

**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**

## **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 Actions.

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

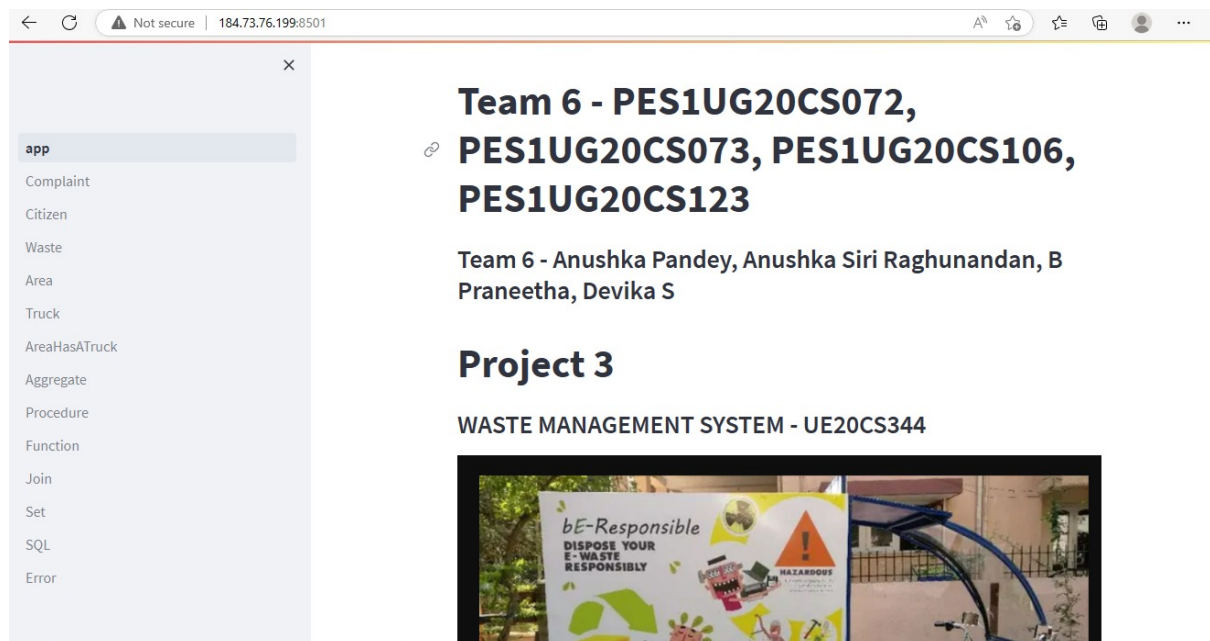
```
Last login: Tue Apr 18 16:51:45 2023 from 106.206.75.195
ubuntu@ip-172-31-84-209: ~$ cd Waste-Management-System/
ubuntu@ip-172-31-84-209: ~$ streamlit run app.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: http://172.31.84.209:8501
External URL: http://184.73.76.199:8501
```

## The CRUD Web App



# 1. Create command

app

Complaint

Citizen

Menu

Create

U5

Pearl S

Date of Birth

2013/04/18

Gender

Female

Phone Number

89975645679

Address Details

House No

B2

Street

1st S main road

Area

Jayanagar

City

Bengaluru

Area ID

A4

Register Citizen

Successfully added citizen : U5

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

app

Complaint

Citizen

Menu

Read

PES1UG20CS072, PES1UG20CS073,  
PES1UG20CS106, PES1UG20CS123

Citizen Details

View citizen details

View all

	Aadhar No	Name	Gender	dob	Phno	House_no	Street	Area
0	U1	Dhanu	Female	2002-11-01	9845678542	15	3rd Cross	Jayc
1	U2	Ram	Male	2002-07-27	9875846122	C78	5th Cross	Jayc
2	U3	Raman	Male	2001-11-28	8475626645	12	5 Main	Girir
3	U4	Rani	Female	2001-08-06	84756245687	65	7th Main	Gan
4	U5	Pearl S	Female	2013-04-18	89975645679	B2	1st S main road	Jayc
5	U6	Anuj Sharma	Male	2023-04-05	8990456745	A12	1st D main road	Ban

### 3. Delete command

app

Complaint

Citizen

Menu

Delete

## PES1UG20CS072, PES1UG20CS073, PES1UG20CS106, PES1UG20CS123

### Citizen Details

#### Delete citizen details

View Data

	Aadhar No	Name	Gender	dob	Phno	House_no	Street	Area
0	U1	Dhanu	Female	2002-11-01	9845678542	15	3rd Cross	Jaya
1	U2	Ram	Male	2002-07-27	9875846122	C78	5th Cross	Jaya
2	U3	Raman	Male	2001-11-28	8475626645	12	5 Main	Girir
3	U4	Rani	Female	2001-08-06	84756245687	65	7th Main	Gan
4	U5	Pearl S	Female	2013-04-18	89975645679	B2	1st S main road	Jaya
5	U6	Anuj Sharma	Male	2023-04-05	8990456745	A12	1st D main road	Ban

app

Complaint

Citizen

Menu

Delete

Select

U6

Delete

Deleted: 'U6'

Updated Data

	Aadhar No	Name	Gender	dob	Phno	House_no	Street	Area_nam
0	U1	Dhanu	Female	2002-11-01	9845678542	15	3rd Cross	Jayanagai
1	U2	Ram	Male	2002-07-27	9875846122	C78	5th Cross	Jayanagai
2	U3	Raman	Male	2001-11-28	8475626645	12	5 Main	Girinagar
3	U4	Rani	Female	2001-08-06	84756245687	65	7th Main	Gandhi Ba
4	U5	Pearl S	Female	2013-04-18	89975645679	B2	1st S main road	Jayanagai

## 4. Update command

app

Complaint

Citizen

Menu

Update

# PES1UG20CS072, PES1UG20CS073, PES1UG20CS106, PES1UG20CS123

## Citizen Details

### Update citizen details

Current data

	Aadhar No	Name	Gender	dob	Phno	House_no	Street	Area_nam
0	U1	Dhanu	Female	2002-11-01	9845678542	15	3rd Cross	Jayanagai
1	U2	Ram	Male	2002-07-27	9875846122	C78	5th Cross	Jayanagai
2	U3	Raman	Male	2001-11-28	8475626645	12	5 Main	Girinagar
3	U4	Rani	Female	2001-08-06	84756245687	65	7th Main	Gandhi Ba
4	U5	Pearl S	Female	2013-04-18	89975645679	B2	1st S main road	Jayanagai

app

Complaint

Citizen

Menu

Update

Citizen Details to Edit

U3

Enter User ID

U3

Name

Raman Singh

Date of Birth

2001/11/28

Gender

Male

Phone Number

8475626645

Address Details

House No

12

Street

5 Main

Area

Girinagar

City

Bengaluru

Area ID

A5

app  
Complaint  
Citizen  
Menu  
Update

City

Bengaluru

Area ID

A5

Update Citizen

Successfully edited citizen : U3

Updated data

	Aadhar No	Name	Gender	dob	Phno	House_no	Street	Area
0	U1	Dhanu	Female	2002-11-01	9845678542	15	3rd Cross	Jay
1	U2	Ram	Male	2002-07-27	9875846122	C78	5th Cross	Jay
2	U3	Raman Singh	Male	2001-11-28	8475626645	12	5 Main	Giri
3	U4	Rani	Female	2001-08-06	84756245687	65	7th Main	Gan
4	U5	Pearl S	Female	2013-04-18	89975645679	B2	1st S main road	Jay

## EC2 instance

aws Services Search [Alt+S]

Instances (1/2) Info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
072_073_106_123	i-03059edbc5b130774	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-184-73-76-199.compute-1.amazonaws.com
my-web-server	i-01a60c6d1203b331d	Stopped	t2.micro	-	No alarms	us-east-1b	-

Instance: i-03059edbc5b130774 (trial\_nginx)

Details Security Networking Storage Status checks Monitoring Tags

Instance summary Info

Instance ID  
i-03059edbc5b130774 (trial\_nginx)

IPv6 address  
-

Public IPv4 address  
184.73.76.199 | open address

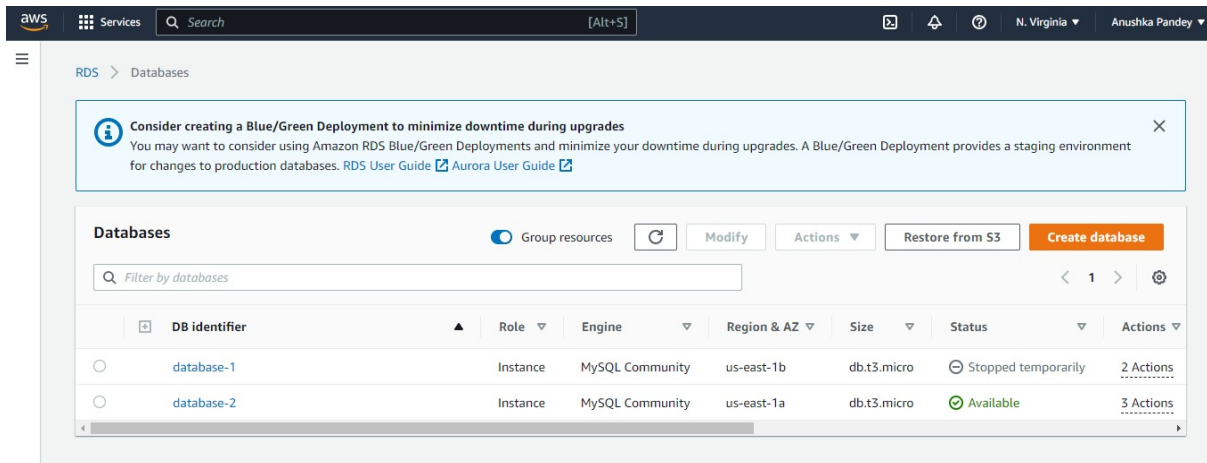
Instance state  
Running

Private IPv4 addresses  
172.31.84.209

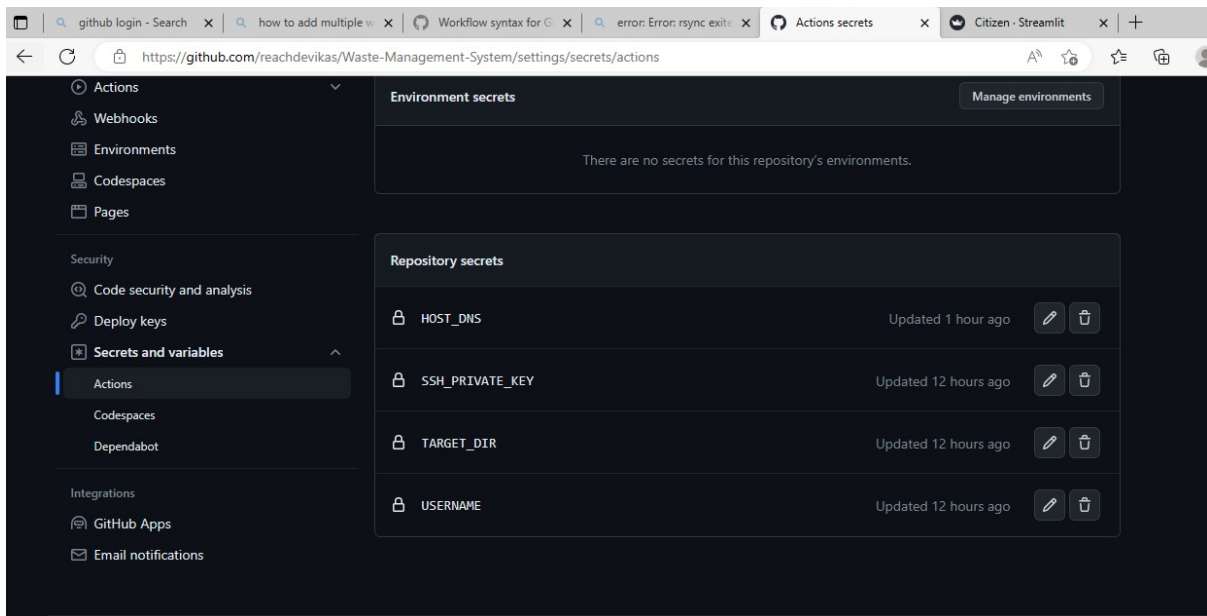
Public IPv4 DNS  
ec2-184-73-76-199.compute-1.amazonaws.com | open address



## AWS RDS connection with the app's MySQL database



## Github repository secrets



The screenshot displays the MySQL Workbench environment. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar contains several sections: MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup), and Administration (Schemas). The main workspace is titled 'Query 1' and contains a SQL script. The script starts with creating a database 'cc\_project' and using it. It then creates several tables: 'AREA' with columns for area\_id, name, latitude, and longitude; 'TRUCK' with columns for v\_no, driver\_name, and driver\_no; 'CITIZEN' with columns for aadhar\_no, name, gender, dob, and phno; 'COMPLAINT' with columns for compl\_id, compl\_msg, compl\_status, and compl\_date; 'WASTE' with columns for w\_id, non\_bio\_wt, bio\_wt, c\_date, and uid; and 'AreaHasATruck' with columns for area\_id and vid. The script also includes an INSERT statement for the 'AREA' table. The right-hand pane shows a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

```

1 • create database cc_project;
2 • use cc_project;
3 • CREATE TABLE AREA (area_id varchar(15), name varchar(50), latitude float, longitude float, PRIMARY KEY (area_id));
4
5 • CREATE TABLE TRUCK (v_no varchar(15), driver_name varchar(50), driver_no varchar(15), PRIMARY KEY (v_no));
6
7
8 • CREATE TABLE CITIZEN (aadhar_no varchar(15), name varchar(50), gender varchar(10), dob date, phno varchar(15));
9
10
11 • create table COMPLAINT (compl_id varchar(15), compl_msg varchar(255), compl_status varchar(25), compl_date date, PRIMARY KEY (compl_id));
12 • ALTER TABLE COMPLAINT ADD UNIQUE (compl_id);
13
14 • create table WASTE (w_id varchar(15), non_bio_wt float, bio_wt float, c_date date, uid varchar(15), vid varchar(15), PRIMARY KEY (w_id));
15 • ALTER TABLE WASTE ADD UNIQUE (w_id);
16
17 • create table AreaHasATruck (area_id varchar(15), vid varchar(15), FOREIGN KEY (area_id) REFERENCES AREA (area_id));
18
19 • INSERT into AREA values
20 ('A1','Banashankari',10.4,61.2),
21 ('A2','Jayanagar',15.783,84.1),

```

A screenshot of a web browser window showing the nginx welcome page. The browser's address bar at the top displays a warning icon, the text "Not secure", and the IP address "184.73.76.199". The main content area of the browser has a light gray background and features the heading "Welcome to nginx!" in a large, bold, black font. Below the heading, there is a paragraph of text: "If you see this page, the nginx web server is successfully installed and working. Further configuration is required." This is followed by another paragraph: "For online documentation and support please refer to [nginx.org](https://nginx.org). Commercial support is available at [nginx.com](https://nginx.com)." The final line of text on the page is "Thank you for using nginx." in an italicized font. On the far right edge of the browser window, a small portion of a sidebar with a search icon is visible.

```

Last login: Tue Apr 18 17:59:11 2023 from 106.206.75.195
ubuntu@ip-172-31-84-209: ~$ cd Waste-Management-System/
ubuntu@ip-172-31-84-209: ~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-04-18 16:41:45 UTC; 1h 38min ago
     Docs: man:nginx(8)
   Process: 2134 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 2136 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 2137 (nginx)
    Tasks: 2 (limit: 1141)
   Memory: 3.5M
      CPU: 210ms
   CGroup: /system.slice/nginx.service
           └─2137 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           └─2138 "nginx: worker process"

```

## Github actions(before uploading the file)

```
ubuntu@ip-172-31-84-209: /Waste_Management_System$ ls
B_PES1UG20CS072_Anushka_Pandey_A4.pdf      'Screenshot (11075).png'  cc_sql_anushka.txt  function.sql  main_banner.png  requirements.txt  trigger2.sql
PES1UG20CS123_Insert.sql                  database.py               func.sql            modification_trigger.sql  pic_article.webp  trig.sql
PES1UG20CS123_Waste_Management_System.sql  app.py                  func.sql            modification_trigger.sql  procedure.sql      trigger.sql
ubuntu@ip-172-31-84-209: /Waste_Management_System$
```

## Github actions(after uploading the file)

```
ubuntu@ip-172-31-84-209: /Waste_Management_System$ ls
'Assignment 5- Introduction to AWS Key Management Service.docx'  'Screenshot (11075).png'  database.py  modification_trigger.sql  requirements.txt
B_PES1UG20CS072_Anushka_Pandey_A4.pdf                          'Screenshot (11075).png'  func.sql     func.sql                  trig.sql
PES1UG20CS123_Insert.sql                                       database.py               function.sql  pic_article.webp         trigger.sql
PES1UG20CS123_Waste_Management_System.sql                      app.py                   func.sql     pic_article.webp         trigger2.sql
ubuntu@ip-172-31-84-209: /Waste_Management_System$
```

## **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.