1. Create a Flask application with an /api route. When this route is accessed, it should return a JSON list. The data should be stored in a backend file, read from it, and sent as a response.

Solution:-

1. Create file

flask\_api\_app

app.py

data.json

2)Backend Data File (data.json)

{"id": 1, "name": "Anju", "role": "Developer"},

{"id": 2, "name": "sanu", "role": "Designer"},

{"id": 3, "name": "appi", "role": "Manager"}

3)Flask Application ([app.py](http://app.py))

from flask import Flask, jsonify

import json

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

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

def get\_data():

# Read data from backend file

with open("data.json", "r") as file:

data = json.load(file)

# Return JSON response

return jsonify(data)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

4)Run the Application:-

Run =python app.py

Output:-json

[

{"id": 1, "name": "Anju", "role": "Developer"},

{"id": 2, "name": "sanu", "role": "Designer"},

{"id": 3, "name": "appi", "role": "Manager"}

]

Explanation:-

1. **data.json** → stores backend data.
2. **json.load(file)** → reads the file content and converts it into a Python list/dictionary.
3. **jsonify(data)** → converts-Python object into a JSON response (so Flask sends proper JSON with headers).
4. **@app.route('/api')** → defines the /api endpoint.
5. **debug=True** → runs the server in development mode with hot reload.

2)Create a form on the frontend that, when submitted, inserts data into MongoDB Atlas. Upon successful submission, the user should be redirected to another page displaying the message **"Data submitted successfully"**. If there's an error during submission, display the error on the same page without redirection.

Solution:-

1)Install required libraries

pip install flask pymongo dnspython

2) Flask App ([app.py](http://app.py)):-

from flask import Flask, render\_template, request, redirect, url\_for

from pymongo import MongoClient

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

client = MongoClient("your\_mongodb\_atlas\_connection\_string")

db = client["test\_db"] # Database name

collection = db["users"] # Collection name

@app.route('/')

def index():

return render\_template('form.html')

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

def submit():

try:

name = request.form.get('name')

email = request.form.get('email')

collection.insert\_one({"name": name, "email": email})

return redirect(url\_for('success'))

except Exception as e:

return render\_template('form.html', error=str(e))

@app.route('/success')

def success():

return render\_template('success.html')

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

app.run(debug=True)

3)Frontend Form (templates/form.html)

<!DOCTYPE html>

<html>

<head>

<title>Submit Data</title>

</head>

<body>

<h2>Submit Your Data</h2>

<form method="POST" action="/submit">

<label>Name:</label>

<input type="text" name="name" required><br><br>

<label>Email:</label>

<input type="email" name="email" required><br><br>

<button type="submit">Submit</button>

</form>

{% if error %}

<p style="color:red;">Error: {{ error }}</p>

{% endif %}

</body>

</html>

4)Run the App:-

python app.py

Explanation:-

1. Frontend Form → User enters Name + Email.
2. /submit route → Reads form data using request.form.
3. MongoDB Insert → collection.insert\_one({...}) inserts data into MongoDB Atlas.
4. On success → Redirects to /success.
5. On error → Returns form.html again with an error message (without redirection).