**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 10 March 2025 |
| Team ID | SWTID1741256813147528 |
| Project Name | Rhythmic Tunes |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example:** Rhythmic Tunes

**Reference:** [**https://open.spotify.com/**](https://open.spotify.com/)

## Components & Technologies:

**Frontend Technologies**

* **React.js: A popular JavaScript library for building user interfaces, especially single-page applications.**
* **Redux: For state management, ensuring that your app's state is predictable and easy to debug.**
* **Material-UI: A React component library that implements Google's Material Design, providing a sleek and modern look.**

**Backend Technologies**

* **Node.js: A JavaScript runtime built on Chrome's V8 engine, perfect for building scalable network applications.**
* **Express.js: A minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.**
* **MongoDB: A NoSQL database that stores data in JSON-like documents, making it flexible and scalable.**

**Music-Specific Components**

* **Web Audio API: A high-level JavaScript API for processing and synthesizing audio in web applications.**
* **Tone.js: A framework built on top of the Web Audio API, providing a more musical and higher-level abstraction for creating interactive music in the browser.**
* **MIDI.js: A library that allows you to play MIDI files and manipulate MIDI data in the browser.**

**Additional Tools and Libraries**

* **Socket.io: For real-time, bidirectional communication between web clients and servers, useful for features like live collaboration or real-time updates.**
* **OAuth: For secure authentication, allowing users to log in using their existing accounts from services like Google or Facebook.**
* **Jest: A delightful JavaScript testing framework with a focus on simplicity, ensuring your app is robust and bug-free.**

**Deployment and Hosting**

* **AWS (Amazon Web Services): For scalable cloud computing services, including EC2 for hosting your application and S3 for storing media files.**
* **Heroku: A cloud platform that lets you build, run, and operate applications entirely in the cloud, simplifying deployment and scaling.**

**User Experience Design**

* **Figma: A collaborative interface design tool that allows you to create, share, and test designs with your team.**
* **Adobe XD: A vector-based user experience design tool for web apps and mobile apps, enabling you to create wireframes.**

## Application Characteristics:

**Core Functionality**

* **Music Playback**: High-quality audio playback with support for various file formats (e.g., MP3, WAV, FLAC).
* **Playlist Management**: Users can create, edit, and share playlists with ease.
* **Search & Discovery**: Advanced search capabilities to find songs, artists, albums, and playlists. Recommendations based on user preferences and listening history.

**User Interface**

* **Intuitive Design**: A clean, user-friendly interface that is easy to navigate.
* **Responsive Design**: Ensures the app works seamlessly on various devices and screen sizes.
* **Customizable Themes**: Users can personalize the look and feel of the app with different themes and color schemes.

**Social Features**

* **User Profiles**: Allow users to create profiles and follow others.
* **Activity Feed**: Display recent activities of users' friends, such as new playlists or liked songs.
* **Sharing Options**: Enable users to share their favorite tracks and playlists on social media platforms.

**Advanced Features**

* **Offline Mode**: Allow users to download songs and playlists for offline listening.
* **Lyrics Display**: Show synchronized lyrics while the song is playing.
* **Equalizer**: Provide an equalizer with preset modes and customizable options to enhance audio quality.
* **Audio Visualization**: Offer visual representations of the audio, such as waveforms or spectral displays.

**Performance & Scalability**

* **Low Latency**: Ensure minimal delay in music playback and app interactions.
* **Scalable Infrastructure**: Use cloud services to handle a growing number of users and data efficiently.
* **High Availability**: Implement measures to ensure the app remains accessible and operational at all times.

**Security**

* **User Data Protection**: Secure user data with encryption and comply with privacy regulations.
* **Secure Authentication**: Implement secure login methods, such as OAuth, and support multi-factor authentication.

**Analytics & Monitoring**

* **User Analytics**: Track user behavior and preferences to improve the app's features and recommendations.
* **Performance Monitoring**: Continuously monitor the app's performance and fix issues promptly.

**Monetization**

* **Subscription Model**: Offer premium features through subscription plans.
* **Ad-Supported Model**: Include advertisements for users on the free tier, ensuring they are non-intrusive.

**References:**

[**React.js Documentation**](https://react.dev/)

[**Node js Best Practice**](https://nodejs.org/en/learn/getting-started/introduction-to-nodejs)

[**JSON Web Server Referance**](https://www.npmjs.com/package/json-server)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)