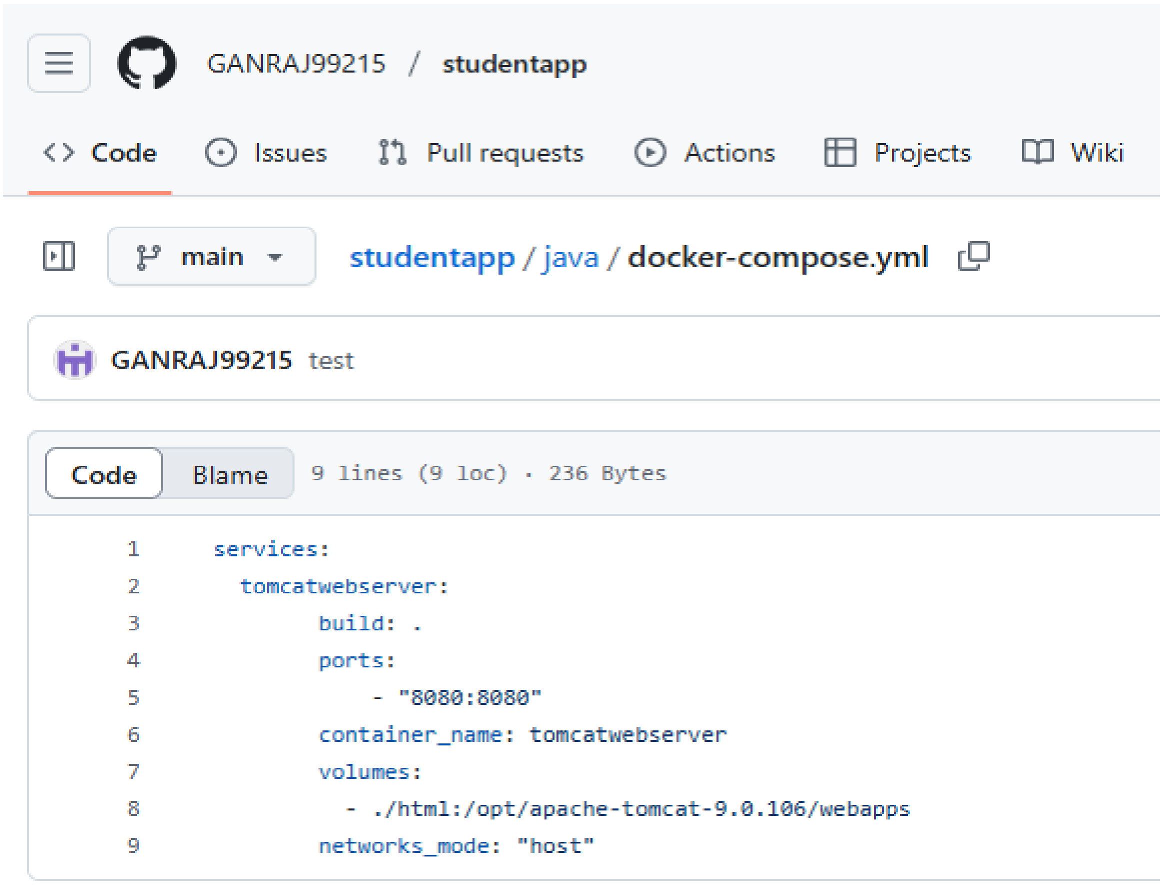
DOCKER COMPOSE : -

First we will write docker compose file



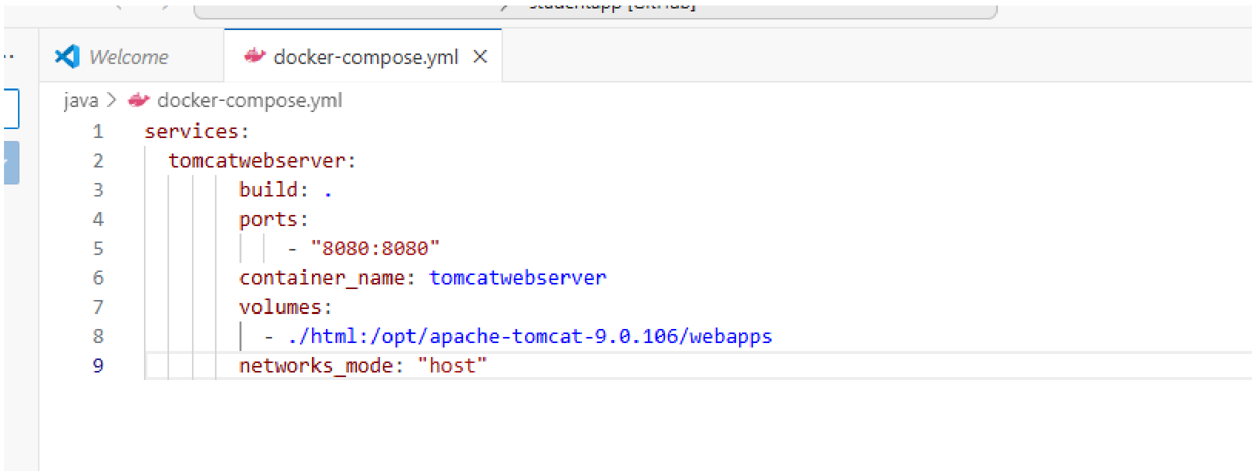
```
services:
  tomcatwebserver:
        build: .
        ports:
            - "8080:8080"
        container_name: tomcatwebserver
        volumes:
          - ./html:/opt/apache-tomcat-9.0.106/webapps
        networks_mode: "host"
```

As you can see in the snap we have write down the context (comments, arguments) that need to be compose our docker. here is split explanation about it.

Explanation:

Key	Meaning
`services`	Defines containers to run. Here, only one: `tomcatwebserver`.
`build: .`	Uses the Dockerfile from the current directory.
`ports`	Maps port `8080` of container to host. Useful only if `network_mode` isn't `host`.
`container_name`	Gives a custom name to the container.
`volumes`	Mounts `./html` folder into Tomcat's `webapps/`. Replaces default WAR.
`networks_mode: "host"`	Shares host's network. Skips Docker's virtual network. Not needed unless required for performance or access.

See we have created a docker compose file to automate the manual process of building containers



```
services:
  tomcatwebserver:
      build: .
      ports:
        - "8080:8080"
      container_name: tomcatwebserver
      volumes:
       - ./html:/opt/apache-tomcat-9.0.106/webapps
      networks_mode: "host"
```

We can also create two or more containers using same docker file but you have be careful on container ports and network (volumes/build if other location have to create).

```
services:
 tomcat1:
  build: .
  ports:
   - "8081:8080"
  container_name: tomcat1
  volumes:
   - ./html1:/opt/apache-tomcat-9.0.106/webapps
 tomcat2:
  build: .
  ports:
   - "8082:8080"
  container_name: tomcat2
  volumes:
 - ./html2:/opt/apache-tomcat-9.0.106/webapps
```

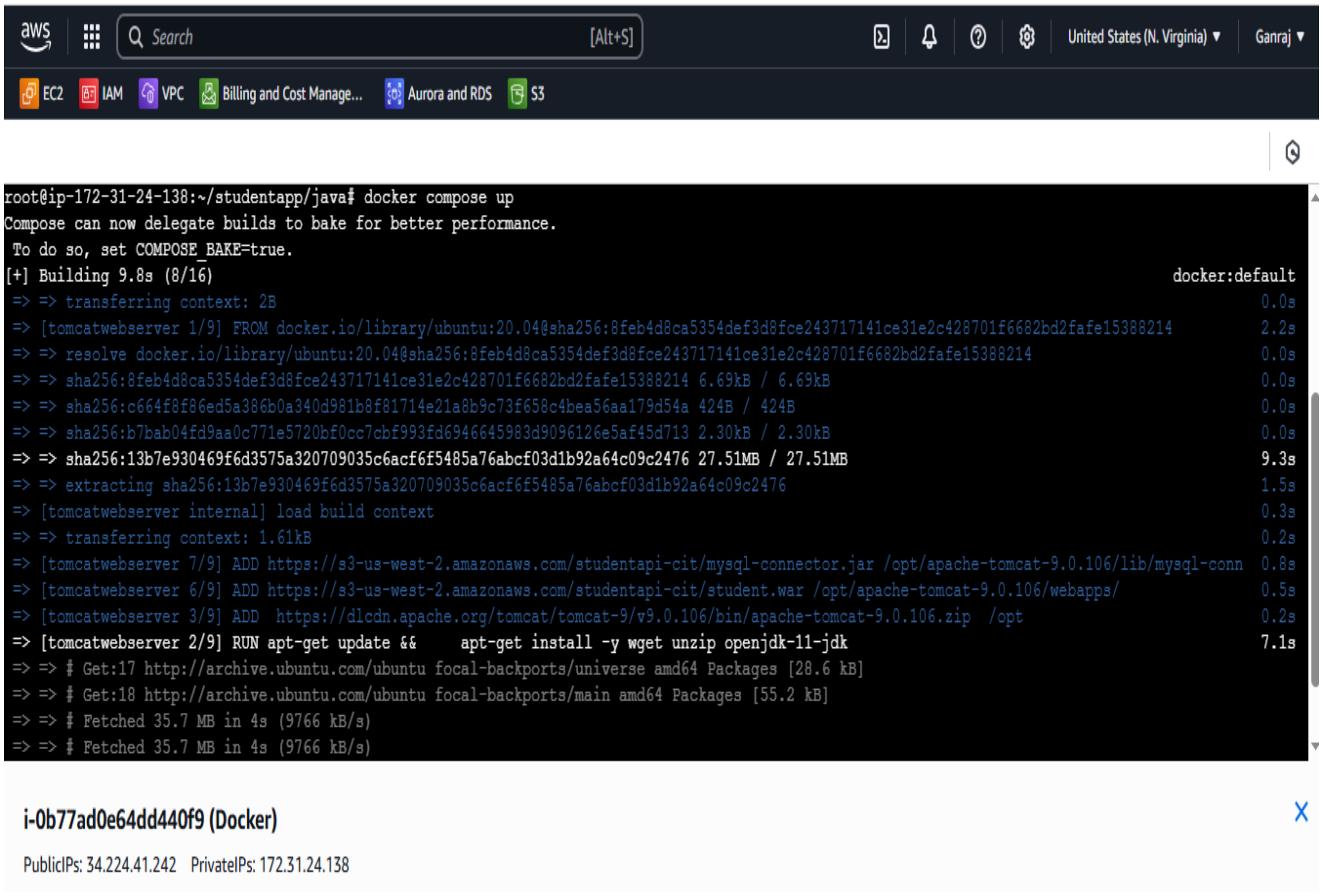
Now we will run this compose file in ec2 server and cloning our github repository to host or build container for our project.

Do the recommended process first do the docker installation on to ubuntu server and clone the repository

When you gone to your directory where as the docker compose file you have to run a command

docker compose up

using this command you are running docker compose but in a foreground of your ubuntu server.As you can see in the below screeshot.



```
root@ip-172-31-24-138:~/studentapp/java# docker compose up
Compose can now delegate builds to bake for better performance.
 To do so, set COMPOSE_BAKE=true.
[+] Building 9.8s (8/16)                                                                                          docker:default
 => => transferring context: 2B                                                                                              0.0s
 => [tomcatwebserver 1/9] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682bd2fafe15388214          2.2s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682bd2fafe15388214                  0.0s
 => => sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682bd2fafe15388214 6.69kB / 6.69kB                                0.0s
 => => sha256:c664f8f86ed5a386b0a340d981b8f81714e21a8b9c73f658c4bea56aa179d54a 424B / 424B                                    0.0s
 => => sha256:b7bab04fd9aa0c771e5720bf0cc7cbf993fd6946645983d9096126e5af45d713 2.30kB / 2.30kB                                0.0s
 => => sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c2476 27.51MB / 27.51MB                              9.3s
 => => extracting sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c2476                                     1.5s
 => [tomcatwebserver internal] load build context                                                                            0.3s
 => => transferring context: 1.61kB                                                                                          0.2s
 => [tomcatwebserver 7/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar /opt/apache-tomcat-9.0.106/lib/mysql-conn  0.8s
 => [tomcatwebserver 6/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war /opt/apache-tomcat-9.0.106/webapps/    0.5s
 => [tomcatwebserver 3/9] ADD  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.106/bin/apache-tomcat-9.0.106.zip  /opt          0.2s
 => [tomcatwebserver 2/9] RUN apt-get update &&      apt-get install -y wget unzip openjdk-11-jdk                             7.1s
 => => # Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [28.6 kB]
 => => # Get:18 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
 => => # Fetched 35.7 MB in 4s (9766 kB/s)
 => => # Fetched 35.7 MB in 4s (9766 kB/s)
```

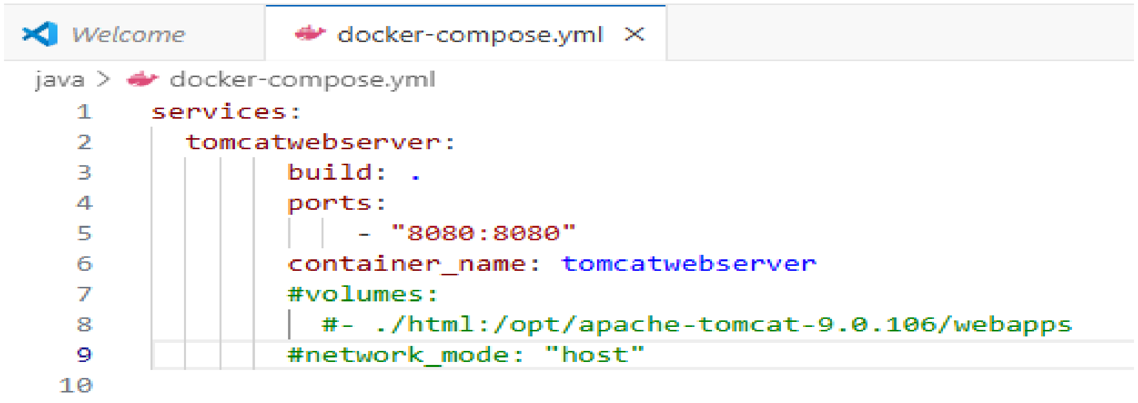
i-0b77ad0e64dd440f9 (Docker)

PublicIPs: 34.224.41.242   PrivateIPs: 172.31.24.138

For running docker compose on background use command

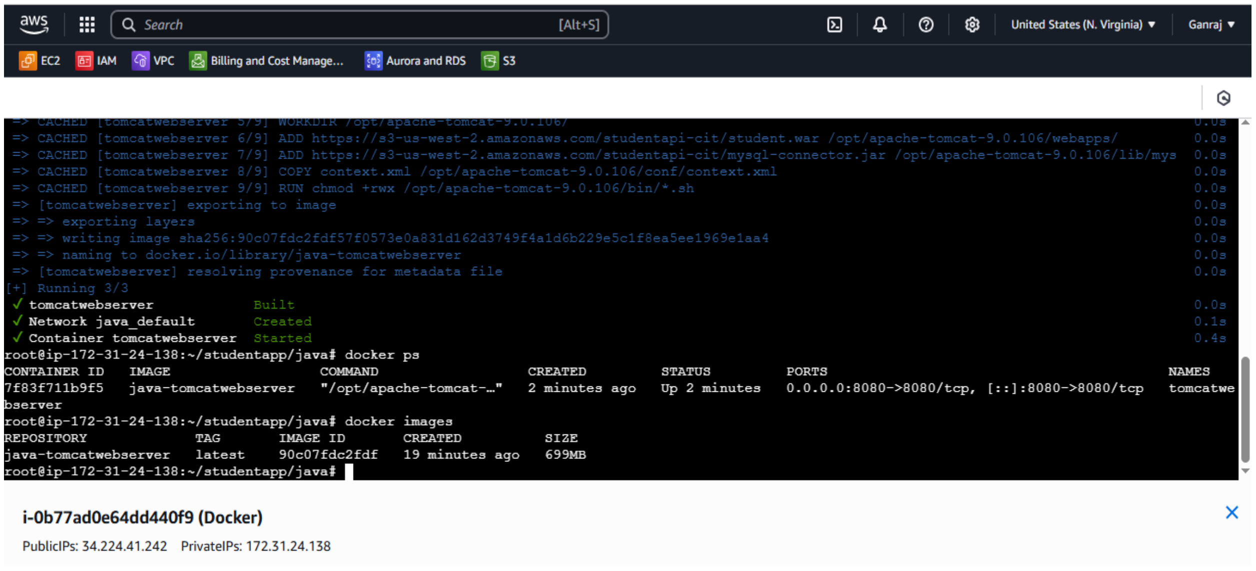
docker compose up -d

Now we will run without network mode and without mounting volume on server.



```yaml
services:
  tomcatwebserver:
     build: .
     ports:
        - "8080:8080"
     container_name: tomcatwebserver
     #volumes:
       #- ./html:/opt/apache-tomcat-9.0.106/webapps
     #network_mode: "host"
```

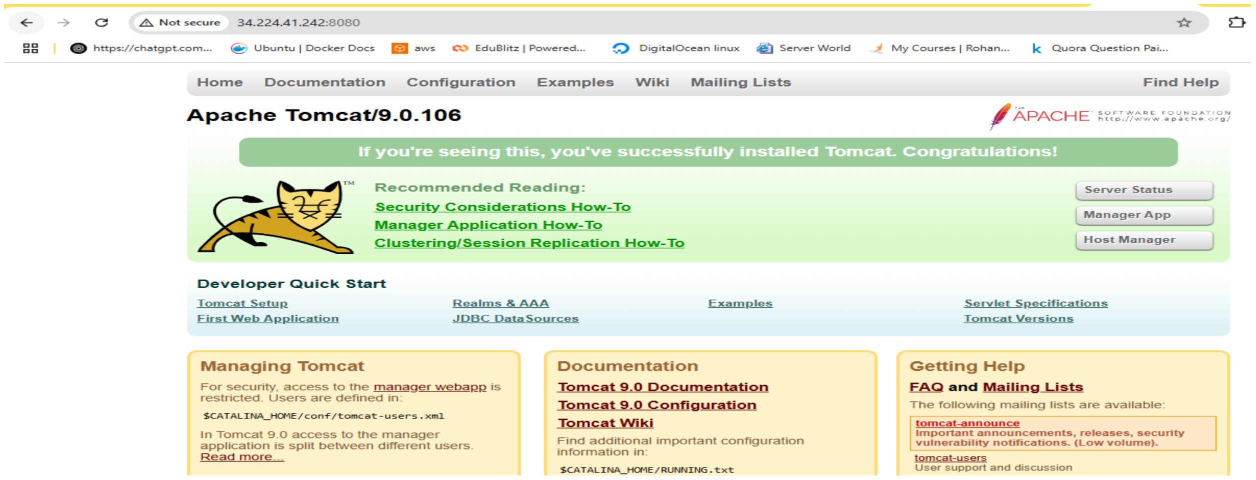
As you can se below snap we have achieved container running using docker compose more like automated.



```
=> CACHED [tomcatwebserver 5/9] WORKDIR /opt/apache-tomcat-9.0.106/                                                        0.0s
=> CACHED [tomcatwebserver 6/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/student.war /opt/apache-tomcat-9.0.106/webapps/      0.0s
=> CACHED [tomcatwebserver 7/9] ADD https://s3-us-west-2.amazonaws.com/studentapi-cit/mysql-connector.jar /opt/apache-tomcat-9.0.106/lib/mys 0.0s
=> CACHED [tomcatwebserver 8/9] COPY context.xml /opt/apache-tomcat-9.0.106/conf/context.xml                               0.0s
=> CACHED [tomcatwebserver 9/9] RUN chmod +rwx /opt/apache-tomcat-9.0.106/bin/*.sh                                         0.0s
=> [tomcatwebserver] exporting to image                                                                                    0.0s
=> => exporting layers                                                                                                     0.0s
=> => writing image sha256:90c07fdc2fdf57f0573e0a831d162d3749f4a1d6b229e5c1f8ea5ee1969e1aa4                                0.0s
=> => naming to docker.io/library/java-tomcatwebserver                                                                     0.0s
=> [tomcatwebserver] resolving provenance for metadata file                                                                0.0s
[+] Running 3/3
 ✓ tomcatwebserver            Built                                                                                        0.0s
 ✓ Network java_default       Created                                                                                      0.1s
 ✓ Container tomcatwebserver  Started                                                                                      0.4s
root@ip-172-31-24-138:~/studentapp/java# docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED         STATUS         PORTS                                       NAMES
7f83f711b9f5   java-tomcatwebserver   "/opt/apache-tomcat-…"   2 minutes ago   Up 2 minutes   0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp   tomcatwe
bserver
root@ip-172-31-24-138:~/studentapp/java# docker images
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
java-tomcatwebserver   latest    90c07fdc2fdf   19 minutes ago   699MB
root@ip-172-31-24-138:~/studentapp/java#
```

i-0b77ad0e64dd440f9 (Docker)

PublicIPs: 34.224.41.242   PrivateIPs: 172.31.24.138



Apache Tomcat/9.0.106

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

$CATALINA_HOME/conf/tomcat-users.xml

In Tomcat 9.0 access to the manager application is split between different users.
Read more...

Documentation

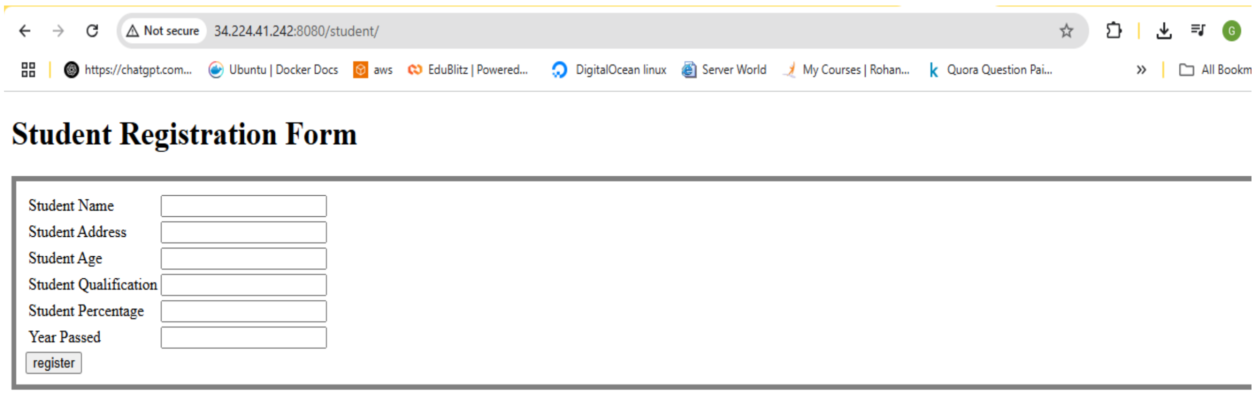
Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki
Find additional important configuration information in:

$CATALINA_HOME/RUNNING.txt

Getting Help

FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion

And Finally our webpage is running . whereas our data (Code Frontend & backend) is stored in S3 bucket.



Student Registration Form

Student Name	
Student Address	
Student Age	
Student Qualification	
Student Percentage	
Year Passed	