Install python and flask.

Then open terminal (command prompt in Windows). Notice the version of python.

$ python

>>> exit()

Upgrade pip

$ python -m pip install --upgrade pip

Install Flask

$ pip install flask

Check if installation is correct and python can import flask.

$ python

>>> import flask

>>> exit()

Start project

Create (by mkdir command) a project directory (e.g. Demo\_SSY) in a path (e.g. C:\\path) for project.

C:\\path\Demo\_SSY

Use IDE to open this folder. (e.g. Visual Studio Code: File > Open Folder > Select Folder). This is a completely empty project.

Create a new python file here: flaskssy.py

Writing below code inside flaskssy.py will create the “Hello World App”.

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

def hello():

return "Hello SSY Analytics!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

To run this, navigate to C:\\path\Demo\_SSY in terminal. Type

$ python flaskssy.py

OR

Set the environment variable FLASK\_APP to the file flaskssy.py

$ set FLASK\_APP=flaskssy.py

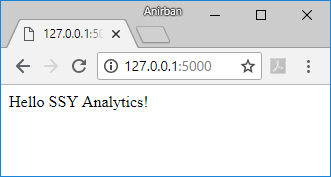
To run flask in debug mode that reflects code changes at every save in project files:

$ set FLASK\_DEBUG=1

Run flaskssy.py by following command:

$ flask run

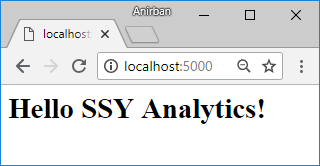
Open browser at web address: http://localhost:5000



Wrap the text in html tags and save:

return "<h1>Hello SSY Analytics!</h1>"

Refresh browser



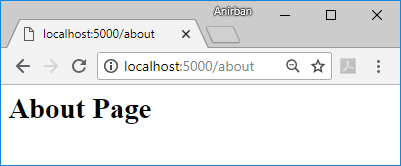
Add about route:

@app.route("/about")

def about():

return "<h1>About Page</h1>"

look in browser: address = <http://localhost:5000/about>



Following code in flaskssy.py does below:

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route("/")

@app.route("/home")

def home():

return "<h1>Home Page</h1>"

@app.route("/about")

def about():

return "<h1>About Page</h1>"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

1. Handles “/” and “/home” route decorators with same home() function.
2. Shows and about page
3. Runs app in debug mode without explicitly setting up environment variables FLASK\_APP and FLASK\_DEBUG. Just

$ python flaskssy.py

Working with Templates

Create a directory called templates inside project directory: C:\\path\Demo\_SSY\templates

Create 2 html files, home.html and about.html in this folder:

<!doctype html>

<html>

<head>

<title></title>

</head>

<body>

<h1>Home Page</h1>

</body>

</html>

<!doctype html>

<html>

<head>

<title></title>

</head>

<body>

<h1>About Page</h1>

</body>

</html>

In the flaskssy.py app file import render\_template and in home route use render\_template(‘home.html’) to return html, instead of returning inline html (in case of /about rout below.

from flask import Flask, render\_template

from forms import RegistrationForm, LoginForm

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = '8bd131fc7ef168cfff2c744fbf6fdeb7'

@app.route("/")

@app.route("/home")

def home():

return render\_template('home.html')

@app.route("/about")

def about():

return "<h1>About Page</h1>"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

Next return render\_template(‘about.html’) for about route as well.

Add dummy data to the application file flaskssy.pt that mock data retrieved from a database call. Next, pass this data to the template as the second parameter to the render\_template() function below.

from flask import Flask, render\_template

from forms import RegistrationForm, LoginForm

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = '8bd131fc7ef168cfff2c744fbf6fdeb7'

testimonials = [

{

'topic': 'Samajik Suraksha Yojana',

'description': 'Social security for unorganised workers',

'writeup': 'The West Bengal Labour Welfare Board is a statutory and autonomous body.',

'date': 'August 20, 2018'

},

{

'topic': 'Fluxionbits',

'description': 'Solutions that create impact',

'writeup': 'Custom Apps, Integration Services and Staffing Solutions',

'date': 'August 22, 2018'

}

]

@app.route("/")

@app.route("/home")

def home():

return render\_template('home.html', posts=testimonials)

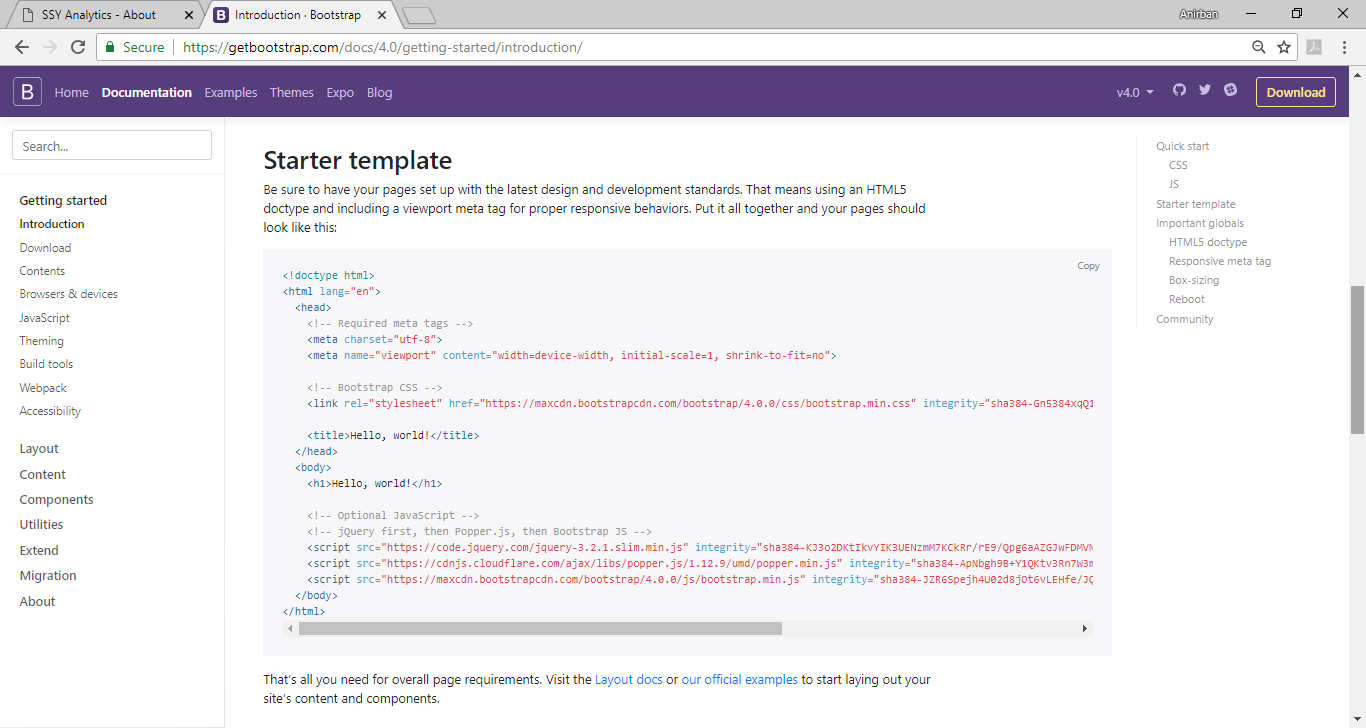
Create a template.html file in the templates directory for common code to be used by all html files.

In the template html file we can add code to publish the data passed for individual pages from the flaskssy.py application file. The resulting html files will look like as follows:

Using bootstarp

Get bootstrap starter template:Search bootstrap documentation starter template. Go to:

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>



Use this to change layout.html page.

Add following meta tags to the <head> tag of template.html, before <title>:

<!doctype html>

<html>

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

{% if title %}

<title>SSY Analytics - {{ title }}</title>

Copy the following javascript tags from bootstrap before closing body tags of template.html.

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

Put the body of the template page inside a div tag.

Refresh browser cache and reload browser (Ctrl+Shift+R) and see css formatting changes appear in home and about pages.

Working with Forms (WT Forms) in Flask

$ pip install flask-wtf

Create new file forms.py in project folder C:\\path\Demo\_SSY

Create 2 form classes RegistrationForm and LoginForm as below with fields and vlaidators:

from flask\_wtf import FlaskForm

from wtforms import StringField, PasswordField, SubmitField, BooleanField

from wtforms.validators import DataRequired, Length, Email

class RegistrationForm(FlaskForm):

username = StringField('Username',

validators=[DataRequired(), Length(min=2, max=20)])

email = StringField('Email', validators=[DataRequired(), Email()])

password = PasswordField('Password', validators=[DataRequired()])

submit = SubmitField('Sign Up')

class LoginForm(FlaskForm):

email = StringField('Email', validators=[DataRequired(), Email()])

password = PasswordField('Password', validators=[DataRequired()])

remember = BooleanField('Remember me')

submit = SubmitField('Login')

Set up SECRET\_KEY app config variable with 16 random hex characters in flaskssy.py app file:

Go to command prompt:

$ python

>>> import secrets

>>> secrets.token\_hex(16)

Paste the key generated at the top of flaskssy.py file

from flask import Flask

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = '8bd131fc7ef168cfff2c744fbf6fdeb7'

...

Import the forms created above into flaskssy.py app file:

Database

$ pip install flask-sqlalchemy