

ADF Practical – 6 Steps

- Submitted by: - Sanket Chheladiya(18bce042)

-Submitted To: - Prof. Dvijesh Bhatt

Q1:-

Demonstrate the concept of AbstractBased User and Custom model for the Social Media Application.

Models.py

```
from django.db import models
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager

class MyAccountManager(BaseUserManager):
    def create_user(self, email, username, password=None):
        if not email:
            raise ValueError('Users must have an email address')
        if not username:
            raise ValueError('Users must have a username')

        user = self.model(
            email=self.normalize_email(email),
            username=username,
        )

        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_superuser(self, email, username, password):
        user = self.create_user(
            email=self.normalize_email(email),
            password=password,
            username=username,
        )
        user.is_admin = True
        user.is_staff = True
        user.is_superuser = True
        user.save(using=self._db)
        return user

class Account(AbstractBaseUser):
    email = models.EmailField(verbose_name="email", max_length=60,
                              unique=True)
```

```

        username= models.CharField(max_length=30, unique=True)
        date_joined= models.DateTimeField(verbose_name='date joined',
auto_now_add=True)
        last_login= models.DateTimeField(verbose_name='last login',
auto_now=True)
        is_admin= models.BooleanField(default=False)
        is_active= models.BooleanField(default=True)
        is_staff= models.BooleanField(default=False)
        is_superuser= models.BooleanField(default=False)

        USERNAME_FIELD = 'email'
        REQUIRED_FIELDS = ['username']
        objects = MyAccountManager()

        def __str__(self):
            return self.email
        # For checking permissions. to keep it simple all admin have ALL
        permissions
        def has_perm(self, perm, obj=None):
            return self.is_admin
        # Does this user have permission to view this app? (ALWAYS YES FOR
        SIMPLICITY)
        def has_module_perms(self, app_label):
            return True

```

settings.py

```
AUTH_USER_MODEL = "CustomUserModel.Account"
```

admin.py

```

from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from CustomUserModel.models import Account

class AccountAdmin(UserAdmin):
    list_display = ('email' , 'username' , 'date_joined', 'last_login',
'is_admin', 'is_staff')
    search_fields = ('email' , 'username',)
    readonly_fields = ('date_joined','last_login')

    filter_horizontal = ()
    list_filter = ()
    fieldsets = ()

admin.site.register(Account, AccountAdmin)

```

Screenshots:

-Create Super User

Before

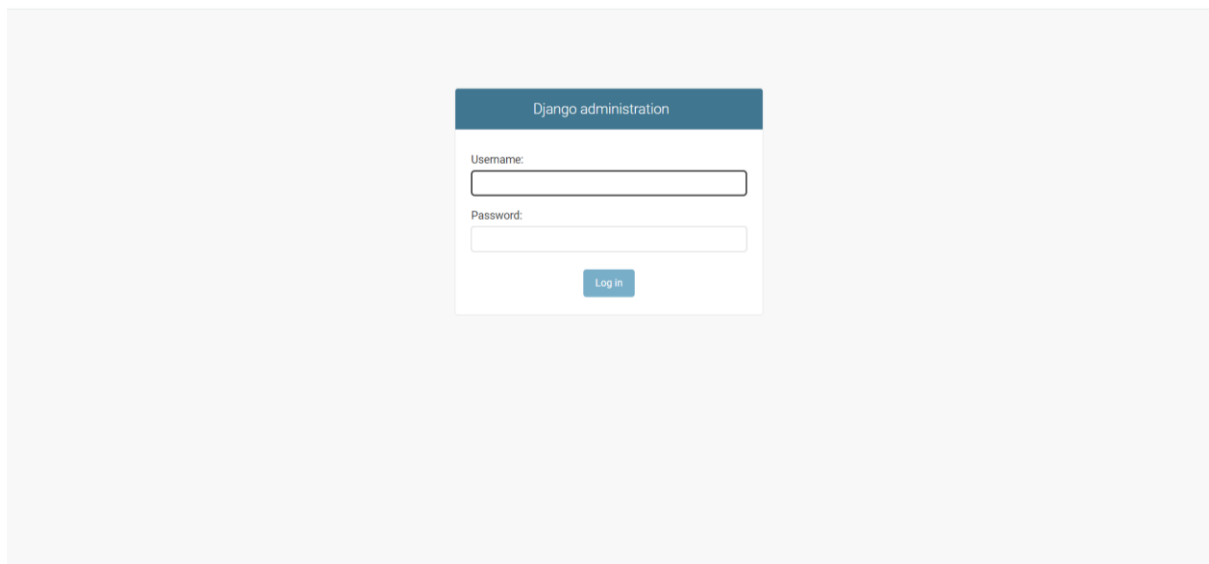
```
Username (leave blank to use 'admin'): xyz
Email address: xyx@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

After

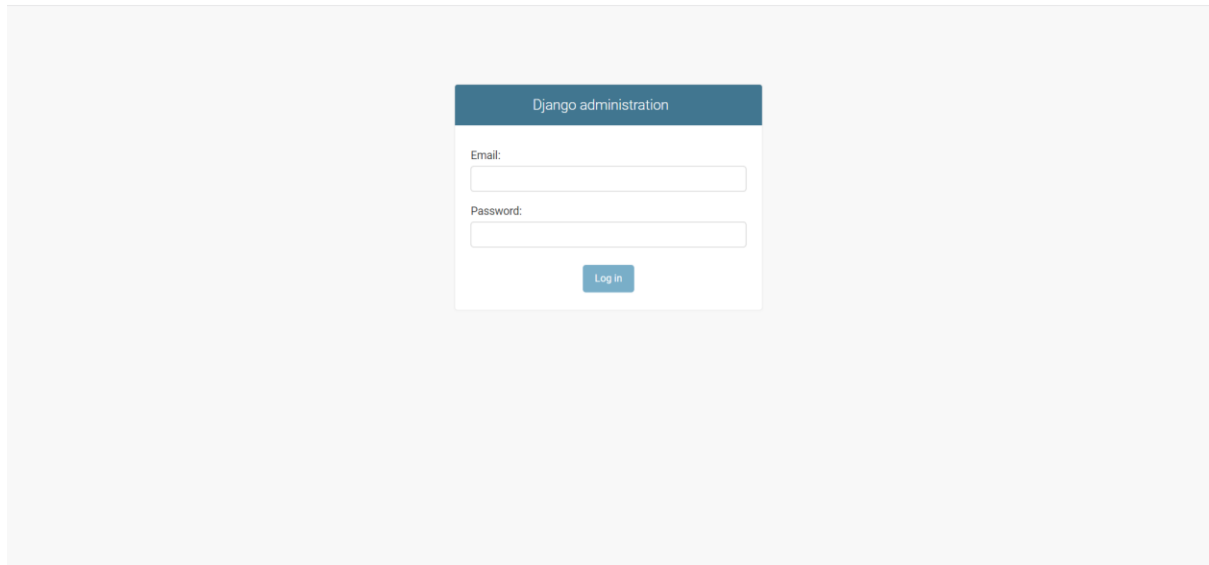
```
Email: asd@gmail.com
Username: asd
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

-Admin Login Page

Before



Before



-Admin panel before

Django administration

WELCOME, SANKET VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Authentication and Authorization > Users

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

USER

Posts + Add

User profiles + Add

Select user to change

Q

Search

Action: Go 0 of 4 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Sanket123	spatel19042001@gmail.com	Sanket	Chheladiya	<div></div>
<input type="checkbox"/>	Sanket543	sanketchheladiya19042001@gmail.com	Sanket	Patel	<div></div>
<input type="checkbox"/>	abc	abc@gmail.com			<div></div>
<input type="checkbox"/>	sanket	sanket@gmail.com			<div></div>

4 users

FILTER

By staff status

All

Yes

No

By superuser status

All

Yes

No

By active

All

Yes

No

ADD USER +

- AUTHENTICATION AND AUTHORIZATION
- Groups [+ Add](#)
 - Users [+ Add](#)
- USER
- Posts [+ Add](#)
 - User profiles [+ Add](#)

Change user

HISTORY

Username:

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

Password: **algorithm: pbkdf2_sha256 iterations: 260000 salt: 0lrvcg***** hash: IM/8Wg*******

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name: Last name: Email address:

Permissions

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☐ Staff status

Designates whether the user can log into this admin site.

☐ Superuser status

Designates that this user has all permissions without explicitly assigning them.

Groups:

Available groups

Filter

Choose all

Chosen groups

Remove all

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

Available user permissions

Filter

- admin | log entry | Can add log entry
- admin | log entry | Can change log entry
- admin | log entry | Can delete log entry
- admin | log entry | Can view log entry
- auth | group | Can add group
- auth | group | Can change group
- auth | group | Can delete group
- auth | group | Can view group
- auth | permission | Can add permission
- auth | permission | Can change permission
- auth | permission | Can delete permission
- auth | permission | Can view permission
- auth | user | Can add user

Choose all

Chosen user permissions

Remove all

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

Important dates

Last login:

Date: 2021-11-10 Today

Time: 16:13:05 Now

Note: You are 5.5 hours ahead of server time.

Date joined:

Date: 2021-11-10 Today

Time: 16:12:09 Now

Note: You are 5.5 hours ahead of server time.

Delete

Save and add another

Save and continue editing

SAVE

-After

Django administration

WELCOME: [ABC@GMAIL.COM](#) [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Customusermodel > Accounts

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

CUSTOMUSERMODEL

Accounts [+ Add](#)

Select account to change

[ADD ACCOUNT](#) [+](#)

Q

Search

Action: Go 0 of 2 selected

<input type="checkbox"/>	EMAIL	USERNAME	DATE JOINED	LAST LOGIN	IS ADMIN	IS STAFF
<input type="checkbox"/>	User@gmail.com	User	Nov. 11, 2021, 5:47 a.m.	Nov. 11, 2021, 5:47 a.m.	<div></div>	<div></div>
<input type="checkbox"/>	abc@gmail.com	abc	Nov. 12, 2021, 4:01 a.m.	Nov. 12, 2021, 4:10 a.m.	<div></div>	<div></div>

2 accounts

Django administration

WELCOME: [ABC@GMAIL.COM](#) [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Customusermodel > Accounts > abc@gmail.com

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

CUSTOMUSERMODEL

Accounts [+ Add](#)

Change account

[HISTORY](#)

abc@gmail.com

algorithm: pbkdf2_sha256 iterations: 260000 salt: CMxEFT***** hash: RyPs7c*****
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Password:

Email:

abc@gmail.com

Username:

abc

☒ Is admin

☒ Is active

☒ Is staff

☒ Is superuser

Date joined:

Nov. 12, 2021, 4:01 a.m.

Last login:

Nov. 12, 2021, 4:27 a.m.

Delete

Save and add another

Save and continue editing

SAVE

Django administration

Home

Logged out

Thanks for spending some quality time with the Web site today.

[Log in again](#)

-----**Thank You**-----