**Implementation of REST API using Spring Boot –**

1. **I have used two tables – Student and Course**

|  |
| --- |
| **Student\_table** |
| Student\_id |
| Student\_name |
| Student\_age |
| Course\_id |

|  |
| --- |
| **Course\_table** |
| Course\_name |
| Course\_Duration |

1. **I have used MySQL Workbench for database**

Following code I have added in my application.properties file

spring.datasource.username=root

spring.datasource.password=Jazz@123

spring.datasource.url=jdbc:mysql://localhost:3306/demo

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

1. **I have also used spring security**

Following is the dependency we need to add in pom.xml

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

1. For main login – use id: admin1 – password: secret1
2. For Postman login – hit localhost:8080/student or course

use id:user1 – password: secret1

1. **I also implemented dockerizing war file**

Following is my docker file content

From tomcat:8.0.51-jre8-alpine

RUN rm -rf /usr/local/tomcat/webapps/\*

COPY ./target/student-management-0.0.1-SNAPSHOT.war /usr/local/tomcat/webapps/ROOT.war

CMD ["catalina.sh","run"]

Commands to be used:

* To build an image in the docker

#$ docker image build -t student-management .

* To run in the container

#$ docker container run -p 8080:8080 -d student-management

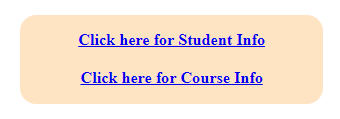
* Container Id will be generated, it is executed using following command

#$ docker container exec -it {container-id} /bin/sh

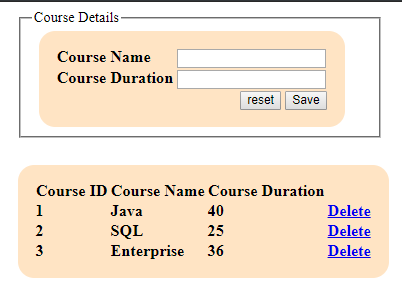
#/usr/local/tomcat # cd webapps/

#/usr/local/tomcat/webapps # ls

1. **First Page – home.jsp**

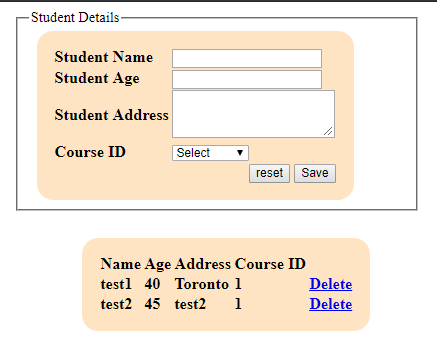


1. **To add courses to the database – click the link to add Course Info, it will redirect to courses.jsp**



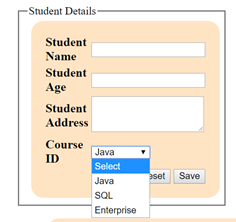
Since data is dummy data is added into my database, data is getting displayed at the bottom

1. **By clicking on “Delete” link, one can delete the record from the database and page will be refreshed**
   1. It will be using deleteById feature which JPA provides
2. **To add Student to the database – click the link to add Student Info, it will redirect to student.jsp**



Since data is dummy data is added into my database, data is getting displayed at the bottom

1. **In the Course ID drop-down, I am fetching data from the database**



Currently, there 3 records are added in the course table

Note: If there are no records are added in the course\_table, data will be returned as null

1. **By clicking on “Delete” link, one can delete the record from the database and page will be refreshed**
   1. It will be using deleteById feature which JPA provides