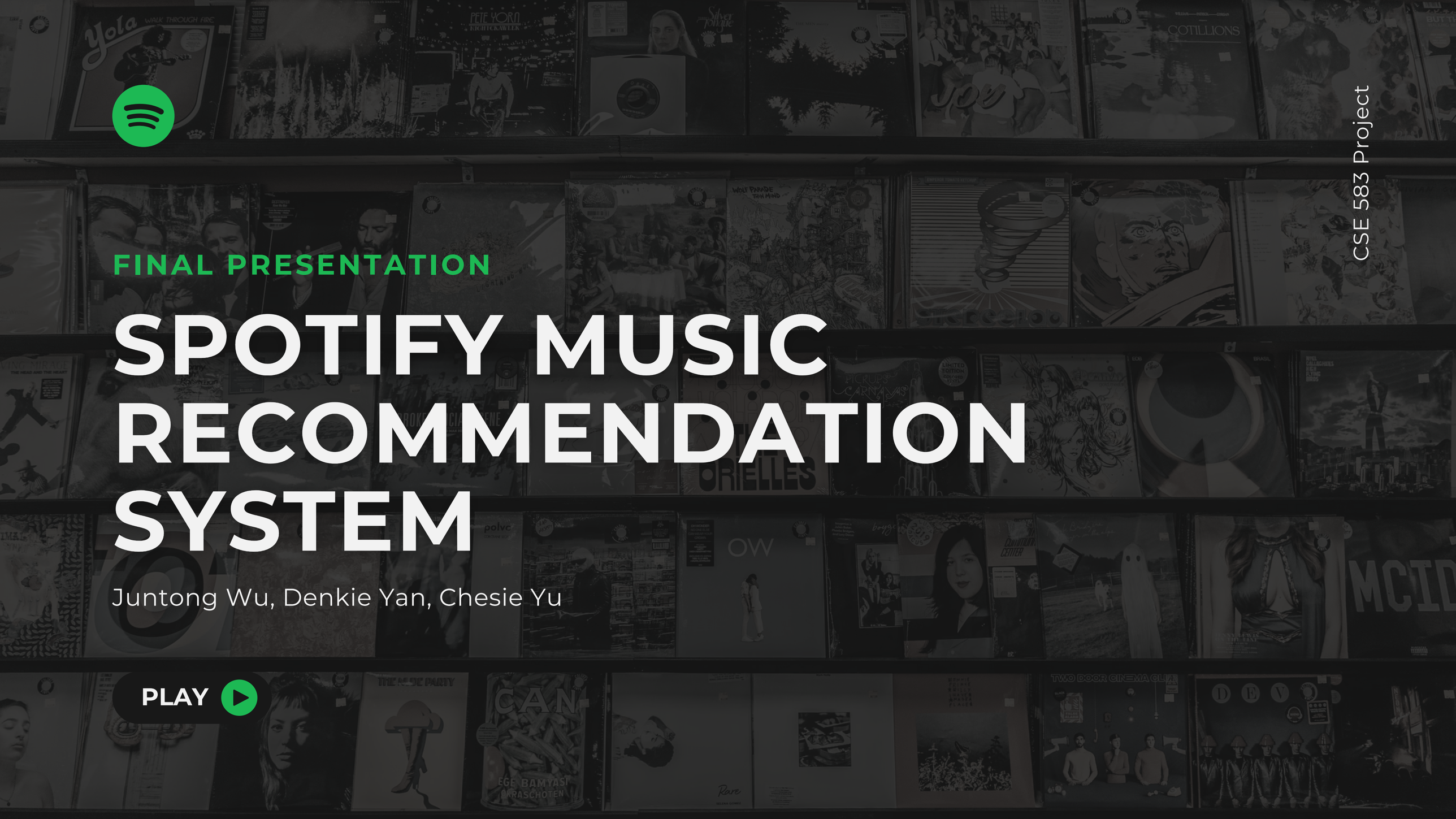


CSE 583 Project

FINAL PRESENTATION

SPOTIFY MUSIC RECOMMENDATION SYSTEM

Juntong Wu, Denkie Yan, Chesie Yu

PLAY

Project Background

- Discover patterns and trends from the Spotify dataset (i.e., music evolution, popular genre, etc.)
- Develop a music recommendation system using collaborative filtering
- Build up an interactive user interface using dash
  - Allows users to explore music trends and characteristics
  - Provides personalized music recommendation to users

Data

Spotify Dataset

- **Track Information:** song name, artist names, etc.
- **Audio Features:** acousticness, danceability, energy, instrumentalness, liveness, loudness, etc
- **Popularity Metrics:** popularity score
- **Temporal Data:** release date
- **Identification Data:** Unique identifiers for song (song ID) and artist (artist ID)

Dummy Dataset



- **Users profile image:** generated using stable-diffusion-v1-5
- **User Songs data:** used for collaborative filtering
  - avg. 50% songs most listened by each user (listening count > 10)
  - avg. each song has 5 users who listened
- **User Friends data:** used for collaborative filtering
  - min. 3 friend relations per user
  - max. 20 friend relations per user

```
# Process the batch
for i in range(start_index, start_index + batch_size):
    # Make sure we don't go out of bounds in the last batch
    if i >= len(users_df):
        break

    # Access the user row by row index
    user = users_df.iloc[i]
    prompt = f"A cute cartoon image of a music lover named {user['first_name']} {user['last_name']} at the age of
    image = pipe(prompt).images[0]

    # Save the image
    image.save(f"./users_profile_3/{user['user_id']}.png")
```

Use Case

Discoverability

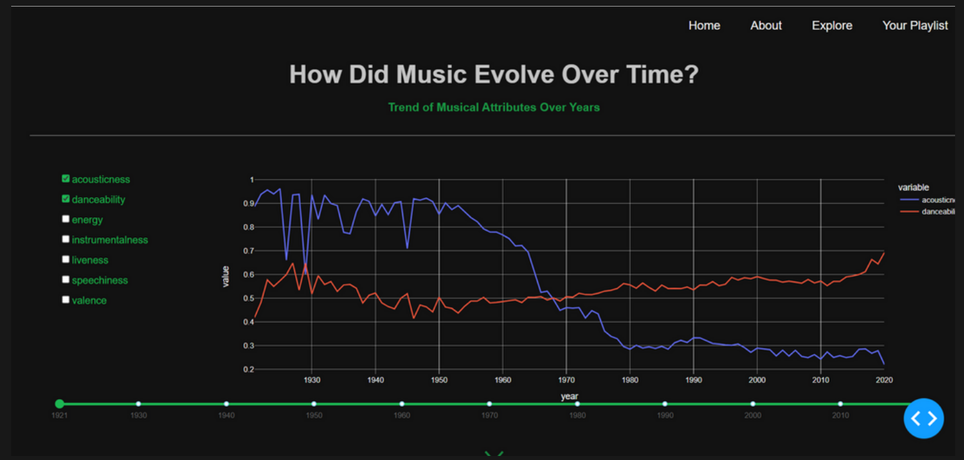
Introduce users to songs and artists they might not have discovered on their own, expanding their musical horizons

Personalization

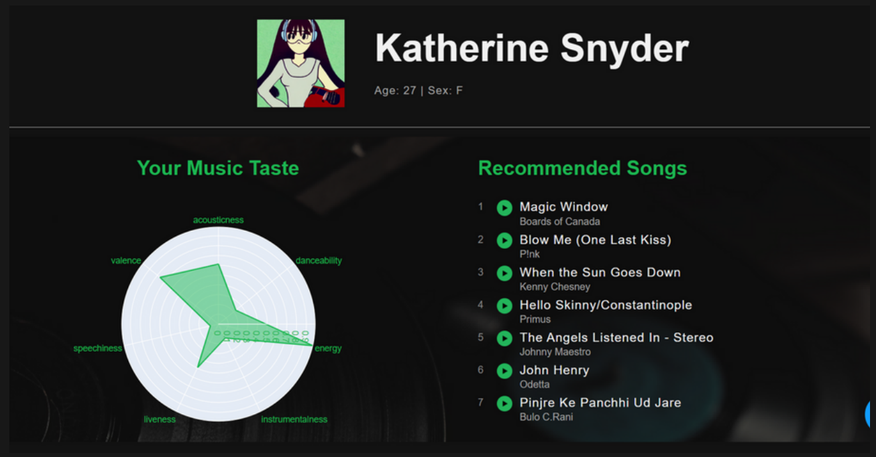
Personalized recommendation based on the user's listening history and music taste

Our User Recommendation Platform:

- Explore music Trends and Characteristics



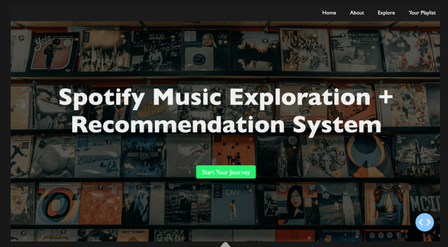
- Personalize Music Recommendation

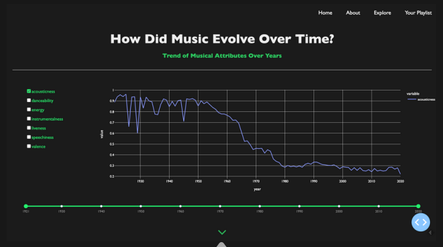


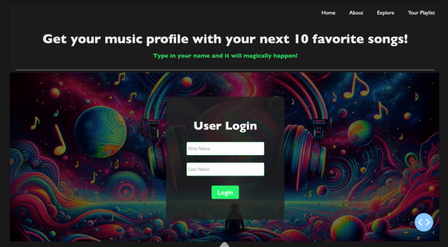
User Flow

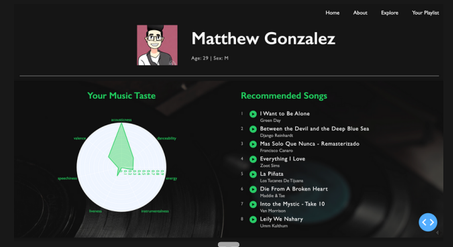
Requires:
User Authentication

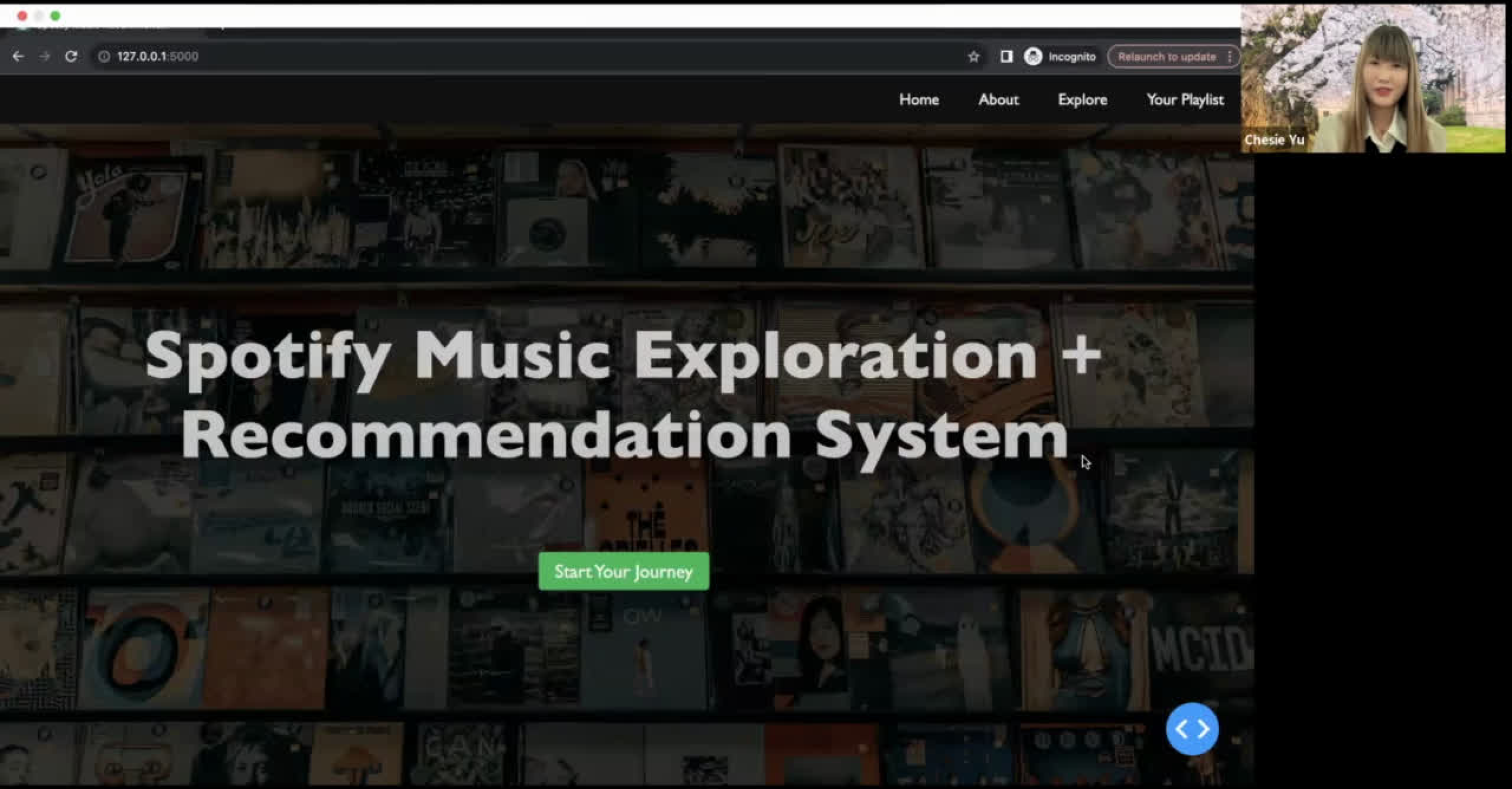
Landing Page → **Explore Page** (Use Case #1: Music Trends + Characteristics) → **Login Page** → **User Profile** (Use Case #2: Music Taste + Personalized Recommendations)



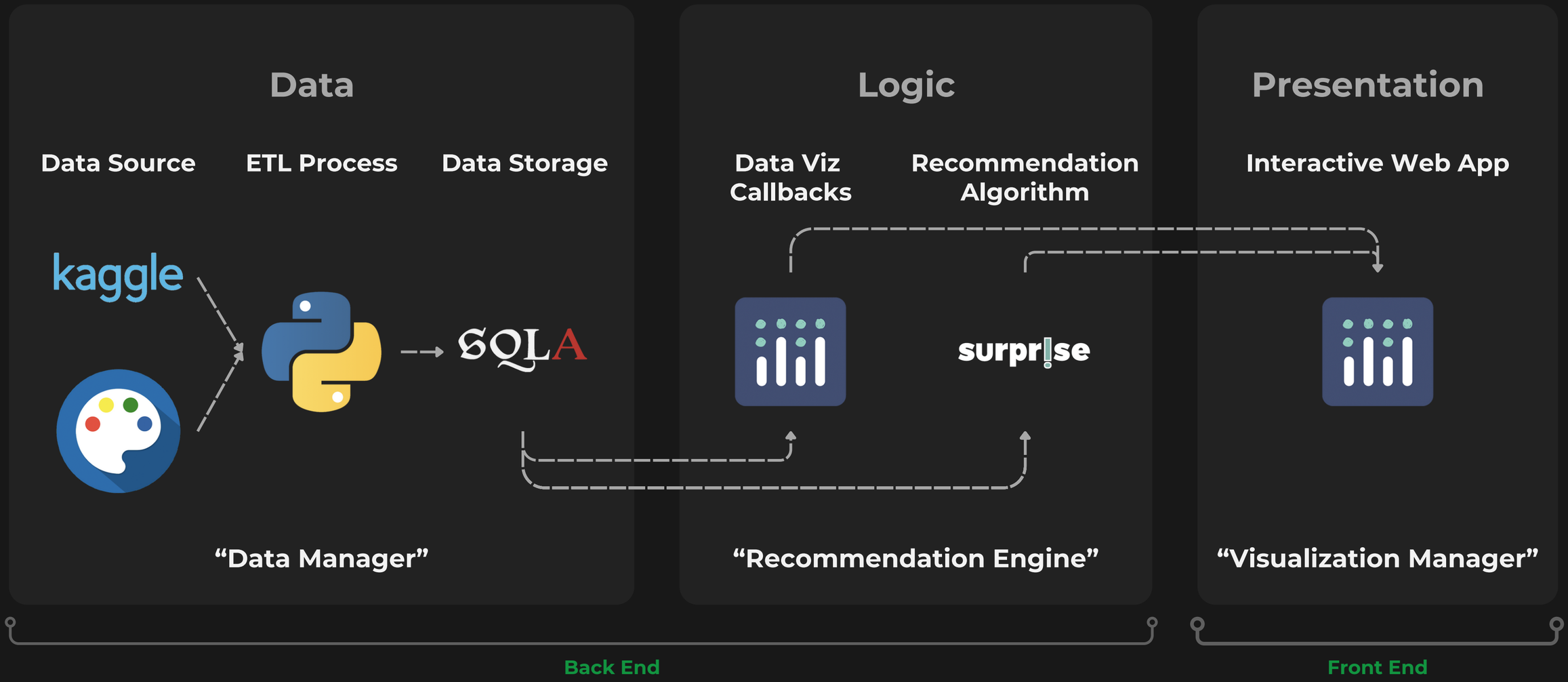




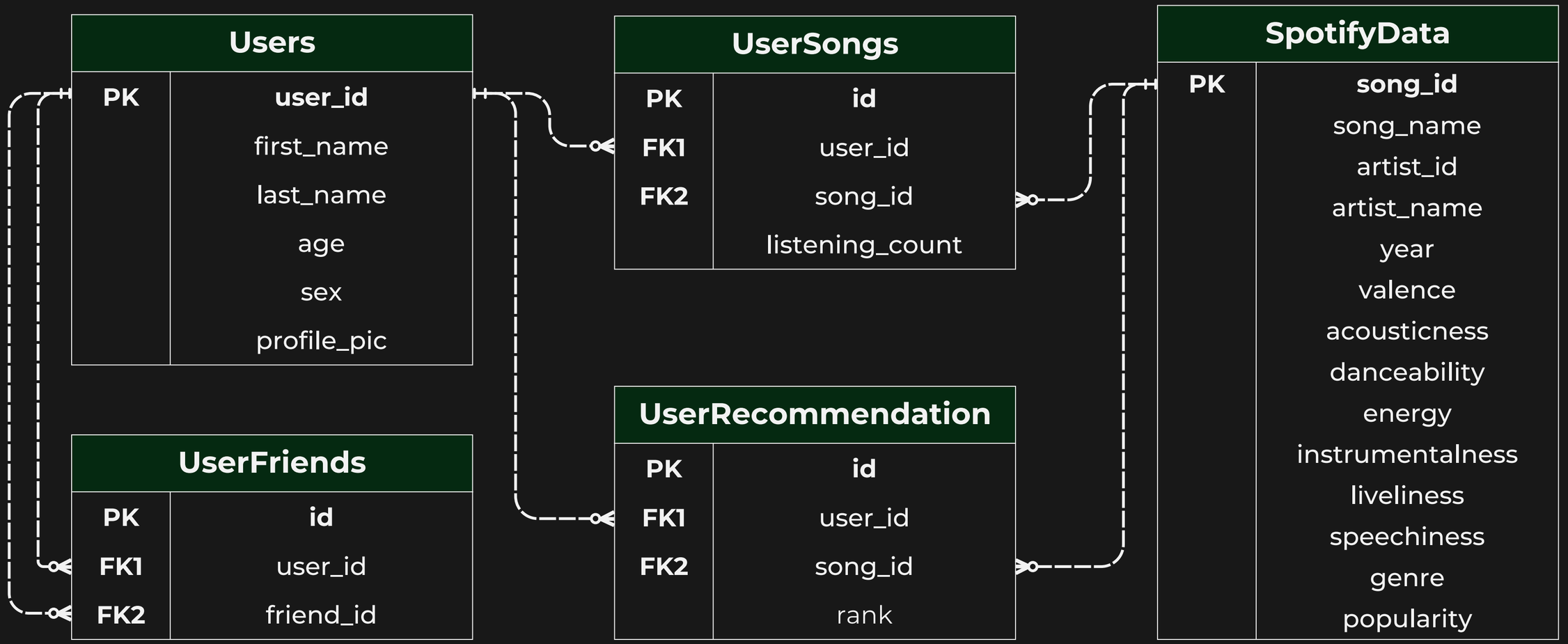



127.0.0.1:5000
Incognito
Relaunch to update
Home
About
Explore
Your Playlist
Chesie Yu
Spotify Music Exploration +
Recommendation System
Start Your Journey

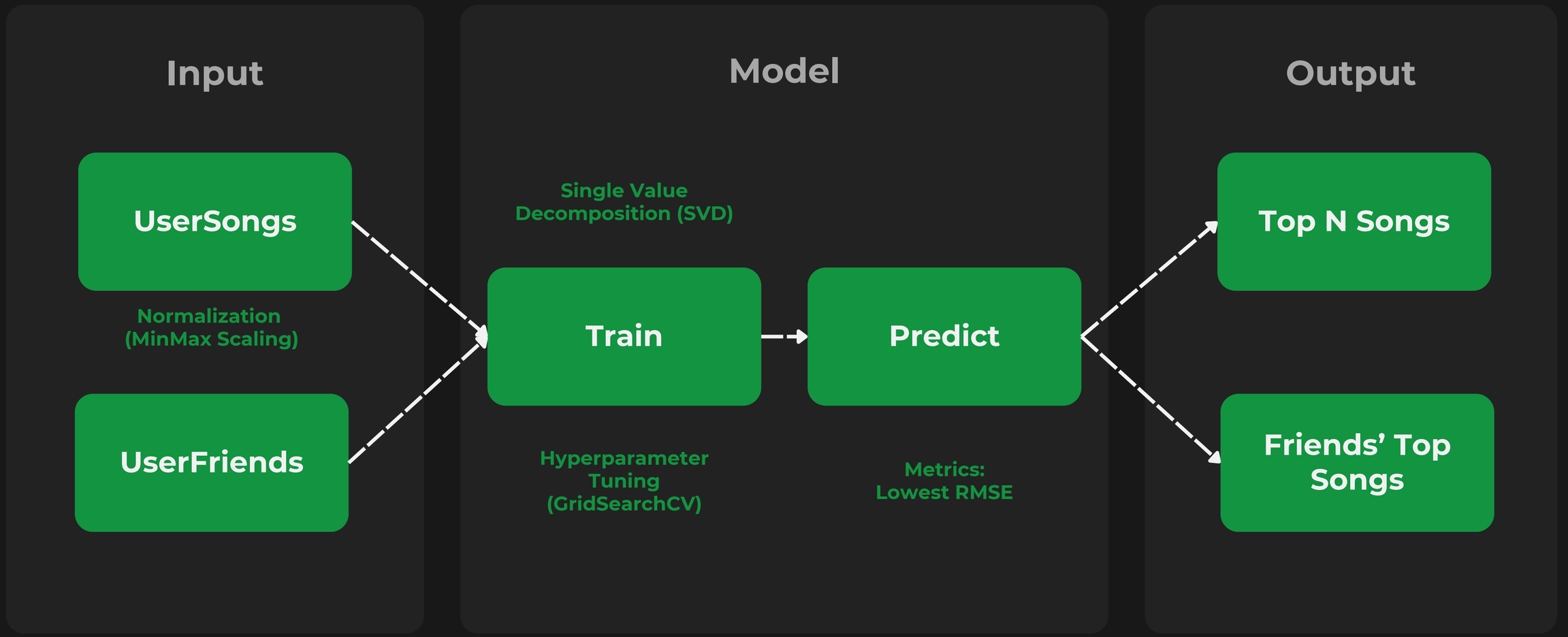
Architecture Diagram



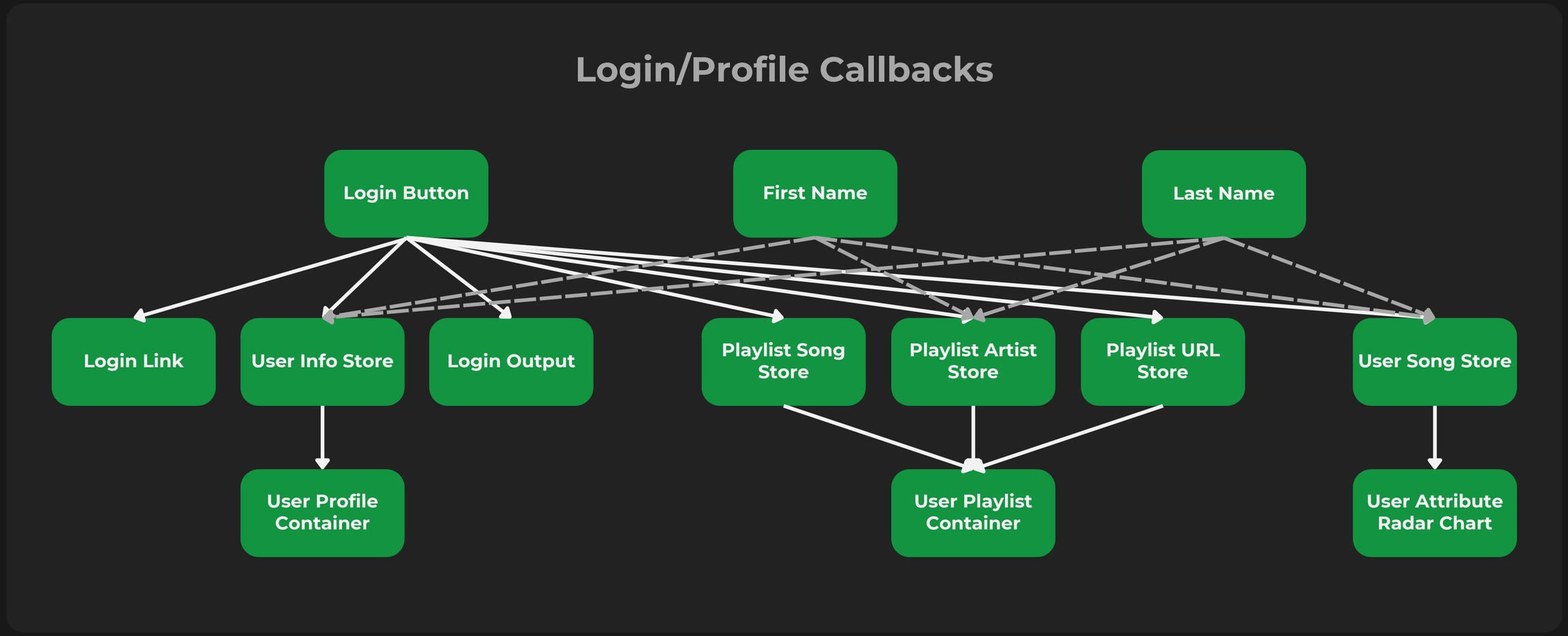
ER Diagram


Users
PK
user_id
first_name
last_name
age
sex
profile_pic
UserSongs
PK
id
FK1
user_id
FK2
song_id
listening_count
SpotifyData
PK
song_id
song_name
artist_id
artist_name
year
valence
acousticness
danceability
energy
instrumentalness
liveliness
speechiness
genre
popularity
UserFriends
PK
id
FK1
user_id
FK2
friend_id
UserRecommendation
PK
id
FK1
user_id
FK2
song_id
rank

Recommendation System



Plotly Dash App



Project Structure

```
Spotify-Music-Recommendation-System (main)
|      # Main file; run this to start the app
|----- app.py
|     |
|      # All the graphics, stylesheets, and scripts used in the project
|----- assets
|     |
|      # All datasets used; in .csv format
|----- data
|     |
|      # Body of displayed page
|----- pages
|     |      about.py
|     |      explore.py
|     |      home.py
|     |      login.py
|     |      recom.py
|     |
|      # All the functionality and callcacks for pages
|----- utils
|     |----- pages
|     |      |----- explore
|     |      |      |      callbacks.py
|     |      |      |      visuals.py
|     |      |----- login
|     |      |      |      callbacks.py
|     |      |      |      usermatch.py
|     |      |----- recom
|     |      |      |      callbacks.py
|     |      |      |      model.py
|     |      |      |      playlist.py
|     |      |      |      profile.py
|     |----- database.py
|     |----- spotipy_db.sqlit
|     |----- page_template.py
```

Typical Dash APP Structure:

- app.py
- assets
- data
- pages
- utils

Project Structure

```
|      # Data preprocessing workflows
|----- preprocess
|      |
|      # All the functionality tests for this project
|----- tests
|      |      _init_.py
|      |      test_database.py
|      |      test_login.py
|      |      test_recom.py
|      |      test_visual.py
|      |
|      # Design documents
|----- doc
|      |      component-spec.md
|      |      functional-spec.md
|      |      technology-review.pdf
|      |      technology-review.pptx
|      |
|      # All images used in README and docs
|----- image
|      |
|      # User guide for app operations
|----- example
|      |      User Guide for Spotify Music Exploration & Recommendation.pdf
|      |      userguide.md
|      |
|      # Environmental file
|----- environment.yml
```

Other vital components:

- **preprocess**
- **tests**
- **doc**
- **image**
- **example**
- **environment.yml**

Lessons Learned

HANDLE DATABASE OPERATIONS CAREFULLY

Using an ORM (Object-Relational Mapping) like SQLAlchemy can simplify database interactions, but it also requires careful handling of sessions and transactions.

MODULE & DIRECTORY STRUCTURE

Ensure a well-organized directory and module structure - design beforehand and keep iterating

IMPORTANCE OF TEST-DRIVEN DEVELOPMENT

Write tests before actual code, which can lead to more reliable, maintainable, and error-free applications

Lessons Learned

HANDLING DYNAMIC CONTENT IN WEB APPS

Have a good understanding of how frontend frameworks interact with backend logic to ensure a seamless and responsive user experience

ENVIRONMENT & DEPENDENCY MANAGEMENT

Using tools like virtual environments, Docker, or Conda can significantly reduce "works on my machine" problems.

Future Work

1. Improve log-in logic by creating user account and passwords
2. Enrich visual types to present more musical and user information
3. Update HTML/CSS design to improve user experience
4. Incorporate content filtering to improve recommendation model performance

CSE 583 Project

THANK YOU!

https://github.com/CSE583-Fall2023-Project/Spotify-Music-Recommendation-System

Juntong Wu: juntongw@uw.edu
Denkie Yan: denkie@uw.edu
Chesie Yu: cyu909@uw.edu

PAUSE