Java Application Deployment with Reverse Proxy on AWS

Objective :

The objective of this project is to securely deploy a Java-based Student Registration Web Application on AWS, using open-source tools such as Apache Tomcat, MySQL, and Nginx. The architecture ensures secure access through a reverse proxy, proper backend isolation, and persistent data storage in an Amazon RDS database.

Infrastructure Setup :

EC2 Instances :

- App Server (Ubuntu EC2):

  - Runs Apache Tomcat to host `student.war`.

  - Accessible internally on port `8080`.

- Reverse Proxy Server (Ubuntu EC2):

  - Configured with Nginx as a reverse proxy.

- Publicly accessible on port `80`.

Security Groups :

- App Server EC2 (Backend):

  - Allows only inbound traffic from Proxy EC2 on port 8080

  - Denies all other public access

- Reverse Proxy EC2:

  - Allows public traffic on port 80

  - No access to port 8080 directly from the internet

Application Deployment :

- Installed Apache Tomcat 9 on the App Server

- Deployed the application student.war to the webapps/ directory

- Placed MySQL Connector (mysql-connector.jar) in Tomcat's lib/ directory to enable DB connectivity

- Confirmed the application runs internally at:
  http://localhost:8080/student

Database Setup (Amazon RDS) :

- Engine: MySQL

- Schema: `studentdb`

- Connectivity: RDS instance is not publicly accessible; accessible from App Server only

- Table:

```
CREATE TABLE students (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100),
email VARCHAR(100),
course VARCHAR(100)
);
```

- Verified database connection from the Java application

- Submitted test entries through the web form

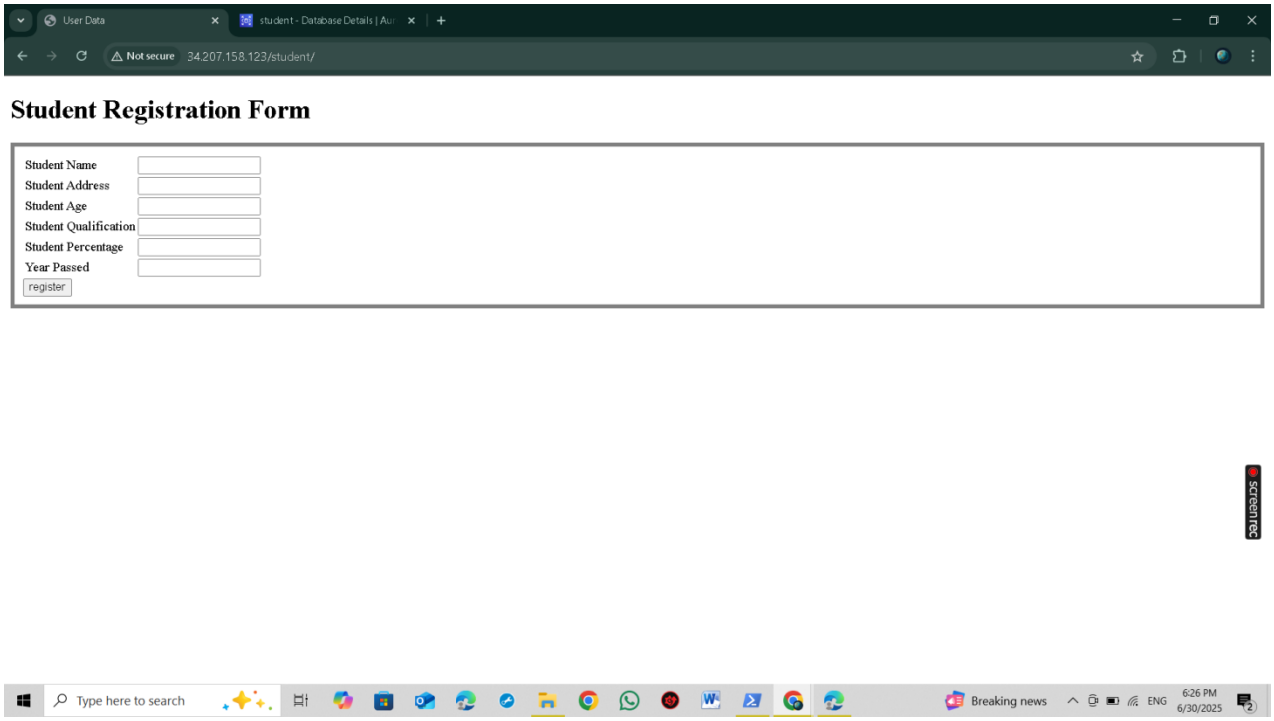
- Checked table using MySQL client and confirmed data insertion

Tools, Services, and Configurations Used :

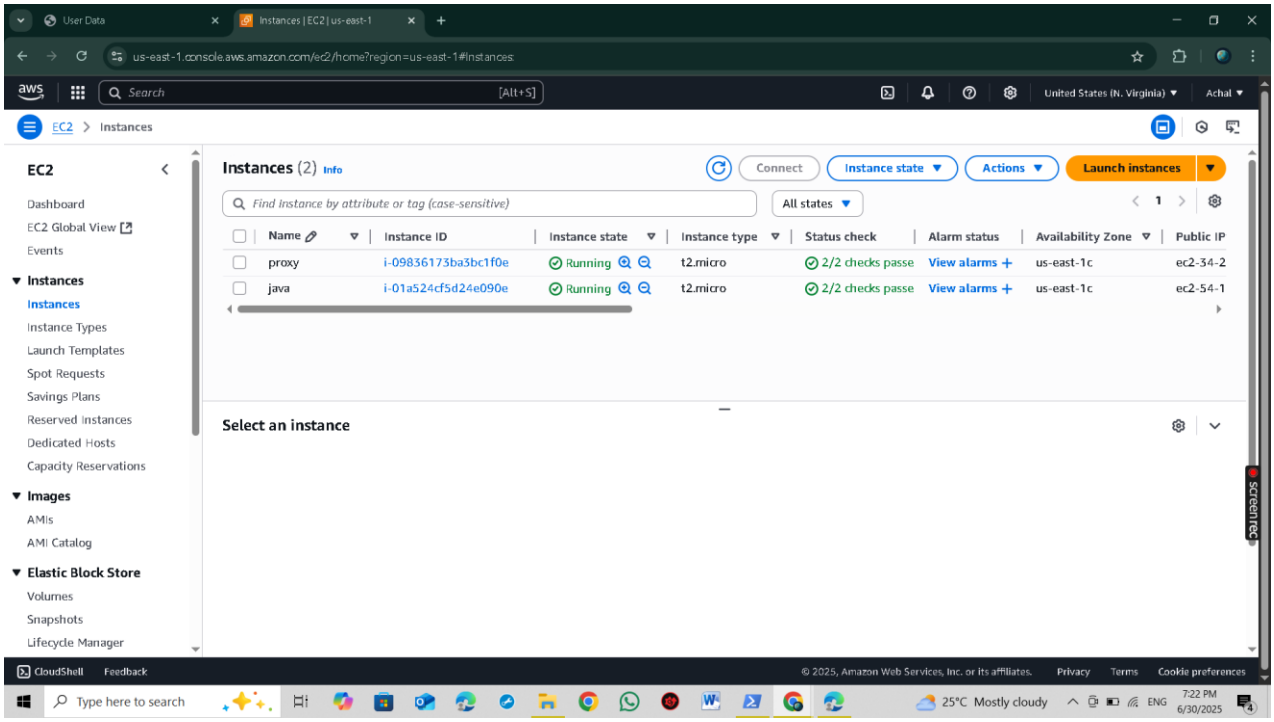
Component	Tool/Service
Web Hosting	Apache Tomcat 9
Application Deployment	student.war
Reverse Proxy	Nginx
Operating System	Ubuntu
DataBase	Amazon RDS (MySQL)
Java DB Connector	Mysql- connector.jar
Cloud Infrastructure	AWS EC2, RDS
Access Control	Security Group

Challenges Encountered & Solutions :

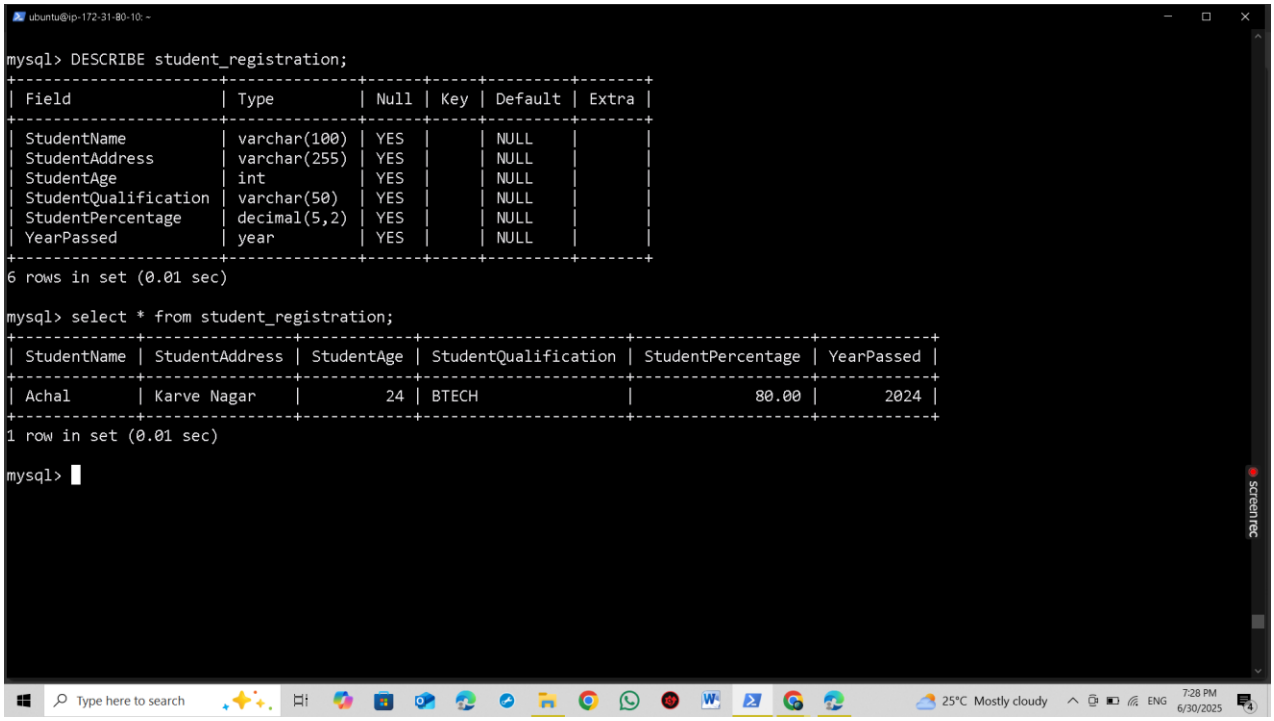
Challenges	Encountered & Solution
Application not connecting to MySQL	Added MySQL Connector JAR to Tomcat's lib/ directory
502 Bad Gateway error on Nginx	Fixed proxy_pass to use internal private IP and ensured App Server was running
Direct access to backend EC2	Applied proper security group rules to public traffic on port 8080



Screenshot: Registration data



Screenshot : Ec2 instance of project and (proxy server)



Screenshot: Database of registration data