**Groovify - Music Streaming App**

**1. Introduction**

Project Title: Groovify – Music Streaming App

Team Members:

Team Leader: P. Gowshika

G. Heshmika

M. Raja Rajeswari

P. Preetha

S. Swetha

Groovify is a vibrant, feature-rich music streaming web app designed to deliver a personalized and immersive listening experience. Built with React and styled for expressive UI feedback, Groovify combines sleek design with powerful playback controls—making it more than just a player, but a celebration of sound. Whether you're looping your favorite Tamil hits, shuffling pop playlists, or curating your own glowing favorites, Groovify adapts to your rhythm. With Like and Favorite icons, animated transitions, and a responsive layout, every interaction feels premium

**2. Project Overview**

**Purpose:** To deliver a personalized and immersive music experience through a responsive, feature-rich web application. Groovify allows users to explore songs, manage playback, curate favorites, and unlock premium features—all within a glowing, expressive interface.

**Features:**

* Play, pause, loop, and shuffle songs with seamless audio control
* Like and Favorite songs to build personalized playlists
* Loop and shuffle modes for dynamic listening
* Glowing UI feedback for play/pause, favorites, and active song cards
* Search and filter songs by category, artist, or title
* Smart state management for syncing playback across components

**3. Architecture**

**Component Structure:** Groovify is built using modular React components that handle playback, user interaction, and UI rendering. Key components include:

**SongCard**: Displays individual songs with play/pause, Like, and Favorite controls

**NowPlaying**: Shows the currently playing track with progress, volume, and playback controls

**Sidebar**: Handles navigation between Home, Liked, and Favorites tabs

**Homepage**: Serves as the welcome screen before playback begins

**State Management:** Groovify uses both local state (via useState) and global state (via useRef, useEffect, and optional context) to manage:

Playback status and active song

Liked and favorited songs

Loop and shuffle modes

State is synced across components to ensure consistent playback behavior and UI updates.

**Routing:** Client-side routing is implemented using React Router (or similar) to navigate between:

**/home:** Main dashboard with category tabs and song cards

**/liked:** Displays all liked songs

**/favorites:** Shows favorited tracks

Routing ensures seamless transitions and preserves playback state across views.

# 4. Setup Instructions

**Prerequisites:**

* Windows or macOS operating system
* Node.js LTS version installed (https://nodejs.org/en/download/)
* Internet connection for dependency downloads
* Code editor such as Visual Studio Code
* Modern web browser (Chrome, Firefox, Edge, Safari)

**Installation**:

1. Download and install Node.js LTS version suitable for your OS.
2. Download the project zip folder from the source provided.
3. Extract and open the folder in VS Code.
4. Open the terminal in VS Code and run `npm install` to install dependencies.

**5. Folder Structure**

**Client:** Contains all frontend code including reusable components, page views, styles, and playback logic. This is the heart of Groovify’s expressive and responsive user interface.

**Key folders:**

**components/** – UI building blocks like SongCard, NowPlaying, Sidebar, GlowingButton, and AudioControls

**pages/** – Route-level views such as Homepage and Dashboard

**styles/** – Modular CSS files for layout, transitions, and glowing effects

**assets/** – Icons, fallback images, and category visuals

**Utilities:** Includes helper functions and modules that support playback, filtering, and user experience enhancements.

# 6. Running the Application

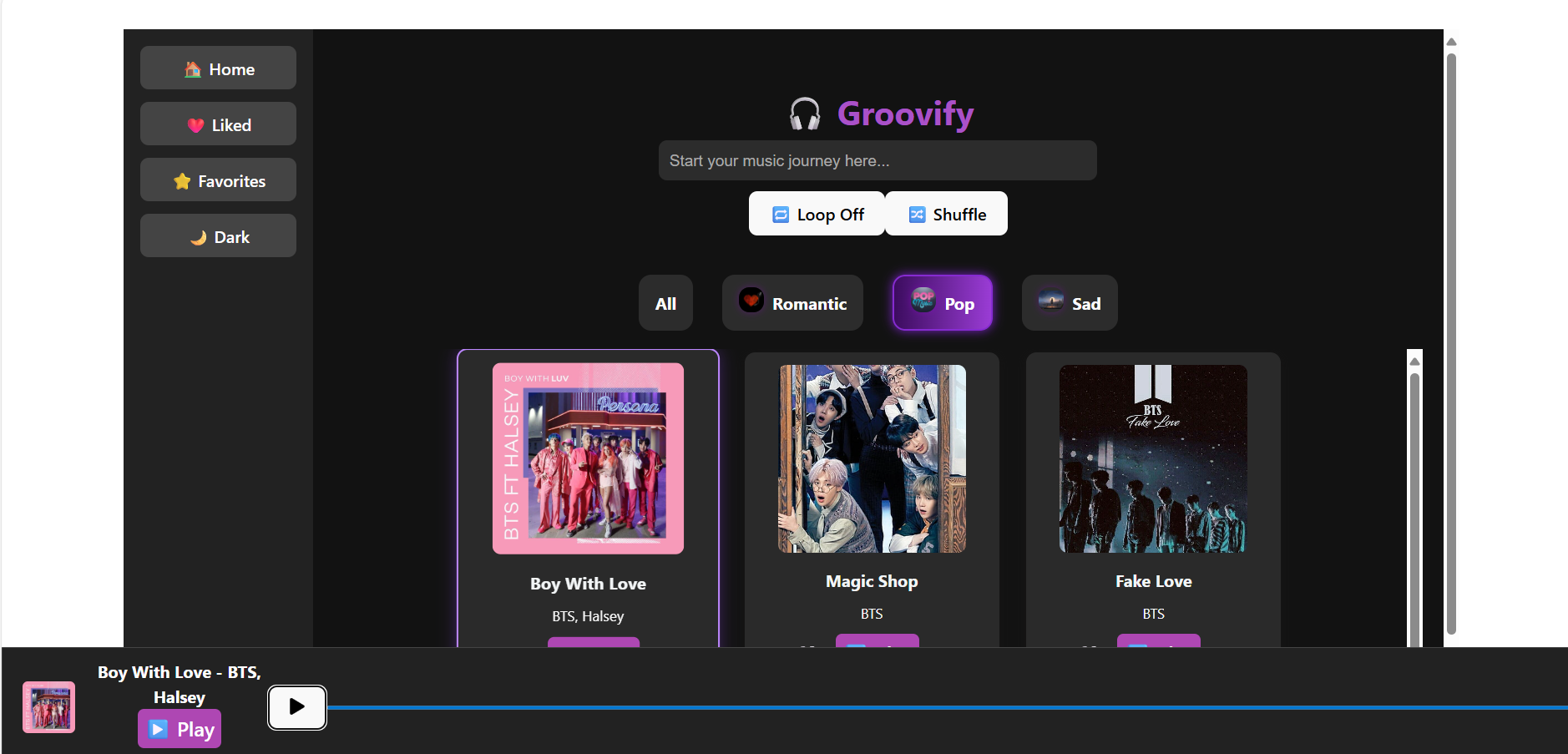
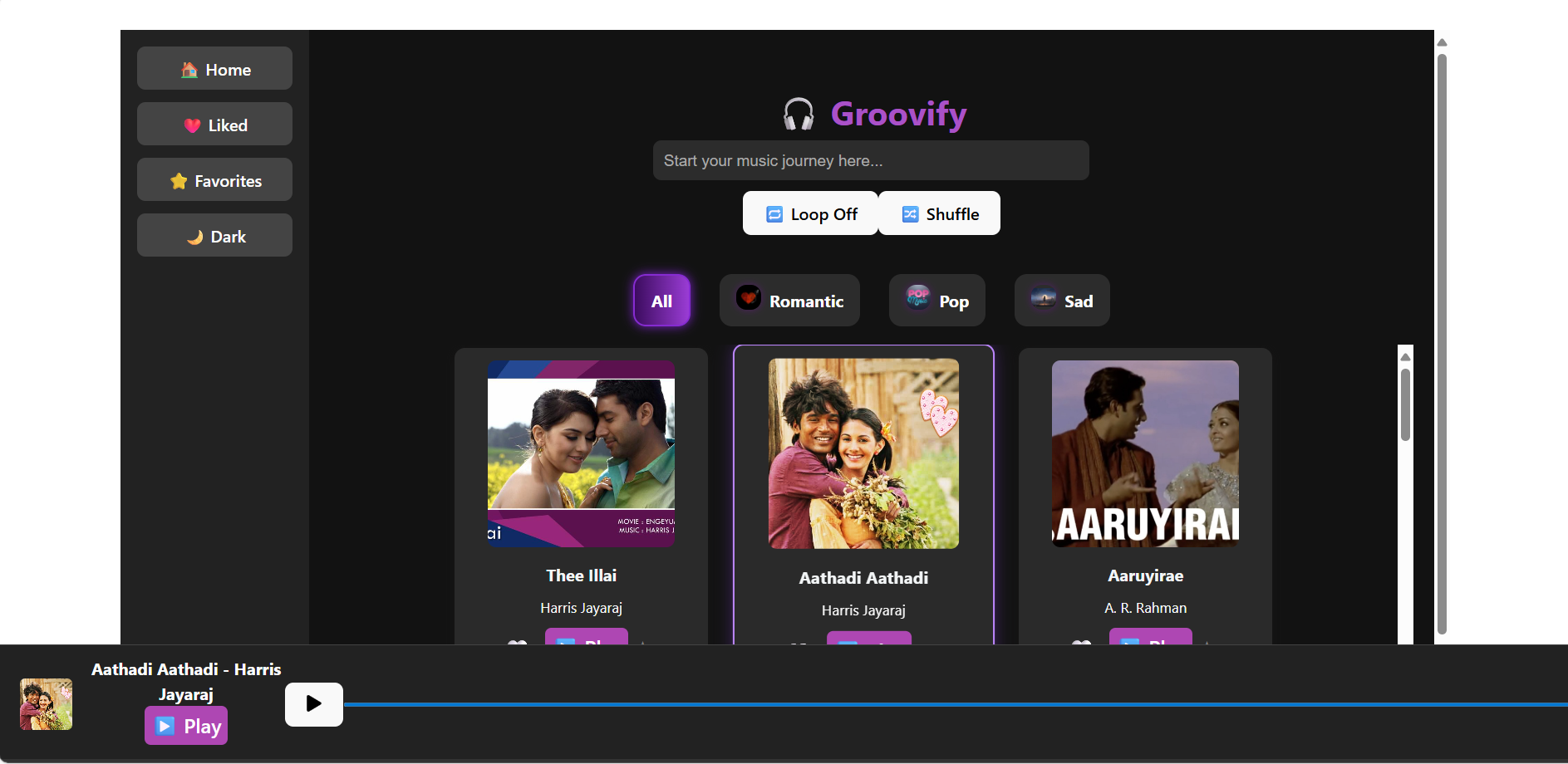
Frontend: After dependencies are installed, start the application by running the command `npm run dev`. This will open the app in the default browser at http://localhost:5173.

**7. Component Documentation**

**Key Components:** SongCard Displays individual songs with title, artist, image, and playback controls. Includes Like and Favorite icons with glowing feedback. Handles local audio playback and syncs with global state.

**8. User Interface:** Groovify features a responsive and interactive user interface designed to deliver a seamless music experience across desktop, tablet, and mobile devices. The layout adapts fluidly to screen sizes, ensuring intuitive navigation and consistent playback behavior.

# 9. Screenshots or Demo



**10. Future Enhancements**

Advanced user authentication and profile customization Enable users to create detailed profiles, and manage preferences.

Multi-language and regional music support Expand the app to support Tamil, Korean, and other regional languages with localized UI and curated playlists.

Theme builder and UI customization Let users personalize their experience with custom gradients, glowing borders, and animated transitions.

AI-powered playlist generation Use machine learning to auto-generate playlists based on mood, tempo.

**Conclusion**

**Groovify** is a powerful and expressive music streaming application designed to deliver a seamless, personalized listening experience. With intuitive playback controls, glowing UI feedback, and real-time synchronization across components, Groovify transforms the way users interact with their favorite songs.

From looping Tamil hits to curating glowing favorites, Groovify empowers users to explore, organize, and enjoy music effortlessly. Its modular architecture, responsive design, and

premium-ready features make it a scalable solution for both casual listeners and music enthusiasts.

As Groovify continues to evolve—with authentication, premium access, and mobile support—it stands as a bold, creative platform that celebrates sound, design, and user expression.