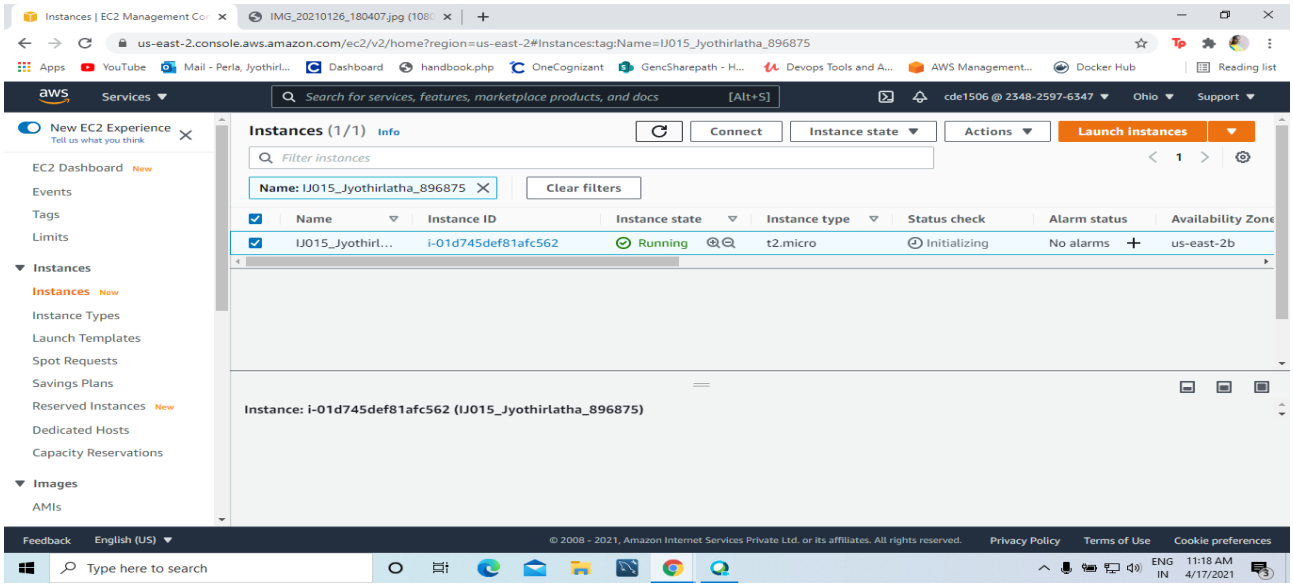
AWS, CI/CD

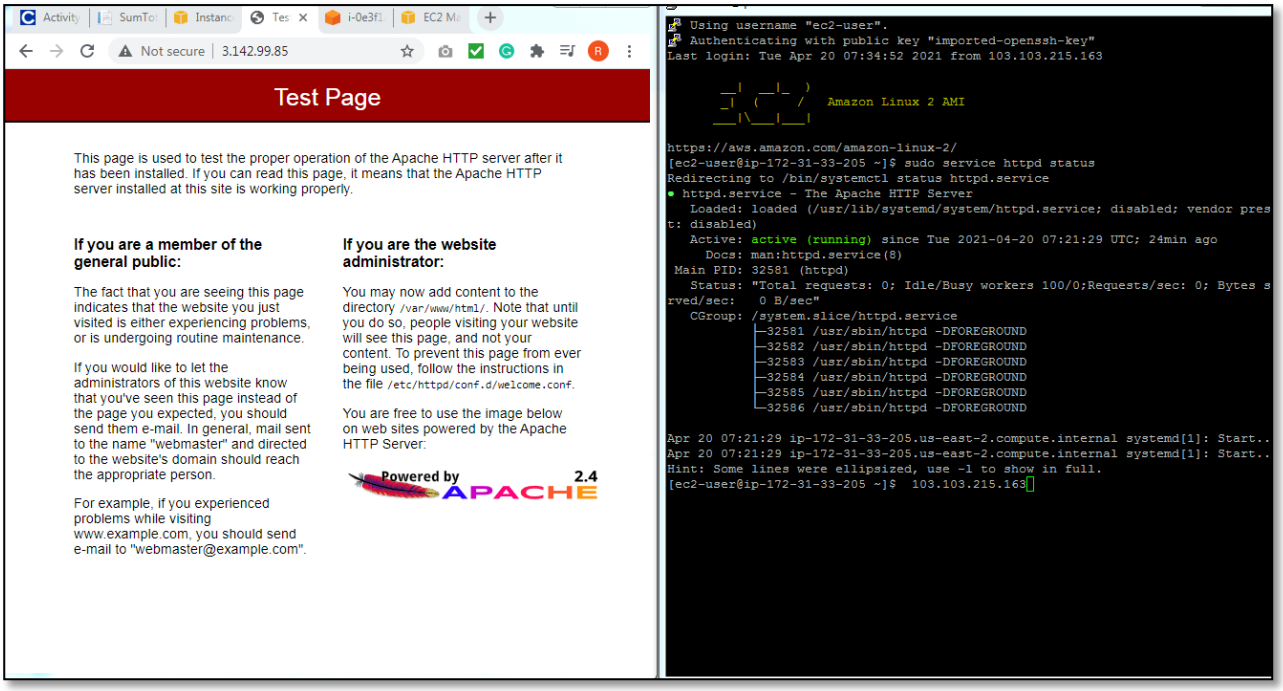
DAY-ONE

▪ [EC2-Hands-on]

Create an EC2 instance, connect to it from your local system and install apache web server on the EC2 instance.



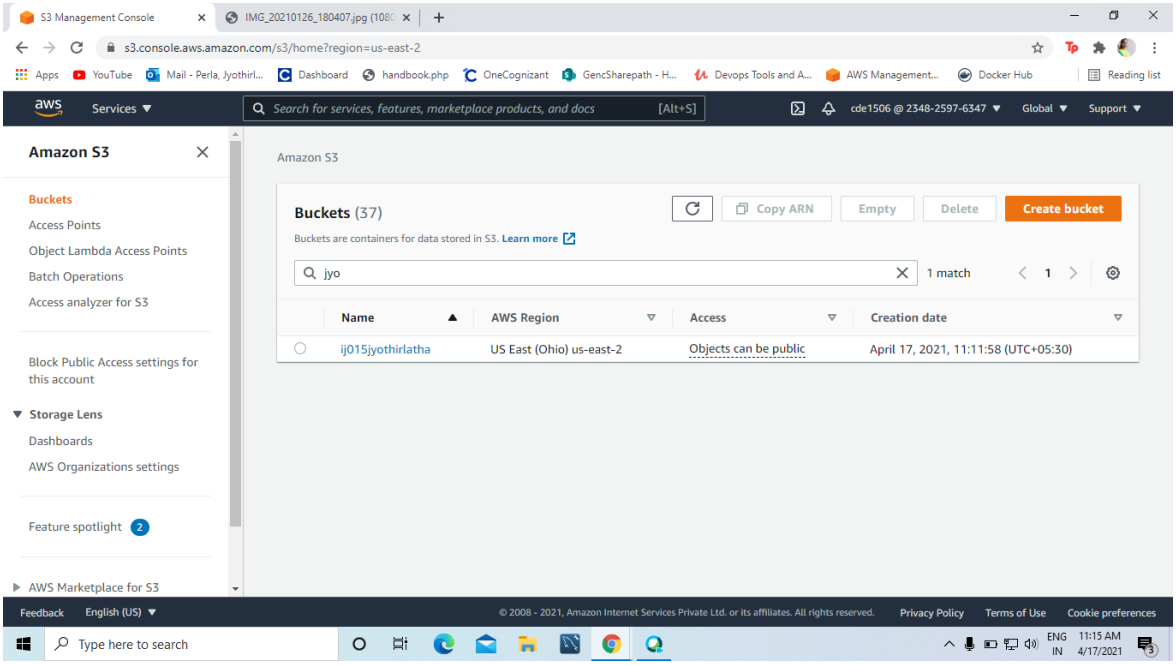
This indicates my instance is up and running



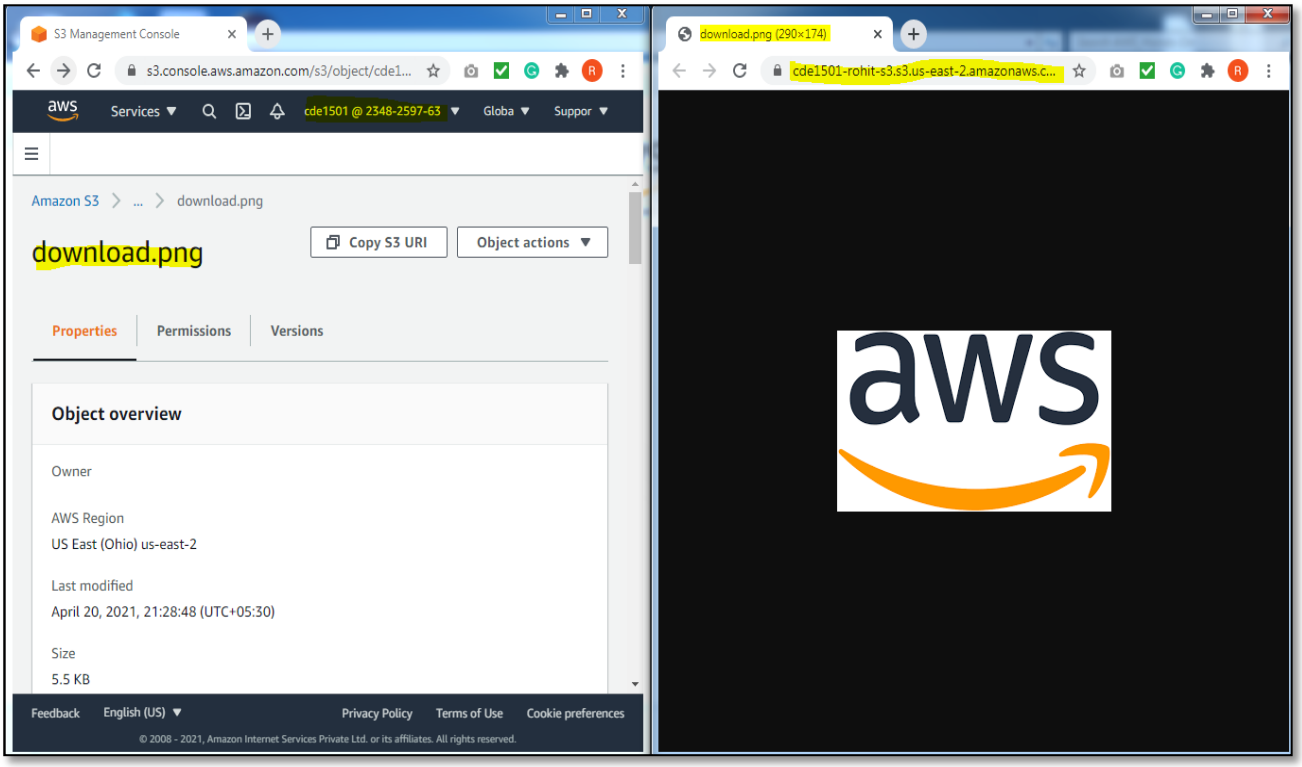
Type the public IPV4 address on the browser url bar and we should get the above shown screen

- **[S3-Hands-on]**

Create a S3 bucket and store an object in it. Enable to object for public access so that anyone can access it through a web browser.



Once successfully S3 created, we will be taken to the above shown

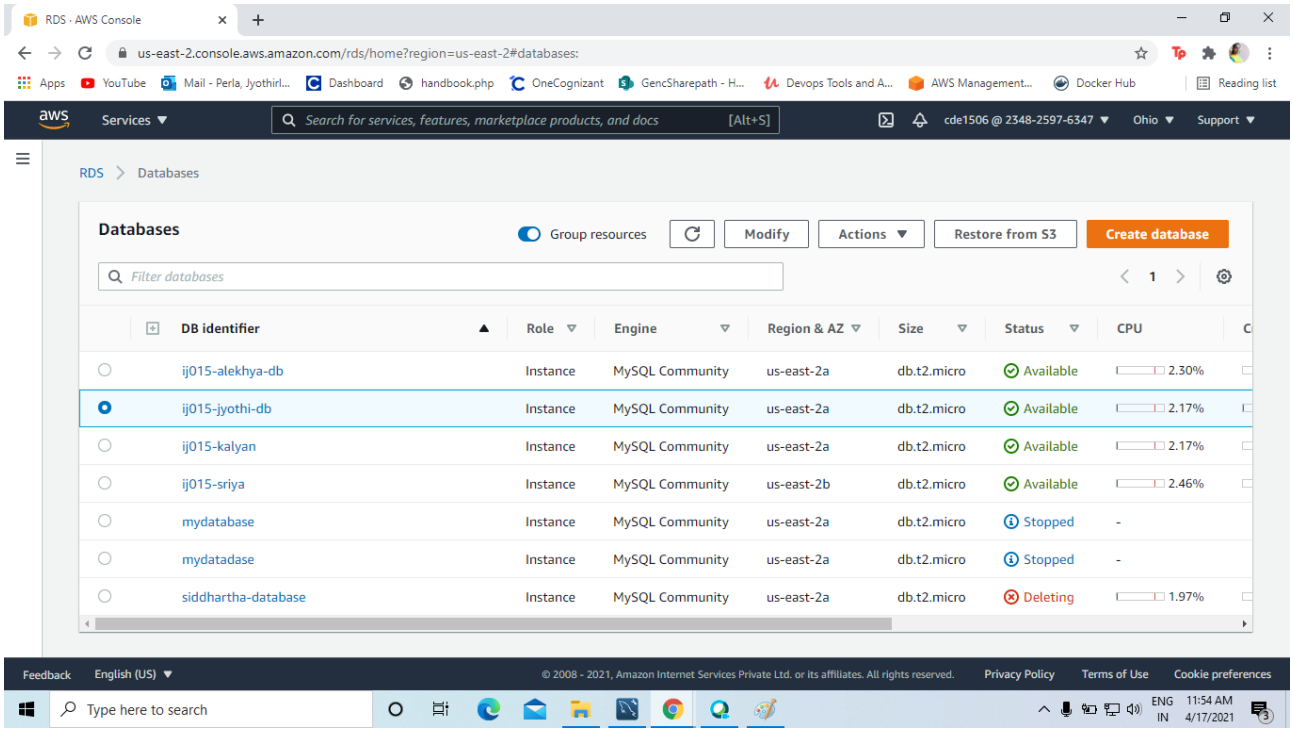


Access the object by clicking in the Object URL and we can view the object in the browser

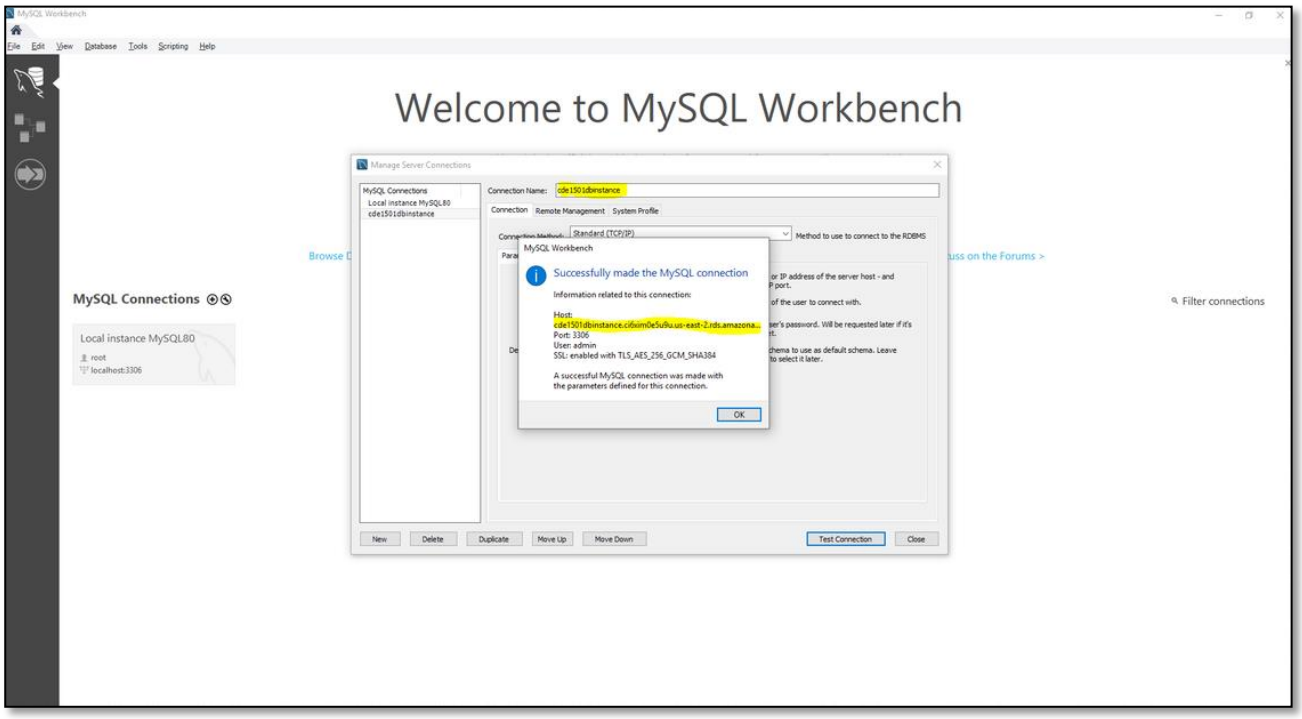
DAY-TWO

- **[RDS-Hands-on]**

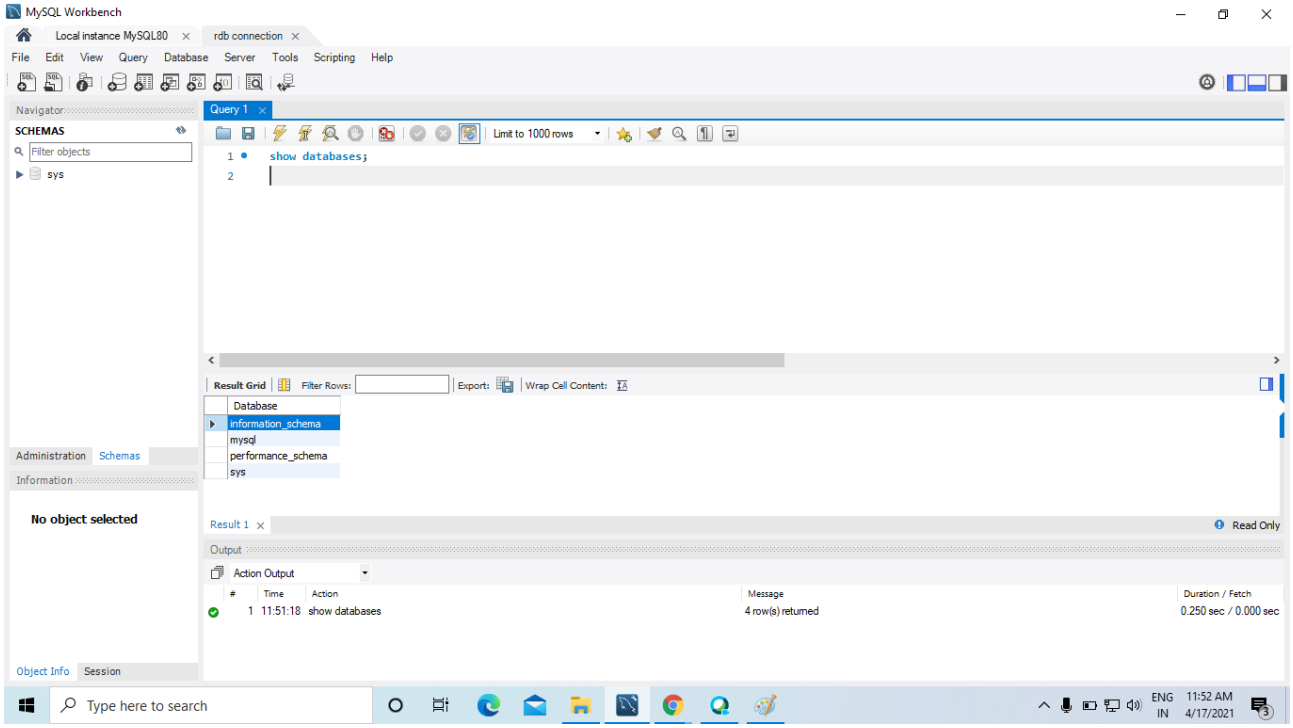
Create a RDS database in AWS and access it through the local client tool.



We can see above screen that our database is being created



We will be connected to the RDS MySQL Server

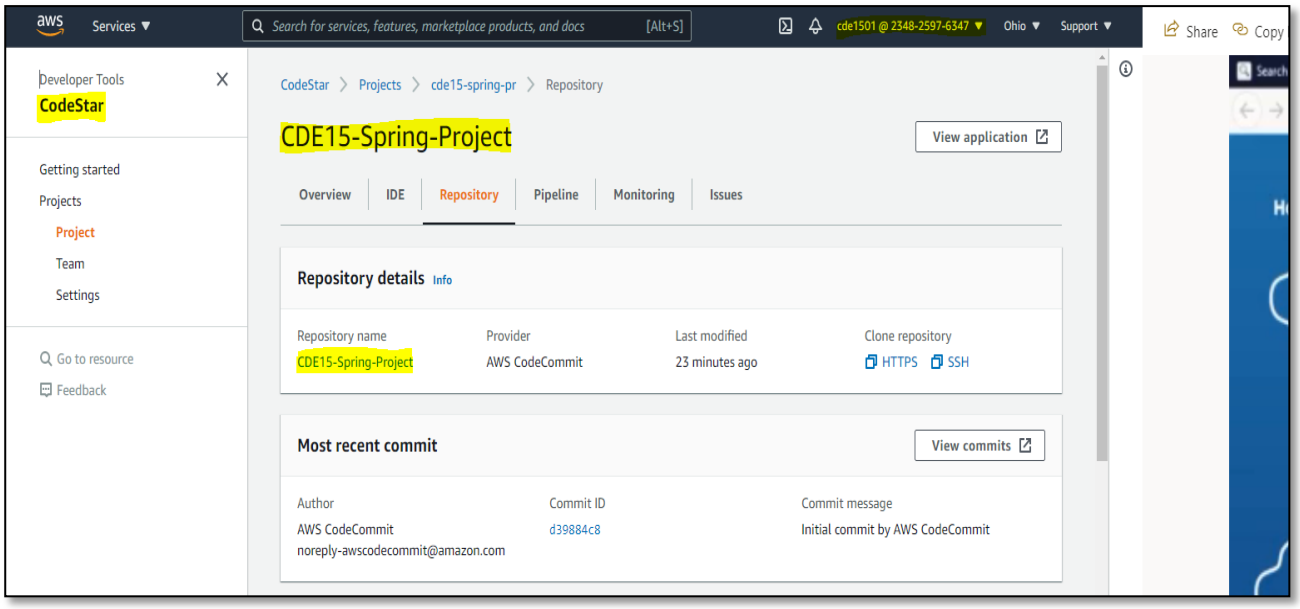


We can see above screen that we are connected to the RDS MySQL server and query window is opened. Here we can issue any sql commands we want to execute against the RDS database

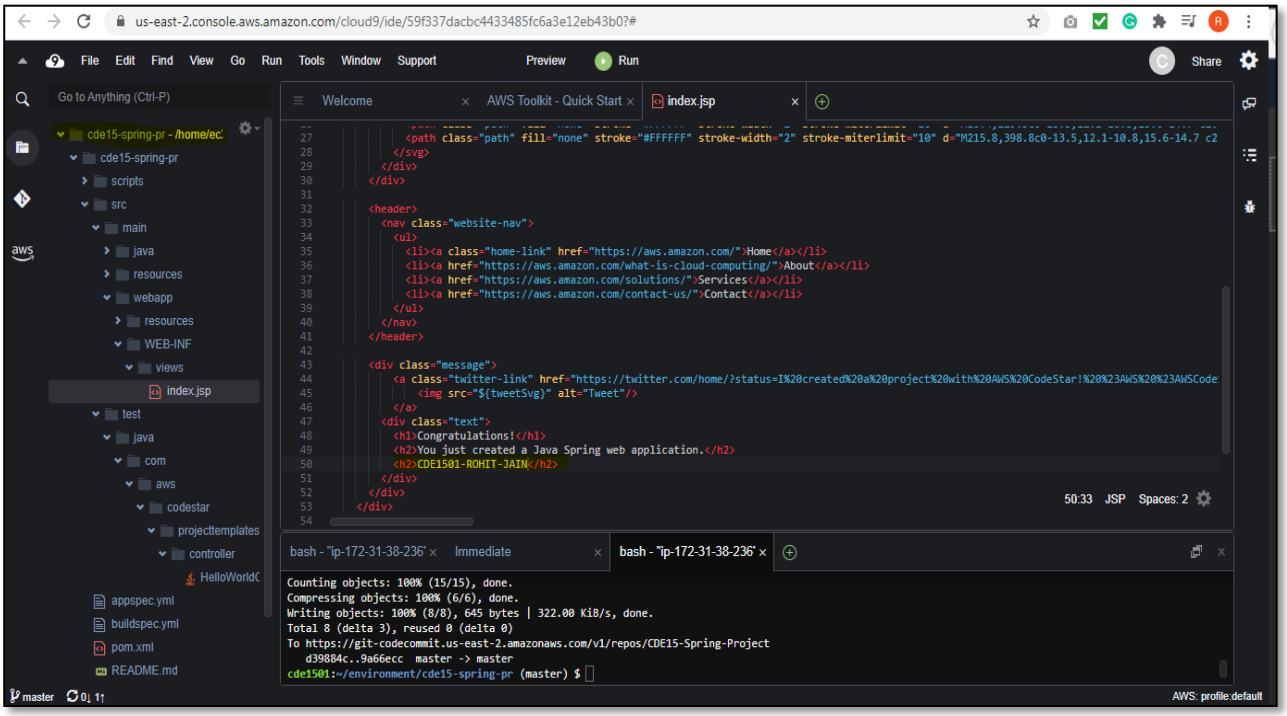
DAY-THREE

- **[CCID-lab-hands-on-practise]**

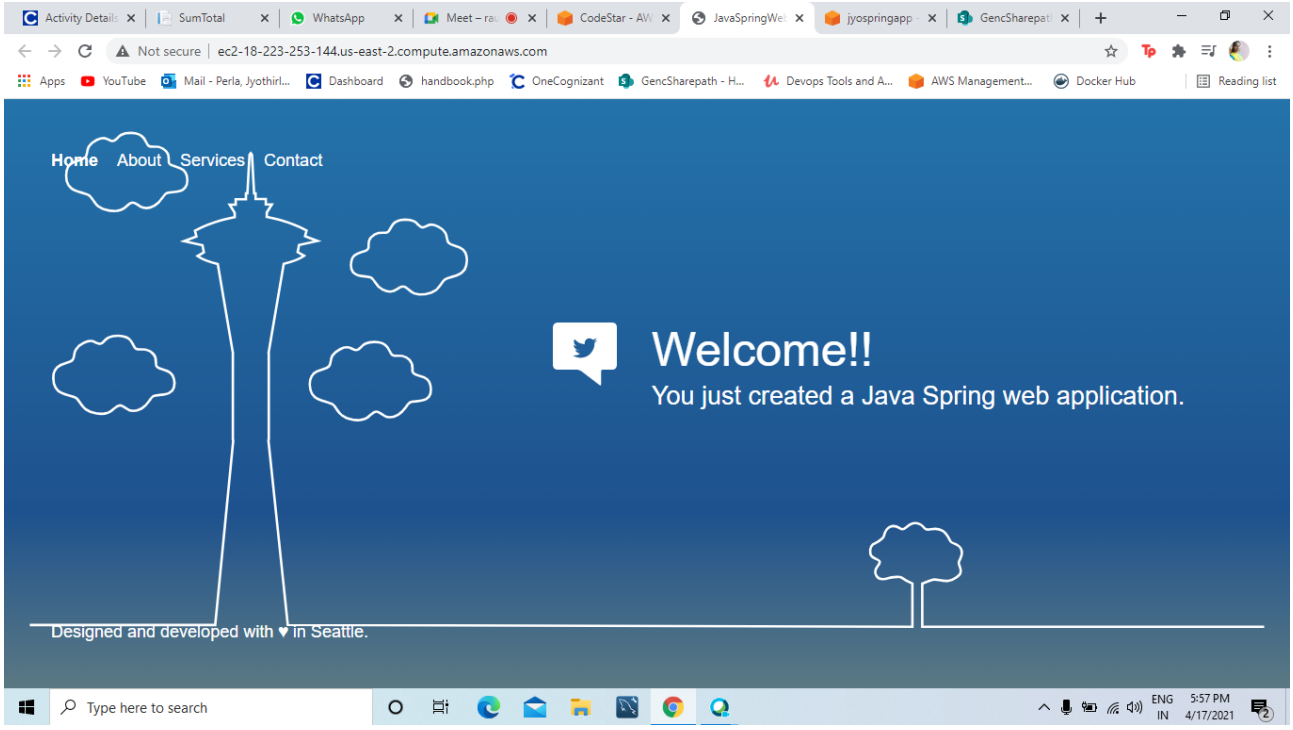
We will able to deploy a spring web application using a Continuous Integration (CI)/ Continuous Delivery (CD) pipeline and the IDE provided by AWS.



We can see above screen that SPRING PROJECT is created



AWS IDE

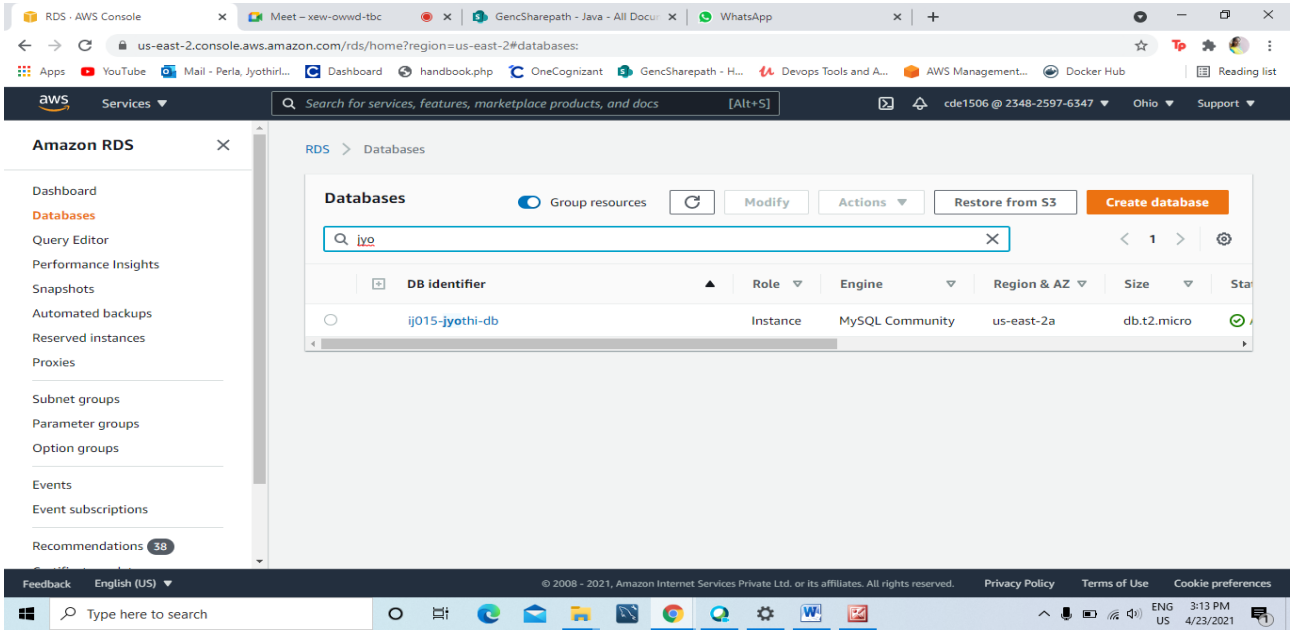


Successfully deploy a spring web application using CI/CD

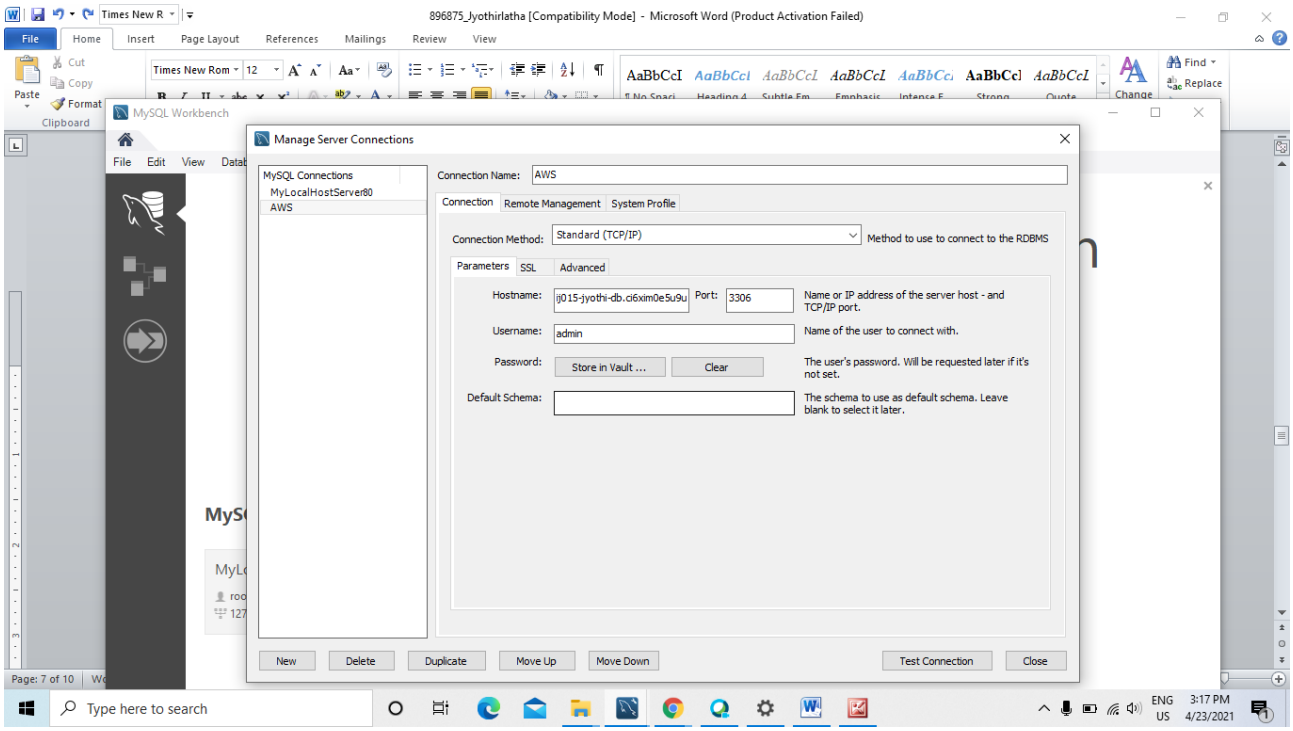
DAY-FOUR

- **[Spring-REST-with-RDS-Backend]**

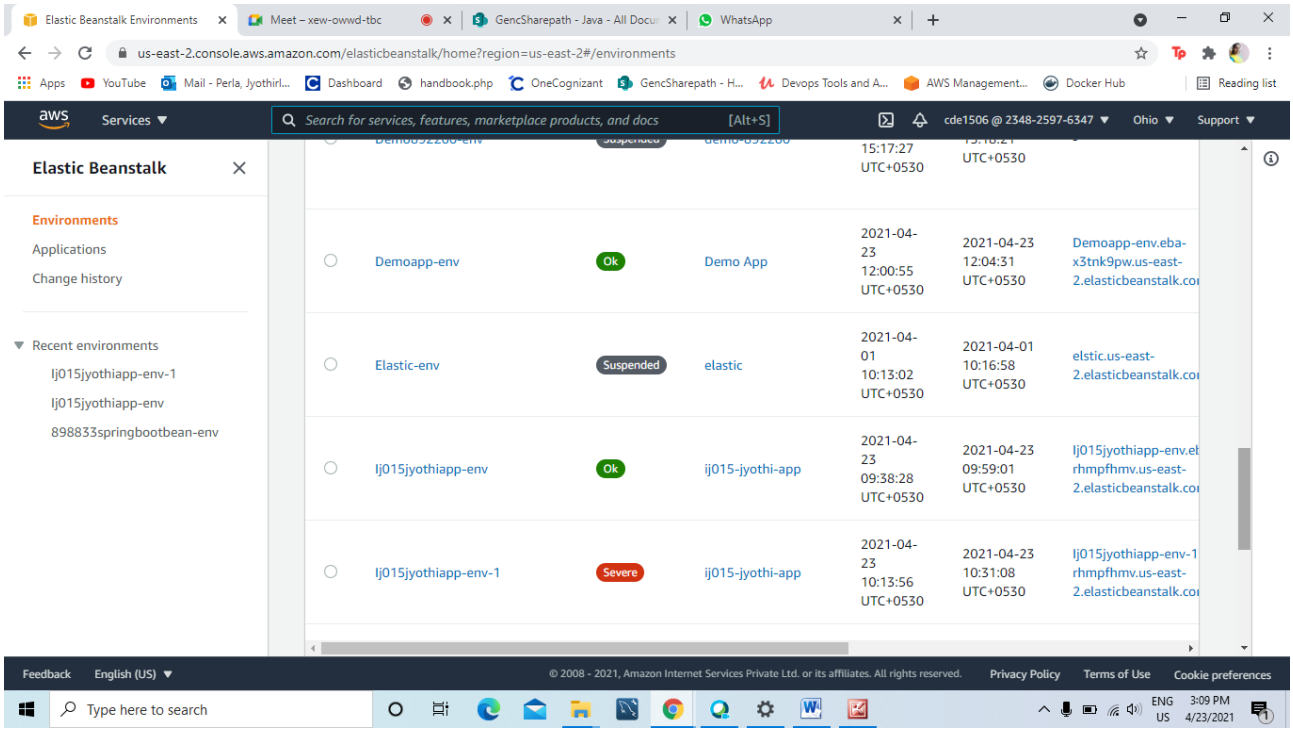
Create a Spring REST application that perform Read and Insert operation on RDS database. Deploy the application in AWS Elastic Beanstalk and access the application from anywhere.



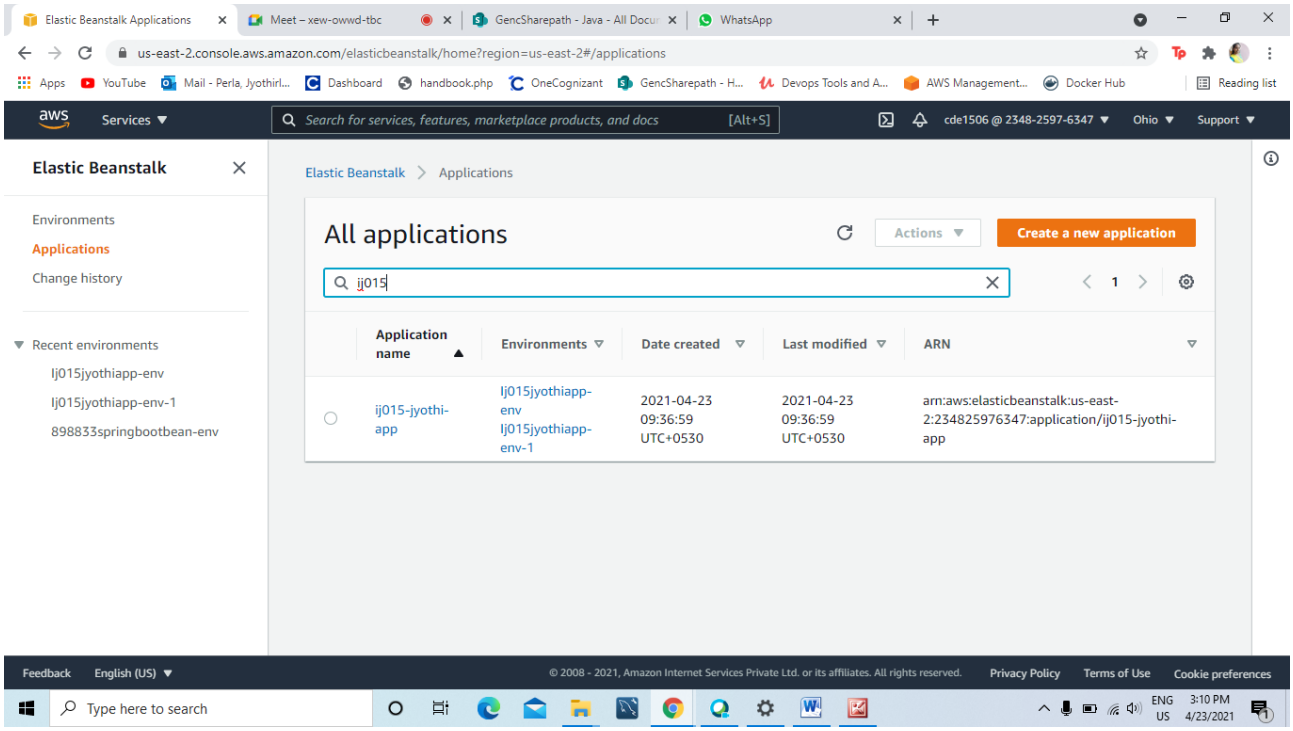
We can see above screen that our database is being created



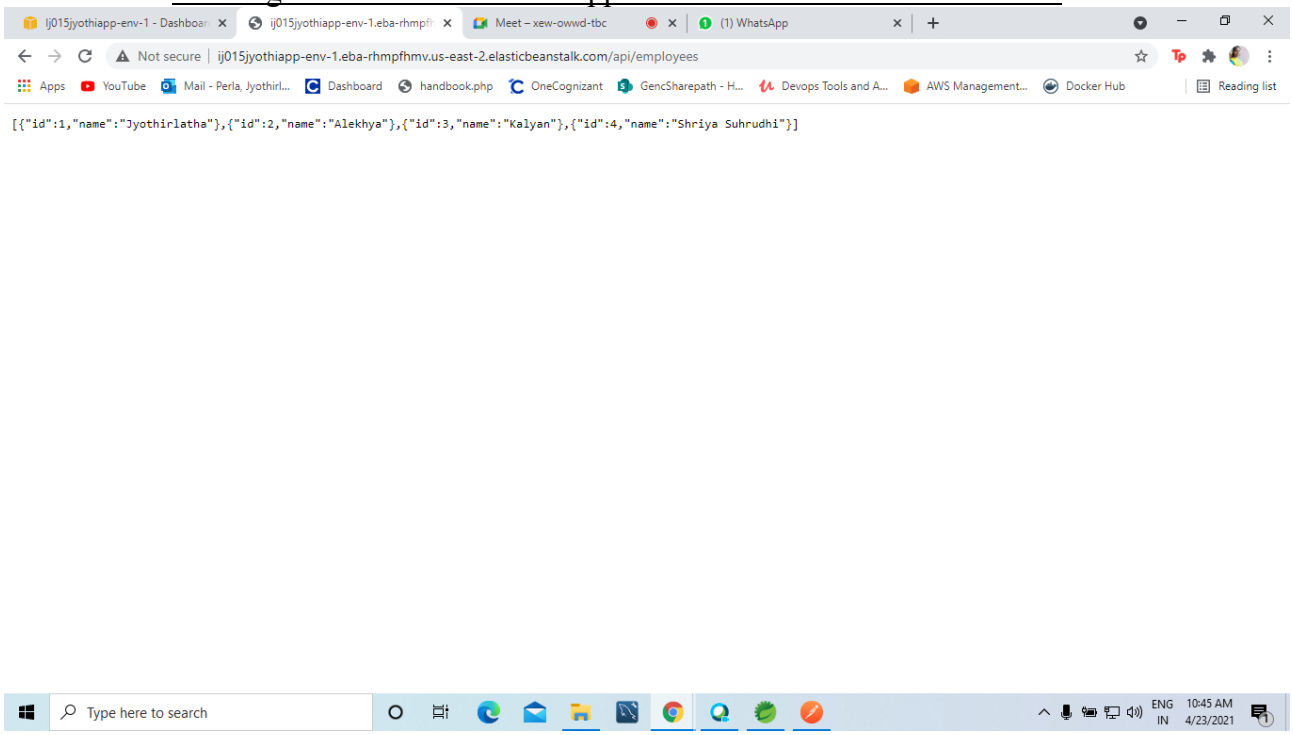
We will be connected to the RDS MySQL Server



Creating the Elastic Beanstalk environment shown in the screen above



Creating the Elastic Beanstalk Application shown in the screen above

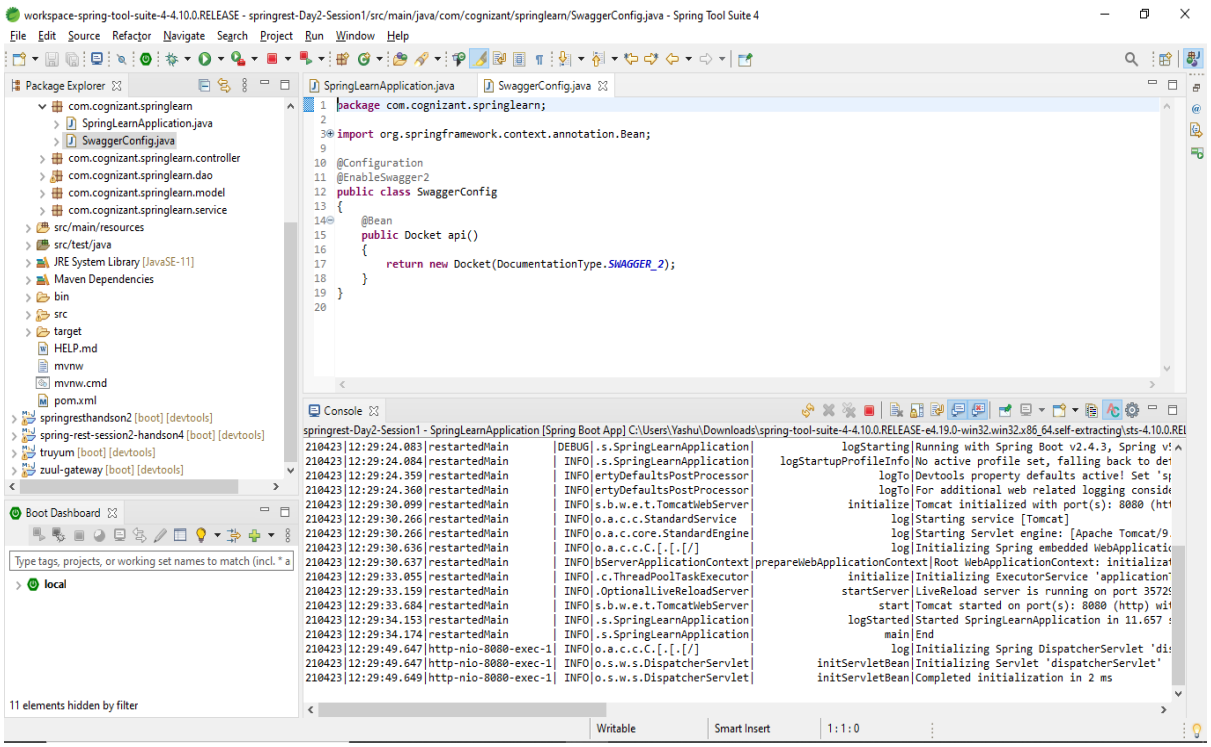


Start the application locally and type the url in the browser window as shown above

DAY-FIVE

- **[Swagger]**

Make use of Swagger to create documentation for RESTful / microservices

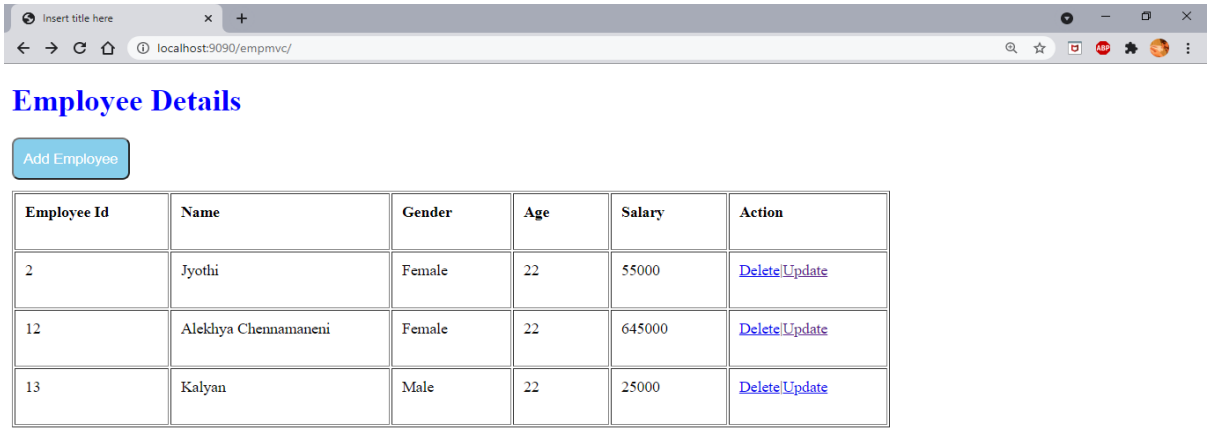


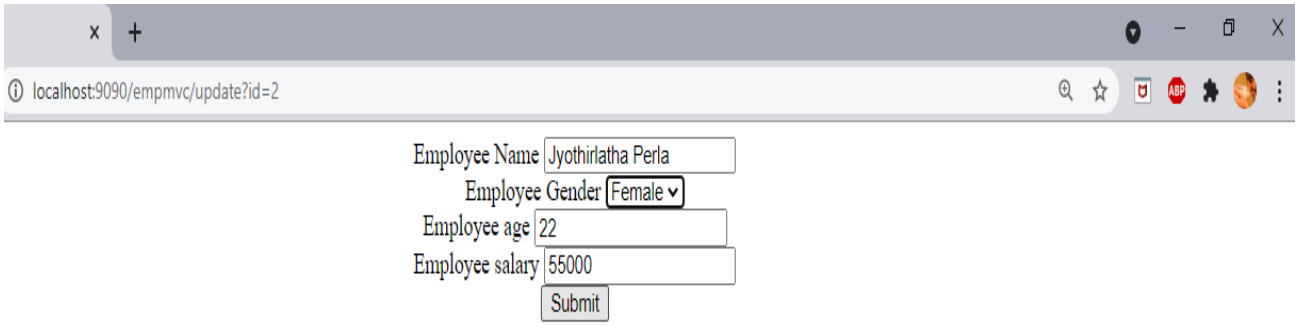
Included the Swagger dependencies & create a Swagger configuration class as shown below.

api endpoint “localhost:8080/v2/api-docs” and complete API documentation of your service

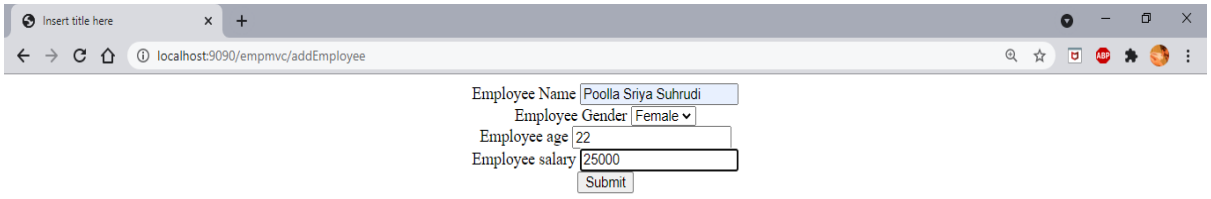
DAY – SIX

- **[Spring MVC Client for Spring REST Service]**


Insert title here
localhost:9090/empmvc/
Employee Details
Add Employee
Employee Id
Name
Gender
Age
Salary
Action
2
Jyothi
Female
22
55000
Delete|Update
12
Alekhya Chennamaneni
Female
22
645000
Delete|Update
13
Kalyan
Male
22
25000
Delete|Update


localhost:9090/empmvc/update?id=2
Employee Name Jyothirlatha Perla
Employee Gender Female
Employee age 22
Employee salary 55000
Submit


Insert title here
localhost:9090/empmvc/update?id=2
Updated Successfully


Insert title here
localhost:9090/empmvc/addEmployee
Employee Name Poolla Sriya Suhrudi
Employee Gender Female
Employee age 22
Employee salary 25000
Submit

Employee Added SuccessFully

Employee Details

Add Employee

Employee Id	Name	Gender	Age	Salary	Action
2	Jyothirlatha Perla	Female	22	55000	Delete|Update
12	Alekhya Chennamaneni	Female	22	645000	Delete|Update
13	Kalyan	Male	22	25000	Delete|Update
15	Poolla Sriya Suhrudi	Female	22	25000	Delete|Update

Employee deleted Successfully