**A**

**Summer Training Project**

**On**

Wandoc collaboration

**Session:** 2022-23

**B. Tech 4th Year**



#### Department of Computer Science and Engineering

**Bundelkhand Institute of Engineering and Technology**

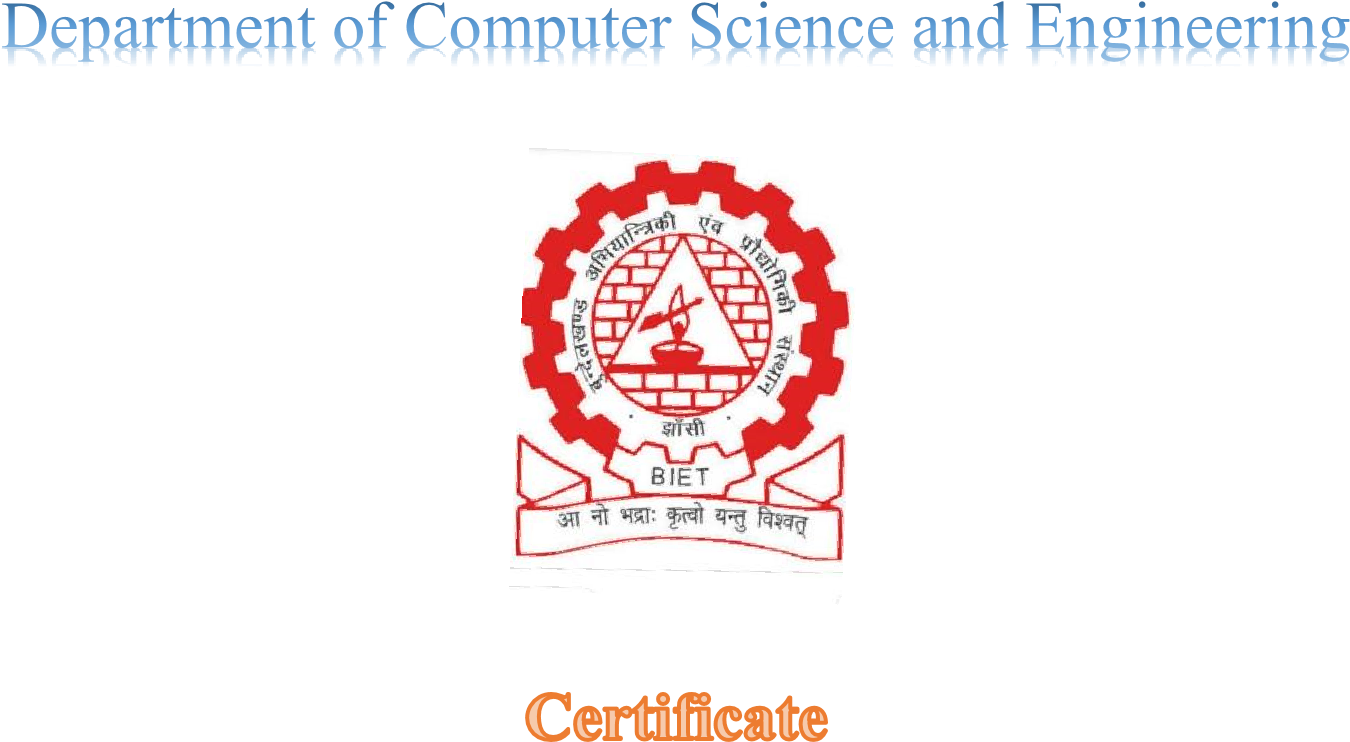
**Jhansi (U.P.) India – 284128**

**Submitted By: Submitted To:**

Avinash Kumar Prajapati (1904310017) Prof. A.K. Solanki

**BUNDELKHAND INSTITUTE OF**

**ENGINEERING AND TECHNOLOGY**



Certified that this is a bonafied record of the project entitiled “Document Scanner” completed successfully by **Avinash kumar Prajapati (1904310017)** of the 7th semester, Computer Science and Engineering, under the guidance of respected **Prof. A.K. Solanki** during this academic year 2022-23.

**Verified by Guide:**

**…………………………..**

**Prof. A.K. Solanki**

**(Depatment of Computer Science and Engineering) ACKNOWLEDGEMENT**

We are deeply indebeted to our respected Head of the department **Prof. A.K. Solanki** for guiding us .The team is also grateful to our project guide, **Er. Umakant Ahirwar** and respected faulty member **Dr. Yashpal Singh, Dr. R.N. Verma, Dr. S.K. Gupta, Er. Shashank Gupta, Er. Shivam Pandey, Dr. Shailendra Pratap** and **Er. Anuj Kumar Yadav** and for their indomitable contributionand guidance without which the completion of this project would have been possible.

Our sincerest thanks to all teachers, seniors and colleagues whose help and guidance brought this project to successful completion.

**Submitted By:**

Avinash Kumar Prajapati (1904310009)

#### TABLE OF CONTENTS

**SR.NO. CONTENT PAGE NO.**

1. Introduction……………………..……………………………………………………. 5
2. Objective………………………….……………………………………….....………..5
3. Scope of the Project…………….....……………………………….……….…..……...6
4. Overall Description…………...….....…………………………….………….….….6 - 8

4.1.Hardware and Software required…...………………...……………….……….…..6

4.2.Summary of the Project…………………...……………………...…….…………..7

4.3.Advantage of the Project…………………………….……………………………..7

1. Coding…………………………………………………………………………….8 - 15
2. Screenshots …...…………………………………………………………………16 – 23
3. Testing……………………..…………………………………………………….24 - 25
   1. Unit of Testing……………..…………………………………………………….24
   2. Test for Final Output…………………...…………………………………………24 7.3 Test for What we got finally.…………..……………………………………25
4. Conclusion……………………………………………………………………………25
5. Reference……………………………………………………………………………..26

# 1. INTRODUCTION

A developer platform where user can ask problems regarding development and technologies. Although there are many platforns that provides these services but still they become very complex and all the answer are becoming old fashioned that is out of version. The main goal of our platform is to provide most recent and most updated version of solution :

What can a user do?

* Ask Questions.
* Search for solution.
* Save the solution for future .
* Like the solution and question also .
* Report any invalid solution ..

# 2.OBJECTIVE

While developing this platform we try to resolve and achieve the following problems and goals.

* How should I make a system that we can also be friend with the other developer..
* We had used django-friendship package after it’s modification..
* Providing a relaxing and professional UI (Neomorphic UI ).
* We also provide a system for moderation over the comment solutions.
* Anyone can report the commets if the comments are too irrevalent of their question.
* You have your own profile where you can see your all writen questions.

# 3. SCOPE OF THE PROJECT

* It’s totally open-source and It’s main feature is you can be friend with someone who has solved your problem.
* You can also block someone if that user is irretative.
* You can write your own solutions for othe user’s question asked.

#### 4. OVERALL DESCRIPTION

Our mechanism converts the randomly taken images of the documents in any shape to get the only the desired part with no loss in the data. The output images have enhanced quality. We can also scan any noisy document to get the clear and sharp image which can be printed. It is also equipped with the window to set the min and max threshold values.

## 4.1. HARDWARE AND SOFTWARE REQIUREMENTS

SOFTWARE REQUIREMENTS

* Operating system- Windows 10 and above is used as the operating system as it is stable and supports more features and is more user friendly.
* Python is used to write the whole controls code.
* Use of Django Web Framework for website.
* Use of django-allauth for authentication
* Use of django-friendship for follow/unfollow system.
* Css for Neomorphic design..

HARDWARE REQUIREMENTS

* Intel core i5 11th and above generation processor is used because it is faster than other processors and provide reliable and stable and we can run our pc for longtime. By using this processor we can keep on developing our project without any worries.
* Ram minimum of 4gb is used as it will provide fast reading and writing capabilities and will in turn support in smooth processing.

## 4.2. SUMMARY OF THE PROJECT

1. Creating Profile.
2. Asking questions and making your profile more professional.
3. Getting followup or follow other users..
4. Get the Most relevent solution.
5. Like the solution if it’s valid or dislike if it’s invalid.
6. Save the posts to bookmark..
7. Share the post.

## 4.3 ADVANTAGE OF THE PROJECT

* We can collaborate with other developers.
* We can write our own solutions and post it and also can write the experiments.
* We can block the spammers
* We can also chat with others in comment system.

###### 5. COD and Website: ( myproject1.pythonanywhere.com )

**Manage.py file which manages the app server:**

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks.""" import os import sys

def main():

"""Run administrative tasks.""" os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'wandoc.settings')

try:

from django.core.management import execute\_from\_command\_line except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

) from exc

execute\_from\_command\_line(sys.argv) if \_\_name\_\_ == '\_\_main\_\_':

main()

**Model for the Web project:** from django.conf import settings

# Create your models here.

from django.contrib.auth.models import AbstractUser, User

from django.db import models from taggit.managers import TaggableManager from django.urls import reverse import uuid import random from django.db import IntegrityError

user = settings.AUTH\_USER\_MODEL

class customUser(AbstractUser):

user\_identity = models.CharField(max\_length=255 ,editable = True, unique=True) bio = models.CharField(max\_length=255, help\_text='Write about yourself in 200 charectors', null=True,blank=True) social\_link = models.CharField(max\_length=255, help\_text='Enter your approach link.' , null=True,blank=True) profile\_pic = models.ImageField(default='Capture.JPG') is\_private = models.BooleanField(default=False)

def get\_absolute\_url(self):

return reverse('wandocApp:userProfilePage\_other', args=[self.user\_identity])

def save(self, \*args, \*\*kwargs):

success = False while not success:

u\_t = str(uuid.uuid4()) if customUser.objects.filter(user\_identity = u\_t):

print(customUser.objects.filter(user\_identity = u\_t)) success = False else:

super(customUser, self).save(\*args, \*\*kwargs) success = True

class Post(models.Model):

STATUS\_CHOICES =(

('draft', 'Draft'),

('published', 'Published'),

)

id = models.AutoField(primary\_key=True) problem\_title = models.CharField(max\_length=255) # problem\_encountered = models.TextField( blank=True) problem\_detail = models.TextField()

author = models.ForeignKey(customUser, on\_delete=models.CASCADE)

popular = models.BooleanField(default=False) recommended = models.BooleanField(default=False)

slug = models.CharField(max\_length=120)

tags = TaggableManager()

status =models.CharField(max\_length=10, choices=STATUS\_CHOICES, default='draft') updated\_date = models.DateTimeField(auto\_now=True) create\_date = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return self.problem\_title

def get\_absolute\_url(self):

return reverse('wandocApp:detailpage', args=[self.slug, self.author.user\_identity])

**Urls settings:**  from django.urls import path, include from .views import ( homePage, userPage, userProfilePage, writeQuery, editprofile, detailpage, userProfilePage\_other, reciepent\_user, accept\_user,

reject\_user, cancel\_user, remove\_friend, friend\_management\_system, management\_cancel\_user, block\_friend, remove\_block, manage\_remove\_block, anonymousUserPage, cancel\_user\_on\_detail, reciepent\_user\_on\_detail,

)

app\_name = "wandocApp"

urlpatterns = [ path("", homePage, name="homepage"), path("accept\_user/<u\_tk>", accept\_user, name="accept\_user"), path("profile/<u\_token>/", userProfilePage\_other, name="userProfilePage\_other"), path("anonymous-user-page", anonymousUserPage, name="anonymoususerpage"), path("block\_friend/<u\_tk>", block\_friend, name="block\_friend"), path("cancel\_user/<u\_tk>", cancel\_user, name="cancel\_user"),

path(

"cancel\_user\_on\_detail/<u\_tk>/<slug>", cancel\_user\_on\_detail, name="cancel\_user\_on\_detail",

),

path(

"friend-management-system/", friend\_management\_system, name="friend\_management\_system",

),

path(

"management\_cancel\_user/<u\_tk>", management\_cancel\_user, name="management\_cancel\_user",

),

path("manage\_remove\_block/<u\_tk>", manage\_remove\_block, name="manage\_remove\_block"), path("<slug>/<u\_tk>/", detailpage, name="detailpage"),

path(

"reciepent\_user\_on\_detail/<u\_tk>/<slug>", reciepent\_user\_on\_detail, name="reciepent\_user\_on\_detail",

),

path("reject\_user/<u\_tk>", reject\_user, name="reject\_user"), path("remove\_block/<u\_tk>", remove\_block, name="remove\_block"), path("remove\_friend/<u\_tk>", remove\_friend, name="remove\_friend"), path("requesting/<u\_tk>", reciepent\_user, name="reciepent\_user"), path("write-query/", writeQuery, name="writeQuery"), path("user-page", userPage, name="userscreen"), path("user-profile", userProfilePage, name="userprofilepage"), path("user-profile-edit/", editprofile, name="editprofile"),

]

**Settings.py file which manages the whole project.**

"""

Django settings for wandoc project.

Generated by 'django-admin startproject' using Django 3.2.

For more information on this file, see https://docs.djangoproject.com/en/3.2/topics/settings/

For the full list of settings and their values, see https://docs.djangoproject.com/en/3.2/ref/settings/

"""

import os from pathlib import Path from django.contrib.messages import constants as messages

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-z2!^)d1csl^x-l%i0cycr3n+s\_=ef1^#8%oui&zyked23i!\*!u'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'django.contrib.sites',

'django\_comments\_xtd',

'django\_comments',

'taggit',

'django\_summernote',

'wandocApp',

# 'friend',

'friendship',

'allauth',

'allauth.account',

'allauth.socialaccount',

]

X\_FRAME\_OPTIONS = 'SAMEORIGIN'

# SUMMERNOTE\_THEME = 'bs4'

SUMMERNOTE\_CONFIG = {

'width': '100%',

'height': '500px',

'iframe': True,

'airMode': True,

'disableDragAndDrop': True,

'summernote': {

'toolbar': [

['style', ['style']],

['font', ['bold','italic', 'underline', 'clear','strikethrough', 'superscript', 'subscript']],

['fontname', ['fontname']],

['fontsize', ['fontsize',]],

['color', ['color']],

['para', ['ul', 'ol', 'paragraph']],

['table', ['table']],

['insert', ['link', 'picture', 'video','hr']],

['view', ['undo','redo','fullscreen', 'codeview', 'help']],

],

},

}

SITE\_ID = 1

MESSAGE\_TAGS = {

messages.ERROR: 'danger'

}

# comment-xtd

COMMENTS\_APP = 'django\_comments\_xtd'

EMAIL\_HOST = "smtp.gmail.com"

EMAIL\_PORT = "587"

EMAIL\_HOST\_USER = 'aviavi61106@gmail.com'

EMAIL\_HOST\_PASSWORD = '@$df@3183\*vinash#kumar'

EMAIL\_USE\_TLS = True

DEFAULT\_FROM\_EMAIL = "Helpdesk <helpdesk@yourdomain>"

COMMENTS\_XTD\_CONFIRM\_EMAIL =False

# To help obfuscating comments before they are sent for confirmation. COMMENTS\_XTD\_SALT = (b"Timendi causa est nescire. "

b"Aequam memento rebus in arduis servare mentem.") # Source mail address used for notifications.

COMMENTS\_XTD\_FROM\_EMAIL = "aviavi61106@gmail.com"

# Contact mail address to show in messages.

COMMENTS\_XTD\_CONTACT\_EMAIL = "helpdesk@example.com"

COMMENTS\_XTD\_APP\_MODEL\_OPTIONS = {

'wandocApp.post': {

'allow\_flagging': True,

'allow\_feedback': True,

'show\_feedback': True,

# 'who\_can\_post': 'all' # Valid values: 'all', users' }

}

COMMENTS\_XTD\_MAX\_THREAD\_LEVEL = 3 # default is 0

COMMENTS\_XTD\_LIST\_ORDER = ('-thread\_id', 'order') # default is ('thread\_id', 'order')

AUTHENTICATION\_BACKENDS = [

# Needed to login by username in Django admin, regardless of `allauth` 'django.contrib.auth.backends.ModelBackend',

# `allauth` specific authentication methods, such as login by e-mail

'allauth.account.auth\_backends.AuthenticationBackend',

]

LOGIN\_REDIRECT\_URL = "wandocApp:userprofilepage"

# ACCOUNT\_ADAPTER = 'wandocApp.adapter.MyAccountAdapter'

MIDDLEWARE = [

'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware',

'django.middleware.common.CommonMiddleware',

'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddleware',

'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'wandoc.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

'APP\_DIRS': True,

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

WSGI\_APPLICATION = 'wandoc.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

# Password validation

# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

{

'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

},

{

'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

},

]

# Internationalization

# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC\_URL = '/static/'

# Default primary key field type

# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, 'static'),

]

STATIC\_URL = '/static/'

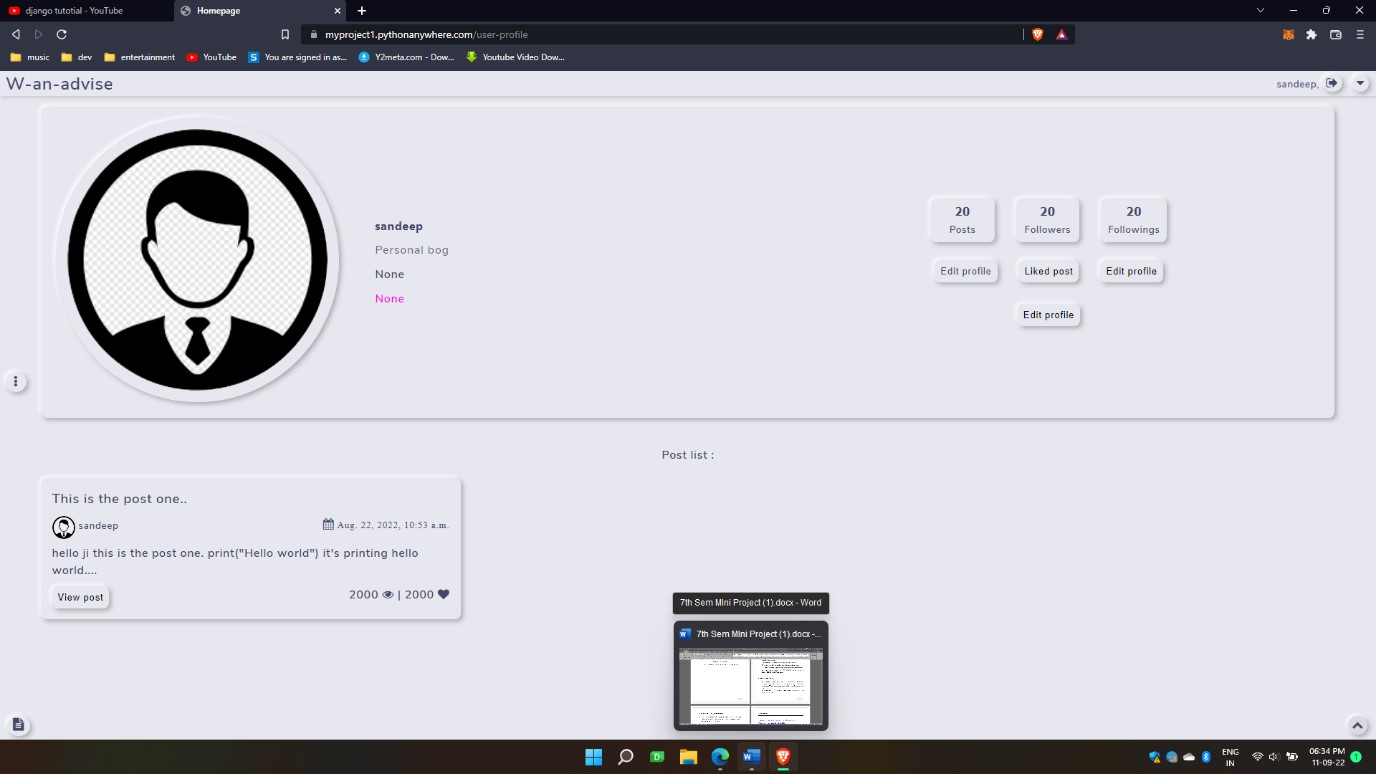
MEDIA\_URL = '/media/'

STATIC\_ROOT = os.path.join(BASE\_DIR, 'static\_cdn')

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media\_cdn')

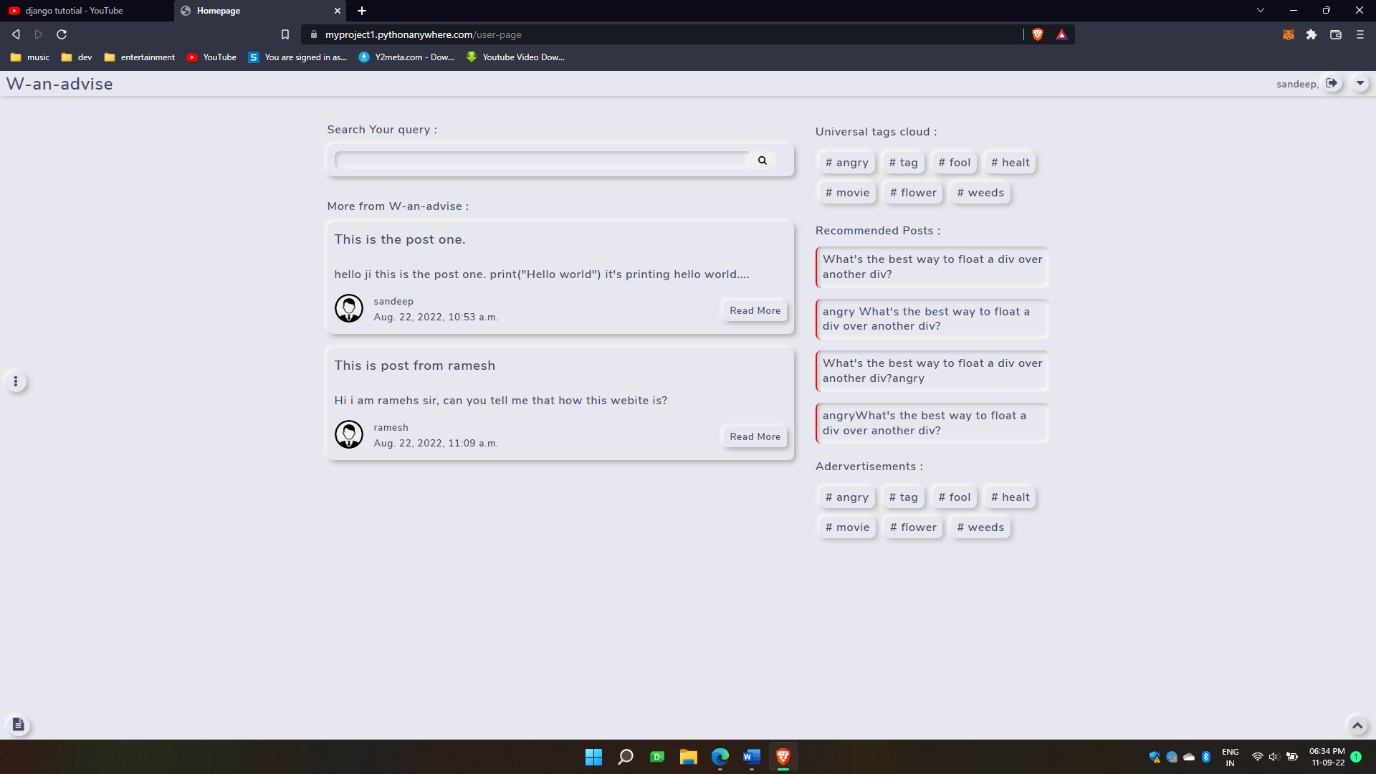
AUTH\_USER\_MODEL ="wandocApp.CustomUser" ---------------------------------------------------------------------------------------------------------------

# 6. SCREENSHOTS



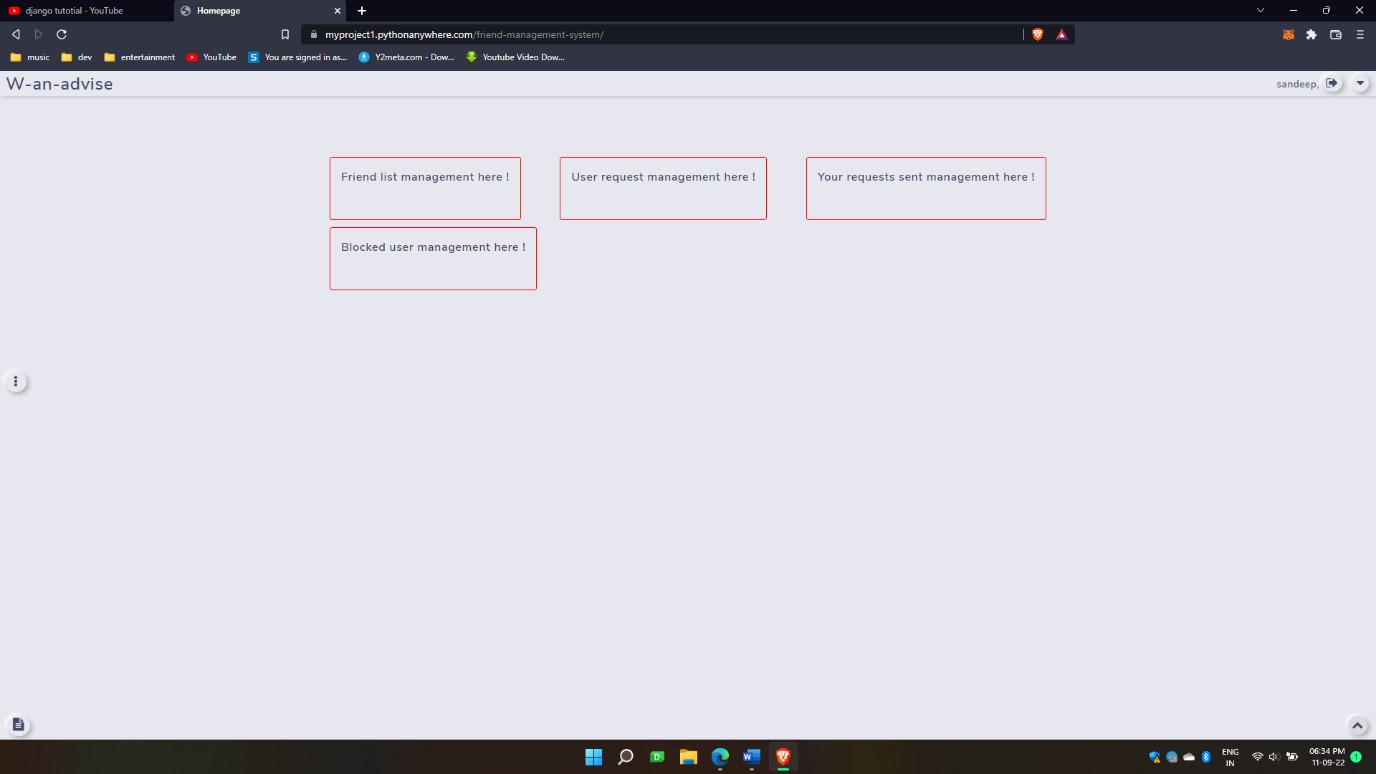
**FIG 6.1** Original Image

In Fig 6.1 we have illustrated the profile page for user



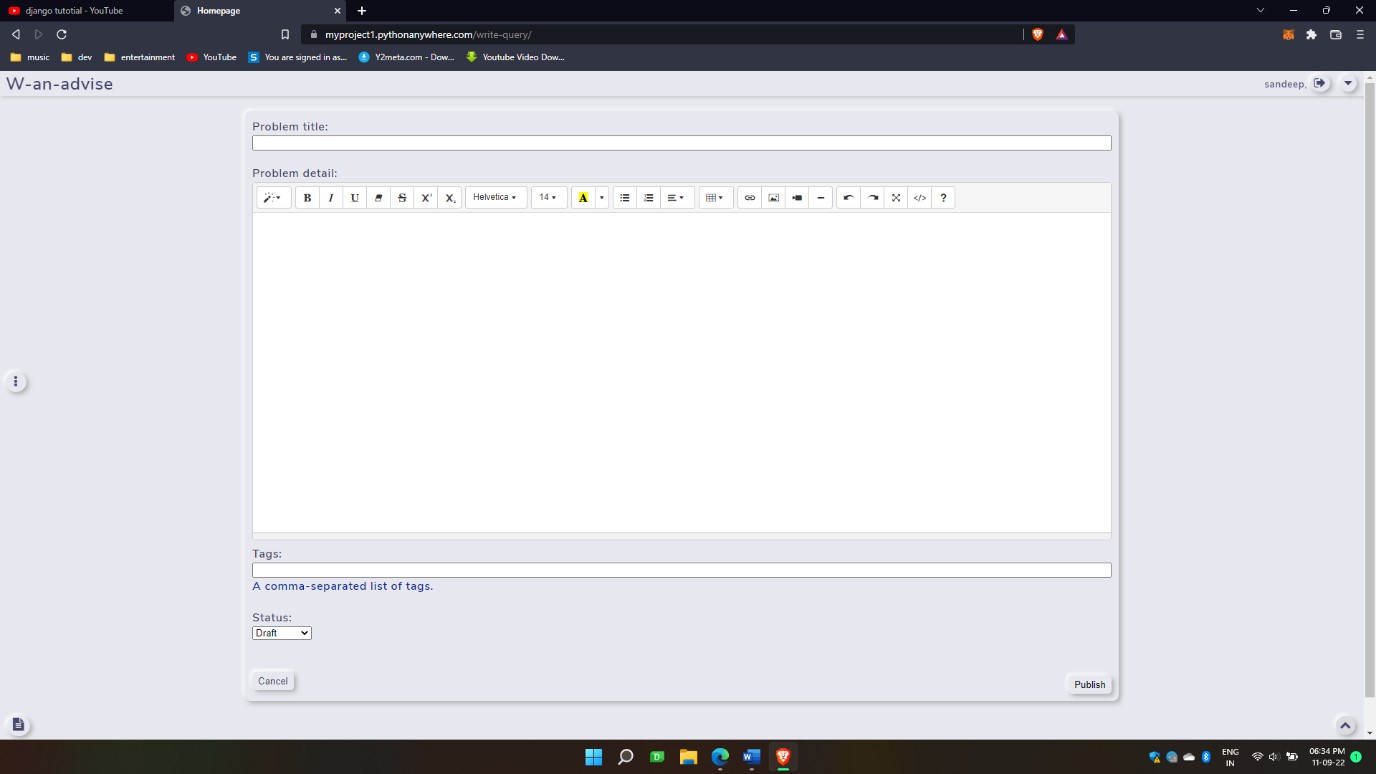
**FIG 6.2** Exloring the posts.

Fig 6.2 Displays the exploring the post which is latest..



**FIG 6.3** Connections

In Fig 6.3 It show the management of connections with other people.



**FIG 6.4** Writing the post

Fig 6.4 We can write the post also.

# 7. TESTING

The aim of the system testing process was to determine all defects in our project .The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not.Our Project went through two differnent levels of testing.

##### 7.1 Unit Testing

Unit testing is done on each of the following intermediate stages.

* **Creating user successful**
* **Logout successful**
* **Login successful**
* **Writing post succcessful**
* **Sending friend request successful**
* **Blocking the annoying user successful**

### 7.2 Test for Final Output

• **Test for Getting the suitable Page detail** – Designing the website whiv=ch trract user with no flaws are also has been taken care of.

### 7.3 Test for Different Ranges of the Images

• **Test for Multiple Stages** – We have taken perform the all acions and no eroors are found at any page and you can easly create an your account for free.

# 8.CONCLUSION

In this current time collaboration with people like developer and engineers are very important things and it can lead new ideas and technologies and inovation.

#### 9. REFERENCE

1. [Django documentation | Django documentation | Django (djangoproject.com)](https://docs.djangoproject.com/en/4.1/)
2. <https://www.youtube.com/watch?v=rHux0gMZ3Eg>