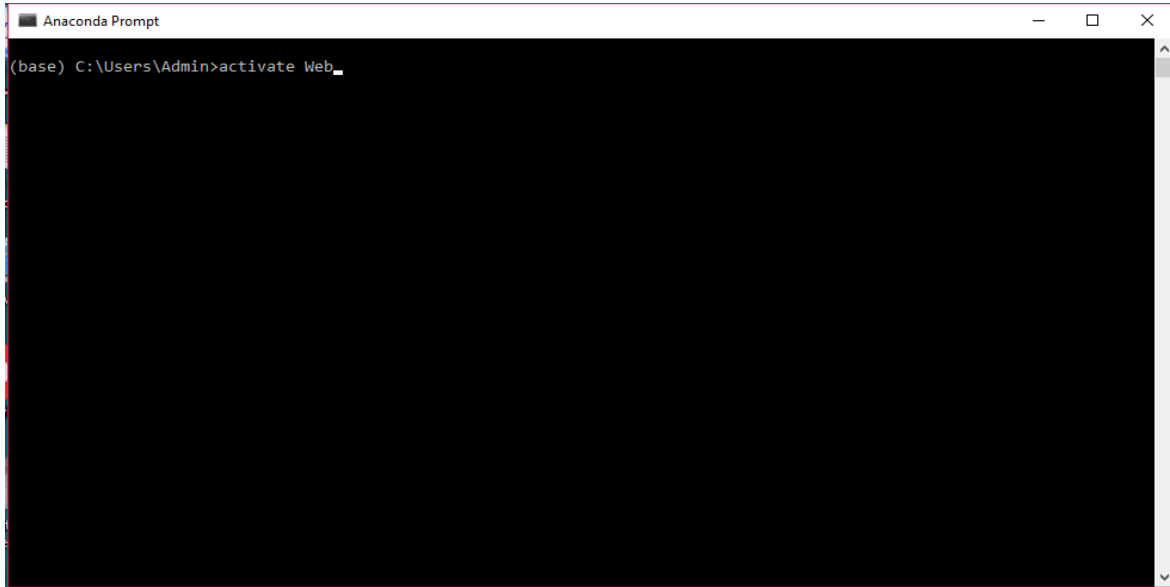


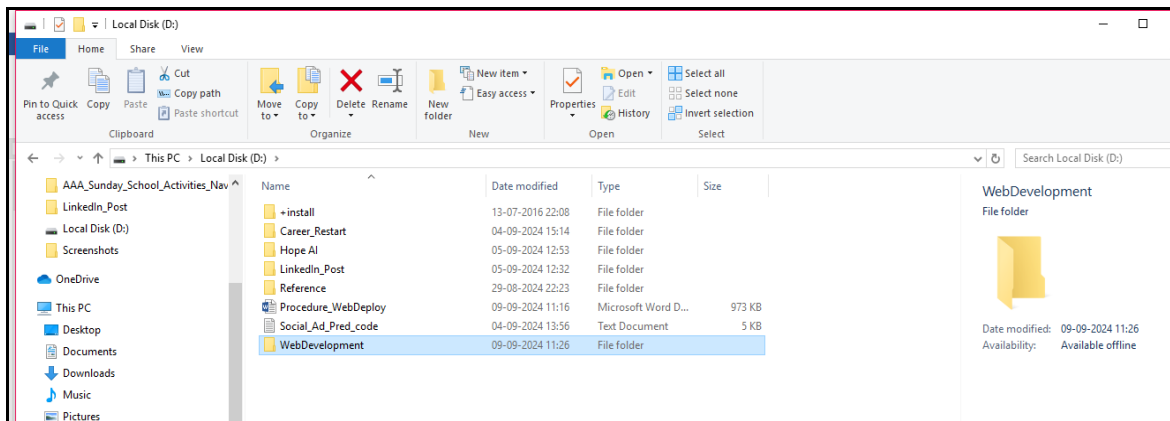
# Django Web Development Machine Learning Project

## Step 1: Creating Virtual Environment

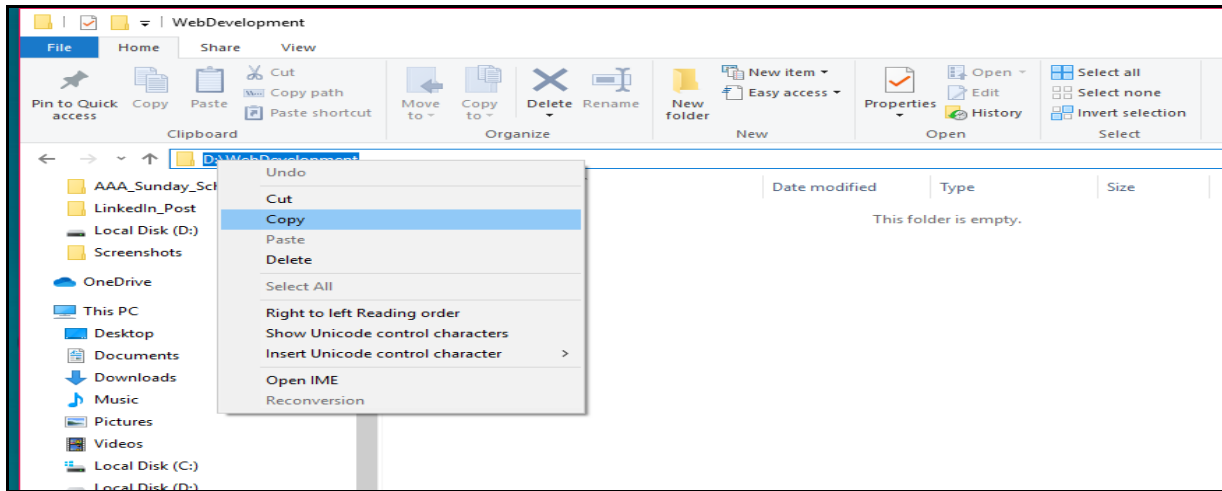


## Step 2: Create a New Folder in the local disk.

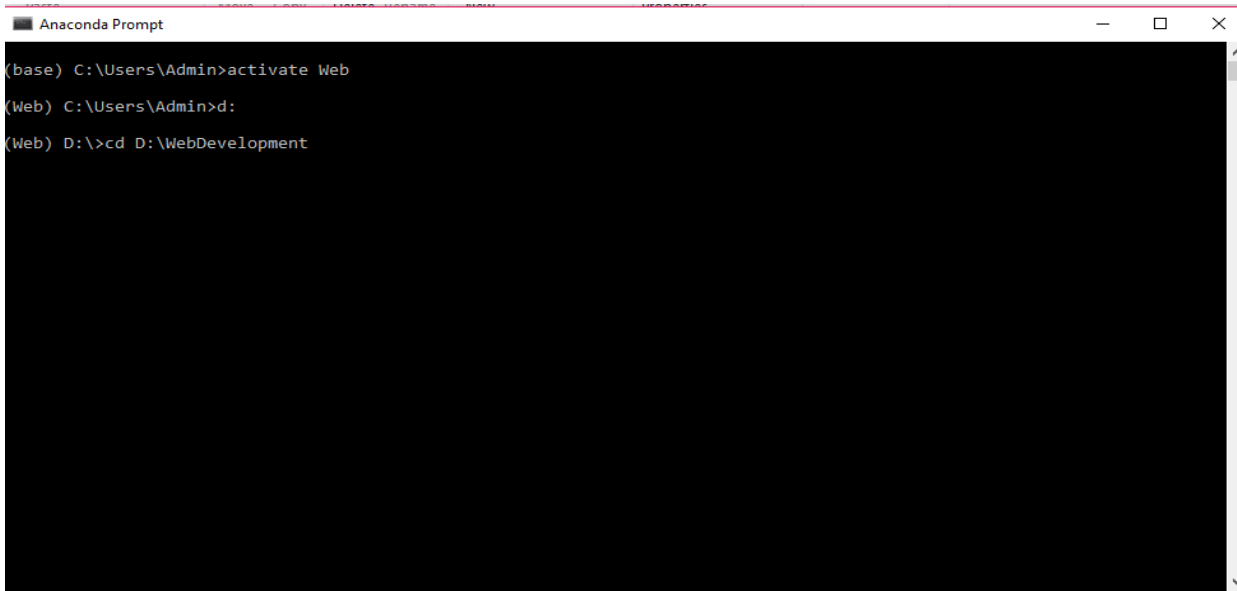
Select Local Disk (D:) > New Folder (eg: WebDevelopment)



### Step 3: Copy the Folder path

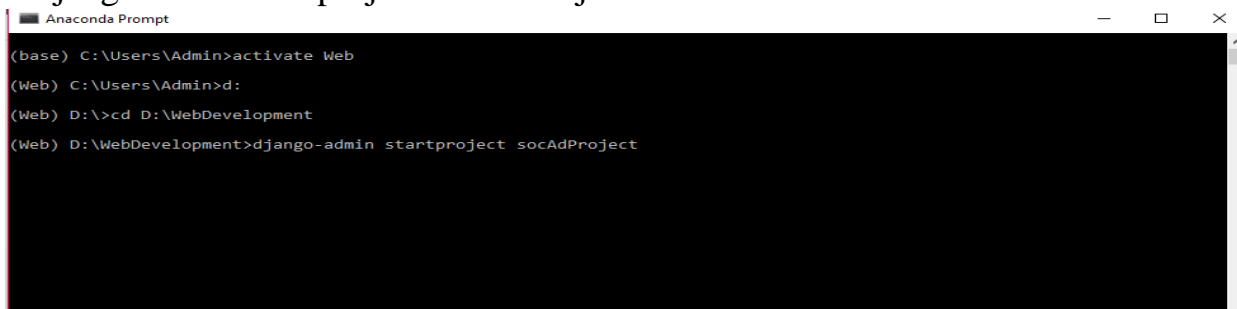


### Step 4: In the Anaconda command prompt, paste the copied folder path.

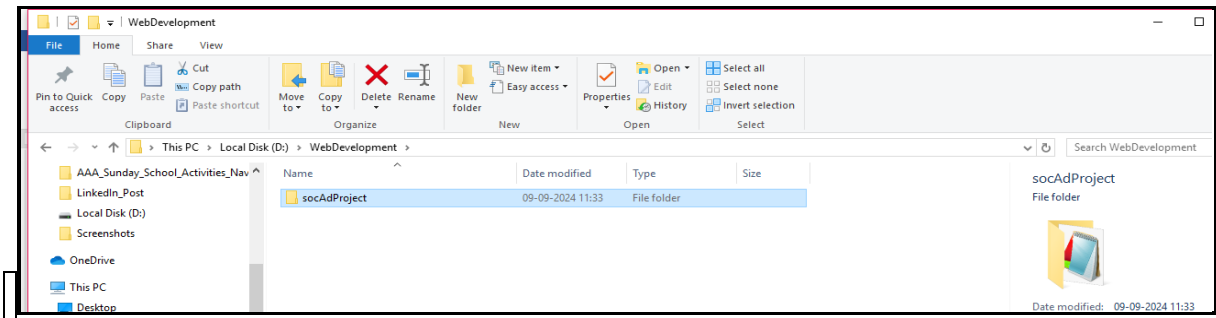


### Step 5: Create a project folder inside the newly created folder using the following command,

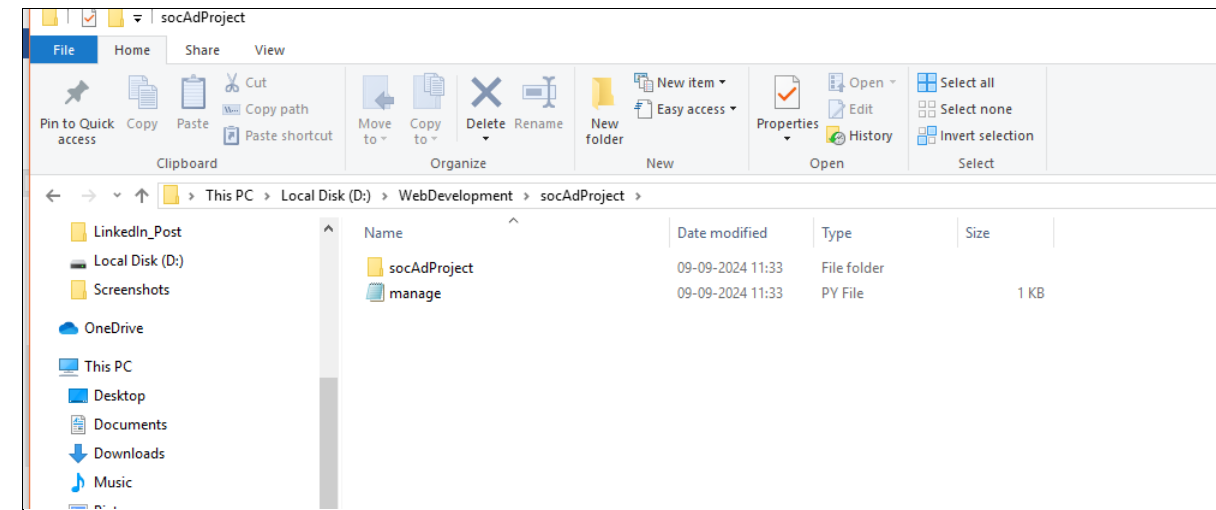
> django-admin startproject socAdProject



**Step 6:** A folder for the project (eg. socAdProject) is created inside the newly created folder.

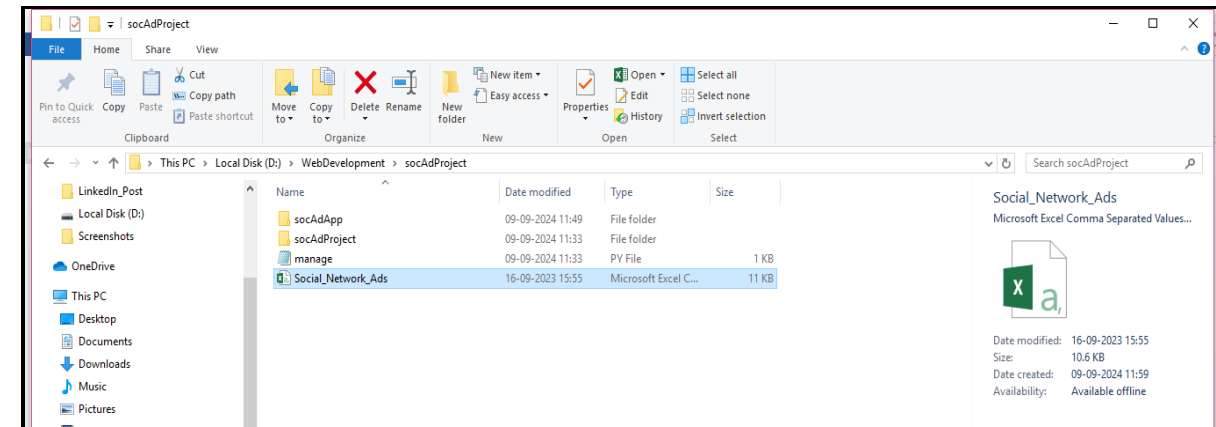


**Step 7:** Inside the Project folder, one Project file folder and a manage.py file will be created.



**Step 8:** Copy and paste the Machine Learning Project dataset (.csv file) inside the Project File Folder.

Local Disk (D:) > WebDevelopment > socAdProject > .csv file



**Step 9:** Navigate to the created folder in the anaconda command prompt using the command, > cd socAdProject

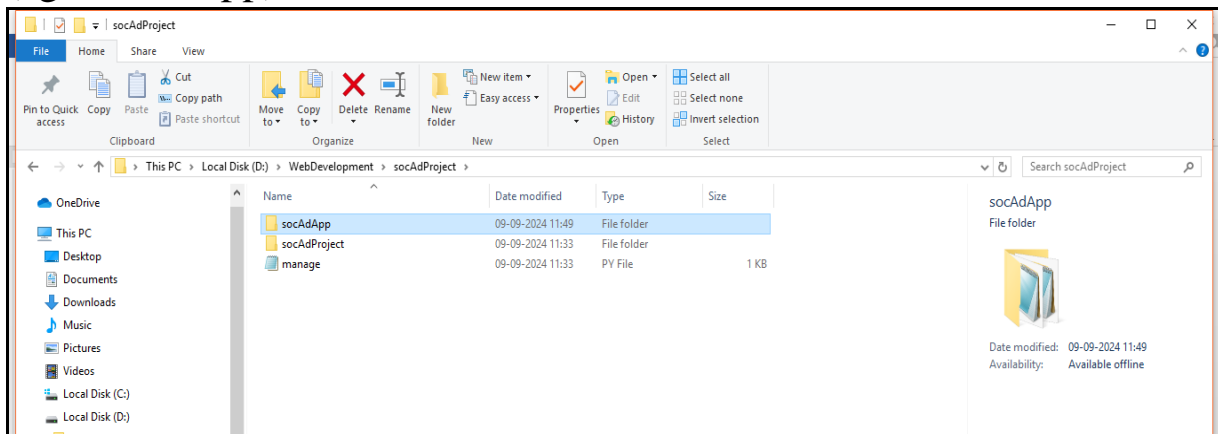
```
Anaconda Prompt
(base) C:\Users\Admin>activate Web
(Web) C:\Users\Admin>d:
(Web) D:\>cd D:\WebDevelopment
(Web) D:\WebDevelopment>django-admin startproject socAdProject
(Web) D:\WebDevelopment>cd socAdProject_
```

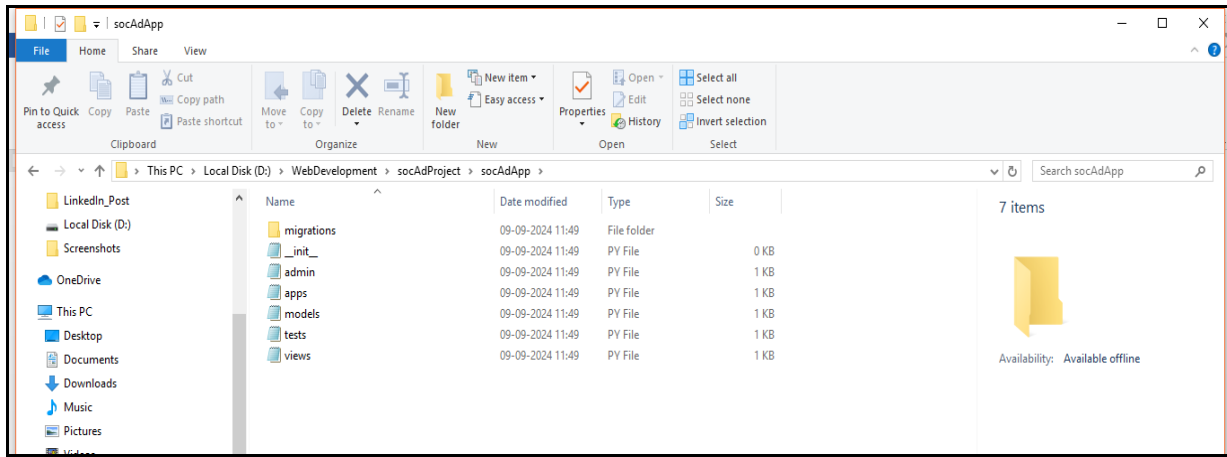
**Step 10:** Create a new folder (eg. socAdApp) for the Application using the following command,

> django-admin startapp socAdApp

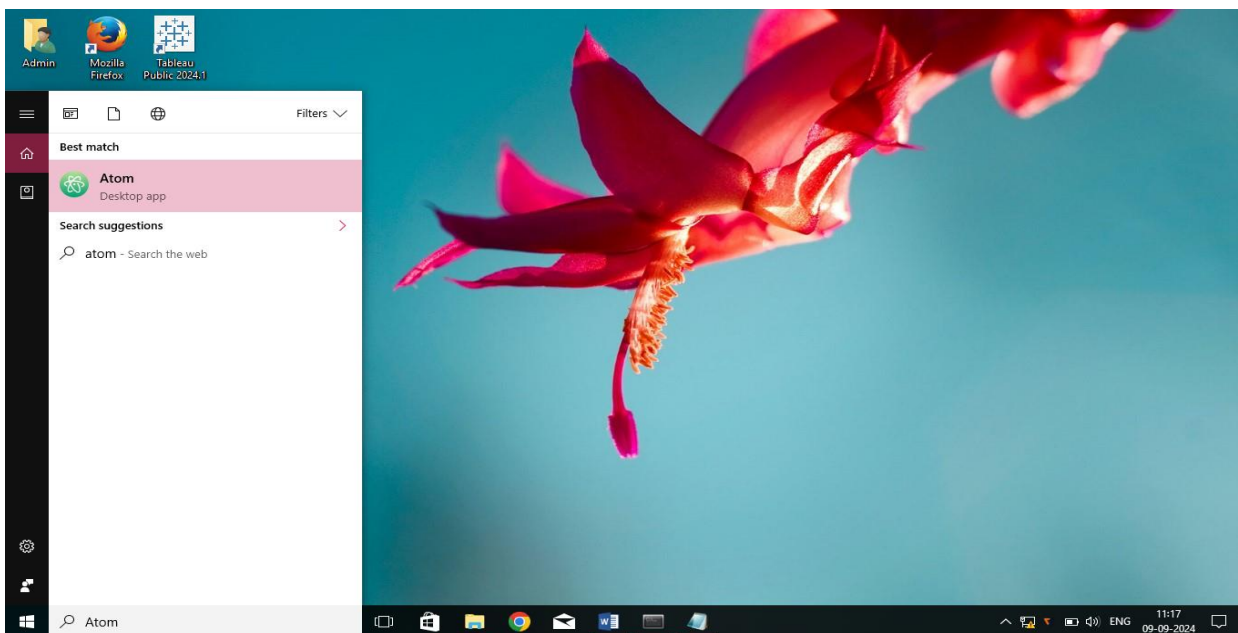
```
Anaconda Prompt
(base) C:\Users\Admin>activate Web
(Web) C:\Users\Admin>d:
(Web) D:\>cd D:\WebDevelopment
(Web) D:\WebDevelopment>django-admin startproject socAdProject
(Web) D:\WebDevelopment>cd socAdProject
(Web) D:\WebDevelopment\socAdProject>django-admin startapp socAdApp_
```

**Step 11:** Go to the Project folder in your local disk, you can notice an App file folder (eg. socAdApp) will be created with some files.

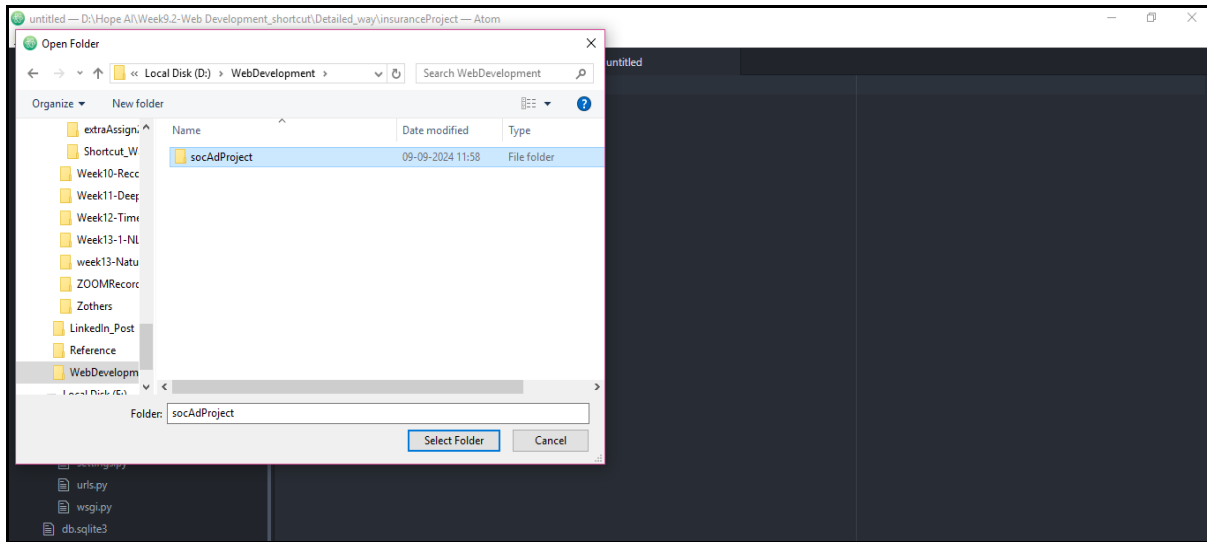




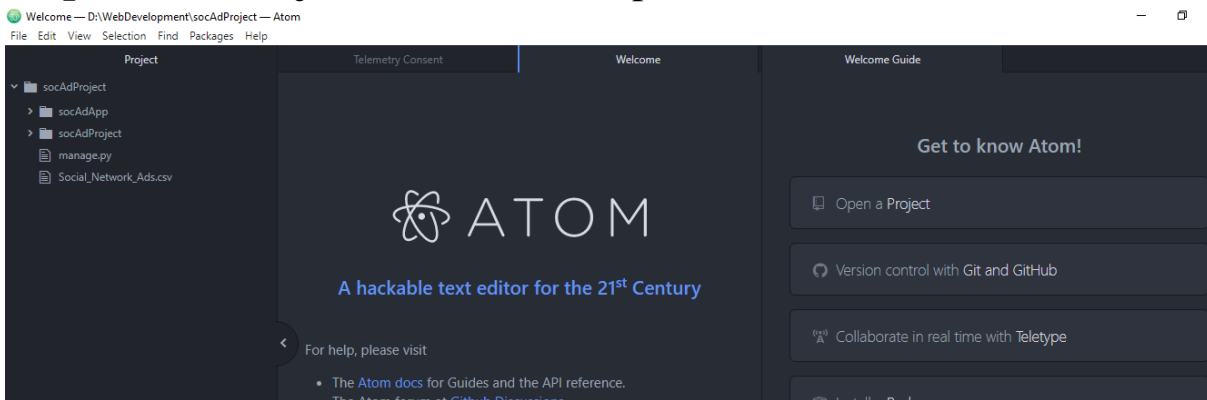
**Step 12:** Now, open the Atom Editor



**Step 13:** Select File > Open Folder > socAdProject ( Project Folder Name)



**Step 14:** The Project folder will be opened inside the Atom editor.



**Step 15:** Open the ckdProject folder in the Atom Editor.

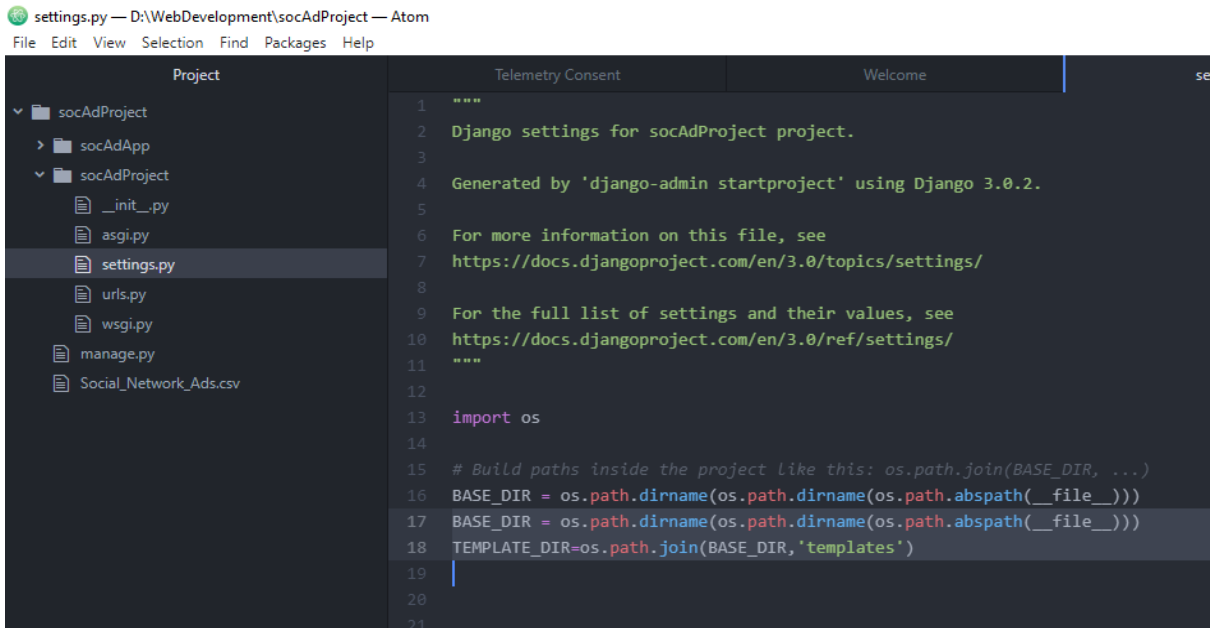
[Note: Do the changes in the forms.py, models.py, views.py, urls.py in our newly created Project folder by referring the above ckdproject folder]

**Step 16:** Click > socAdProject > settings.py

1. Copy the above lines from the ckdProject folder

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```
TEMPLATE_DIR=os.path.join(BASE_DIR,'templates')
```



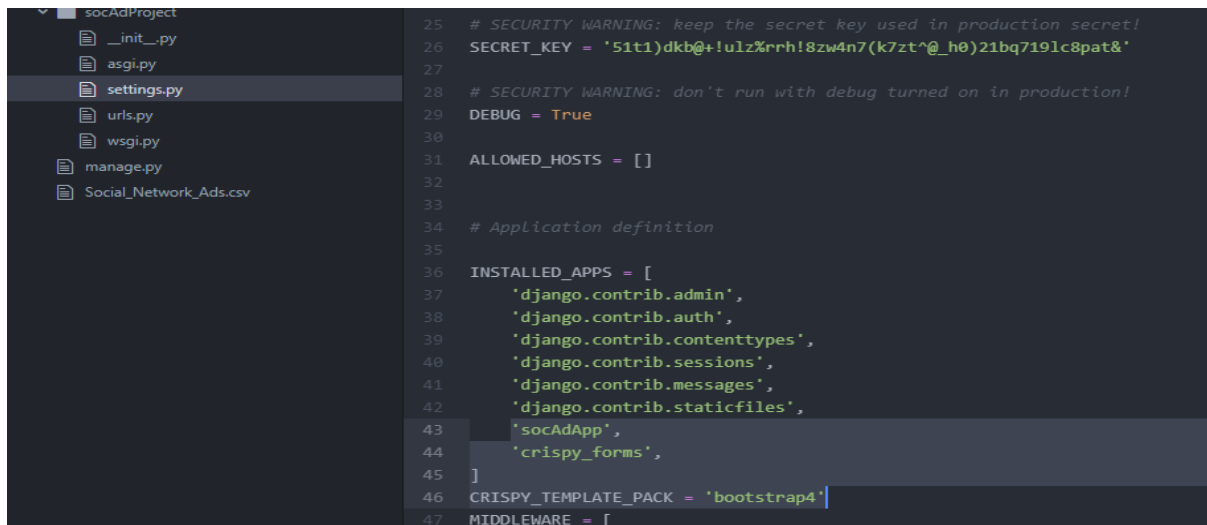
```
settings.py — D:\WebDevelopment\socAdProject — Atom
File Edit View Selection Find Packages Help

Project
  socAdProject
    socAdApp
    socAdProject
      __init__.py
      asgi.py
      settings.py
      urls.py
      wsgi.py
      manage.py
      Social_Network_Ads.csv

Telemetry Consent Welcome

1 """
2 Django settings for socAdProject project.
3
4 Generated by 'django-admin startproject' using Django 3.0.2.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.0/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.0/ref/settings/
11 """
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
18 TEMPLATE_DIR=os.path.join(BASE_DIR,'templates')
19
20
21
```

- a) Scroll down, inside the installed Apps enter the following lines,  
'socAdApp', [Note: enter the 'App' name of the project]  
'crispy\_forms',  
CRISPY\_TEMPLATE\_PACK= 'bootstrap4'



```
socAdProject
  __init__.py
  asgi.py
  settings.py
  urls.py
  wsgi.py
  manage.py
  Social_Network_Ads.csv

25 # SECURITY WARNING: keep the secret key used in production secret!
26 SECRET_KEY = '51t1)dkb@+!ulz%rrh!8zw4n7(k7zt^@_h0)21bq719lc8pat&'
27
28 # SECURITY WARNING: don't run with debug turned on in production!
29 DEBUG = True
30
31 ALLOWED_HOSTS = []
32
33
34 # Application definition
35
36 INSTALLED_APPS = [
37     'django.contrib.admin',
38     'django.contrib.auth',
39     'django.contrib.contenttypes',
40     'django.contrib.sessions',
41     'django.contrib.messages',
42     'django.contrib.staticfiles',
43     'socAdApp',
44     'crispy_forms',
45 ]
46 CRISPY_TEMPLATE_PACK = 'bootstrap4'
47 MIDDLEWARE = [
```

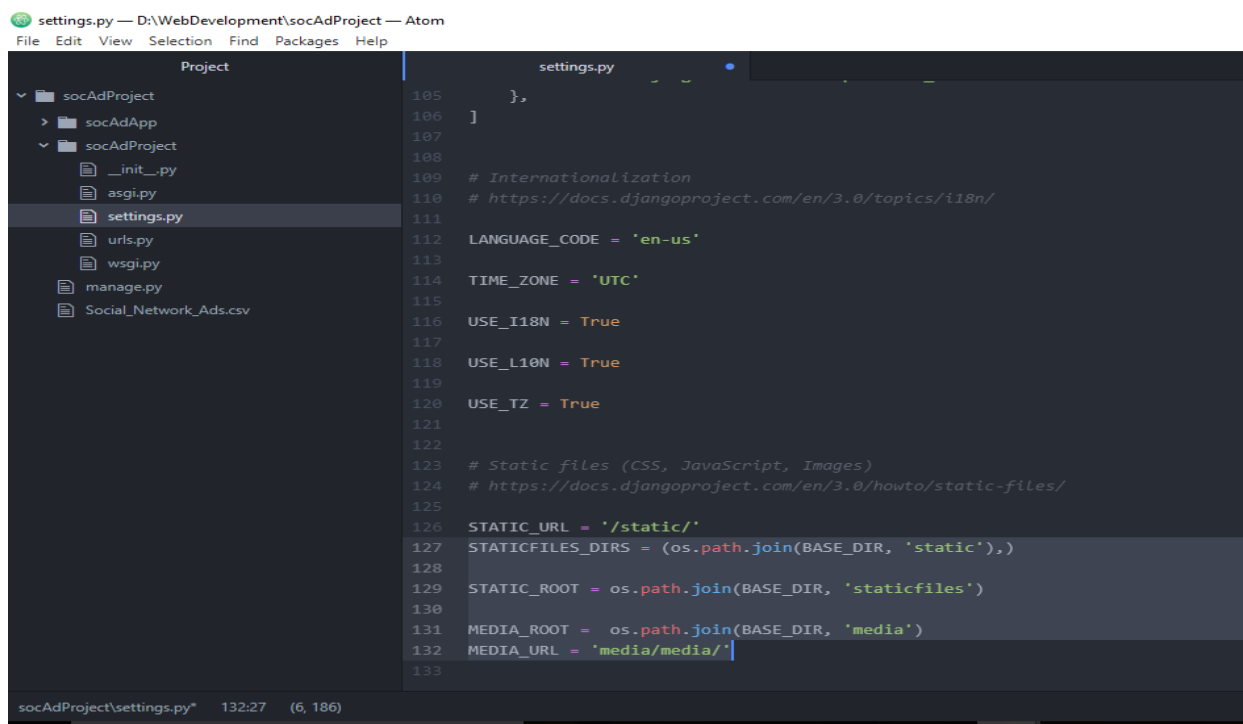
- b) In the Templates variable, enter the line 'DIRS': [TEMPLATE\_DIR,],  
c) Scroll down and come to the STATIC\_URL = '/static/', enter the following lines:

STATICFILES\_DIRS = (os.path.join(BASE\_DIR, 'static'),)

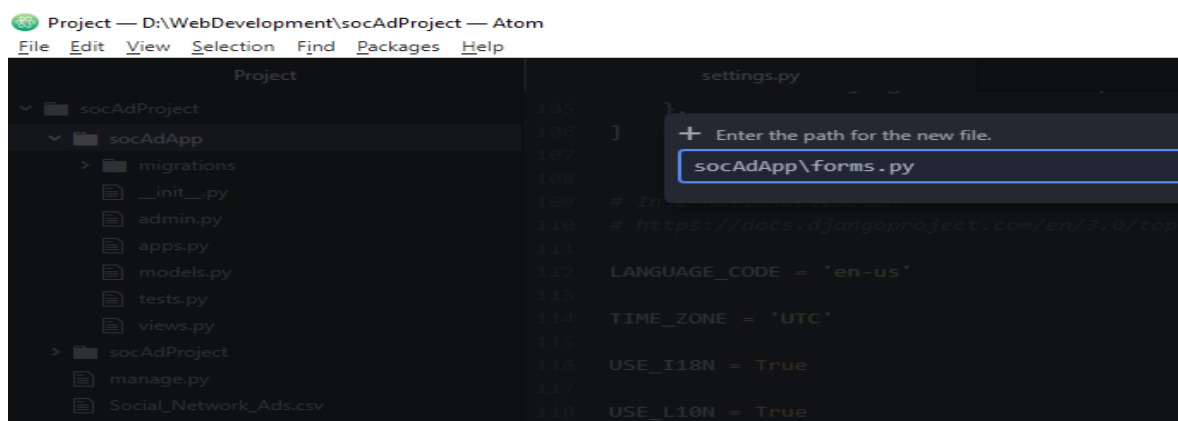
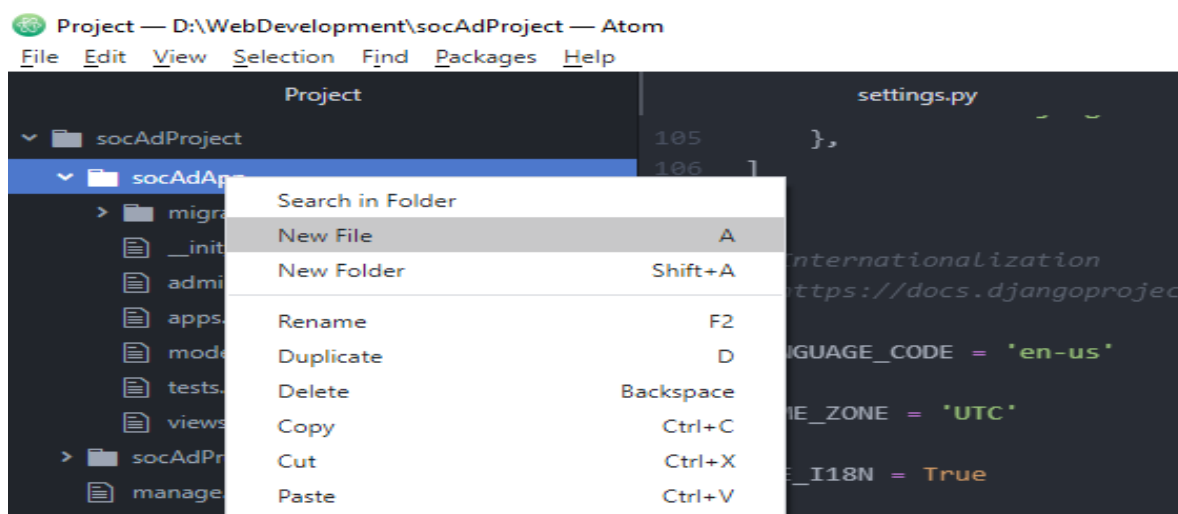
STATIC\_ROOT = os.path.join(BASE\_DIR, 'staticfiles')

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

MEDIA\_URL = 'media/media/'



**Step 17:** Select socAdApp > right click > click New File > enter forms.py

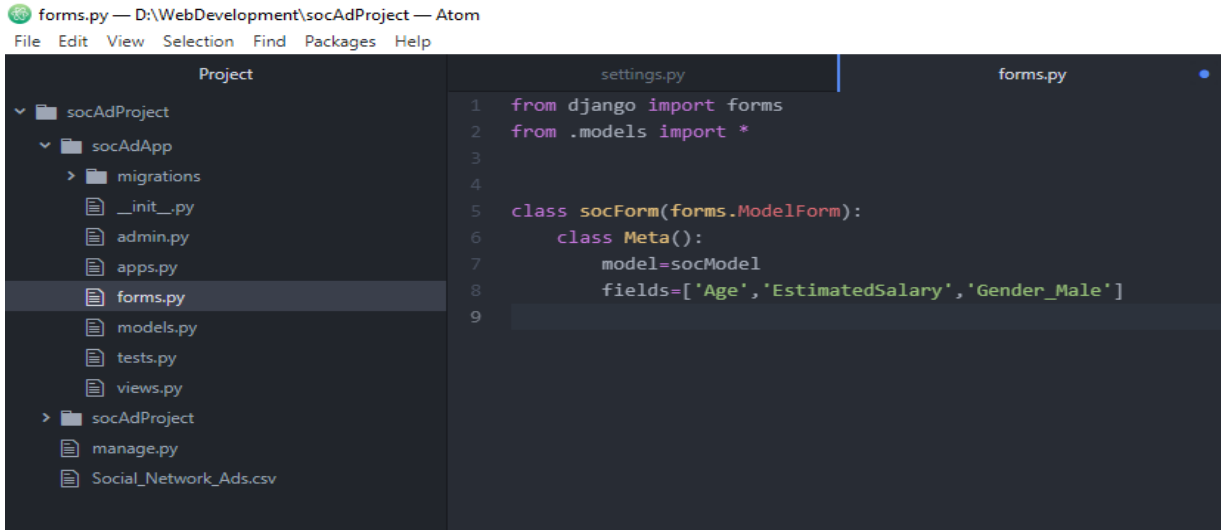




**Step 18:** Inside the forms.py, enter the following lines

1. Give a name for your form (eg. socForm)
2. Give a name for your model. (eg. model= socModel)
3. Enter the input variables name from your dataset in the fields variable .

(Note: Input variable name should be same as it is in the dataset)

The screenshot shows the Atom editor interface. On the left, the 'Project' sidebar displays the file structure of 'socAdProject', with 'forms.py' selected under 'socAdApp'. The main editor pane shows the code in 'forms.py'.

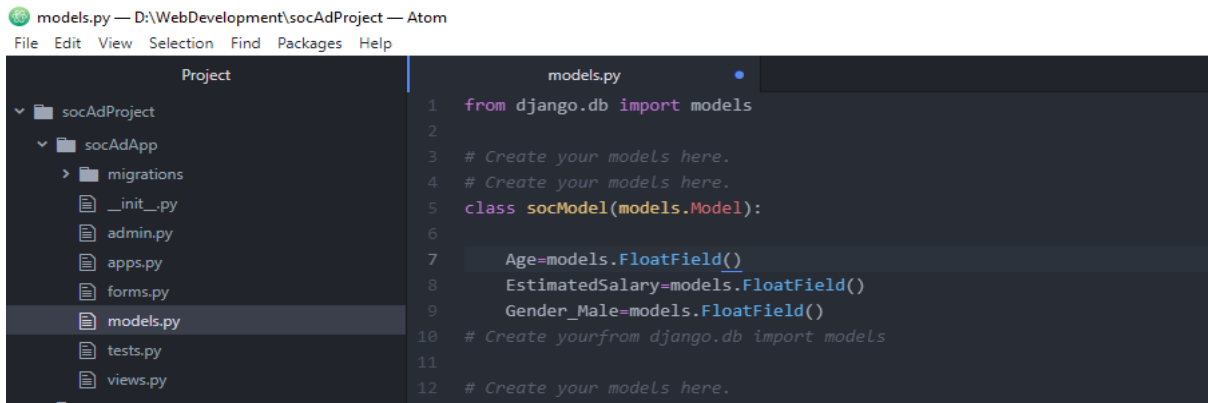
```
forms.py — D:\WebDevelopment\socAdProject — Atom
File Edit View Selection Find Packages Help

Project
  socAdProject
    socAdApp
      migrations
      __init__.py
      admin.py
      apps.py
      forms.py
      models.py
      tests.py
      views.py
    socAdProject
      manage.py
      Social_Network_Ads.csv

settings.py
forms.py
1 from django import forms
2 from .models import *
3
4
5 class socForm(forms.ModelForm):
6     class Meta():
7         model=socModel
8         fields=['Age','EstimatedSalary','Gender_Male']
9
```

**Step 19:** Select socAdApp > models.py,enter the following lines

1. The datatype for the given input variables are assigned.

The screenshot shows the Atom editor interface. On the left, the 'Project' sidebar displays the file structure of 'socAdProject', with 'models.py' selected under 'socAdApp'. The main editor pane shows the code in 'models.py'.

```
models.py — D:\WebDevelopment\socAdProject — Atom
File Edit View Selection Find Packages Help

Project
  socAdProject
    socAdApp
      migrations
      __init__.py
      admin.py
      apps.py
      forms.py
      models.py
      tests.py
      views.py
    socAdProject
      manage.py
      Social_Network_Ads.csv

models.py
1 from django.db import models
2
3 # Create your models here.
4 # Create your models here.
5 class socModel(models.Model):
6
7     Age=models.FloatField()
8     EstimatedSalary=models.FloatField()
9     Gender_Male=models.FloatField()
10 # Create yourfrom django.db import models
11
12 # Create your models here.
```

**Step 20:** Copy the lines from ckdProject > ckdApp > views.py and paste it in the socAdApp > views.py.

Changes to be made are :

1. Assign a variable name for each input variables in which the input values are received for the backend process

views.py — D:\WebDevelopment\socAdProject — Atom

File Edit View Selection Find Packages Help

Project

- socAdProject
  - socAdApp
    - migrations
      - \_\_init\_\_.py
      - \_\_init\_\_.py
      - admin.py
      - apps.py
      - forms.py
      - models.py
      - tests.py
      - views.py

models.py

```

52 filenot_url= reverse_lazy('filenot')
53 def get(self, request, *args, **kwargs):
54     form = self.form_class()
55     return render(request, self.template_name, {'form': form})
56 def post(self, request, *args, **kwargs):
57     #print('inside post')
58     form = self.form_class(request.POST, request.FILES)
59     #print('inside form')
60     if form.is_valid():
61         form.save()
62         data_age=request.POST.get('Age')
63         data_es=request.POST.get('EstimatedSalary')
64         data_gm=request.POST.get('Gender_Male')
65         #print (data)
66         #dataset=pd.read_csv("prep.csv",index_col=None)
67         #dicc={'yes':1,'no':0}

```

views.py

2. Paste the Machine Learning project source code .

3. In the return statement of the def post(): function, enter the newly assigned variable names and the variable out to view the output.

views.py — D:\WebDevelopment\socAdProject — Atom

File Edit View Selection Find Packages Help

Project

- templates
  - aboutus.html
  - base.html
  - create.html
  - fail.html
  - filenot.html
  - out.html
  - process.html
  - ref.html
  - succ\_msg.html
- socAdProject
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - forms.py
  - models.py
  - tests.py
  - urls.py
  - views.py

base.html

succ\_msg.html

aboutus.html

views.py

```

62 data_age=request.POST.get('Age')
63 data_es=request.POST.get('EstimatedSalary')
64 data_gm=request.POST.get('Gender_Male')
65 import pandas as pd
66 dataset=pd.read_csv("Social_Network_Ads.csv")
67 dicc={'yes':1,'no':0}
68 dataset=pd.get_dummies(dataset,drop_first=True)
69 dataset=dataset.drop("User ID",axis=1)
70 independent=dataset[['Age','EstimatedSalary','Gender_Male']]
71 dep=dataset["Purchased"]
72 from sklearn.model_selection import train_test_split
73 X_train,X_test,Y_train,Y_test=train_test_split(independent,dep,test_size=1/3,random_state=0)
74 #predicting the output value using Gaussian Navie Bayes classifier
75 from sklearn.naive_bayes import GaussianNB
76 classifier=GaussianNB()
77 classifier.fit(X_train,Y_train)
78 Y_pred=classifier.predict(X_test)
79 from sklearn.metrics import confusion_matrix
80 cm=confusion_matrix(Y_test,Y_pred)
81 from sklearn.metrics import classification_report
82 clf_report=classification_report(Y_test,Y_pred)
83 data =np.array([data_age,data_es,data_gm])
84 if data.dtype.kind in 'uO': # 'U' for unicode, 'O' for object (generic Python objects)
85     data = data.astype(float)
86 out=classifier.predict(data.reshape(1,-1))
87 return render(request, "succ_msg.html", {'data_age':data_age,'data_es':data_es,'data_gm':data_gm,'out':out})
88 else:
89     return redirect(self.failure_url)

```

forms.py

models.py

tests.py

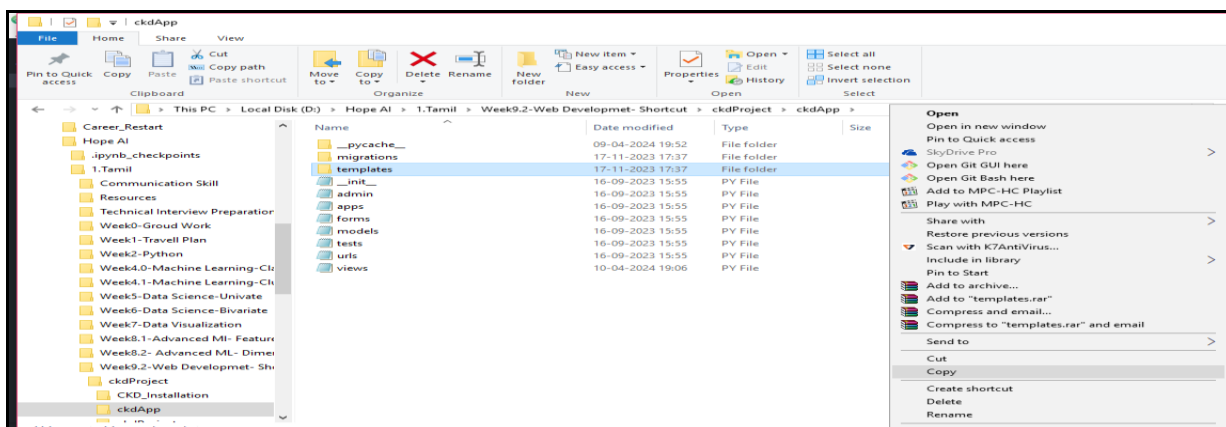
out.html

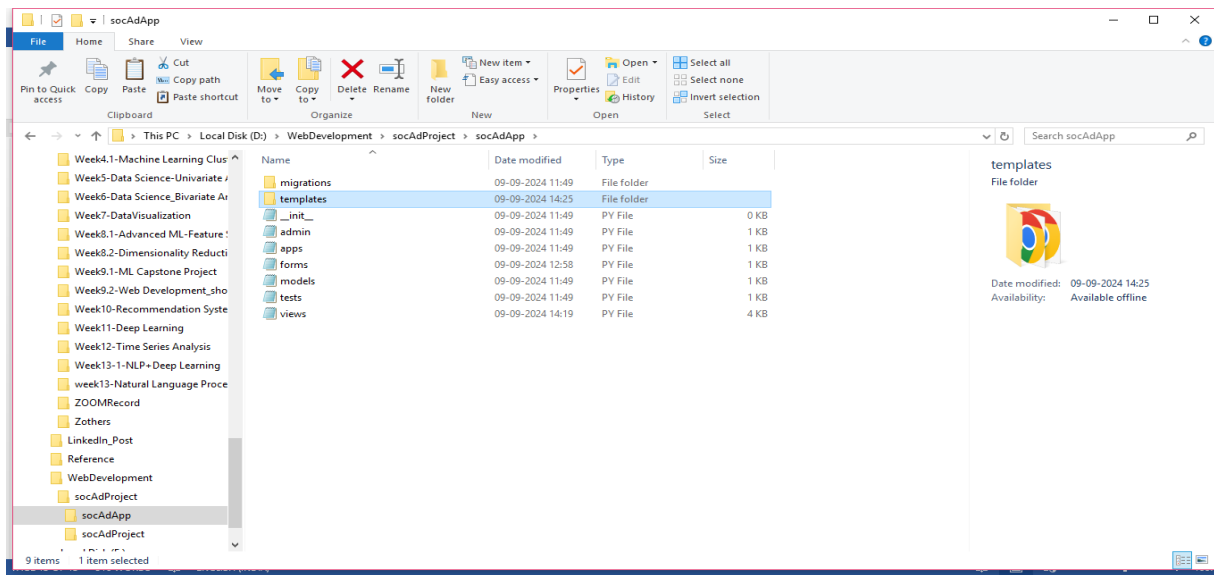
filenot.html

create.html

urls.py — s...

**Step 21:** Open the ckdproject folder , Select ckddapp > templates > copy the file and paste it in the socAdProject > socAdApp folder.





## Step 22: Click templates > succ\_msg.html > change the following

1. Change the Heading of the project (eg. Social Network Ad Classification)
2. Change the input variables name to be displayed in the output page.
3. If it is ML classification project,

```
{% if out == 1 %}
```

```
<p>Probability to Purchase is high</p>
```

```
{% else %}
```

```
<p>Probability to purchase is low</p>
```

```
{% endif %}
```

succ\_msg.html — D:\WebDevelopment\socAdProject — Atom

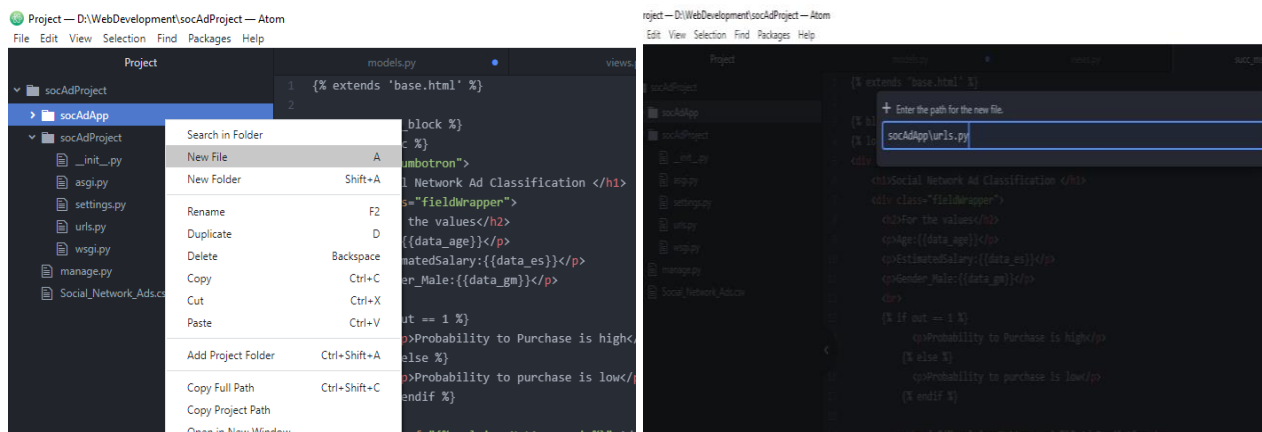
```
File Edit View Selection Find Packages Help

Project
  templates
    aboutus.html
    base.html
    create.html
    fail.html
    fileno.html
    out.html
    process.html
    ref.html
    succ_msg.html
    _init_.py
    admin.py
    apps.py
    forms.py
    models.py
    tests.py
    urls.py
    views.py
  socAdProject
    __pycache__
    _init_.py
    ...

base.html | succ_msg.h... | aboutus.html | views.py | forms.py | models.py | tests.py | out.html | fileno.html | create.html | ur...

1  {% extends 'base.html' %}
2
3  {% block body_block %}
4  {% load static %}
5  <div class="jumbotron">
6    <h1>Social Network Ad Classification </h1>
7    <div class="fieldWrapper">
8      <h2>For the values</h2>
9      <p>Age:{{data_age}}</p>
10     <p>EstimatedSalary:{{data_es}}</p>
11     <p>Gender_Male:{{data_gm}}</p>
12     <br>
13     {% if out == 1 %}
14       <p>Probability to Purchase is high</p>
15     {% else %}
16       <p>Probability to purchase is low</p>
17     {% endif %}
18
19     <a href="{% url 'socAdApp:soc' %}">Ad Prediction</a>
20     <br>
21
22     <br>
23   </div>
24 {% endblock %}
```

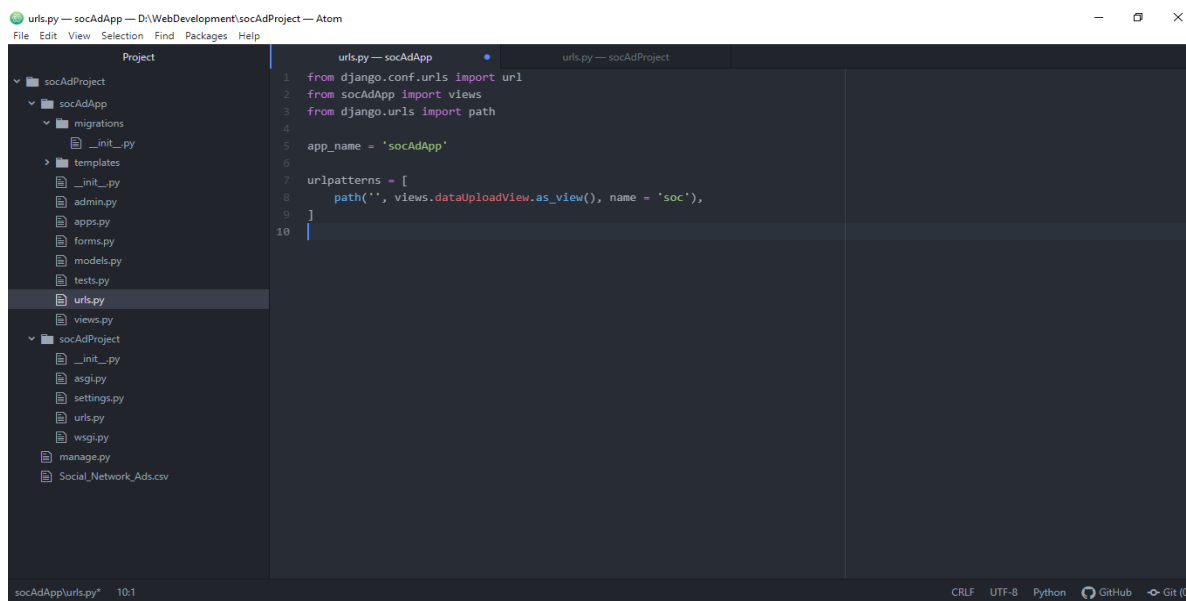
## Step 23: Right click socAdApp > Select New File > enter urls.py



1. Select > socAdApp > urls.py and paste the copied lines from the ckdProject > ckdApp > urls.py file

Changes to be made are:

1. Enter the project App name in the libraries installed (eg. from socAdApp import views)
2. Enter the project App name in the app\_name variable. (eg. socAdApp)
3. Give a name for the dataUploadview (eg. soc)

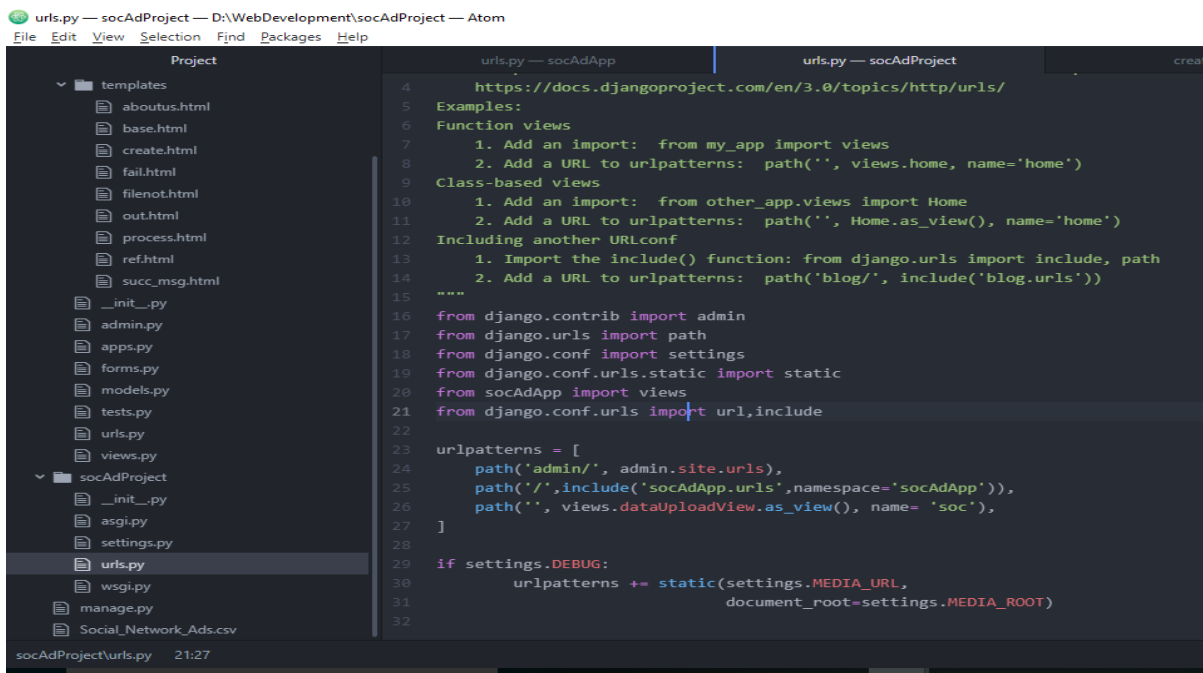


## Step 24: Select socAdProject > urls.py py and paste the copied lines from the ckdProject > ckdProject > urls.py file

Changes to be made are:

1. Enter the project App name in the libraries installed (eg. from socAdApp import views).
2. In the urlspattern, change the name of the urls to be included as “socAdApp.urls” and the namespace as “socAdApp”.

### 3. Give a name for the dataUploadview (eg. soc)



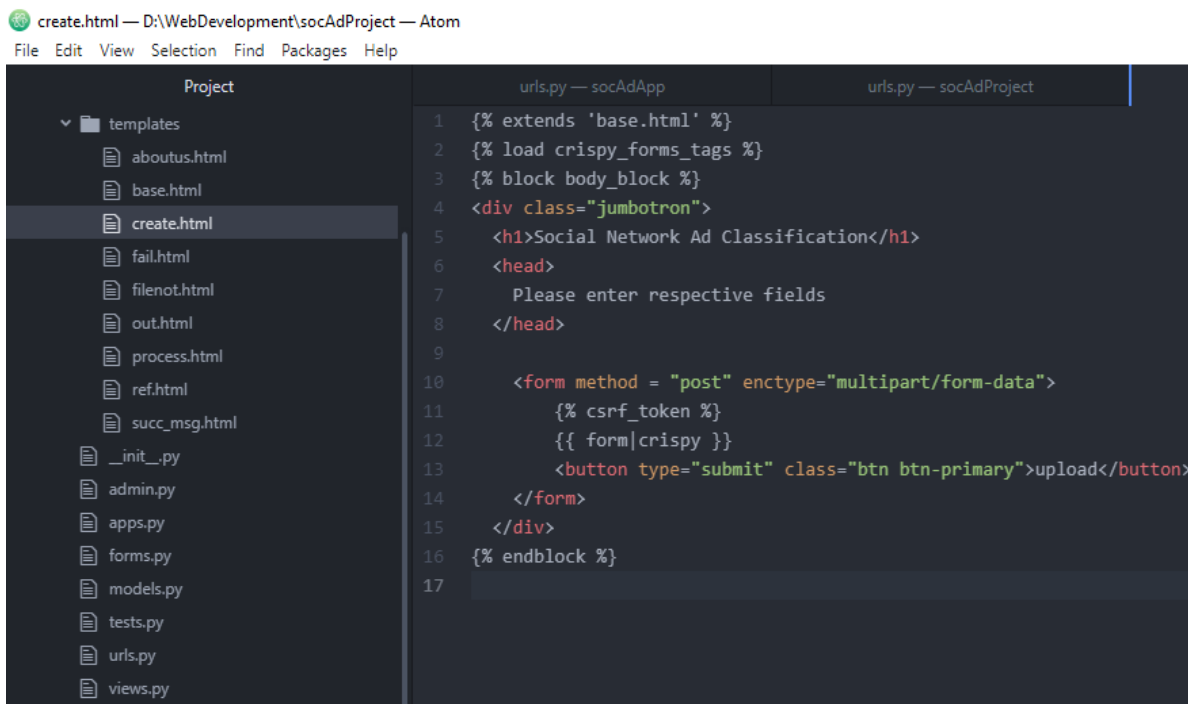
The screenshot shows the Atom editor with the file `urls.py` open. The left sidebar displays the project structure, including a `templates` folder and a `socAdProject` folder. The main editor area shows the following Python code:

```
4 https://docs.djangoproject.com/en/3.0/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path
18 from django.conf import settings
19 from django.conf.urls.static import static
20 from socAdApp import views
21 from django.conf.urls import url, include
22
23 urlpatterns = [
24     path('admin/', admin.site.urls),
25     path('', include('socAdApp.urls', namespace='socAdApp')),
26     path('', views.dataUploadView.as_view(), name='soc'),
27 ]
28
29 if settings.DEBUG:
30     urlpatterns += static(settings.MEDIA_URL,
31                           document_root=settings.MEDIA_ROOT)
32
```

## Step 25: Select socAdApp > templates > create.html

Changes to be made are:

1. Change the name of the Project to be displayed.  
(eg. Social Network Ad Classification)



The screenshot shows the Atom editor with the file `create.html` open. The left sidebar displays the project structure, including a `templates` folder and a `socAdProject` folder. The main editor area shows the following HTML code:

```
1 {% extends 'base.html' %}
2 {% load crispy_forms_tags %}
3 {% block body_block %}
4 <div class="jumbotron">
5     <h1>Social Network Ad Classification</h1>
6     <head>
7         Please enter respective fields
8     </head>
9
10     <form method = "post" enctype="multipart/form-data">
11         {% csrf_token %}
12         {{ form|crispy }}
13         <button type="submit" class="btn btn-primary">upload</button>
14     </form>
15 </div>
16 {% endblock %}
17
```

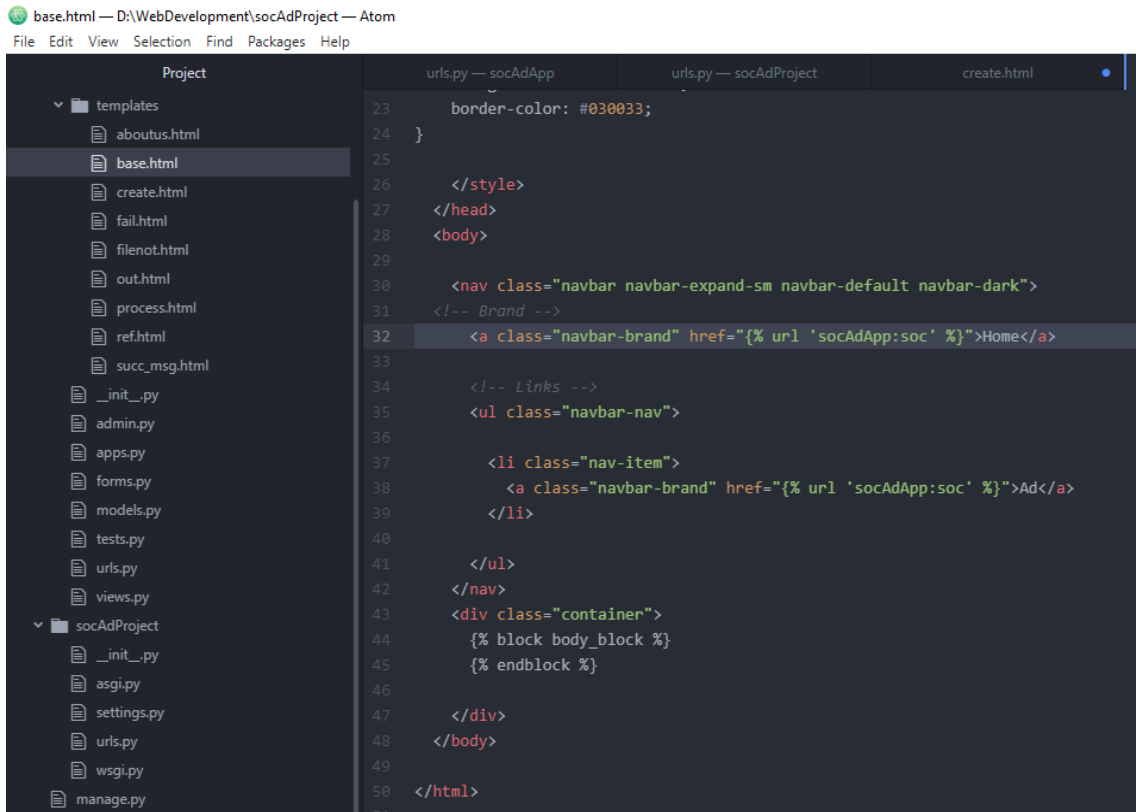
## Step 26: Select socAdApp > templates > base.html

Do the following changes,

1. Rename the Project App name and namespace (eg. socAdfApp:soc)

```
<a class="navbar-brand" href="{ % url 'socAdApp:soc' % }">Home</a>
```

```
<a class="navbar-brand" href="{ % url 'socAdApp:soc' % }">Ad</a>
```



**Step 27:** Select File > Save All.

**Step 28:** Come to the anaconda command prompt and enter the commands,

```
>python manage.py migrate
```

```
>python manage.py makemigrations socAdApp
```

```
>python manage.py createsuperuser
```

```
username:SocialAd
```

```
pwd:soc1234
```

```
confirm pwd: soc1234
```

```
>python manage.py migrate --run-syncdb
```

```
>python manage.py runserver
```

```
(base) C:\Users\Admin>activate Web
(Web) C:\Users\Admin>d:
(Web) D:\>cd D:\WebDevelopment
(Web) D:\WebDevelopment>django-admin startproject socAdProject
(Web) D:\WebDevelopment>cd socAdProject
(Web) D:\WebDevelopment\socAdProject>django-admin startapp socAdApp
(Web) D:\WebDevelopment\socAdProject>python manage.py migrate_
```

```
Anaconda Prompt - python manage.py runserver
(Web) D:\WebDevelopment\socAdProject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, socAdApp
Running migrations:
  No migrations to apply.

(Web) D:\WebDevelopment\socAdProject>python manage.py makemigrations socAdApp
No changes detected in app 'socAdApp'

(Web) D:\WebDevelopment\socAdProject>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, socAdApp
Running migrations:
  No migrations to apply.

(Web) D:\WebDevelopment\socAdProject>python manage.py createsuperuser
Username (leave blank to use 'admin'): WebDeploy
Email address:
Password:
Password (again):
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

(Web) D:\WebDevelopment\socAdProject>python manage.py migrate --run-syncdb
Operations to perform:
  Synchronize unmigrated apps: crispy_forms, messages, staticfiles
  Apply all migrations: admin, auth, contenttypes, sessions, socAdApp
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
Running migrations:
  No migrations to apply.

(Web) D:\WebDevelopment\socAdProject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 09, 2024 - 17:13:30
Django version 3.0.2, using settings 'socAdProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[09/Sep/2024 17:13:41] "GET / HTTP/1.1" 200 2666
```

# Output

Input Form:

1:8000

127.0.0.1:8000

D

### Social Network Ad Classification

Please enter respective fields

Age\*

EstimatedSalary\*

Gender Male\*

upload

## Output Page

HomeAd

### Social Network Ad Classification

#### For the values

Age:54.0

EstimatedSalary:35000.0

Gender\_Male:0.0

Probability to Purchase is high

Ad Prediction