Delhi Metro Rail Corporation

An Internship Report

on

A file-sharing web application for DMRC employees using python-django

Submitted By-

VEDANTA SHARAN(21BCE2970)

ANUSHKA JAIN (21BCE0038)

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering(Core)

Vellore Institute of Technology, Vellore **Under the guidance of:**

Mr. Manu Bhardwaj

Mr. Saurabh Sharma

ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks to Mr. Saurabh Sharma and Mr. Manu Bhardwaj and all other esteemed engineers of Delhi Metro Rail Corporation for having given me a chance to complete my training. I convey my thanks for their cooperation valuable guidance, constructive criticism, and constant encouragement. I am highly grateful to the Delhi Metro Railway Corporation for giving me an opportunity for the training and for providing the required facilities. It was a great experience working in the institution and learning from such knowledgeable engineers. I hope the knowledge gained here during the training would help me in the development of my career as an engineer.

ABSTRACT

Rapid urbanization and population growth in places like Delhi have raised the demand for reliable public transportation. The Delhi Metro, known for its extensive network and simple connections, is relied on daily by millions of passengers. Therefore systematic and secure sharing of data especially confidential is extremely important for the organization.

Thus, this project was an attempt of creating a personal application for the Delhi Metro Corporation.

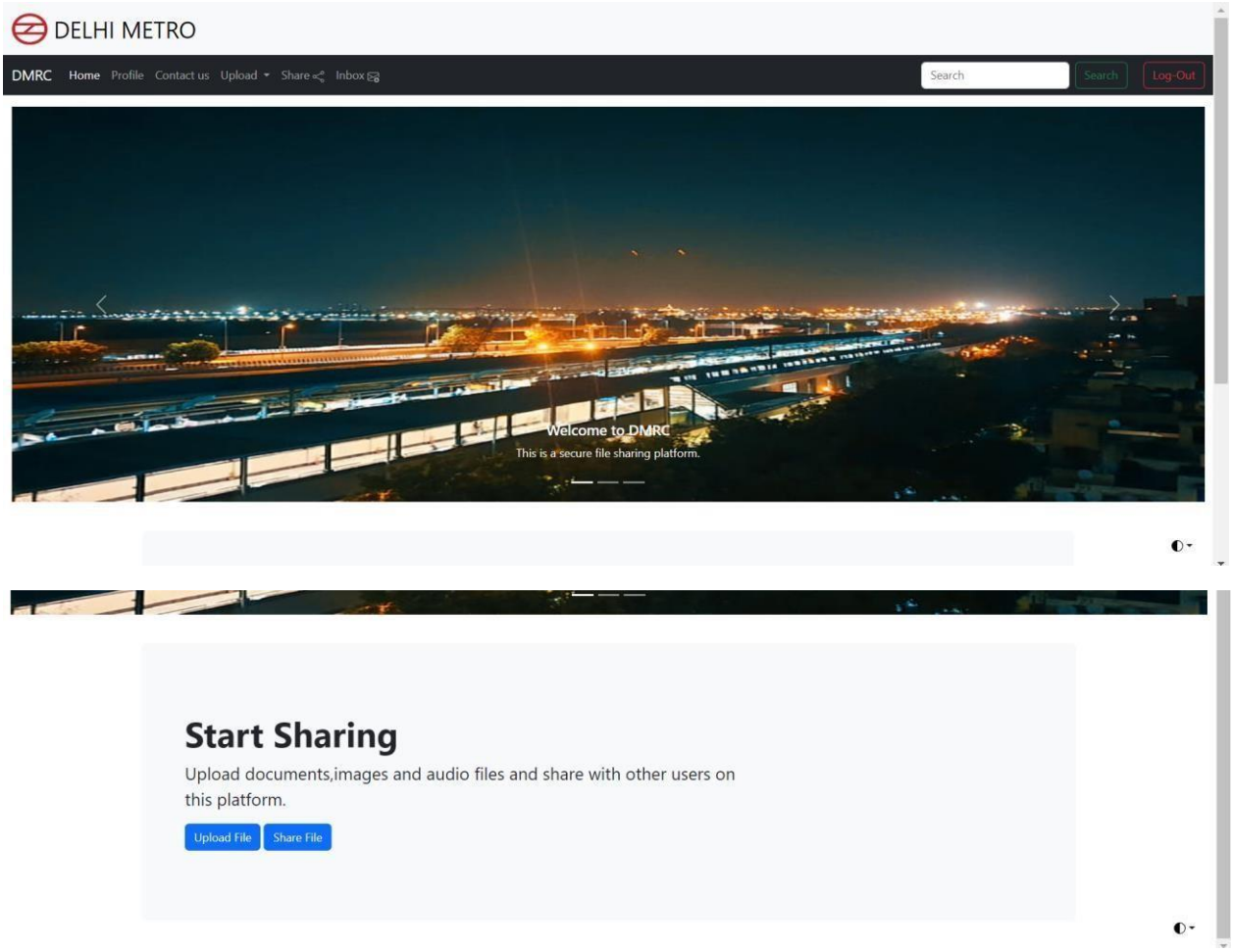
CONTENT

INTRODUCTION

The Delhi Metro has quickly become an integral part of the city's transportation network. Since its introduction in 2002, it has helped tremendously with the transportation needs of the city's expanding population. The metro system has benefitted millions of commuters by cutting down on traffic and offering a more comfortable mode of transportation because of its extensive coverage and efficient connectivity.

Effective communication is essential for the functioning of any organisation. It is important that the data being shared in the organisation remains intact and only accessible by authorised members of the organisation.

Thus , this project aims to create a file sharing platform for DMRC which has security measures put in place to keep the data safe from any hackers or unethical use. It uses Django and its in-built functions along with html and css code connected to a No-Sql database(MongoDB) to create a fully functioning website.



OBJECTIVE:

The aim is to create a website which allows users to create an account, log in to the site , save files in their accounts for future use and share files with other users along with proper security measures to keep the data safe from potential threats.

TECHNOLOGIES USED:

- Python
- Django

- MongoDB
- Html ,Css , Javascript using Bootstrap

Project Description

The web application is created using python's web framework Django.

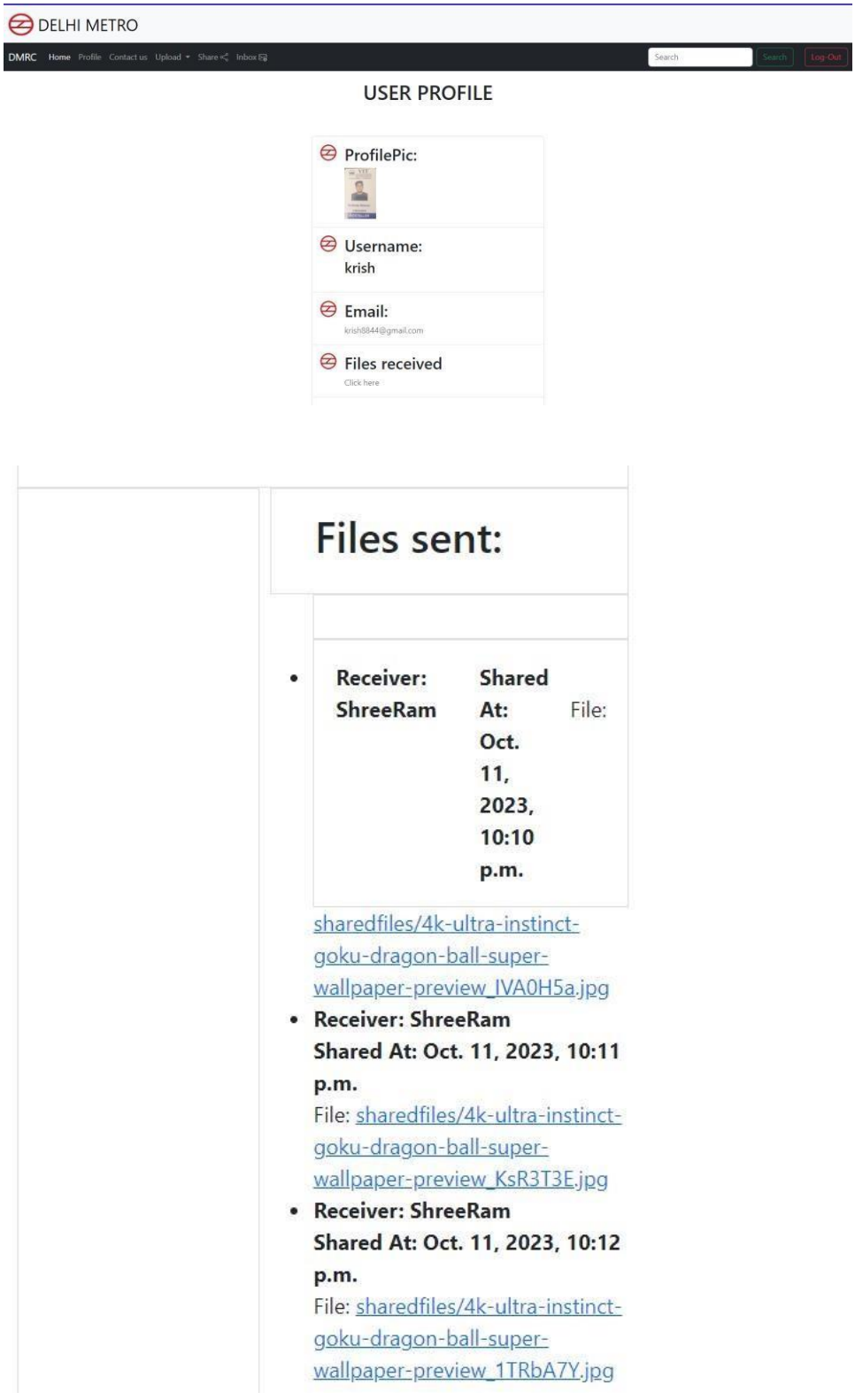
It consists of the following components:

1. The Home page:

   It consists of a carousel from bootstrap with images of personal choice and can be changed at any time. Apart from the carousel it has two buttons which are linked to two pages i.e the file upload page and the file sharing page respectively.

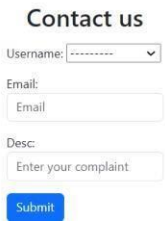
2.The Profile page:

- It consists of the profile of the user. It has a profile-picture component which has the profile picture set to the dmrc logo as default.
- It displays the username,email and a list of files sent to another user with the receiver's name, date and time and the file itself altogether.
- It also has a 'Files received' component which on clicking sends the user to the inbox page.



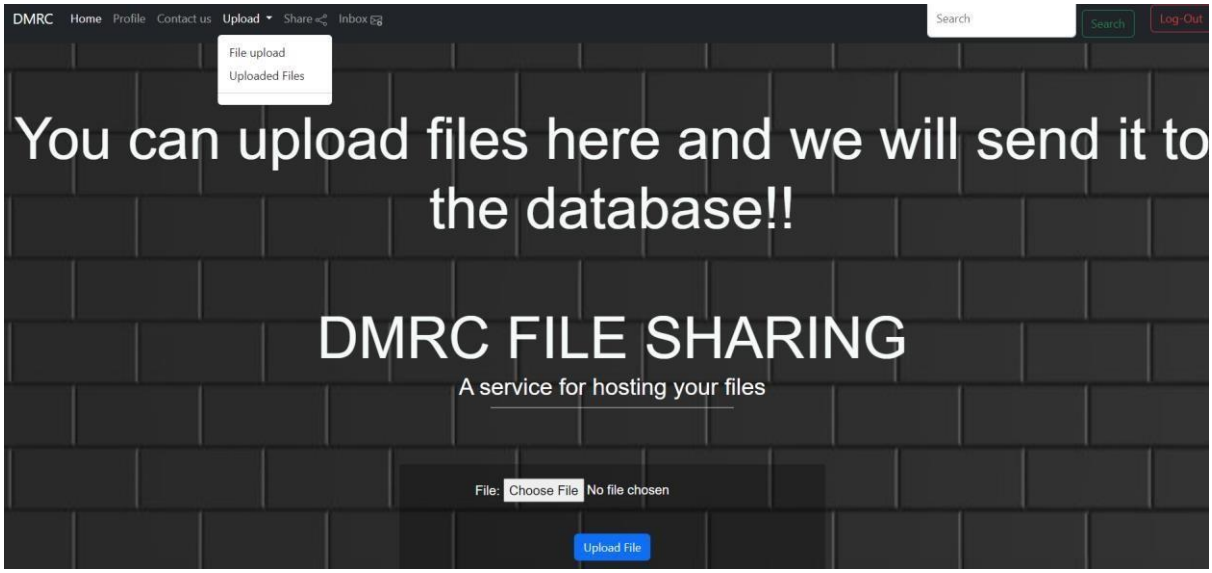
3.The Contact Us page:

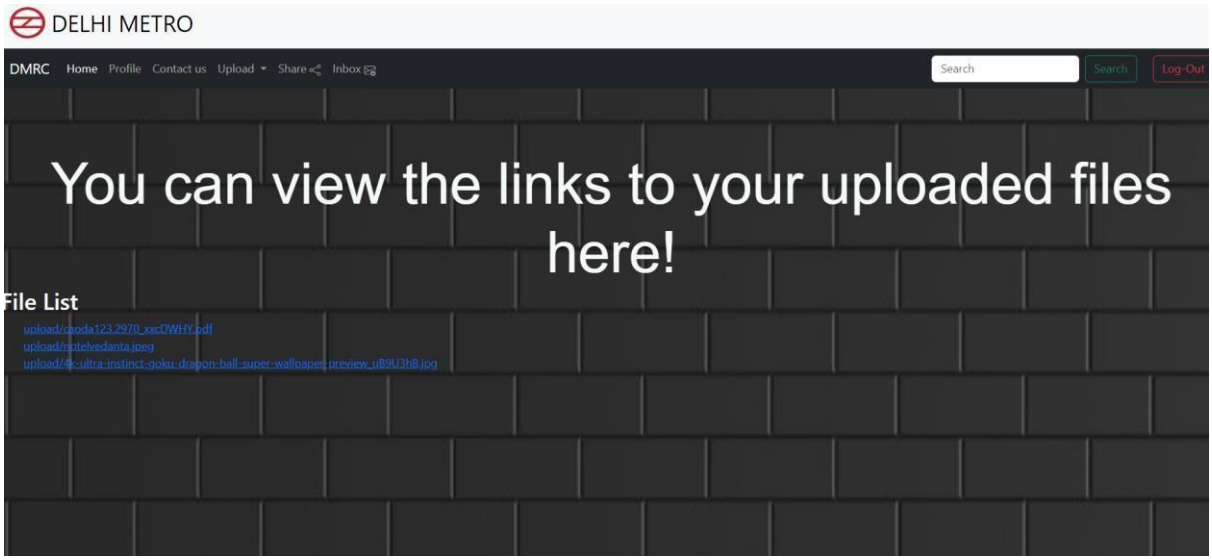
- It consists of the username,email and a description of the issue the user is facing.
- The data after submission can be viewed in the mongodb database as well as the Djangoadmin's portal.



4.The Upload dropdown:

- It has two options:
- File Upload: It allows the user to upload the file and save the link to the file on the Uploaded files page.
- Uploaded files: It consists of the list of file links that the user has uploaded.

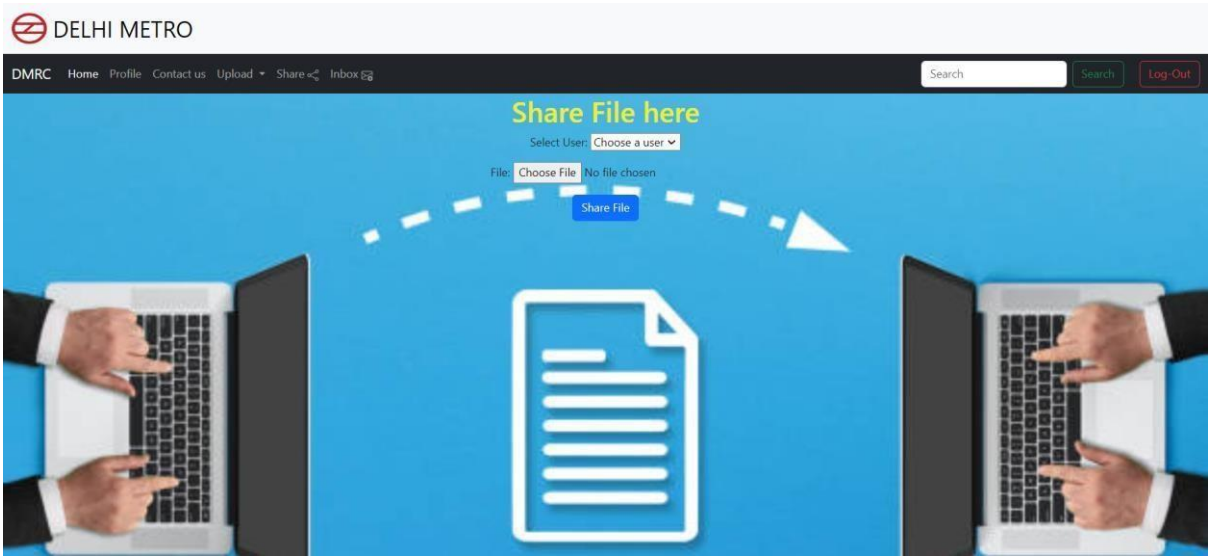




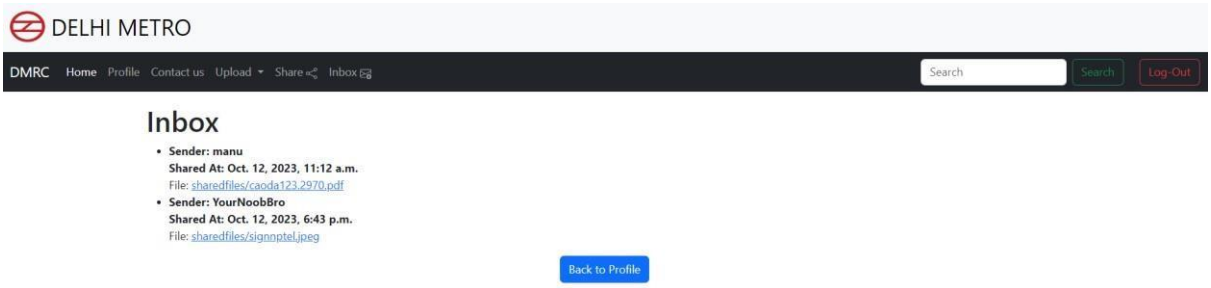
5. Share page: It consists of two fields:

- Receiver:to whom the file is sent to.
- File: The file to be shared.
- After sharing the user is redirected to the profile page with a success message and the file sent can be viewed in the

file sent field of the profile page.

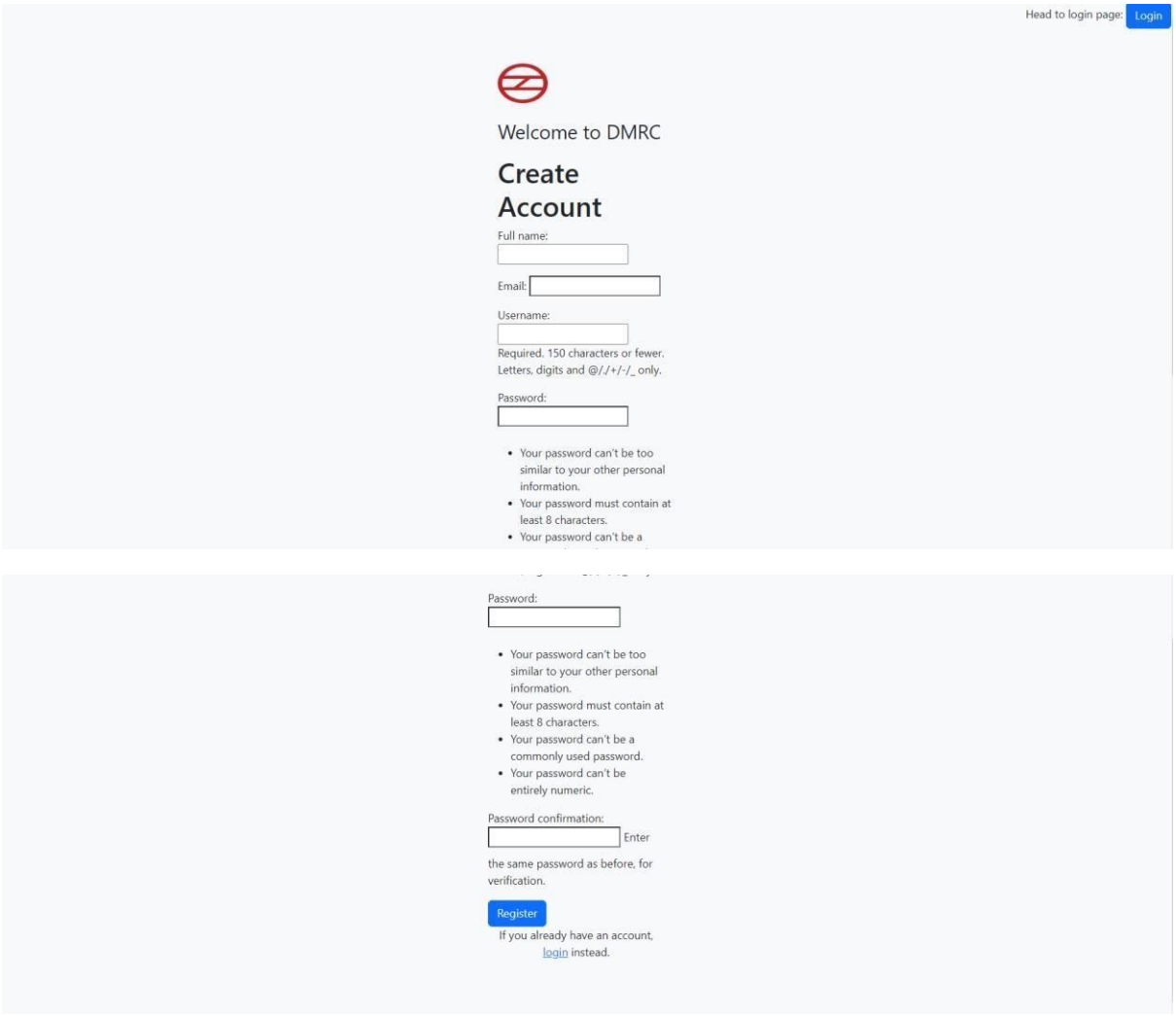


6.Inbox page: It consists of the list of files that have been sent by other users along with the sender's name and the date and time of sharing.

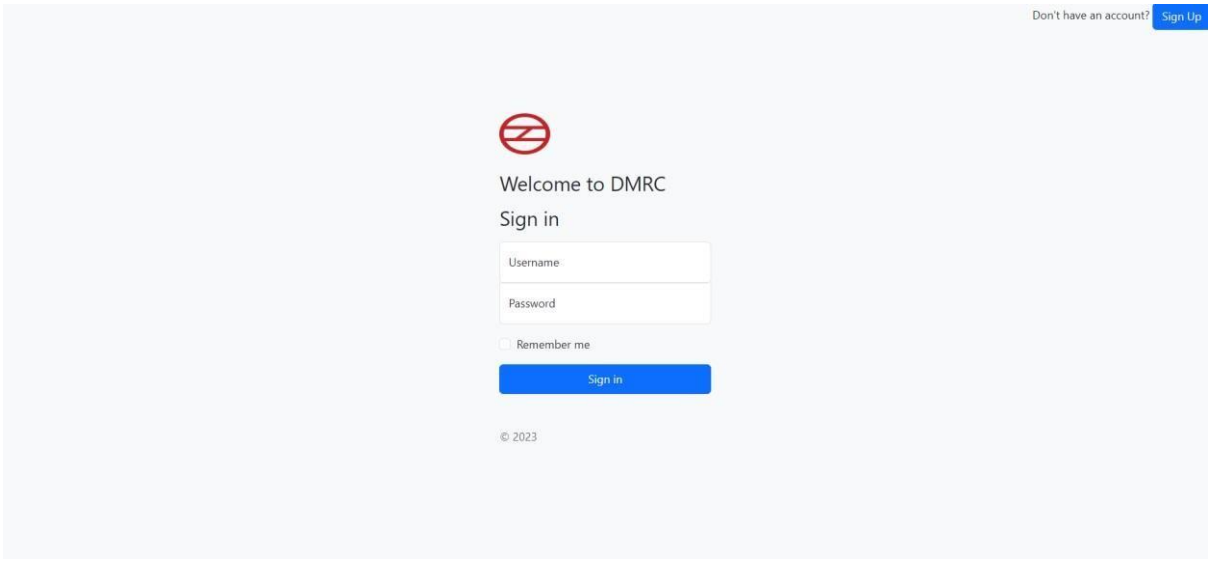


7.Signup page:

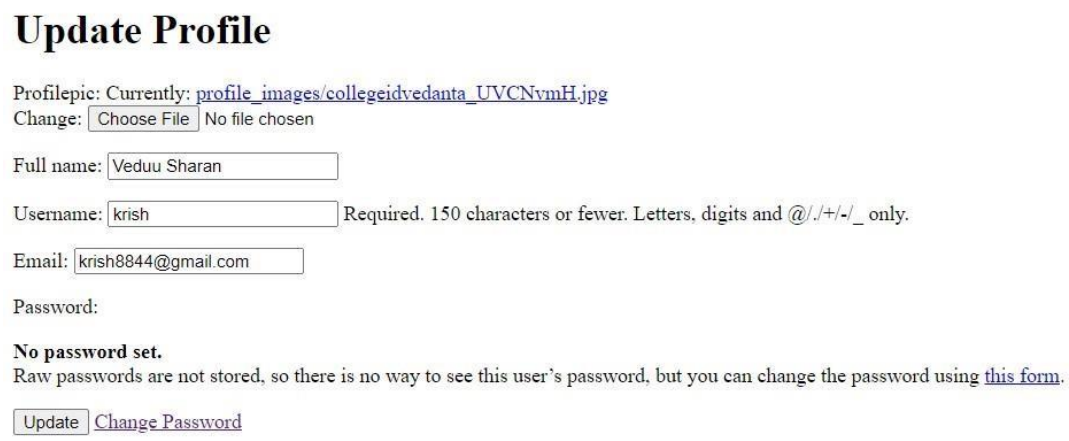
The signup page is used to create a custom user that can access the website by logging in and share files. Django's inbuilt UserCreationForm was used to create the page and it has security measures to ensure safe password creation.



8. Login Page: Finally , the login page allows the user to access the website by using their username and password. Only valid users created by the signup form(apart from the django's superuser) can access the website.



9.The update Profile page:



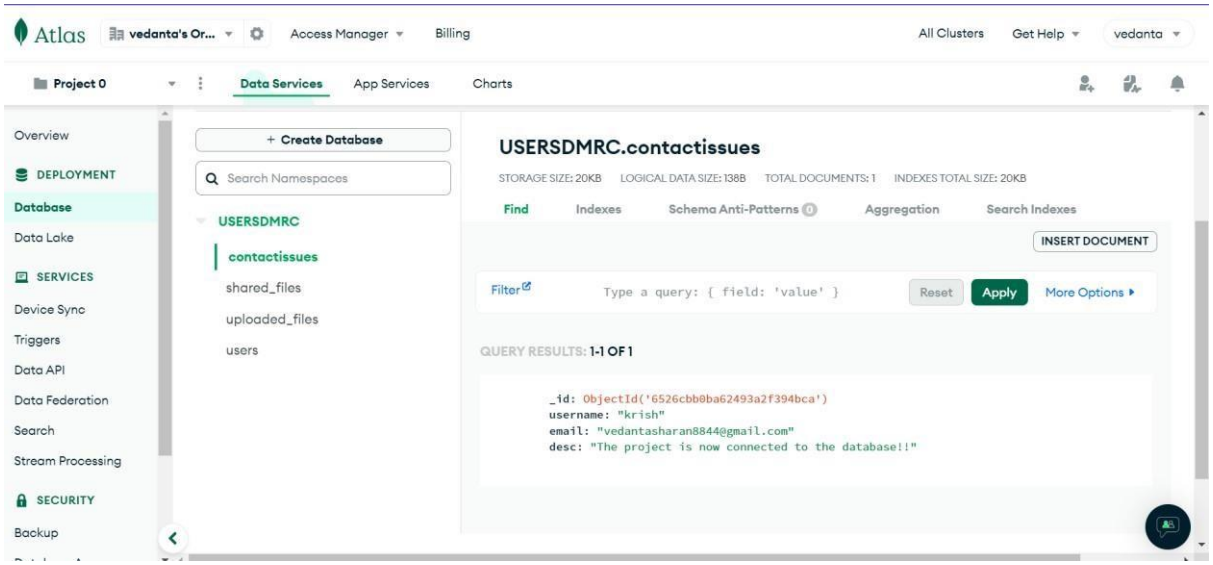
This page allows to update the information of the users and also allows them to change their password.

DATABASE DESIGN

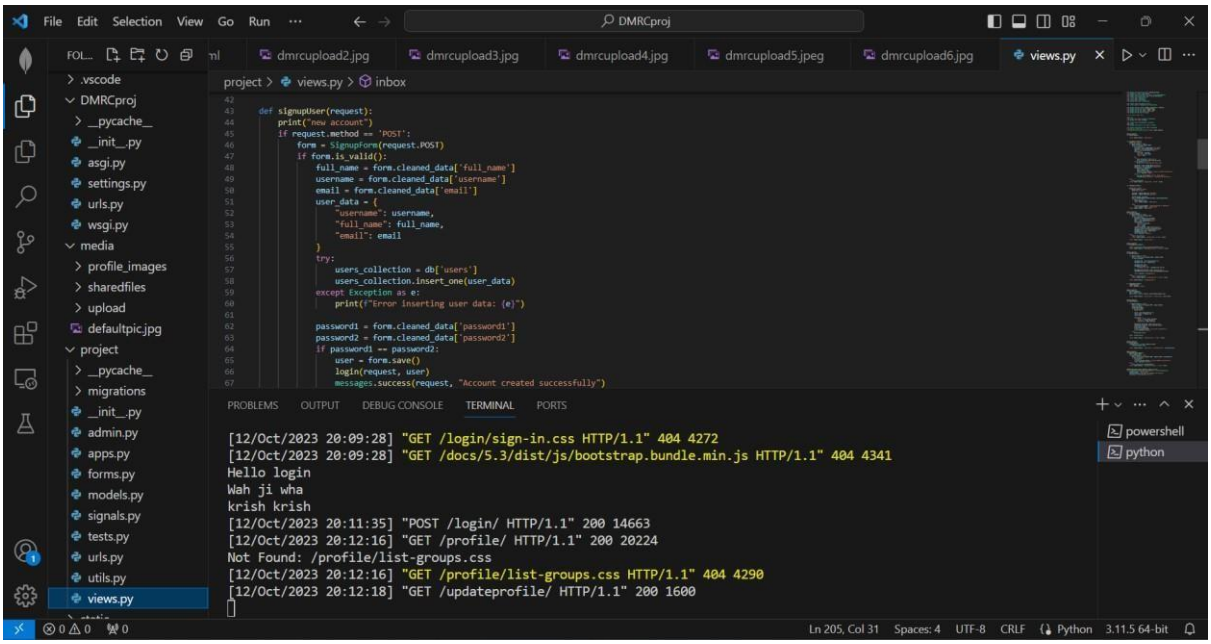
A No-Sql database MongoDB is used. The Django project is connected to the cloud storage of MongoDB called MongoDB atlas using PyMongo.

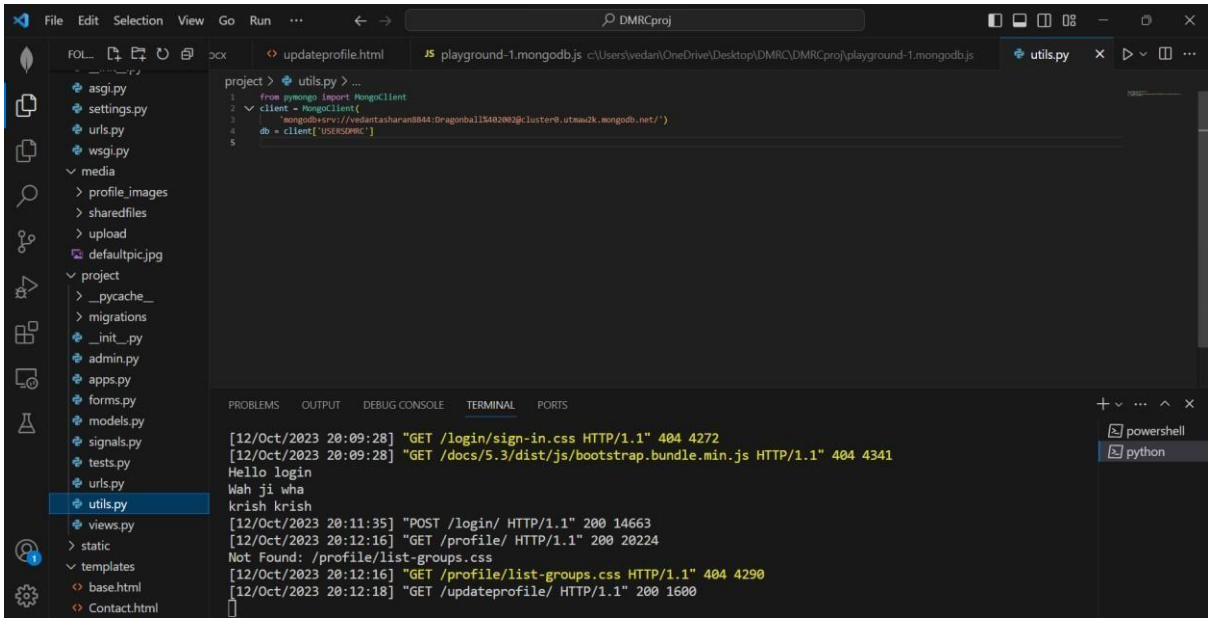
Four collections have been included to store data:

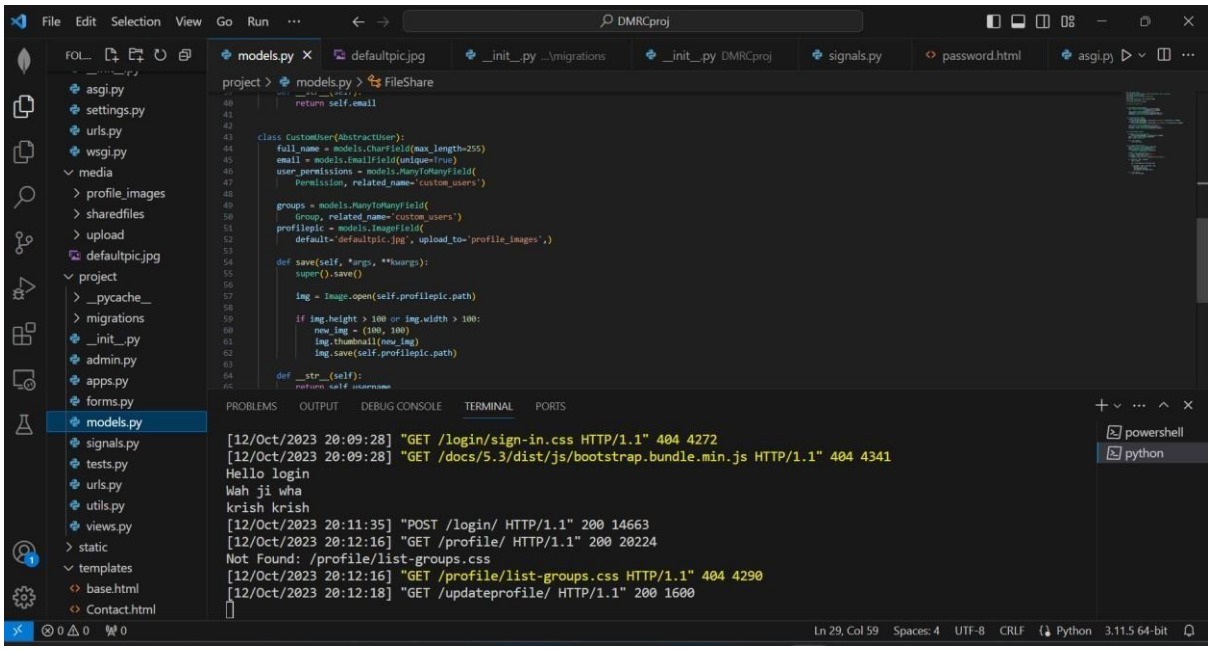
1. Users: To store the required user information.
2. Contactissues: To store the data posted in the ‘Contact Us’ page.
3. Uploaded_files: To store the file path of the uploaded files on the file upload page.
4. Shared_files: To share the file path of the shared files on the file share page.

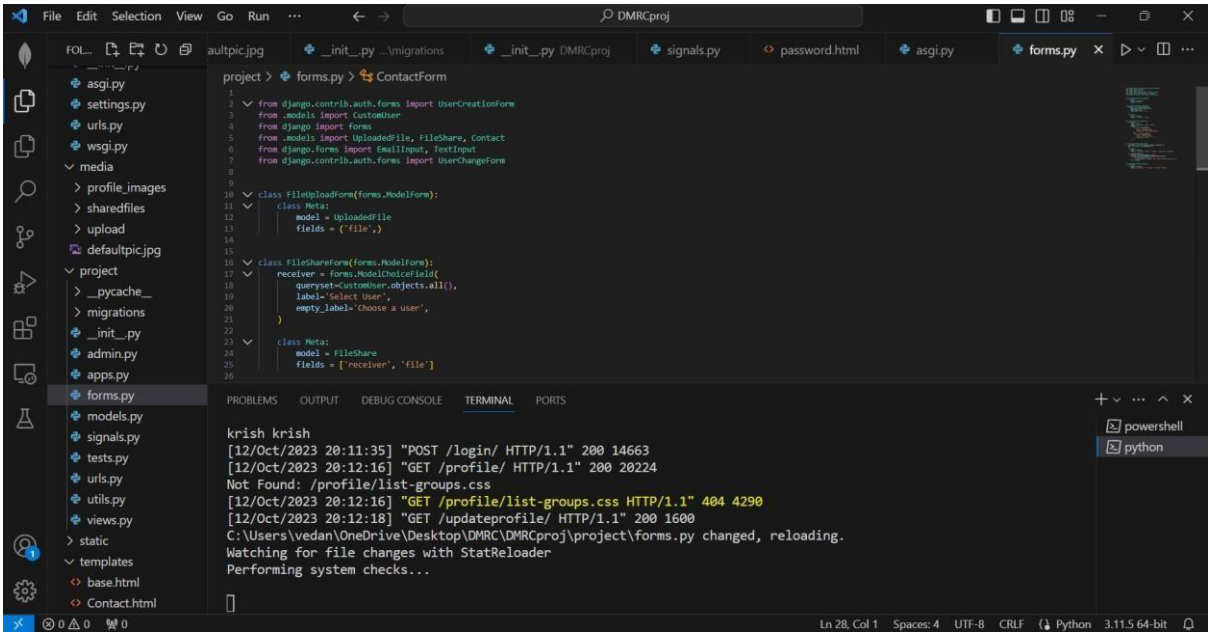


CODE SNIPPETS




File Edit Selection View Go Run
DMRCproj
updateprofile.html
playground-1.mongodb.js c:\Users\vedan\OneDrive\Desktop\DMRC\DMRCproj\playground-1.mongodb.js
utils.py
project > utils.py > ...
from pymongo import MongoClient
client = MongoClient(
'mongodb+srv://vedantasharan8844:Dragonball%402002@cluster0.utmaw2k.mongodb.net/')
db = client['USERSDMRC']
asgi.py
settings.py
urls.py
wsgi.py
media
profile_images
sharedfiles
upload
defaultpic.jpg
project
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
signals.py
tests.py
urls.py
utils.py
views.py
static
templates
base.html
Contact.html
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
python
[12/Oct/2023 20:09:28] "GET /login/sign-in.css HTTP/1.1" 404 4272
[12/Oct/2023 20:09:28] "GET /docs/5.3/dist/js/bootstrap.bundle.min.js HTTP/1.1" 404 4341
Hello login
Wah ji wha
krish krish
[12/Oct/2023 20:11:35] "POST /login/ HTTP/1.1" 200 14663
[12/Oct/2023 20:12:16] "GET /profile/ HTTP/1.1" 200 20224
Not Found: /profile/list-groups.css
[12/Oct/2023 20:12:16] "GET /profile/list-groups.css HTTP/1.1" 404 4290
[12/Oct/2023 20:12:18] "GET /updateprofile/ HTTP/1.1" 200 1600


File Edit Selection View Go Run
DMRCproj
models.py
defaultpic.jpg
__init__.py ...\migrations
__init__.py DMRCproj
signals.py
password.html
asgi.py
project > models.py > FileShare
return self.email
class CustomUser(AbstractUser):
full_name = models.CharField(max_length=255)
email = models.EmailField(unique=True)
user_permissions = models.ManyToManyField(
Permission, related_name='custom_users')
groups = models.ManyToManyField(
Group, related_name='custom_users')
profilepic = models.ImageField(
default='defaultpic.jpg', upload_to='profile_images',)
def save(self, *args, **kwargs):
super().save()
img = Image.open(self.profilepic.path)
if img.height > 100 or img.width > 100:
new_img = (100, 100)
img.thumbnail(new_img)
img.save(self.profilepic.path)
def __str__(self):
asgi.py
settings.py
urls.py
wsgi.py
media
profile_images
sharedfiles
upload
defaultpic.jpg
project
__pycache__
migrations
__init__.py
admin.py
apps.py
forms.py
models.py
signals.py
tests.py
urls.py
utils.py
views.py
static
templates
base.html
Contact.html
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
python
[12/Oct/2023 20:09:28] "GET /login/sign-in.css HTTP/1.1" 404 4272
[12/Oct/2023 20:09:28] "GET /docs/5.3/dist/js/bootstrap.bundle.min.js HTTP/1.1" 404 4341
Hello login
Wah ji wha
krish krish
[12/Oct/2023 20:11:35] "POST /login/ HTTP/1.1" 200 14663
[12/Oct/2023 20:12:16] "GET /profile/ HTTP/1.1" 200 20224
Not Found: /profile/list-groups.css
[12/Oct/2023 20:12:16] "GET /profile/list-groups.css HTTP/1.1" 404 4290
[12/Oct/2023 20:12:18] "GET /updateprofile/ HTTP/1.1" 200 1600
Ln 29, Col 59 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit



RESULTS AND DISCUSSION

The project was created by using Django integrated with a MongoDB database. The user can create an account, login,update the automatically created profile,write complaints to the administrator,upload and save files and share files with other users. The security features while user creation are implemented using django's built in features like Django.auth.

CONCLUSION AND FUTURE SCOPE

An attempt was made to create a secure file sharing platform for the Dmrc employees. The project has a lot of future scope related to design, security features and database implementation with a powerful framework like Django.

REFERENCES

- https://docs.djangoproject.com/en/4.2/
- https://getbootstrap.com/
- https://www.mongodb.com/
- www.google.com
- Stackoverflow.com • Youtube Github link to project:

https://github.com/Anushkajain2003/-file-sharing-web-application-