



DOS MUSIC

Lorenzo Stigliano

136174

Progetto di Tecnologie Web 2021

SOMMARIO

Introduzione	2
Glossario	4
Obiettivi del sito	5
Descrizione del progetto	6
Utenti.....	6
Database e Modelli.....	7
User.....	7
Post	7
Feedbacks (commenti, reports, likes)	7
Schema Completo delle tabelle utilizzate	8
Utilizzo dell'Applicazione.....	10
Tecnologie utilizzate e Scelte di Sviluppo (le scelte di sviluppo sono da inserire)	11
Backend	11
Comunicazione	11
Frontend	11
Organizzazione del progetto	12
Test Svolti	13
Risultati.....	14
Istruzione per l'installazione.....	23
Problemi Riscontrati	23

INTRODUZIONE

La storia di internet e dei servizi di streaming musicale è sempre stata legata e potremmo sancire come una delle prime dati significative di questo legame il 1993. In quell'anno nasce IUMA (Internet Underground Music Archive). L'idea era quello di avere un luogo in cui artisti indipendenti potessero caricare le proprie tracce e condividerle senza dover passare attraverso alcuna etichetta musicale.



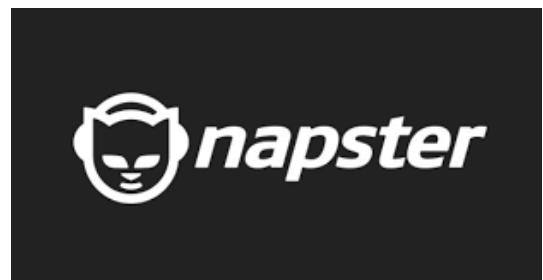
Le canzoni che venivano caricate su IUMA si potevano ascoltare e scaricare direttamente dal sito (funzionalità implementata post release); da notare come con i servizi di streaming odierni sono rare le volte in cui scarichiamo la musica in locale. Dopo qualche anno IUMA fu acquisita da EMUSIC che ne ampliò le funzionalità.



Sul finire degli anni 90 si è affacciato sul mercato un nuovo programma: Napster. Il suo successo

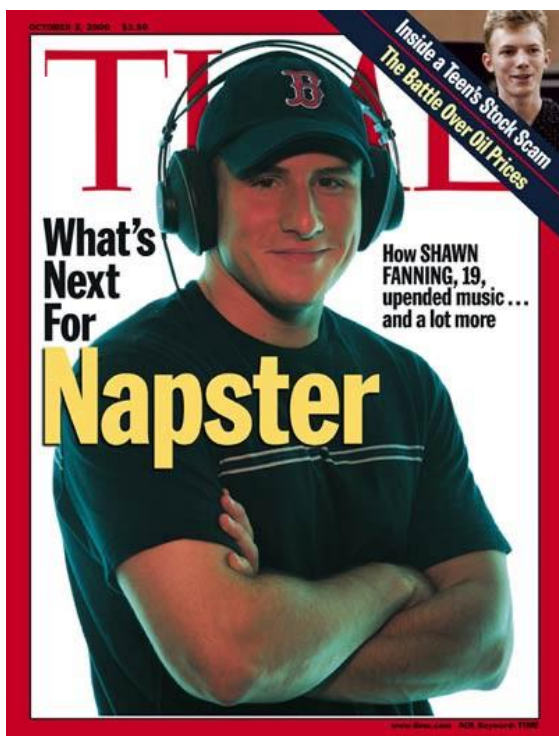
fu travolgente e si basava principalmente su due caratteristiche:

- Condivisione delle canzoni fra utenti
- Formato mp3



La prima caratteristica consentiva letteralmente di condividere la musica con un semplice click del mouse, mentre la seconda permetteva di avere un formato musicale che occupasse poco spazio su disco e fosse quindi anche più veloce e pratico da scaricare.

Napster portava però con sé un problema etico non di poco conto: poter concedere ad utenti qualunque di scaricare e condividere canzoni, album, live session, di artisti etichettati, voleva dire non pagare alcun tipo di diritto d'autore. La situazione scatenò le inimicizie di molti artisti che decisero infine di denunciare Napster ed i suoi creatori Sean Parker e Shawn Fanning.

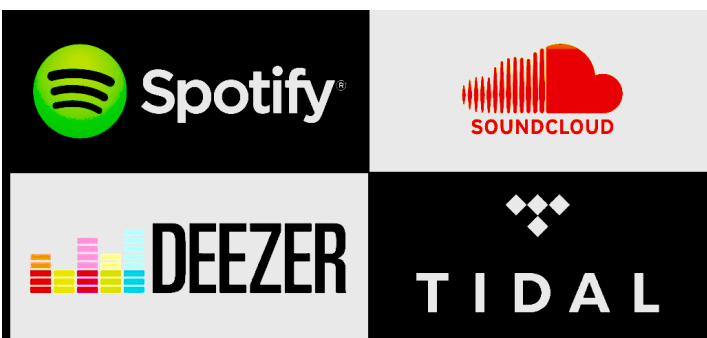


L'onda di Napster venne interrotta l'11 luglio del 2001, a causa delle varie cause legali che ne portarono alla chiusura. In realtà, a seguito di questa prima chiusura, si cercò di trasformare la formula iniziale in un servizio a pagamento, ma gli accordi con le major non andarono per il verso giusto e nel 2002 il progetto venne abbandonato.

Il testimone di Napster venne presto passato ad Apple, che nel 2003 lanciò iTunes: libreria musicale online che permetteva di scaricare canzoni sui dispositivi Apple al prezzo di 0,99 cent/canzone.

La grande svolta a seguito dell'esperimento di iTunes è stata quella dei servizi di streaming su abbonamento: Spotify, Tidal, Deezer ...

Questi si basano sul concetto di avere un canone mensile, ma danno modo di usufruire di una libreria musicale pressoché infinita e compresa anche di Podcast e Live Radio di ogni genere e sorta. Uno dei problemi di questi servizi è che,



come nel caso di Spotify, sono spesso tecnicamente in perdita: per quanto diventino de facto un aspetto importante per la vita di moltissime persone, faticano in realtà a generare revenues in maniera sufficiente, giacché i costi di mantenimento e di diritti con le major spesso superano di gran lunga i ricavi degli abbonamenti mensili.

Il caso di Spotify è interessante perché per quanto il servizio sia tecnicamente in perdita costante, è diventato così popolare che l'idea di chiudere non è attualmente contemplata.

DOS Music si pone quindi all'interno di questo scenario e si propone come un'alternativa a servizi Freemium, per la condivisione delle tracce musicali di artisti senza etichetta (indipendenti). Lo stile del sito si ispira in parte alla cultura Vaporwave, che prende a piene mani dall'immaginario pop degli anni 80 e 90. Il sito appare quindi come sistema operativo DOS (in particolare a MS DOS 3.10) con cui poter interagire.

Dalle animazioni, allo stile grafico, alla palette colori super saturata applicata alle immagini caricate sul sito, ... l'obiettivo è stato cercare di ricreare quel tipo di esperienza.

Tale stile grafico contestualizza il genere musicale preponderante all'interno del sito: lofi, declinato e contaminato da moltissimi altri generi. Questo genere musicale mi accompagna dall'inizio della pandemia in tutte le mie giornate davanti ad un video terminale e con questa scelta di tema volevo rendergli omaggio.

GLOSSARIO

<u>Termine</u>	<u>Sinonimi</u>
Utente	User, Iscritto
Post	Song, Canzone, Contenuto Multimediale
Genere Musicale	Categoria, Tag, Genere
Commento/Like (mi piace)	Feedback, Riscontro
Modulo di iscrizione/creazione post/creazione commento	Form

OBIETTIVI DEL SITO

1. Upload di contenuti musicali in formato musicale vario (mp3, wav, ...) ed editing post caricamento da parte dell'utente creatore
2. Sistema di feedback (likes e commenti)
3. Possibilità di ricerca di contenuti all'interno del sito
4. Streaming dei contenuti multimediali all'interno del portale
5. Creazione di tags nuovi (also known as: generi musicali) da associare alle canzoni caricate sul sito
6. Editing informazioni legate al profilo utente
7. Report dei post

UTENTI

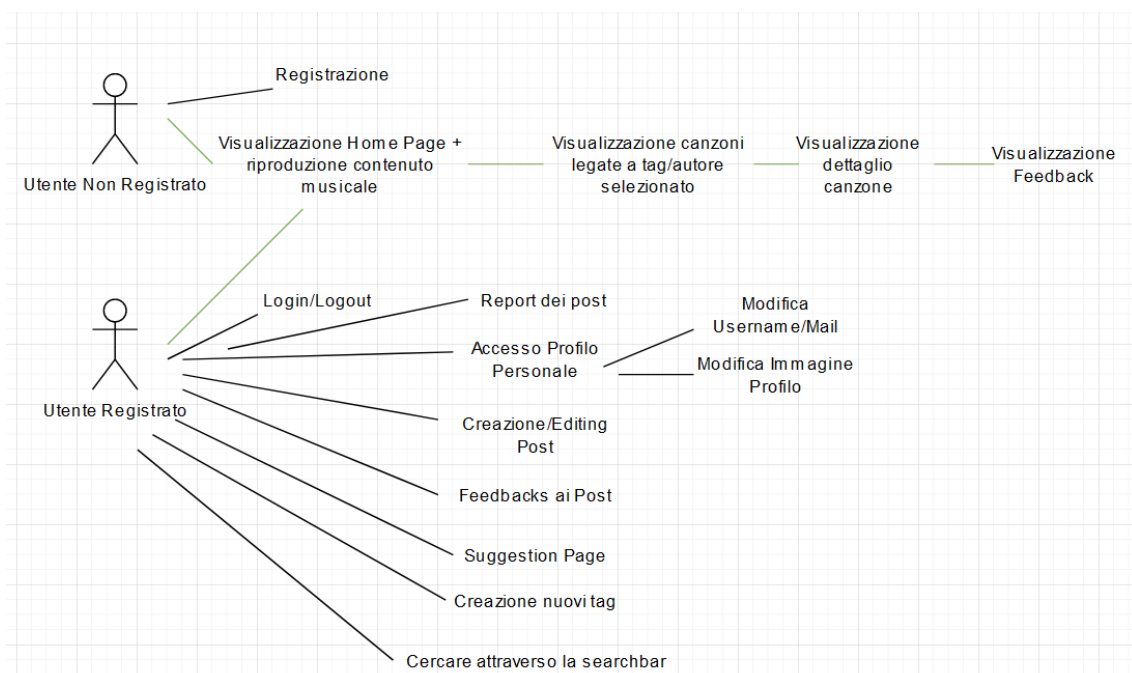
Gli utenti di DOS Music sono divisi in due principali categorie:

1. Utenti non registrati:

- Accedere alla Homepage
- Vedere il dettaglio di un post e relative categorie, autore e feedbacks
- Ascoltare musica direttamente dal sito

2. User, normali utenti della piattaforma, che sono in grado di:

- caricare musica all'interno della piattaforma
- lasciare un feedback
- cercare altri contenuti all'interno del sito
- creare tag nuovi da associare alle canzoni
- segnalare post inappropriati
- editare il loro profilo utente



DATABASE E MODELLI

In quanto database è stato scelto di implementare SQLITE per praticità d'uso.

USER

auth_user	
id	integer
password	varchar(128)
last_login	datetime
is_superuser	bool
username	varchar(150)
last_name	varchar(150)
email	varchar(254)
is_staff	bool
is_active	bool
date_joined	datetime
first_name	varchar(150)

Per il modello user è stato utilizzato quello presente all'interno dell'App Django Auth. Ogni utente può avere solamente un profilo e genere infiniti post, ognuno di questi è posseduto solamente da un utente.

POST

Tale modello contiene tutte le caratteristiche del contenuto musicale che l'utente andrà a caricare su DOS Music: titolo, data, autore, copertina, genere, traccia audio, album, cantante. Post è collegato al modello Comment e Report per cui ogni post può avere infiniti commenti/report ma ognuno di essi è legato solamente ad un singolo post.

blog_post	
id	integer
title	varchar(100)
date_posted	datetime
author_id	integer
header_image	varchar(100)
category	varchar(255)
audio	varchar(100)
album	varchar(100)
singer	varchar(100)

FEEDBACKS (COMMENTI, REPORTS, LIKES)

blog_comment		blog_report	
id	integer	id	integer
author	varchar(200)	author	varchar(200)
text	text	text	text
created_date	datetime	created_date	datetime
approved_comment	bool	approved_report	bool
post_id	integer	post_id	integer

Ecco le tabelle che si occupano del commento/report e funzionalità di feedback tramite like. L'ultima funzionalità citata è gestita come campo molti a molti; questo permette di creare una relazione tra un post e infiniti utenti. La funzione che poi ritorna

effettivamente il numero di likes altro non è che un contatore di "relazioni" che possiede un determinato post con un numero N di user.

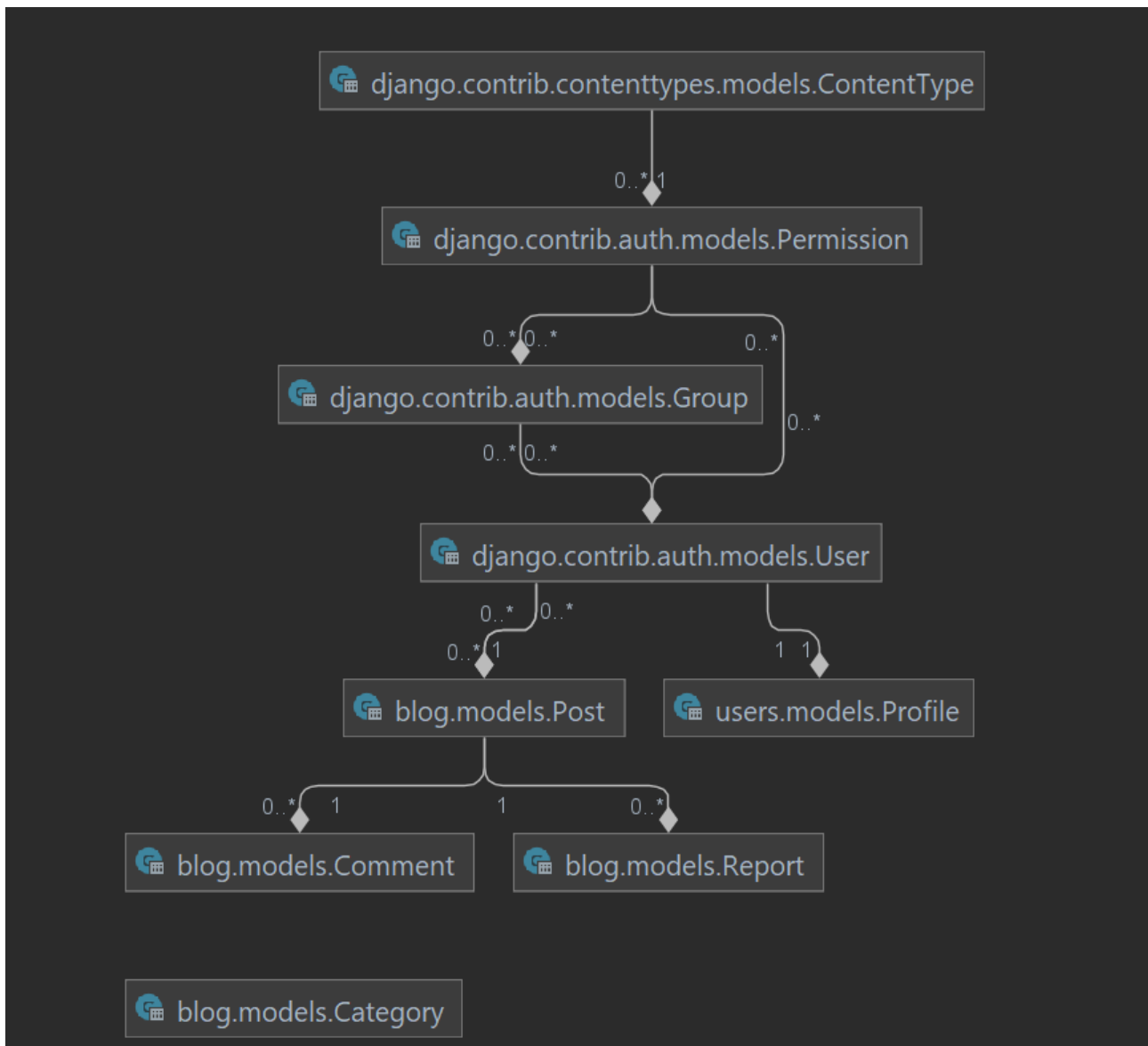

```
class Post(models.Model):
    ...

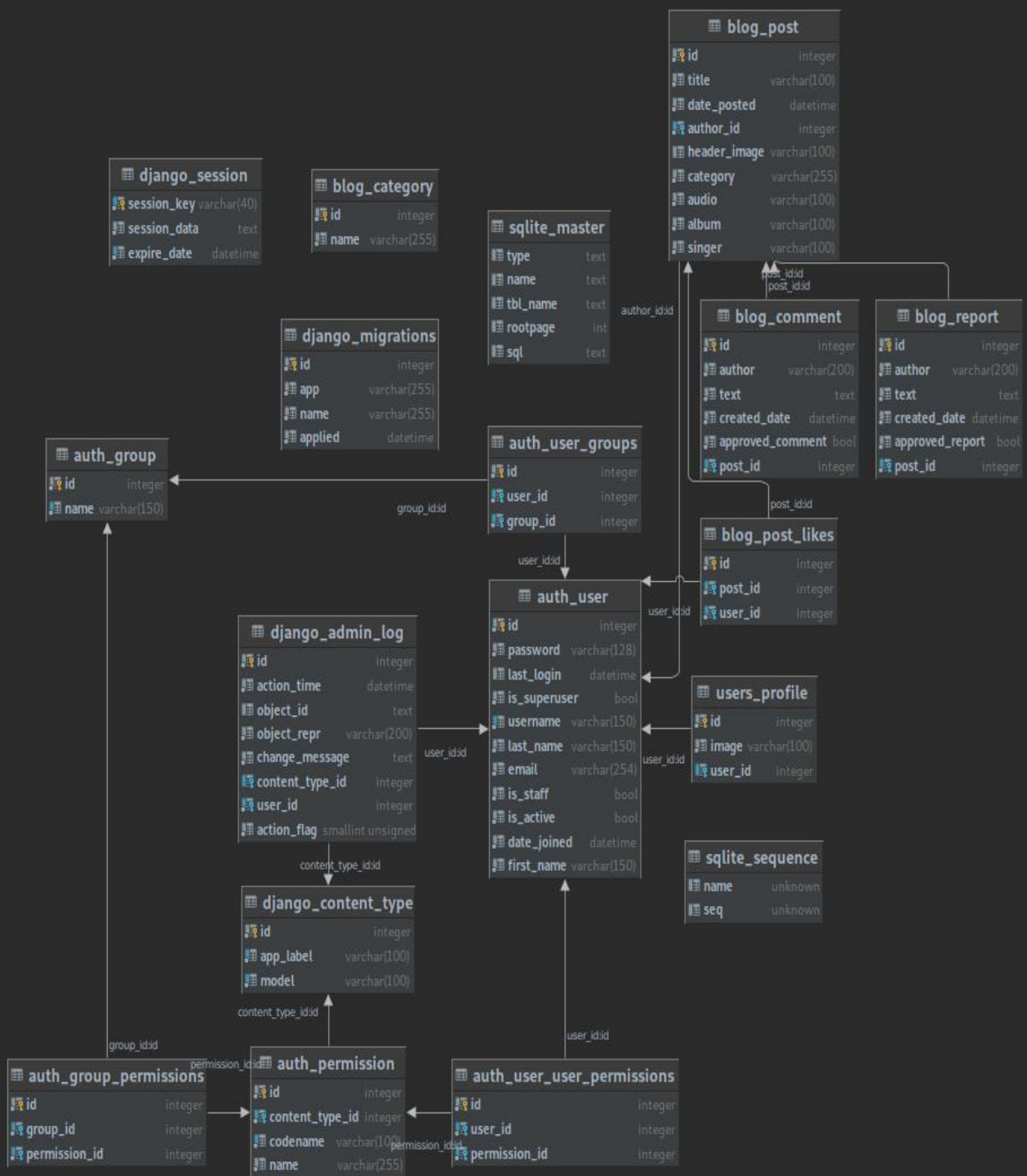
    likes = models.ManyToManyField(User, related_name="blog_post")

    def total_likes(self):
        return self.likes.count()
```

blog_post_likes	
id	integer
post_id	integer
user_id	integer

SCHEMA COMPLETO DELLE TABELLE UTILIZZATE





UTILIZZO DELL'APPLICAZIONE

- Ingresso sulla piattaforma: l'utente accede alla piattaforma come utente anonimo e può navigare solamente attraverso la homepage, il dettaglio dei contenuti presenti in homepage e la pagina di registrazione. Qualora tenti di interagire con il sistema di feedback della piattaforma verrà automaticamente reindirizzato alla pagina di registrazione del sito, ma può comunque ascoltare i brani che visualizza sulla piattaforma.
- Fase di registrazione: qualora l'utente decida di registrarsi, potrà farlo attraverso il form di registrazione all'interno della pagina di registrazione. Inserendo i propri dati personali verrà infine creato un account, che potrà essere accesso attraverso il proprio username e la propria password. Il profilo utente è modificabile post creazione.
- Creazione di un post: una volta registrati all'interno del sito sarà possibile caricare contenuto multimediale e interagire attraverso i feedback con gli altri contenuti del sito. Una volta creato un post sarà anche possibile modificarlo.
- Feedback ad altri contenuti: gli utenti registrati possono quindi lasciare un like o un commento ai contenuti presenti sulla piattaforma e qualora dovesse ritenere inopportuni alcuni contenuti possono sempre segnalarlo con l'apposito form all'ADMIN della piattaforma.
- Ricerca di contenuti: in quanto utente registrati sarà possibile usufruire delle funzionalità di ricerca messe a disposizione dal sito.
- Visualizzazione pagina suggeriti: come utenti iscritti si potrà anche visualizzare la pagina dei post consigliati da DOS Music.

TECNOLOGIE UTILIZZATE E SCELTE DI SVILUPPO (LE SCELTE DI SVILUPPO SONO DA INSERIRE)

BACKEND

È stato utilizzato Django secondo le modalità vista durante il corso di tecnologie web e sono stati sfruttati alcuni plugins:

- Pillow: per caricare le immagini (profilo/per le copertine delle canzoni) all'interno del sito.
- Ckeditor: aggiunge maggiore possibilità di editing al contenuto testuale dei post.
- Crispy Forms: crea dei "form" grazie ai quali gli utenti possono registrarsi o caricare contenuto multimediale all'interno della piattaforma.
- Baker: per creare test per i modelli in maniera veloce e comoda. L'idea alla base è che se ho un modello con molti campi, io non voglio scrivere manualmente dei valori per ciascuno di essi, ma preferisco affidare questo compito a Baker.

COMUNICAZIONE

Per il momento, le comunicazioni tra frontend e backend avvengono tramite il protocollo HTTP (non ideale il sistema dal punto di vista della sicurezza). Non è escluso che, in fase di deployment, si possa aggiungere un layer di sicurezza alle comunicazioni e passando a HTTPS.

FRONTEND

Il frontend è stato realizzato tramite Bootstrap e lo stile del sito è stato ottenuto andando ad integrare il tema Bootstrap i386, che in origine sarebbe orientato a siti in WordPress che sfruttano Bootstrap, ma integrato all'interno del framework Django. Il motivo del tema è il sistema DOS e conferisce alla piattaforma un aspetto estremamente retrò, grazie anche agli effetti grafici in JavaScript che compongono la pagina come se fosse un vecchio terminale ogni volta che si naviga all'interno del sito.

ORGANIZZAZIONE DEL PROGETTO

Il progetto è diviso principalmente in tre parti:

- Django_Project: che contiene tutte quelle che sono le impostazioni legate al sito.
- Blog: che ospita tutta la gestione del contenuto multimediale che viene caricato all'interno del sito.
- Users: che contiene i modelli che riguardano la gestione degli utenti.

L'idea è alla base di tale divisione è stata quella di rendere quanto più modulare possibile il sito per un futuro utilizzo di alcune sue parti (in particolare per il modulo User). Per quel che riguarda i Post la scelta è stata quella di tenere una applicazione unica per la creazione del contenuto multimediale, che fosse però internamente divisa in modo tale che potesse essere di facile gestione l'implementazione di altri moduli legati ai post, come è stato per l'implementazione del Report, dei Commenti, ...

TEST SVOLTI

Per effettuare i test sulla piattaforma si è ricorso alla metodologia di utilizzo dei file Test.py che hanno permesso di andare a saggiare la bontà delle seguenti funzionalità:

- Registrazione alla piattaforma, con conseguente creazione di un utente

```
class ProfileTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = User.objects.create_user(username='jacob', email='jacob@gmail.com',
        password='prova1234')

    def test_details(self):
        request = self.factory.get('/profile')
        request.user = self.user
        response = profile(request)
        self.assertEqual(response.status_code, 200)
```

- Upload di contenuto multimediale sulla piattaforma

```
class PostTest(TestCase):
    def testPosting(self):
        def test_event_model(self):
            '''event = Post.objects.create(
                title="Some title",
                audio="/media/cash.mp3",
                header_image="/media/images/cash.jpg",
                date_posted=datetime.time,
                author="The body",
                singer="The body",
                album="the-slug",
                likes=800,
                category='rock'
            )'''

        user_post = baker.make(Post, title="Post Test")
        self.assertEqual(str(user_post), "Post Test")
```

- Test per la homepage


```
class HomePageTest(TestCase):

    def testHome(self):
        client = Client()
        response = client.get(reverse('blog-home'))
        self.assertEqual(response.status_code, 200)
```

RISULTATI

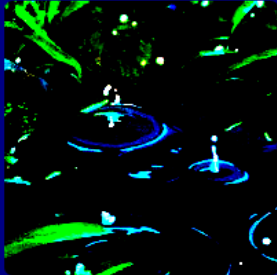
DOSMUSIC
Suggestion
New Category
Add Song
Profile
Logout
< Search >

New Songs



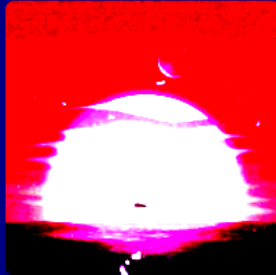
>>> I Heed A Map <<<
giovanni lofi September 12, 2021

0:42 / 2:39




>>> Sweet Sad Afternoon <<<
giovanni trap September 12, 2021

0:00 / 2:15




>>> Positivity <<<
giovanni lofi September 12, 2021

0:00 / 2:14



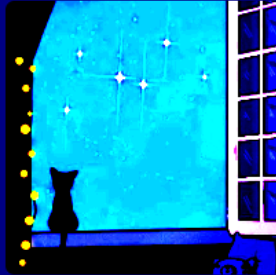
>>> Isolation <<<
giovanni lofi September 12, 2021

0:00 / 2:08



>>> Mirrorverse <<<
maxbubblegum vaporwave September 12, 2021

0:00 / 4:27



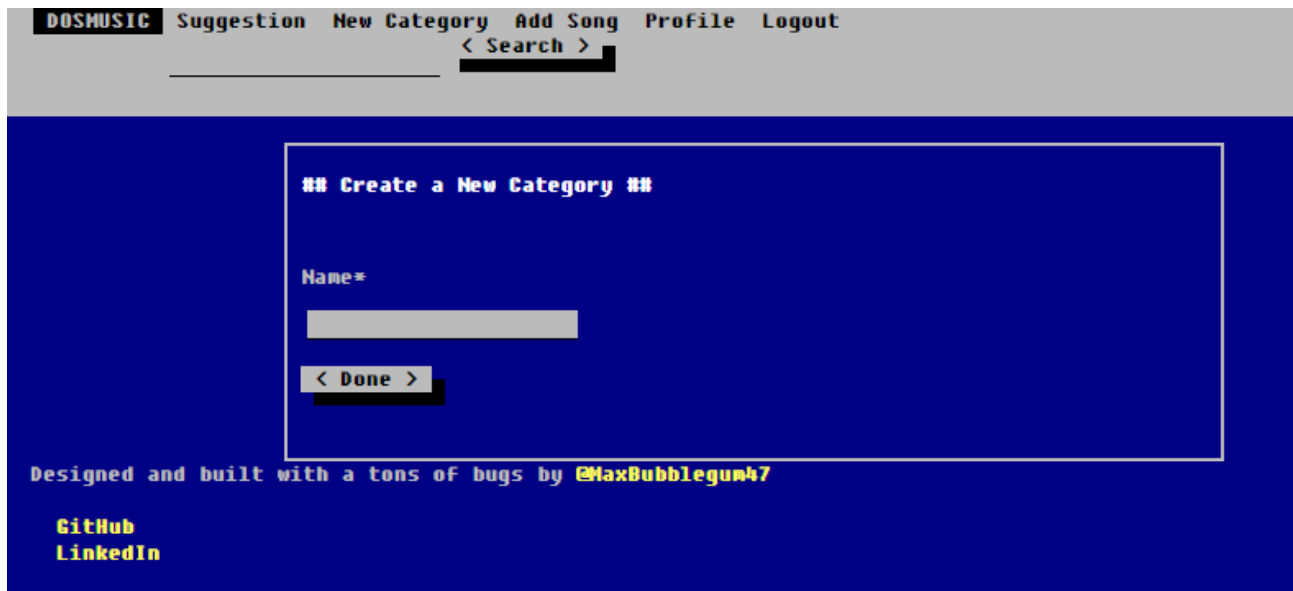
>>> Thoughts <<<
maxbubblegum trap September 12, 2021

0:00 / 1:55

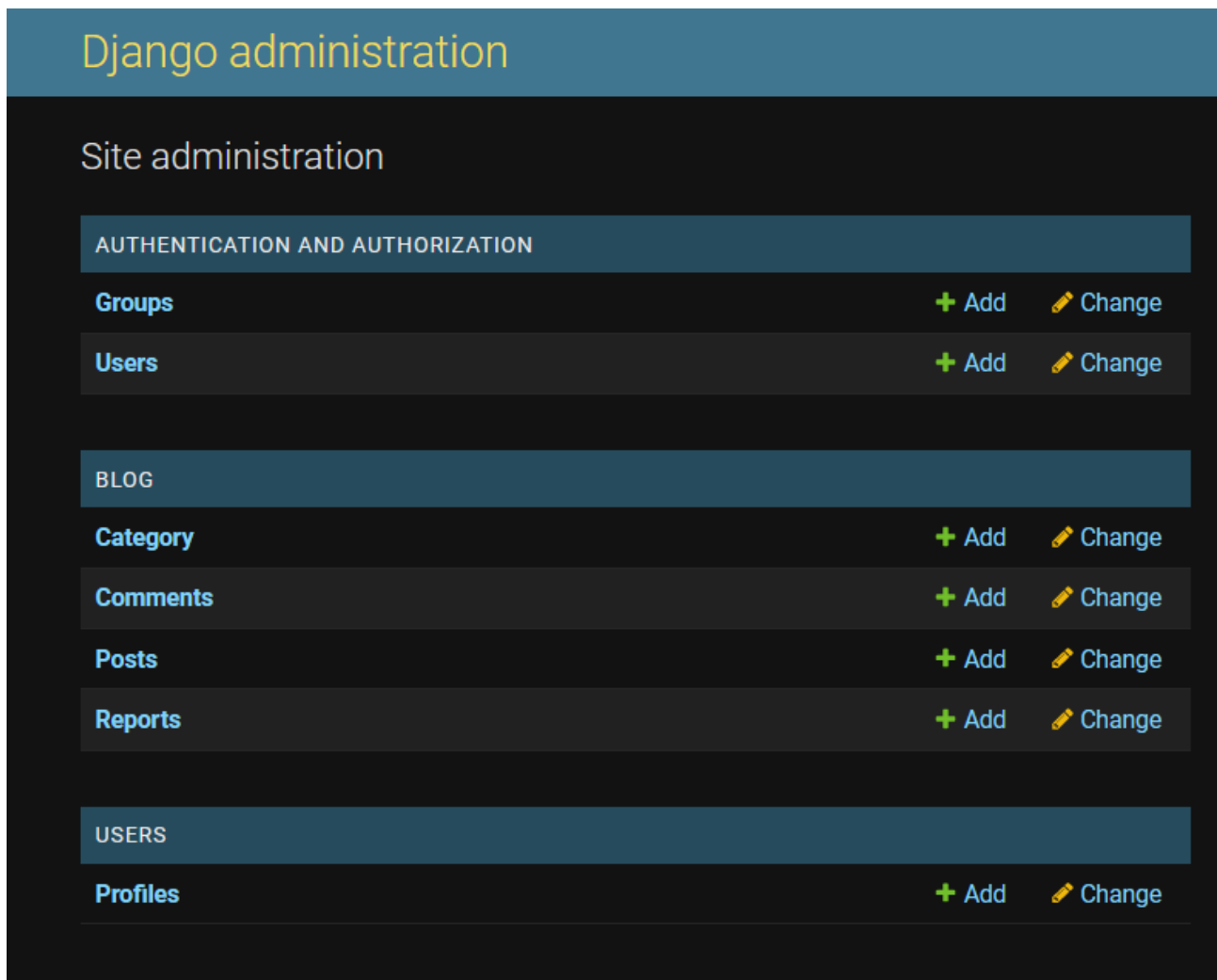
< 1 >
< 2 >
< Next >
< Last >

Designed and built with a tons of bugs by @MaxBubblegum47

[GitHub](#)
[LinkedIn](#)



Creazione di una nuova categoria



Pannello di amministrazione 1

Join Today

Username*

**Required. 150 characters or fewer.
Letters, digits and @/./+/-/_
only.**

Email*

Password*

Your password can't be too
similar to your other personal
information.
Your password must contain at
least 8 characters.
Your password can't be a
commonly used password.
Your password can't be entirely
numeric.

Password confirmation*

**Enter the same password as before,
for verification.**

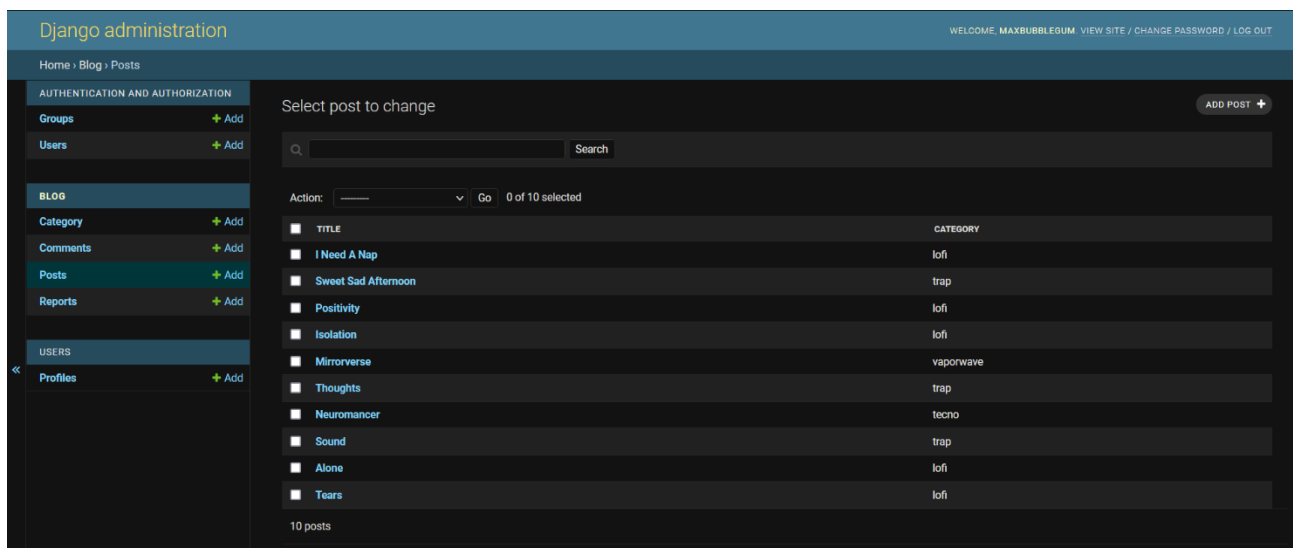
< Sign Up >

Already Have An Account? [Sign In](#)

Designed and built with a tons of bugs by [@MaxBubblegun47](#)

[GitHub](#)

[LinkedIn](#)



Pannello di amministrazione 2

DOSMUSIC [Suggestion](#) [New Category](#) [Add Song](#) [Profile](#) [Logout](#)

[< Search >](#)

New Song

Title*

Album

Singer

Header image

Nessun file selezionato

Category

Audio*

Nessun file selezionato

Designed and built with a tons of bugs by [@MaxBubblegum47](#)

[GitHub](#)
[LinkedIn](#)

Form per inserimento nuova canzone



maxbubblegum
257544@studenti.unimore.it

< User's Post >

Edit Profile

Username*

maxbubblegum

Required. 150 characters or fewer.
Letters, digits and @/./+/-/_
only.

Email*

257544@studenti.unimo

Image*

Currently: **profile_pics/wp4397508-
death-stranding-fragile-
wallpapers.jpg**

Change:

Scegli file Nessun file selezionato

< Update >

Designed and built with a tons of bugs by @MaxBubblegum47

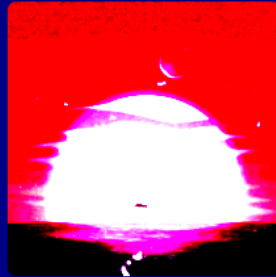
[GitHub](#)

Most Liked Songs



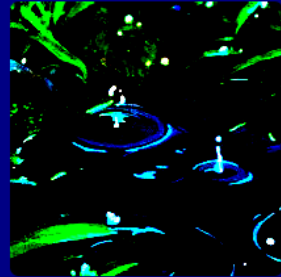
>>> **Isolation** <<<
giovanni lofi September 12, 2021

▶ 0:37 / 2:08 🔊 ⋮



>>> **Positivity** <<<
giovanni lofi September 12, 2021

▶ 0:00 / 2:14 🔊 ⋮



>>> **Sweet Sad Afternoon** <<<
giovanni trap September 12, 2021

▶ 0:00 / 2:15 🔊 ⋮



>>> **I Need A Map** <<<
giovanni lofi September 12, 2021

▶ 0:00 / 2:39 🔊 ⋮

[< First >](#) [< Previous >](#) [< 1 >](#) [< 2 >](#)

Designed and built with a tons of bugs by [@MaxBubblegun47](#)

[GitHub](#)
[LinkedIn](#)

Post's Report

Text*

[< Report >](#)

Designed and built with a tons of bugs by [@MaxBubblegum47](#)

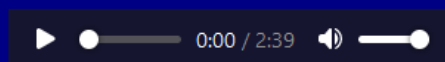
[GitHub](#)

[LinkedIn](#)



Comments

>>> I Need A Map <<<
giovanni lofi September 12, 2021




[< Report >](#) [< Like >](#) - 0 Likes

[< Add comment >](#)

Designed and built with a tons of bugs by @MaxBubblegum47

[GitHub](#)
[LinkedIn](#)



>>> I Need A Map <<<
giovanni lofi September 12, 2021

0:23 / 2:39

< Report > < Like > - 2 Likes

< Add comment >

Comments

test

I'm in love with that song!!!

maxbubblgum

so chill

Visualizzazione likes e commenti

ISTRUZIONE PER L'INSTALLAZIONE

Link per scaricare il progetto:

- https://github.com/MaxBubblegum47/django_project.git

-

pip install -r requirements.txt

PROBLEMI RISCONTRATI

La maggior parte dei problemi in termini di sviluppo, oltre alla comprensione del framework di sviluppo, sono stati sicuramente l'implementazione del sistema di feedback e la gestione del tema di bootstrap.

In primis perché è stata necessaria moltissima sperimentazione per raggiungere un risultato ottimale in termini di funzionalità del backend e integrazione delle nuove modalità con la GUI del sito.

In secondo luogo, l'implementazione del tema è risultata difficoltosa dal momento che non vi fosse documentazione riguardo l'implementazione del tema all'interno di framework al di fuori di WordPress. Oltretutto il tema si serve di alcune classi specifiche HTML, che hanno portato ad un riadattamento di tutto il sito (precedentemente costruito con un'altra versione di Bootstrap e con un altro tema).

Diverso invece il problema della responsività del sito: con alcuni browser ho notato come possa cambiare la sensibilità del sito agli input degli utenti, nel senso che alle volte finché l'animazione di caricamento pagina non è terminata non sarà possibile effettuare determinate operazioni (questo l'ho notato molto su Google Chrome). Sto ancora indagando e qualora ne dovessi venire a capo aggiornerò il progetto di conseguenza.