Spotify Music Genre Explorer

Introduction

Spotify Music Genre Explorer is an interactive Python 3.9+ web application built using the Streamlit framework. It allows the user to explore music by genre using real-time data pulled from the Spotify Web API. It shows track information, album artwork, popularity metrics, artist popularity charts, audio previews, and a geographical demo map. With this application, the user can interact with widgets such as dropdowns, sliders, radio buttons, text inputs, and checkboxes. The project targets Human–Computer Interaction design principles to create an intuitive, effective, and engaging user experience.

Usability Goals

The primary usability goals for the Spotify Music Genre Explorer were to create an interface that is effective, efficient, easy to learn, and satisfying for users. The application tries to ensure effectiveness by allowing users to retrieve music data based on genre quickly and with a minimum amount of effort. Efficiency is supported by a clean layout, such as having controls in the sidebar. This allows users to change settings and explore results without navigating elsewhere. Much focus was put into learnability, which was achieved by using clearly labeled widgets and a simple workflow that a new user can understand immediately. Error prevention is ensured by limiting user inputs through dropdowns and sliders, as well as through helpful system messages like warnings, errors, and exceptions. Overall, the app is designed in a way that provides maximum satisfaction to its user through their engagement with the visual experience of album art, audio previews, charts, and customizable views that make exploration intuitive and enjoyable.

Design Process

This project's design process incorporates user research, usability testing, and planning for accessibility. User research began by identifying potential users, such as music enthusiasts and casual Spotify listeners, and garnering insight into their expectations of a digital music exploration tool. Feedback suggested that users wanted to see visual elements, like album artwork, as well as have ease of navigation and efficiency in getting popular tracks. Several rounds of usability testing were conducted with friends, whose feedback was taken into consideration. With their feedback, I was able to know there was a need to support both table and column views, and add an artist popularity chart. To add on, accessibility was also considered for the entire design process, including sticking to high contrast for text, clearly labeling items, and providing fallback messages when an image could not be found. These steps resulted in an inclusive user-centered application.

API Integration

The application communicates with the Spotify Web API using the Spotipy library to get music and artist information based on the selected genre. When users choose a genre, the app sends a

request to Spotify's search endpoint and receives track metadata, which includes track names, album titles, popularity scores, album art URLs, preview audio URLs, and artist identifiers. Additional API calls are made to get artist popularity, which are used to create the artist popularity bar chart. I had several challenges during this process, such as inconsistent data being returned for certain genres and having occasional absence of artist popularity data. These issues were addressed by implementing fallback and warnings to inform users of missing data.

Interactive Widgets Used

The application uses a variety of interactive widgets to increase user control and interaction. The sidebar contains a text input box that allows a user to optionally input their name, a select box to choose a genre, a slider to choose how many tracks are retrieved, and a radio button to switch between the column-based and table-based views. Additional checkboxes toggle optional visual elements such as the random map demo and the artist popularity chart. A main button labeled "Explore Genre" initiates all data retrieval and content updates. These widgets work to create a responsive, interactive environment that lets users intuitively and engagingly explore the app.

HCI Design Principles

The Spotify Music Genre Explorer was developed using Human–Computer Interaction principles in order to enhance usability and user experience. All interactive controls have been kept in the sidebar for visibility, making it easy for the user to find options and easily understand what is available. The application offers success, warning, and error messages, and dynamic visual elements like album artwork, charts, and audio previews of the system responses keep the user well-informed. Consistency in the interface was maintained through using uniform formatting, repetitive layout patterns, and predictable behaviors of widgets. This enables users to go through the application easily. Flexibility is supported through options that can be changed by the user, such as the table and column views, whether to show charts or not, and whether to show the map. Error prevention was another important consideration by limiting user inputs through already defined lists of genres, sliders, and checkboxes, catching API errors safely, and displaying them.

Conclusion

The Spotify Music Genre Explorer is successfully able to integrate real-time Spotify API data with an interactive Streamlit interface. The app is visually pleasing, user-friendly, and built around HCI principles. It fulfills all technical and usability requirements.