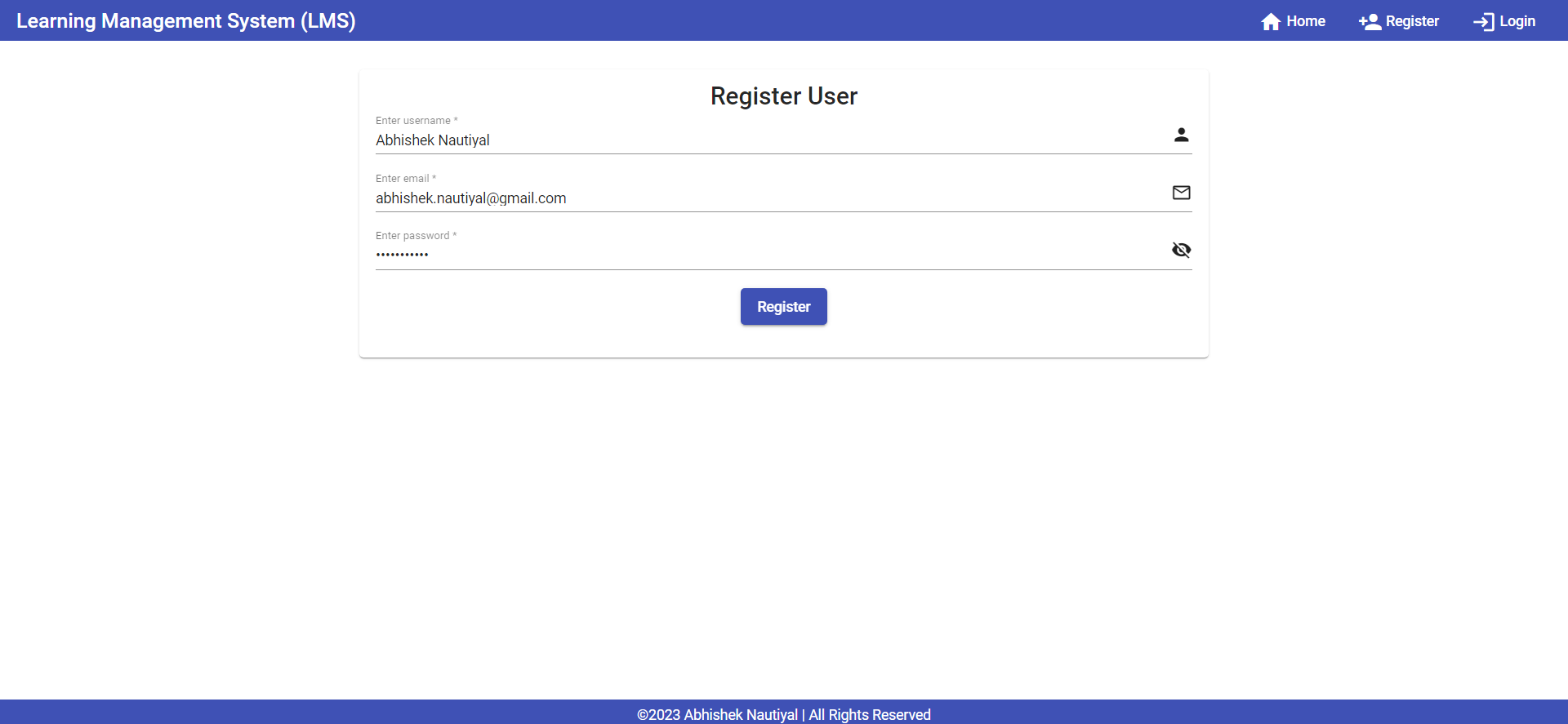
**Project – Learning Management System (LMS)**

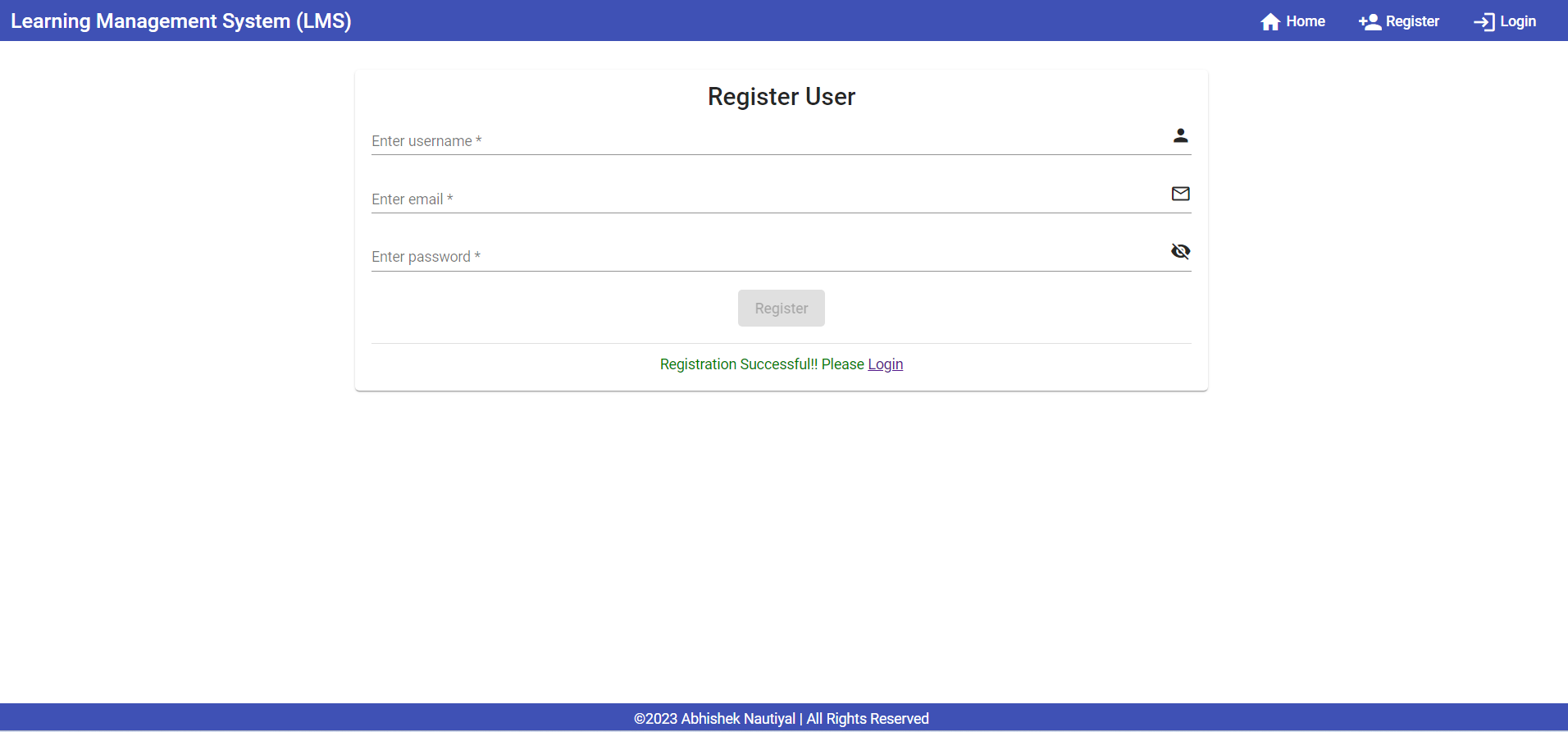
**Front End –**

Below is the UI where system is asking user either to Register or Login, Material Angular is used for component building. JWT is used for authentication and token is stored in local storage.

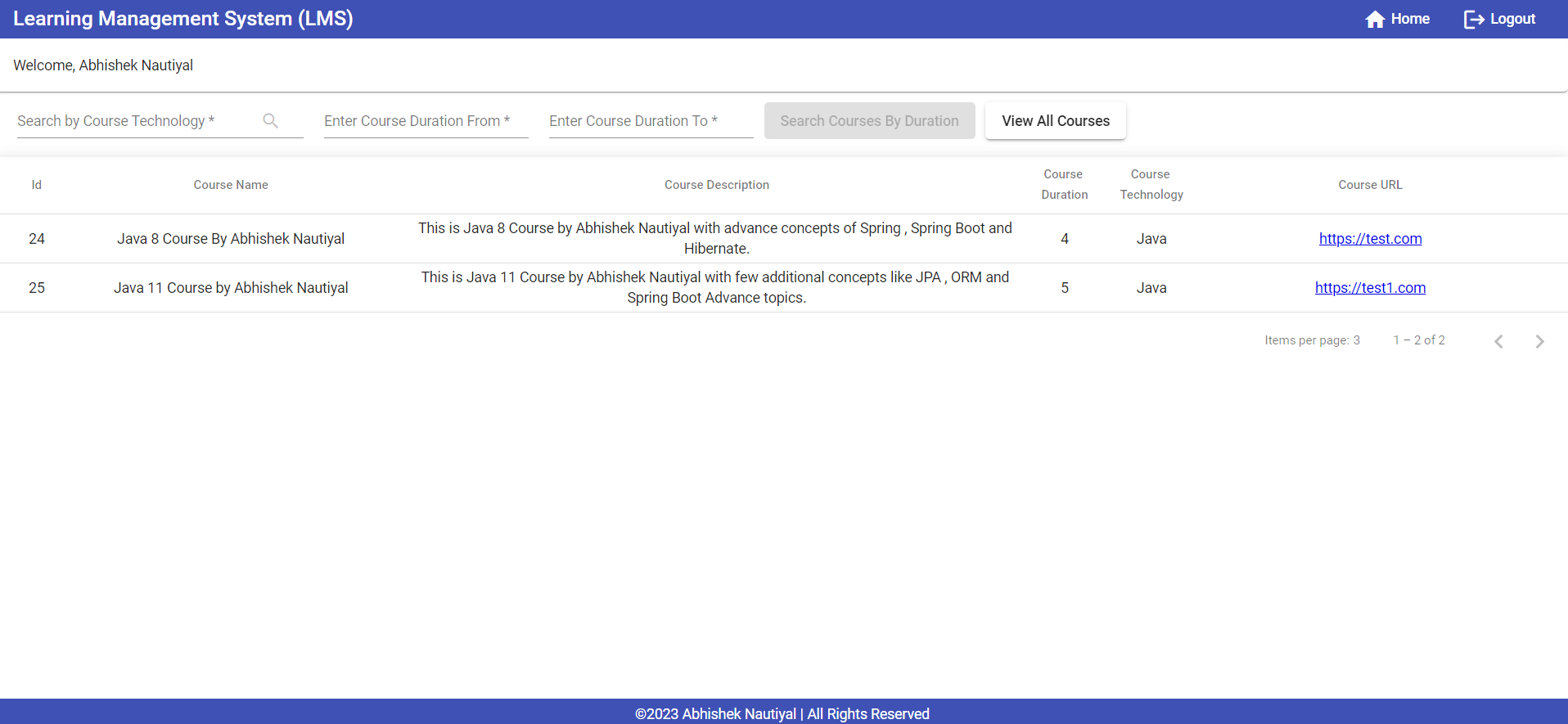
Below is the User Registration Page -



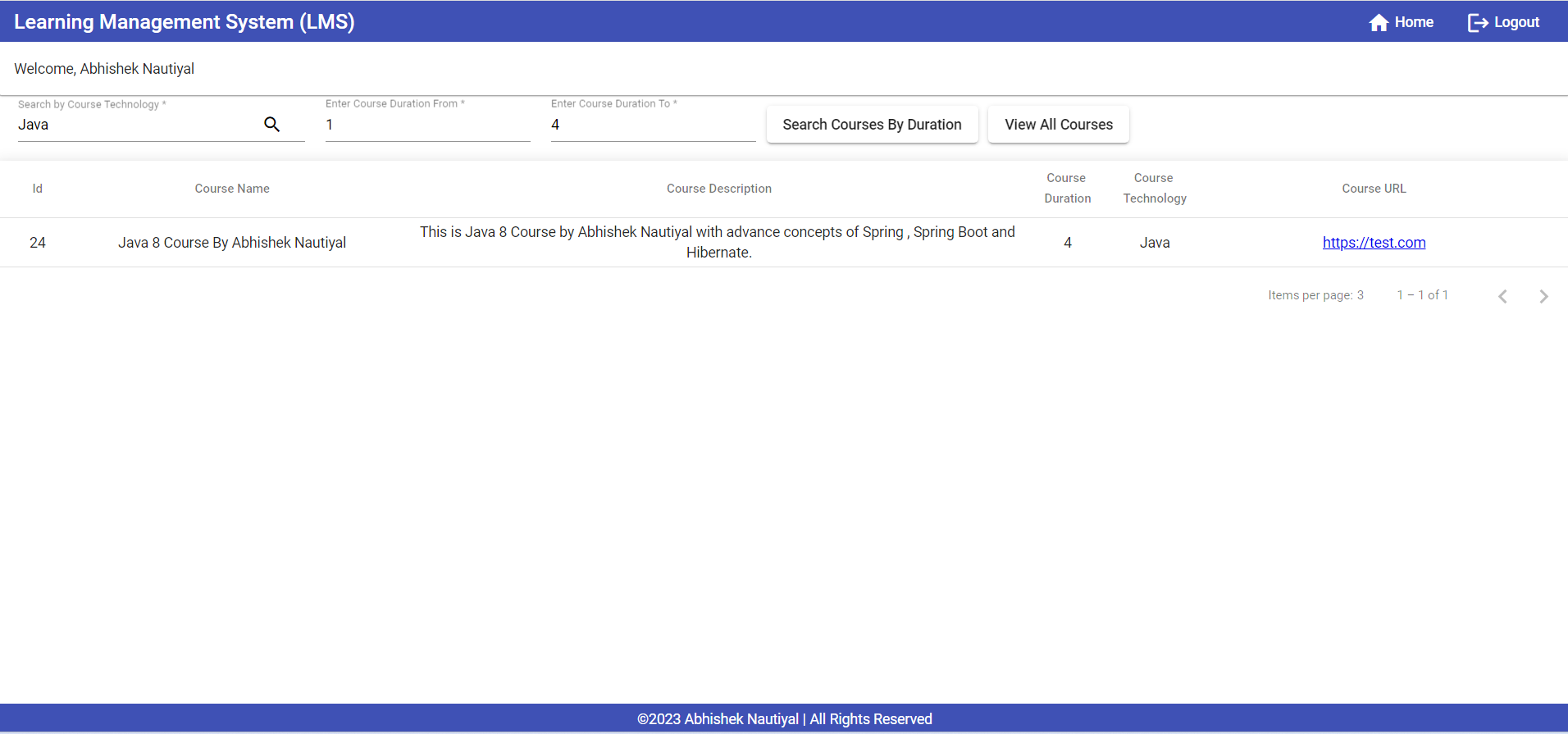
On successful registration of User, one can login and access the Home Page –



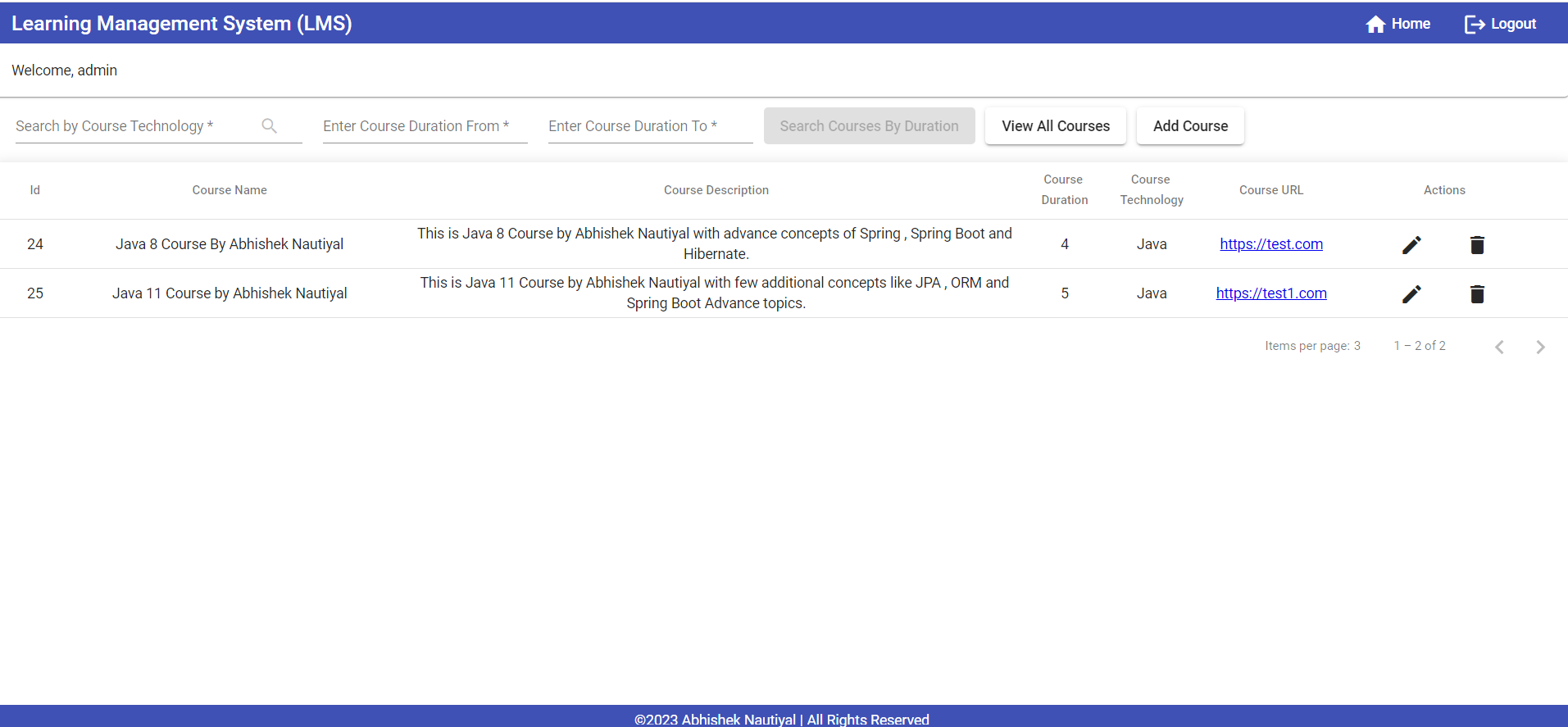
When user login below is the home page where user can search courses based on Course Technology or Based on Course Technology along with Course Duration –



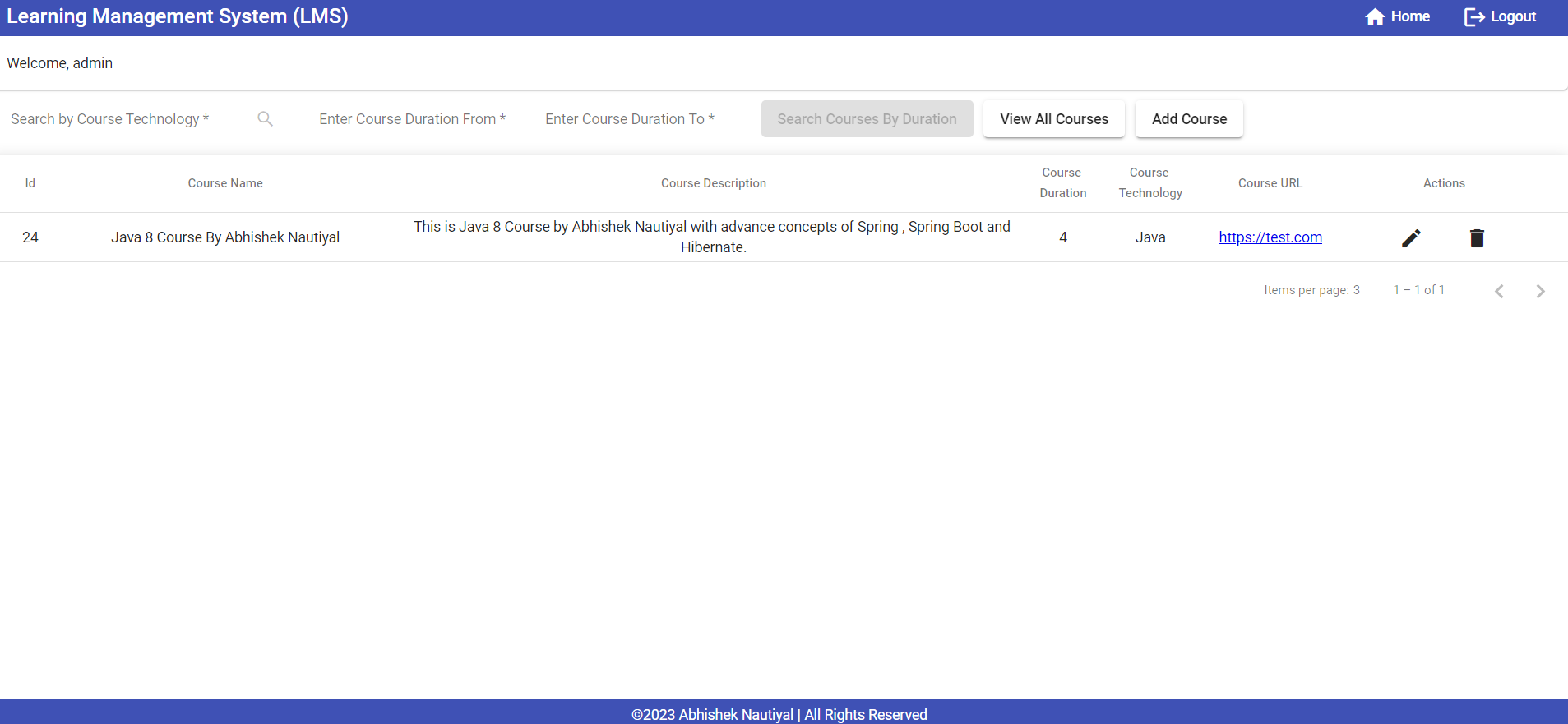
Search functionality based on Course Technology and Course Duration –



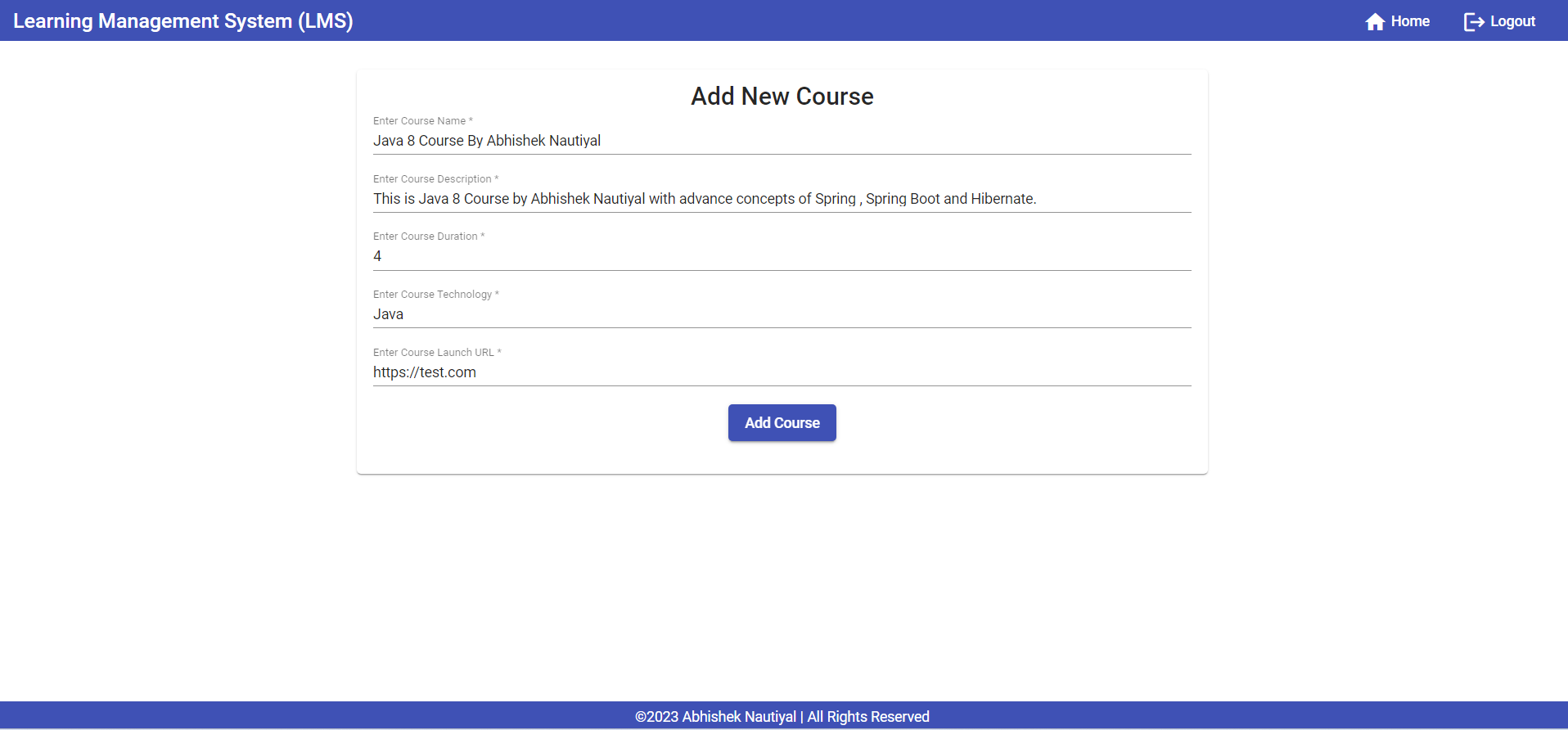
When user with Admin Role login below is the Screen – Admin can perform add, edit and delete operations along with search.



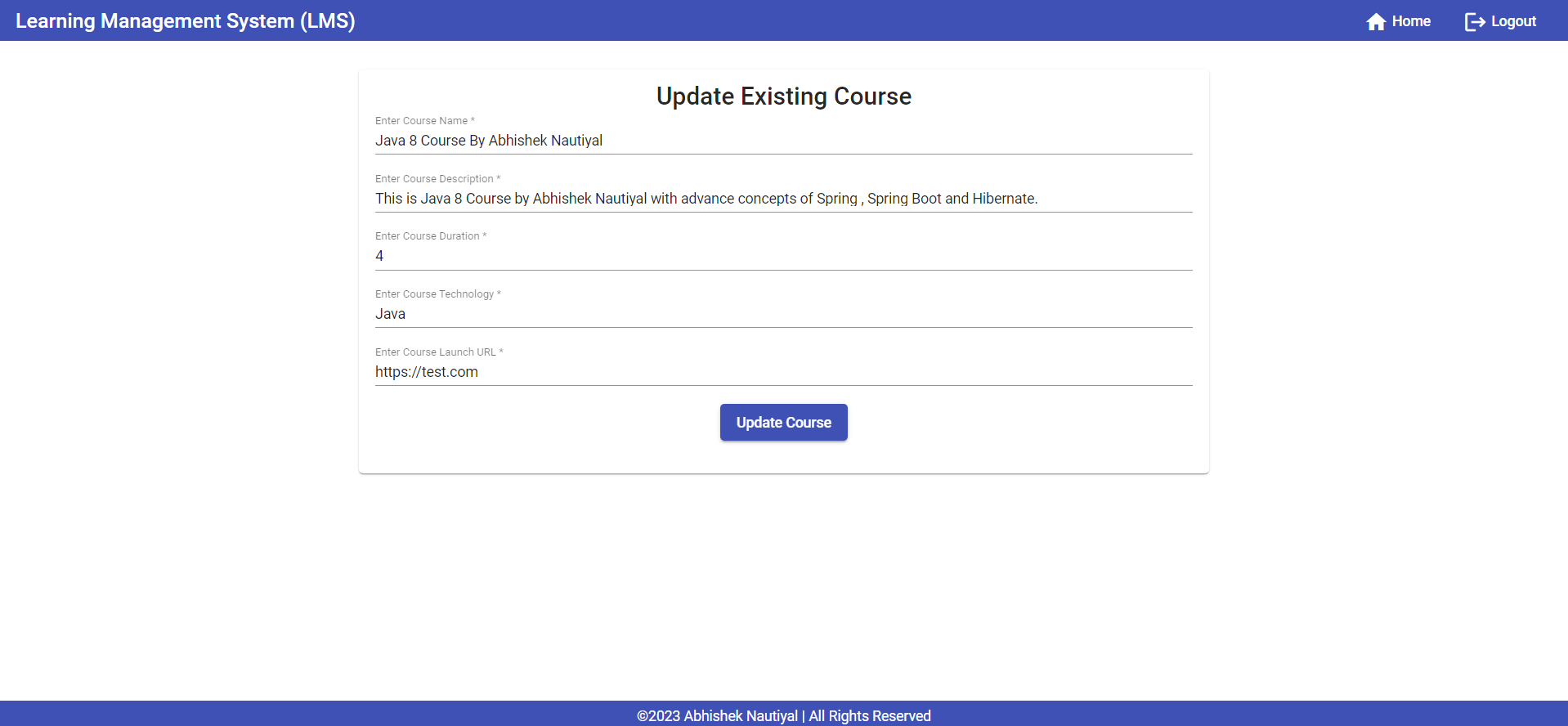
Functionality when admin click on delete icon to delete course -



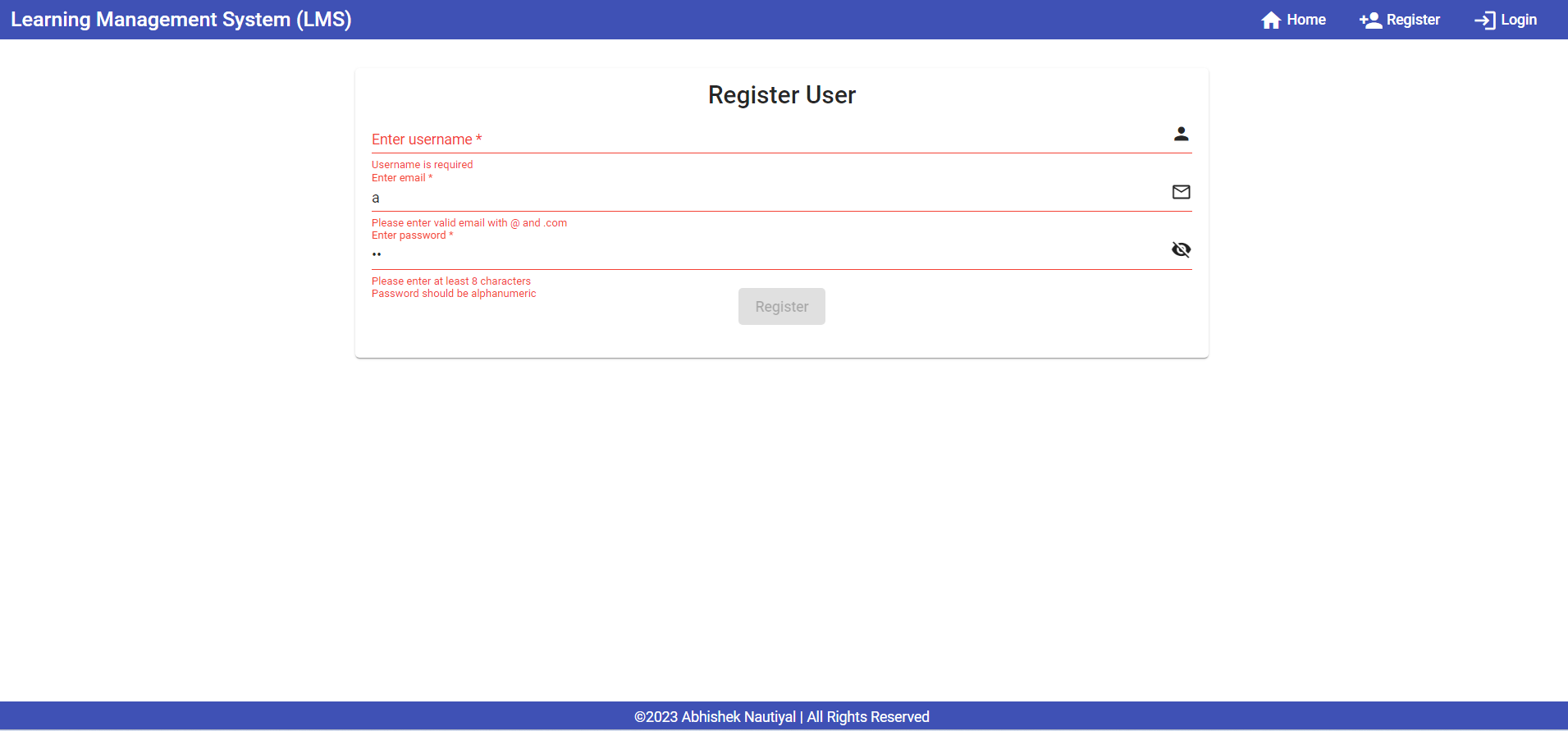
Below is the Add Course screen where Admin can add course –



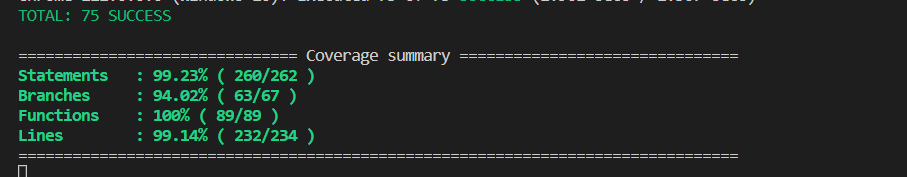
Below is the Update Course screen where Admin can update course –

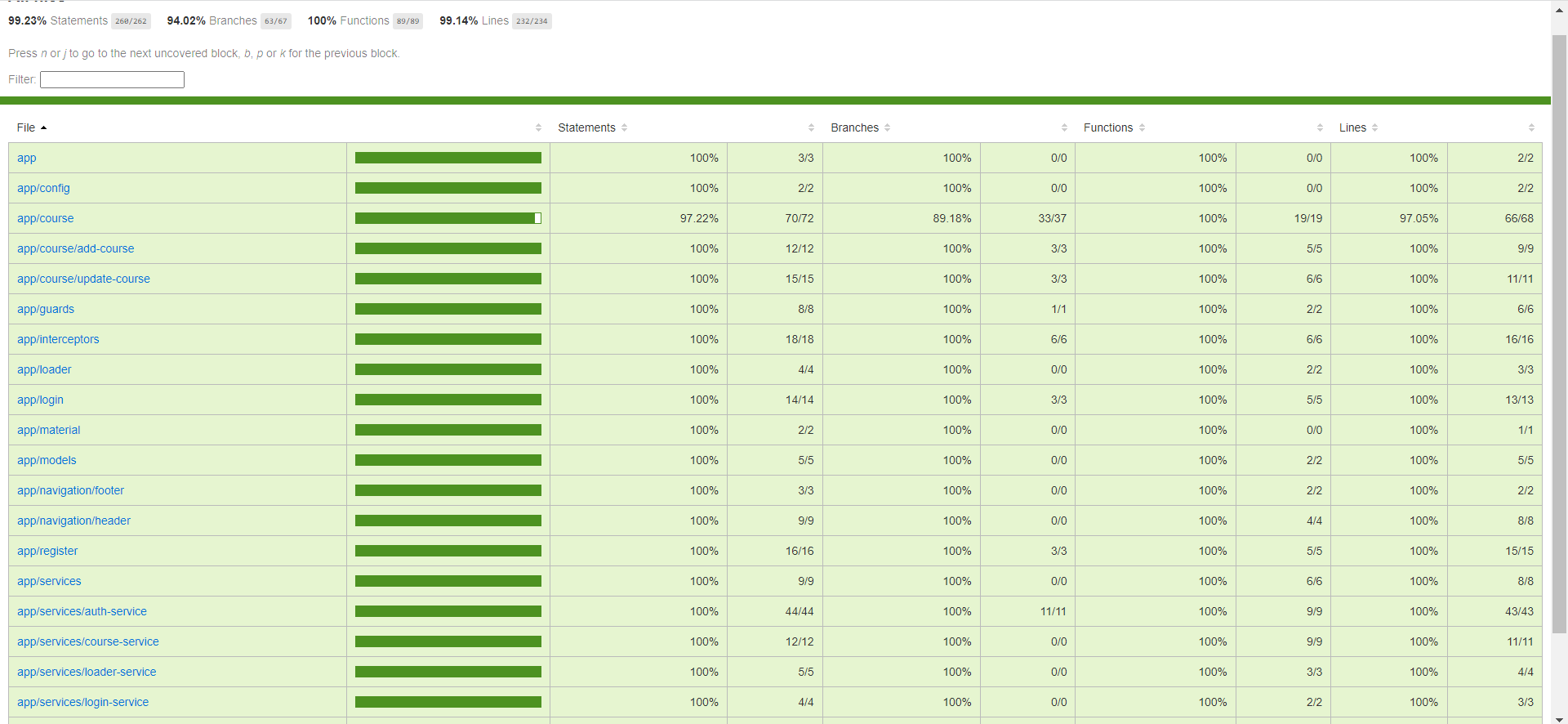


Front End User Validation is Added for Register and Login page -



Below is the unit test case implementation along with code coverage for Front End –

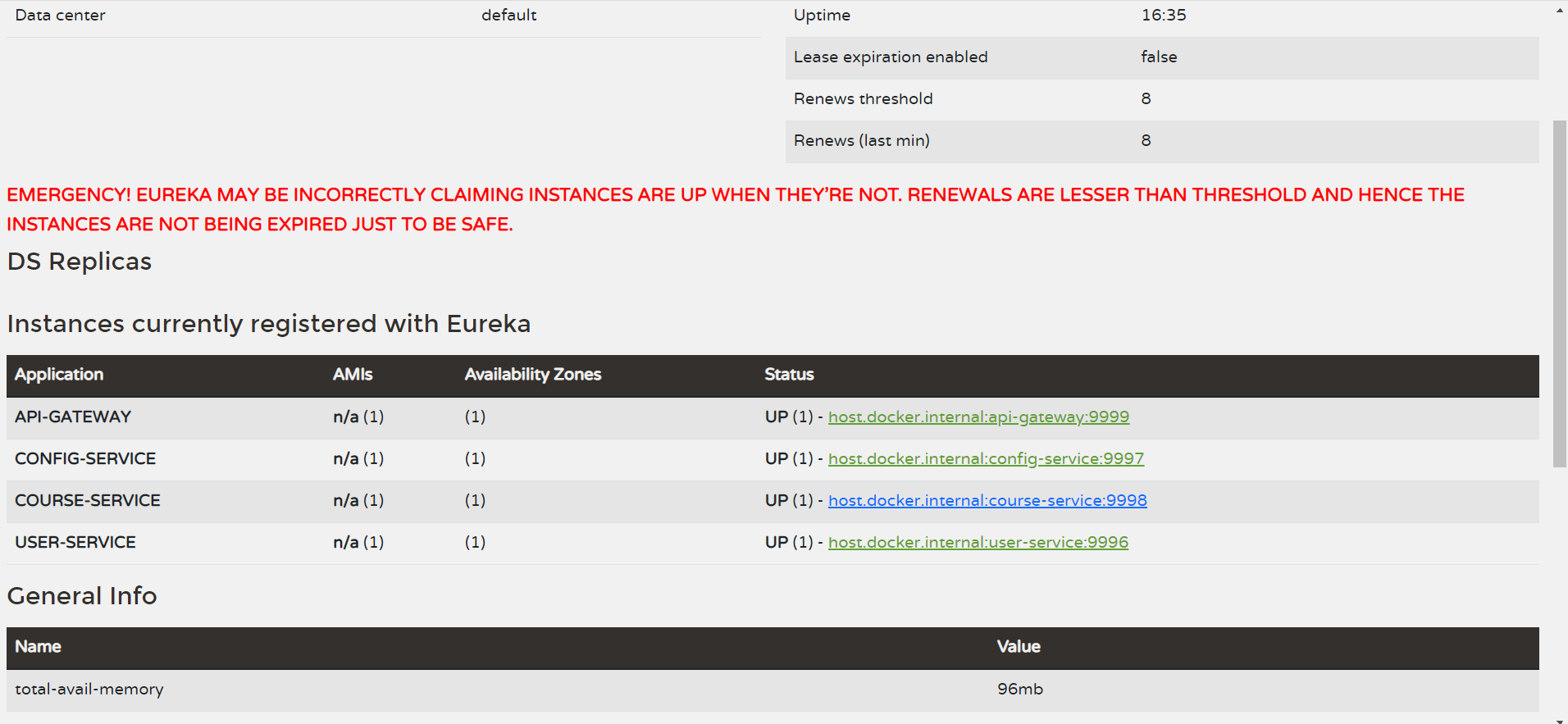




**Back End –**

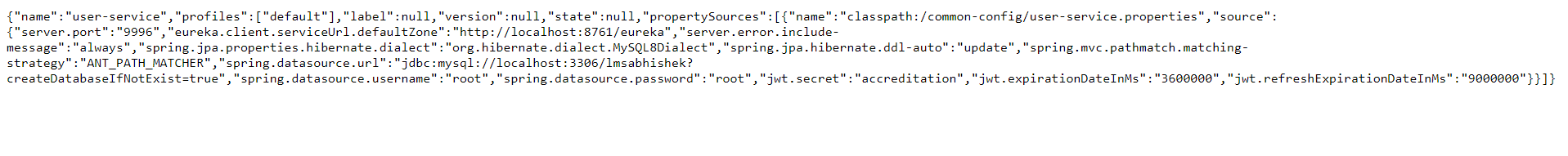
5 microservices are created as part of Back End – api-gateway, service-registry, config-service, course-service and user-service

Below is the service-registry – eureka page accessed via api-gateway service –

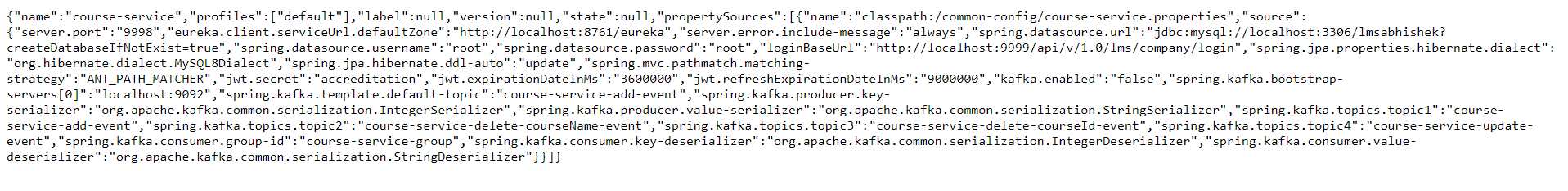


Config-service is created for maintaining single configuration file based on micro-service –

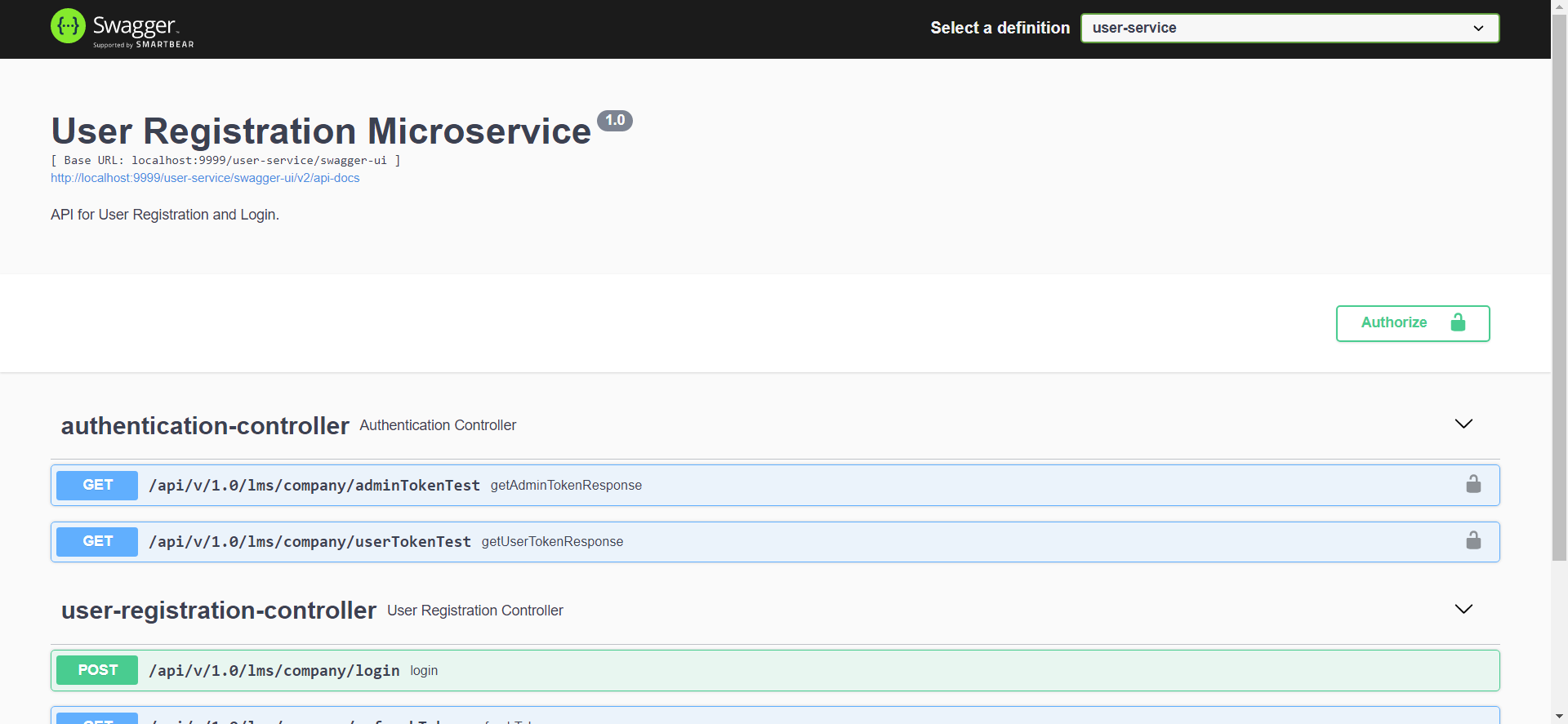
Example user-service configuration present in config-service looks like below –



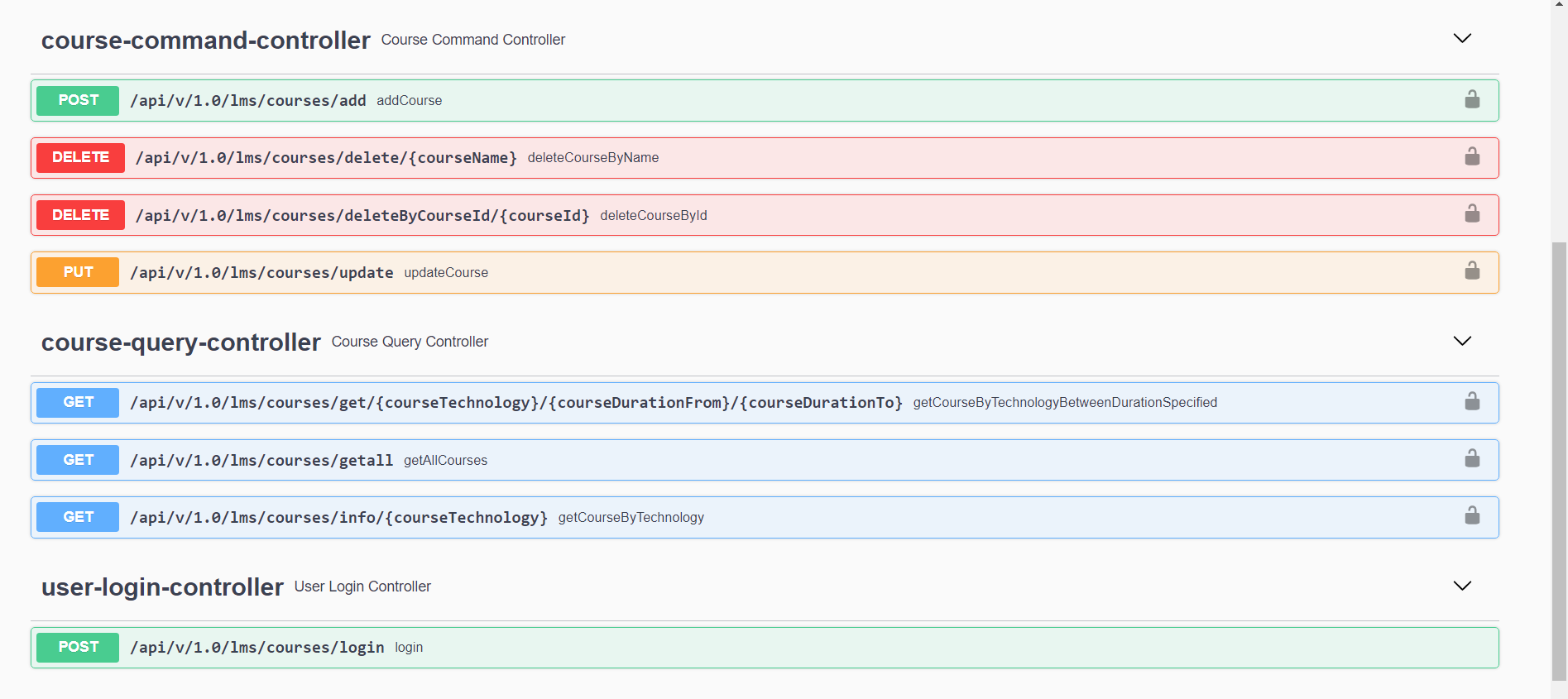
Similarly, course-service configuration present in config-service looks like below –



Below is the Swagger-UI page which can be accessed via api-gateway service and based on service definition selected one can test Api – end points.

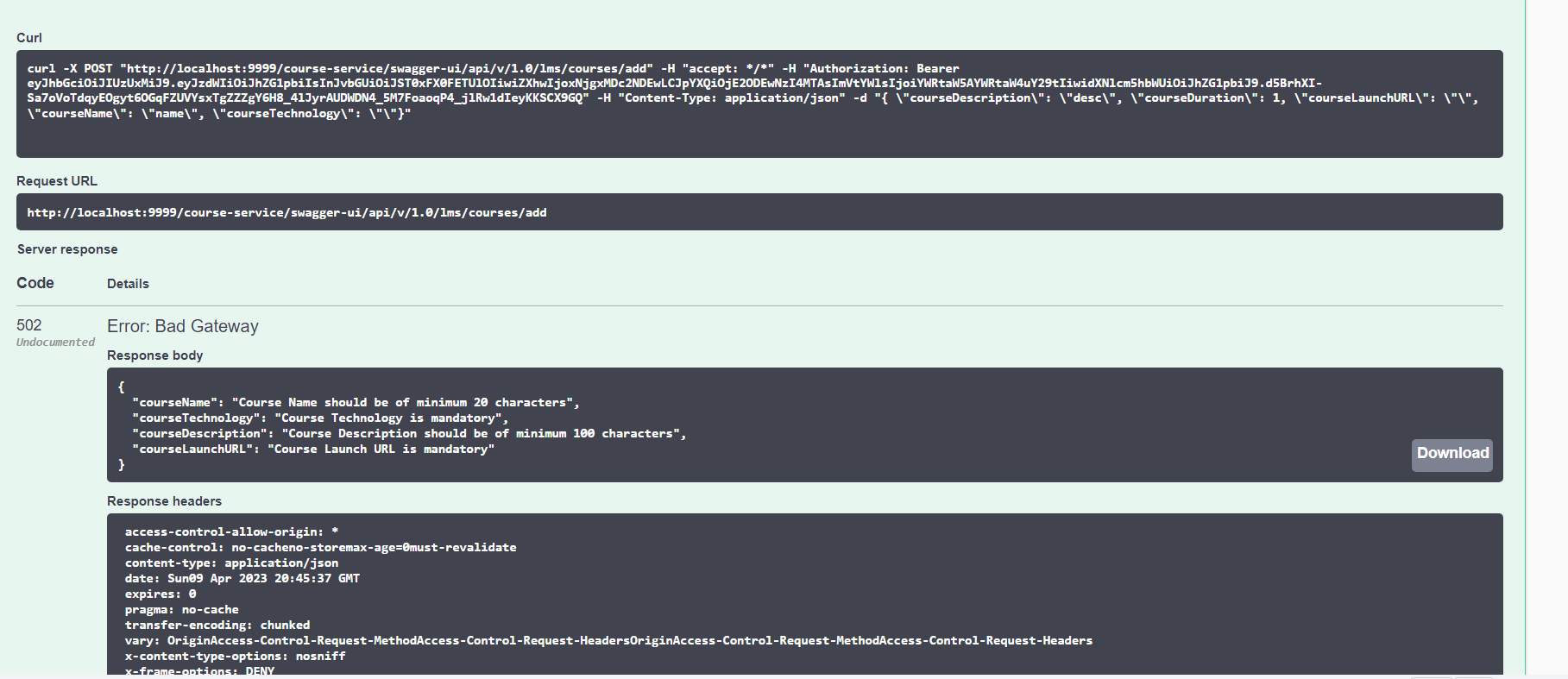


Below is course-service swagger page – where end points can be tested after passing access token



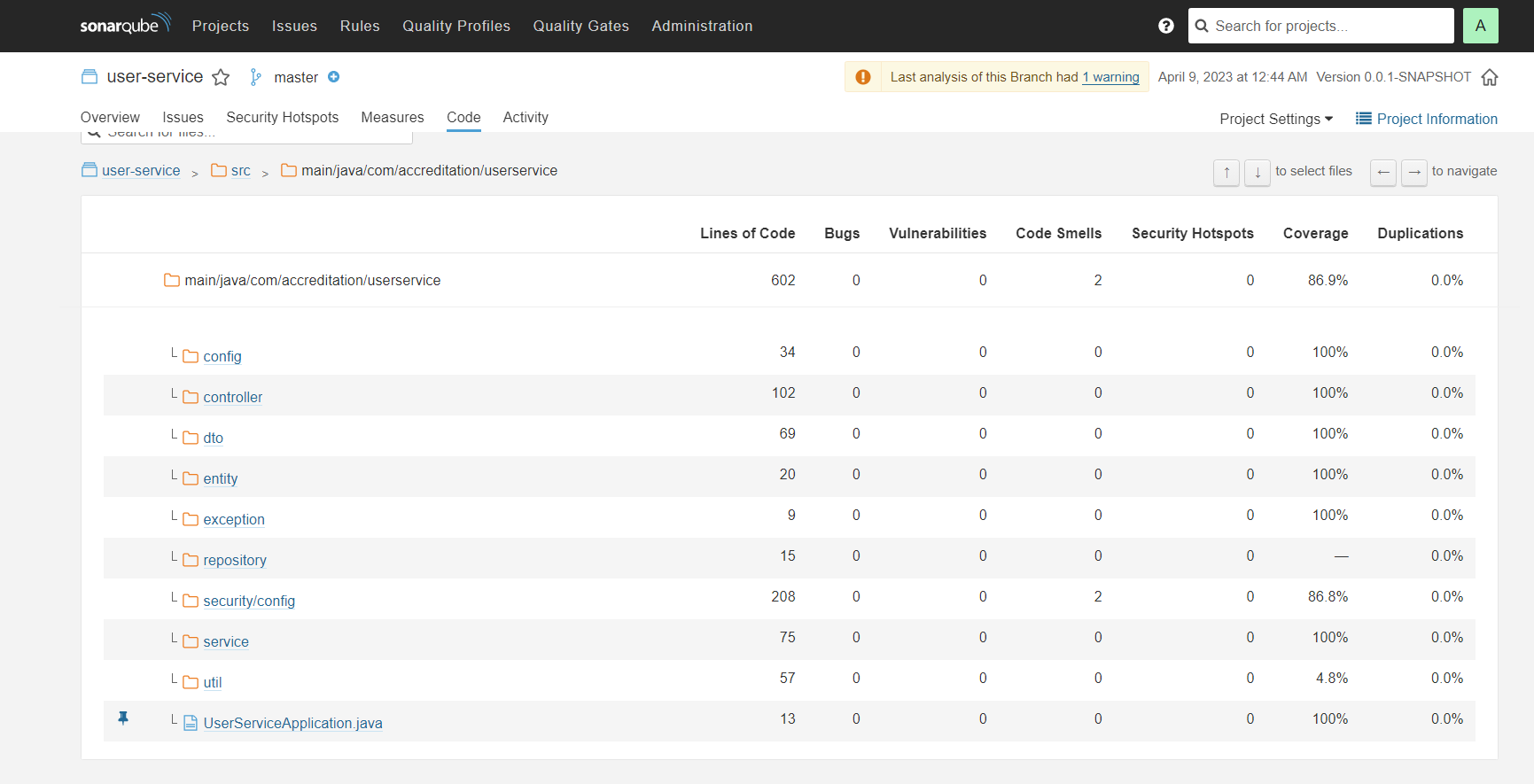
**Note – Communication between course-service and user-service is via Rest Template, we are calling /login end point from course-service which is internally triggering user-service login end point.**

Server-side validation is implemented for invalid input –

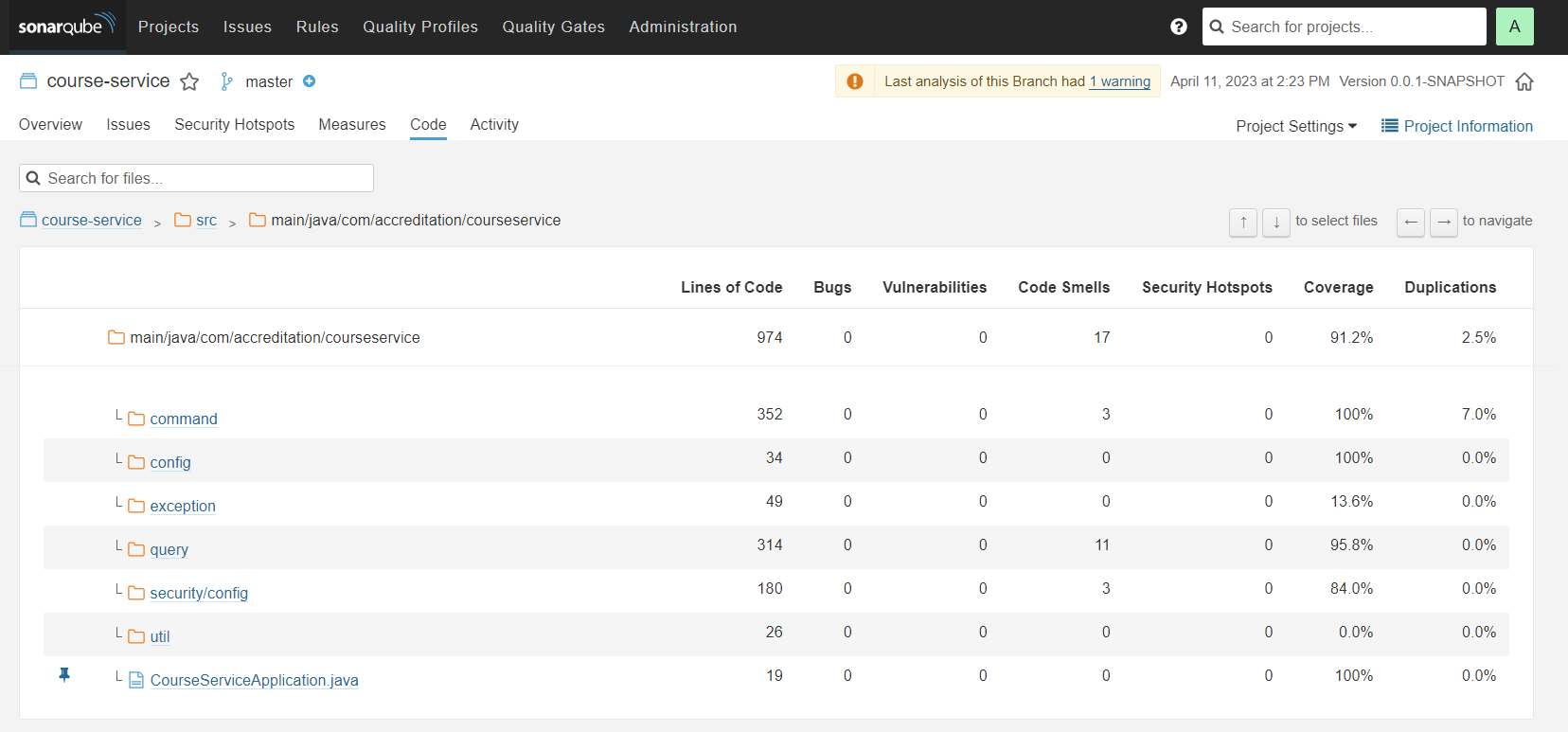


SonarQube is integrated for code coverage and security hotspot check –

Below is for user-service -

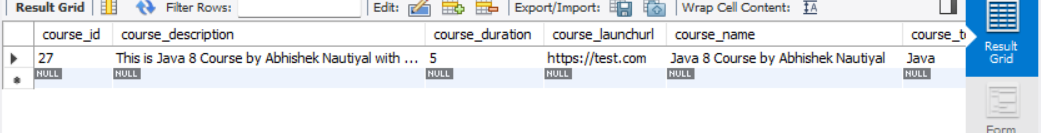


Below is for course-service –

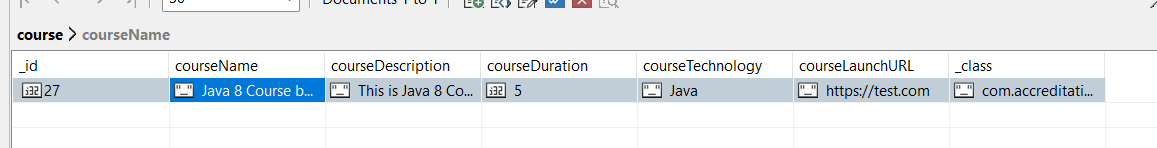


CQRS is also implemented where we are syncing MySQL and MongoDB using Kafka –

Below is the snippet from MySQL DB -



Below is the snippet from Mongo DB -



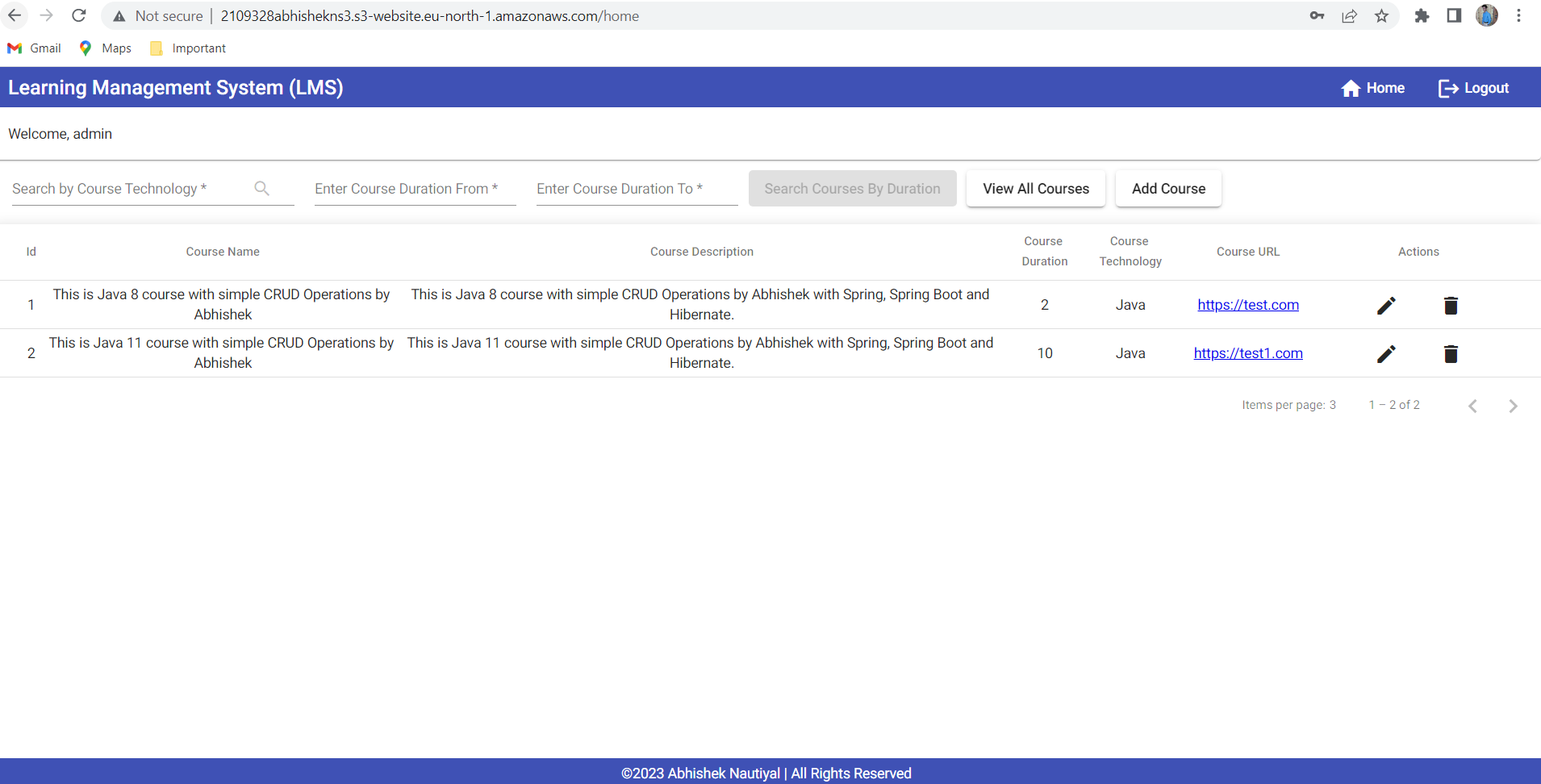
**Note – kafka.enabled property should be true in course-service property file present in config-service to sync MySQL and Mongo else only MySQL which will be referred to fetch records.**

**Cloud -**

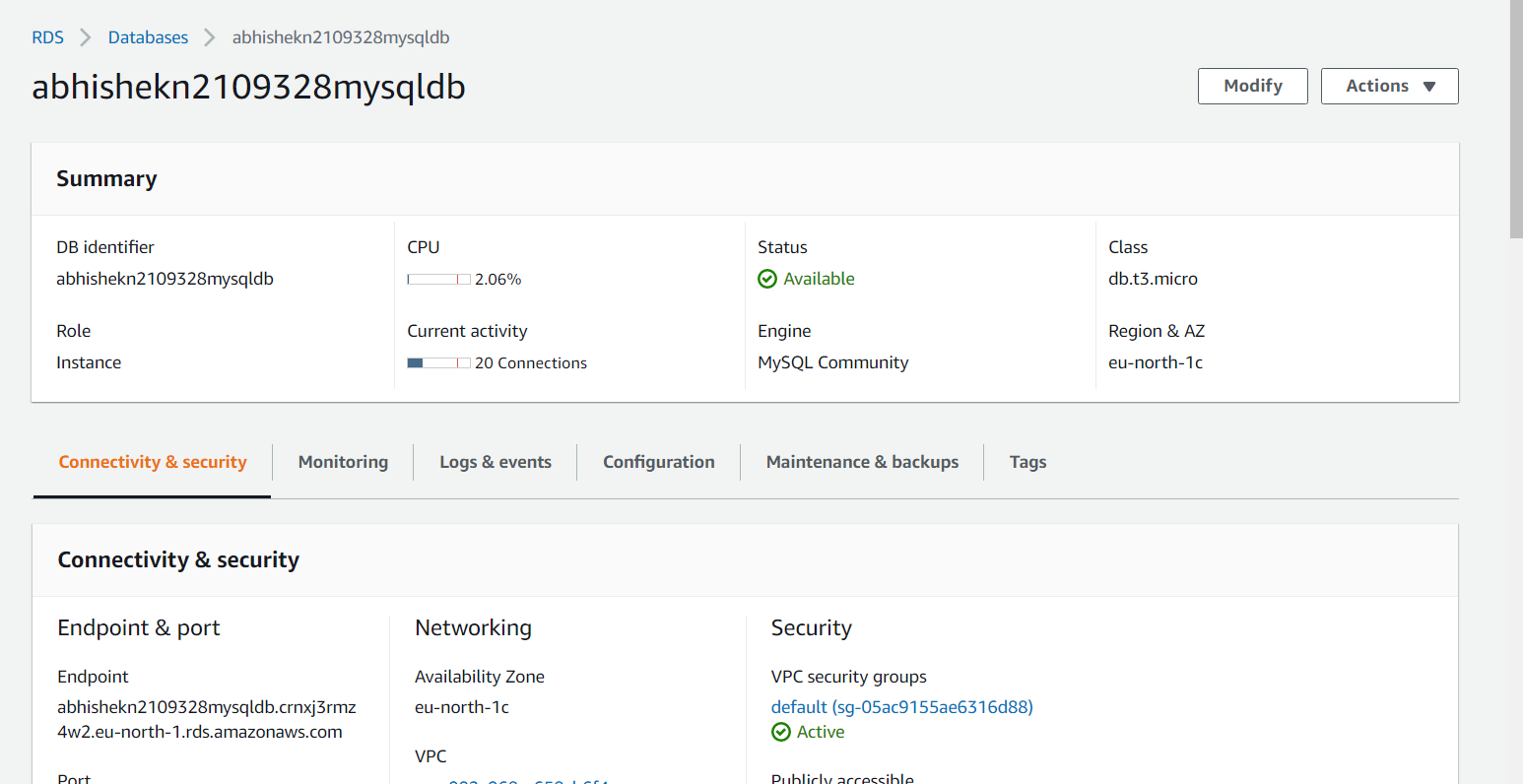
Front End is hosted via Static website hosting of S3 – Below is the URL -

[**http://2109328abhishekns3.s3-website.eu-north-1.amazonaws.com**](http://2109328abhishekns3.s3-website.eu-north-1.amazonaws.com/home)

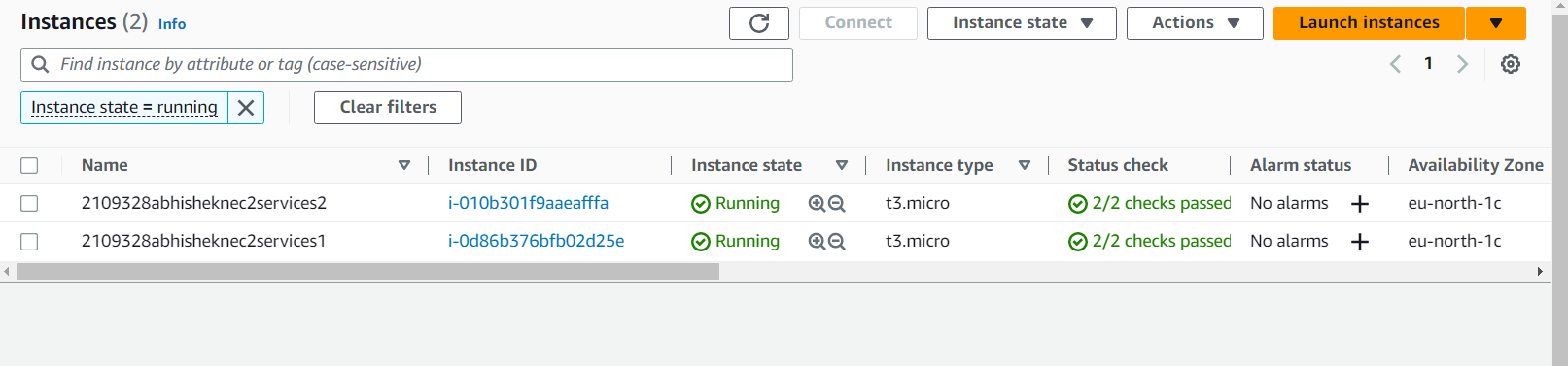
Below is the snippet of application working fine via Static website hosting –



RDS database is created in AWS – Refer below snippet –

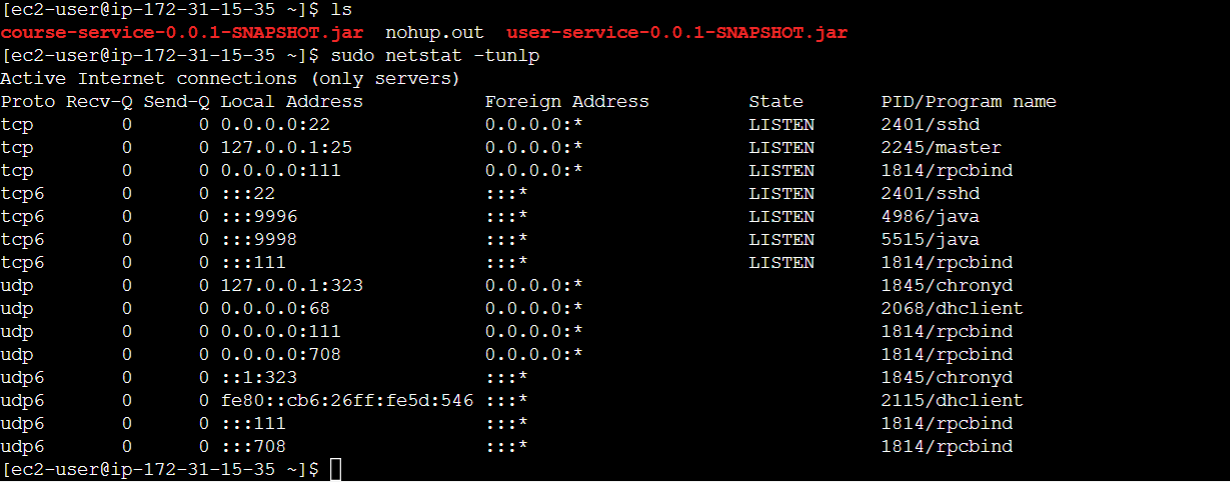


Two EC2 instances are created and in one instance 3 jar files are deployed and in other instance 2 jar files are deployed –



**Note- Reason for creating 2 EC2 instance was VM was getting freeze when all jar file deployed in one instance and started.**

Refer below snippet user-service is running on port 9996 and course-service is running on port 9998

****

Refer below snippet api-gateway is running on port 9999, service-registry is running on port 8761 and config-service is running on port 9997

****

All end points of different micro-service can be accessed via api-gateway end point –

Refer below few URL for reference –

Eureka URL - <http://13.49.167.12:9999/eureka/web/>

User Service configuration via Config Service URL - <http://13.49.167.12:9999/user-service/default/>

Course Service configuration via Config Service URL - <http://13.49.167.12:9999/course-service/default/>

Swagger URL - <http://13.49.167.12:9999/swagger-ui/>