## ****1. Research on Flask Basics****

### ✅ ****What is Flask?****

Flask is a lightweight **Python web framework** used for building web applications and APIs. It's known for being **simple, flexible, and easy to use**.

### ✅ ****How to Install and Run Flask****

1. Install Flask using pip:

pip install flask

1. Create a simple app.py file:

from flask import Flask

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

@app.route('/')

def home():

return "Hello, Flask!"

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

app.run(debug=True)

1. Run the Flask app:

python app.py

1. The app will run at http://127.0.0.1:5000/.

### ✅ ****How to Define API Routes and Return JSON Responses****

Use Flask’s @app.route() decorator to define API routes:

from flask import Flask, jsonify

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

@app.route('/api/greet')

def greet():

return jsonify({"message": "Welcome to my Flask API!"})

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

app.run(debug=True)

### ✅ ****What is Jinja2, and How is It Used in Flask?****

**Jinja2** is a **templating engine** in Flask that allows you to insert Python variables into **HTML templates**. It helps in dynamically rendering data on web pages.

Example:

1. **Template (**templates/index.html**)**:

<h1>Welcome, {{ name }}!</h1>

1. **Flask Route (**app.py**)**:

from flask import Flask, render\_template

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

@app.route('/welcome/<name>')

def welcome(name):

return render\_template('index.html', name=name)

## ****2. Build Your Flask API****

### ✅ ****Flask App Structure****

/flask-api-mini-project

│── app.py

│── /templates

│ ├── index.html

│── /static

│── README.md

### ✅ ****Implement the API Endpoints in**** app.py

from flask import Flask, jsonify, request, render\_template

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

# Sample data

items = [

{"id": 1, "name": "Laptop", "price": 750},

{"id": 2, "name": "Phone", "price": 500},

{"id": 3, "name": "Headphones", "price": 100}

]

# 1️⃣ Endpoint: Welcome message

@app.route('/', methods=['GET'])

def welcome():

return jsonify({"message": "Welcome to the Flask API!"})

# 2️⃣ Endpoint: Get list of items

@app.route('/items', methods=['GET'])

def get\_items():

return jsonify({"items": items})

# 3️⃣ Endpoint: Get item by ID

@app.route('/items/<int:item\_id>', methods=['GET'])

def get\_item(item\_id):

item = next((item for item in items if item['id'] == item\_id), None)

if item:

return jsonify(item)

return jsonify({"error": "Item not found"}), 404

# 4️⃣ Endpoint: Add a new item (POST request)

@app.route('/api/items', methods=['POST'])

def add\_item():

data = request.get\_json()

new\_item = {

"id": len(items) + 1,

"name": data['name'],

"price": data['price']

}

items.append(new\_item)

return jsonify(new\_item), 201

# 5️⃣ Render items on an HTML page using Jinja2

@app.route('/items/html')

def show\_items():

return render\_template('index.html', items=items)

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

app.run(debug=True)

### ✅ ****Jinja2 Template for Rendering Items (****templates/index.html****)****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Items List</title>

</head>

<body>

<h1>Available Items</h1>

<ul>

{% for item in items %}

<li>{{ item.name }} - ${{ item.price }}</li>

{% endfor %}

</ul>

</body>

</html>

## ****3. Testing Your API****

### ✅ ****Using Postman or Curl****

* **GET all items**

curl http://127.0.0.1:5000/api/items

* **GET specific item (ID = 1)**

curl http://127.0.0.1:5000/api/items/1

* **POST a new item**

curl -X POST http://127.0.0.1:5000/api/items -H "Content-Type: application/json" -d '{"name": "Tablet", "price": 300}'

## ****4. Submission Requirements****

### ✅ ****GitHub Upload****

1. **Create a GitHub repository** (e.g., flask-api-mini-project).
2. **Push your code**:

git init

git add .

git commit -m "Initial commit"

git branch -M main

git remote add origin https://github.com/yourusername/flask-api-mini-project.git

git push -u origin main

### ✅ ****README File****

Create a README.md file with:

# Flask API Mini-Project

## Setup Instructions

1. Install Flask:

pip install flask

2. Run the Flask app:

python app.py

3. API Endpoints:

- `GET /api/welcome` - Returns a welcome message.

- `GET /api/items` - Returns a list of items.

- `GET /api/items/<id>` - Returns a specific item by ID.

- `POST /api/items` - Adds a new item.

## Testing

Use \*\*Postman\*\* or \*\*Curl\*\* to test API responses.

## ✅ ****Final Deliverables****

✔ **Flask API with working endpoints**  
✔ **Jinja2 template rendering items**  
✔ **README file**  
✔ **Postman/Curl tests**  
✔ **GitHub repository with all files**