Harmonify – Documentation

Project Overview

Harmonify is a web-based music search and lyrics application that allows users to explore songs, retrieve lyrics, and access tracks on **Spotify**. The project integrates multiple APIs and features an advanced filtering system, pagination, and a user-friendly playlist functionality. The primary focus during development was overcoming technical challenges such as **API limitations, caching issues, and backend optimization** while ensuring a seamless user experience. The project was developed over approximately **2-3 weeks** through iterative improvements and problem-solving.

Development Process & Key Challenges

Phase 1: Backend Setup & Basic UI (1-2 Weeks)

The development started with the backend, setting up API requests for retrieving song and lyrics data. Initially, **Musixmatch** was used, but due to limitations, it was later replaced with **Genius** and finally **LyricsOvh**, which proved to be the most reliable option. Once API integration was stable, the **HTML structure** for the **Home**, **Search**, and **Lyrics** pages was created, providing a clean foundation without JavaScript or CSS.

An early version of the **playlist feature** was added, though it lacked persistence and reset upon refresh. The UI was then refined with **CSS styling** and a **logo** to establish a visual identity.

Phase 2: Enhancing UI, API Handling & Bug Fixes (2-3 Weeks)

To improve user experience, **album covers** and **direct Spotify links** were integrated via the **Spotify API**. The backend was optimized by restructuring API requests and introducing caching, reducing response times.

Pagination was also enhanced, increasing results from **20 to 50 per page**, and input sanitization was implemented to handle special characters in search queries. Additionally, structured **error handling** was introduced to prevent API failures and improve stability.

Phase 3: Advanced Features & Code Optimization (4-5 weeks)

With the foundation in place, filters were introduced to allow refined searches, requiring backend modifications to handle multiple parameters efficiently. To improve code maintainability, main.go was split into modular files. CSS was separated from HTML, simplifying future styling changes.

A persistent JSON-based playlist system was implemented, replacing the earlier version. Navigation was restructured to remove reliance on javascript.history.back, preventing user flow issues. The UI was further refined with hover effects, animated buttons, and pop-ups using JavaScript, and a "Copy Lyrics" button was added for convenience. A login button was introduced, with user passwords securely stored in JSON files, along with a registration page to allow new users to sign up.

Phase 4: Final Enhancements & Optimization (6-7 weeks)

The search result limit was expanded from **50 to 900**, significantly improving search capabilities. A **custom error page** and a **FAQ page** were added to assist users.

To enhance interactivity, **animated rotating song cards** were implemented, improving visual engagement. The project file structure was also optimized by separating **CSS and JS**, restructuring the backend, and grouping assets such as images and JSON files for better organization.

Final Deployment & Conclusion (6-7 weeks)

Harmonify evolved through continuous iterations, with each phase enhancing functionality and user experience. Addressing API issues, refining UI elements, and optimizing backend operations resulted in a **smooth, efficient, and feature-rich music search application**.

Key Achievements:

- Seamless **lyrics retrieval** with multiple API integrations.
- **Persistent playlist functionality** with JSON storage.
- **Advanced filtering and pagination** for better user control.
- **Interactive UI enhancements** including animations and pop-ups.
- **Optimized backend** for faster response times and reliability.

The **development journey of Harmonify** was driven by problem-solving and incremental improvements, ensuring a well-rounded and high-performing application. The final product successfully balances **performance, usability, and scalability**, making it a powerful tool for music lovers.