How to Implement Password Reset Functionality in Django Using Built-in Views

Implementing password reset functionality in your Django application can significantly improve user experience and security. Fortunately, Django provides built-in views and forms that simplify this process. In this blog, we'll walk you through setting up a complete password reset system in Django.

Prerequisites

Before we start, make sure you have Django installed and a project set up. You can install Django using pip: **pip install django**

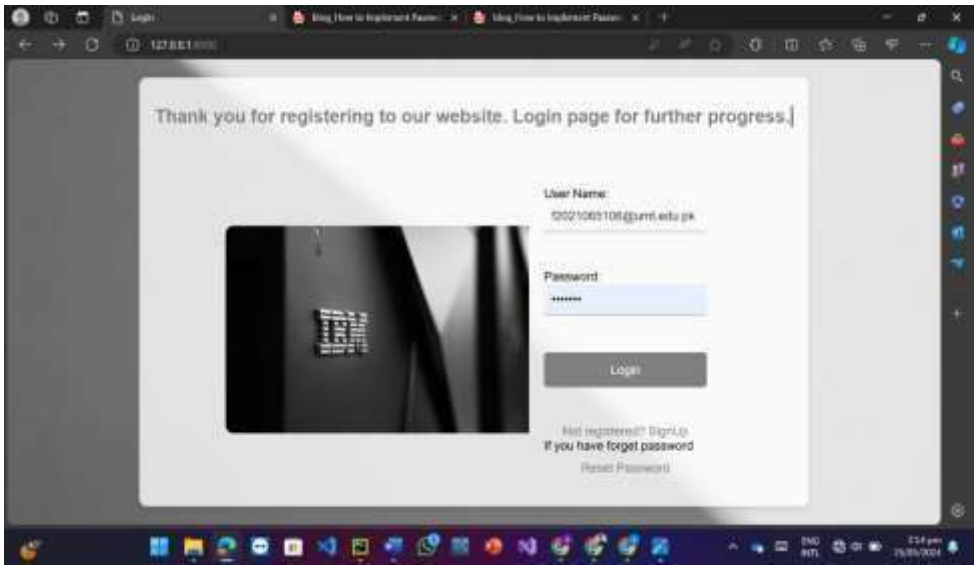
If you don't have a project yet, create one:

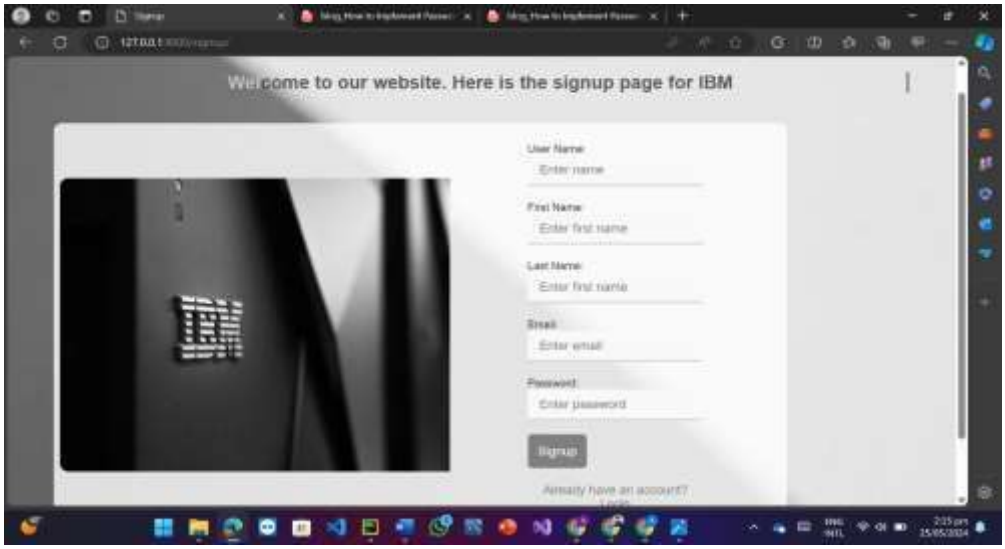
django-admin startproject myproject

cd myproject And create an app:

python manage.py startapp myapp

To implement the password reset you must have already built a signup login page in which you want reset password. I had already made a signup and login page so I can implement password reset functionality.





Step 1: Configure Email Settings

Django's password reset functionality relies on sending emails. First, you need to configure your email settings in settings.py. Here's an example using a Gmail account:

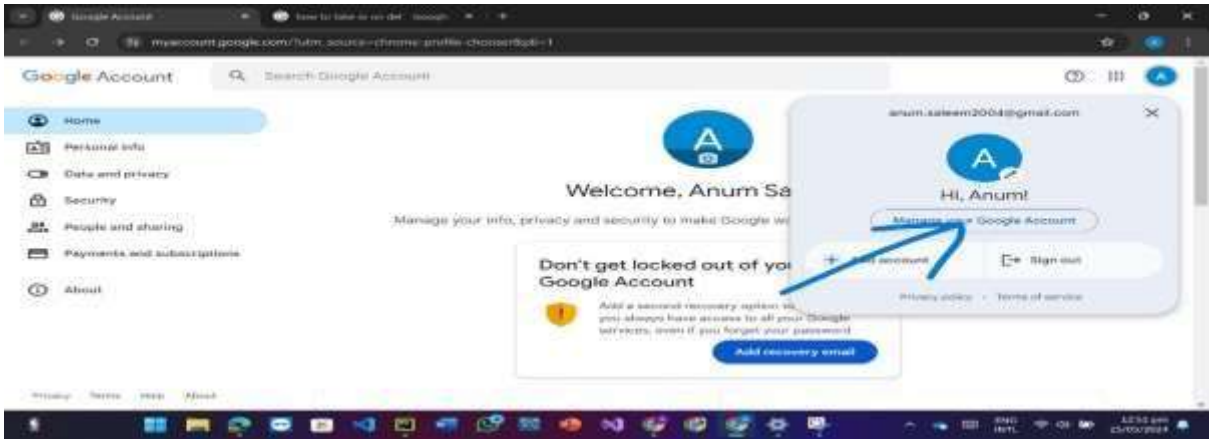
settings.py

```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_USE_TLS = True
EMAIL_PORT = 587  # or you can use 465
EMAIL_HOST_USER = 'anum.saleem2004@gmail.com' EMAIL_HOST_PASSWORD
= ' ongj khzd kicf nlgf  '
'
```

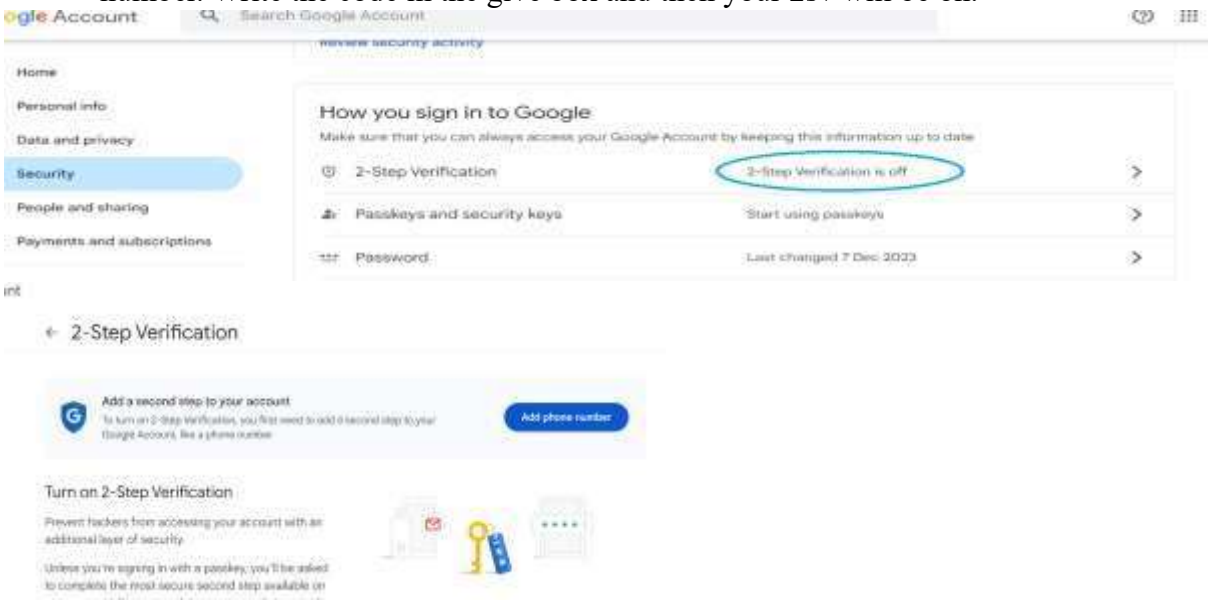
All the above code is written as it is except the email and password. This is the built-in code given by Django.Make sure to replace the email and password with your actual email and password. For production, consider using environment variables or Django's django-environ package for better security.

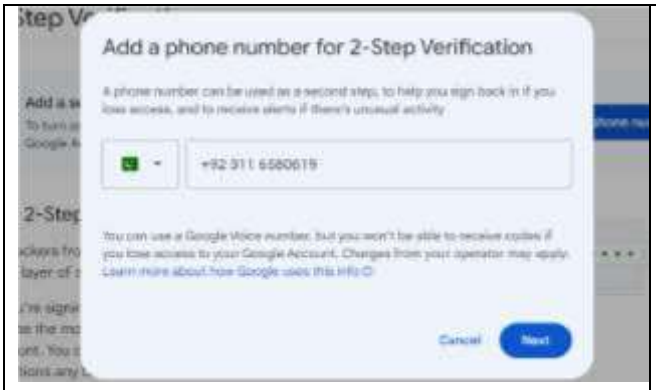
You can get this EMAIL-HOST-PASSWORD by your gmail account. Follow the following steps

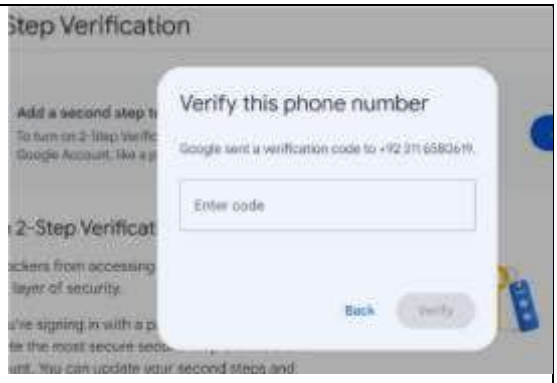
1. First go to your Gmail account and then go to manage your account.



2. Then go to security and turn on the 2 steps verification for your account. To turn on the 2sv you can add your phone number then a code will be sent to you on that phone number. Write the code in the give box and then your 2sv will be on.



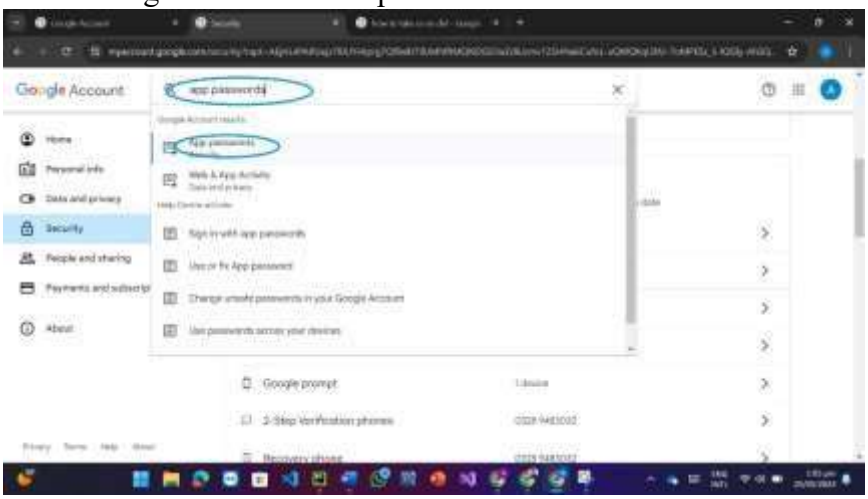


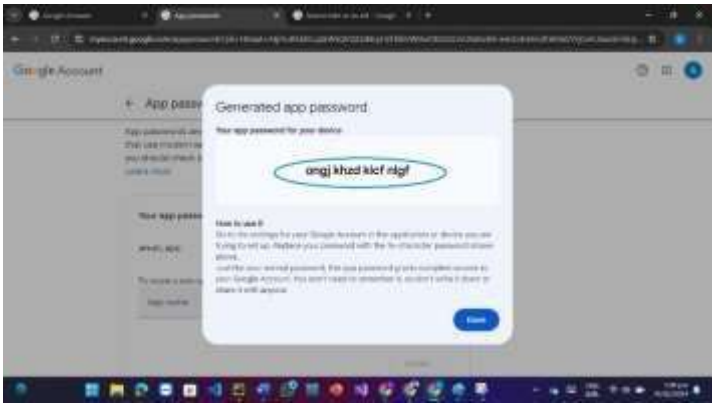


3. Here it shows that 2sv is on.

4. Now to get the password you have to create the app. Go to search bar of your account and type **app passwords** and then you will get that option and go to app passwords and create your app. I created my app as **Anum_app** then I get this password. This password will be given to you once and you have to create another app to get another password: '**ongj khzd kicf nlgf**' Do not share this password with anybody as it can be the reason for your account to be hacked and use by others.

Here is the guide for these steps.





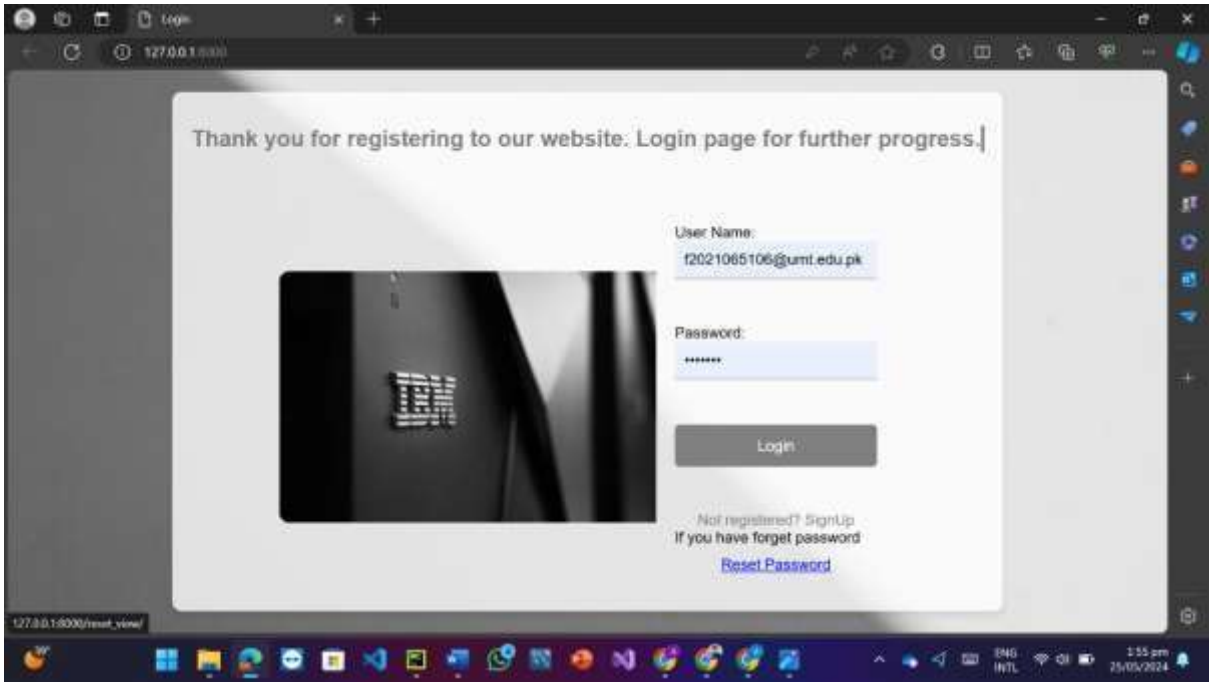
Step 2: Add URL Patterns

Next, we need to include Django's built-in authentication views in our URL configuration.

Open app(urls.py) in your project directory: **myapp/urls.py**

```
from django.urls import path
from . import views
from django.contrib.auth.views import
PasswordResetView,PasswordResetDoneView,PasswordResetConfirmView,PasswordReset
CompleteView
urlpatterns = [
    path('', views.login_user, name='login'),
    path('signup/', views.signup_user, name='signup'),
path('home/', views.home, name='home'),
path('activate/<str:id>/', views.activate, name='activate'),
path('logout/', views.mylogout, name='logout'),
    path('reset_view/', PasswordResetView.as_view(), name='reset_view'),
    path('reset_done/', PasswordResetDoneView.as_view(), name='password_reset_done'),
path('password_confirm/<uidb64>/<token>/',
PasswordResetConfirmView.as_view(),name='password_reset_confirm'),
path('password_reset_complete/', PasswordResetCompleteView.as_view(),
name='password_reset_complete'),
]
```

use the name of the paths same as given in the above code as they are built-in in Django so we cannot change them. By changing them this will give error.

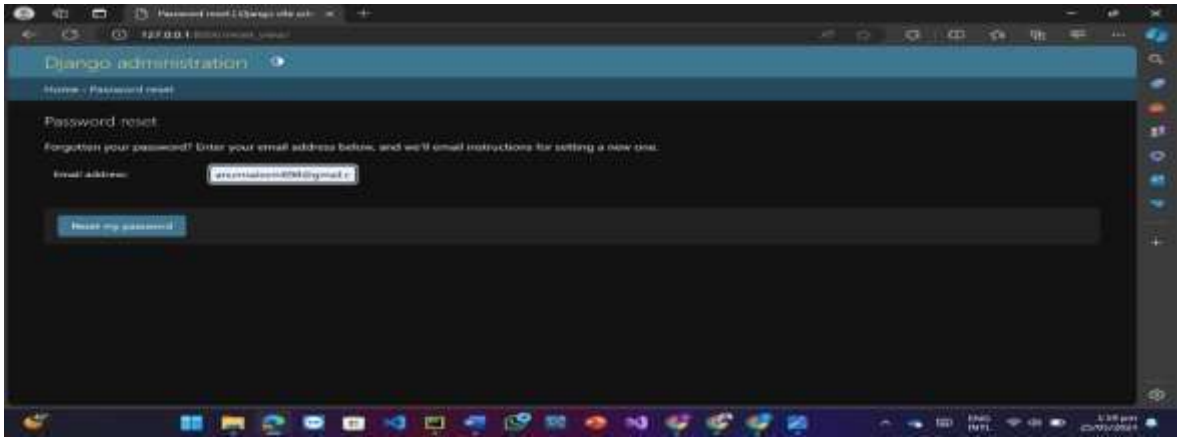


Go to pass reset link if you forgot your password.

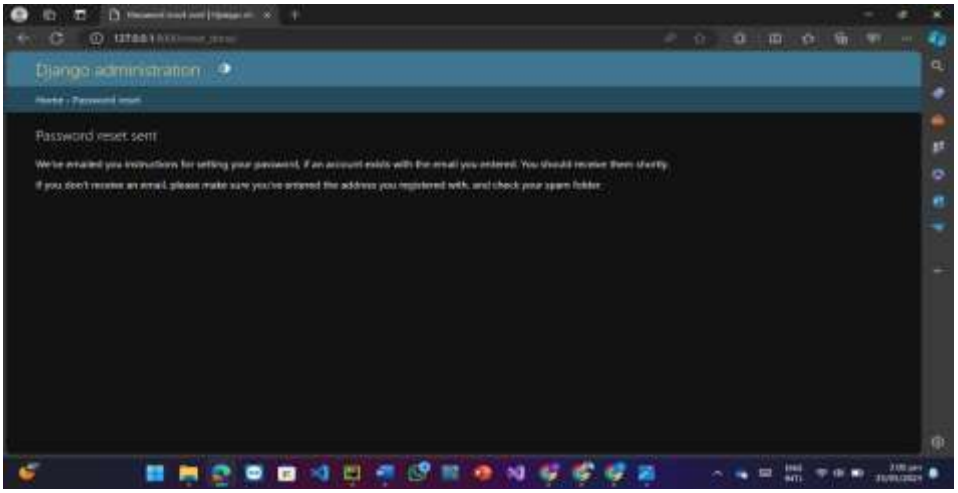
This setup includes the necessary URL patterns for password reset views:

PasswordResetView: Renders the form to submit an email for password reset.

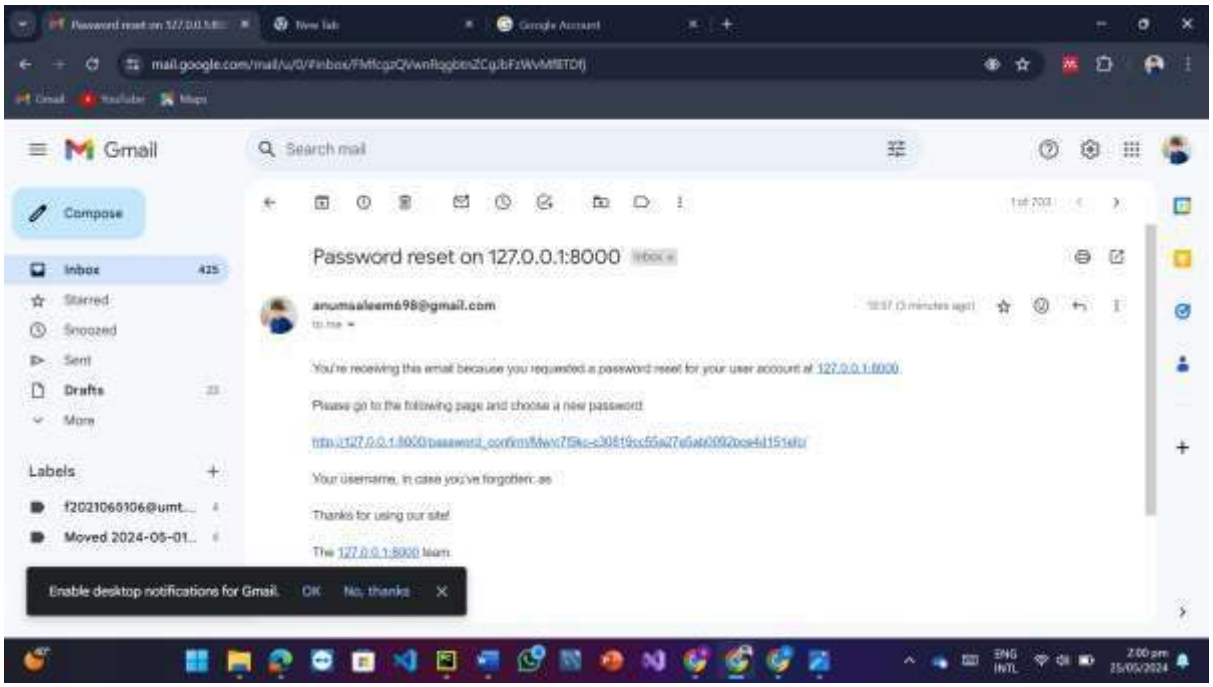
The email should be valid then you can reset your password as verification and authentication are also happened.



PasswordResetDoneView: Informs the user that the email has been sent.

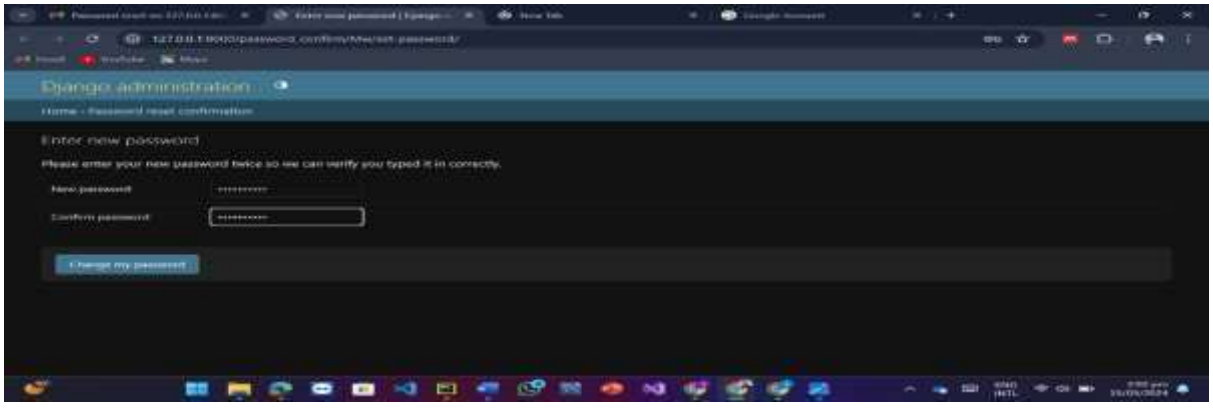


Here I got an email and i will go through the link to change my password.

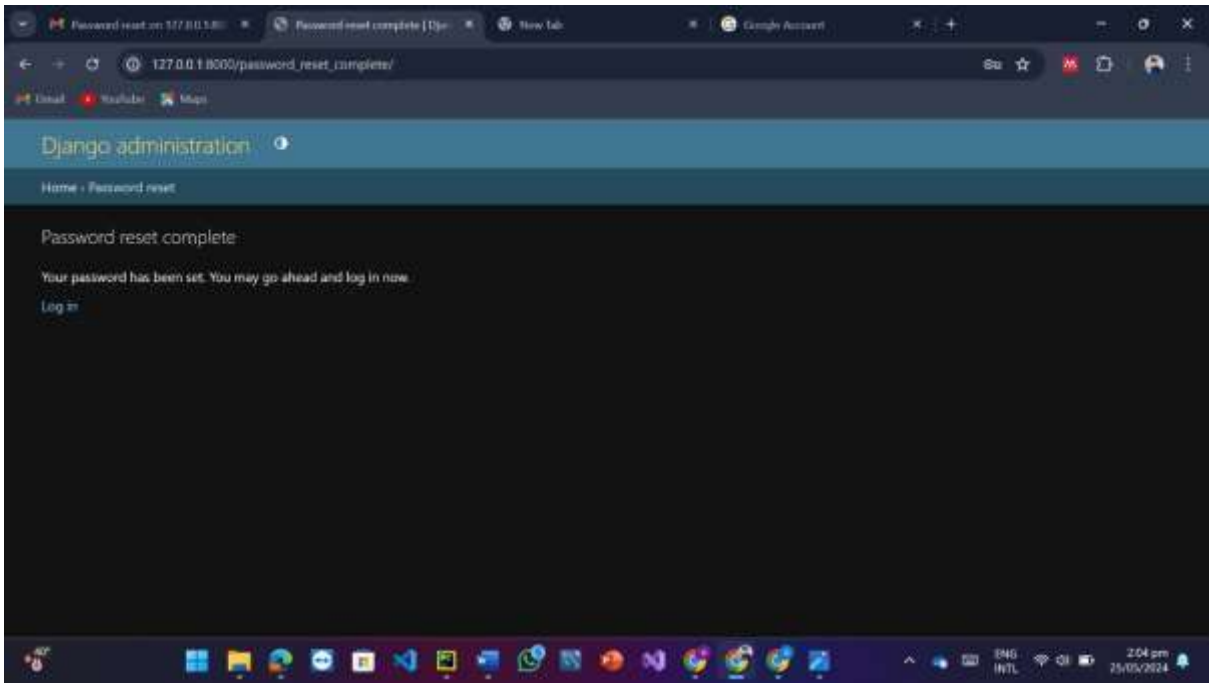


PasswordResetConfirmView: Allows the user to set a new password.

By going through the link this tab opens which ask for the new password and confirm password. It asks you for the strong password as authentication is also happend at the backened.



PasswordResetCompleteView: Informs the user that the password has been reset successfully.



These are all built in views and the builtin templates will be open then if you want to make your website design according to your desired website then create your owntemplates and place them in the path instead of using builtin templates.

Conclusion

With these steps, you have successfully implemented password reset functionality in your Django application using built-in views. This feature enhances user experience by allowing users to recover their accounts securely and efficiently. Django's built-in tools make this process straightforward and easy to integrate into your application.