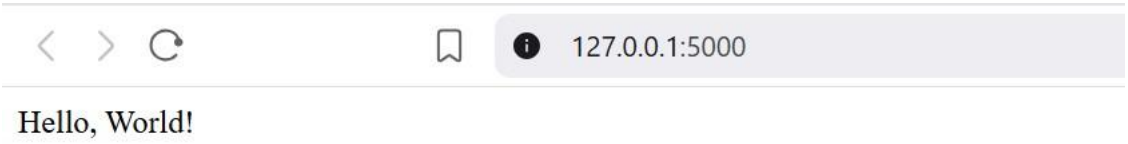
1. Create a simple view that returns "Hello, World!" and map it to a URL using Python Flask.

app.py

```
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)
```



Hello, World!

2. 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.

1. Project Structure

```
flask_app/
|— app.py
|— templates/
    └— social.html
```

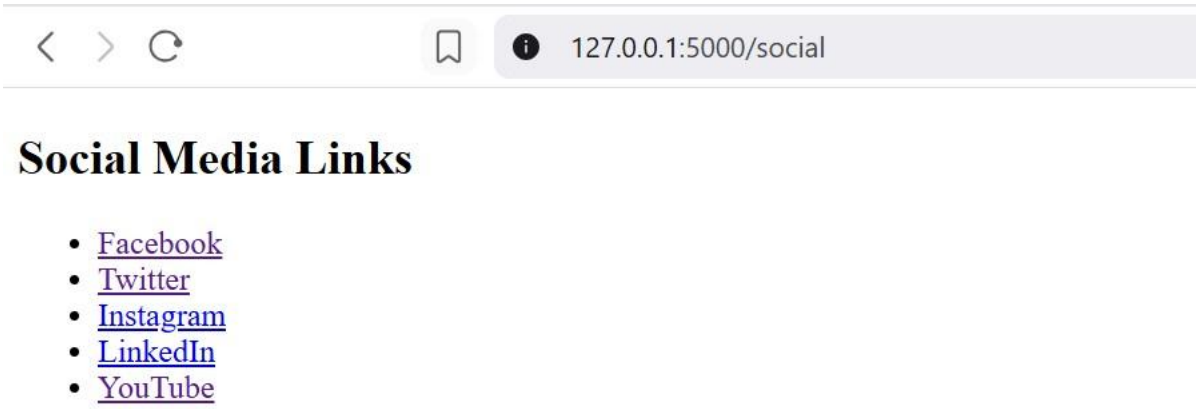
app.py

```
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)
```

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.

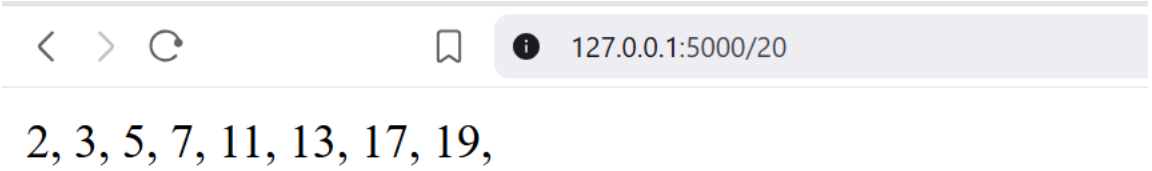
Example:

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

app.py

```
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:

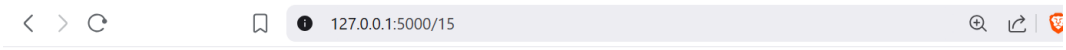
1. Displays a message `"Please add a number to the URL, like /5 or /10"` when a user visits the home page (`"/"`).
2. Accepts an integer from the URL (e.g., `/7`).
3. Generates and returns the first *N* Fibonacci numbers, where *N* is the integer passed in the URL.

Example:

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

app.py

```
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)
```



First 15 Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377,

5. Create a Flask application that:

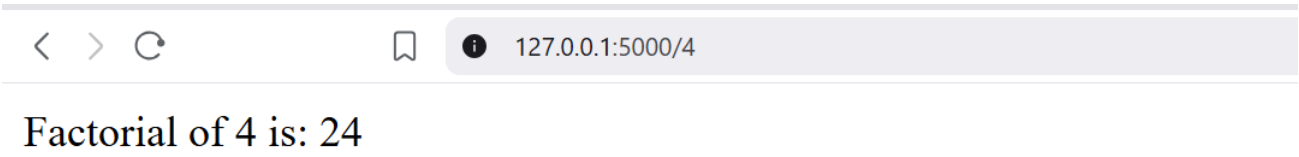
1. Displays a message **"Please add a number to the URL, like /5 or /10"** when a user visits the home page (/).
2. Accepts an integer from the URL (e.g., /6).
3. Calculates the **factorial** of the given number and displays the result in the browser.

Example:

- Visiting http://127.0.0.1:5000/6 should return:
  Factorial of 6 is: 720

app.py

```
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 factorial(number):
    # Calculate factorial
    fact = 1
    for i in range(1, number + 1):
        fact *= i

    # Return the factorial result as a string
    return f"Factorial of {number} is: {fact}"

# Run the Flask app
if __name__ == '__main__':
    app.run(debug=True)
```



6. Create a Flask application to navigate between multiple links in a webpage.

1.Program structure

```
flask_app/
│
├── app.py
│
└── templates/
    ├── index.html
    ├── about.html
    └── social.html
```

app.py

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/social')
```

```
def social():
    return render_template('social.html')

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

index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Home Page</title>
</head>
<body>
  <div >
     <nav>
      <a href="/">Home</a> |
      <a href="/about">About</a> |
      <a href="/social">Social</a>
     </nav>
    <h1>Hello...</h1>
    <p>This is the Home Page.</p>
  </div>
</body>
</html>
```

About.html

```
<!DOCTYPE html>
<html>
<head>
  <title>About Page</title>
</head>
<body>

    <nav>
      <a href="/">Home</a> |
      <a href="/about">About</a> |
      <a href="/social">Social</a>
     </nav>

    <h1>About Us</h1>
    <p>This is the About Page.</p>
</body>
</html>
```

Social.html

```
<!DOCTYPE html>
<html>
<head>
  <title>links</title>
</head>
<body>
  <div class="box">
    <nav>
      <a href="/">Home</a> |
      <a href="/about">About</a> |
      <a href="/social">Social</a>
     </nav>
    <h1>Our Logo:</h1>
    <img src="image_path.jpg " target="_blank" >
   </div>
</body>
</html>
```

7. Write a Flask app with a /contact page containing a form (Name, Message) and a /submit route that displays the submitted data using both **POST** and **GET** methods.

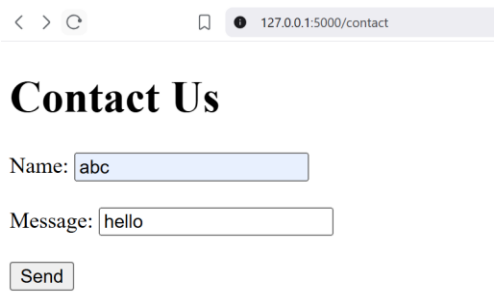
app.py

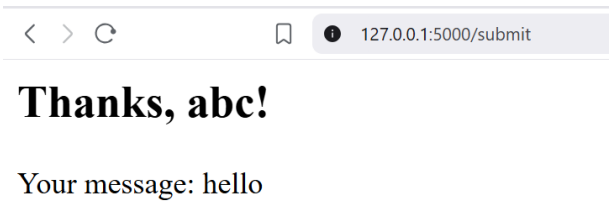
```
from flask import Flask, render_template,request

app = Flask(__name__)

@app.route('/contact')
def contact():
  return render_template('contacts.html')

# Route to handle form submission
@app.route('/submit', methods=['POST', 'GET'])
def submit():
  if request.method == 'POST':
    name = request.form['username']
    msg = request.form['message']
```

```
    else:
        name = request.args.get('username')
        msg = request.args.get('message')
    return f"<h2>Thanks, {name}!</h2><p>Your message: {msg}</p>"

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

contacts.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Page</title>
</head>
<body>
    <h1>Contact Us</h1>
  <!-- <form action="/submit" method="GET"> -->
    <form action="/submit" method="POST">
        <label>Name:</label>
        <input type="text" name="username"><br><br>

        <label>Message:</label>
        <input type="text" name="message"><br><br>

        <button type="submit">Send</button>
    </form>
</body>
</html>
```





8. Create a Flask application that connects to a MySQL database **flaskdb** containing a table **users(id, name, email)**.

1. Establish a connection to MySQL using **mysql.connector**.
2. Create a route **/users** that executes the SQL query SELECT * FROM users.
3. Return the result of the query in **JSON format** using jsonify.