



Title - Sravanam Music Streaming and Editor Software

Micro Project Report Submitted to
Institute of Advance Computing for submission of
Bachelor of Technology degree with
specialization in
Cyber Security & Forensic
2023-2024

Supervised by

Dr. Hare Ram Shah

Submitted by

1. Yash Jain (21ADV3CSF0023)
2. Name: Srajal Tadge (21ADV3CSF0020)
3. Khushi Rai (21ADV3CSF0013)

(Software Requirement Specification Document)

Abstract - To create a Spotify clone using HTML, CSS, Java, and an editor, start by structuring the app with HTML, including sections for header, main content, and footer, then style it with CSS for layout and aesthetics. Integrate JavaScript for dynamic functionality like navigation, search, playback controls, user authentication, and API integration with a backend service. Choose an editor like Visual Studio Code and set up plugins for linting, auto-completion, and version control with Git to streamline development. With this approach, you can efficiently build a responsive and interactive Spotify clone tailored to your design and feature requirements.

Introduction-

A Sravanam Music Streaming and Editor Softwaremusic player is a web-based application that aims to replicate the core features and user experience of the popular music streaming platform. It involves the use of JavaScript, HTML, CSS, VS Code, and Live Server to create an interactive and visually appealing interface, handle audio playback, “Play/Pause,” “Next” and “Previous” buttons.

JavaScript plays a central role in the development process as it enables the implementation of dynamic features and functionality. It is used to handle user interactions, control audio playback. JavaScript frameworks like React or Angular can be used to facilitate efficient component-based development.

HTML is utilized to structure the web pages and define the layout of the Sravanam Music Streaming and Editor Software. It allows for the creation of various elements, such as buttons, input fields, and containers, which provide the necessary structure for the user interface.

CSS is responsible for styling the application and creating an appealing visual experience. It is used to define the colors, fonts, spacing, and overall aesthetic of the Sravanam Music Streaming and Editor Software. CSS frameworks like Bootstrap or Material-UI can be employed to streamline the styling process.

VS Code serves as the primary code editor for development. It provides a range of features and extensions that enhance productivity, including syntax highlighting, code completion, and debugging tools. Its powerful ecosystem allows developers to efficiently write and organize code.

Live Server, on the other hand, is a local development server that allows for real-time testing and previewing of the application. It facilitates a seamless development workflow by automatically refreshing the browser whenever changes are made to the code, providing instant feedback.

To build a Sravanam Music Streaming and Editor Software, one would start by analyzing the other music platform to understand its features, design, and functionality. From there, the application can be divided into various components, such as audio player controls, “Play/Pause,” “Next” and “Previous” buttons.

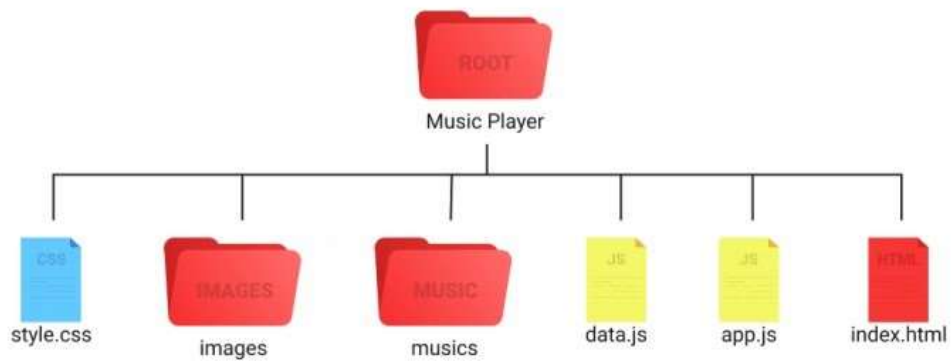
The development process would involve creating the necessary HTML structure for the user interface, applying CSS styles to achieve a visually appealing design, and implementing JavaScript logic to handle user interactions, audio playback functionality.

Testing is crucial to ensure the application functions as expected across different browsers and devices.

Flowchart

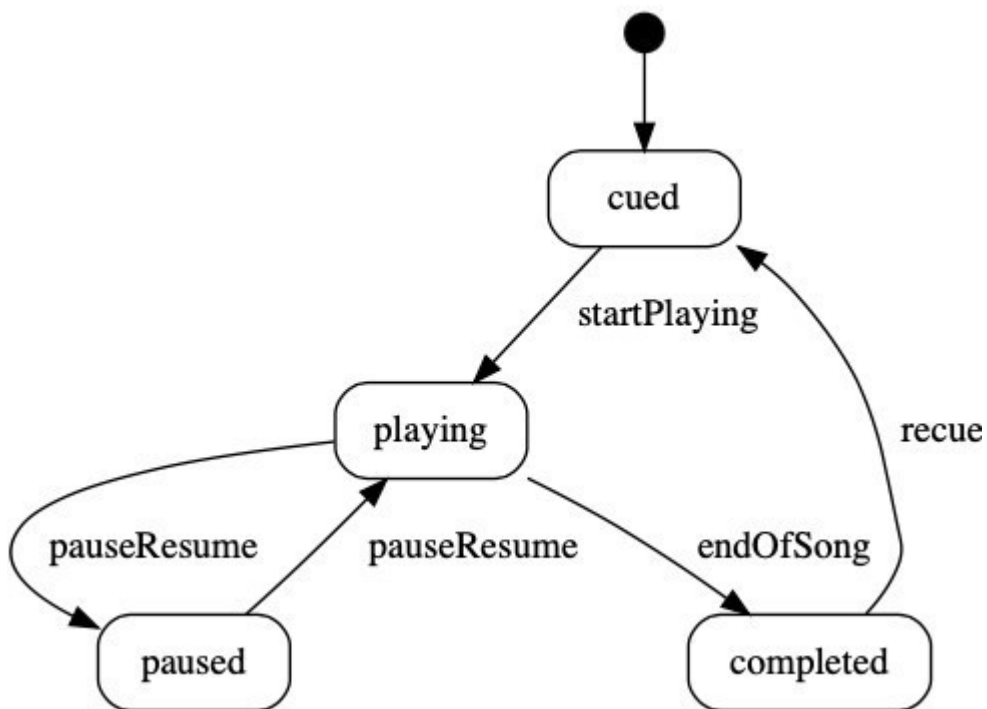
To create a Spotify clone using HTML, CSS, JavaScript, and an editor, begin by designing a flowchart outlining the user interface and interactions.

Start with a main flow indicating the user journey from login/signup to homepage navigation, playlist browsing, song playback, and social features. Branch off to handle alternative paths like search functionality and user profile management.



Each step in the flowchart corresponds to HTML/CSS elements and JavaScript functionality, implemented and styled in the chosen editor.

Iterate on the flowchart as you develop, ensuring seamless transitions and user experiences throughout the Spotify clone's interface.



Creating a Sravanam Music Streaming and Editor Software

Building a Sravanam Music Streaming and Editor Software using JavaScript, HTML, CSS, VS Code, and Live Server involves creating a user-friendly interface, implementing audio playback functionality.

JavaScript is the backbone Sravanam Music Streaming and Editor Software, responsible for handling dynamic aspects like user interactions and data manipulation. It enables features such as play, pause, and skip functionality.

HTML is used to structure the web pages, defining elements like buttons, input fields, and containers that make up the user interface. CSS styles these elements to create an appealing and visually consistent design, including color schemes, fonts, and layouts.

VS Code is utilized as the code editor, offering features like syntax highlighting, code completion, and debugging tools to streamline the development process. Live Server, a local development server, provides real-time testing and previews of the application, ensuring a smooth development experience.

Testing is essential to ensure the functions correctly. Live Server facilitates real-time testing during development, and additional testing frameworks.

In conclusion, building a Sravanam Music Streaming and Editor Software requires a combination of JavaScript, HTML, CSS, VS Code, and Live Server. By replicating core features and developers can create a web-based music player with audio playback functionality, providing users with a familiar and enjoyable music streaming experience.

Software Requirements:

- Windows 11(ultimate, enterprise)
- Visual Studio 17.1.5
- HTML

- CSS
- JavaScript
- Live Server

Hardware Components:

- Processor – RYZEN 7 5000 SERIES
- SSD – 500GB
- Memory – 16GB RAM
- Laptop

Advantages:

- Familiar User Experience: Users will find the interface and functionality of the clone familiar and intuitive. This allows for a seamless transition and easy adoption.
- Customizability: Building the scratch allows for customization and tailoring to specific requirements. Developers have full control over the design, features, and functionalities of the application.
- Flexibility and Scalability: JavaScript, HTML, and CSS provide a flexible development environment, allowing developers to create dynamic and interactive user interfaces. The modular structure of JavaScript frameworks enables scalability, making it easier to add new features and enhance existing ones.
- Cross-platform Compatibility: The web-based nature of the Sravanam Music Streaming and Editor Software ensures compatibility across different operating systems, including Windows, macOS, Linux, and mobile platforms like iOS and Android.
- Learning Opportunity: Building a Sravanam Music Streaming and Editor Software provides an excellent learning opportunity for developers to enhance their skills in web development, JavaScript, HTML, CSS, and related technologies. It allows for hands-on experience with building complex applications and working with APIs.

Disadvantages:

- Complexity: Developing a Sravanam Music Streaming and Editor Software can be a complex undertaking, requiring a deep understanding of the other music platform and its functionalities. Handling audio playback, especially for less experienced developers.

- **Design Consistency:** While efforts can be made to replicate the Sravanam Music Streaming and Editor Software user interface and design, achieving the exact same level of polish and consistency may be challenging. Designing an interface of Sravanam Music Streaming and Editor Software aesthetic and user experience requires careful attention to detail and may require additional design resources.
- **Legal and Copyright Considerations:** Creating a Sravanam Music Streaming and Editor Software raises potential legal and copyright concerns. Using copyrighted music or logos without proper licensing or authorization can result in legal issues. It's essential to understand and comply with copyright laws and licensing requirements to avoid any legal complications.
- **Performance:** Web-based applications can face performance challenges, especially when dealing with large amounts of data or complex interactions. Ensuring smooth audio playback and responsiveness may require optimization techniques and performance tuning.
- **Learning Curve:** Building a Sravanam Music Streaming and Editor Software may involve a steep learning curve, particularly for developers who are new to web development or specific technologies like JavaScript frameworks. Acquiring the necessary skills and knowledge to implement the desired features and functionalities may require additional time and resources.

Applications:

- **Music Streaming Service:** The primary application of the Sravanam Music Streaming and Editor Software is to provide users with a web-based music streaming service.
- **Educational Purposes:** The Sravanam Music Streaming and Editor Software can serve as a valuable educational resource for developers learning web development, JavaScript, HTML, CSS, and related technologies.
- **Portfolio Project:** Developing a Sravanam Music Streaming and Editor Software can be a showcase project for developers to demonstrate their skills and expertise in web development. It can be included in a portfolio to showcase their ability to create interactive web applications with audio playback and data integration.

Product Architecture and Components

The architecture of Sravanam Music Streaming and Editor Software web-based application using JavaScript, HTML, CSS, VS Code, and Live Server consists of several key components. Here are the major components and their roles:

- HTML: Provides the structure and markup of the web pages, including elements like buttons, input fields, and containers.
- CSS: Styles the HTML elements, including layout, colors, fonts, and visual effects, to create an appealing user interface.
- JavaScript: Implements the dynamic and interactive behavior of the application, handling user interactions.

Development Environment:

- VS Code: Serves as the code editor for writing and editing the application's HTML, CSS, and JavaScript code. It provides features like syntax highlighting, code completion, and debugging tools.
- Live Server: Facilitates real-time testing and previewing of the application by hosting it on a local development server. It automatically refreshes the browser when changes are made to the code.

Draw the DFD(data flow diagram) diagram

Data flow Diagram of Creating a Spotify Clone Using HTML, CSS & JavaScript Only