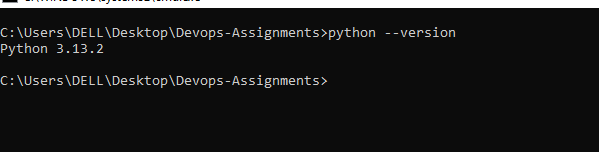
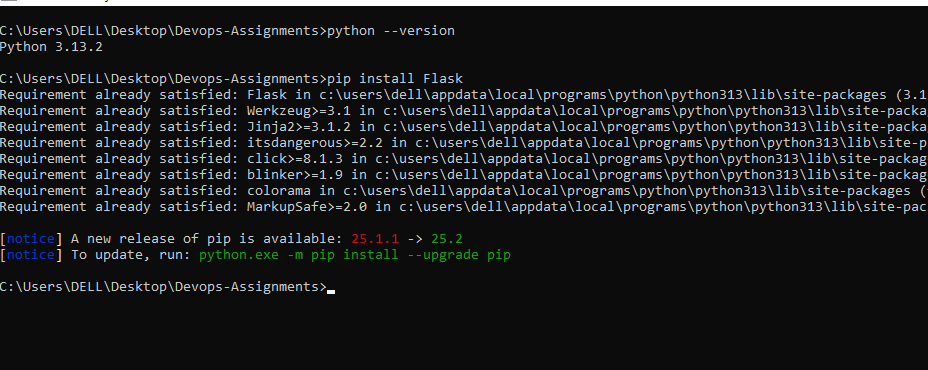
1. Create a Flask application with an /api route. When this route is accessed, it should return a JSON list. The data should be stored in a backend file, read from it, and sent as a response.

To create a flask application we need to first have python installed. Once that is done we next need to install flask using pip. I have python already installed in my machine and I verified it using the below command



Once that is done we can install flask franework using pip or pipx which package manager is installed on ur machine. I have pip installed on my machine so I installed flask using the same.



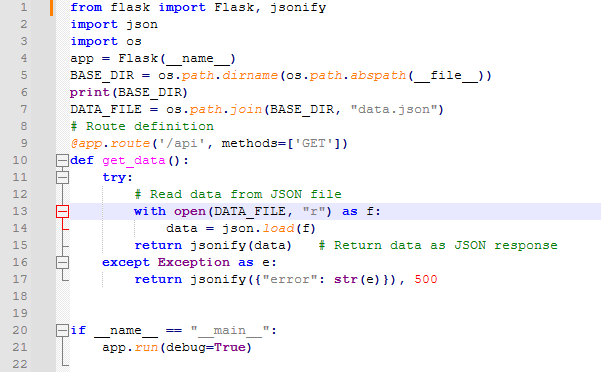
Once flask is also installed now lets focus on creating a small flask application. For that 1st lets start by creating a file called app.py which acts a entry point to ur flask application and also does the following tasks:

 **Creates the Flask application object**

 **Defines routes** (the URLs your app responds to)

 **Runs the application.**

**Enough with the background now lets focus on the task of creating a /api route. Open the app.py file and write the code shown below**

****

**What it does??**

**Lines 1-3 : we are importing some of the python’s standard libraries that are required for performing the tasks.**

app = Flask(\_\_name\_\_) → Initializes your Flask app.

**Lines 5-7**

* **\_\_file\_\_ → is a special variable in Python that gives you the current script’s filename (including its path).**
* **os.path.abspath(\_\_file\_\_) → converts that into an absolute path (so instead of ./app.py, you get something like C:/Users/Pisher/project/app.py)**
* **The final output of line 5 will be C:/Users/Pisher/project/app.py from here we are extract till /project and line 7 appends my json file name to the base directory**

**Line 9: we define a route called /api . methods=[“get”] means this route only accepts get method. The get\_data() executes when the /api route is accessed. So the actual functionality of the route is defined in the get\_data function**

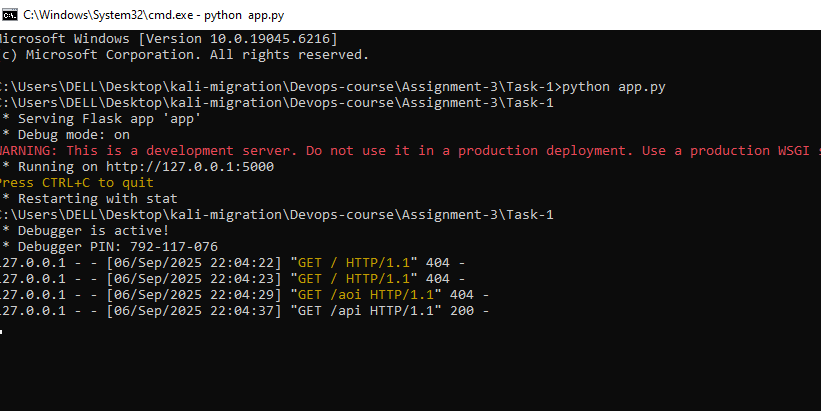
**The get\_data function does the following task**

**1 opens the file data.json in read mode and parses the contents of the file and**

**2 display the content of the file in json format when the /api route is accessed.**

**To run this application, open up a terminal and run python app.py and then goto browser and access the below url**

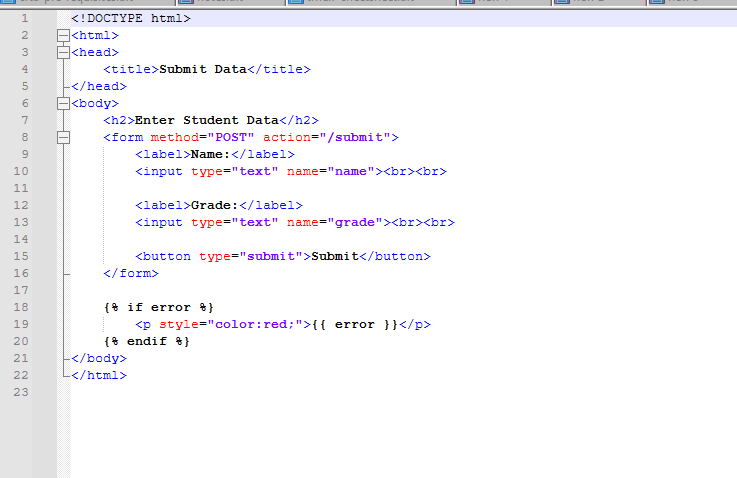
<localhost:5000/api>



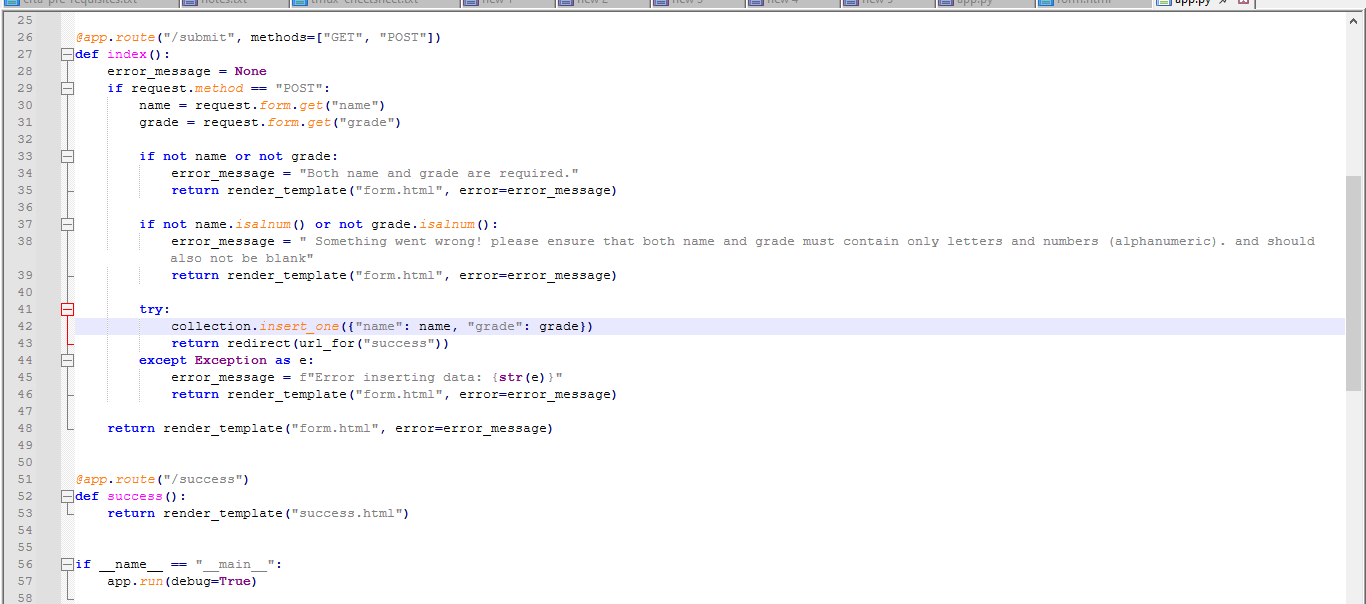


1. Create a form on the frontend that, when submitted, inserts data into MongoDB Atlas. Upon successful submission, the user should be redirected to another page displaying the message **"Data submitted successfully"**. If there's an error during submission, display the error on the same page without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

First let’s create the front-end form to submit data to the backend.



The above page creates a form with 2 input fields for student name and grade which when submitted goes to the /submit endpoint. Now lets create the backend file and define 2 routes /submit and /success.



The /submit route is accessible via both get and post methods.

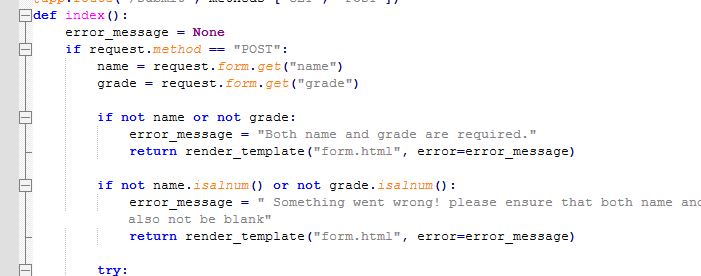
The index method first checks recieves data from the front end via

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

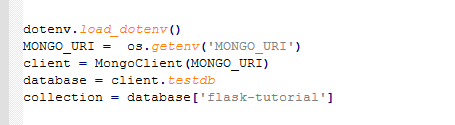
grade = request.form.get("grade")

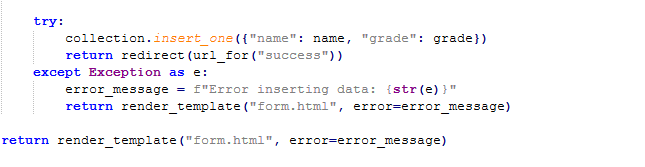
And then does some basic validation to check if

* Any of the input fields is empty or
* Any of the fields contains any special chars other than alptha numeric chars. If any of these conditions are satisfied the application throws an error page with link to get back to the previous page

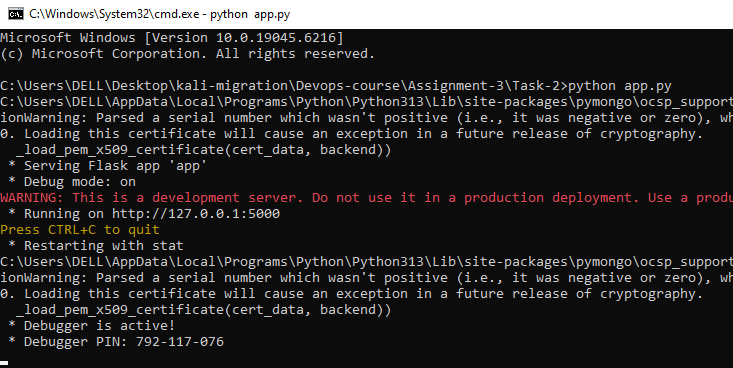


If no error occurs, it connects to the mongo cloud DB and then creates a collection called testdb and inside collection it creates a database called flask-tutorial and the inserts the form data. Once the insertion is completed the application redirects us to a success page. One thing to note is the connection string is not hardcoded in the code here for security reasons its defined in the environment file and accessed programmatically

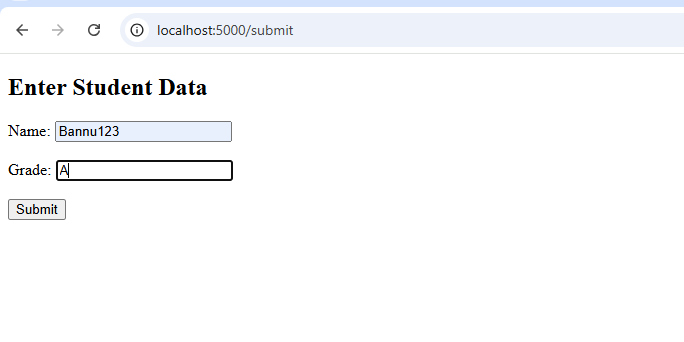


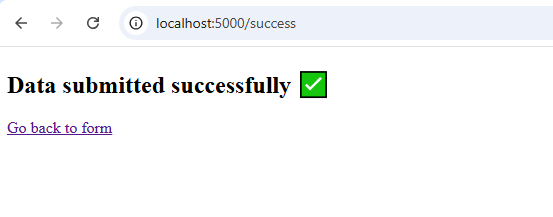


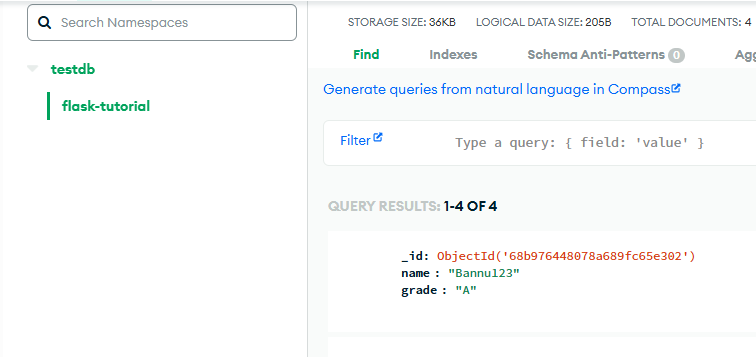
To run this application just open a terminal and run python app;.py



Enter any data and submit the form once its successfully submitted the data is entered in the DB.







**Submission Guidelines -:** Attach Screenshots or command along with explanation and submit in doc(google doc or microsoft doc) format also attach github repo link