# Web Worksheet 6: Introduction to Flask

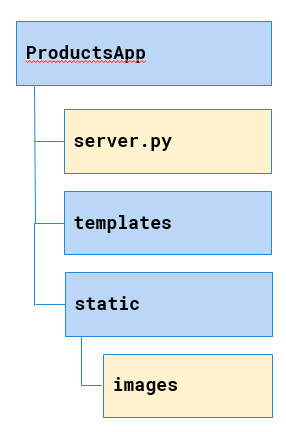
## References:

* <https://www.tutorialspoint.com/flask/index.htm>. Application, Routing.
* Miguel Grinberg (2014). *Flask Web Development(1st Ed.)*.O’Reilly. Chapter 2.

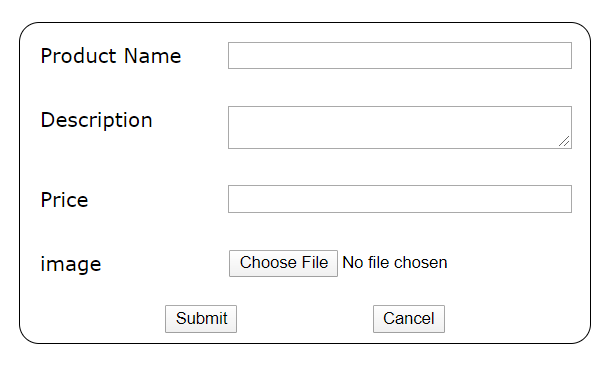
Perquisites: Completed the form worksheet in previous lesson. (Web Worksheet 5)

**Part 1: Web App Folder Hierachy**

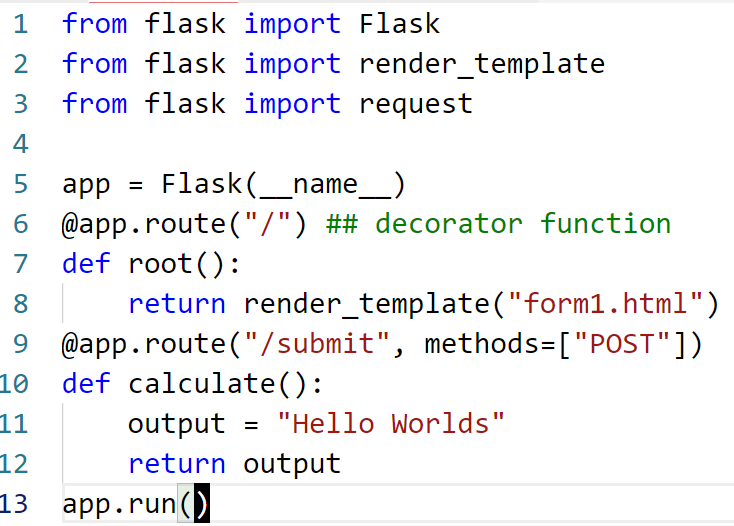
1. Make a new **folder** to house your Flask project. Let’s call it ProductsApp, which is our first project. You can create the folder on your desktop for easy access.
2. Inside ProductsApp, add a new **file**. Call it server.py. This is where we are going to write our Python code.
3. Add two subfolders under ProductsApp named templates and static



1. Copy your form1.html (whatever name you use) to the **templates** folder. (form1 was created from the previous lesson, Web Worksheet 5)



1. Modify your form1.html so that
   1. the form is submitted to the URL "/submit" and use the "POST" method to submit the form. (The full URL is http://localhost:5000/submit) . When a relative path is specified ("/submit"), the web browser will send it to where the existing web page comes from.
   2. Every input element has a name attribute with a value identifying the input data.
2. In the server.py file enter the following code:



1. Run the python program. (from IDLE or VS Code)

232

1. Open the Chrome web browser and access the Web Application via the URL <http://localhost:5000>.
2. You should see form1 rendered on your browser.
3. Enter data for the text fields and select a image file to upload.
4. You should see "Hello Worlds" being displayed.
5. Modify the code in form\_submit so that the message "Hello, you have submitted a product <product> , <description> that costs <price>" is displayed on the web browser instead. <product>, <description> and <price> are the text that the user entered in the form.
6. Add code in form\_submit to save the data submitted in the form to a file named **"products.txt"**. Each time a form is submitted, a new line in the form <product>,<description>,<price> is appended to the file.
7. We will handle the image later.

NOTE:

When a form is posted to the web app, you will be able to access the data from the form using a dictionary object **request.form.** The keys are the values that you defined for the name attributes of the input element.

**Part 2 Saving the image file on the Web Server**

1. In Part 1, the web form only sends text. In order to send binary content (image file) you need to add the form attribute enctype with a value of "multipart/form-data".
2. The file data is no longer stored in the **request.form** dictionary object, but in the **request.files** dictionary object.
3. The following code snippet is used to check for file attachment and if presence, save it in the /static/ folder on the web server.

| img = request.files.get("image\_file") # image\_file is the name attribute of the input file element in the form  if img:  img\_file\_name= img.filename  img.save(f"static/images/{img\_file\_name}")  # add code to add the img\_file\_name as the 4th column in the file products.txt |
| --- |

1. The final contents of the products.txt should looked like this:

Baby Rockie,Outdoor Backpack,42.00,backpack.png

Zenon EOS6D,SLR Pro Camera,1392.00,camera.png

Zenva glasses,A classical design,342.00,glasses.jpg

Smart Band,4/7 heart rate monitoring,365.00,watch.png

1. The folder static/images should have the images files.