|  |  |
| --- | --- |
| Assignment No. | 1 |
| Name | JANANI R |
| Registration No. | 921319104069 |
| Batch No. | B7-1A3E |

# Question-1:

**Createregistrationpageinhtmlwithusername,email,andphonenumberandby usingPOSTmethoddisplayitinnexthtmlpage.**

# index.html

<!DOCTYPEhtml>

<html>

<head>

<metacharset="utf-8">

<metaname="viewport"content="width=device-width,initial-

scale=1">

<title>Registration</title>

<linkrel="stylesheet"href="{{

url\_for('static',filename='styles/index.css')}}">

</head>

<body>

<center>

<br><br>

<h1>Registration</h1><br>

<formaction="{{url\_for('result')}}"method="post">

<table>

<tr>

<td>

</td>

<td>

<label>Name</label>

:<inputtype="text"class="name-input

namemb-3"id="name"name="name">

</td>

</tr>

<tr>

<td>

</td>

<td>

<label>Email</label>

:<inputtype="email"class="name-

inputnamemb-3"id="email"name="email">

</td>

</tr>

<tr>

<td>

</td>

<td>

<label>Mobile</label>

:<inputtype="number"class="name-

inputnamemb-3"id="mobile"name="mobile">

</td>

value="Submit">

</tr>

</table>

<br><br>

<inputclass="btnbtn-outline-primary"type="submit"

</form>

</center>

</body>

</html>

# result.html

<!DOCTYPEhtml>

<html>

<head>

<metacharset="utf-8">

<metaname="viewport"content="width=device-width,initial-

scale=1">

<title>Profile</title>

<linkrel="stylesheet"href="{{

url\_for('static',filename='styles/index.css')}}">

</head>

<body>

<center>

<br><br>

<h1>Welcome!!!</h1>

<br>

<formaction="{{url\_for('result')}}"method="post">

<table>

<tr>

</tr>

<tr>

</tr>

<tr>

<h3>Name:{{name}}<br></h3>

<h3>Email:{{email}}<br></h3>

<h3>Mobile:{{mobile}}<br></h3>

</tr>

</table>

<br><br>

</form>

</center>

</body>

</html>

# app.py

from flask import \*;importos

app=Flask(name)

@app.route('/',methods=['GET','POST'])defhome():

if request.method=='POST':name = request.form["name"]email=request.form["email"]

mobile=request.form["mobile"]

returnredirect(url\_for('result',name=name,email=email,mobile=mobile))

returnrender\_template('index.html')

@app.route("/result",methods=['GET','POST'])defresult():

name = request.form.get('name')email= request.form.get('email')mobile=request.form.get('mobile')

returnrender\_template('result.html',name=name,email=email,mobile=mobile)

ifname== "main":app.run(debug=True,port=2807)

# OUTPUT:



**Question-2:**

# DevelopaFlaskprogramwhichshouldcontainatleast5packagesusedfrompypi.org

*#1.NUMPY*

importnumpyas np

arr=np.array([[-1, 2, 0, 4],

[4,-0.5,6,0],

[2.6,0,7,8],

[3,-7, 4,2.0]])

print("InitialArray:")print(arr)

InitialArray:

[[-1. 2. 0. 4.]

[4.-0.56. 0.]

[2.60. 7. 8.]

[3.-7. 4. 2.]]

*#2.PANDAS*

importpandasaspd

s1 =pd.Series([1,3,4,5, 6,2,9])

s2=pd.Series([1.1,3.5,4.7, 5.8,2.9, 9.3])

s3=pd.Series(['a', 'b','c','d','e'])Data={'first':s1,'second':s2,'third':s3}

df=pd.read\_csv('/content/sample\_data/ds\_salaries.csv')print(df)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Unnamed: 0 | work\_year | experience\_level | employment\_type | \ |
| 0 | 0 | 2020 | MI | FT |  |
| 1 | 1 | 2020 | SE | FT |  |
| 2 | 2 | 2020 | SE | FT |  |
| 3 | 3 | 2020 | MI | FT |  |
| 4 | 4 | 2020 | SE | FT |  |
| .. | ... | ... | ... | ... |  |
| 602 | 602 | 2022 | SE | FT |  |
| 603 | 603 | 2022 | SE | FT |  |
| 604 | 604 | 2022 | SE | FT |  |
| 605 | 605 | 2022 | SE | FT |  |
| 606 | 606 | 2022 | MI | FT |  |

job\_titlesalarysalary\_currencysalary\_in\_usd

\

1. DataScientist 70000 EUR 79833
2. Machine LearningScientist260000 USD 260000
3. BigData Engineer 85000 GBP 109024

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | Product DataAnalyst | | 20000 | | USD | | 20000 |
| 4 | MachineLearning Engineer | | 150000 | | USD | | 150000 |
| .. | ... | | ... | | ... | | ... |
| 602 | DataEngineer | | 154000 | | USD | | 154000 |
| 603 | DataEngineer | | 126000 | | USD | | 126000 |
| 604 | DataAnalyst | | 129000 | | USD | | 129000 |
| 605 | DataAnalyst | | 150000 | | USD | | 150000 |
| 606 | AIScientist | | 200000 | | USD | | 200000 |
|  | employee\_residence | remote\_ratio | | company\_location | | company\_size | |
| 0 | DE | 0 | | DE | | L | |
| 1 | JP | 0 | | JP | | S | |
| 2 | GB | 50 | | GB | | M | |
| 3 | HN | 0 | | HN | | S | |
| 4 | US | 50 | | US | | L | |
| .. | ... | ... | | ... | | ... | |
| 602 | US | 100 | | US | | M | |
| 603 | US | 100 | | US | | M | |
| 604 | US | 0 | | US | | M | |
| 605 | US | 100 | | US | | M | |
| 606 | IN | 100 | | US | | L | |

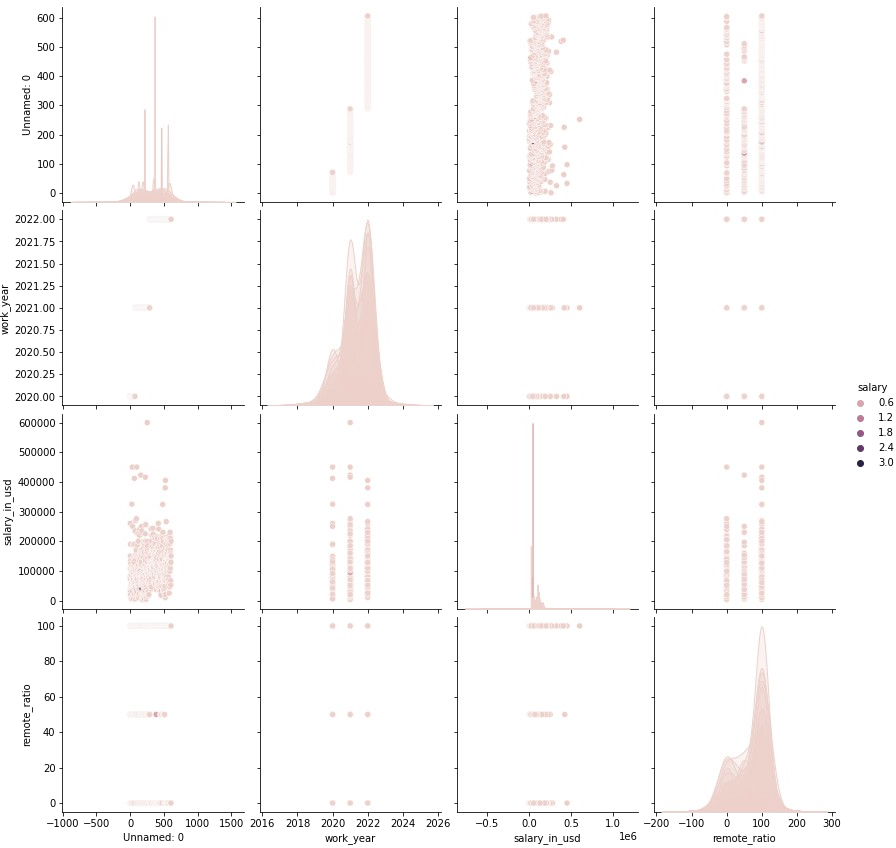
[607rows x12 columns]

*#3.SEABORN*

importseabornassns

sns.pairplot(df,hue="salary",height=3)

<seaborn.axisgrid.PairGrid at0x7f32024d5650>



*#4.TENSORFLOW*

importtensorflowas tf

mnist =tf.keras.datasets.mnist

(x\_train, y\_train),(x\_test,y\_test)=mnist.load\_data()x\_train,x\_test=x\_train/255.0,x\_test/ 255.0

model=tf.keras.models.Sequential([tf.keras.layers.Flatten(input\_shape=(28, 28)),tf.keras.layers.Dense(128, activation='relu'),tf.keras.layers.Dropout(0.2),tf.keras.layers.Dense(10)

])

*#5.PYTZ*

frompytzimporttimezonefromdatetime importdatetime

format= "%Y-%m-%d %H:%M:%S %Z%z"

now\_utc = datetime.now(timezone('UTC'))print(now\_utc.strftime(format))

now\_asia= now\_utc.astimezone(timezone('Asia/Kolkata'))print(now\_asia.strftime(format))

2022-10-0716:36:09UTC+0000

2022-10-0722:06:09IST+0530