

22-07-2020:

=====

Good Morning Respected Sir/Mam

Django-Project Creation:

- python -V
- django-admin --version
- django-admin startproject "projectname"

-> It automatically creates Admin app with Project name

-> To run a server we need to run a command that is "python manage.py
runserver"

User Creation App:

-> We need to navigate for project location

-> so we need to run a command in command prompt for creation of a user
app

- django-admin startapp "userappname"

Configuring Admin app in Settings.py,Urls.py and in Userapp creating a
function in Views.py:

-> In FDP3 you can observe settings.py so in this file we need to register our
app (i.e., Application definition).

-> We need to open urls.py just we need to add url path and our app function
we need to import our app to this location

-> from myapp import views

we need to give a path for that function

```
-> urlpatterns =[
    path('hi/',views.hello,name="hq"),
]
-> hi -> url that can be entered in browser
-> hello -> to perform that action
-> name -> forms -> linking
```

Response with parameters in an url:

=> http://localhost:8000/s/raju->User

Hello Good Morning raju

```
=> url
    -> path('s/<str:r>/',views.hg),
=> views
    -> def hg(request,r):
        return HttpResponse("Hello "+r)
```

HTML,CSS,Javascript,Jquery:

=====

Html -> Templates => .html

url:

```
path('rw/<str:name>/',views.fr),,
```

views:

```
def fr(re,name):
    return render(re,'myapp/sample.html')
```

render:

render(re,'.html',{})

Templates: -> myapp -> sample.html
 -> usr -> sample.html

Templates:

-> We need to create a folder with a name templates that should be inner of
userapp
-> Inner of Templates folder we need to create another folder with
userappname
-> There it self we need to create our .html files

23-07-2020:

=====

- ❖ Navigating data from one html file to another
- ❖ Inline styles
- ❖ Internal Styles:
 - We have 3 ways:
 - by using element(tag)
 - by using id
 - by using class

24-07-2020:

=====

Yesterday Topics Covered:

-> Data sending from url to .html

- > Created Form and applied some styling for that form
- > Styling to a form by using selectors like tagname(i.e., element,), id, class.
- > Different ways to apply styling to a .html

Good Morning Respected Sir/Mam

CSS:

=====

Inline -> This can be used in inner of a start tag that can be changed by using property:value

Internal -> This styling should be used with a style tag
elementname,idname,classname
property:value

-> This internal style should be inner of head tag

External -> We need to pass all those styling to another .css file that should be accessed within a head tag

Ex:

first.css

```
h1
{
    //property:value;
    color:black;
}
```

```
<head>
<link rel="stylesheet" href="" type="text/css">
</head>
```

css,js,images -> external files -> static

Static:

-> We need to create static folder inner of our app

-> After creation then we need to create another folder for accessing .css files

-> css -> .css

-> js -> .js

-> images -> .jpg, .png, images extensions

urls:

path('exc/', views.extc)

Views:

```
def extc(request):  
    return render(request, 'reg.html')
```

inline -> it will not breaks for new line

Ex: span, i

block -> it will displays in new line

Ex: div

first.css

form

```
{  
    background-color: cadetblue;  
    margin-left: 30%;  
    margin-right: 30%;  
    margin-top: 70px;  
}
```

Responsive purpose just we are using Bootstrap

-> Css,javascript,jquery

- css
- jquery
- js

Javascript:

- for validations,action

```
<head>
  <script>
    alert("Hello")
  </script>
</head>
```

```
<body>
<script>
function at()
{
  alert("Hello")
}
</script>
</body>
```

external js

sample.js

```
var a = prompt("Enter first value")
var b = prompt("Enter second value")
r = parseInt(a)+parseInt(b)
alert(r)
```

25-07-2020:

=====

Sample.js

```
function add()
{
    var a = document.getElementById("fval").value
    var b = document.getElementById("sval").value
    var c = parseInt(a)+parseInt(b)
    //alert(c)
    document.getElementById("out").innerText="Addition of "+a+"and
"+b+"is: "+(parseInt(a)+parseInt(b))
}
```

Bootstrap:

- Inbuilt packages [css,js]
- css -> bootstrap.min.css
- js -> bootstrap.min.js
- fonts -> bootstrap

Responsiveness

Online

Bootstrap cdn:

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></scrip
t>
```

Bootstrap.html code:

```
-----
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></scrip
t>
</head>
<body>
<div class="container">
  <div class="jumbotron" style="margin-top:50px">
    <h1>Welcome to FDP Program</h1>
  </div>
  <div class="row">
    <div class="col-md-6">
      <div class="jumbotron">
        <h2>Login</h2>
        <form method="POST" action="#">
          <input type="text" name="uname"
class="form-control" placeholder="Username" required><br/>
```



```

        <input type="password" name="pwd"
class="form-control" placeholder="Password" required><br/>
        <input type="submit" value="Login" class="btn
btn-primary">
    </form>
</div>
</div>
</div>
<div class="col-md-6">
    <div class="jumbotron">
        <h2>Register</h2>
    </div>
</div>
</div>
</div>
</body>
</html>

```

Offline:

Static -> css/js/images

Database:

Without creating of user models

-> "python manage.py migrate"

-> tables -> default -> admin

-> User[table -> auth_user]

Sqlite: db browser for sqlite

-> open database [project location][.sqlite3]

we can create Admin by command prompt

-> python manage.py createsuperuser

-> username:

-> email id:

-> password:

-> confirm password:

-> 8 y/n

-> Admin

-> roles,permissions,

Models.py:

Own tables

-> python manage.py makemigrations

-> migrations[folder][0002.py] -> delete

[interfacing file] -> x table

-> python manage.py migrate

Models.py:

Own tables

-> python manage.py makemigrations

-> migrations[folder][0002.py] -> delete

[interfacing file] -> x table

-> python manage.py migrate

ORM: -> Object Relational Mapping

Models.py:[table -> field name]

Usreg -> tablename

field name = datatype()

- CharField(max_length="230")
- IntegerField()
- FloatField()
- ImageField()
- FileField()

```
class Usreg(models.Model):  
    username = CharField()  
    pawd = CharField()  
    age = IntegerField()  
    sal = FloatField()
```

myapp_Usreg -> default