

# Creating a simple django project

---

## Create Django Project

---

```
# Create a virtual environment  
python -m venv env
```

```
# Activate the virtual environment  
source env/bin/activate
```

```
pip install django
```

```
# Start the django project  
django-admin startproject aws_demo_project  
cd aws_demo_project
```

```
# Create an app for our project  
python manage.py startapp aws_app
```

```
# aws_demo_project/settings.py
```

```
INSTALLED_APPS = [  
    ...  
    'aws_app',  
]
```

```
# aws_demo_project/urls.py
```

```
from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('aws_app.urls')),  
]
```

Create a new file: `aws_app/urls.py`

```
# aws_app/urls.py

from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name="index"),
    path('<str:username>', views.greet, name='greet'),
]
```

We have not yet created the views so lets create that:

```
# aws_app/views.py

from django.shortcuts import render

def index(request):
    return render(request, 'aws_app/index.html')

def greet(request, username):
    return render(request, 'aws_app/greet.html', {
        "username" : username
    })
```

We have not yet created the templates so lets create it:

create `aws_app/templates/aws_app/index.html` and `aws_app/templates/aws_app/greet.html`

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Index</title>
</head>
<body>
    <h1>This is the main page</h1>
</body>
</html>
```

greet.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{username}}</title>
</head>
<body>
  <h1>Hello {{username}}</h1>
</body>
</html>
```

Now lets see whether this is working:  
start the server by:

```
python manage.py runserver
```

In the browser go to `http://localhost:8000`

You can see the index.html

goto `http://localhost:8000/rahul`

you can see hello rahul

Which means everything is working.

## Now lets create some database stuff:

### First install mysql in the machine

```
sudo apt install mysql-server
sudo mysql_secure_installation
```

Now lets enter the mysql and create a new user

```
sudo mysql
CREATE USER 'rahul'@'localhost' BY '123456';
GRANT ALL PRIVILEGES ON *.* TO 'rahul'@'localhost' WITH GRANT OPTION;
exit
```

Now lets try to enter mysql using newly created user

```
mysql -u rahul -p

# Lets create a database
CREATE DATABASE aws_demo_db;
```

```
SHOW DATABASES;  
# This will show all the databases
```

## Now lets come back to django

To connect django with mysql install mysqlclient

```
pip install mysqlclient
```

If this shows any error

```
sudo apt-get install python-dev python3-dev  
sudo apt-get install libmysqlclient-dev  
pip install mysqlclient
```

```
# aws_demo_project/settings.py
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'aws_demo_db',  
        'USER': 'rahu1',  
        'PASSWORD': '123456',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```

Now lets migrate all the databases provided by the django itself, for that

```
python manage.py migrate
```

Now lets check the tables are created in the database, for that ,

```
mysql -u rahu1 -p  
USE aws_demo_db;  
SHOW TABLES;  
  
# You can see this, These are the tables created by django itself  
+-----+  
| Tables_in_aws_demo_db |  
+-----+  
| auth_group |
```

```
| auth_group_permissions      |
| auth_permission            |
| auth_user                  |
| auth_user_groups           |
| auth_user_user_permissions |
| django_admin_log           |
| django_content_type        |
| django_migrations          |
| django_session             |
+-----+

```

## Now lets create our own model !!

```
# aws_app/models.py
from django.db import models

class Post(models.Model):
    title = models.CharField(max_length=50)
    content = models.TextField()
    image = models.ImageField(upload_to='uploads/')

    def __str__(self):
        return self.title

```

Inorder to add images we have to install pillow:

```
pip install pillow
```

Now lets migrate this to the database for that:

```
python manage.py makemigrations
python manage.py migrate

```

Now if you check the tables in our database you can see this table there!!

## Django Admin

so inorder to add posts we want to access admin panel so lets create the admin site,

```
python manage.py createsuperuser
```

Inorder to add the model to the admin site:

```
# aws_app/admin.py

```

```
from django.contrib import admin
from .models import Post

admin.site.register(Post)
```

Now if you go to <http://localhost:8000/admin/>

You can see the Posts model there, You can add new post there

After uploading an image, you can see that image will be saved in the uploads folder in our project

## Now lets render the details of database to our templates

```
# aws_app/views.py
from .models import Post

def posts(request):
    posts = Post.objects.all()
    return render(request, 'aws_app/posts.html', {
        "posts" : posts
    })
```

Now lets create the url for this view

```
# aws_app/urls.py

urlpatterns = [
    ...

    path('posts/', views.posts, name="posts"),
]
```

We didn't created the html page so lets create the template

create a new file `aws\_app/templates/aws\_app/posts.html`

```
<!--posts.html -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Posts</title>
</head>
<body>
```

```

<h1>ALL POSTS</h1>

{% for post in posts %}
    <div class="post">
        <h1>{{post.title}}</h1>
        
        <p>{{post.content}}</p>
    </div>
{% endfor %}
</body>
</html>

```

Now lets add some styling for this webpage

create this folder:

```
aws_app/static/css/index.css
```

Add this to index.css

```

.post{
    width: 60%;
    margin: auto;
    margin-bottom: 30px;
    background-color: #222;
    color: #fff;
    padding: 30px;
}
.post h1{
    color: #ffc107;
}
.post img{
    width: 300px;
}

```

Now lets link this css to the `posts.html`

For that add `{% load static %}` at the top of `posts.html` and inside head lets link the css by

```
<link rel="stylesheet" href="{% static 'css/index.css' %}">
```

if you go to `http://localhost:8000/posts/`

You can see the posts that you have entered through the admin panel

If the styling doesnt happen, just restart the server, it will be fine

**But wait where is the image ???**

inorder to render the image in webpage we have to do some additional stuffs :

```
1. # aws_demo_project/settings.py
```

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')  
MEDIA_URL = '/uploads/'
```

```
2. # aws_demo_project/urls.py
```

```
from django.conf import settings  
from django.conf.urls.static import static  
  
urlpatterns += static(settings.MEDIA_URL,  
document_root=settings.MEDIA_ROOT)
```

now just restart the server and delete the posts that is created and also you can delete the uploads folder in your project folder

Now just create a new post and woow, you can see the posts in the webpage, at

```
http://localhost:8000/posts/
```

Now before deploying the application we have to gather all the static files:

for that add this line to `aws_demo_project/settings.py` :

```
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

Then to gather all the static files, just type:

```
python manage.py collectstatic
```

now you will see that a new static folder is created in the root directory of you project. This is similar to the media folder creted earlier, when we upload an image.

```
# aws_demo_project
```

```
ALLOWED_HOSTS = ['*']
```

Finally lets create a requirements.txt file by `pip freeze > requirements.txt`

Now our project is ready for deployment