Dharmsinh Desai University, Nadiad
Faculty of technology, Department of Computer Engineering
Subject : Software Project
Lab Manual

Lab 8 : Web Application development with Python and Django 2 : Working with views and templates

Introduction: Django framework is designed on Model – View – Template architecture. Every django application contains models.py, views.py and urls.py file. Models.py file defines data for application. All tables descriptions are defined in model.py file. Views.py file defines programs to manipulate and access data defined in models.py file. Various functionalities defined in views.py can be access on frontend (web pages) via templates. References to these templates (html, css, scripts etc) are stored in urls.py. This tutorial shows basic example to work with views.py and templates. Follow the step describe here to create a basic application which uses views and template. (Some steps are repeated from previous lab manual. If required, refer to lab-6 and 7 manual for details.).

Step 0 : Create a django project :

```
$django-admin startproject viewdemo
```

Step 1 : Create an App:

```
$python manage.py startapp viewdemoapp
```

Step 2: Register new app in settings.py : Add an entry for viewdemoapp in settings.py ***(viewdemo/settings.py)*** as follows,

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'viewdemoapp'
]
```

Step 3 : Update project and app's urls.py file : To access the webpage defined in 'viewdemoapp' we need to define url of the page in projects urls.py ***(viewdemo/urls.py)***. Update the existing urls.py ***(vewidemo/urls.py)*** file as follows,

```
from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^', include('viewdemoapp.urls')),
]
```

inculde('viewdemoapp.urls') defines location of urls file as **/viewdemoapp/urls.py**. Now go to viewdeomapp folder and create a new urls.py file.

```
from django.conf.urls import url
from viewdemoapp import views

urlpatterns = [
    url(r'^', views.HomePageView.as_view()),
]
```

This code imports the views from ***viewdemoapp*** app and expects a view called ***HomePageView*** to be defined.

Step 4: Define view: Open viewdemoapp's views.py file ***(viewdemoapp/views.py)*** and create following views.

```
from django.shortcuts import render
from django.views.generic import TemplateView

# Create your views here.
class HomePageView(TemplateView):
    def get(self, request, **kwargs):
        return render(request, 'index.html', context=None)
```

This file defines a view called ***HomePageView***. Django views take in a `request` and return a `response`. In our case, the method `get` expects a HTTP GET request to the url defined in our `urls.py` file. Once a HTTP GET request has been received, the method renders a template called `index.html` which is just a normal HTML file which could have special Django template tags written alongside normal HTML tags.

Step 5 : Define template: Now we need to define a template which will be rendered whenever view.py is called. Inside the application folder ***(/viewdemoapp/)*** create a template directory.

```
$mkdir templates
```

Go to the newly created templates directory and create and index.html page. Write following code in the index.html page.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Howdy!</title>
    </head>
    <body>
        <h1>Hi! Django views and template test successful!</h1>
    </body>
</html>
```

Step 6 : Now deploy the system with migration commands

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

Step 7 : Run server

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

Step 8: Open web browser and check output at following url.

```
http://127.0.0.1:8000/viewdemoapp
```

Exercise: Create HTML pages (front end) for your project. Also, update models.py as per the table design of your project (If not already completed in last lab).