**Activate Django and run server**

>cd C:\Reena Dangi\python\_stack\my\_environments

> call djangoPy3Env\Scripts\activate

>cd C:\Reena Dangi\python\_stack\django\project\_one

>python manage.py runserver

Create first project and app

>django-admin startproject your\_project\_name\_here

>cd your\_project\_name\_here

>python manage.py runserver (see server is running)

Create an app inside project

>python manage.py startapp your\_app\_name\_here

>python manage.py runserver

>code .

Create url.py at app level

> create urls.py in your app

**from django.contrib import admin**

**from django.urls import path**

**urlpatterns = [**

**path('', views.home, name='app2-home'),**

**]**

>Point project url.py to app url.py

from django.contrib import admin

from django.urls import path,include

from django.conf.urls import url

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',include('app2.urls')),

]

>create a method in views which will point to html template

def home(request):

return render(request, "home.html")

**Adding templates and static files**

>create a folder templates in root app (app2->templates)

>create another folder in templates with same name as app(app2->templates->app2)

>create home.html in that folder(app2->templates->app2->home.html)

>go to setting.py and add app name ('app2.apps.App2Config',) app2 is appname.apps.get method name from apps.py

>go to views -> change method - >

def home(request):

return render(request,'app2/home.html')

**>create folder ‘static’ in root**

>create app name folder under static (app2->static->app2->main.css)

>create main.css

>access stylesheet in home.html

**Adding Base.html , as a template**

**Create base.html**

{%load static%}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=About, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Document</title>

    <link rel="stylesheet" href="{%static 'app2/main.css'%}">

</head>

<body>

      {% block content %}{% endblock %}

</body>

</html>

* **Delete everything from home.html and index.html and write this –**
* {% extends "app2/base.html" %}
* {% block content %}
* <H1>This is Home</H1>
* {% endblock content%}

**Using bootstrap in base.html**

**This is the base template you have to use to use bootstarp**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>Hello, world!</title>

</head>

<body>

<h1>Hello, world!</h1>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</body>

</html>

**Admin Page**

1. **Cmd prompt**
2. **Create database - > Migration**

**>python manage.py makemigrations (detects changes and update DB)**

**>python manage.py migrate (this create db )**

**>python manage.py createsuperuser (give user name/email/pass/passagain) (reena/reena)**

**>** <http://localhost:8000/admin/>

**>login and create new users – test**

**Create model / database structure**

<https://www.youtube.com/watch?v=aHC3uTkT9r8&list=PL-osiE80TeTtoQCKZ03TU5fNfx2UY6U4p&index=5>

**ORM(object relation model)**

**>create post model in models**

from django.db import models

from django.utils import timezone

from django.contrib.auth.models import User

# Create your models here.

# each class will be a table

class post(models.Model):

    title=models.CharField(max\_length=100)

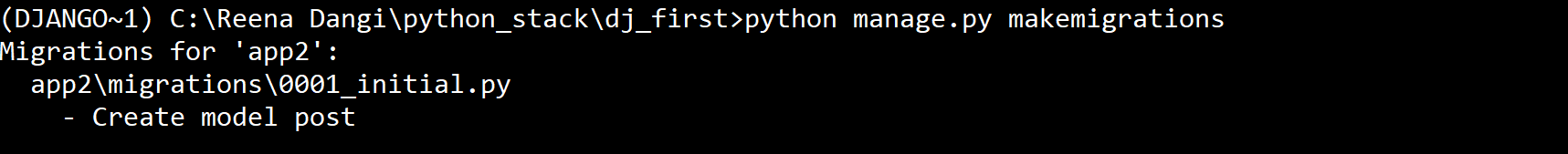
    content=models.TextField()

    date\_posted=models.DateTimeField(default=timezone.now)

    author=models.ForeignKey(User,on\_delete=models.CASCADE)

**>Go to cmd prompt -**

**>python manage.py makemigrations   
 this will create new model named post (table name post)**



**Now go and check migration folder – 0001**

* **Python manage.py sqlmigrate app2 0001 ( this will show what query will run when you are migrating)**
* **Python manage.py migrate (this will create db table named post)**
* **Activate Django shell – which will allow us to look at some sql line by line**
* **Python manage.py shell**
* **>>from app2.models import post / import \***

**>>**from django.contrib.auth.models import User

* **User.objects.all()**
* **User.objects.first()**
* **User.objects.filter(username=’test’).first()**
* **user= User.objects.filter(username=’test’).first()**
* **user**
* **user.id**
* **user= User.objects.get(id=1)**
* **Post.objects.all() – no post**
* **Post\_1=Post(title=’blog1’,content=’First post!’, author=user)**
* **Post\_1.save()**
* **post.objects.all()**
* **Post.object.create (title=’blog1’,content=’First post!’, author=user)**
* **Go back to models and add – a dunder object(\_\_**
* def \_\_str\_\_(self):
* return self.title
* **Restart shell**
* **Import …..**
* **Post.objects.all()**
  + **Shows data**

**>create user again**

**>user=User.objects.filter(username=”test”)**

**>user**

**>post\_2=Post(title=’Blog2’,content=’Blog2’,author\_id=user.id)**

**>post\_2.save()**

**>Post.objects.all()**

**>post=Post.objects.first()**

**>post**

**>post.content**

**>post.date\_posted**

**>post.author.email**

**> user.post\_set ( it will return all post)**

**>user.post\_set.all() ( it will return all post by that user)**

**>user.post\_set.create(title=’blog 3’……) (no need to run save)**

**>post.objects.all()**

**>exit shell**

**Get this data in view-**

**Pass this data in home page and render it !**

from django.shortcuts import render

from django.http import HttpResponse

from .models import post

# Create your views here.

def home(request):

    context={

        'post':Post.objects.all()

    }

    return render(request,'app2/home.html',context)

def about(request):

    return render(request,'app2/about.html',context)

**Now change home.html**

{% extends "app2/base.html" %}

{% block content %}

<H1>Welcome Home ! {{request.session.first\_name}}</H1>

{%for post in posts%}

<article class="media content-section">

    <div class="media-body">

        <div class="article-metadata">

            <a class="mr-2" href="#">{{ post.author }}</a>

            <small class="text-muted">{{ post.date\_posted|date:"F d, Y" }}</small>

        </div>

        <h2><a class="article-title" href="#">{{ post.title }}</a></h2>

        <p class="article-content">{{ post.content }}</p>

    </div>

</article>

{%endfor%}

{% endblock content%}

**How do we add post to admin ? so that admin can edit post ?**

**Register model in Admin.py**

* **Go to admin.py**
* from django.contrib import admin
* # Register your models here.
* from .models import post
* admin.site.register(post)

**Try going to admin page and update post and it works !!!**

**Create one to many**

**class Author(models.Model):**

**name = models.CharField(max\_length=255)**

**created\_at = models.DateTimeField(auto\_now\_add=True)**

**updated\_at = models.DateTimeField(auto\_now=True)**

**class Book(models.Model):**

**title = models.CharField(max\_length=255)**

**author = models.ForeignKey(Author, related\_name="books")**

**created\_at = models.DateTimeField(auto\_now\_add=True)**

**updated\_at = models.DateTimeField(auto\_now=True)**

**Create many to many**

**class Book(models.Model):**

**title = models.CharField(max\_length=255)**

**created\_at = models.DateTimeField(auto\_now\_add=True)**

**updated\_at = models.DateTimeField(auto\_now=True)**

**class Publisher(models.Model):**

**name = models.CharField(max\_length=255)**

**books = models.ManyToManyField(Book, related\_name="publishers")**

**created\_at = models.DateTimeField(auto\_now\_add=True)**

**updated\_at = models.DateTimeField(auto\_now=True)**

**Login and registration**

1. **Before you can use inbuilt forms and models for USER , Make migrations and migrate**
2. **Create Admin account** 
   1. <http://localhost:8000/admin>
   2. Create a super user
   3. **Reena/reena.dangi@gmail.com/codingdojo**
3. **Create an app users app**
4. **Change installed\_apps in settings.py**
5. **\*\*From Django.contib.forms import UserCreationForm**
6. **View.py -> create a view ->register ->**
   1. **Form=UserCreationForm()**
   2. **Return render(request, ‘users/register.html’, {‘form’:form})**
7. **Create a base.html and use that as a base template**
8. **Create templates->users->register.html**
9. {% extends "users/base.html" %}
10. {% block content %}
11. <H1>Registration</H1>
12. <div class="content-section">
13. <form method="POST">
14. {% csrf\_token %}
15. <fieldset class="form-group">
16. <legend class="border-bottom mb-4">Join Today</legend>
17. {{form}}
18. </fieldset>
19. <div class="form-group">
20. <button class="btn btn-outlin-info" type="submit">Sign up!</button>
21. </div>
22. </form>
23. <div class=border-top pt-3>
24. <small class="text-muted">
25. Already have an account?
26. <a href="{%url 'login' %}" class="ml-2">Sign in</a>
27. </small>
28. </div>
29. </div>
31. {% endblock content%}
32. **URLs.PY at root** 
    1. **From users import views**
33. from django.contrib import admin
34. from django.urls import path,include
35. from users import views as user\_views
36. urlpatterns = [
37. path('admin/', admin.site.urls),
38. path('register/',user\_views.register,name="register"),
40. ]

 Change form to {{form.as\_p}} to make it look better

* **Check if data is post data – then we want to create a form which has data**
* def register(request):
* if request.method=='POST':
* form=UserCreationForm(request.POST)
* if form.is\_valid():
* username=form.cleaned\_data.get('username')
* else:
* form=UserCreationForm()
* context={'form':form}
* return render(request,'users/register.html',context)

**Now show messages -** from django.contrib import messages

if form.is\_valid():

            username=form.cleaned\_data.get('username')

            messages.success(request,f'Account created for {username}')

            return redirect("/")

**>Now add message in base.html**

  {%if messages%}

                {%for message in messages%}

                <div class="alert alert-{{message.tags}}">

                    {{message}}

                </div>

                {%endfor%}

                {%endif%}

* **Now see how to save data**
* **Form.save()**
* if form.is\_valid():
* form.save()
* username=form.cleaned\_data.get('username')
* messages.success(request,f'Account created for {username}')
* print("\*"\*40,messages)
* return redirect("/register/")
* **Now we need to add few more fields to our form** 
  + **Email**
  + **First name**
  + **Last name**
* **Cerate a file – forms.py**
* from django import forms
* from django.contrib.auth.models import User
* from django.contrib.auth.forms import UserCreationForm
* class UserRegisterForm(UserCreationForm):
* email=forms.EmailField()
* first\_name=forms.CharField(min\_length=2, max\_length=12)
* last\_name=forms.CharField(min\_length=2, max\_length=12)
* class Meta:
* model=User
* fields=['username','first\_name','last\_name','email','password1','password2']
* **Now change views**
* from .forms import UserRegisterForm
* **replace UsercreationForm with UserRegisterForm**
* **Awesome !!!!! It worked ..create a new user and validate that it works**
* **Now change cSS**
* **Install crispy forms**
* **Pip install djago-crispy-forms**
* **Setting.py ->installed apps**
  + **‘crispy\_forms’,**
  + **Very bottom ->**
  + CRISPY\_TEMPLATE\_PACK='bootstrap4'
* **register.html ->** 
  + **{% load crispy\_forms\_tags%}**
  + **{{form|crispy}}**

**>Now this looks Awesome !!!!**

**Login/Logout modules**

* **Urls.py**
* **Cerate login.html and logout.html**
* **Update urls.py**
* from django.contrib import admin
* from django.urls import path,include
* from users import views as user\_views
* # Django provides login and logout views
* from django.contrib.auth import views as auth\_views
* urlpatterns = [
* path('admin/', admin.site.urls),
* path('register/',user\_views.register,name="register"),
* path('login/',auth\_views.LoginView.as\_view(template\_name='users/login.html'),name="login"),
* path('logout/',auth\_views.LogoutView.as\_view(template\_name='users/logout.html'),name="logout"),
* ]

**>Run – it will give error accounts/profile/ not found , now change path**

**>**

**How to add image to your profile**

* **Settings.py**
* MEDIA\_ROOT =  os.path.join(BASE\_DIR, 'media')
* MEDIA\_URL = '/media/'

<img class="mr-3 mt-5" width=300 src="{{destination.album\_cover.url}}" alt="">

**How to add static/image files**1. Store your static files in a folder called **static** in your app. For example **my\_app/static/my\_app/example.jpg**.  
2. Make sure that **django.contrib.staticfiles** is included in your [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/3.0/ref/settings/#std:setting-INSTALLED_APPS).  
3. In your settings file, define [**STATIC\_URL**](https://docs.djangoproject.com/en/3.0/ref/settings/#std:setting-STATIC_URL), for example:

STATIC\_URL = '/static/'

4. Store your static files in a folder called **static** in your app. For example **my\_app/static/my\_app/example.jpg**.

**Google photo API**

1. pip install --upgrade google-api-python-client

The google-auth-oauthlib and google-auth-httplib2 libraries for user authorization.

1. pip install --upgrade google-auth-oauthlib google-auth-httplib2

Create or select a project in the [API Console](https://console.developers.google.com/). Complete the following tasks in the API Console for your project:

1. In the [library panel](https://console.developers.google.com/apis/library), search for the API you are calling and make sure it is enabled for your project.

[**https://accounts.google.com/o/oauth2/auth?response\_type=code&client\_id=836787780852-7mthov4o51ukrt8kutceob8odk918c2p.apps.googleusercontent.com&redirect\_uri=http%3A%2F%2Flocalhost%3A63416%2F&scope=http%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.metadata.readonly&state=jP5Lh5PIVVSjBooNhQnmyhDjS0uXqK&access\_type=offline**](https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=836787780852-7mthov4o51ukrt8kutceob8odk918c2p.apps.googleusercontent.com&redirect_uri=http%3A%2F%2Flocalhost%3A63416%2F&scope=http%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.metadata.readonly&state=jP5Lh5PIVVSjBooNhQnmyhDjS0uXqK&access_type=offline)

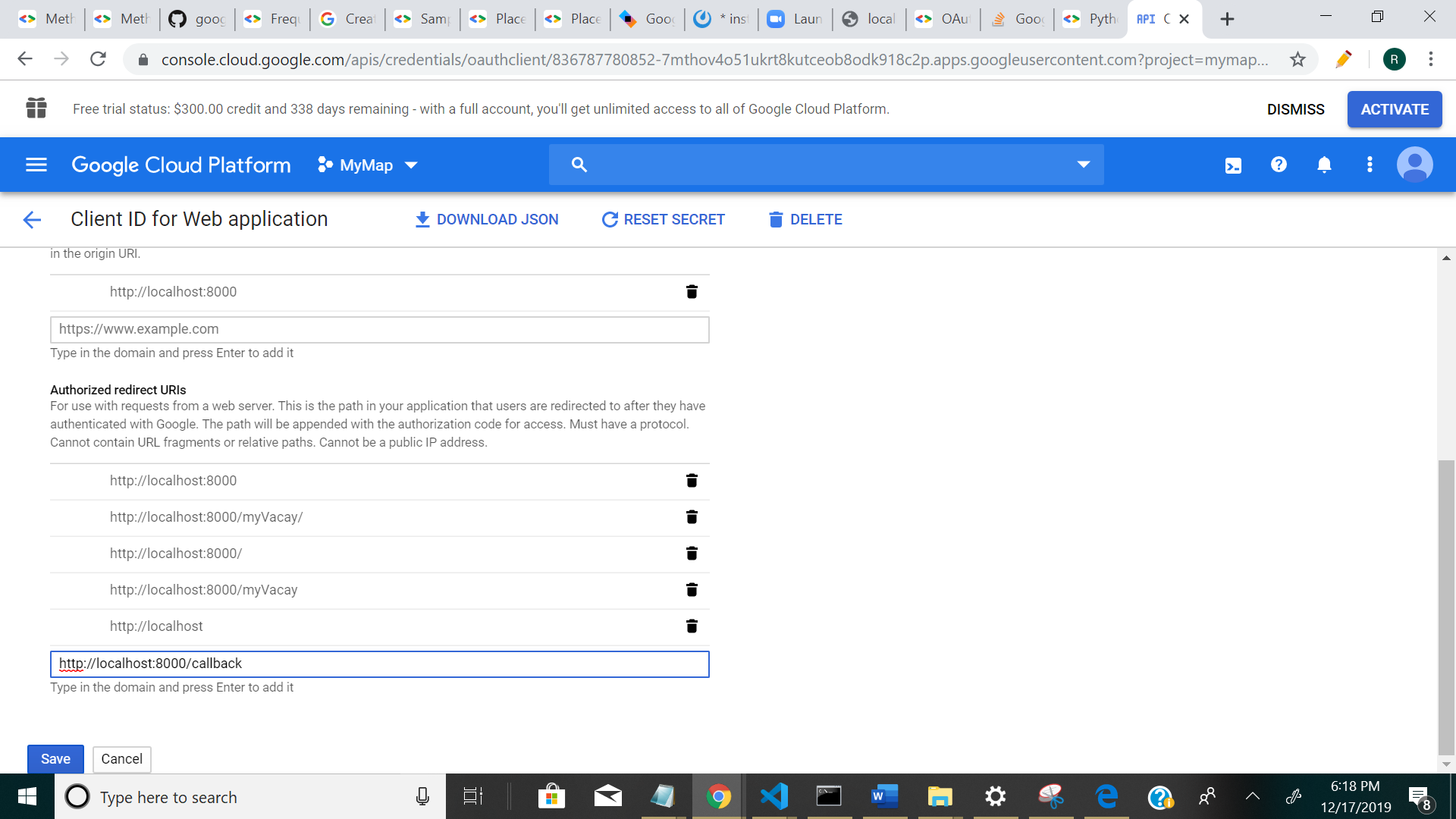
**I', 'ChIJ7cv00DwsDogRAMDACa2m4K8']**

**[17/Dec/2019 18:04:26] "GET /myVacay/ HTTP/1.1" 200 4475**

**flow <google\_auth\_oauthlib.flow.InstalledAppFlow object at 0x049D7700>**

**Please visit this URL to authorize this application: https://accounts.google.com/o/oauth2/auth?response\_type=code&client\_id=836787780852-7mthov4o51ukrt8kutceob8odk918c2p.apps.googleusercontent.com&redirect\_uri=http%3A%2F%2Flocalhost%3A8000%2F&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fphotoslibrary.readonly&state=7yXRDGZ2n20xJGTOo9OEKNkRPuSZ1S&access\_type=offline**

**[17/Dec/2019 18:05:03] "GET /?state=7yXRDGZ2n20xJGTOo9OEKNkRPuSZ1S&code=4/uQGht1StLcznpb8IoQ2CTf9xst0Y6XRlRBh47Wm0Gy\_1Hp6ojbf4mRIAWkFf1siSGsr1HqgUowlNyBAXyYAxIO4&scope=https://www.googleapis.com/auth/photoslibrary.readonly HTTP/1.1" 200 7242**



**localhost:8000/myVacay/?state=UwA8YyrM9737ATSY0jocfMKyN34FOS&code=4/uQHSYHmKVLvA-7dio74wiAU2bVpD75nEUajrh7nyFhT57pjj8lVWR3KOIy9iPAhA2OJMy42THs3NPbt\_hboaH1Y&scope=https://www.googleapis.com/auth/photoslibrary.readonly**

**Working code –**