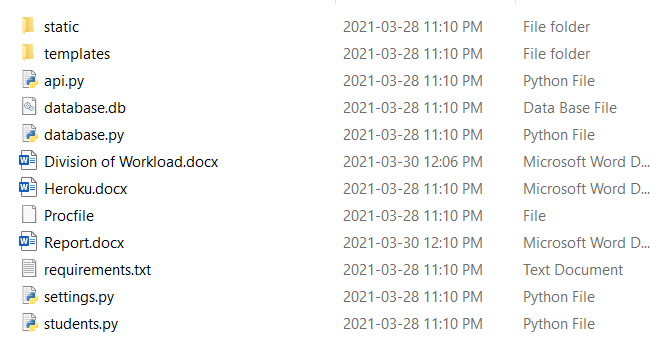
**Student API**

RESTful API using Flask

By Umutcan Asutlu and Ramon Vilarins

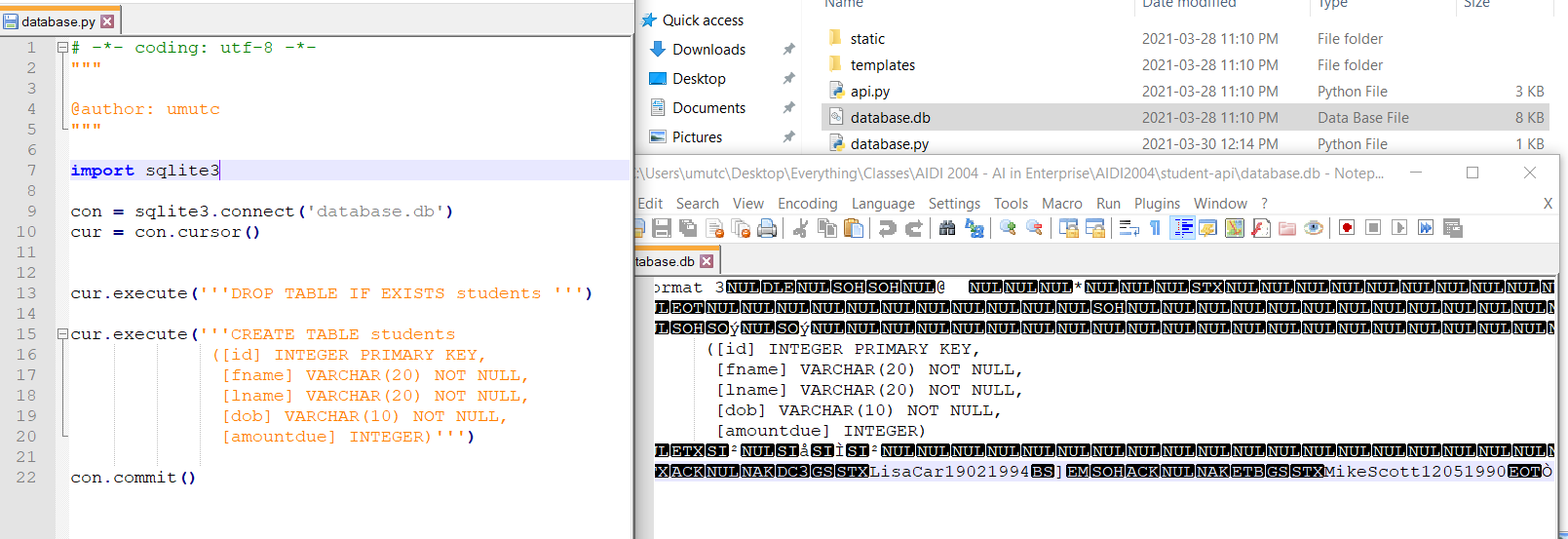
**API Directory**

We used python code for creating database and API. The flask API’s front-end is done by using HTML5 and CSS3. It was supposed to be uploaded to Heroku so Procfile and requirements.txt are generated.



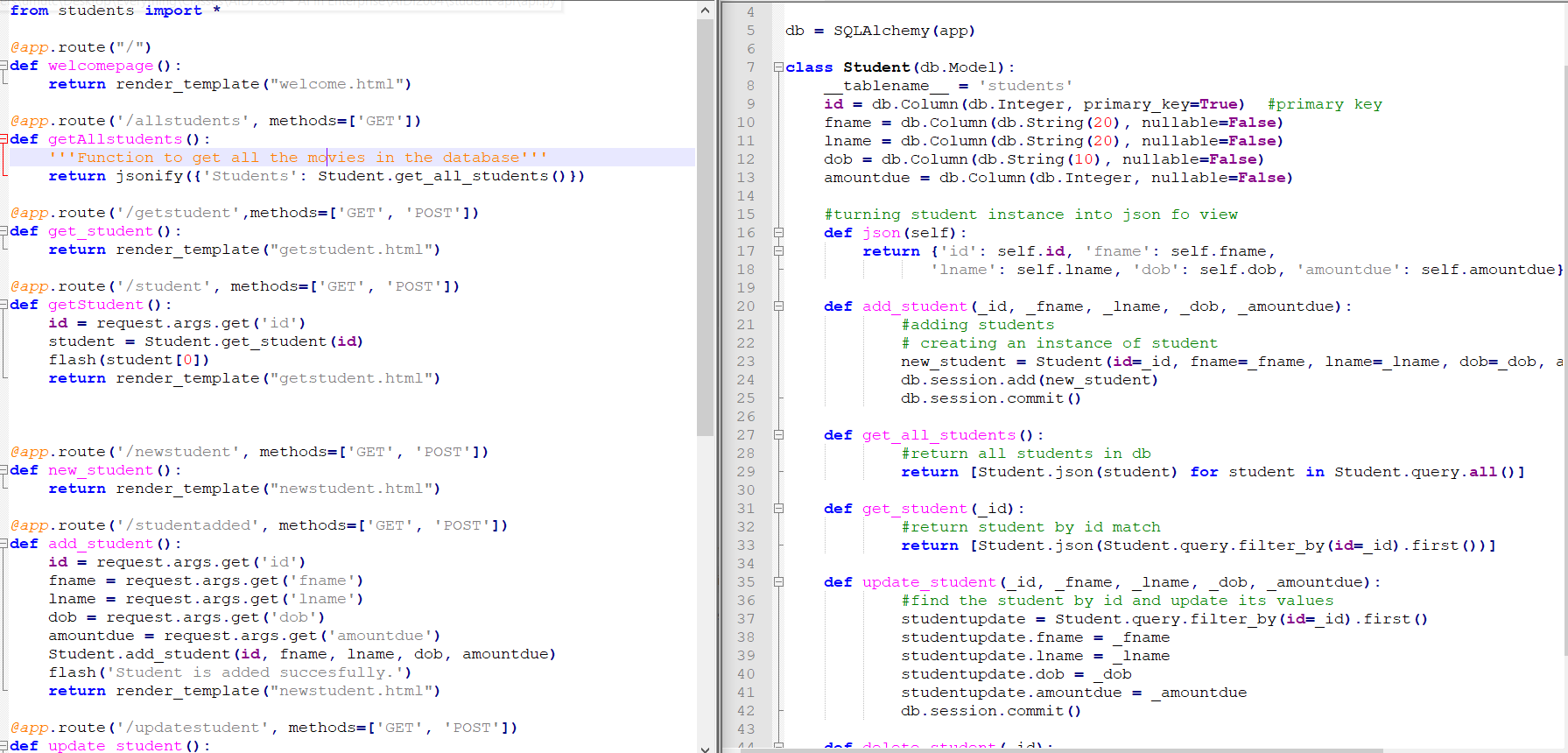
**Database**

Database is created using Python code and with SQL Alchemy it is connected to API.



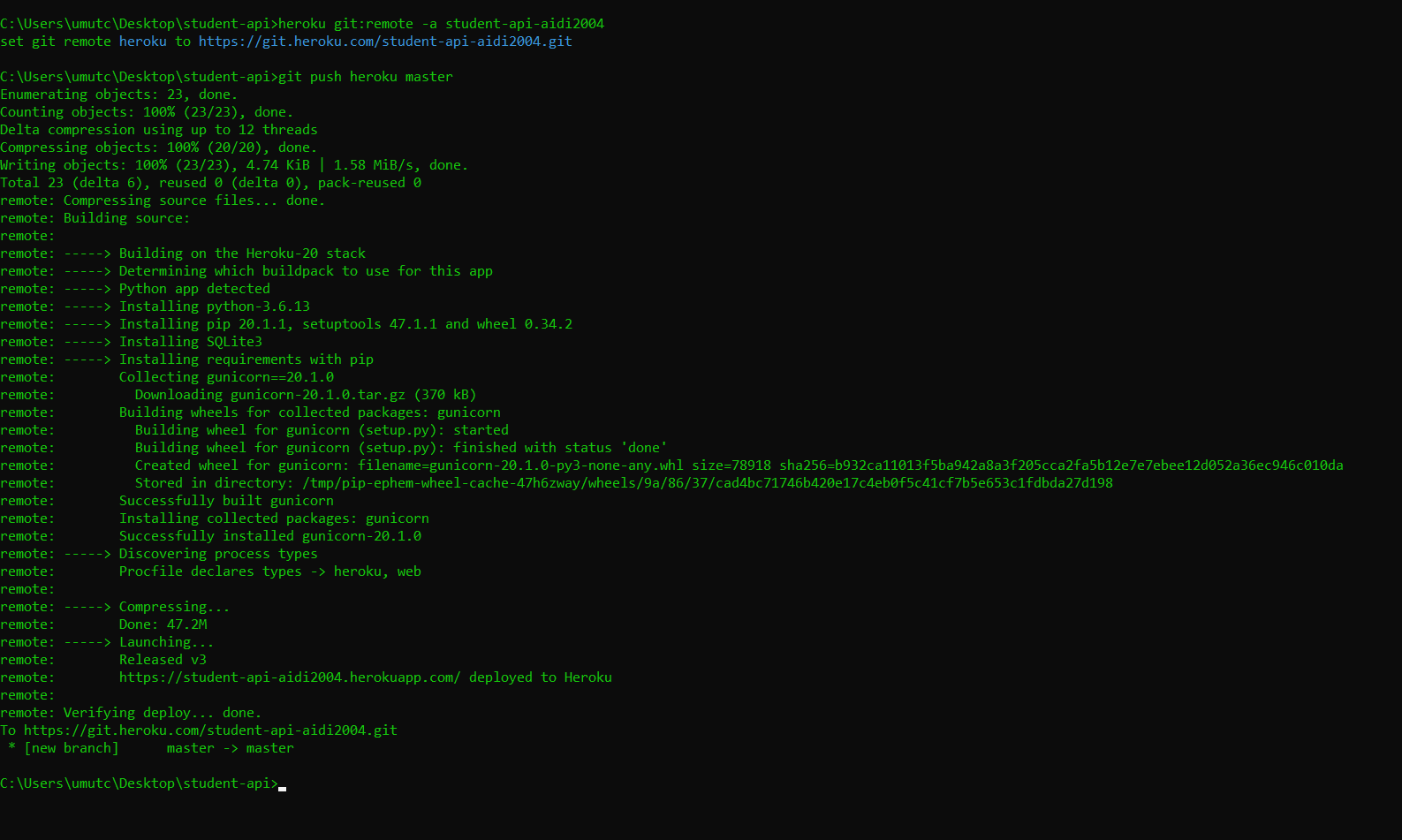
**API**

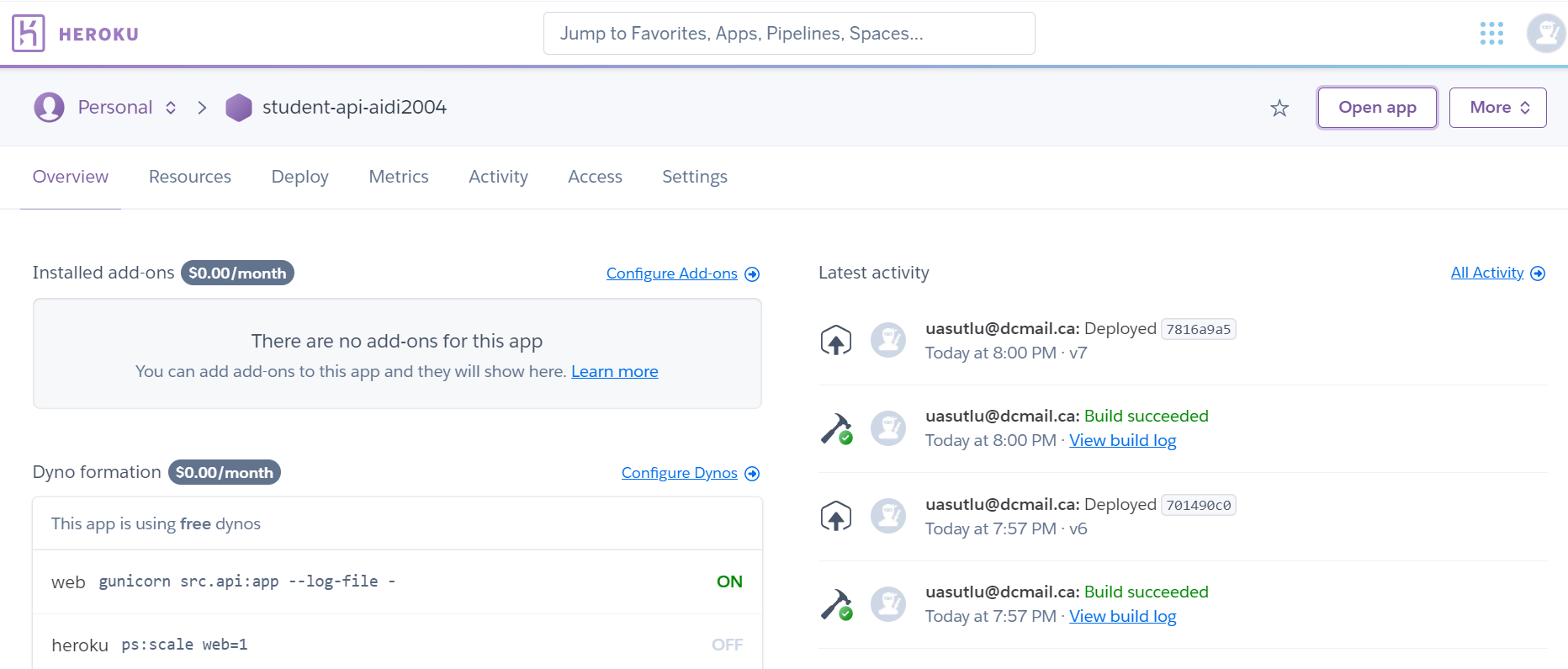
API has different pages for each CRUD operations. These operations are applied on student object.

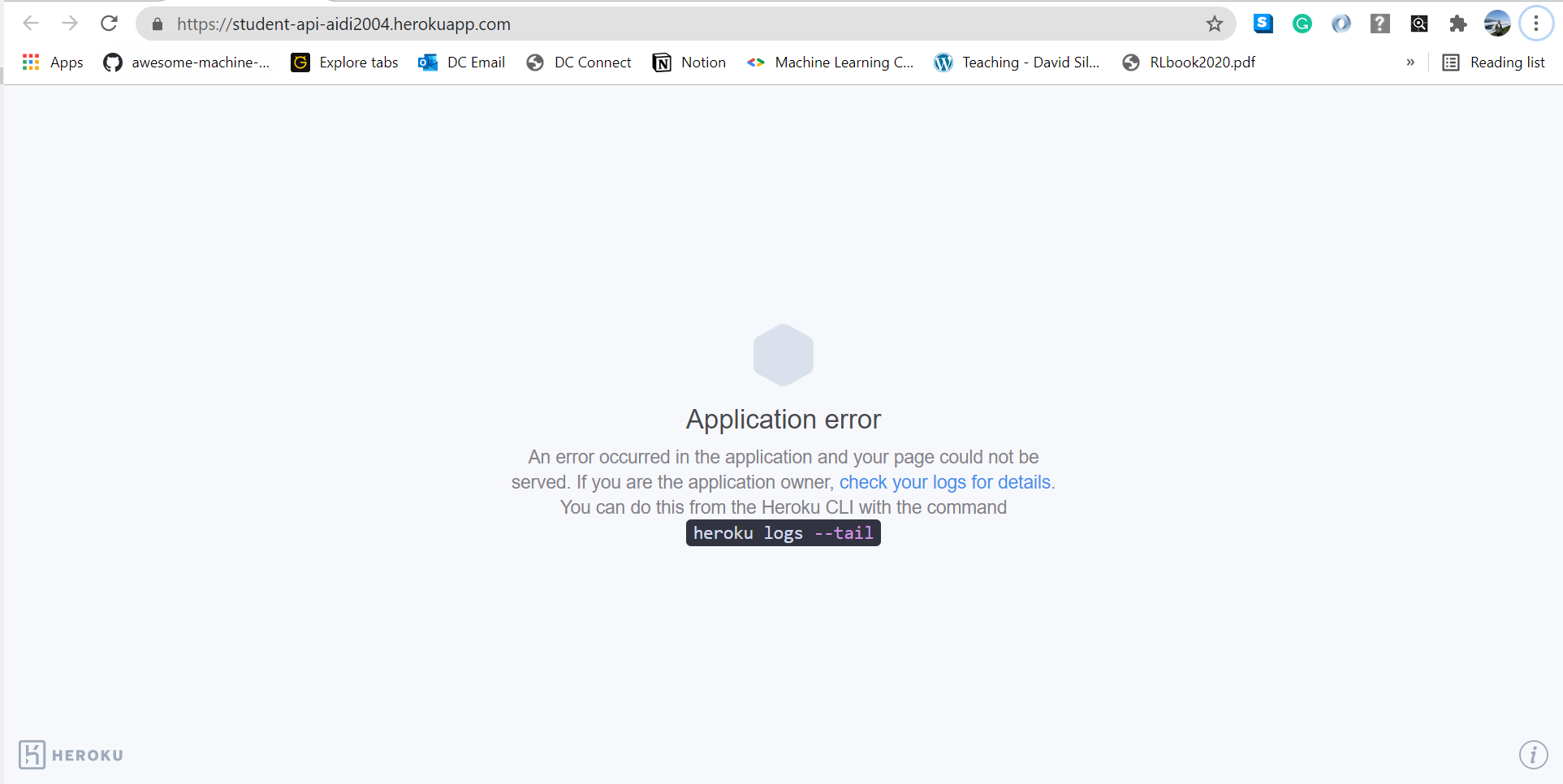


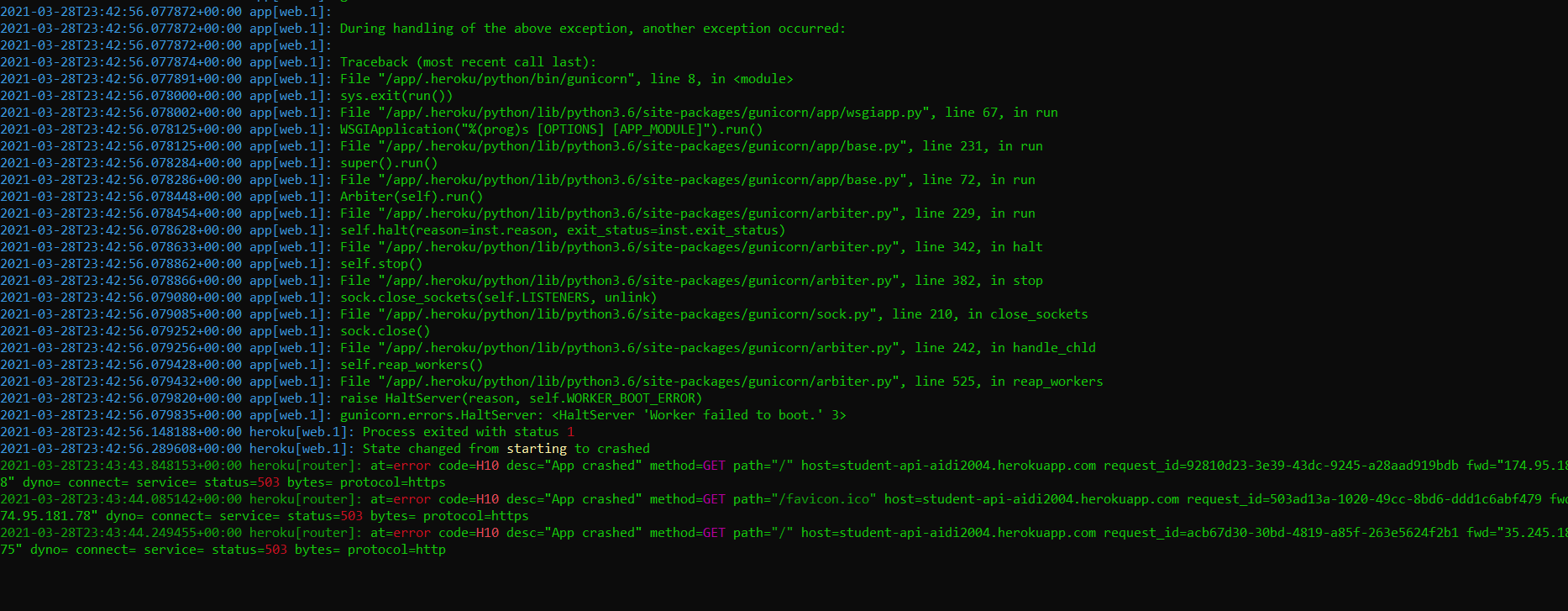
**Usage**

We tried to deploy it to Heroku but for unknown reasons the app kept crashing even though the deployment was successful.



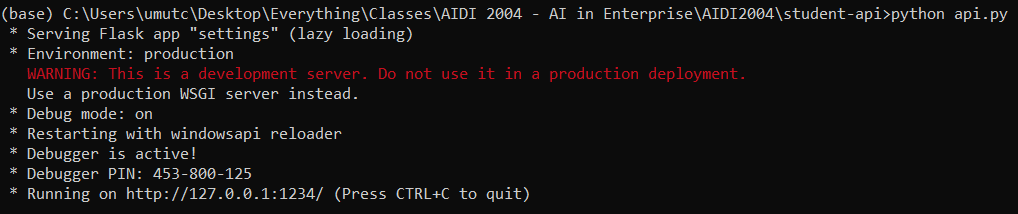


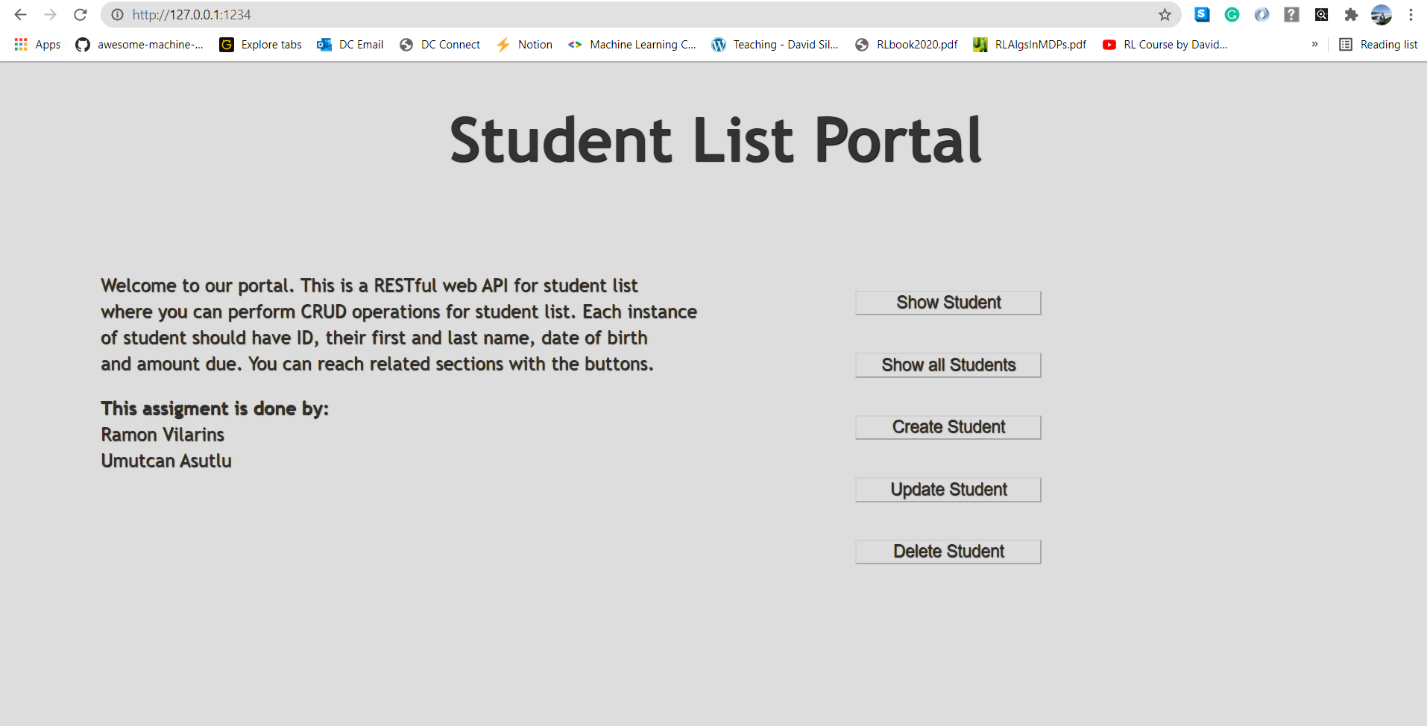




**Local View**

However, it is still possible to use the API on local computer.

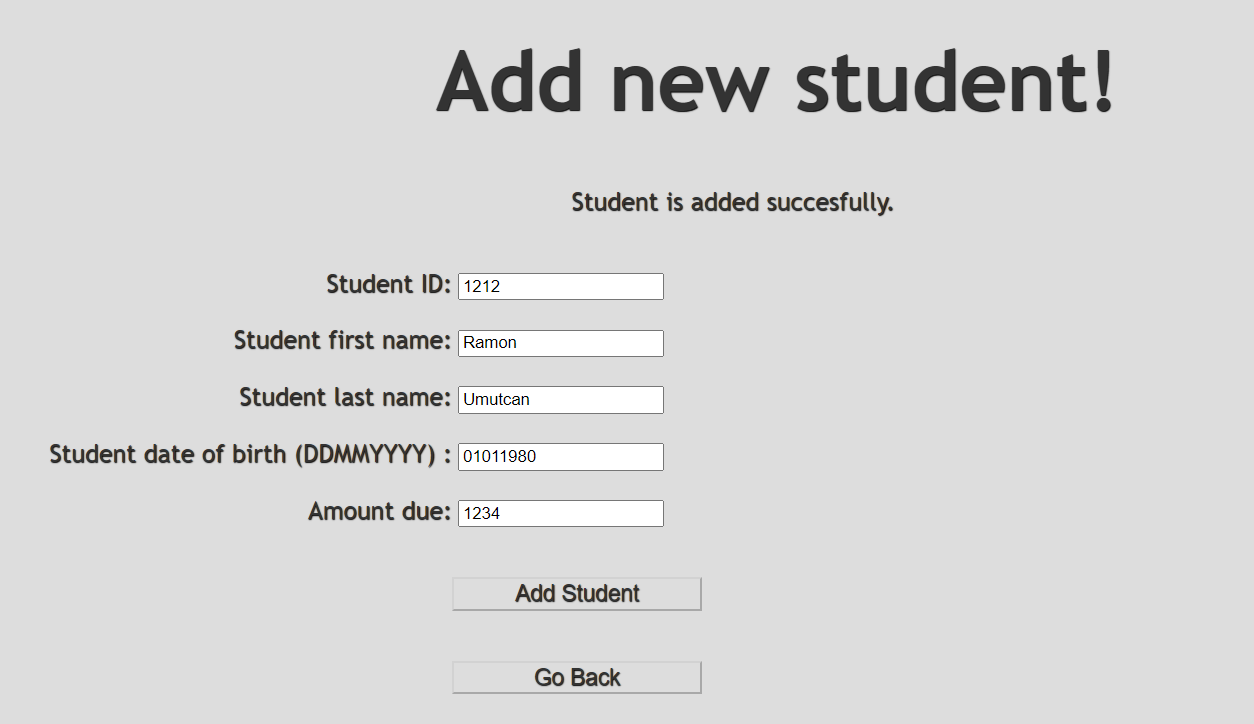


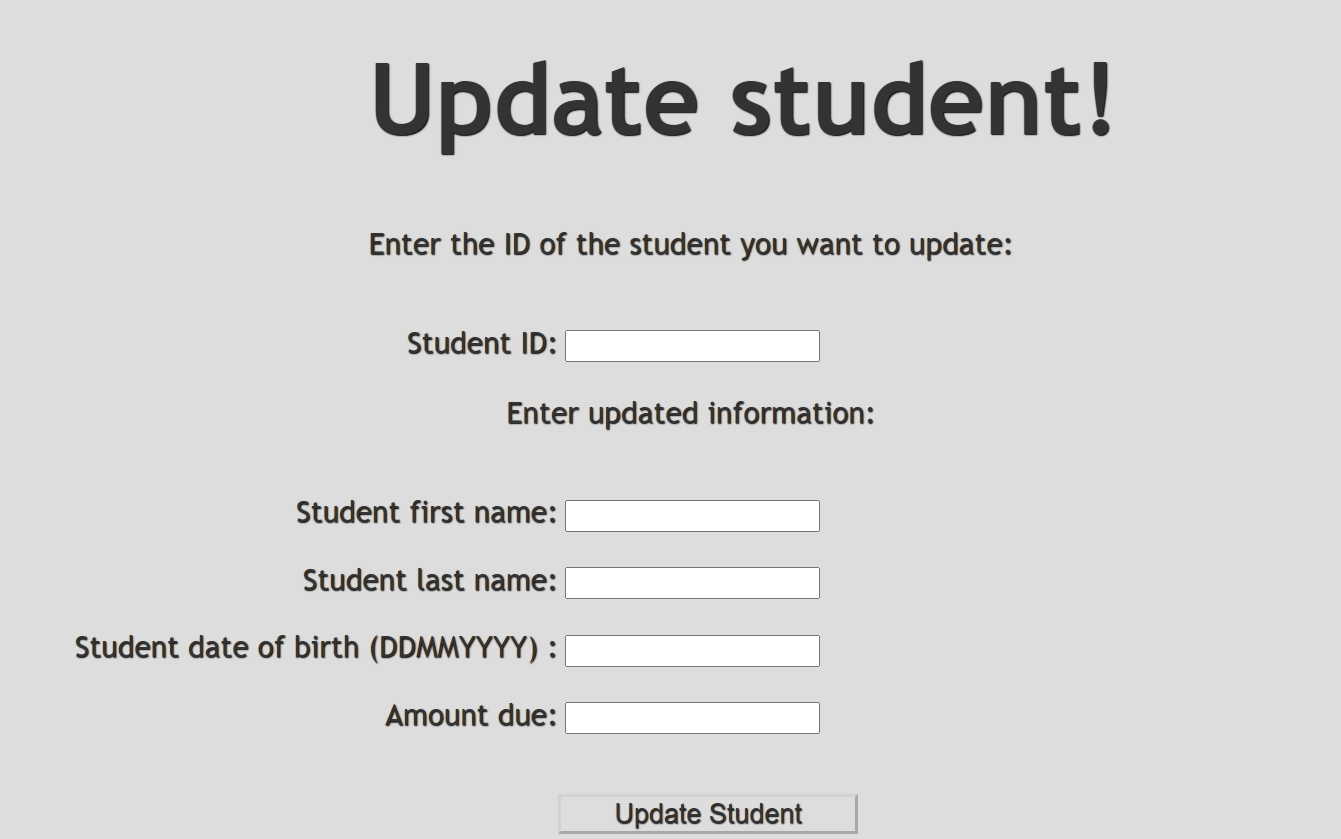


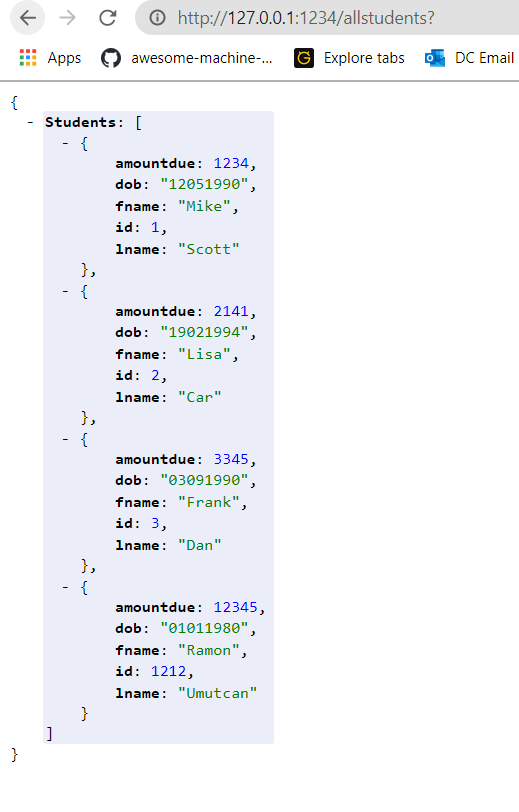
**Notes**

It can show student by given ID, it can create, delete, or update any student. The API is fully connected to the database. ID is primary key, but it is defined by the user, it is not auto generated. For show, update and delete the ID must be valid within the database.

To protect API logic and show it could have been used by services like POSTMAN and such, show all students page is prepared in JSON view.







**Division of Workload**

|  |  |
| --- | --- |
| **Task** | **Name** |
| API Architecture | Ramon |
| API Functions | Umutcan |
| Flask CRUD Operations | Ramon |
| Flask Front-End (HTML/CSS) | Umutcan |
| Database | Umutcan |