Progetto Django

Feisbùcc

Antonio Emanuel Balas, Andrea Bertini, Damiano De Vivo, Nikita Piraino

Feisbùcc

Obiettivo, creare un social network:
dalla pagina di login alla gestione del database

GRUPPO 3

Feisbùcc > /usr/bin/python

Indice

Utilizzo

Modello E-R

Database e Pagina Login

Contenuti

Feisbùcc - Utilizzo

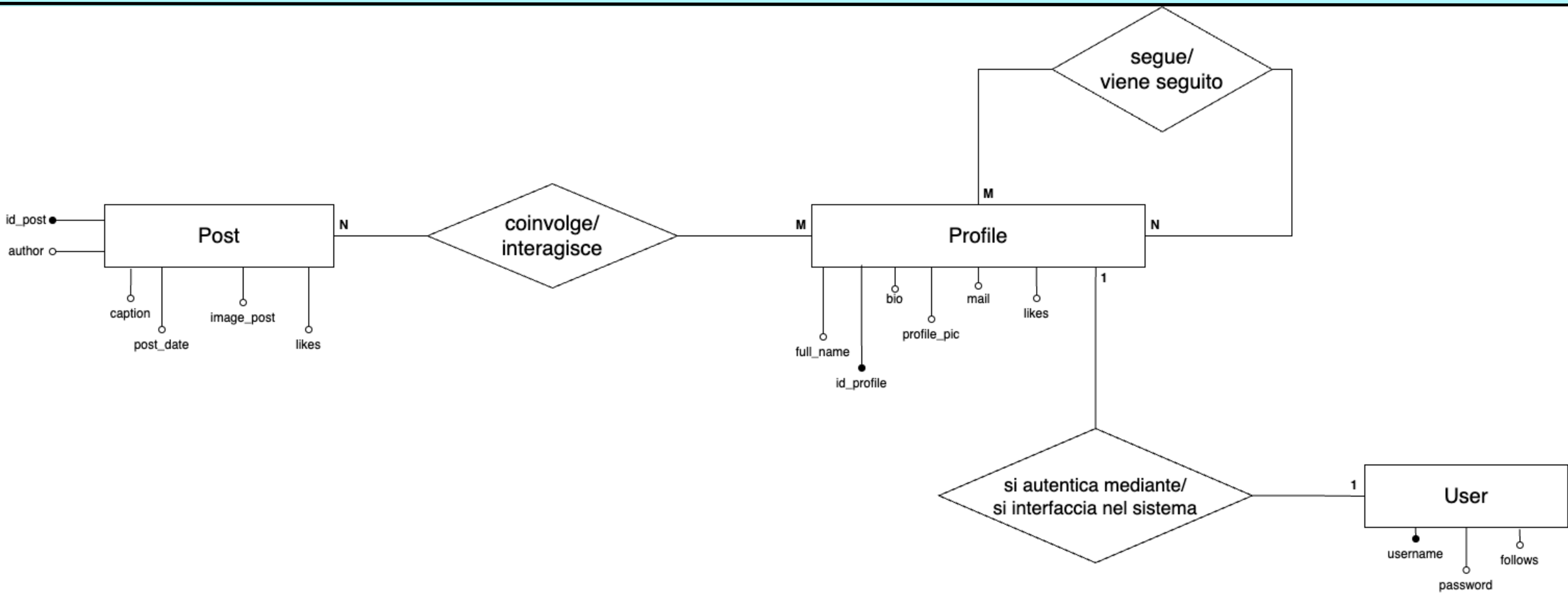
Funzionalità disponibili

Creazione di post

Interazione con i post degli altri utenti

Personalizzazione del profilo

Feisbùcc - Modello E-R



Feisbùcc - Database e Login

```
class SignupForm(UserCreationForm):

    username = forms.CharField(widget=forms.TextInput(attrs={'placeholder':'Username'}))
    password1 = forms.CharField(widget=forms.PasswordInput(attrs={'placeholder':'Password'}))
    password2 = forms.CharField(widget=forms.PasswordInput(attrs={'placeholder':'Repeat Password'}))

    class Meta:
        model = User
        fields = ('username', 'password1', 'password2')


class LoginForm(forms.Form):
    username = forms.CharField(widget=forms.TextInput(attrs={'placeholder':'Username'}))
    password = forms.CharField(widget=forms.PasswordInput(attrs={'placeholder':'Password'}))
```

forms.py: pagina di login

Feisbùcc - Contenuti

```
class Profile(models.Model):
    user = models.OneToOneField(User,on_delete=models.CASCADE)
    fullname = models.CharField(max_length=255,null=True, blank=True,default='vuoto')
    bio = models.TextField(blank=True)
    profile_pic = models.ImageField(default='profile/blank-profile-picture.png',upload_to='profile/')
    mail = models.EmailField(blank=True)
    follows = models.ManyToManyField(
        "self", related_name="followed_by", symmetrical=False, blank=True
    )

    def __str__(self):
            return str(self.user)

    def n_followere(self):
        return self.follows.count()

    def n_following(self):
        profiles = Profile.objects.all()
        n = 0
        for p in profiles:
            foll = p.follows.all()
            for k in foll:
                if k.user.username == str(self.user):
                    n = n+1
        return n-1
```

Codice models: gestione dei profili, gestione dei followers.

Feisbùcc - Contenuti

```
@receiver(post_save, sender=User)
def create_profile(sender, instance, created, **kwargs):
    if created:
        user_profile = Profile(user=instance)
        user_profile.save()
        user_profile.follows.add(instance.profile)
        user_profile.save()


class Post(models.Model):
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    caption = models.TextField(blank=True)
    post_date = models.DateField(auto_now_add=True)
    image_post = models.ImageField(null=True, blank=True, upload_to="images/")
    likes = models.ManyToManyField(User, related_name='blog_posts',blank=True)

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

    def get_profilePic(self):
        profiles = Profile.objects.all()
        for p in profiles:
            if p.user.username == str(self.author):
                return p.profile_pic

    def __str__(self):
        return str(self.author)
```

Continuo codice di models:
uso di @receiver, strumento che una volta ricevuti i contenuti di user li inserisce nel profilo.

Feisbùcc - Contenuti

```
from django.urls import path

from . import views

urlpatterns = [
    path('',views.feisbucc,name='home'),
    path('login/',views.user_login,name='login'),
    path('register/',views.register,name='register'),
    path('logout/',views.user_logout,name='logout'),
    path('addPost/',views.add_post,name='addPost'),
    path('profile/<profile_id>',views.profileV,name='profile'),
    path('profile/editProfile',views.edit_profile,name='editProfile')
]
```

urls.py: gestione URLS del social network, creando dei percorsi da una view all’altra

Grazie dell’Attenzione

GRUPPO 3