Title Page

Title:
Flask API and MongoDB Integration

Submitted by:
Sai Kukkapalli

Course:
DevOps / Python / Full Stack

Institution:
TuteDude

1. Introduction

This project demonstrates a simple full-stack web application using Python Flask and MongoDB. It includes:

- A backend API that reads data from a file and returns JSON.
- A frontend form that submits data to a database.
- A success page on successful submission.
- Error handling without redirect if submission fails.

2. Objective

The main objectives of this project are:

- To create a Flask API endpoint that returns JSON data from a backend file.
- To design a frontend form that submits data to MongoDB.
- To display a success message after successful submission.

- To handle errors gracefully without redirecting the user.

---

3. Technologies Used

- Python 3
- Flask (Backend Framework)
- MongoDB (Using MongoDB Compass – Local Database)
- PyMongo (MongoDB Python Driver)
- HTML (Frontend)
- VS Code (Editor)

---

4. Project Folder Structure

```
flask-api-project/
│
├── app.py
├── data.json
├── requirements.txt
├── templates/
│   ├── index.html
│   └── success.html
│
```

(Insert Screenshot of Folder Structure Here)

---

5. Explanation of Each Component

5.1 data.json

This file stores backend data which is read by the Flask API.

Example content:

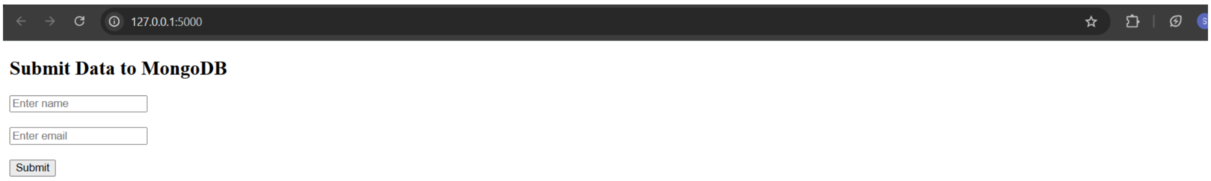
```
[
  { "id": 1, "name": "Sai" },
  { "id": 2, "name": "Rahul" },
  { "id": 3, "name": "Anu" }
]
```

5.2 Flask Backend (app.py)

The Flask application performs:

- Connects to local MongoDB using PyMongo.
- Exposes `/api` route to return JSON data from `data.json`.
- Shows a form at `/`.
- Inserts submitted data into MongoDB.
- Shows success page if insertion is successful.
- Shows error on same page if insertion fails.

Outputs:

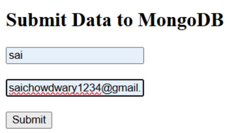


127.0.0.1:5000

Submit Data to MongoDB

Enter name

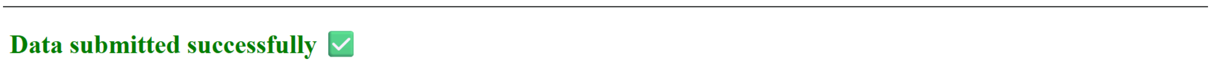
Enter email



Submit Data to MongoDB

sai

saichowdwary1234@gmail.



Data submitted successfully ✅

Conclusion

In this project, a Flask-based backend API and a MongoDB-integrated frontend form were successfully developed. The application demonstrates file handling, API creation, database insertion, and frontend-backend integration.