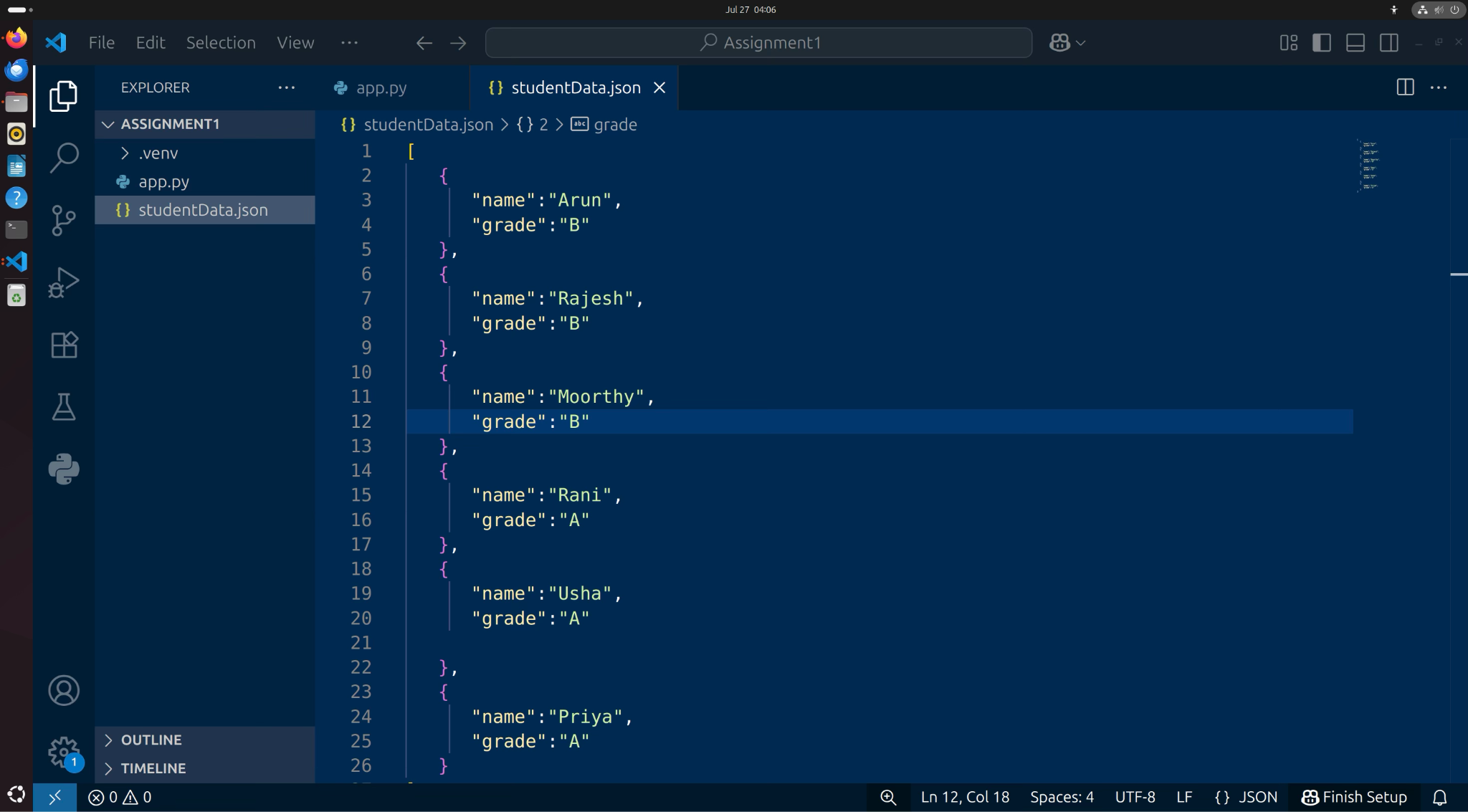
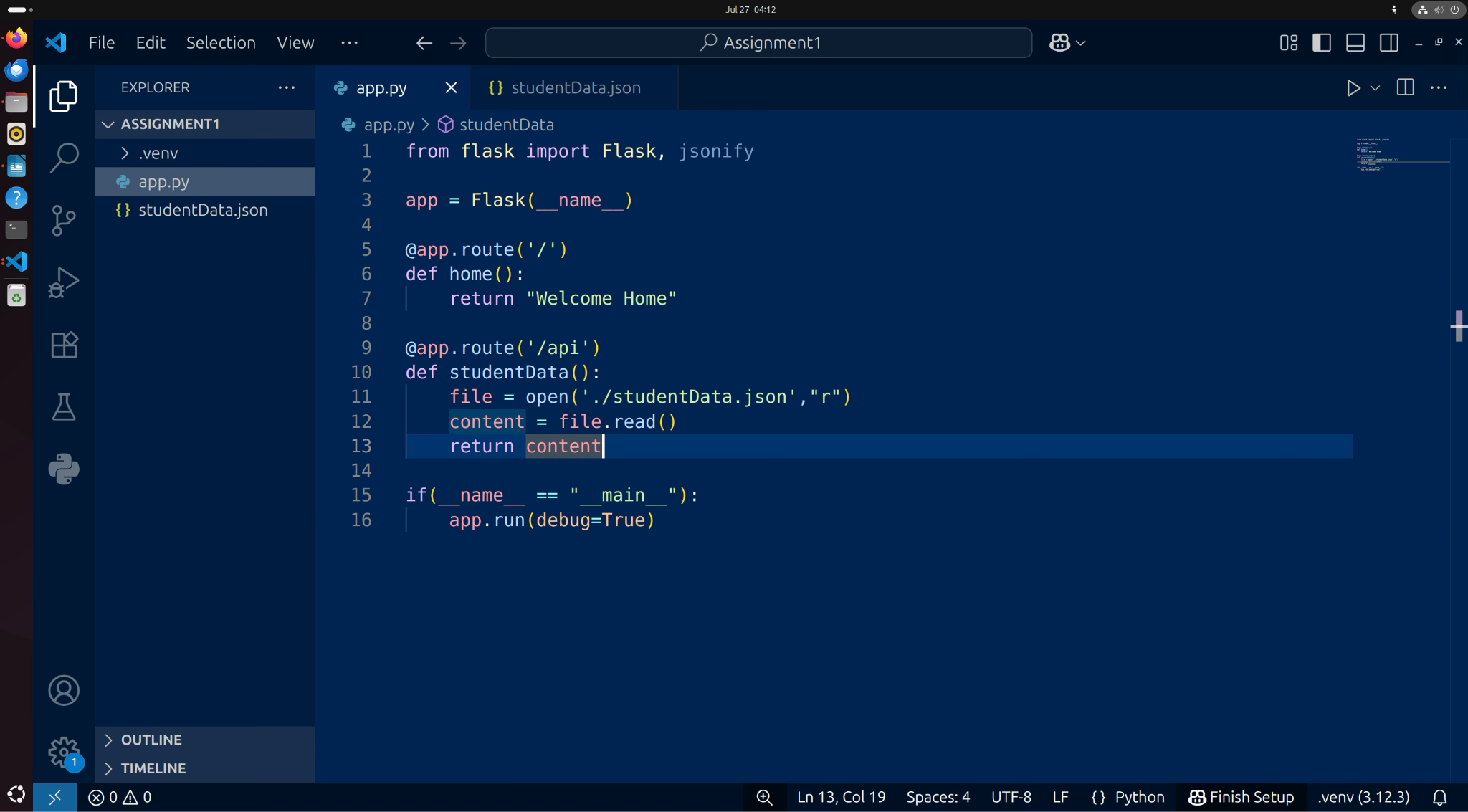
1. Create a Flask application with an /api route. When this route is accessed, it should return a JSON list. The data should be stored in a backend file, read from it, and sent as a response.

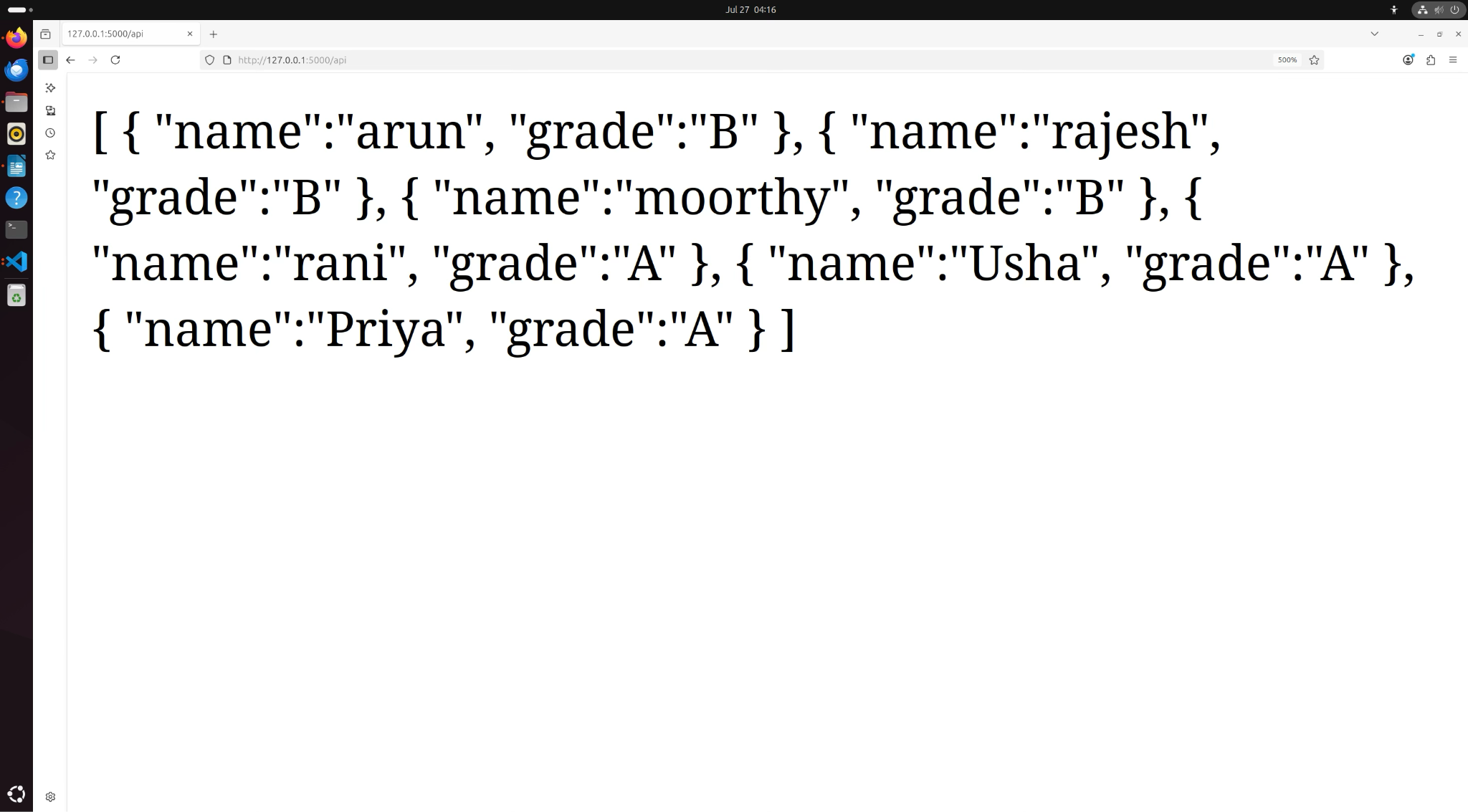
Github: <https://github.com/noorsadiya60/Assignments/tree/main/FlaskAssignment/FlaskAppToReadJSONData>



1. studenData.json file contains students name and their grade.



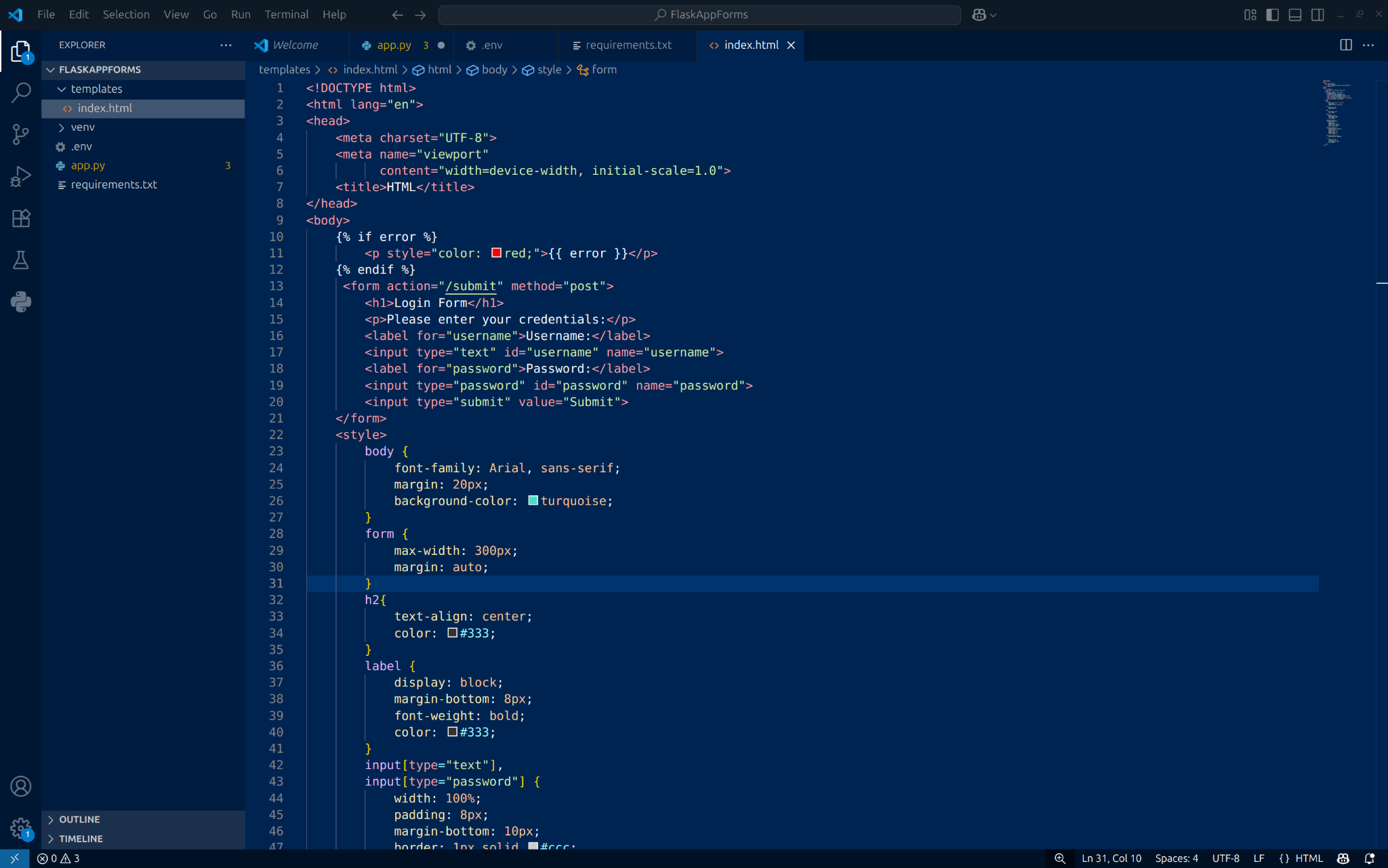
2. app.py contains Flask application for reading json file and display the content



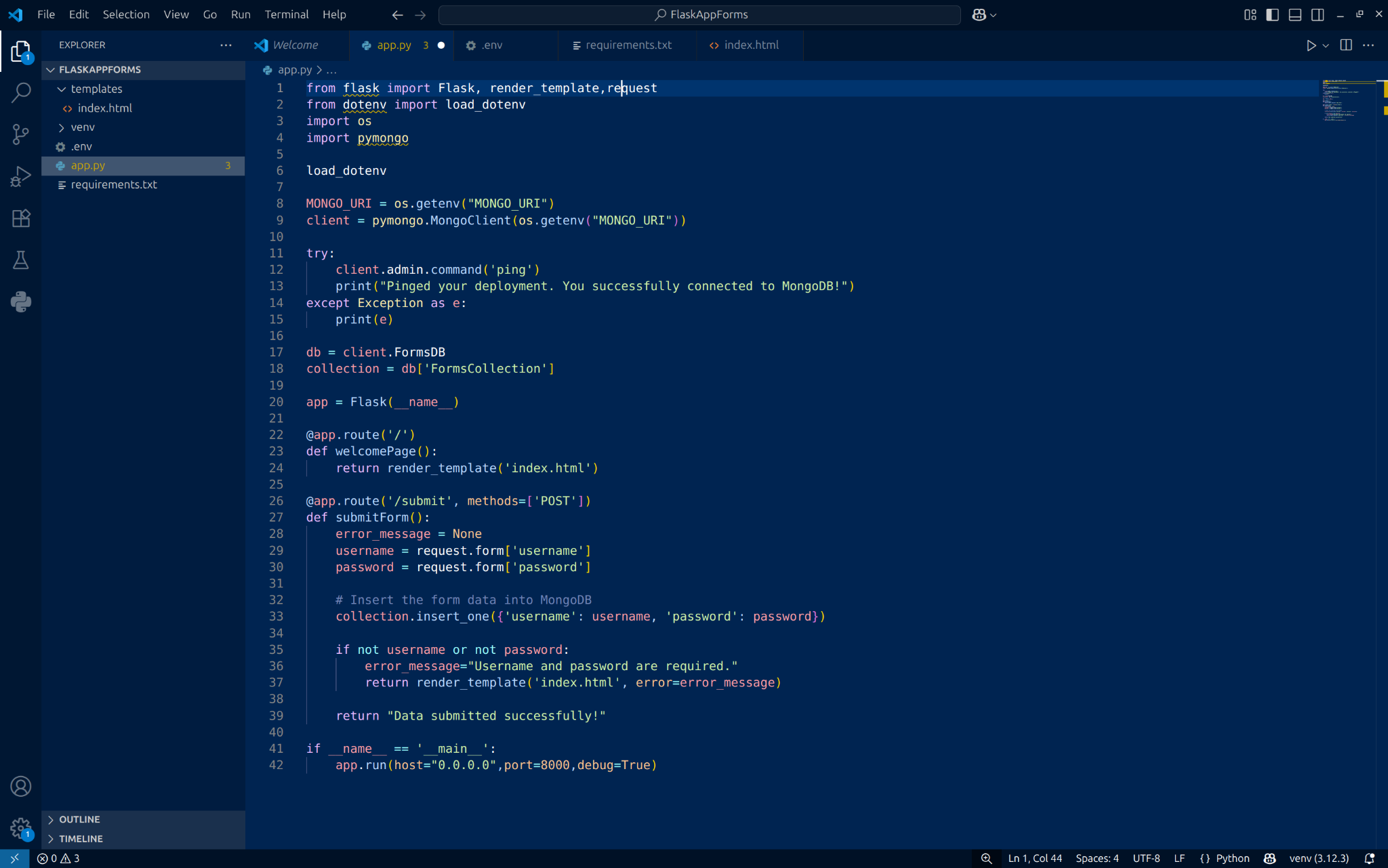
3. output

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.

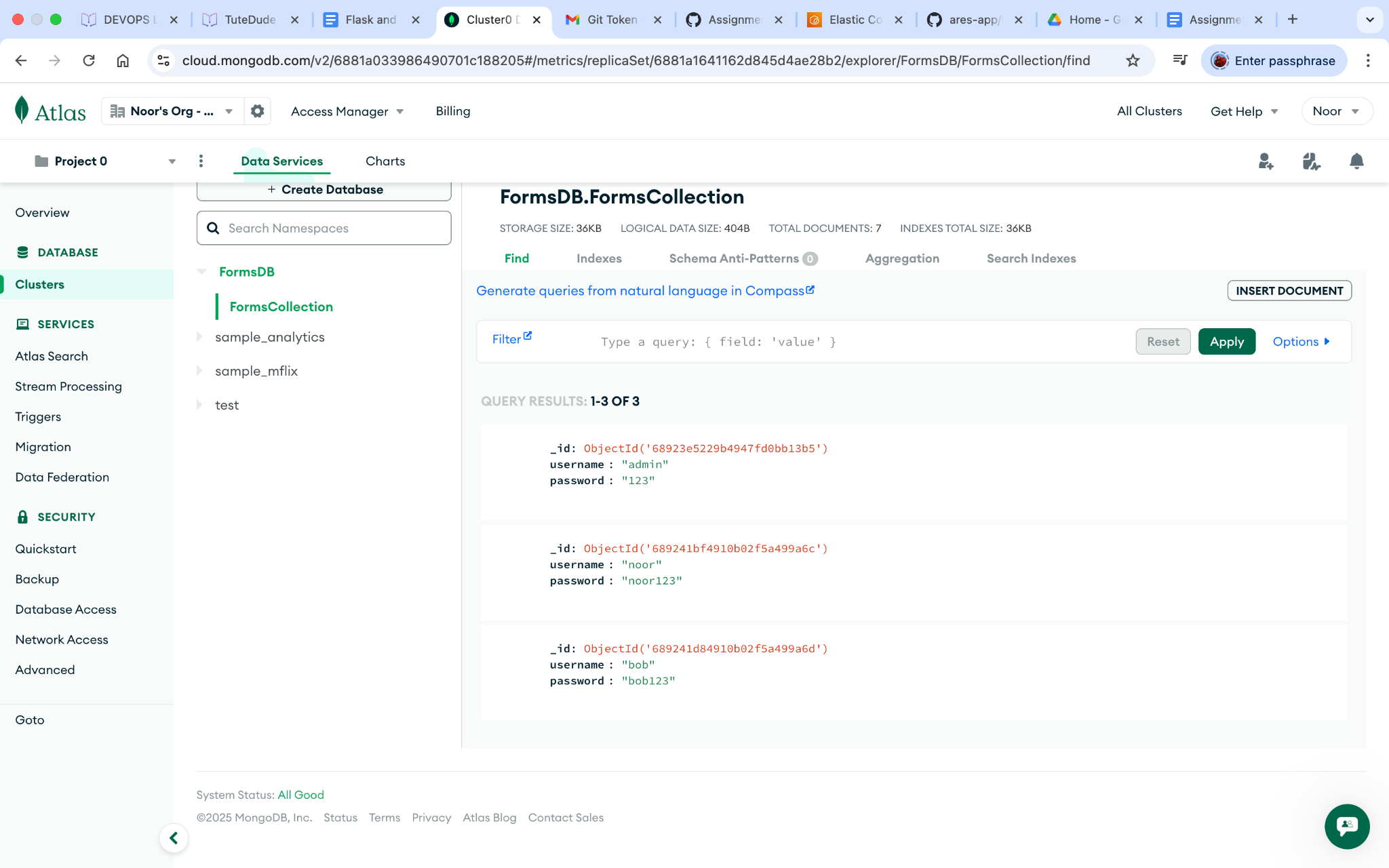
Github: <https://github.com/noorsadiya60/Assignments/tree/main/FlaskAssignment/FlaskAppForms>



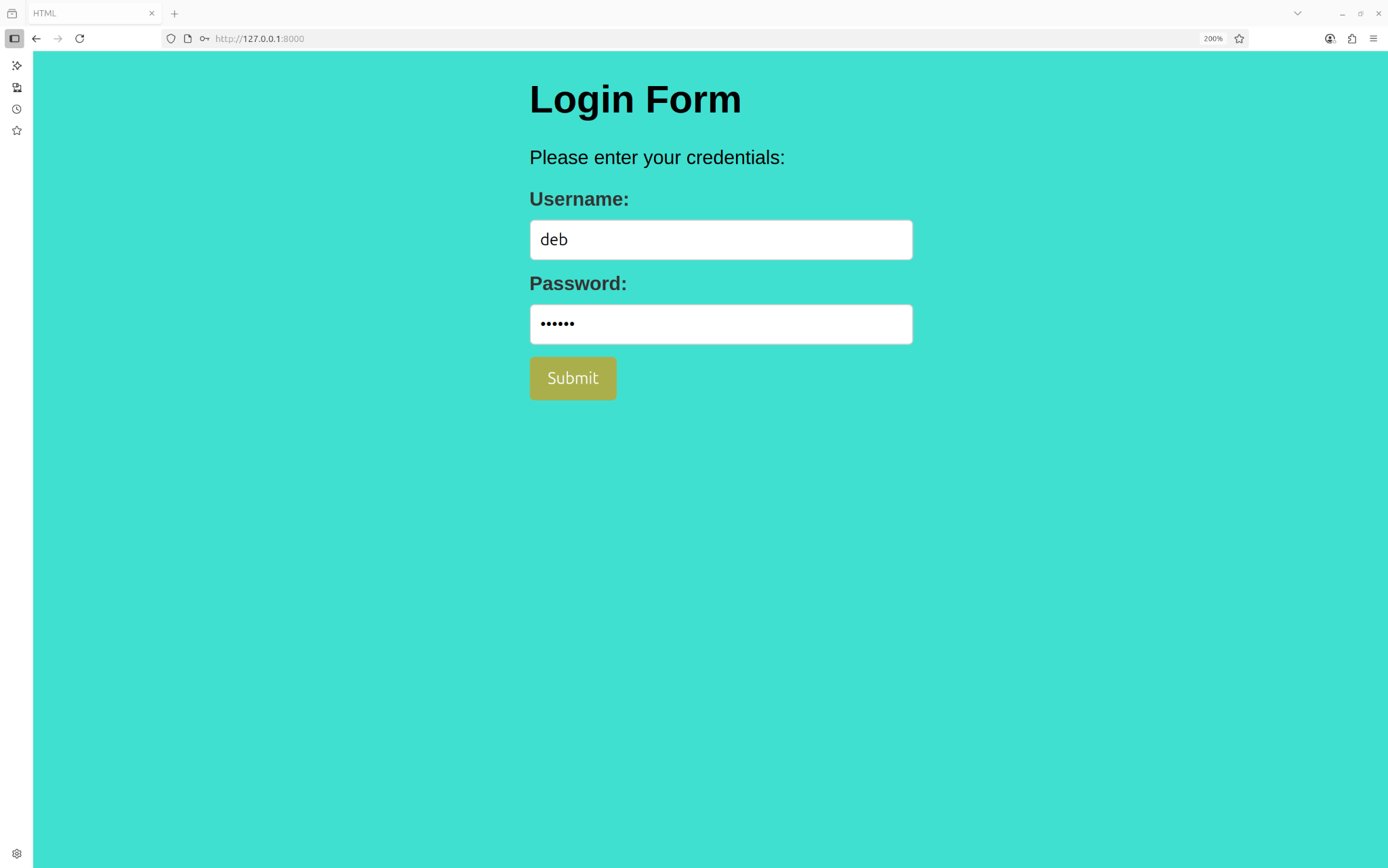
1. Index.html



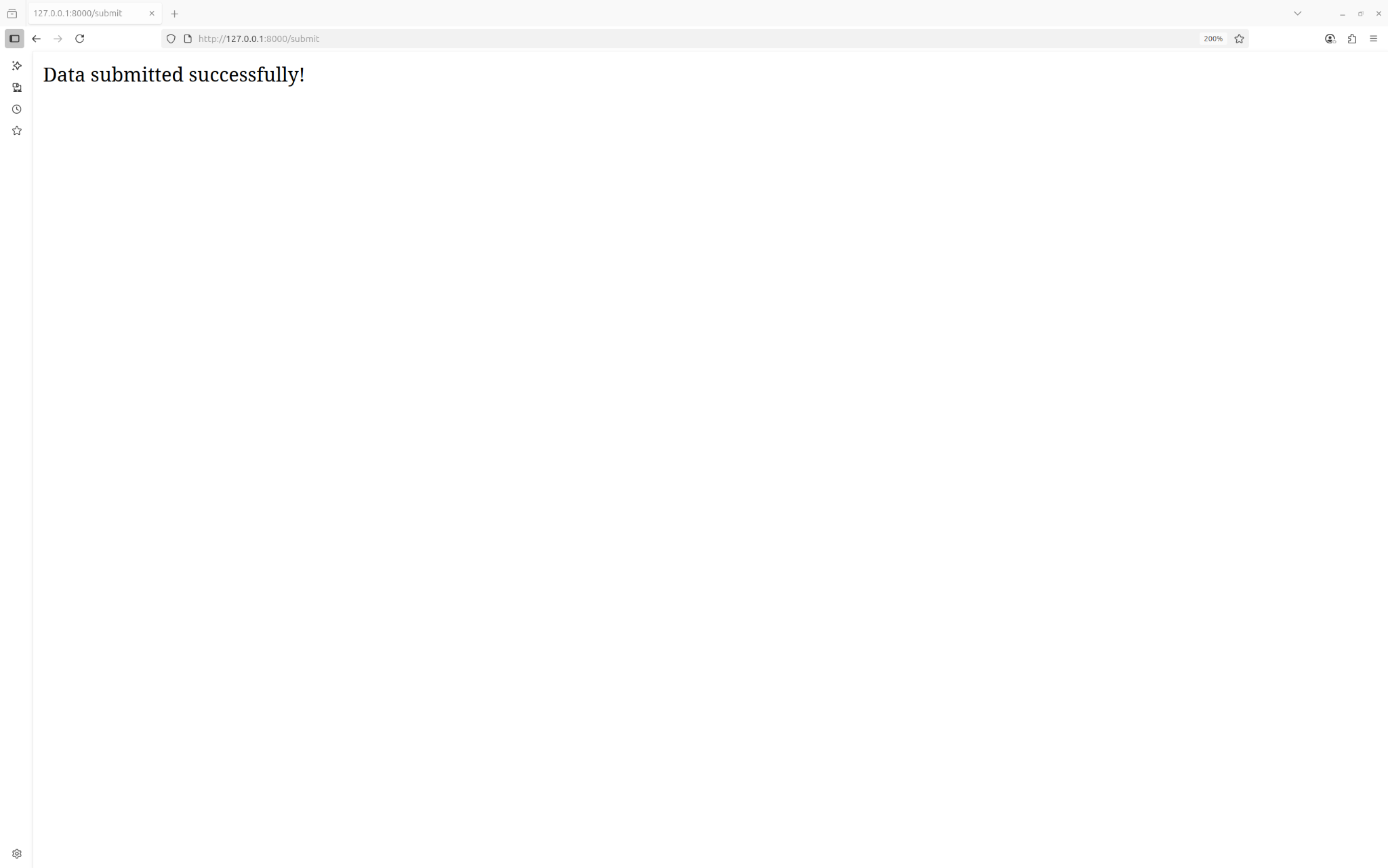
1. [app.py](http://app.py) - imported modules for Flask,load\_dotenv,request,render\_template,pymongo. Connected to MongoDB by storing mongodb uri in the .env file. Created FormsDB database and FormsCollection collection. Code in [app.py](http://app.py) will render index.html which has a form that accepts username and password. Upon clicking the submit button it insert data into the mongoDB database. If any field is missing it returns the error(This is shown in the output images).



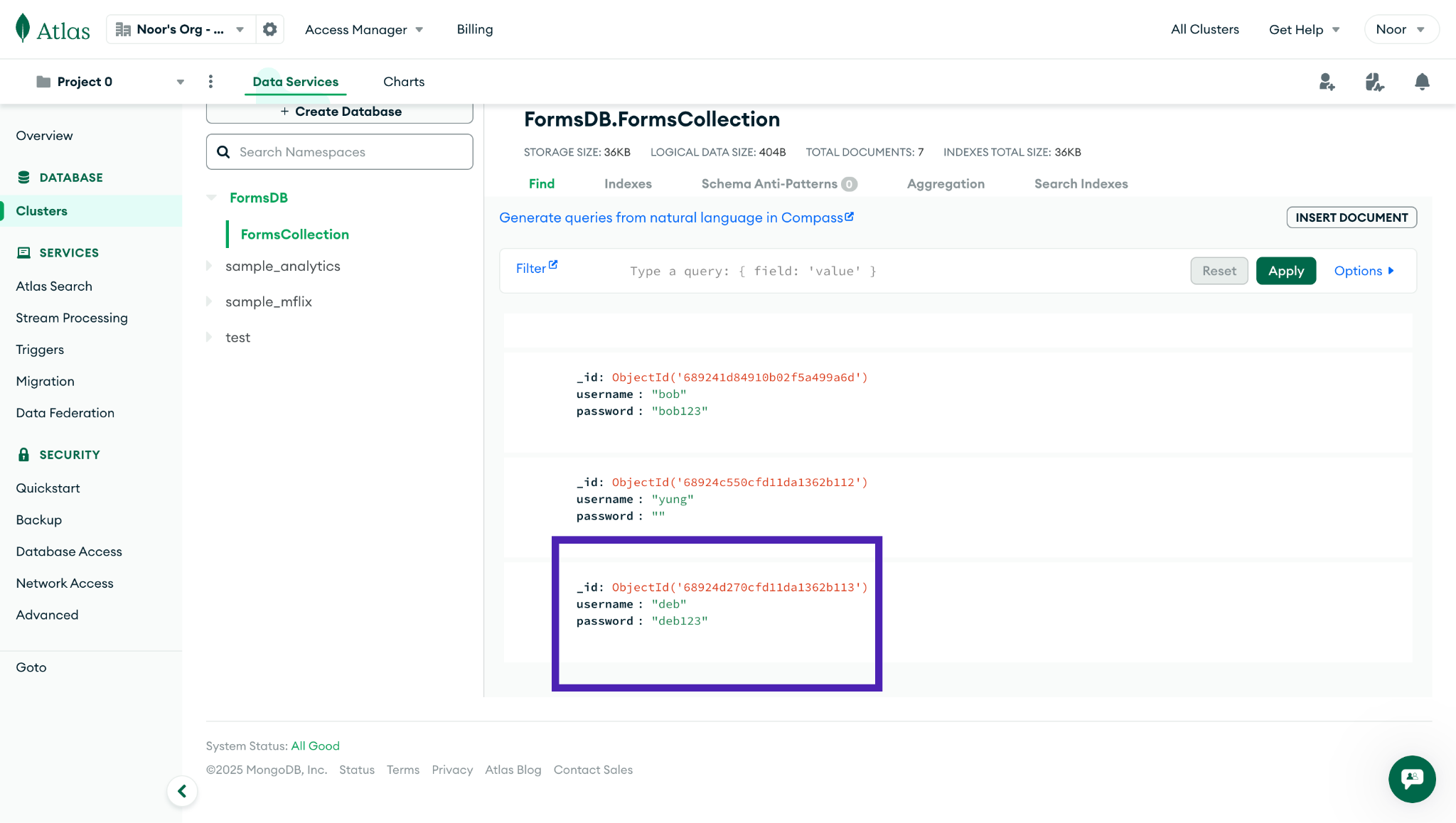
3 . MongoDB - FomsDB database and FormsCollection is successfully created.



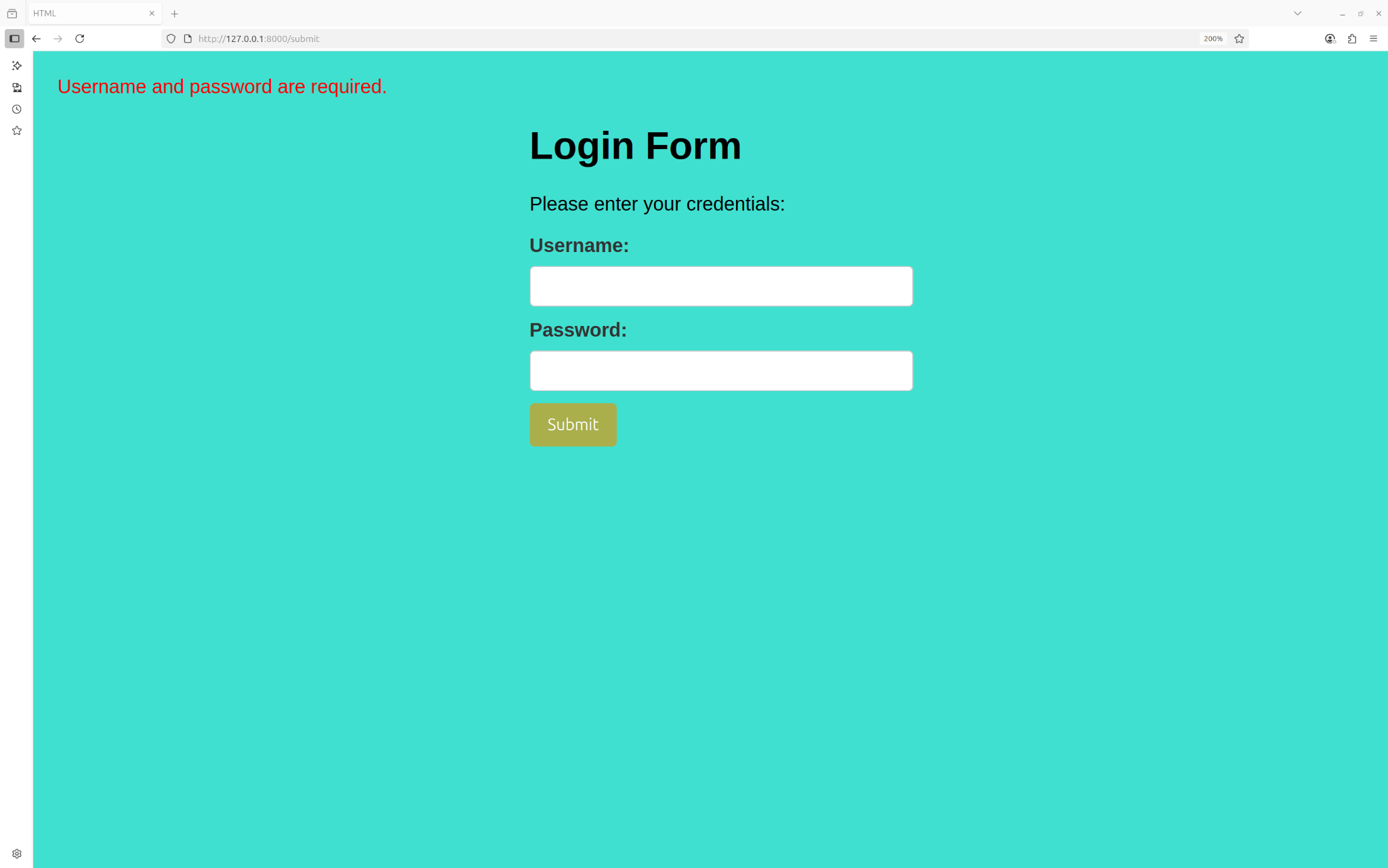
4. Ouput1: deb username and password is entered.



5. Output2: “Data submitted successfully!” message getting displayed upon successful submission.



6. Username and password successfully got inserted into the DB.



7. Error being displayed on the same page

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