**SECTION ONE (first\_test\_django\_project and first\_app)**

**Django Project**

* It act as a parent directory, it mean Overall an Web app or Web application
* To create the django project using django cli command : django-admin startproject <project\_name>
* This cmd create one folder(parent dir) with the givename, within that it will create one more folder(secondary directory) with the same name and manage.py file also created
* Few more files created under the second folder like \_\_init\_\_.py, setting.py,urls.py,asgi.py and wagi.py
* It is also called as **Project scope Files**
* To start the server : python manage.py runserver <PORT\_NUMBER> is used, it should run the parent directory

**Django App**

* It is sub-dir or sub-component
* It is main purpose is the make the huge project into multiple App/Module. For example, If we have Social Media application, we may have Message features, video call features. So we can put two app as message and video calling into single Django Project (Social Media Application)
* Goto the parent directory of the Django Project and create any number of apps into it.
* To create App : python manage.py startapp <APP\_NAME>
* This cmd will create a folder by given name and within that few files created like app.py admin.py , model.py , view.py and test.py
* It is also called as **App Scope Files**
* If needed or if our application big, we can create urls.py in the App Scope File
* We can have N no of App inside the Project
* Each app may have many views related files so we can configure it in the url.py which in the App Scoped Folders
* Then all Url.py file in the each app scoped folder, can be mapped into the url.py in the secondary folder which is created at the time of Project creation
* We can see the **first\_app** project which is done based the above two section.

**SECTION TWO (view\_route\_urls)**

**Views and URL**

* View means what information needs to show to the end-user
* URL means from where that information is shown to the end-user
* View is the webpage, using URL we can route to the desired view from the browser
* URL configuration can be in the both Project(Project scoped file) level and App(App scoped file) level using the urls.py file
* Using path() and include() Django functions we can configure the URLS in the urls.py files
* Urlpattern is the array type of variable used to wrap the path() and include() in the urls.py
* Connecting View to Url using path() function
* In the path() first argument is the **url** and second argument is the **view or function** and third optional arguments is **kwargs** and fourth optional arguments is **name**

**Dynamic Views**

* This is solution for the creating a each and every view whenever it is required
* Python Dictionary object will be helpful to make the View to ender dynamically
* Passing the topic as an argument into the view we can render the dynamic view with the help of dictionary.
* Path convertor can be done in Django, it means, we can receive the any data type from the request

**Redirects and 404**

* In the dynamic view we have one issue, if any value given which is not in the dictionary, it gives error
* We should handle it by giving proper response.
* We can use the Django 404 page to that error
* Redirects used to redirect the one page to another page using Http

**URL names and reverse()**

* In the above redirects is not dynamic way so reserve() will be sued
* For each url we can give a name
* That name can be used in the reverse() method across Django project

Connecting to Templates

* We wont want to type HTML code or HTML response directly into the view.py file
* To address this, template comes into the picture
* So we can move all out HTML files into separate folder and make the our view.py to communicate within them
* Recommended way to store the templates in the App level, but for now we can store it in the Project top level
* Edit the setting.py(TEMPLATES.DIRS key value) to inform Django Project to communicate the Template folder and View.py