

Universidad Distrital Francisco José de Caldas



**UNIVERSIDAD DISTRICTAL
FRANCISCO JOSÉ DE CALDAS**

Technical report MusiCat

Advanced Programming

Tito Alejandro Burbano Plazas

Bogotá D.C.

1. Project Definition

The project consists of the development of monolithic software for a music streaming website. In terms of technical decisions, the project was limited to the processes of streaming music, creating and managing playlists.

It is a business model focused on providing streaming music services to users over the Internet.

As for the interested parties, we consider that the main interested parties in software with these characteristics are the artists who would publish their music on the site for the reach of many people and all the users who will benefit from being able to play music online.

Now, among the tools necessary to develop this software, and taking into account that it will be a monolith, it was decided to use Python for the development of the backend together with the FastAPI framework. For the graphical interface it was decided to use HTML, CSS, JavaScript and the Django framework. Finally, PostgreSQL will be used for the database, due to its reputation for reliability and scalability in production environments.

2. User stories

As a user, I want a platform where I can listen to a wide variety of songs so I can enjoy my favorite music at any time.

As a user, I want to be able to see a search bar on the home page so I can easily find my favorite songs or artists.

As a user, I want to have the option to log in or register if I don't have an account, so I can save my preferences and playlists.

As a user, I want to create an account to save my favorite songs and playlists, so I can access them easily in the future.

As a user I want to access the main page and be able to select the song I want to listen to, so I can listen it.

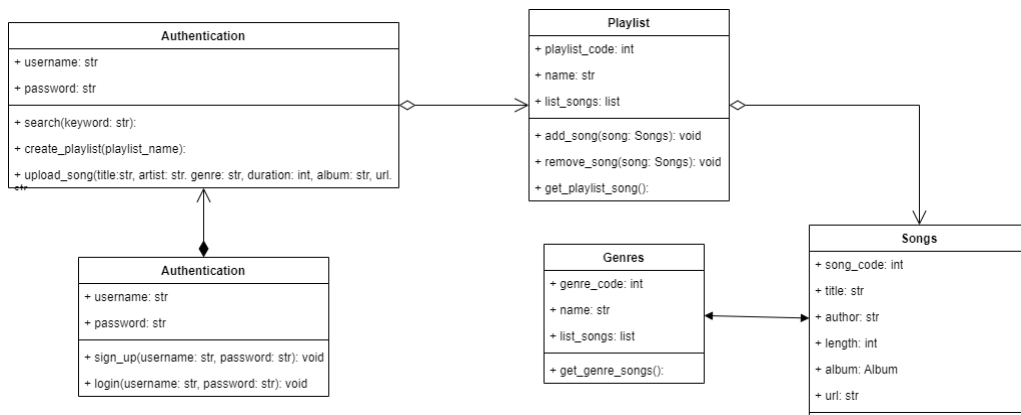
As a user I want to see the songs in the main interface, with name, artist and genre and thus select the one I want to listen to, so I can select one.

As a user, I want to be able to create and manage my own playlists, adding or removing songs as I want, so I can maintain my tastes.

As a user, I would like to have a play button next to each song so I can start listening to it immediately.

As a user I want to be able to upload new songs to the platform, providing details such as title, artist, genre, duration, album and URL, so I can add my own songs.

3. Class Diagram



4. About the classes in the backend

- Authentication: This class receives 'username' and 'password' as parameters, allowing the user to enter the system or register to access the system.
- User: This class works as a representation of a MusiCat user, where any user can; search and upload songs to the platform, you can create and delete playlists, and in turn add songs to them.
- Song: This class is the representation of what a song is within MusiCat.
- Genre: This class is the representation of a musical genre, where the user can navigate through various musical genres and see which songs are within them.
- Playlist: This class represents the operation of a playlist within the website, where the user can add or delete songs to it, and also browse the songs contained.

5. Web services

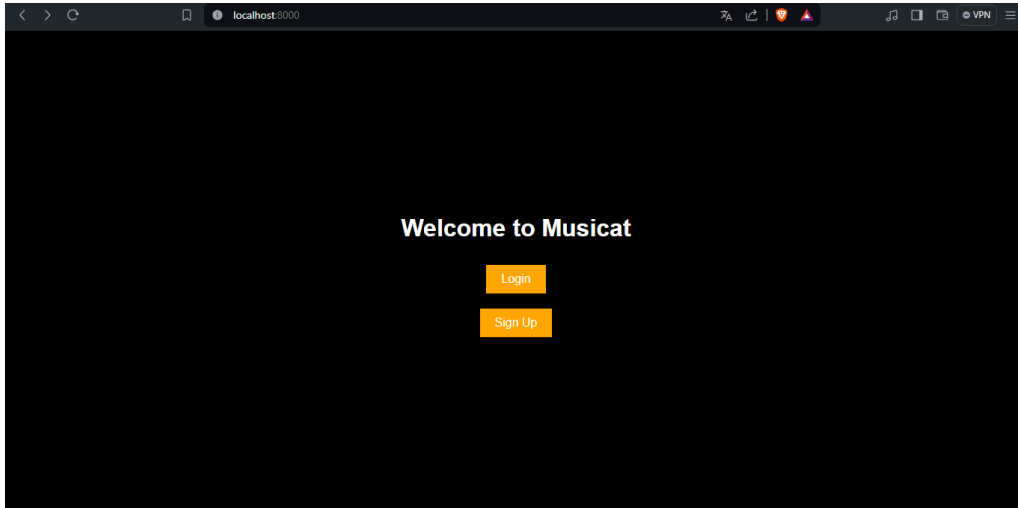
This is a list of the site's web services, each one calls a method of a certain class, you can see the documentation for these within the code.

- `@app.get("/genre/{genre_name}/songs")`
- `@app.post("/playlist/{playlist_name}/add-song")`
- `@app.delete("/playlist/{playlist_name}/remove-song")`
- `@app.get("/playlist/{playlist_name}/songs")`
- `@app.post("/register")`
- `@app.post("/login")`
- `@app.get("/search")`
- `@app.post("/create-playlist")`

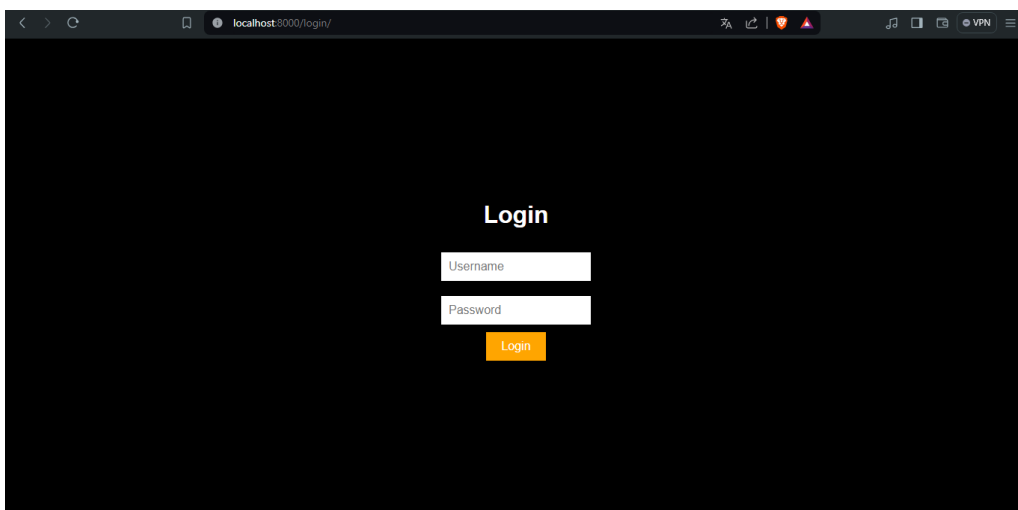
- `@app.post("/add-song-to-playlist")`
- `@app.post("/upload-song")`

6. Django URLs and Mock ups for the site

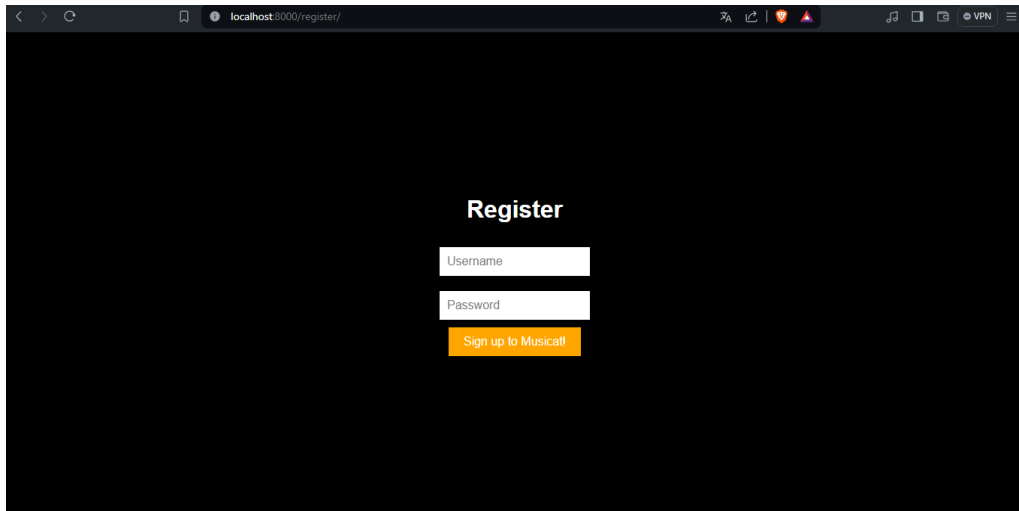
- `path("", views.index, name='index')`: It is the main site, where the user chooses if they want to log in to the site, or register.



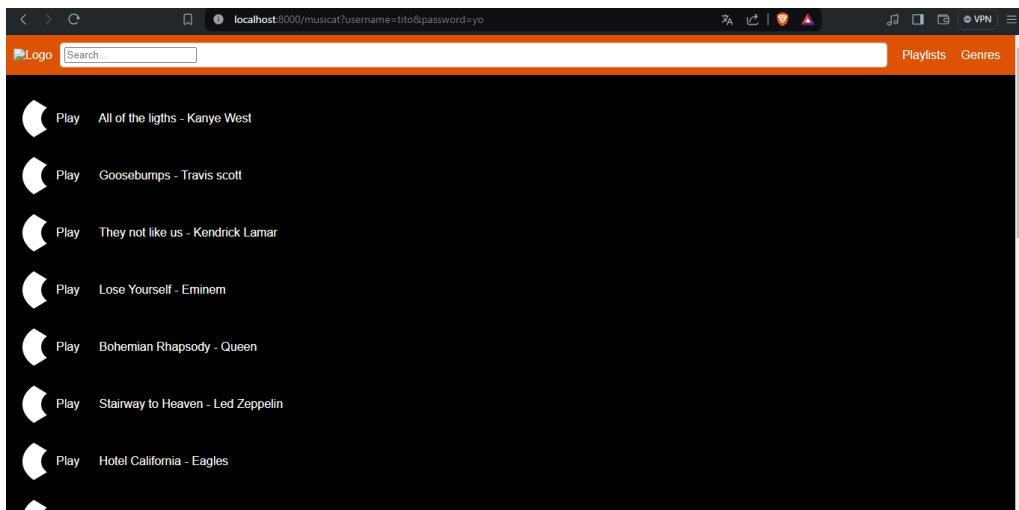
- `path('login/', views.login, name='login')`: On this site the user enters a data validation, where if they are correct they would go to the home page.

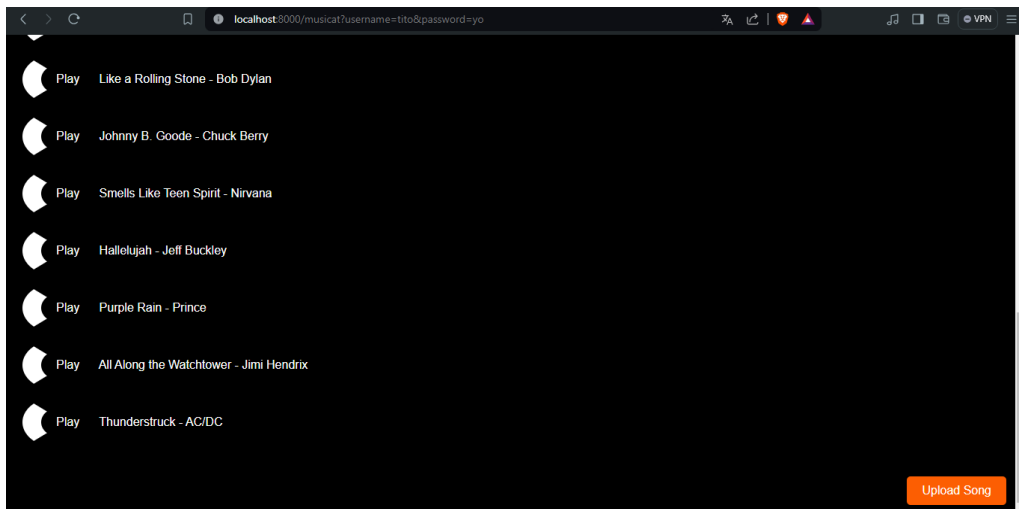


- `path('register/', views.register, name='register')`: On this page the user registers by giving their data, when doing so the user is sent to the Login page.

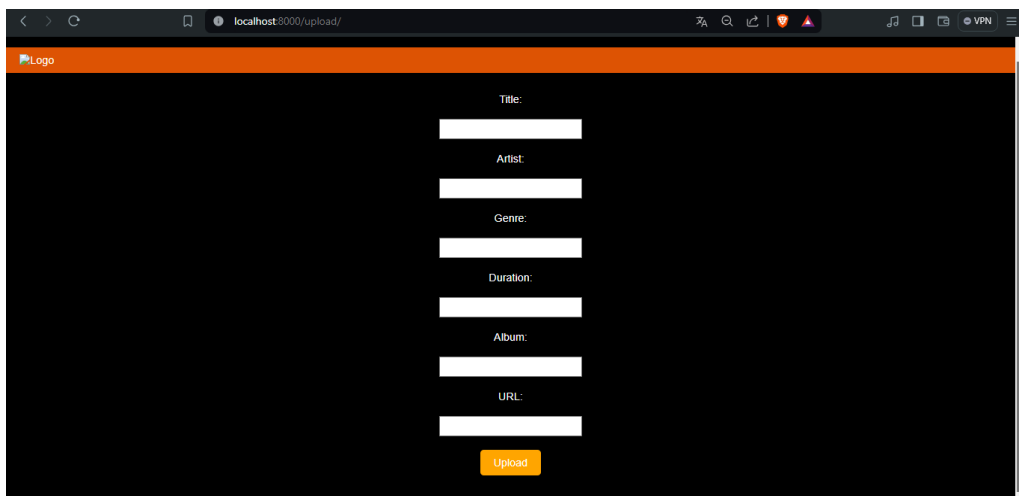


- `path('musicat', views.musicat, name='musicat')`: It is the main page, where the user can choose the action they are going to perform, within of course those offered by MusiCat.

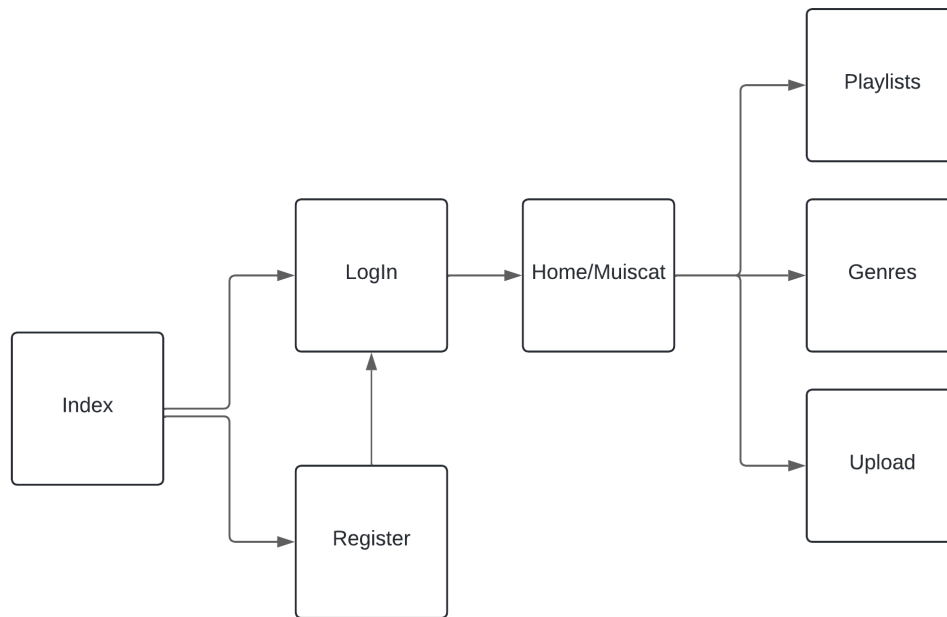




- `path('upload/', views.upload, name='upload')`: On this page the user can upload a song.



7. Navigation Map



8. Thanks for reading it completely!