**Python Django Web framework**

**Experiment – 4.1**

**Aim:**

How to setup a project?

**Program:**

1. **Install python –**python.org
2. **Install pip-** Open command prompt and enter following command-

python -m pip install - pip

1. **Install virtual environment-**Enter following command in cmd-

python -m pip install virtualenv

1. **Set Virtual environment-**

virtualenv env\_site or python -m virtualenv demoEnv

1. Change directory to env\_site by this command-

cd env\_site

1. Go to Script directory inside env\_site and activate virtual environment-

cd Scripts

activate

**7**.Install Django

pip install django

#### Creating a Project

Lets’ check how to create a basic project using Django after you

have installed it in your pc.

* + To initiate a project of Django on Your PC, open Terminal and Enter the following command

django-admin startproject projectName

Example- django-admin startproject mysite

* + A New Folder with name projectName will be created. To enter in the project using terminal enter command

cd projectName

Now run,

Python manage.py runserver

Now visit <http://localhost:8000/>

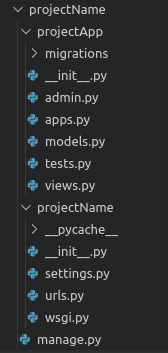
#### Creating an App

To create a basic app in your Django project you need to go to directory containing manage.py and from there enter the command :

python manage.py startapp projectApp

Example- python manage.py startapp polls

Now you can see your directory structure as under :

****

To consider the app in your project you need to specify your project name in INSTALLED\_APPS list as follows in settings.py:

Application definition

INSTALLED\_APPS = [

     'django.contrib.admin',

     'django.contrib.auth',

     'django.contrib.contenttypes',

     'django.contrib.sessions',

     'django.contrib.messages',

     'django.contrib.staticfiles',

     'projectApp.app.projectAppConfig',

]

**Experiment – 4.2**

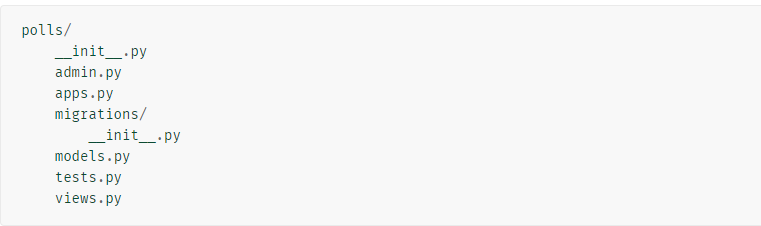
**Aim:**

How to display Hello World in browser using Django?

**Program:**

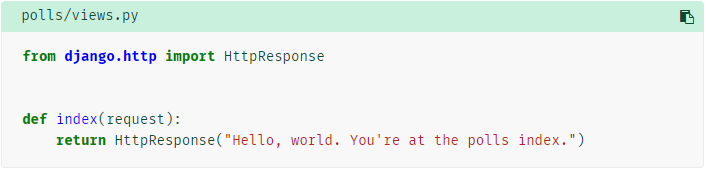
1. **Create an app**

python manage.py startapp polls

That’ll create a directory **polls**, which is laid out like this: ****

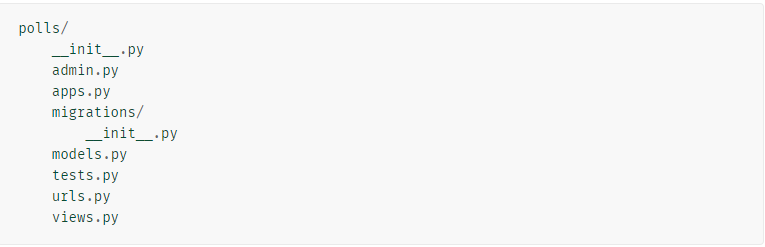
1. **Write your first view**

Open the file **polls/views.py** and put the following Python code in it:

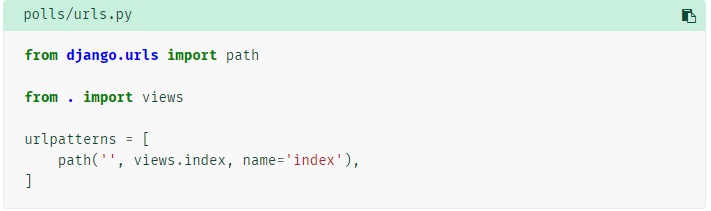
****

**3.** To call the view, we need to map it to a URL - and for this we need a URLconf.

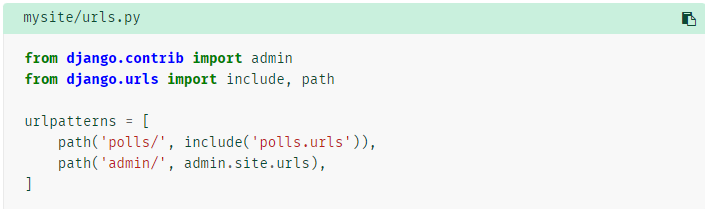
To create a URLconf in the polls directory, create a file called **urls.py**.

****

**4**.In the **polls/urls.py** file include the following code:



1. The next step is to point the root URLconf at the **polls.urls** module. In **mysite/urls.py**, add an import for **django.urls.include** and insert an [**include()**](https://docs.djangoproject.com/en/4.0/ref/urls/#django.urls.include) in the **urlpatterns** list, so you have:

****

1. Run the server

python manage.py runserver

Go to <http://localhost:8000/polls/> in your browser, and you should see the text “Hello, world. You’re at the polls index.”, which you defined in the **index** view.

**Experiment – 4.3**

**Aim:**

**How to apply migrations & What is migrations does?**

**Program:**

## **Database setup**

Now, open up **mysite/settings.py**. It’s a normal Python module with module-level variables representing Django settings.

By default, the configuration uses SQLite. If you’re new to databases, or you’re just interested in trying Django, this is the easiest choice.

If you wish to use another database, install the appropriate [database bindings](https://docs.djangoproject.com/en/4.0/topics/install/#database-installation) and change the following keys in the [**DATABASES**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-DATABASES) **'default'** item to match your database connection settings:

* [**ENGINE**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-DATABASE-ENGINE) – Either **'django.db.backends.sqlite3'**, **'django.db.backends.postgresql'**, **'django.db.backends.mysql'**, or **'django.db.backends.oracle'**. Other backends are [also available](https://docs.djangoproject.com/en/4.0/ref/databases/#third-party-notes).
* [**NAME**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-NAME) – The name of your database. If you’re using SQLite, the database will be a file on your computer; in that case, [**NAME**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-NAME) should be the full absolute path, including filename, of that file. The default value, **BASE\_DIR / 'db.sqlite3'**, will store the file in your project directory.

**mysite/settings.py**

By default, [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-INSTALLED_APPS) contains the following apps, all of which come with Django:

* [**django.contrib.admin**](https://docs.djangoproject.com/en/4.0/ref/contrib/admin/#module-django.contrib.admin) – The admin site. You’ll use it shortly.
* [**django.contrib.auth**](https://docs.djangoproject.com/en/4.0/topics/auth/#module-django.contrib.auth) – An authentication system.
* [**django.contrib.contenttypes**](https://docs.djangoproject.com/en/4.0/ref/contrib/contenttypes/#module-django.contrib.contenttypes) – A framework for content types.
* [**django.contrib.sessions**](https://docs.djangoproject.com/en/4.0/topics/http/sessions/#module-django.contrib.sessions) – A session framework.
* [**django.contrib.messages**](https://docs.djangoproject.com/en/4.0/ref/contrib/messages/#module-django.contrib.messages) – A messaging framework.
* [**django.contrib.staticfiles**](https://docs.djangoproject.com/en/4.0/ref/contrib/staticfiles/#module-django.contrib.staticfiles) – A framework for managing static files.

These applications are included by default as a convenience for the common case.

python manage.py migrate

The [**migrate**](https://docs.djangoproject.com/en/4.0/ref/django-admin/#django-admin-migrate) command looks at the [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-INSTALLED_APPS) setting and creates any necessary database tables according to the database settings in your **mysite/settings.py** file and the database migrations shipped with the app.

By running **makemigrations**, you’re telling Django that you’ve made some changes to your models

**Experiment – 4.4**

**Aim:**

How to access Django admin panel?

**Program:**

### Creating an admin user

python manage.py createsuperuser

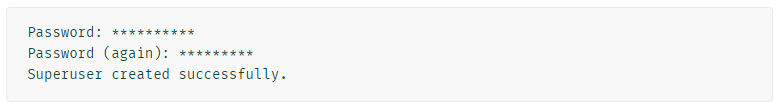
Enter your desired username and press enter.

****

You will then be prompted for your desired email address:

****

The final step is to enter your password. You will be asked to enter your password twice, the second time as a confirmation of the first.

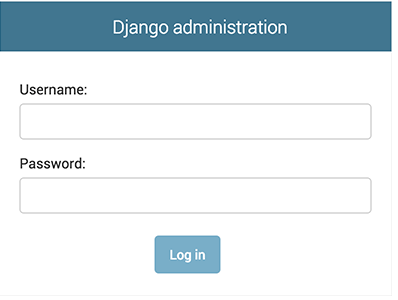
****

### Start the development server

The Django admin site is activated by default. Let’s start the development server and explore it.

python manage.py runserver

Now, open a web browser and go to “/admin/” on your local domain – e.g., <http://127.0.0.1:8000/admin/>. You should see the admin’s login screen

****

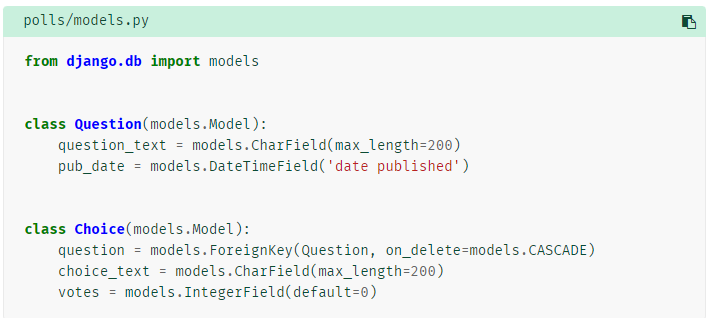
**Experiment – 4.5**

**Aim:**

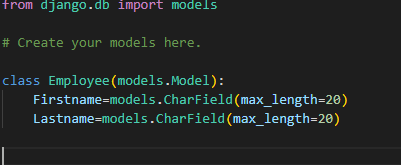
How to create a table in database & what is ORM?

**Program:**

Edit the **polls/models.py**

****

## **Example 2**



## **Activating models**

 First we need to tell our project that the **polls** app is installed.

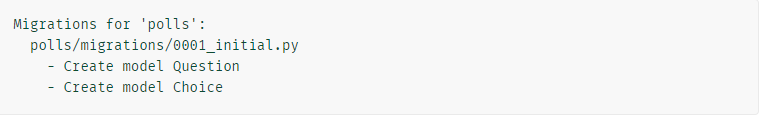
To include the app in our project, we need to add a reference to its configuration class in the [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-INSTALLED_APPS) setting. The **PollsConfig** class is in the **polls/apps.py** file, so its dotted path is **'polls.apps.PollsConfig'**. Edit the **mysite/settings.py** file and add that dotted path to the [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/4.0/ref/settings/#std-setting-INSTALLED_APPS) setting. It’ll look like this:

****

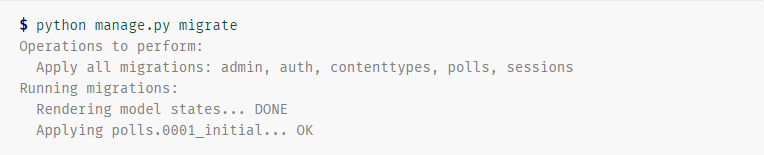
Now Django knows to include the **polls** app. Let’s run another command:

python manage.py makemigrations polls

You should see something similar to the following:

****

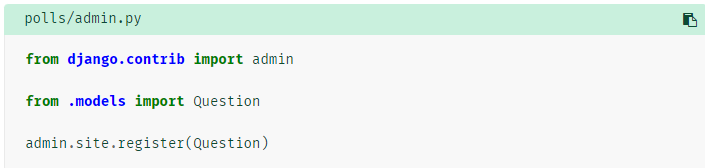
Now, run [**migrate**](https://docs.djangoproject.com/en/4.0/ref/django-admin/#django-admin-migrate) again to create those model tables in your database:

****

* Change your models (in **models.py**).
* Run [**python manage.py makemigrations**](https://docs.djangoproject.com/en/4.0/ref/django-admin/#django-admin-makemigrations) to create migrations for those changes
* Run [**python manage.py migrate**](https://docs.djangoproject.com/en/4.0/ref/django-admin/#django-admin-migrate) to apply those changes to the database.

### **Make the poll app modifiable in the admin(Models)**

We need to tell the admin that **Question** objects have an admin interface. To do this, open the **polls/admin.py** file, and edit it to look like this:

****

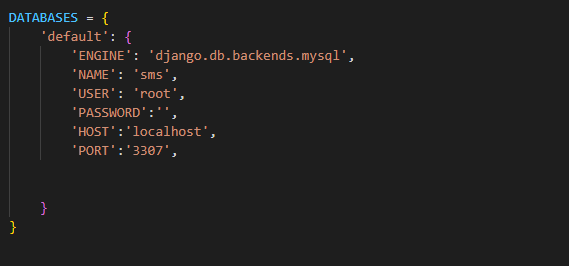
**Experiment – 4.6**

**Aim:**

How to connect MYSQL or … (except sqllite3) in a Django application?

**Program:**

1. Create a database in XAMPP Server
2. Create Models in model.py
3. In settings .py change Database configuration as

****

Engine –Database engine name

Name-Database name

User-username of the database

1. Run the following commands

python manage.py makemigrations

python manage.py migrate

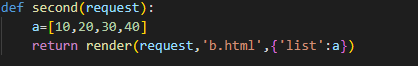
**Experiment – 4.7**

**Aim:**

What is Jinja? Displaying a list of items using Jinja for loop?

**Program:**

Polls/views.py

****

Polls/templates/b.html

****

Run the server

Python manage.py runserver

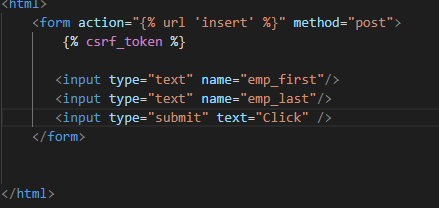
**Experiment – 4.8 -4.9**

**Aim:**

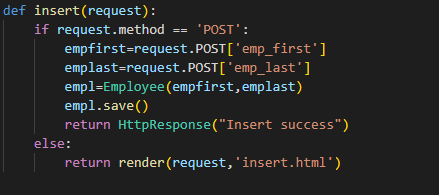
How to methods like POST & GET works? and How to create form using models?

**Program:**

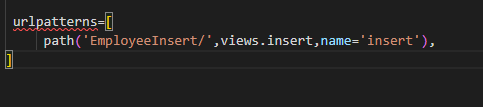
Polls/templates/insert.html

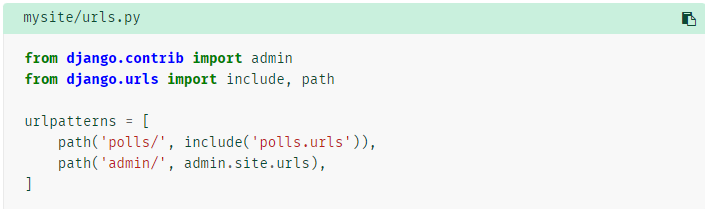


Polls/views.py



Polls/urls.py



****

Run the server

python manage.py makemigrations

python manage.py migrate