Project Report - HarmonyHub: Music Playlist Manager

1. Project Idea

HarmonyHub is a full-stack web application that allows users to create, manage, and enjoy personalized music playlists. Users can register, log in, create playlists, add or remove songs, and explore curated recommendations. The system is designed with a rich and interactive frontend using React and a robust backend using Node.js, Express, and MongoDB.

2. Project Description

The goal of HarmonyHub is to simplify the experience of organizing and playing music through a user-friendly web interface. Each user can create their own playlists, add songs, and manage them dynamically. The frontend is developed using React for dynamic rendering and smooth navigation across pages. The backend provides RESTful APIs for authentication and playlist management, powered by Node.js and Express, with MongoDB serving as the database to store user and playlist data.

This project is divided into two main components:
1. Frontend – Developed using HTML, CSS, JavaScript, and React Framework.
2. Backend – Built using Node.js, Express APIs, and MongoDB for database management.

The project includes 10 pages: Home, Login, Register, Dashboard, Create Playlist, View Playlist, Add Song, About Us, Contact Us, and Team. The backend supports functionalities such as user authentication and playlist CRUD operations.

3. Website Flow (Wireframe)

Below is the logical flow of the HarmonyHub web application:

1. **Landing Page (/)**
   Guide users to log in or sign up.

- **Buttons:** Login → /login, Signup → /signup
- **Footer:** About • Contact • Privacy Policy

2. **Dashboard (/dashboard)**
   Main hub: view and manage playlists.

- **Navbar:** Logo | Search | Logout
- **Sidebar:** Create Playlist (+), My Playlists, Liked Songs, Recently Played
- **Main:** Playlist cards (cover, title, song count, date), "+ New Playlist"

3. **Playlist View (/playlist/:id)**
   Manage songs in a playlist.

- **Header:** Cover, name, edit, delete
- **Song list:** # | Title | Artist | Album | Duration | Actions
- **Controls:** Play All | Shuffle | Add Songs (+)

4. **Song Search (/search)**
   Find and add songs.

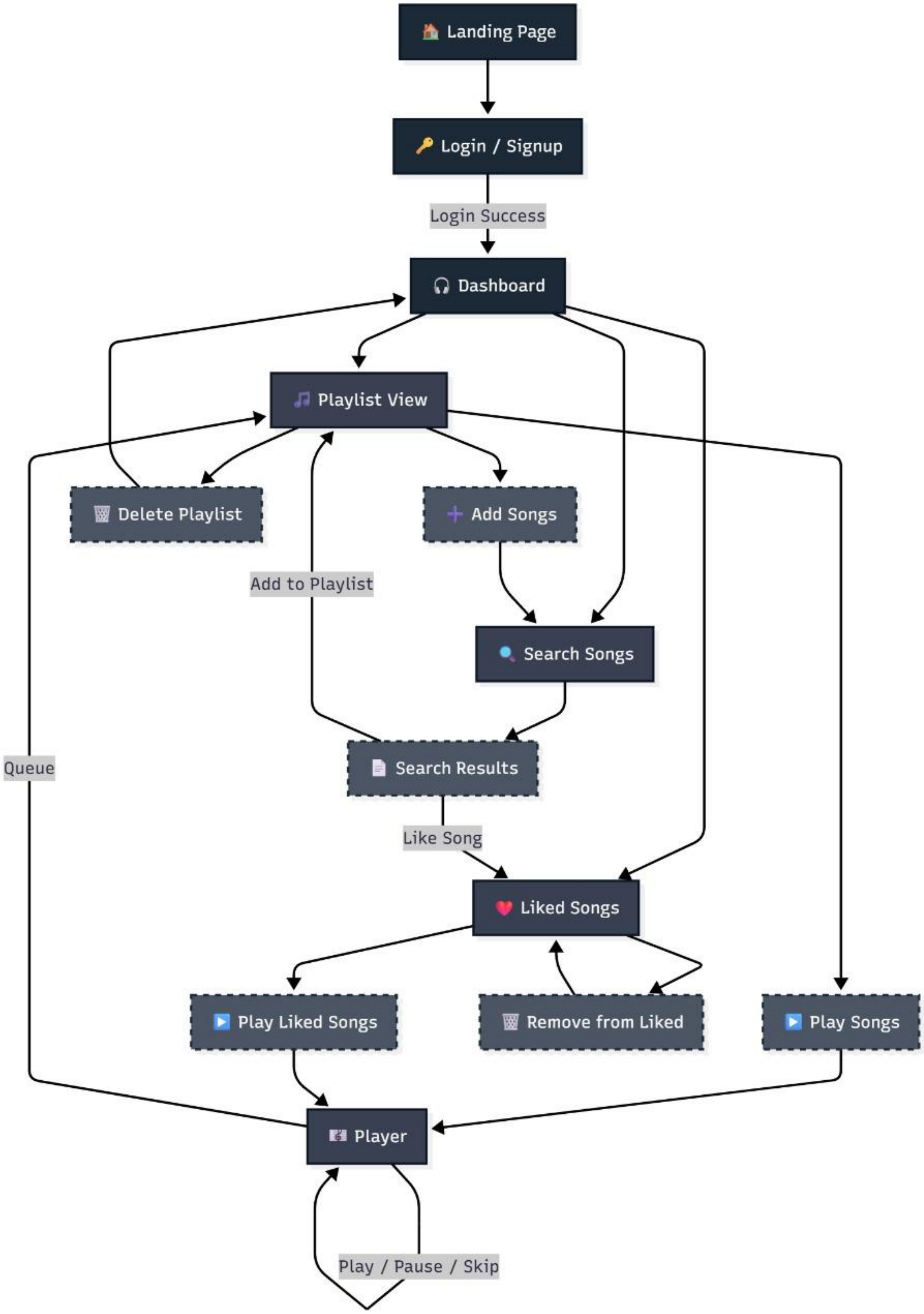
- Search + filters (genre, artist, album)
- **Results:** Song info, Add to Playlist, Like button

5. **Player (Persistent)**
   Visible on all pages.

- Thumbnail, title, artist
- **Controls:** Rewind | Play/Pause | Forward | Shuffle | Loop
- Progress bar, Volume, Queue

6. **Liked Songs (/liked)**
   View all liked songs.

- Filter/Sort, Play all, Remove


🏠 Landing Page
🔑 Login / Signup
Login Success
🎧 Dashboard
🎵 Playlist View
🗑 Delete Playlist
➕ Add Songs
Add to Playlist
🔍 Search Songs
Queue
📄 Search Results
Like Song
❤️ Liked Songs
▶️ Play Liked Songs
🗑 Remove from Liked
▶️ Play Songs
🎚 Player
Play / Pause / Skip

4. Tools and Technologies

Frontend: HTML, CSS, JavaScript, React
Backend: Node.js, Express.js
Database: MongoDB
Other Tools: Postman, VS Code, GitHub

5. Team Details

Developed by: Shreesh Athreya, Swaroop Venkateshwar, Shashank Vinayak