

# Final Project Technical Implementation Plan

- Group name: Web Warriors (Group 3)
- Group members
  - Tharun Achanta (20021462)
  - Bhavin Pragji Patanwadia (20012831)
  - Dhanvin Patel (20016911)
  - Marco Salazar Torres (10471842)
  - Kiko Ferrer (20013384)
- GitHub Repo: <https://github.com/fferrer89/streaming-service>
- The function and general overview of your project; give 1-2 paragraphs on the overall vision
  - Music Streaming Platform
  - Description: In this project we are going to develop a music streaming platform like spotify. It will include features such as user profiles (user and admin), and components such as songs, libraries, playlist creation, liking of songs and playlists, etc. The platform will also provide search and recommendations functionality. Admin users will have the capability to add songs as well as analytics dashboards.
  - Features:
    - Profiles
      - Registration and Login
        - User
        - Admin (uploads and removes songs)
      - Like songs
      - Listening History
    - Playlists
      - Users can create and edit their own playlists
      - Playlists visibility: Private and Public
      - Users can like playlists
    - Library with songs
      - Add a song
      - Remove a song
    - Song:
      - Skip - forward and back
      - Lyrics

- Search and Recommendations
    - Searching for songs, albums, playlists, other user, lyrics
    - Recommendation songs based on previously played songs and liked songs - Personalized recommendations
  - Dashboard with analytics in Admin profile
    - Most listened songs (trending songs)
    - Number of users over time
- Each of your three course technologies you intend on using, as well as what their uses will be
  - Next JS
    - Used for the application structure and server-side routing. Next will handle the pre-rendering of files. We will also use React for client-side rendering of the different components of the application.
  - Redis
    - Used to cache application data and songs for fast music downloads
  - GraphQL
    - API layer to retrieve and add metadata from the front end (songs, profile info, playlists, search results)
  - MongoDB
    - Database for songs, profile info, playlists, search results. Etc
- The two independent technologies, a brief description of each, as well as what their uses will be
  - OAuth 2.0 or/and OpenID Connect (OIDC): A modern authentication and authorization protocol.
    - We plan to use the OIDS to verify our user's identity (authentication) in addition to authorizing the users to protected resources via JSON Web Token. We will also implement an email registering method, so only users with validated emails will be registered.
  - Amazon S3 database to store and retrieve song files
    - Using HLS to fast download of songs
  - Deploying the application and services to the cloud
    - MongoDB - Atlas
    - Next JS - Vercel
    - GraphQL - Heroku