Developing Django Applications

01 - Starting Django Project

We have seen how to create virtual environments and install Django, its now time to start with our first Django Project, but before that we need to start a new project.

Steps -

1. Make sure you have installed Django

```
$ pipenv install django
$ pipenv shell
$ django-admin version
```

2. Start a new project

```
$ django-admin startproject projectName .
```

3. Django starts a new project and installs the default files and directory, you can check them out -

```
$ ls
$ Pipfile Pipefile.lock manage.py projectName
```

02 - Understanding HTTP

Before we learn how django works, its important to understand the "Request & Response" cycle.

The browser sends a 'request' to the server, the server 'responds' with the data, this data is interpreted by the browser and displayed on the screen.

```
Request -> Response
```

For example when you type 'google.com' on the browser window, the browser sends a request to the 'google servers', which sends back data in various forms such as html, css, javascript, images, etc. The browser then makes sense of all the received data and displays the 'google homepage'.

```
user -> browser -> google.com -> request -> server -> response -> data ->
browser -> user
```

03 - How Django works ?

This 'Request' is then passed to 'Views' in Django via 'URL', Django then processes the 'Request', if necessary make calls to database which is structured using 'Models' and sends an appropriate 'Response' which is rendered using 'Templates'.

Lets try to visualize this using a table -

Sr No	Particular	Action	Remark
1.	User	Request	User sends a request via the browser
2.	Django	Receives	Django receives the request
3.	Django	Response	Django process the request and sends back a response
4.	Django	Response	If needed Django interacts with database for creating response
5.	Browser	Render	Browser renders the response using Django Templates

04 - First Django Project

Its time for action and create our first Django project.

1. Open terminal and navigate to the folder where you want to create the project

2. Create a new folder for our project -

```
$ mkdir first_project
$ cd first_project
```

3. Create virtual environment and install Django

```
$ pipenv install django
$ pipenv shell
```

4. Start a new project called 'first_project'

```
$ django-admin startproject first_project .
```

5. View the files and folders structure -

```
.
├── Pipfile
├── Pipfile.lock
├── first_project
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── manage.py

1 directory, 7 files
```

3 / 3

6. We can checkout our project by running the development server which ships with Django

```
$ python manage.py runserver
```

7. The default page -