1. Create a simple view that returns "Hello, World!" and map it to a URL using Python Flask.

from flask import Flask

# Create Flask application

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

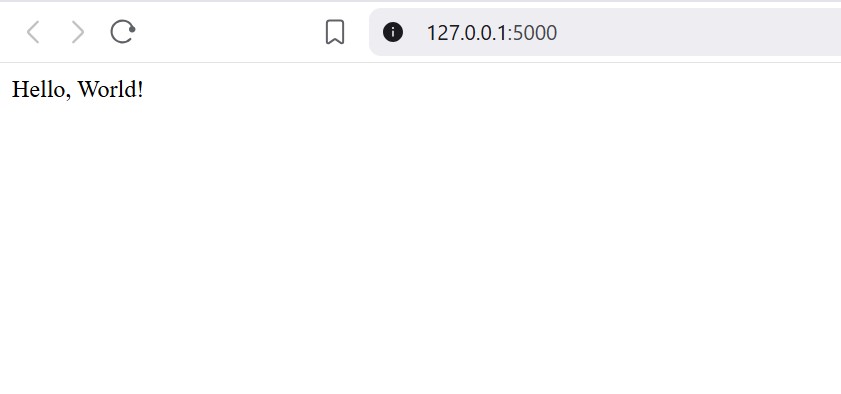
# Define a route and view function

@app.route('/') def hello():

return "Hello, World!"

# Run the application if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)



1. Create a Flask view that displays a list of hyperlinks to various social media websites using a Jinja template, and map it to a URL route.



2. app.py

from flask import Flask, render\_template

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

@app.route('/social') def social\_links():

# List of social media sites

links = {

"Facebook": "https://www.facebook.com",

"Twitter": "https://www.twitter.com",

"Instagram": "https://www.instagram.com",

"LinkedIn": "https://www.linkedin.com",

"YouTube": "https://www.youtube.com"

}

return render\_template("social.html", links=links)

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

app.run(debug=True)

3. templates/social.html

<!DOCTYPE html>

<html>

<head>

<title>Social Media Links</title>

</head>

<body>

<h2>Social Media Links</h2>

<ul>

{% for name, url in links.items() %}

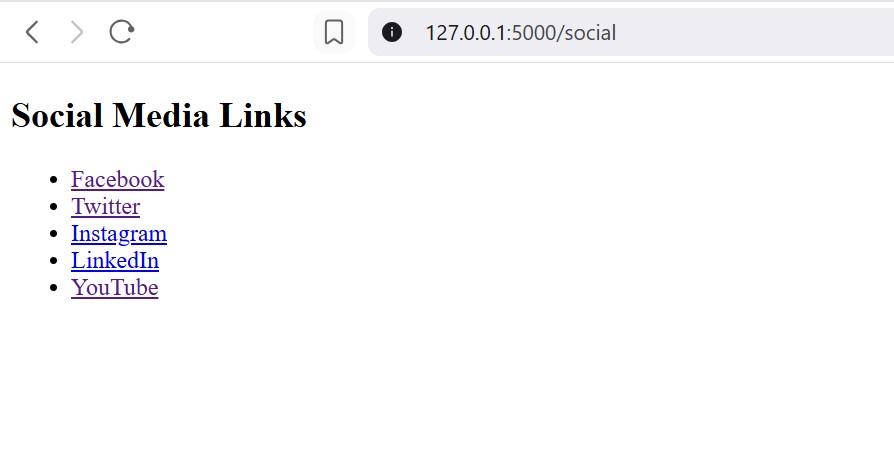
<li><a href="{{ url }}" target="\_blank">{{ name }}</a></li>

{% endfor %}

</ul>

</body>

</html>



3. Write a Flask application that:

1. Displays a message "Please add a number to the URL, like /5 or /10" when a user visits the home page ("/").
2. Accepts an integer from the URL (e.g., /10).
3. Generates and returns all prime numbers up to the given integer as a string.

Example:

* Visiting http://127.0.0.1:5000/10 should return:  
  2, 3, 5, 7,

from flask import Flask

# Create a Flask application instance

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

# Route for the home page

@app.route("/")

def home():

# Message asking user to enter a number in the URL

return "Please add a number to the URL, like /5 or /10"

# Route that accepts an integer from the URL

@app.route("/<int:number>")

def prime(number):

primes = "" # String to hold prime numbers

# Loop through all numbers from 2 to 'number'

for i in range(2, number + 1):

# Check if 'i' is prime

for n in range(2, (i // 2) + 1):

if i % n == 0: # If divisible, not a prime

break

else:

# If no divisor found, it is prime → add to result string

primes += str(i) + ", "

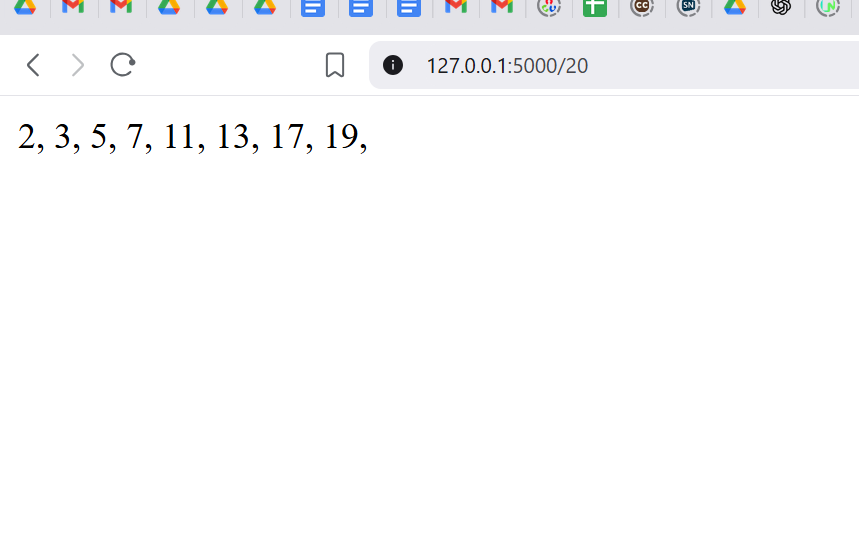
# Return all prime numbers as a string

return primes

# Run the Flask app

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

app.run(debug=True)



4. Create a Flask application that:

1. Displays a message "Please add a number to the URL, like /5 or /10" when a user visits the home page ("/").
2. Accepts an integer from the URL (e.g., /7).
3. Generates and returns the first N Fibonacci numbers, where N is the integer passed in the URL.

Example:

* Visiting http://127.0.0.1:5000/7 should return:  
  First 7 Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8,

from flask import Flask

# Create a Flask application instance

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

# Route for the home page

@app.route("/")

def home():

# Message asking user to enter a number in the URL

return "Please add a number to the URL, like /5 or /10"

# Route that accepts an integer from the URL

@app.route("/<int:number>")

def fibonacci(number):

# String to hold Fibonacci numbers

fibs = "First " + str(number) + " Fibonacci numbers: "

# Initialize first two Fibonacci numbers

fib1, fib2 = 0, 1

# Generate Fibonacci sequence

for i in range(number):

fibs += str(fib1) + ", "

fib1, fib2 = fib2, fib1 + fib2

# Return the Fibonacci sequence as a string

return fibs

# Run the Flask app

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

app.run(debug=True)

