

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.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a project named "FLASK\_MONGO\_PROJECT" containing files ".vscode", "app.py", and "data.json".
- Code Editor:** Displays the "app.py" file with Python code for a Flask application. It includes routes for "/submit" and "/success", and a main block to run the app.
- Terminal:** Shows the command "python app.py" being run, and the resulting output indicating the app is running on port 5000 and accepting connections from 127.0.0.1.
- Status Bar:** Shows the Python version (3.12.10), indexing status ("Indexing completed..."), and file encoding (UTF-8).

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.

The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL "127.0.0.1:5000".
- Page Title:** "Student Form".
- Form Fields:** Two input fields labeled "Name:" and "Course:", and a "Submit" button.
- Page Content:** A success message "Data submitted successfully" is displayed below the form.