Experiment No. 9
Creating web application using Django web framework to demonstrate functionality of user login and registration (also validating user detail using regular expression)
Date of Performance:
Date of Submission:

Experiment No. 9

Title: Creating web application using Django web framework to demonstrate functionality of user login and registration (also validating user detail using regular expression)

Aim: To study and implement web application using Django web framework to demonstrate functionality of user login and registration (also validating user detail using regular expression)

Objective: To introduce Django web framework

Theory:

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Ridiculously fast.

Django was designed to help developers take applications from concept to completion as quickly as possible.

Reassuringly secure.

Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable.

Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.

Code:

```
# File: urls.py
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('register/', views.register, name='register'),
    path('login/', views.login, name='login'),
    path('logout/', views.logout, name='logout'),
]

# File: views.py
from django.shortcuts import render, redirect
from django.contrib.auth.models import User
from django.contrib import auth
import re
```

```
def home(request):
    return render(request, 'home.html')


def register(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        email = request.POST['email']
        if not re.match(r'^[\w.@+-]+$', username):
            return render(request, 'register.html', {'error': 'Username can only contain letters, digits, and @/./+/-/_.'})
        if User.objects.filter(username=username).exists():
            return render(request, 'register.html', {'error': 'Username is already taken.'})
        if User.objects.filter(email=email).exists():
            return render(request, 'register.html', {'error': 'Email is already taken.'})
        user = User.objects.create_user(username=username, password=password, email=email)
        auth.login(request, user)
        return redirect('home')
    else:
        return render(request, 'register.html')


def login(request):
```

```
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = auth.authenticate(username=username, password=password)
        if user is not None:
            auth.login(request, user)
            return redirect('home')
        else:
            return render(request, 'login.html', {'error': 'Invalid username or password.'})
    else:
        return render(request, 'login.html')


def logout(request):
    auth.logout(request)
    return redirect('home')
```

Conclusion: The provided Python program utilizes Django to create a web application demonstrating user registration, login, and logout functionalities. Regular expressions are employed for basic validation of the username during registration. This concise implementation showcases how Django can be leveraged to quickly develop secure and user-friendly web applications with authentication features.