

AIM: Run a basic html page in web browser using python.

STEPS:

- (1) Create a file named app.py .

```
import webbrowser  
import os  
file_path = os.path.abspath('index.html')  
webbrowser.open(f'file:///{file_path}')
```

- (2) Now create the HTML file named index.html .

```
<!DOCTYPE HTML>  
<html>  
<head>  
<title>My First Program </title>  
</head>  
<body>  
<h2>My Dear Students </h2>  
<p>This is my first html page...</p>  
</body>  
</html>
```

Note: Run file using Jupyter lab .

(3) Now Run the Program.

open terminal and ~~run~~ type python
app.py then Enter.

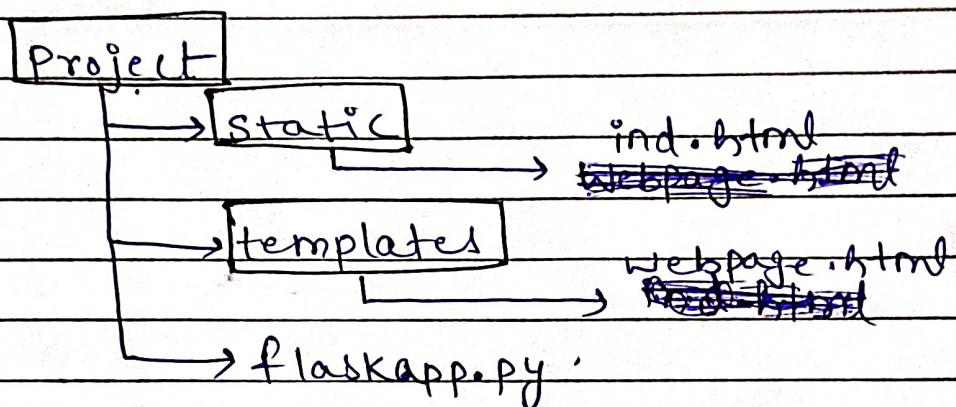
RESULT: The ~~HTML~~ program runs on
browser by the help of python
directly.

Why

AIM: Using flask render html files with hyperlink.

RESOURCES: ~~Jupyterlab~~, Jupyterlab, analonda.

FOLDER MANAGEMENT:



STEPS:

- (1) Open terminal and type ~~Jupyter lab~~ then enter.
- (2) On Jupyter lab create a project folder
→ static and template folder as above shown.
- (3) webpage.html is placed in templates,
ind.html is placed in static and
flaskapp.py is placed in project folder.

(4) flask app.py code :

```

from flask import Flask, render_template
app = Flask(__name__)
@app.route('/')
def home():
    return render_template('webpage.html')
if __name__ == '__main__':
    app.run(debug='True')

```

webpage.html code:

```

<!DOCTYPE HTML>
<html lang='en'>
<head>
    <meta charset='UTF-8'>
    <title> My Html </title>
</head>
<body>
    <h1>Welcome to the html Basics .</h1>
    <p> This is my first Html program </p>
    <a href="{{ url_for('static', filename=
        'index.html') }}> target='_blank'>

```

[visit page-2](#)

```

<ul>
  <li>HTML </li>
  <li>CSS </li>
  <li>JAVASCRIPT </li>
</ul>
</body>
</html>

```

ind.html code:

```

<!DOCTYPE HTML>
<html lang='en'>
  <head>
    <meta charset='UTF-8'>
    <title> second page </title>
  </head>
  <body>
    <h1> It is the second page of HTML </h1>
    <p> This is my second html page
      program. </p>
  </body>
</html>

```

- (5) To Run open terminal & enter
 "python flaskapp.py"

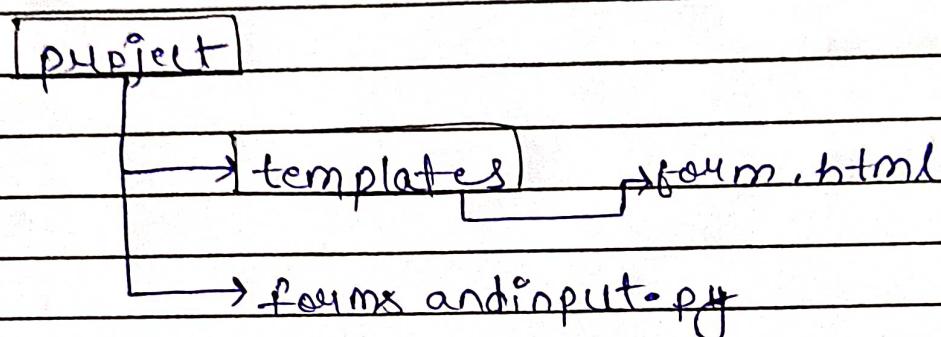
(6) Now click on the port to open browser.

~~RESULT: The hyperlinked programs works fine and rendered on browser using flask.~~

AIM: Form and Input Handling in HTML.

RESOURCES: Jupyter lab, anaconda.

FOLDER MANAGEMENT:



STEPS:

- ① open terminal and type Jupyter lab then enter.
- ② on Jupyter lab create a project folder
→ templates and forms and input.py are inside it → form.html is inside templates.
- ③ forms and input.py code:

```
from flask import FLASK, render_template  
.request  
app = FLASK(__name__)
```

Expt. No. _____

```

@app.route('/', methods = ['GET', 'POST'])
def form_page():
    if request.method == 'POST':
        name = request.form.get('name')
        email = request.form.get('email')
        return f"<h2>Received Input:</h2>
                <p>Name : {name}</p><p>Email:
                {email}</p>"
    request.render_template('form.html')

if __name__ == '__main__':
    app.run(debug = True)

```

Form.html code:

```

<!DOCTYPE html>
<html lang = "en">
    <head>
        <meta charset = "UTF-8">
        <title>form Example</title>
    </head>
    <body>
        <h2>Input form</h2>
        <form method = "POST" action = "/">
            <label for = "name"> Name : </label>
            <input type = "text" id = "name" name =

```

Date: _____

Teacher's Signature: _____

"name" required > ~~

~~

```
<label for="email"> Email: </label>
<input type="email" id="email" name=
"email" required> <br> <br>
```

```
<button type="submit"> Submit </
button>
```

```
</form>
</body>
</html>
```

RESULT: This form and input program
runs perfectly and gives receives
~~the~~ input correctly.