Flask and MongoDB Integration Project

Name: Devijahnavi

Course/Department: DevOps

# 1. Introduction

This project demonstrates a Flask web application integrated with MongoDB. It includes an API endpoint that returns JSON data from a file and a frontend form that inserts data into the MongoDB database. Upon successful data submission, the user is redirected to a success page.

# 2. Prerequisites

Before running the project, ensure the following are installed:  
- Python 3.10 or higher  
- Flask library  
- pymongo library  
- A local or cloud MongoDB instance  
- Any text editor (VS Code, PyCharm, etc.)

# 3. Steps to Execute the Project

Step 1: Create a Flask application with an /api route that returns JSON data from a backend file.  
Step 2: Create a form in HTML that collects user data (name, email) and sends it to Flask.  
Step 3: Use Flask to insert submitted data into MongoDB.  
Step 4: Redirect to a success page if the operation is successful; otherwise display an error.  
Step 5: Run the Flask app and access it via http://127.0.0.1:5000

# 4. Code Summary

app.py contains the Flask backend with routes for displaying the form, handling form submissions, and serving JSON data from data.json.  
Templates folder contains form.html and success.html for the frontend design.

# 5. Output Screenshots

Below are screenshots showing the execution and outputs of the project.

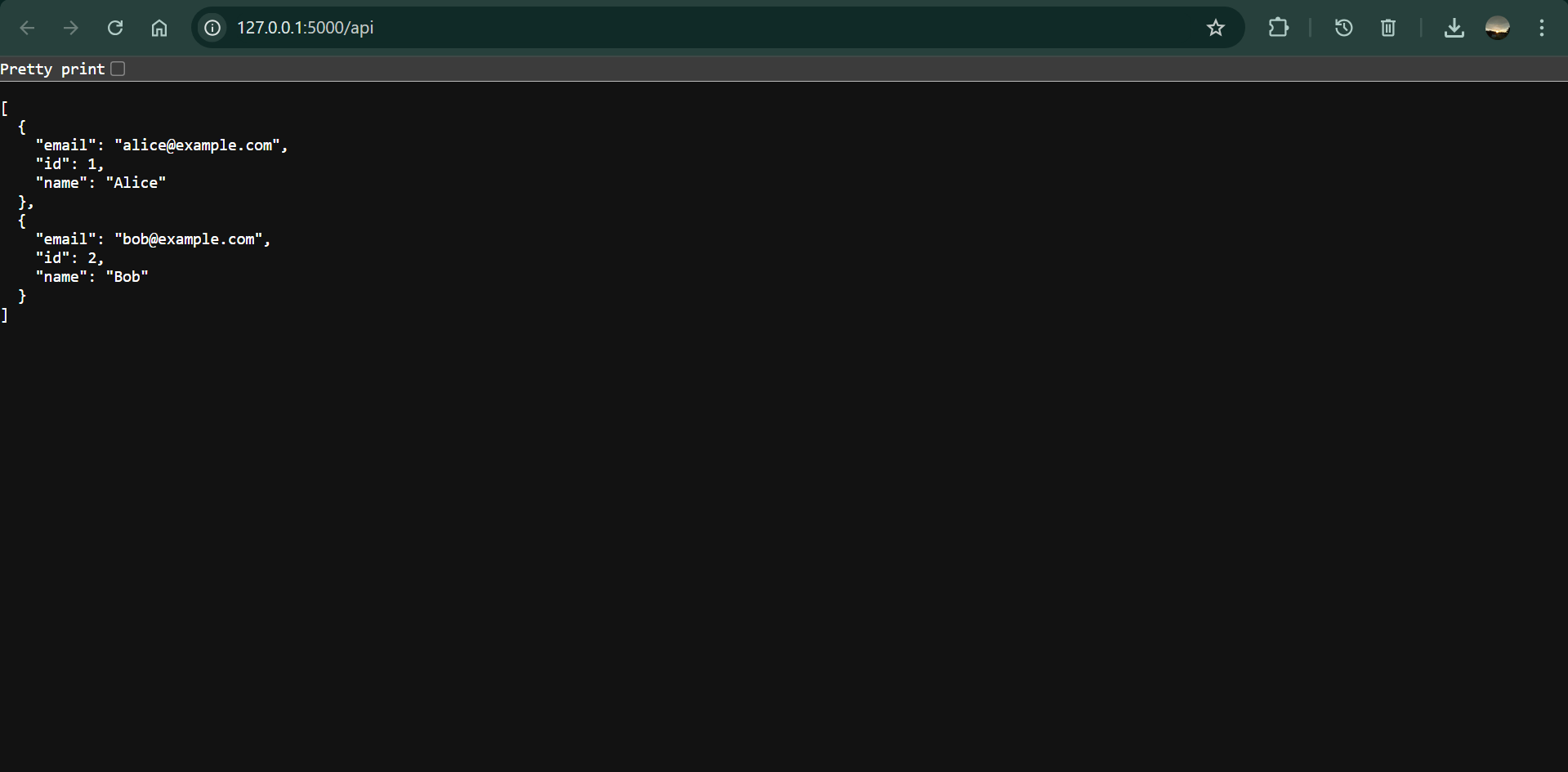


Figure 1: /api route displaying JSON data.

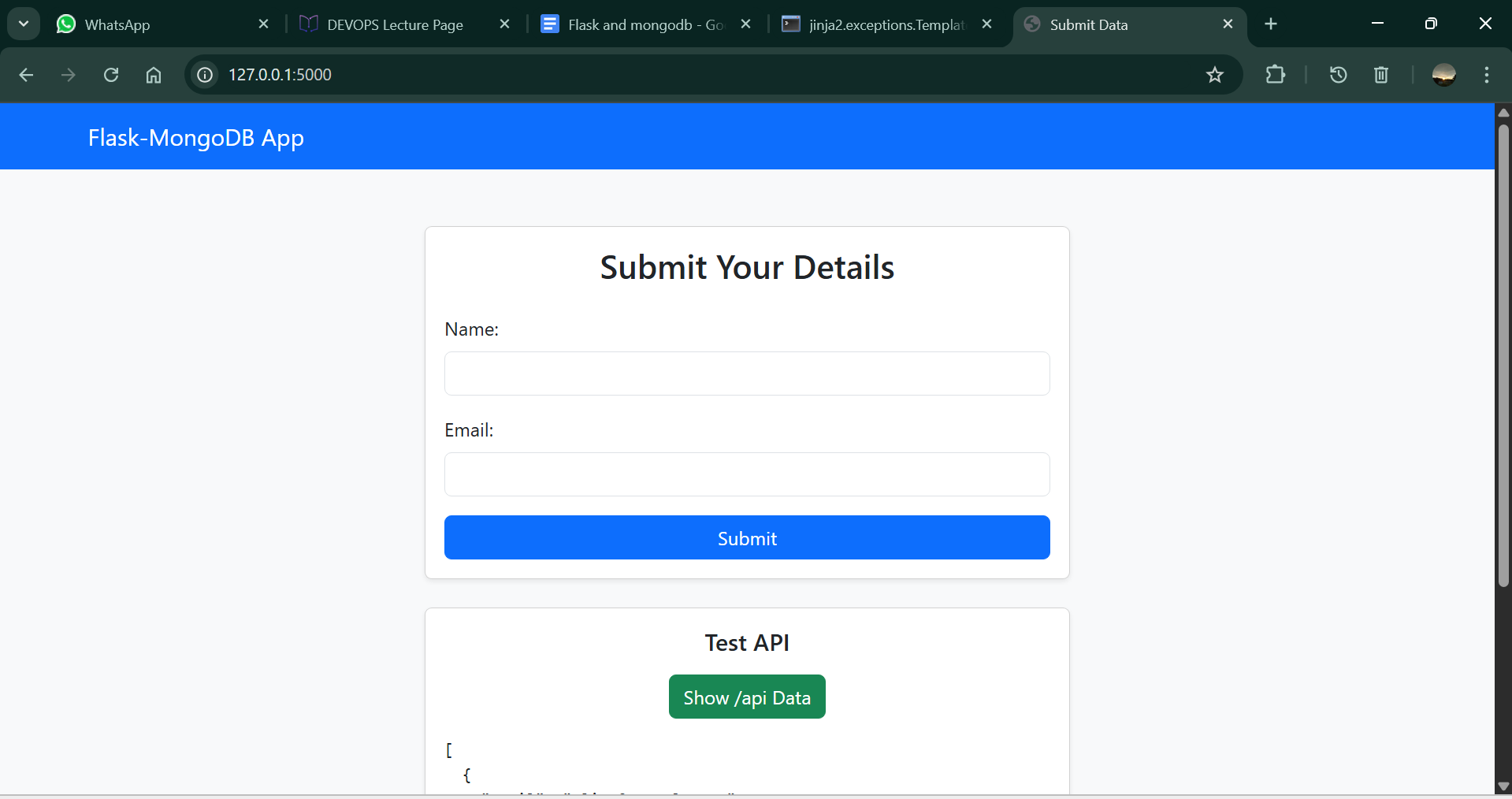


Figure 2: Web form with navigation bar and API test button.

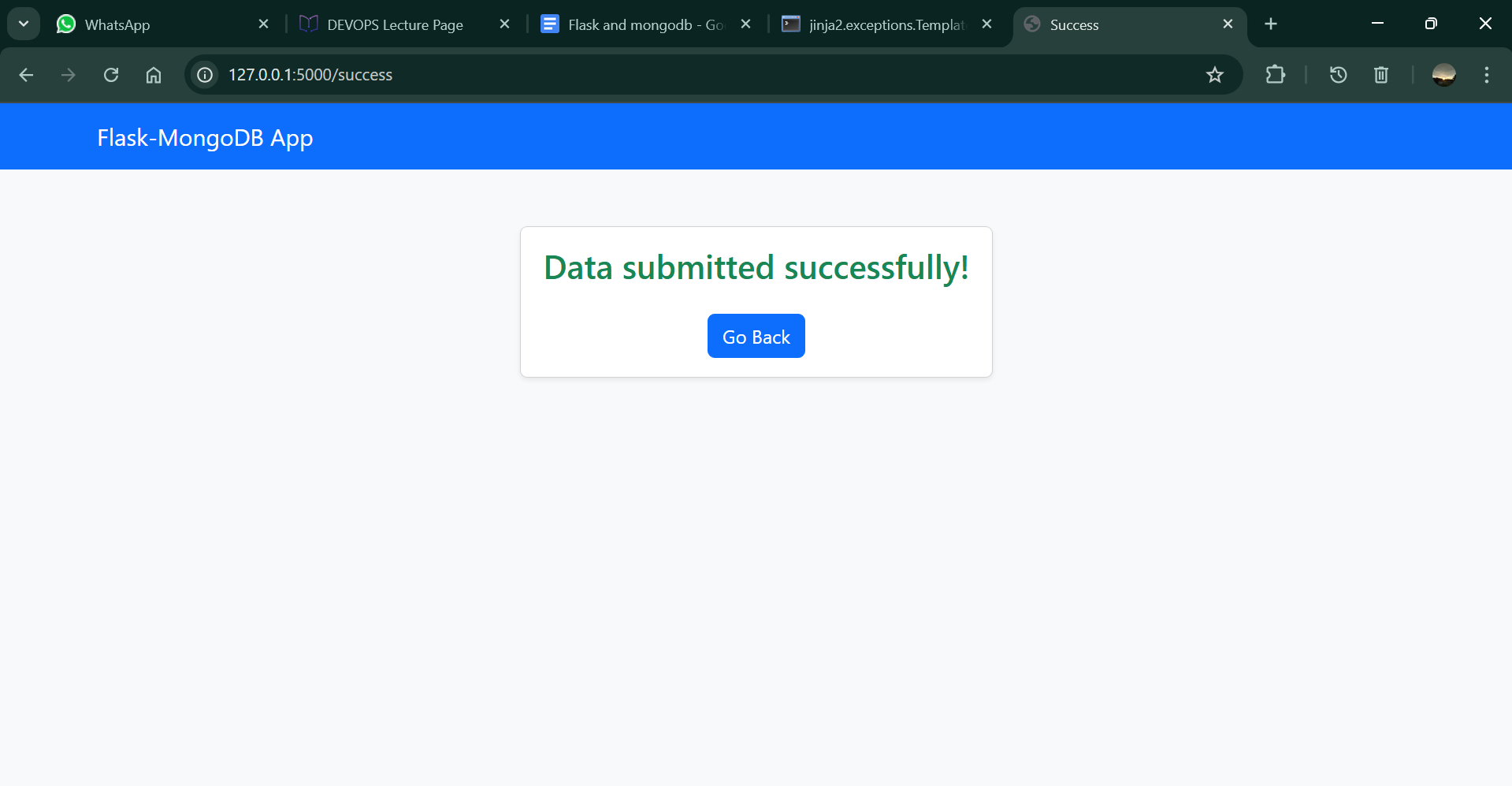


Figure 3: Success page after data submission.

# 6. Conclusion

This project successfully demonstrates how to integrate Flask with MongoDB, handle API responses, and build a simple frontend for data submission. The setup is lightweight and ideal for beginners learning Flask backend development with a database connection.