ONLINE STUDY NOTES PORTAL
PROJECT REPORT

Submitted by:-HEMA PRABHA M

Submitted to:-Reny miss

CLIENT REQUIREMENTS

- Registration :- registration of user , admin ,restaurant owner/staff for using foodordering website
- Login:-The customer and admin, restaurant owner/staff login to the system using user idand password for food ordering/adding/removing food items
- Add/Edit file:-User can add, edit, upload and delete files
- Edit profile:- User can edit their profile details
- Add to favorites:-User can add searched files to their favorite list
- Search :-User can search files which are required for them
- Notifications:-Notifications for user login , crud operations etc..

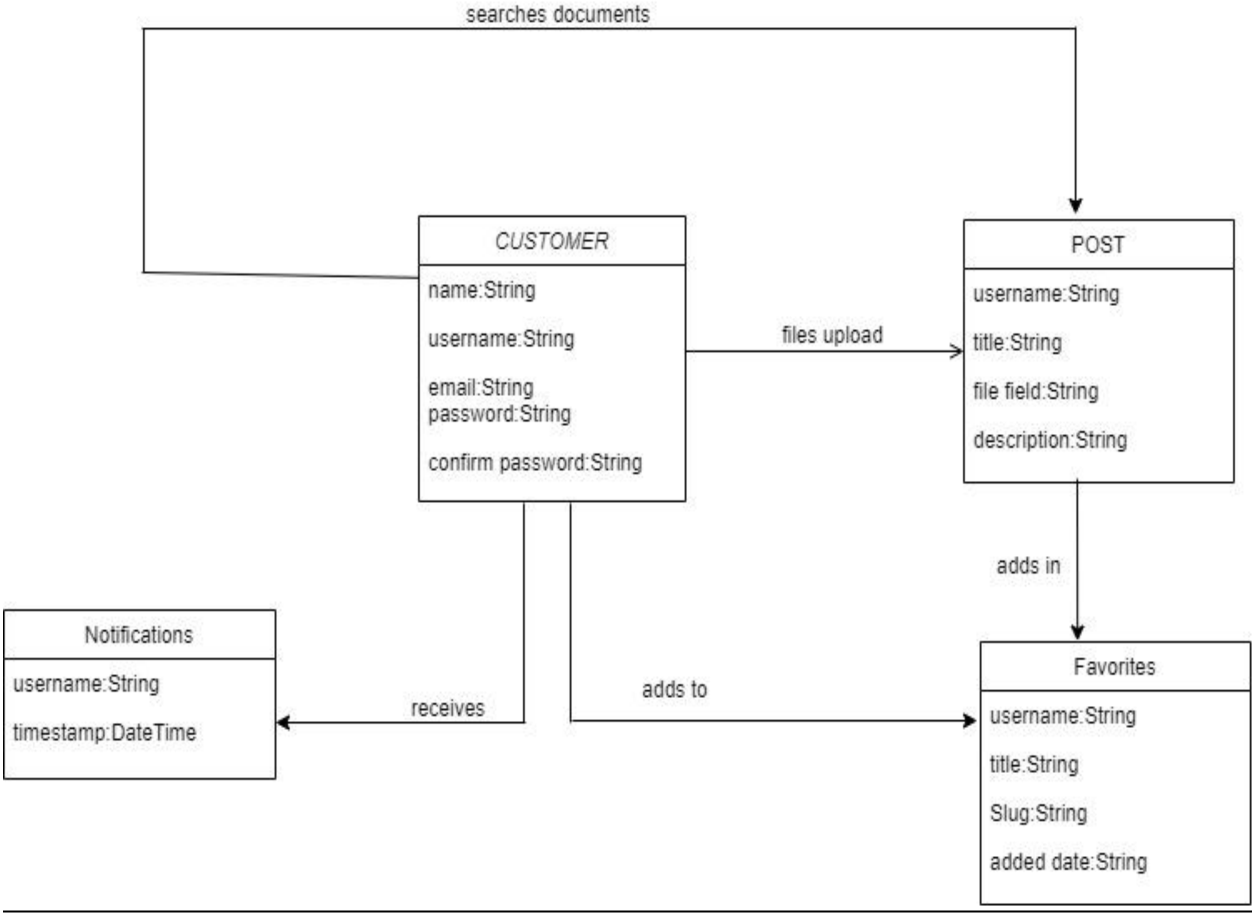
Githublink:- https://github.com/naughtygirlzz/features

MODELS.PY

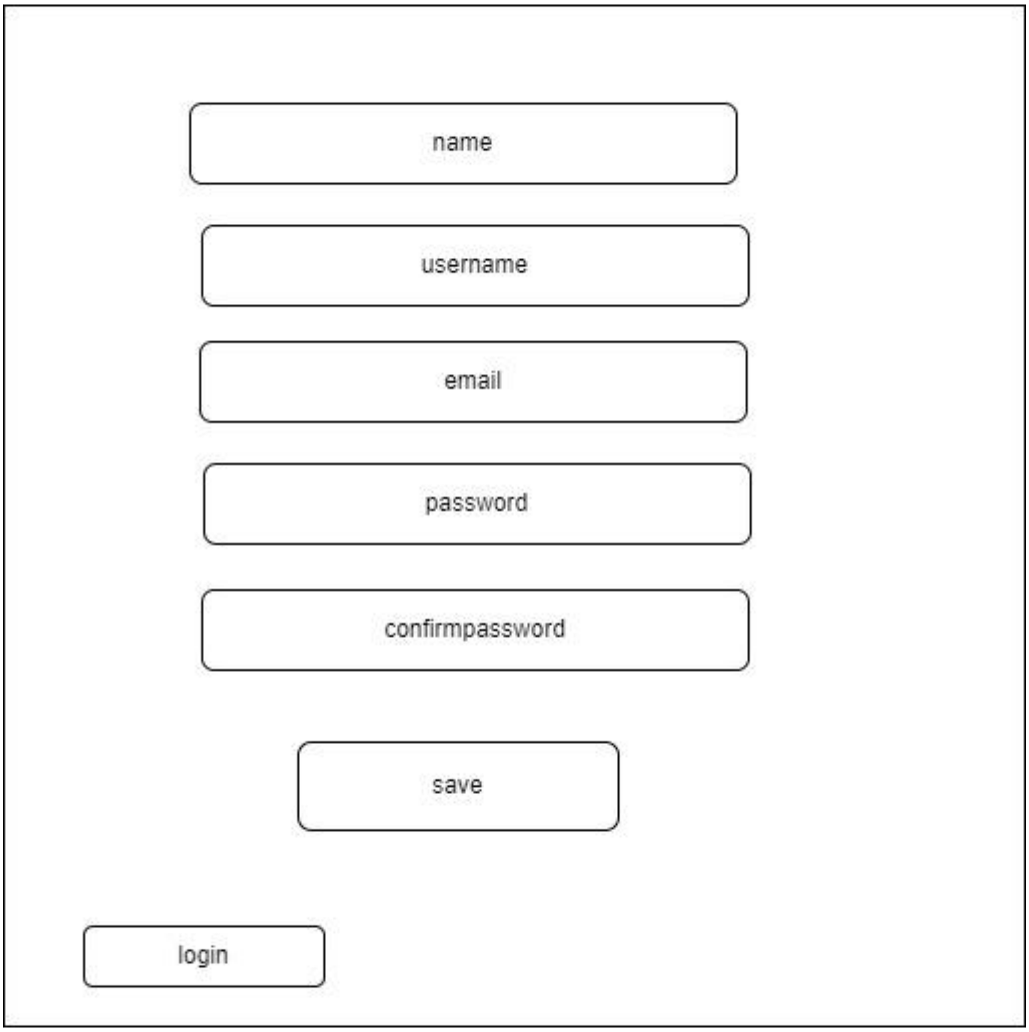
```
from django.db import models
from django.utils import timezone
from django.db.models import Q
from django.contrib.auth.models import User


# Create your models here.
class Customer(models.Model):
    name = models.CharField(max_length=80)
    username = models.CharField(max_length=80,primary_key=True)
    email = models.CharField(max_length=40)
    password = models.CharField(max_length=40)
    confirmpassword = models.CharField(max_length=40)




    def _str_(self):
        return self.username

#file upload class
class PostManager(models.Manager):
    def search(self, query):
        return self.get_queryset().filter(
            Q(title__icontains=query)
        )
```

```
class Post(models.Model):
    id = models.AutoField(primary_key=True)
    title = models.CharField(max_length=200,null=False)
    thumbnail=models.ImageField(upload_to='uploads/',null=True)
    pdf = models.FileField(upload_to="uploads/")
    desc = models.TextField()
    username = models.ForeignKey(Customer, on_delete=models.CASCADE)


    objects = PostManager()

    def _str_(self):
        return self.title



#favorites class
class Favorite(models.Model):
    username = models.ForeignKey(Customer, on_delete=models.CASCADE)
    post = models.ForeignKey(Post, on_delete=models.CASCADE)


    def _str_(self):
        return f"{self.username} - {self.post.title}"



class Notification(models.Model):
    is_read = models.BooleanField(default=False)
    timestamp = models.DateTimeField(auto_now_add=True)
```

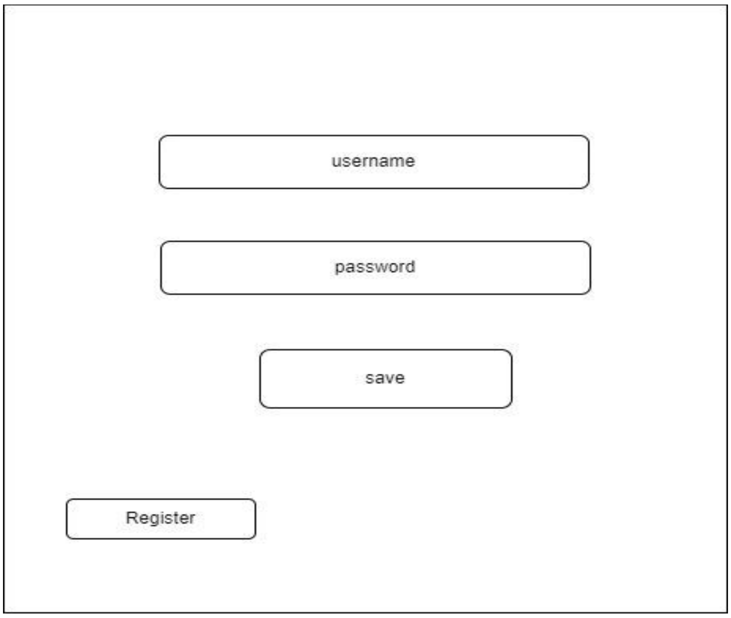
CLASS DIAGRAM



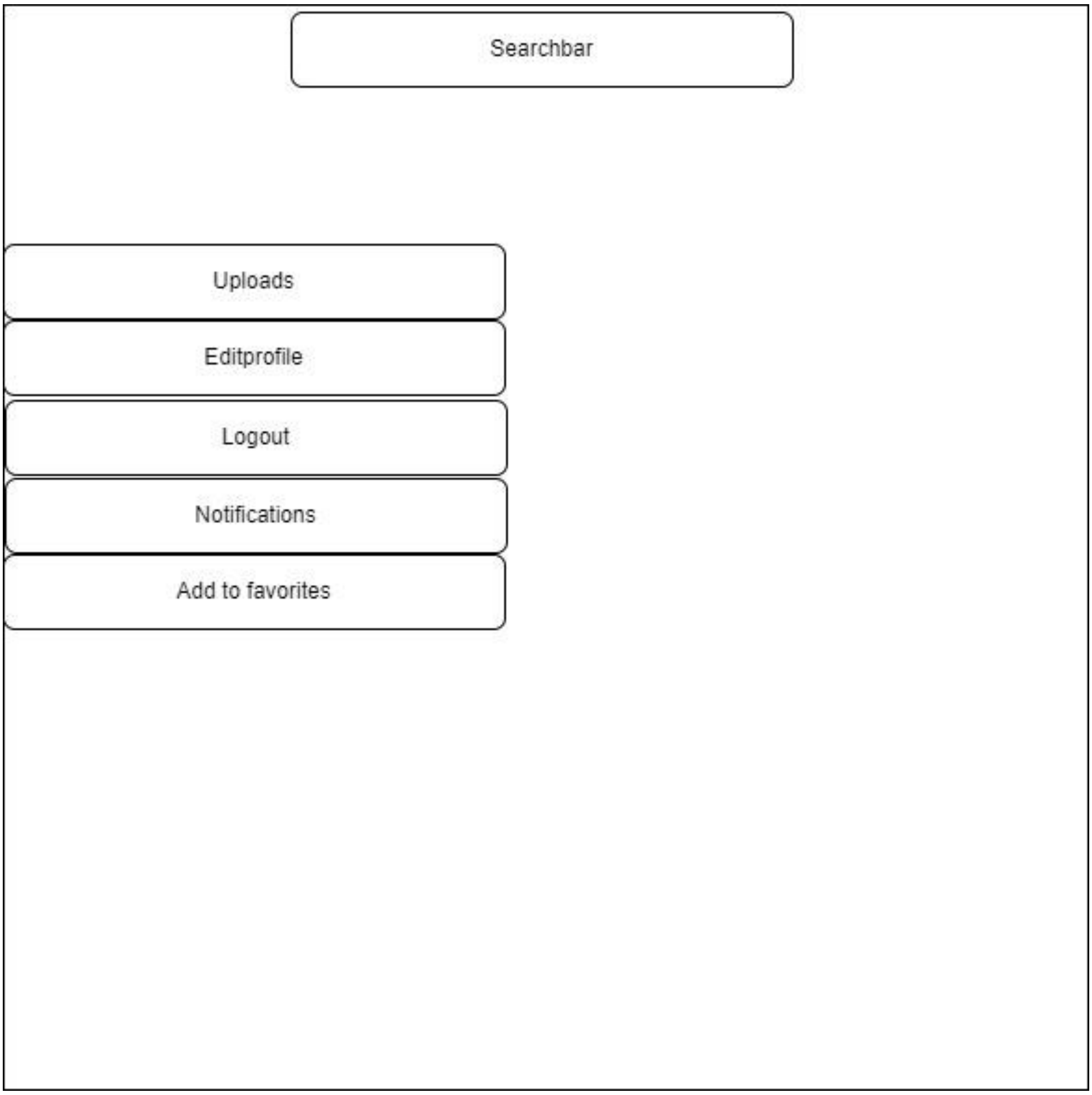
REGISTRATION FORM



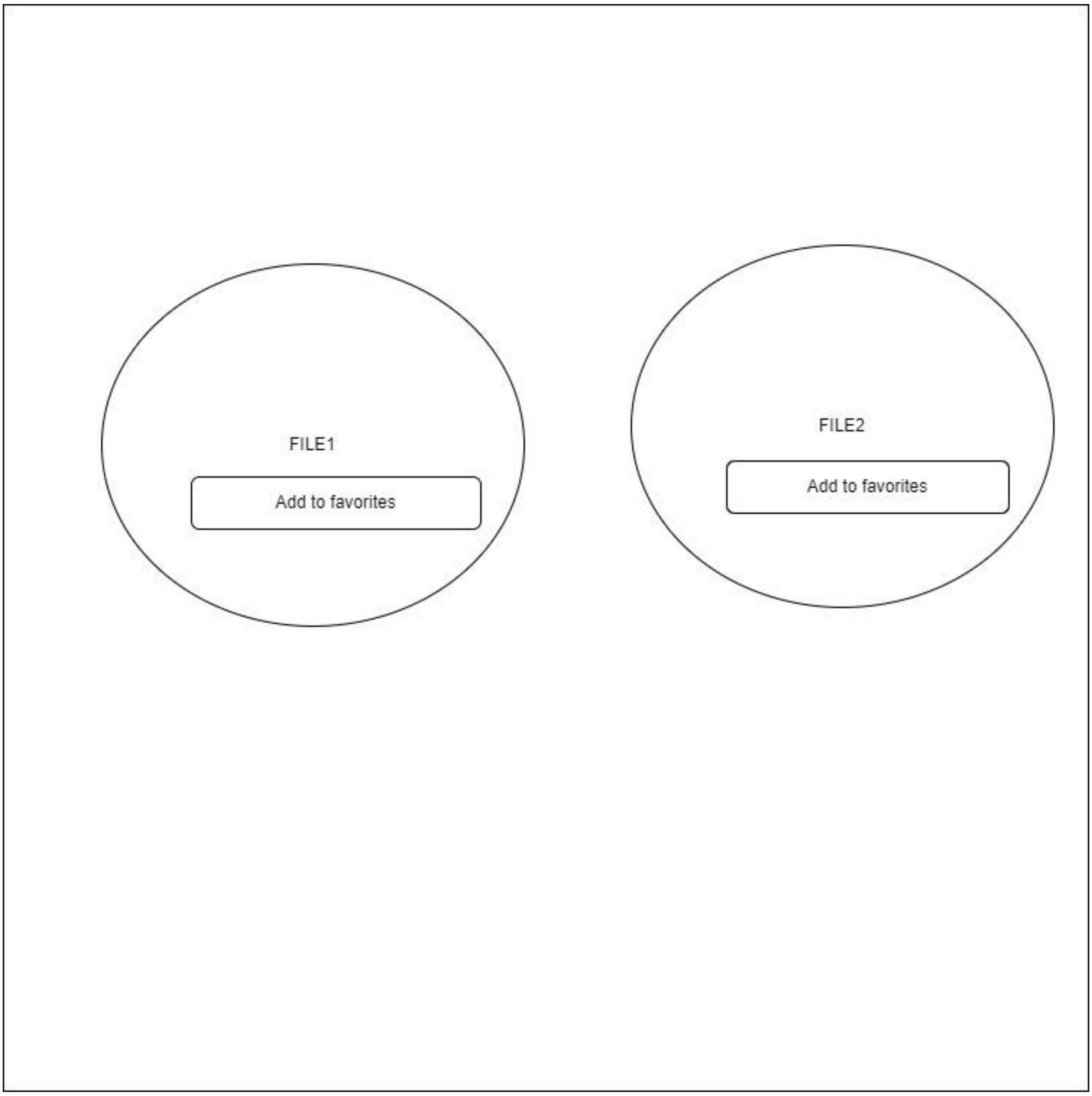
LOGIN FORM



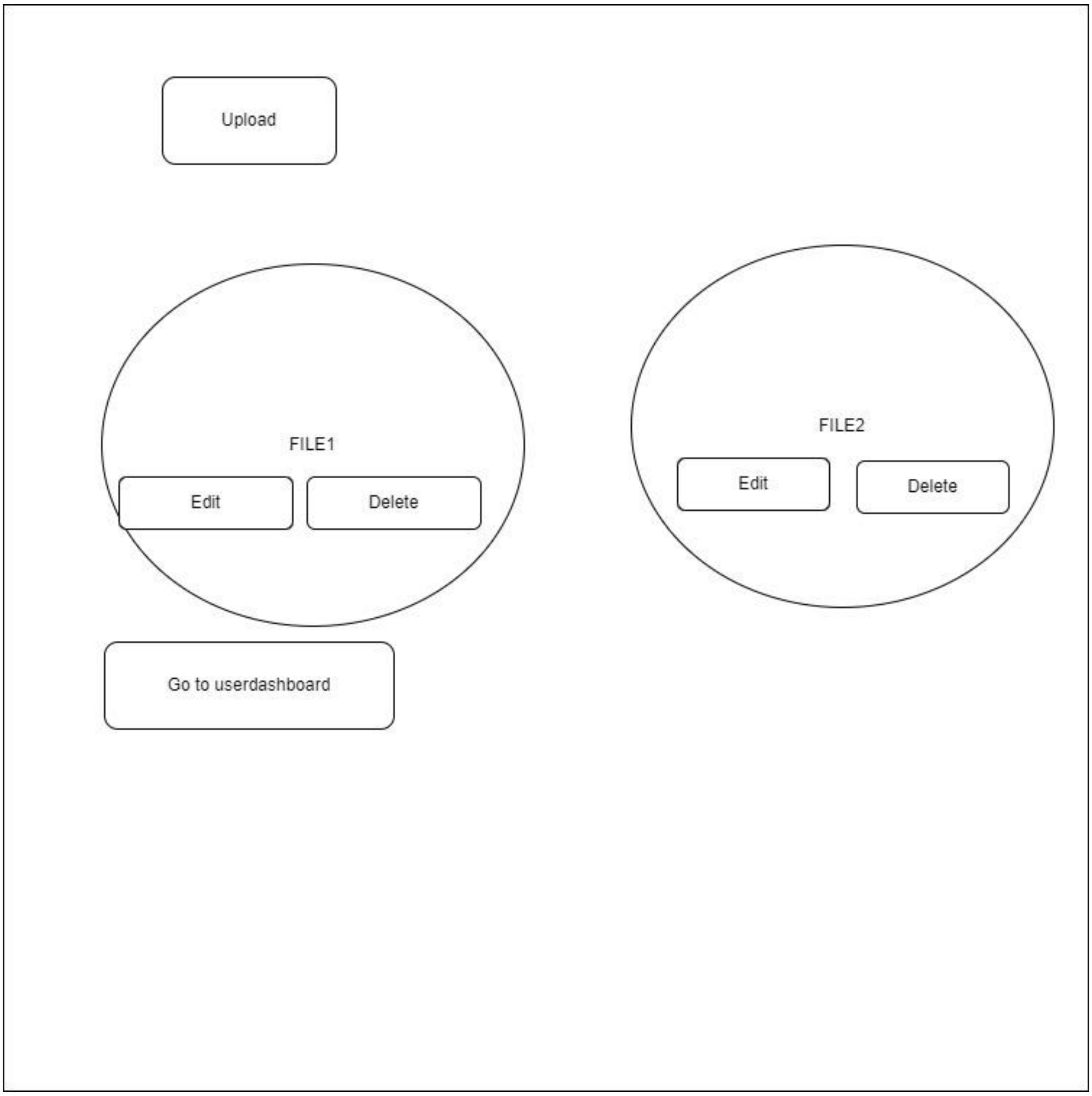
USER DASHBOARD



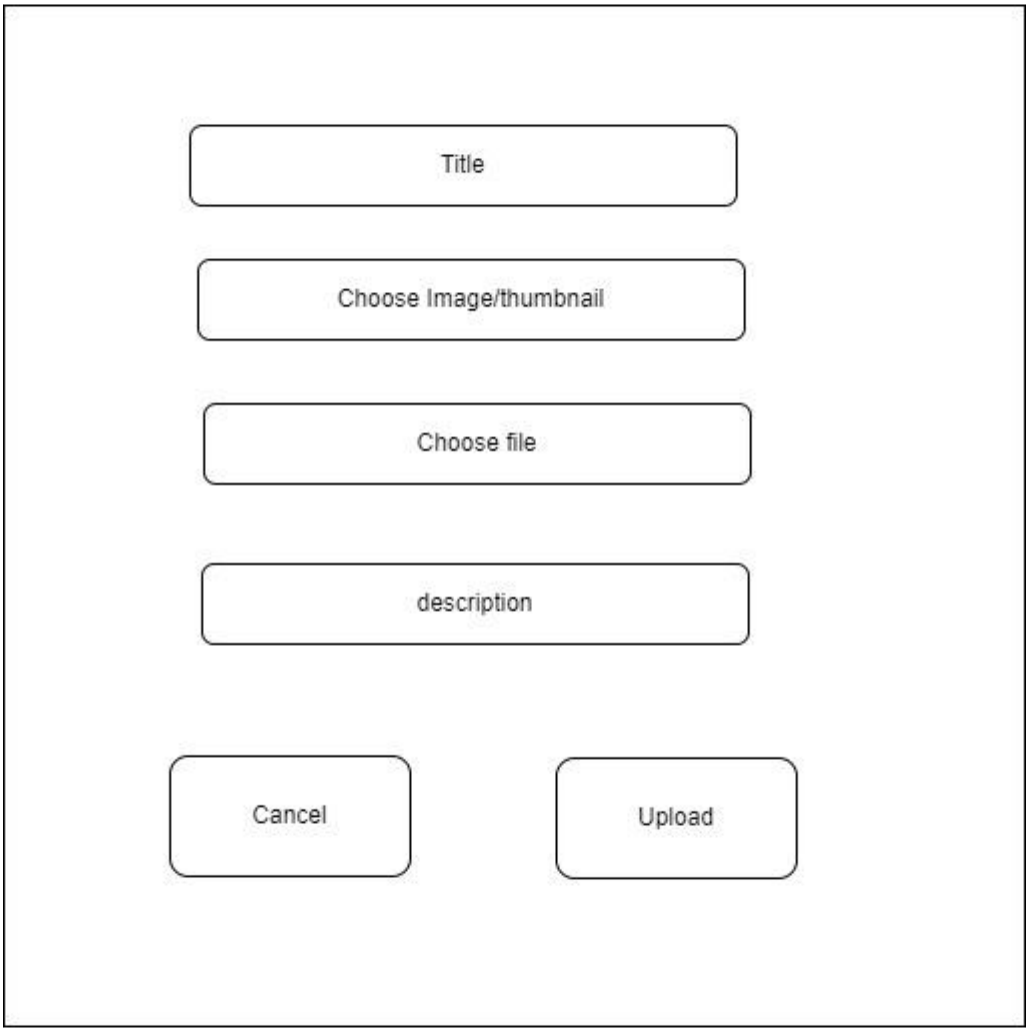
SEARCH



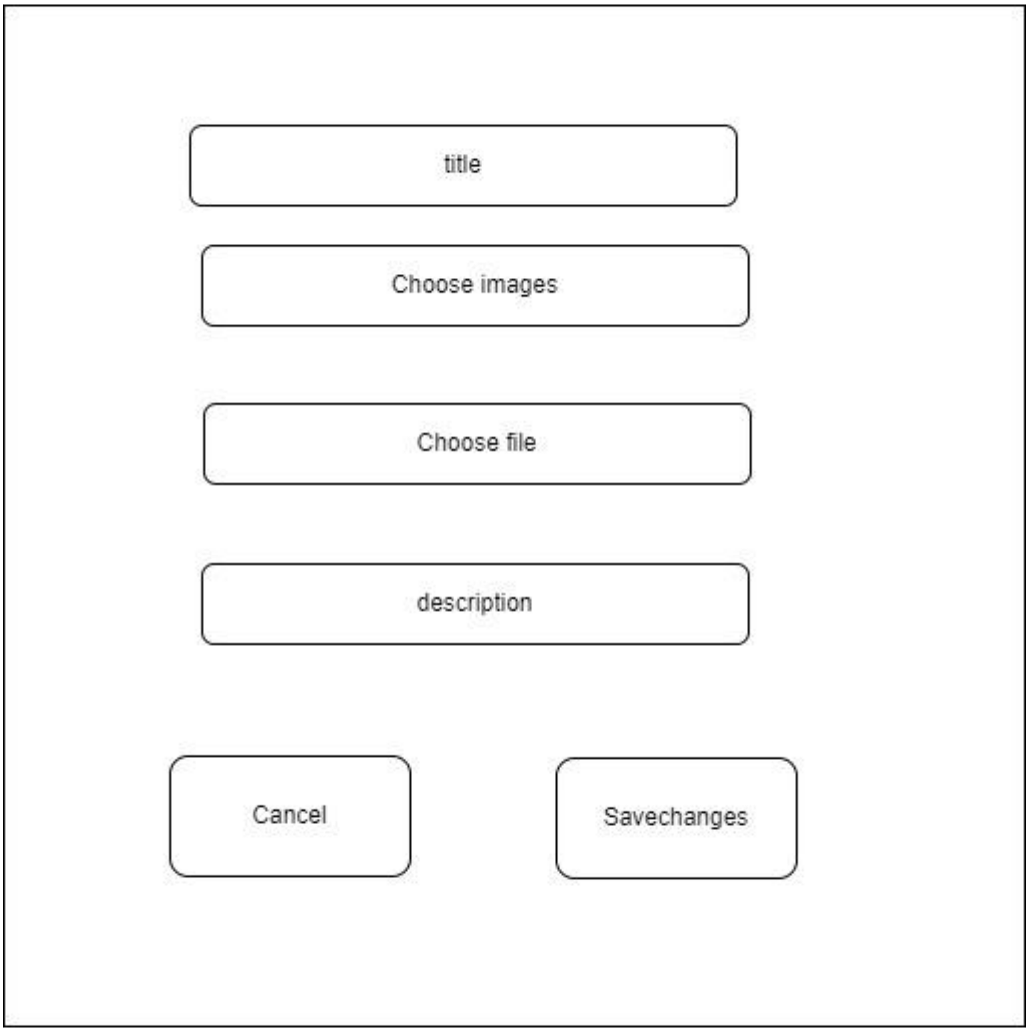
UPLOADDASHBOARD(UPLOADED FILES DISPLAY)



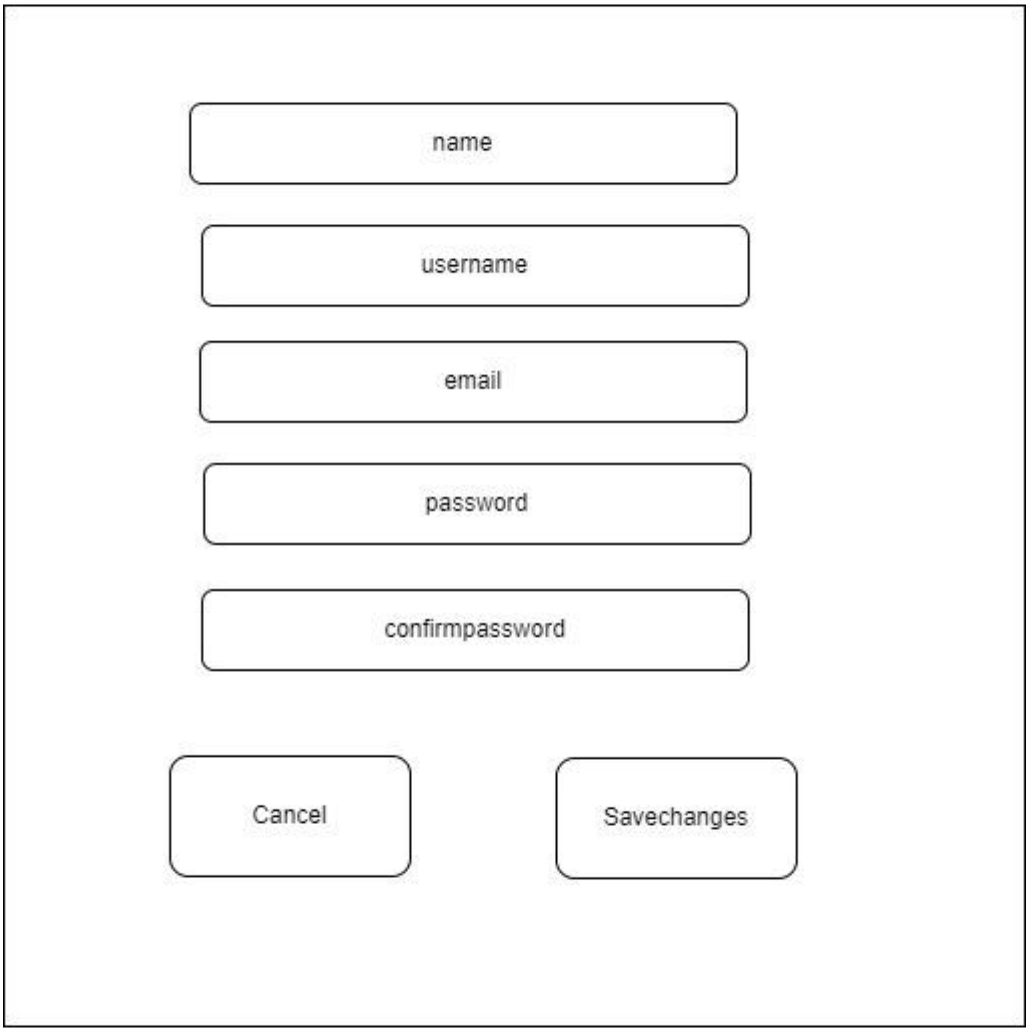
UPLOAD FORM


Title
Choose Image/thumbnail
Choose file
description
Cancel
Upload

EDIT FORM


title
Choose images
Choose file
description
Cancel
Savechanges

EDIT PROFILE FORM


name
username
email
password
confirmpassword
Cancel
Savechanges