

Project Description

The Mixtape Web application is a modern digital platform that revives the nostalgic concept of creating personalized music playlists, inspired by the classic mixtape. It is designed with a vintage aesthetic and allows users to compile playlists using Youtube video links, offering an easy to use interface for adding, organizing and playing tracks. Key features include secure user authentication, a dynamic mixtape creator supporting up to ten tracks , and an embedded YouTube player with custom playback controls. The system automatically extracts video metadata and generates mixtape descriptions for a personalized touch.

Features of this application include

- **User registration and login** - secure authentication with password hashing
- **Session Management** to maintain user state across pages
- **Mixtape Creation** - adding up to 10 YouTube tracks with automated metadata extraction
- **Playback Functionality** - a custom player with play/pause, skip, and progress tracking is displayed
- **Playlist Management** - for saving, retrieving, and displaying past mixtapes
- **UI Theme Switcher** - dark and light modes themes allow the user to choose their desired setting

System Architecture

1.) Frontend (Client Side)

- **HTML5, CSS3, and JavaScript**
- **YouTube iFrame API** for embedded video playback
- **Dynamic UI** for vintage theme
- **Dark/Light Mode toggle**

2.) Backend (Server Side)

- **PHP** for handling HTTP requests and database logic
- **MySQL Database** for storing user accounts, playlists, and track data

3.) Database Structure

- **User Table** to store user credentials (user_id, email, password)
- **Playlists Tables** contains mixtape details (playlist_id, user_id, title, creation_date)
- **Playlists songs table** maps tracks to playlists (Playlist_id, song_id, position)
- **Songs table** caches video metadata (video_id, title, duration, url)

This application is built using a combination of frontend technologies such as HTML, CSS, and JavaScript and PHP backend with MySQL database support. The frontend features a responsive design with a retro theme, while the backend handles user authentication, playlist management, and YouTube API integration for fetching video data and details. Security measures include password hashing, input validation, and prepared statements to prevent SQL injection.

Website Layout

Account Creation and User Login

This application allows the user to input a username and password combination as long as their information is stored in the database. They have the option to create an account, gathering the required information (name, email, and password) and input the data into the users table in the mixtape_db database for the user to be able to log into at any time.

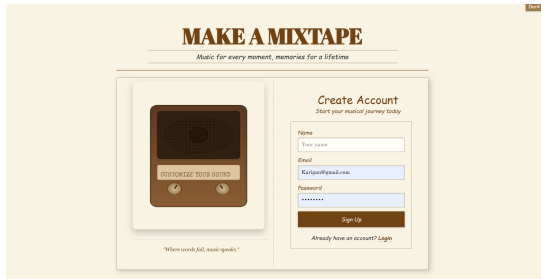


Fig 2. Shows the account creation screen where the user can input their information to create their personal account to create a login which is sent to the database

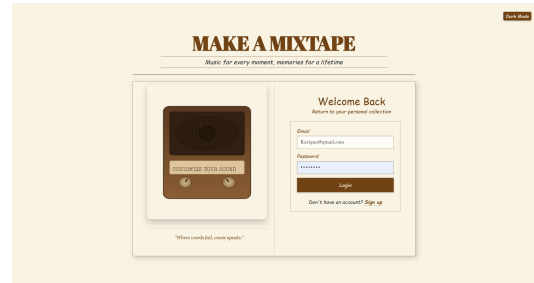


Fig 1. Shows the website application's home page, displaying to the user the login page in which the user can enter a username and password combination to access their tracks

Creating the Mixtape

The website allows for users to create their mixtape name and enter up to 10 youtube links. The user can then hit the create mixtape button and the program will then request the data from the YouTube API. The data is then parsed and used to display the track name and duration. The parsed data is also used to create and store the playlist and playlist song information in the MySQL database.

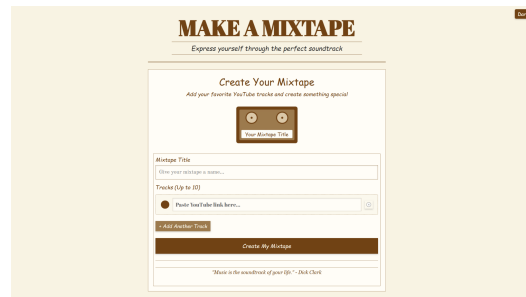


Fig 4. Shows the mixtape creation page requesting the youtube links and the mixtape name

Mixtape Player

The website gathers and parses the video metadata to allow the website to display the videos sound, name, and duration. It sets each of the links as a track and will play through each one in the order it was given. Playback buttons allow the user to rewind, pause, play, and skip through tracks.

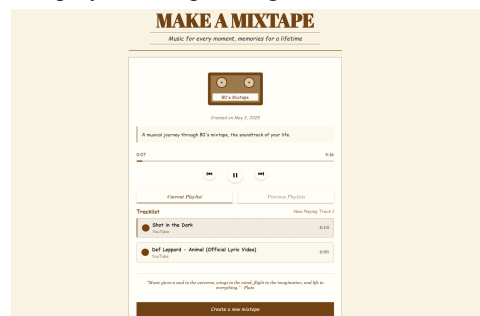


Fig 5 shows the track playing screen. It shows the current tracks the name of the tracks as well as the duration. Playback buttons allow for the user to seamlessly navigate through their mixtape

Previous PlayLists

This page allows the user to access their previously created playlists. Allowing them to effectively navigate between their different mixtapes and the current tracklist being played. The playlists are accessed from the database corresponding to that user's assigned user_id when they made their account. Once the playlists are fetched they are displayed in the previous playlist tab where the user can choose to load the playlist and listen to their track

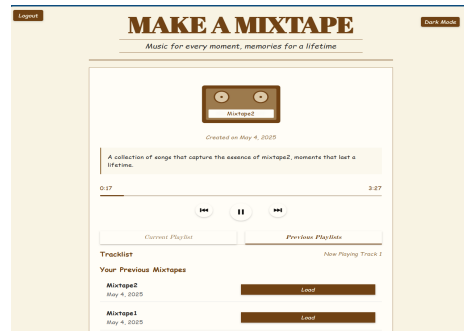


Fig 6 shows the previous mixtapes screen where it displays the users previously created mixtapes and allows them to play them

Individual Contributions

Rahma Seid - Team Leader

Individual contributions include overseeing the project and coordinating progress updates and integration between the front and backend. Designing and managing the MySQL schema and creating relations between the different tables in the database (users=>Playlist=>Songs). Helped develop key backend logic to handle mixtape creation, data storage, and retrieval of user playlists, as well as connecting the backend to the database using PHP.

Fartoon Mohammed - Frontend Developer

Individual tasks included managing front-end development and building responsive HTML/CSS templates for the landing page, login/register page, and mixtape creation page. She added basic interactivity using JavaScript and took care of layout and styling details for login/register page, and mixtape creation page to ensure a smooth user experience. Her contributions also included developing the functionality of a Theme Switcher that allows users to toggle between light and dark modes. Additionally, she set up the logout button functionality, which redirects users back to the login/register page upon logging out.

Aubrey Morgan- Auth & Security Specialist

Tasks included implementing the backend infrastructure to manage user accounts securely. Developing the core PHP scripts for user registration, login, session validation, and logout. These scripts handle password hashing, input validation, session management, and JSON responses to integrate smoothly with the front-end. Managed backend registration logic to ensure a user cannot sign up with an email already present in the database. This work forms the foundation for user authentication and enables personalized access to mixtapes and other user-specific features.

Paroh Ali - Frontend Developer

Contributions include working on frontend development with Fartoon and implementing HTML/CSS/JS into the web application. Also helping to develop the UI theme switcher and integrating it into the site. Paroh also assisted with the UI interactions and ensuring consistent user experience across pages. She created the UI elements for the playback page where users can listen to their mixtapes complete with restart, skip, play, and pause buttons for the user to navigate their tracklist as well as implemented the YouTube API system to retrieve the video metadata to be parsed and used throughout the program.

Sitara Elias-Perez - Backend Developer

Tasks included setting up the PHP server-side logic. As well as handling route requests such as the user login, song upload, and song list retrieval. Sitara also handled the hashed password authentication in the database as well as connecting the users song info requested from the YouTube API to the song table in the mixtape_db database

Karigan Stewart - Report & Analytics Lead

Tasks included setting up and showing users their previous playlists. Connecting the playlist songs and playlists to the database and handling the frontend component of the playlist history allowing the user to pick previous playlists and load the music in the player. As well as created the Project Design Diagram and collected visuals and wrote the final project report.

Project Diagram

