Cloud Computing - Mini Project Report

Deployment of Web App Using AWS Cloud

April 2023

Submitted By:

Anushka Pandey **(PES1UG20CS072)**

Anushka Siri Raghunandan **(PES1UG20CS073)**

B Praneetha **(PES1UG20CS106)**

Devika S **(PES1UG20CS123)**

VI Semester Section B

PES University

Short Description and Scope of the Project

In this project, we have created a simple GUI CRUD application, Waste Management System, using Python Streamlit for the frontend and MySQL for the database. We then connect our database to Amazon RDS and deploy our app in an AWS EC2 instance. Github actions are used to automate your build, test, and deployment pipeline. Nginx is also integrated, which is a web server that can also be used as a reverse proxy, load balancer, mail proxy, and HTTP cache.

Scope of this project includes performing simple CRUD operations on the web application deployed on the EC2 instance. The changes made are persistent and are stored onto the RDS instance.

Any pushes or updates made to the GitHub repository will be reflected on the web application automatically due to the use of GitHub Actiona.

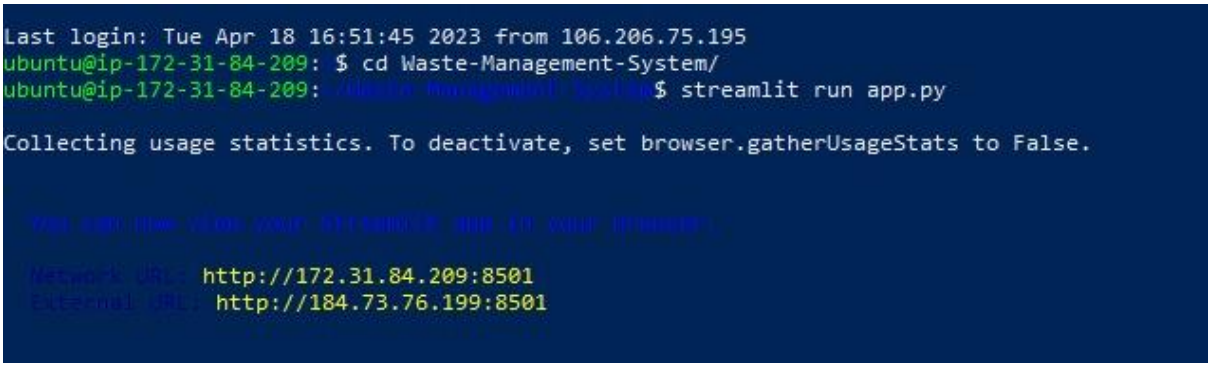
The Nginx server redirects requests from port 80 to our DNS.

Methodology

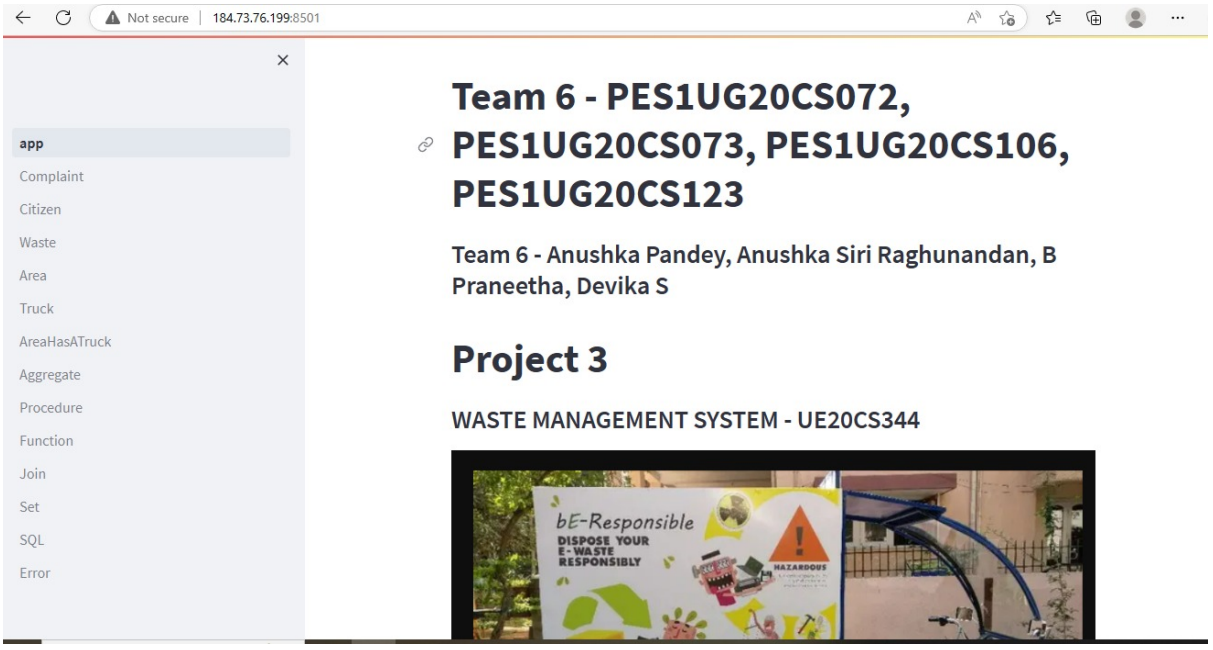
1. We created a simple Streamlit application that can be used to perform CRUD operations. The database used is MySQL.
2. We launched an EC2 instance to deploy the web application created.
3. We created an RDS instance on AWS RDS for the MySQL database.
4. The EC2 instance is then connected to the RDS instance and appropriate changes are made to the inbound security rules.
5. SQLWorkBench is the MySQL client that is used to connect to the RDS instance.
6. The data is then exported into the RDS instance using the SQLWorkBench connection.
7. We install the necessary dependencies like git, pip and streamlit.
8. We then clone the git repository containing the web application files into the EC2 instance.
9. We can now run the streamlit application from the EC2 instance using the command streamlit run app.py
10. We then created the python-app.yml file in the .github/workflows directory to facilitate continuous integration and deployment. Secret keys have been created and used for the same.
11. The nginx server is used to redirect requests from port 80 to our DNS

Testing

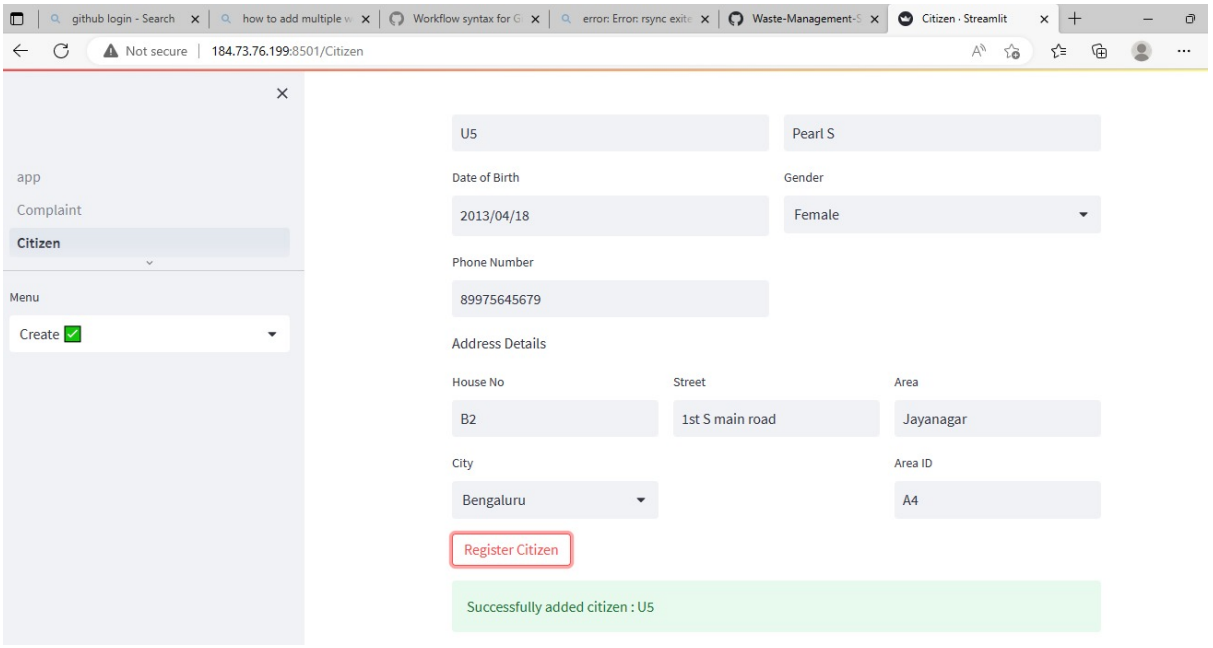
Starting the streamlit app on the EC2 instance



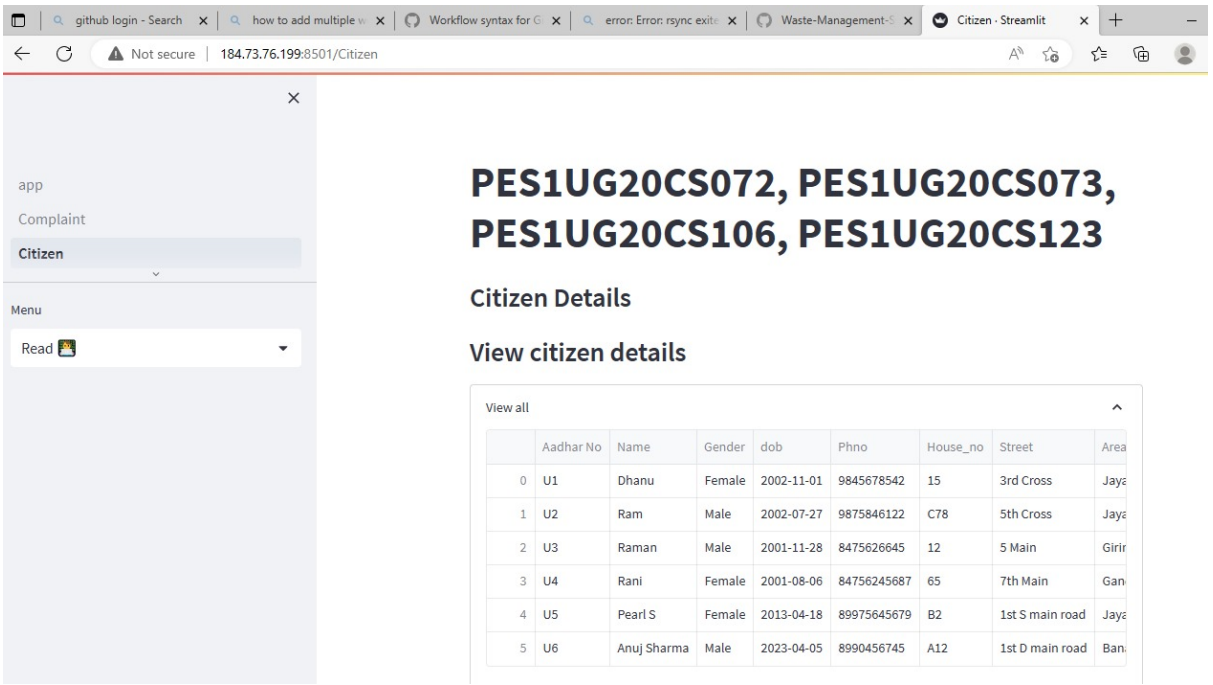
The CRUD Web App



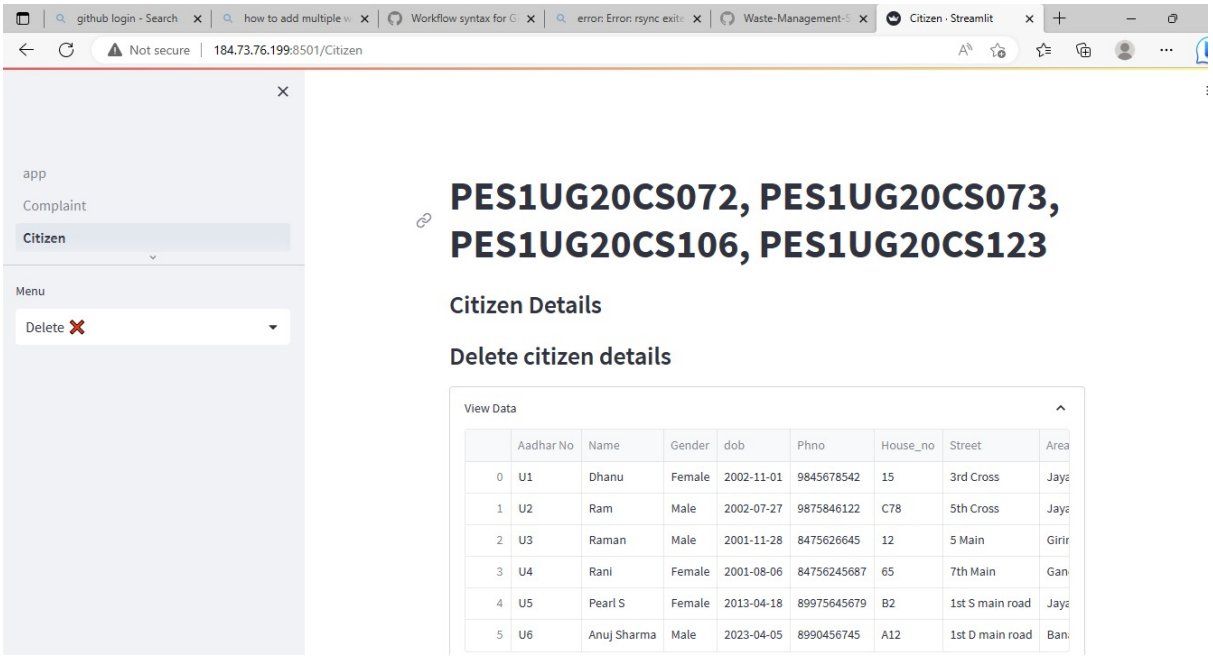
1. Create command

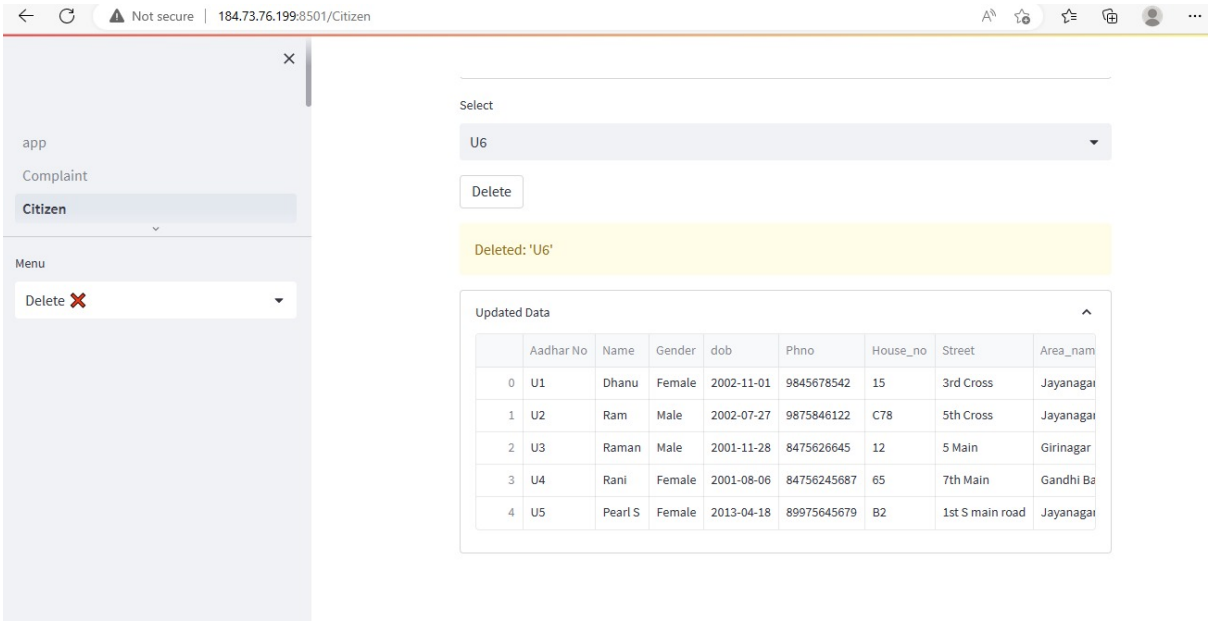


2. Read command (previously added U5 has been added)

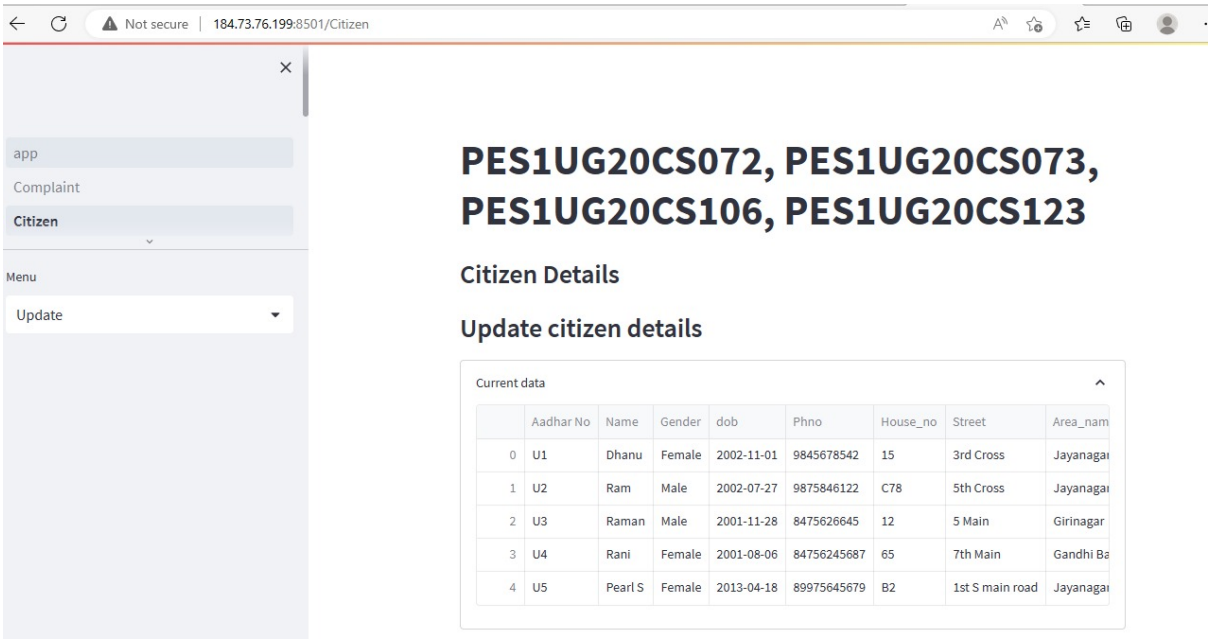


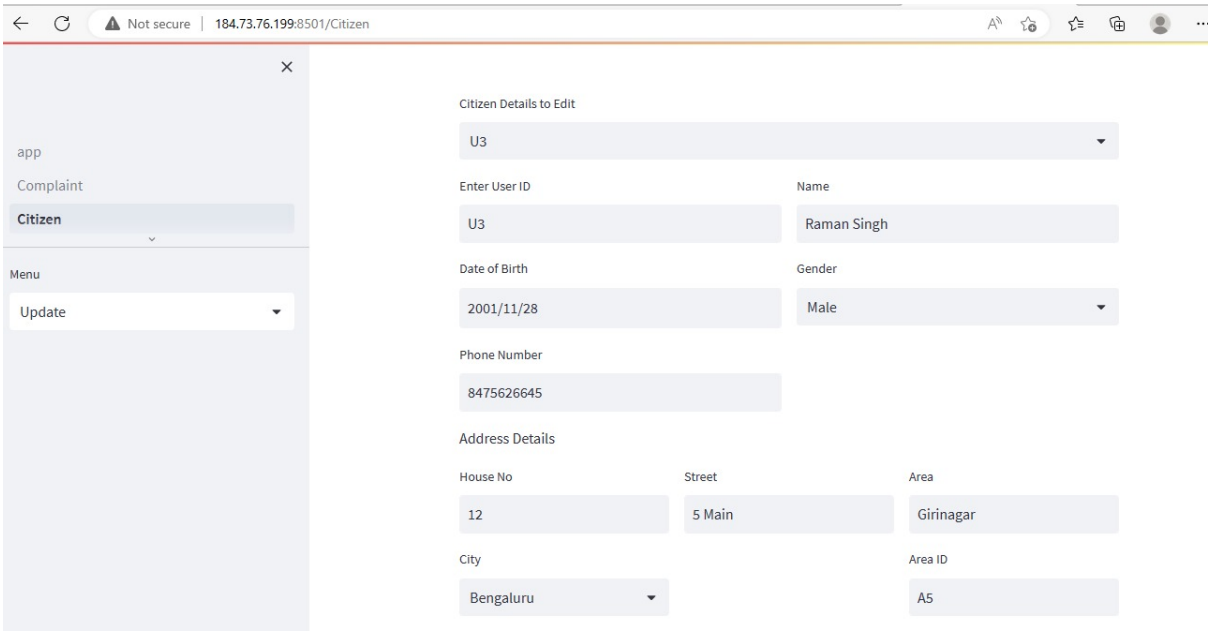
3. Delete command

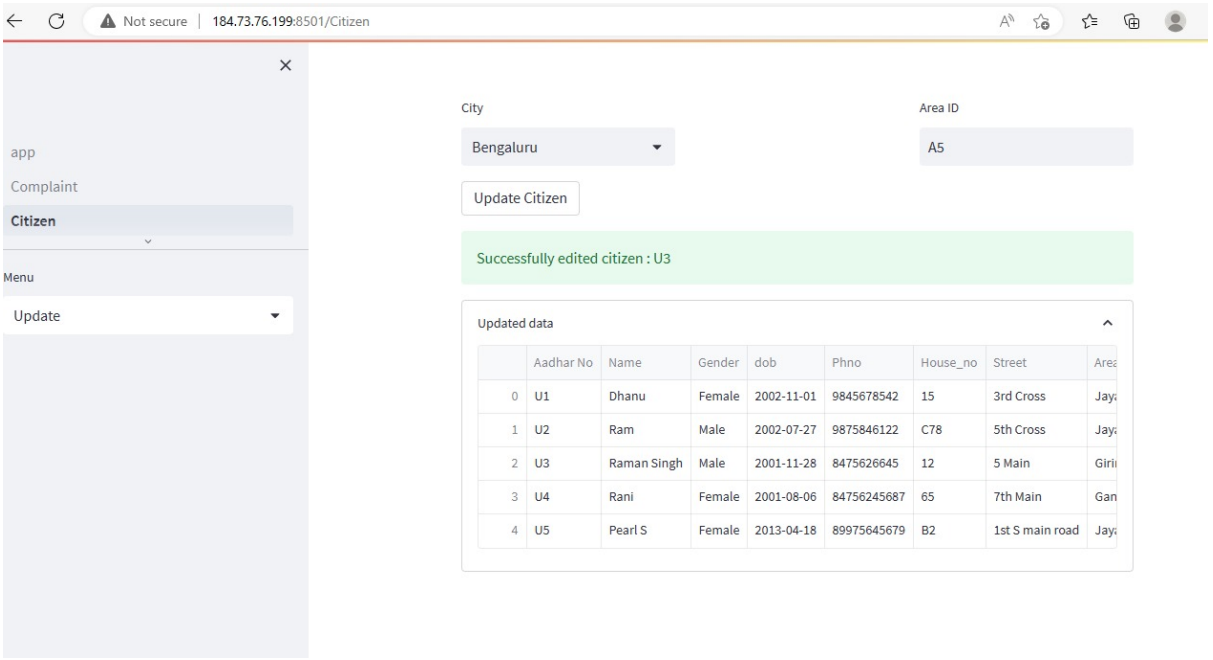




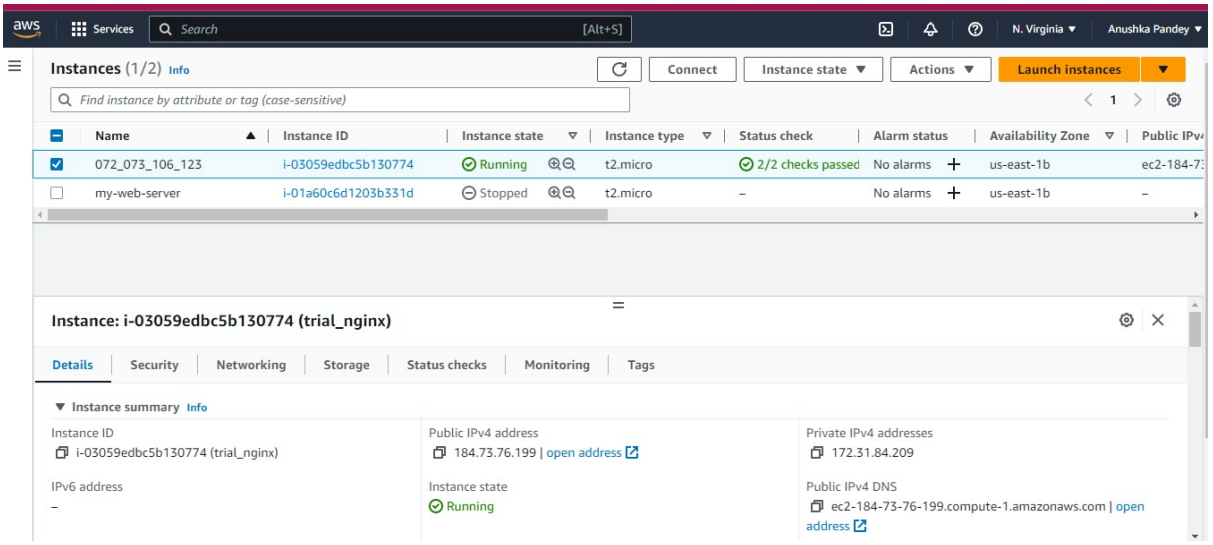
4. Update command



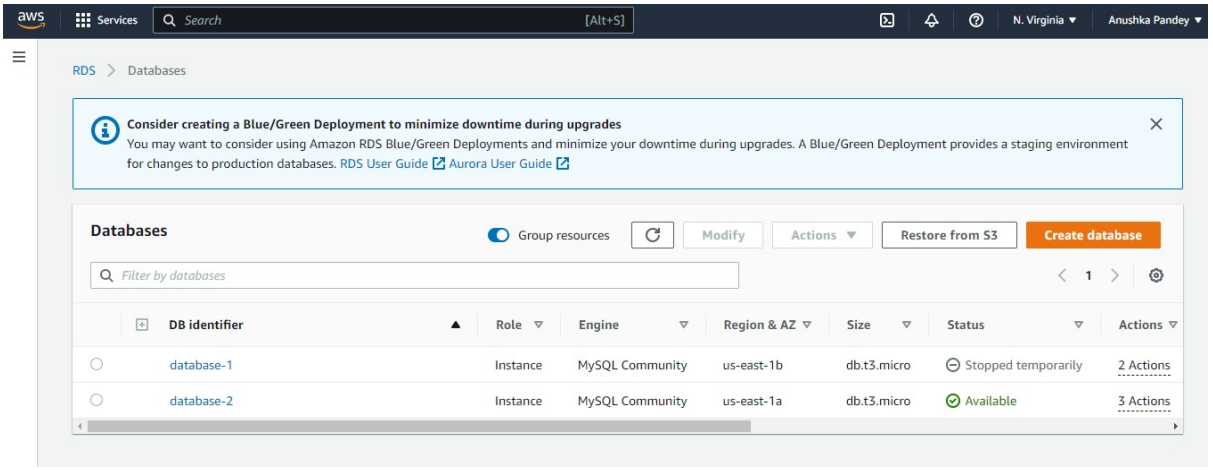




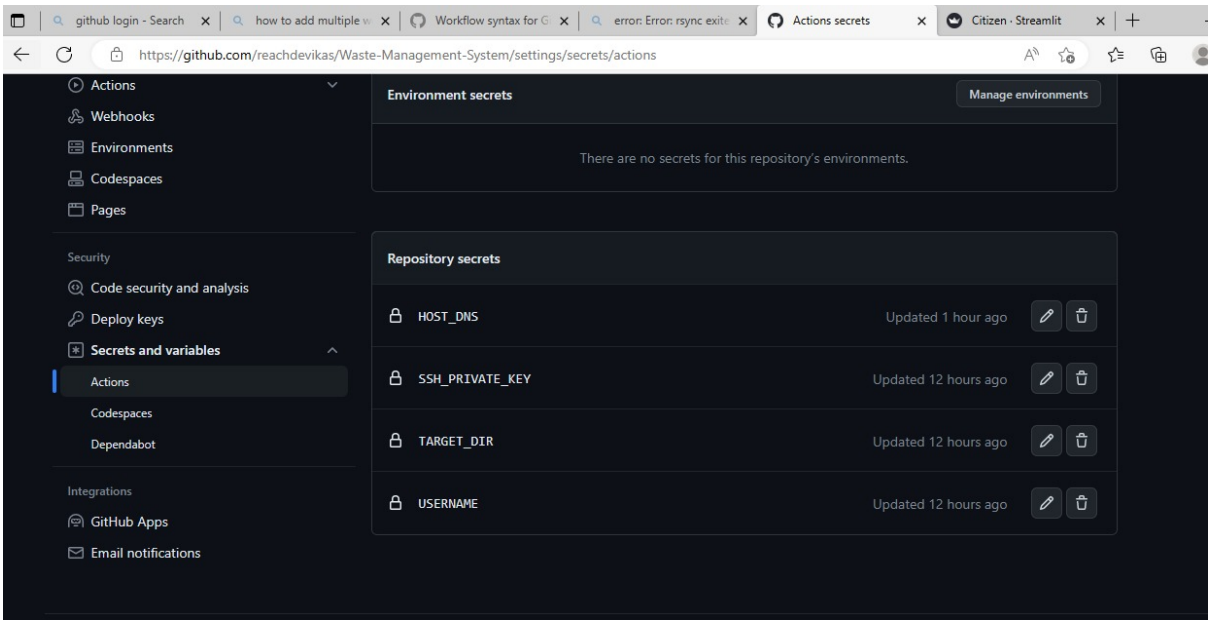
EC2 instance



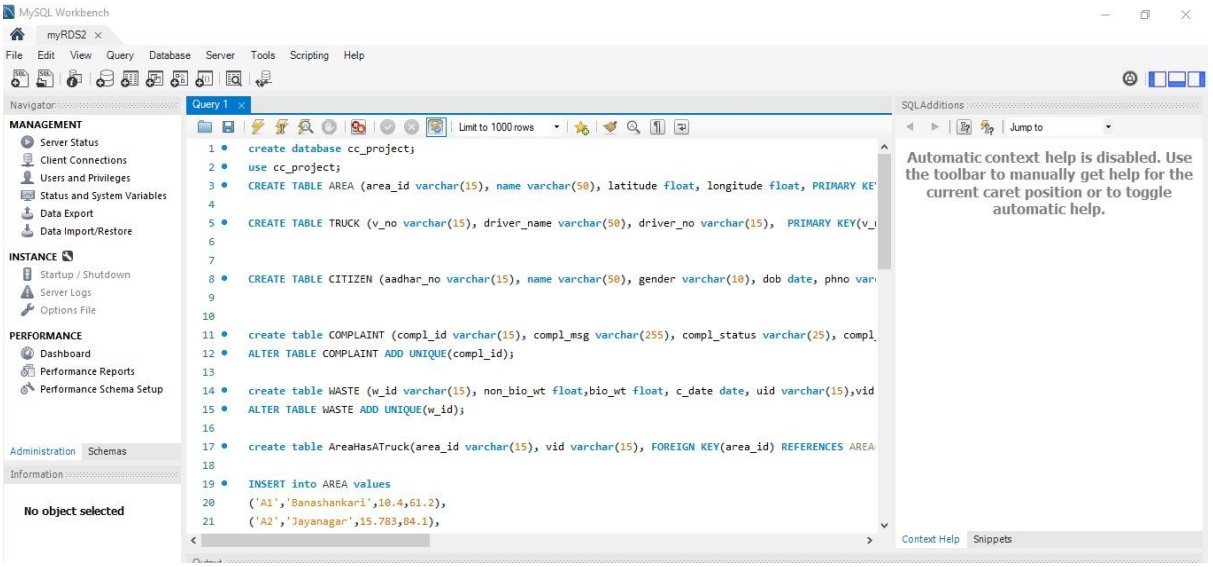
AWS RDS connection with the app’s MySQL database



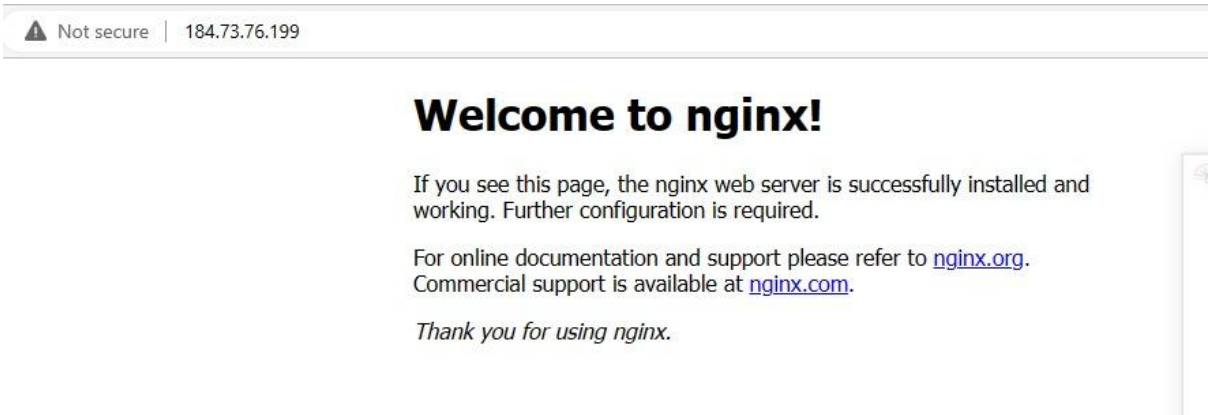
Github repository secrets



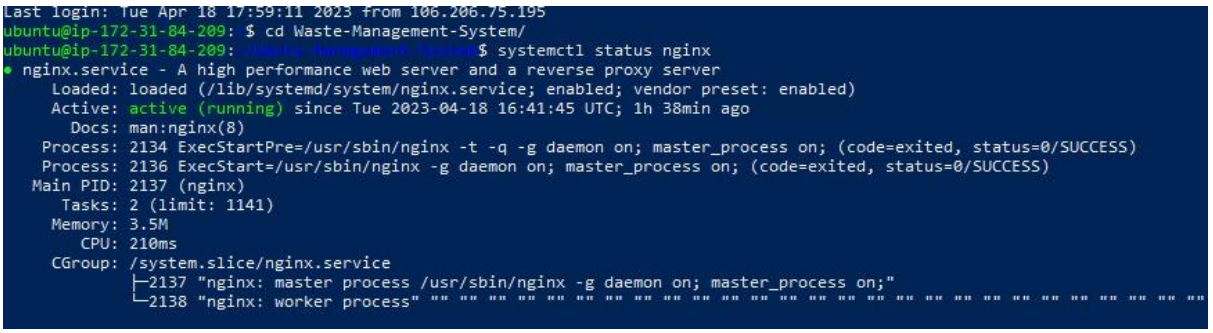
SQL WorkBench connected to the RDS instance and exporting data into the RDS instance



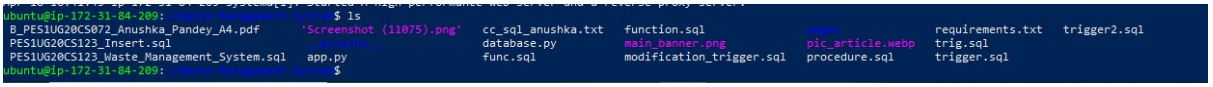
Nginx



Nginx server active



Github actions(before uploading the file)



Github actions(after uploading the file)

Results and Conclusions

1. The web application has been deployed on the EC2 instance
2. The web application is connected to the RDS instance.
3. All CRUD operations can be carried out on the tables on the web application.
4. Any push operations made to the GitHub repository is reflected on the web application automatically.
5. Requests to port 80 nginx server are redirected to our DNS.