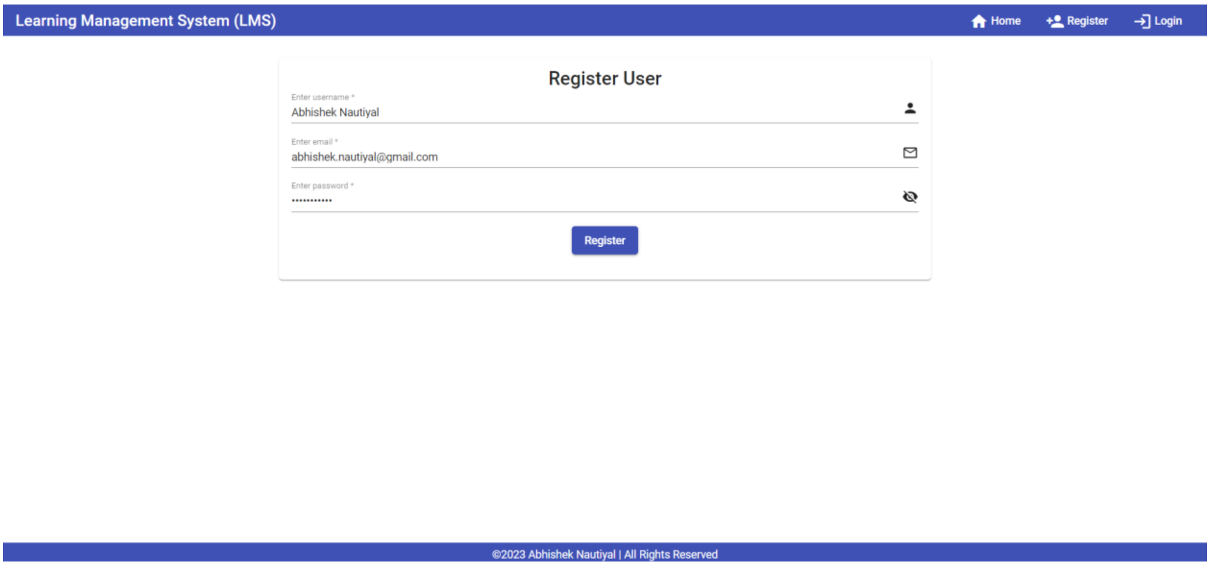
Project – Learning Management System (LMS)

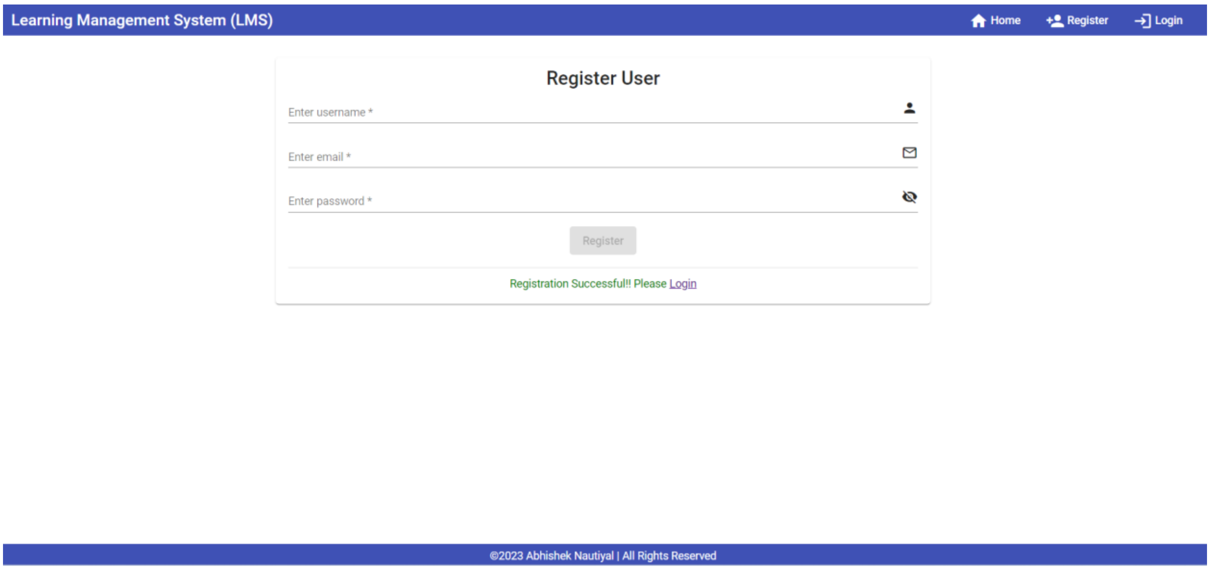
Front End –

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.

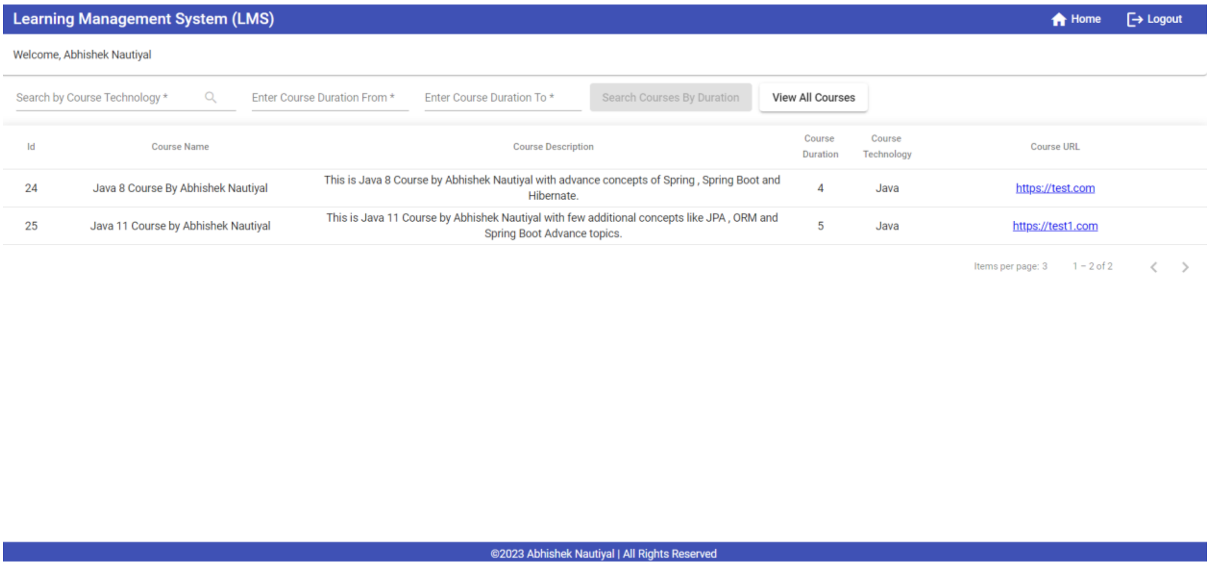
Below is the User Registration Page -



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


Learning Management System (LMS)
Home
Register
Login
Register User
Enter username *
Enter email *
Enter password *
Register
Registration Successful!! Please Login
©2023 Abhishek Nautiyal | All Rights Reserved

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 –


Learning Management System (LMS)
Home
Logout
Search by Course Technology *
Enter Course Duration From *
Enter Course Duration To *
Search Courses By Duration
View All Courses

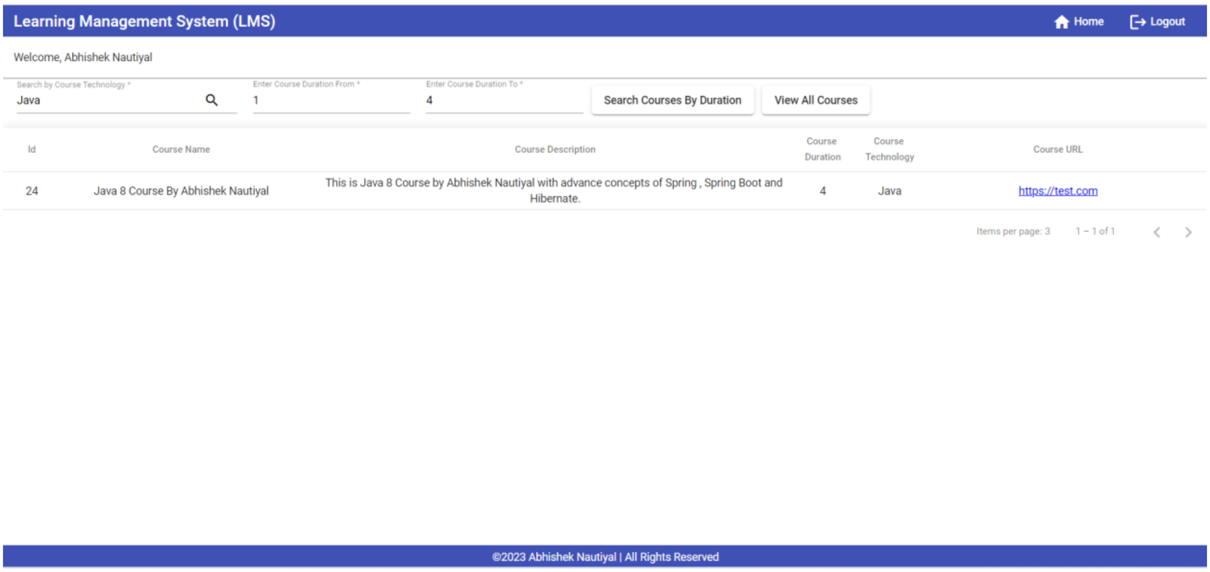
Welcome, Abhishek Nautiyal

Id	Course Name	Course Description	Course Duration	Course Technology	Course URL
24	Java 8 Course By Abhishek Nautiyal	This is Java 8 Course by Abhishek Nautiyal with advance concepts of Spring , Spring Boot and Hibernate.	4	Java	https://test.com
25	Java 11 Course by Abhishek Nautiyal	This is Java 11 Course by Abhishek Nautiyal with few additional concepts like JPA , ORM and Spring Boot Advance topics.	5	Java	https://test1.com

Items per page: 3   1 – 2 of 2

©2023 Abhishek Nautiyal | All Rights Reserved

Search functionality based on Course Technology and Course Duration –



Welcome, Abhishek Nautiyal

Search by Course Technology *: Java

Enter Course Duration From *: 1

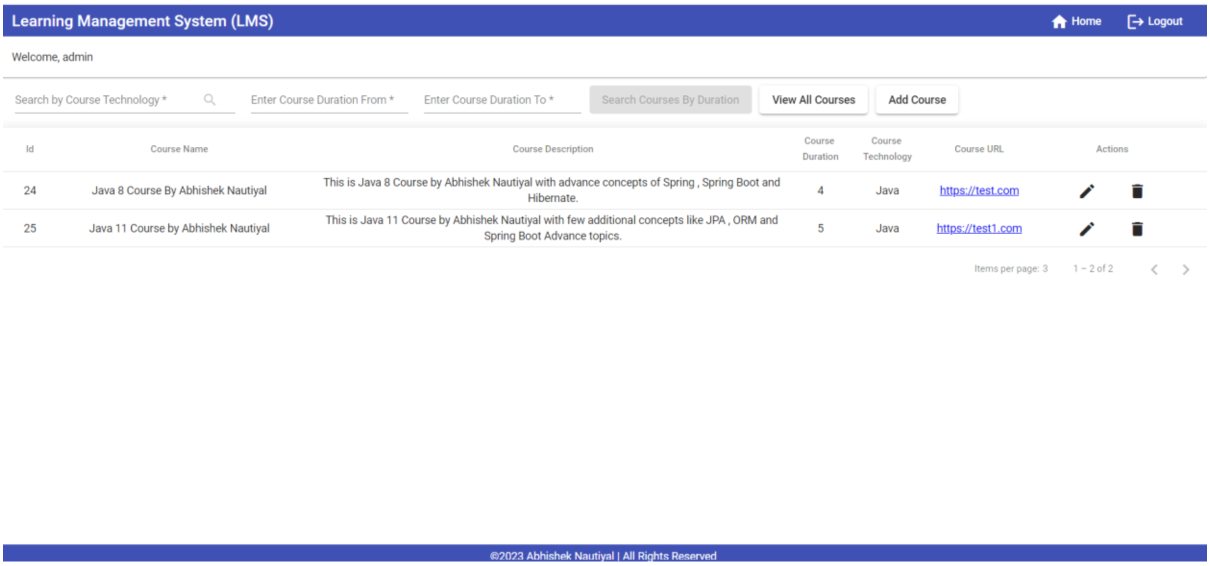
Enter Course Duration To *: 4

Id	Course Name	Course Description	Course Duration	Course Technology	Course URL
24	Java 8 Course By Abhishek Nautiyal	This is Java 8 Course by Abhishek Nautiyal with advance concepts of Spring , Spring Boot and Hibernate.	4	Java	https://test.com

Items per page: 3   1 – 1 of 1

©2023 Abhishek Nautiyal | All Rights Reserved

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


Learning Management System (LMS)
Home
Logout
Search by Course Technology *
Enter Course Duration From *
Enter Course Duration To *
Search Courses By Duration
View All Courses
Add Course

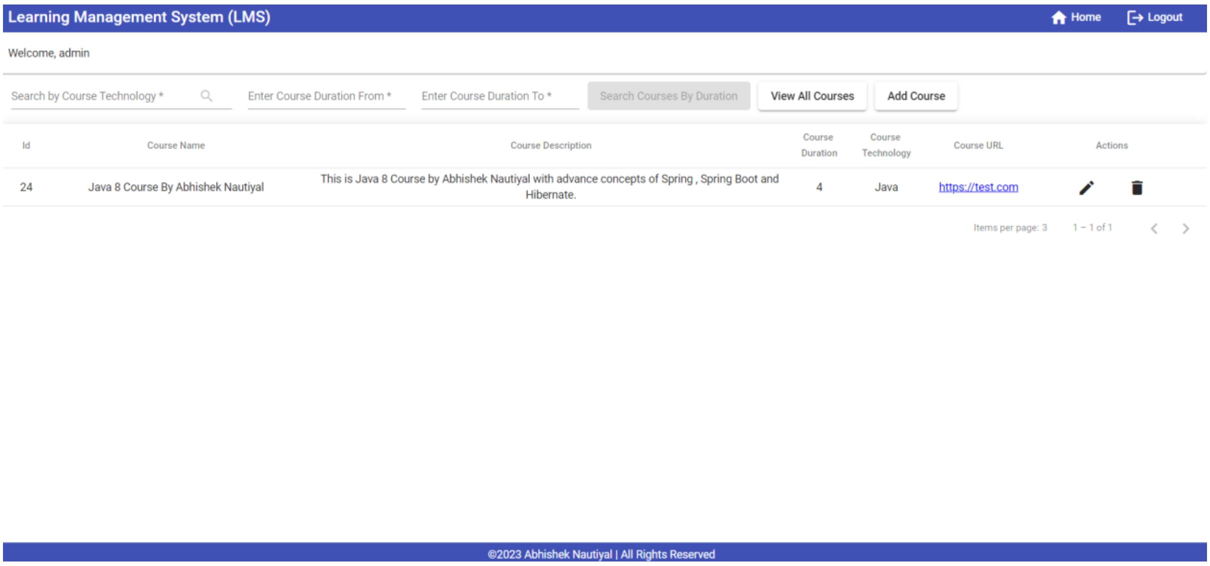
Welcome, admin

Id	Course Name	Course Description	Course Duration	Course Technology	Course URL	Actions
24	Java 8 Course By Abhishek Nautiyal	This is Java 8 Course by Abhishek Nautiyal with advance concepts of Spring , Spring Boot and Hibernate.	4	Java	https://test.com	
25	Java 11 Course by Abhishek Nautiyal	This is Java 11 Course by Abhishek Nautiyal with few additional concepts like JPA , ORM and Spring Boot Advance topics.	5	Java	https://test1.com	

Items per page: 3 1 – 2 of 2

©2023 Abhishek Nautiyal | All Rights Reserved

Functionality when admin click on delete icon to delete course -


Learning Management System (LMS)
Home
Logout
Search by Course Technology *
Enter Course Duration From *
Enter Course Duration To *
Search Courses By Duration
View All Courses
Add Course

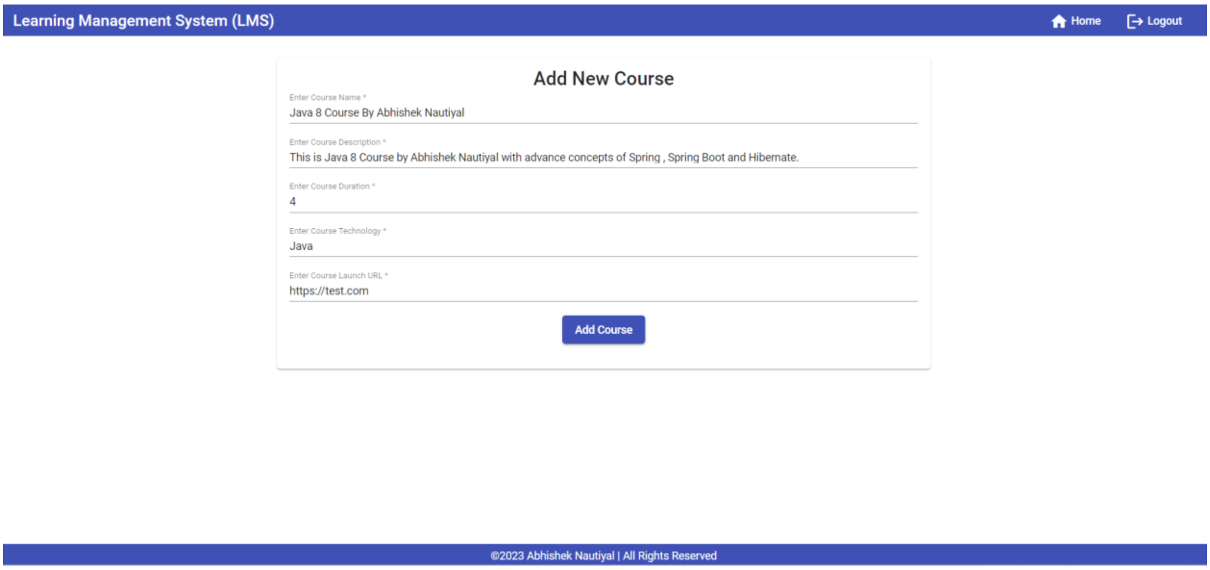
Welcome, admin

Id	Course Name	Course Description	Course Duration	Course Technology	Course URL	Actions
24	Java 8 Course By Abhishek Nautiyal	This is Java 8 Course by Abhishek Nautiyal with advance concepts of Spring , Spring Boot and Hibernate.	4	Java	https://test.com	

Items per page: 3 1 – 1 of 1

©2023 Abhishek Nautiyal | All Rights Reserved

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



Add New Course

Enter Course Name *
Java 8 Course By Abhishek Nautiyal

Enter Course Description *
This is Java 8 Course by Abhishek Nautiyal with advance concepts of Spring , Spring Boot and Hibernate.

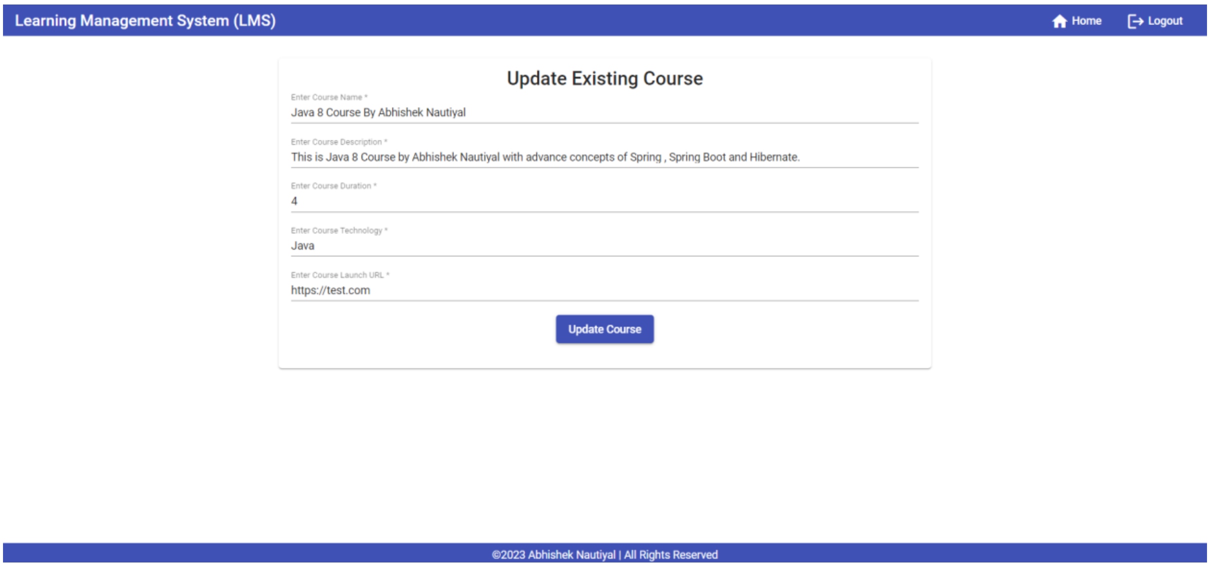
Enter Course Duration *
4

Enter Course Technology *
Java

Enter Course Launch URL *
https://test.com

©2023 Abhishek Nautiyal | All Rights Reserved

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



Update Existing Course

Enter Course Name *
Java 8 Course By Abhishek Nautiyal

Enter Course Description *
This is Java 8 Course by Abhishek Nautiyal with advance concepts of Spring , Spring Boot and Hibernate.

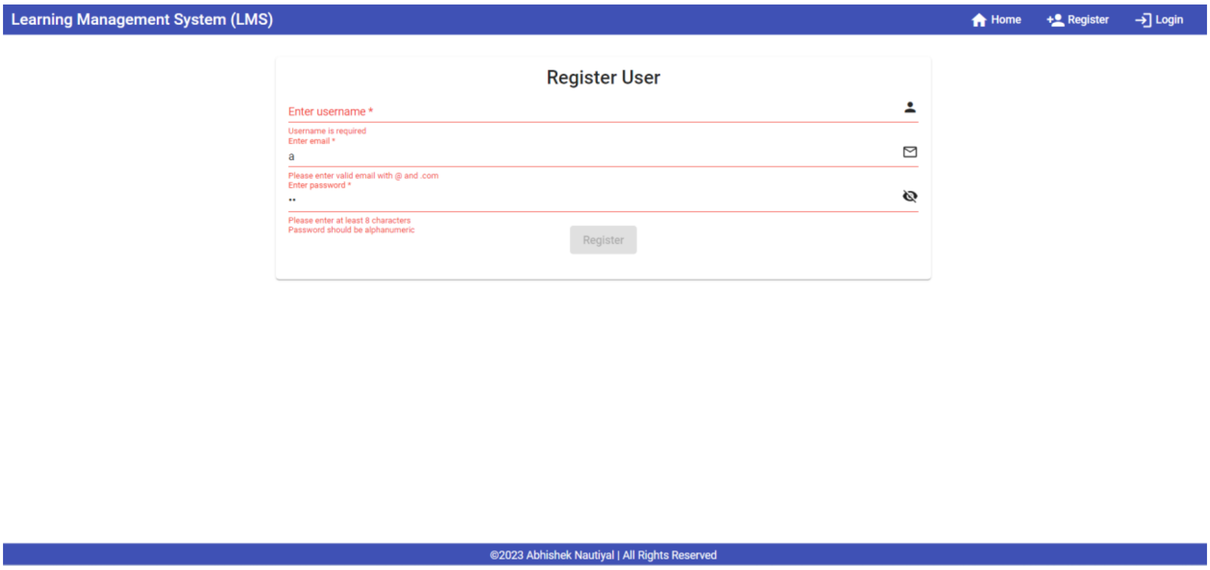
Enter Course Duration *
4

Enter Course Technology *
Java

Enter Course Launch URL *
https://test.com

©2023 Abhishek Nautiyal | All Rights Reserved

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



Register User

Enter username *

Username is required

Enter email *

a

Please enter valid email with @ and .com

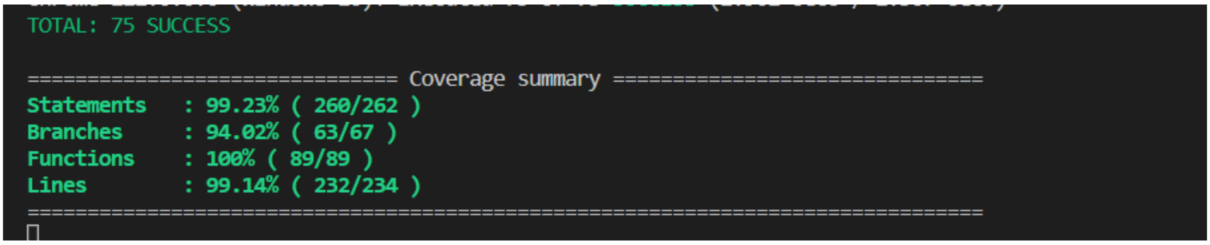
Enter password *

••

Please enter at least 8 characters
Password should be alphanumeric

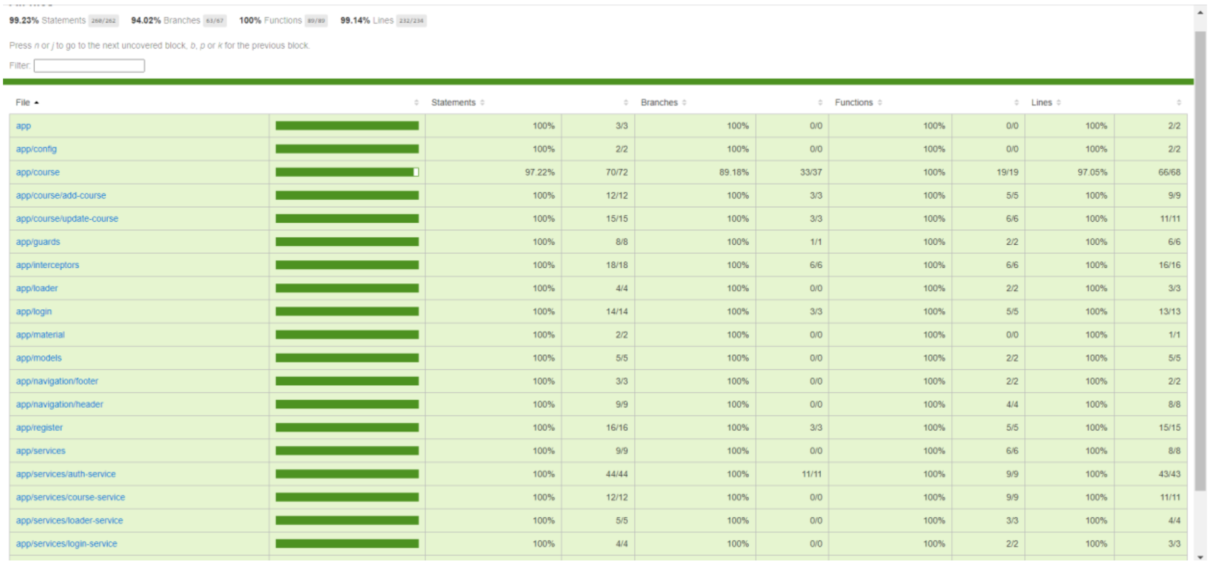
©2023 Abhishek Nautiyal | All Rights Reserved

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



```
TOTAL: 75 SUCCESS

=============================== Coverage summary ===============================
Statements   : 99.23% ( 260/262 )
Branches     : 94.02% ( 63/67 )
Functions    : 100% ( 89/89 )
Lines        : 99.14% ( 232/234 )
================================================================================
```



99.23% Statements 260/262 **94.02%** Branches 63/67 **100%** Functions 89/89 **99.14%** Lines 232/234

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

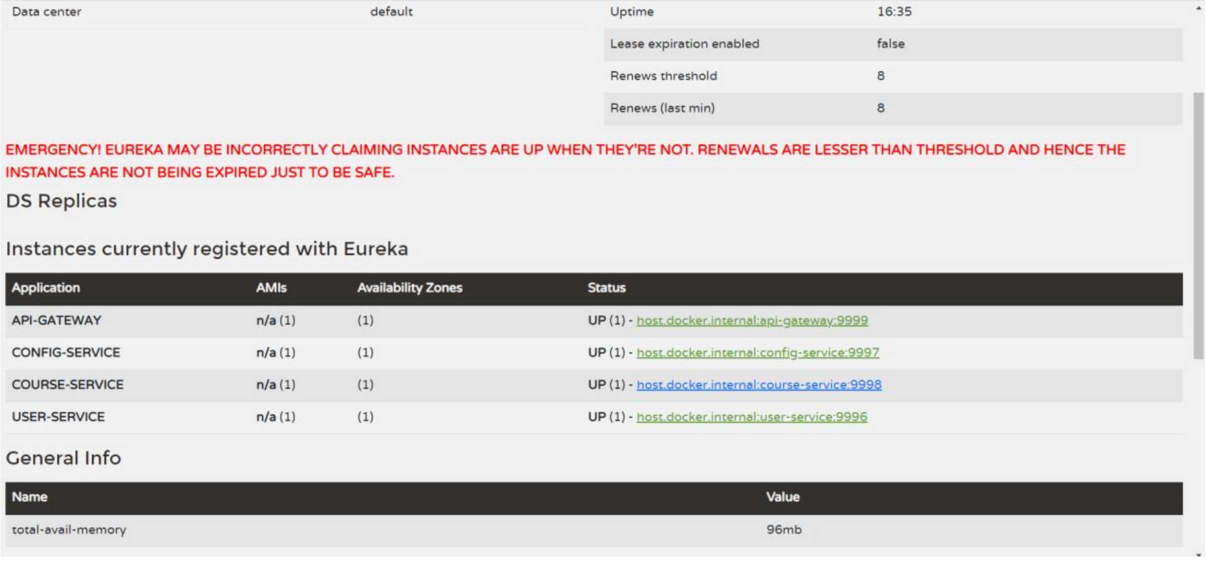
Filter:

File		Statements		Branches		Functions		Lines	
app		100%	3/3	100%	0/0	100%	0/0	100%	2/2
app/config		100%	2/2	100%	0/0	100%	0/0	100%	2/2
app/course		97.22%	70/72	89.18%	33/37	100%	19/19	97.05%	66/68
app/course/add-course		100%	12/12	100%	3/3	100%	5/5	100%	9/9
app/course/update-course		100%	15/15	100%	3/3	100%	6/6	100%	11/11
app/guards		100%	8/8	100%	1/1	100%	2/2	100%	6/6
app/interceptors		100%	18/18	100%	6/6	100%	6/6	100%	16/16
app/loader		100%	4/4	100%	0/0	100%	2/2	100%	3/3
app/login		100%	14/14	100%	3/3	100%	5/5	100%	13/13
app/material		100%	2/2	100%	0/0	100%	0/0	100%	1/1
app/models		100%	5/5	100%	0/0	100%	2/2	100%	5/5
app/navigation/footer		100%	3/3	100%	0/0	100%	2/2	100%	2/2
app/navigation/header		100%	9/9	100%	0/0	100%	4/4	100%	8/8
app/register		100%	16/16	100%	3/3	100%	5/5	100%	15/15
app/services		100%	9/9	100%	0/0	100%	6/6	100%	8/8
app/services/auth-service		100%	44/44	100%	11/11	100%	9/9	100%	43/43
app/services/course-service		100%	12/12	100%	0/0	100%	9/9	100%	11/11
app/services/loader-service		100%	5/5	100%	0/0	100%	3/3	100%	4/4
app/services/login-service		100%	4/4	100%	0/0	100%	2/2	100%	3/3

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 –



Data center	default	Uptime	16:35
		Lease expiration enabled	false
		Renews threshold	8
		Renews (last min)	8

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

Instances currently registered with Eureka

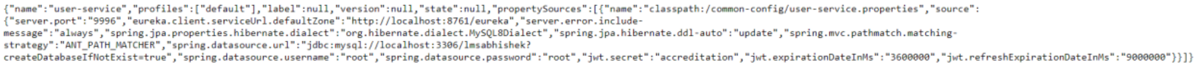
Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - host.docker.internal:api-gateway:9999
CONFIG-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:config-service:9997
COURSE-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:course-service:9998
USER-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:user-service:9996

General Info

Name	Value
total-avail-memory	96mb

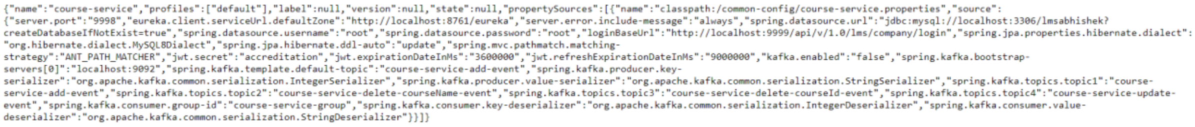
Config-service is created for maintaining single configuration file based on micro-service –

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



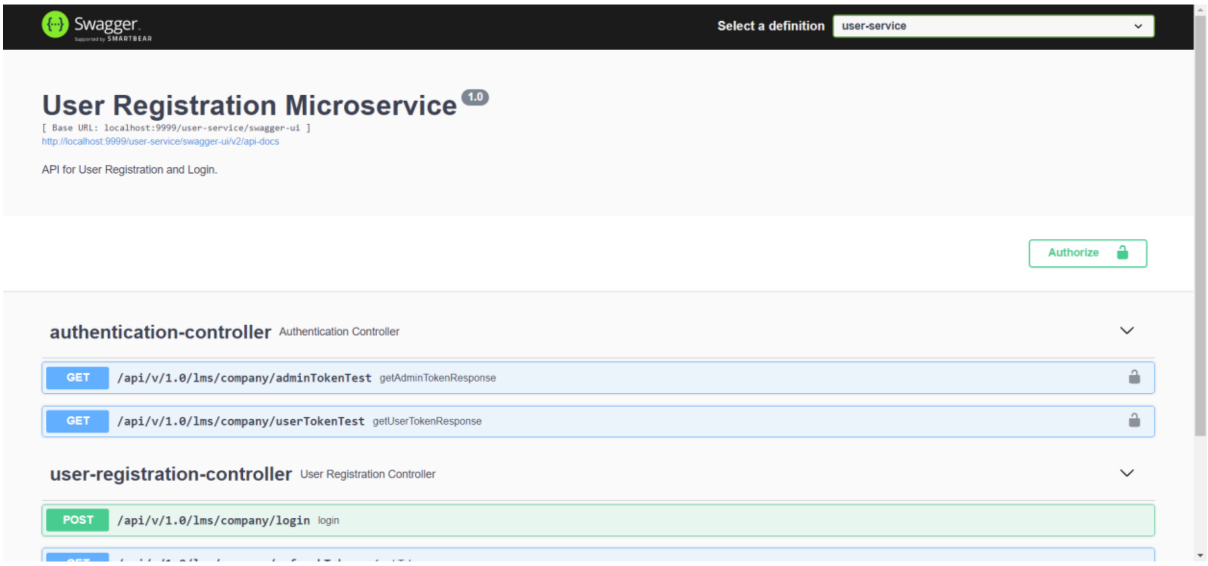
```
{"name":"user-service","profiles":["default"],"label":null,"version":null,"state":null,"propertySources":[{"name":"classpath:/common-config/user-service.properties","source":
{"server.port":"9996","eureka.client.serviceUrl.defaultZone":"http://localhost:8761/eureka","server.error.include-
message":"always","spring.jpa.properties.hibernate.dialect":"org.hibernate.dialect.MySQL8Dialect","spring.jpa.hibernate.ddl-auto":"update","spring.mvc.pathmatch.matching-
strategy":"ANT_PATH_MATCHER","spring.datasource.url":"jdbc:mysql://localhost:3306/lmsabhishek?
createDatabaseIfNotExist=true","spring.datasource.username":"root","spring.datasource.password":"root","jwt.secret":"accreditation","jwt.expirationDateInMs":"3600000","jwt.refreshExpirationDateInMs":"9000000"}}]}
```

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



```
{"name":"course-service","profiles":["default"],"label":null,"version":null,"state":null,"propertySources":[{"name":"classpath:/common-config/course-service.properties","source":
{"server.port":"9998","eureka.client.serviceUrl.defaultZone":"http://localhost:8761/eureka","server.error.include-message":"always","spring.datasource.url":"jdbc:mysql://localhost:3306/lmsabhishek?
createDatabaseIfNotExist=true","spring.datasource.username":"root","spring.datasource.password":"root","loginBaseUrl":"http://localhost:9999/api/v/1.0/lms/company/login","spring.jpa.properties.hibernate.dialect":
"org.hibernate.dialect.MySQL8Dialect","spring.jpa.hibernate.ddl-auto":"update","spring.mvc.pathmatch.matching-
strategy":"ANT_PATH_MATCHER","jwt.secret":"accreditation","jwt.expirationDateInMs":"3600000","jwt.refreshExpirationDateInMs":"9000000","kafka.enabled":"false","spring.kafka.bootstrap-
servers[0]":"localhost:9092","spring.kafka.template.default-topic":"course-service-add-event","spring.kafka.producer.key-
serializer":"org.apache.kafka.common.serialization.IntegerSerializer","spring.kafka.producer.value-serializer":"org.apache.kafka.common.serialization.StringSerializer","spring.kafka.topics.topic1":"course-
service-add-event","spring.kafka.topics.topic2":"course-service-delete-courseName-event","spring.kafka.topics.topic3":"course-service-delete-courseId-event","spring.kafka.topics.topic4":"course-service-update-
event","spring.kafka.consumer.group-id":"course-service-group","spring.kafka.consumer.key-deserializer":"org.apache.kafka.common.serialization.IntegerDeserializer","spring.kafka.consumer.value-
deserializer":"org.apache.kafka.common.serialization.StringDeserializer"}}]}
```

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.



Select a definition: user-service

User Registration Microservice 1.0

[Base URL: localhost:9999/user-service/swagger-ui]
http://localhost:9999/user-service/swagger-ui/v2/api-docs

API for User Registration and Login.

authentication-controller Authentication Controller

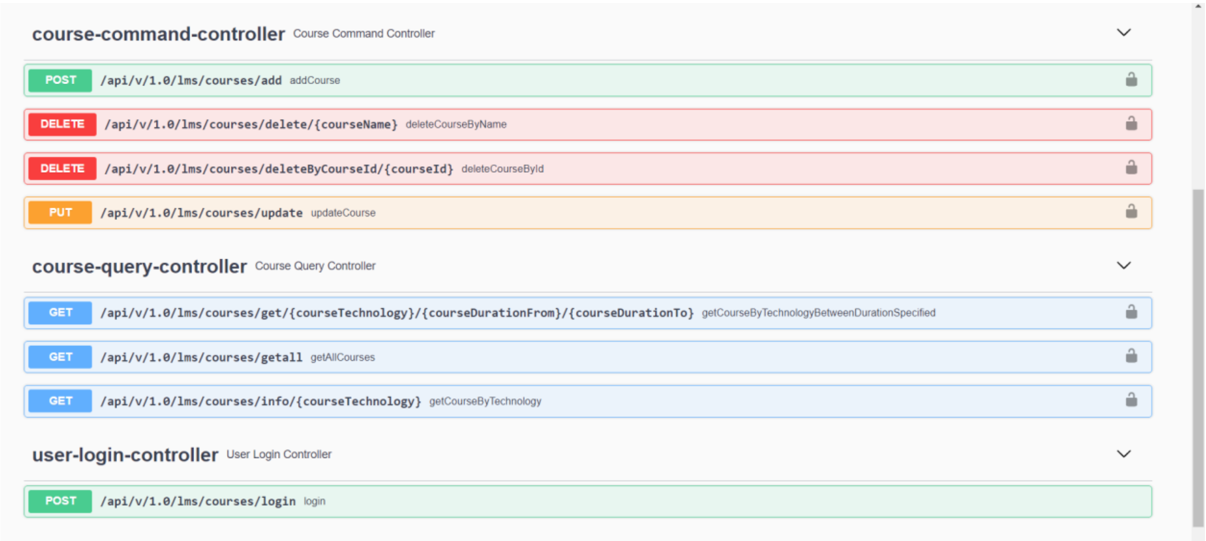
GET /api/v/1.0/lms/company/adminTokenTest getAdminTokenResponse

GET /api/v/1.0/lms/company/userTokenTest getUserTokenResponse

user-registration-controller User Registration Controller

POST /api/v/1.0/lms/company/login login

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



course-command-controller Course Command Controller

POST /api/v/1.0/lms/courses/add addCourse

DELETE /api/v/1.0/lms/courses/delete/{courseName} deleteCourseByName

DELETE /api/v/1.0/lms/courses/deleteByCourseId/{courseId} deleteCourseById

PUT /api/v/1.0/lms/courses/update updateCourse

course-query-controller Course Query Controller

GET /api/v/1.0/lms/courses/get/{courseTechnology}/{courseDurationFrom}/{courseDurationTo} getCourseByTechnologyBetweenDurationSpecified

GET /api/v/1.0/lms/courses/getall getAllCourses

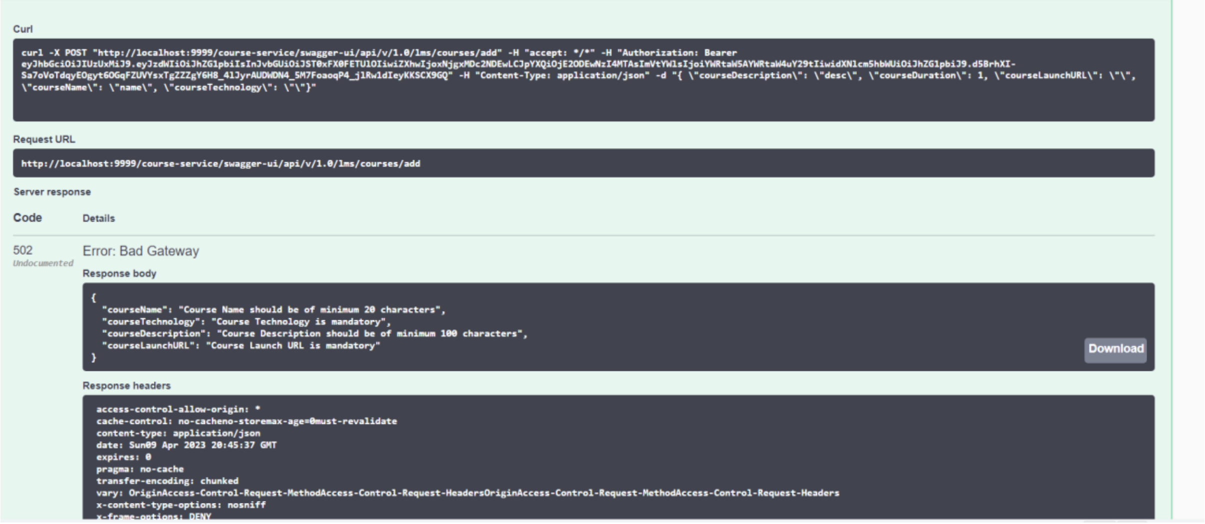
GET /api/v/1.0/lms/courses/info/{courseTechnology} getCourseByTechnology

user-login-controller User Login Controller

POST /api/v/1.0/lms/courses/login login

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 –



Curl

```
curl -X POST "http://localhost:9999/course-service/swagger-ui/api/v/1.0/lms/courses/add" -H "accept: */*" -H "Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsInJvbGUiOiJST0xFX0FETU1OIiwiZXhwIjoxNjgxMDc2NDEwLCJpYXQiOjE2ODEwNzI4MTAsImVtYWlsIjoiYWRtaW5AYWRtaW4uY29tIiwidXNlcm5hbWUiOiJhZG1pbiJ9.d5BrhXI-
Sa7oVoTdqyEOgyt6OGqFZUVYsxTgZZZgY6H8_41JyrAUDWDN4_5M7FoaoqP4_j1Rw1dIeyKKSCX9GQ" -H "Content-Type: application/json" -d "{ \"courseDescription\": \"desc\", \"courseDuration\": 1, \"courseLaunchURL\": \"\",
\"courseName\": \"name\", \"courseTechnology\": \"\"}"
```

Request URL

```
http://localhost:9999/course-service/swagger-ui/api/v/1.0/lms/courses/add
```

Server response

Code Details

502 *Undocumented* Error: Bad Gateway

Response body

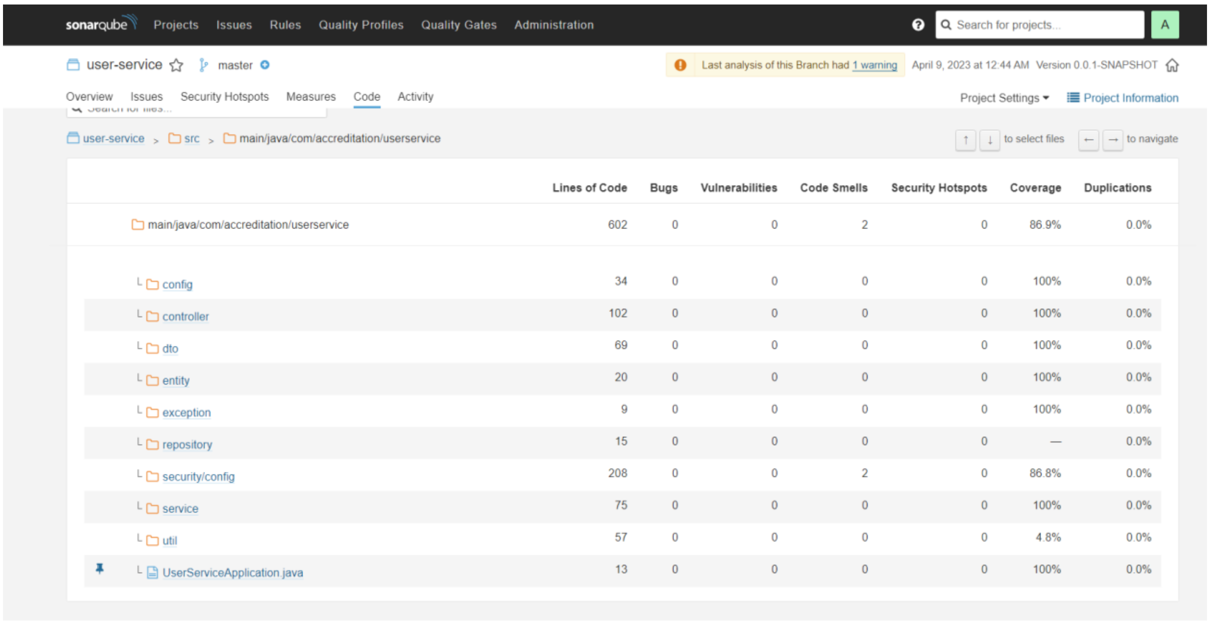
```
{
  "courseName": "Course Name should be of minimum 20 characters",
  "courseTechnology": "Course Technology is mandatory",
  "courseDescription": "Course Description should be of minimum 100 characters",
  "courseLaunchURL": "Course Launch URL is mandatory"
}
```

Response headers

```
access-control-allow-origin: *
cache-control: no-cacheno-storemax-age=0must-revalidate
content-type: application/json
date: Sun09 Apr 2023 20:45:37 GMT
expires: 0
pragma: no-cache
transfer-encoding: chunked
vary: OriginAccess-Control-Request-MethodAccess-Control-Request-HeadersOriginAccess-Control-Request-MethodAccess-Control-Request-Headers
x-content-type-options: nosniff
x-frame-options: DENY
```

SonarQube is integrated for code coverage and security hotspot check –

Below is for user-service -

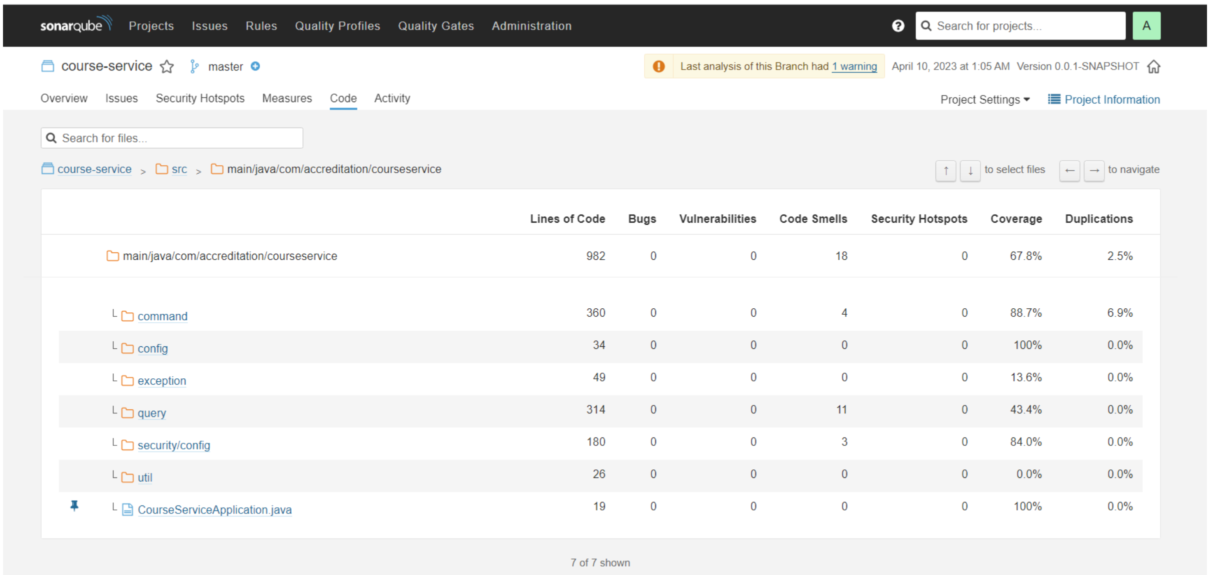


Last analysis of this Branch had 1 warning April 9, 2023 at 12:44 AM Version 0.0.1-SNAPSHOT

user-service > src > main/java/com/accreditation/userservice

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
main/java/com/accreditation/userservice	602	0	0	2	0	86.9%	0.0%
└ config	34	0	0	0	0	100%	0.0%
└ controller	102	0	0	0	0	100%	0.0%
└ dto	69	0	0	0	0	100%	0.0%
└ entity	20	0	0	0	0	100%	0.0%
└ exception	9	0	0	0	0	100%	0.0%
└ repository	15	0	0	0	0	—	0.0%
└ security/config	208	0	0	2	0	86.8%	0.0%
└ service	75	0	0	0	0	100%	0.0%
└ util	57	0	0	0	0	4.8%	0.0%
└ UserServiceApplication.java	13	0	0	0	0	100%	0.0%

Below is for course-service –



Last analysis of this Branch had 1 warning April 10, 2023 at 1:05 AM Version 0.0.1-SNAPSHOT

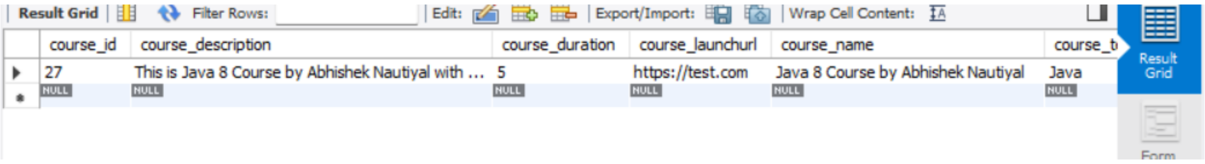
course-service > src > main/java/com/accreditation/courseservice

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
main/java/com/accreditation/courseservice	982	0	0	18	0	67.8%	2.5%
command	360	0	0	4	0	88.7%	6.9%
config	34	0	0	0	0	100%	0.0%
exception	49	0	0	0	0	13.6%	0.0%
query	314	0	0	11	0	43.4%	0.0%
security/config	180	0	0	3	0	84.0%	0.0%
util	26	0	0	0	0	0.0%	0.0%
CourseServiceApplication.java	19	0	0	0	0	100%	0.0%

7 of 7 shown

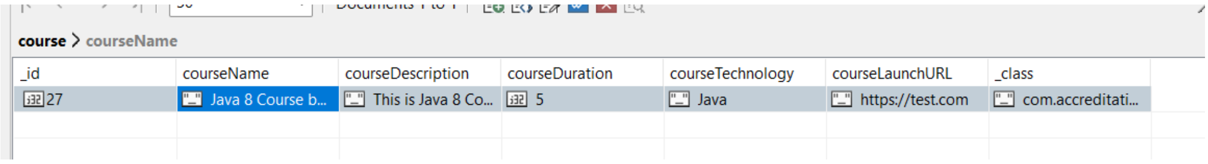
CQRS is also implemented where we are syncing MySQL and MongoDB using Kafka –

Below is the snippet from MySQL DB -



course_id	course_description	course_duration	course_launchurl	course_name	course_t
27	This is Java 8 Course by Abhishek Nautiyal with ...	5	https://test.com	Java 8 Course by Abhishek Nautiyal	Java
NULL	NULL	NULL	NULL	NULL	NULL

Below is the snippet from Mongo DB -



_id	courseName	courseDescription	courseDuration	courseTechnology	courseLaunchURL	_class
27	Java 8 Course b...	This is Java 8 Co...	5	Java	https://test.com	com.accreditati...

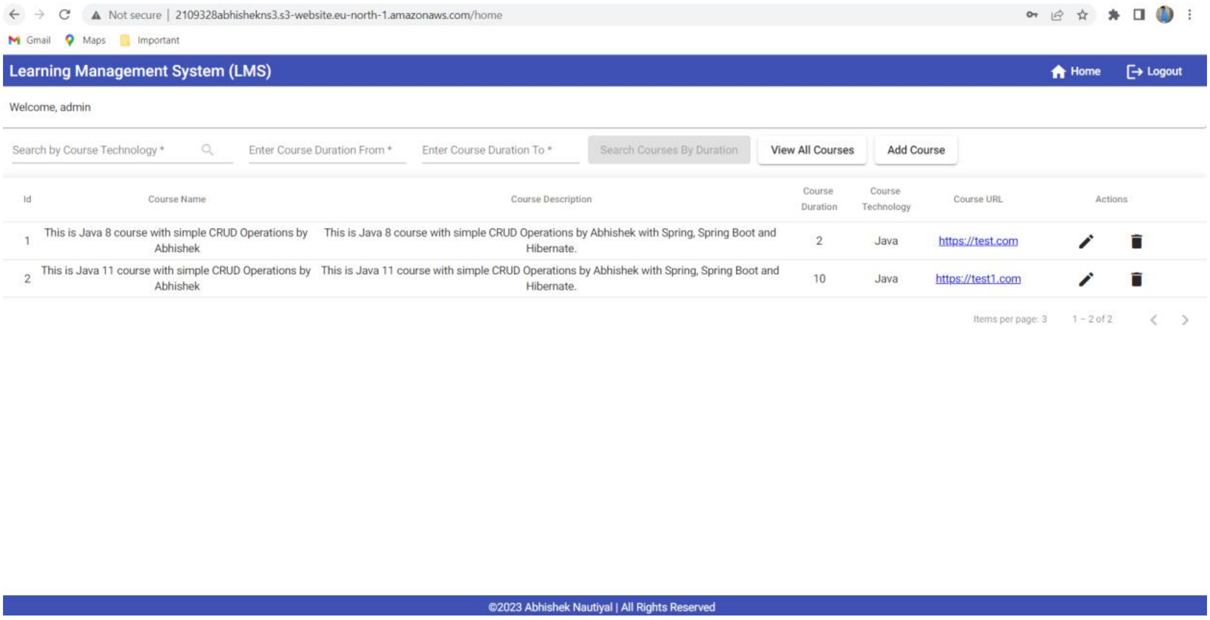
Note – kafka.enabled property should be true in course-service property file present in config-service to sync MySQL and Mongo else only MySQL which will be referred to fetch records.

Cloud -

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

http://2109328abhishekns3.s3-website.eu-north-1.amazonaws.com

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



Not secure | 2109328abhishekns3.s3-website.eu-north-1.amazonaws.com/home
Gmail
Maps
Important
Home
Logout
Search by Course Technology *
Enter Course Duration From *
Enter Course Duration To *
Search Courses By Duration
View All Courses
Add Course

Learning Management System (LMS)

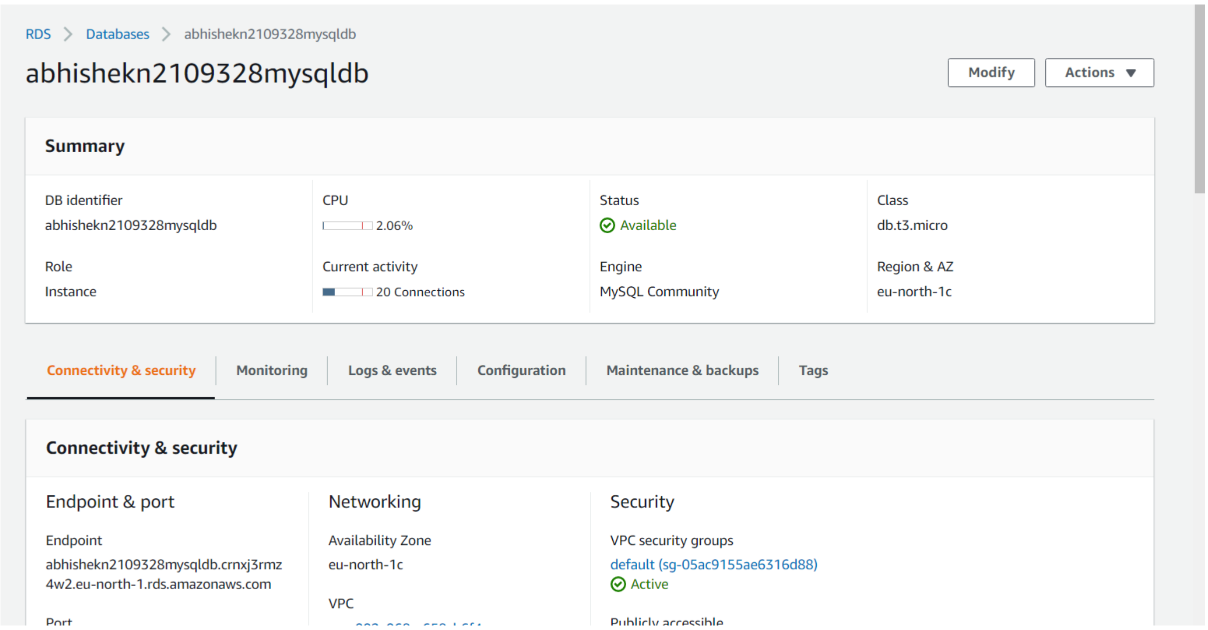
Welcome, admin

Id	Course Name	Course Description	Course Duration	Course Technology	Course URL	Actions
1	This is Java 8 course with simple CRUD Operations by Abhishek	This is Java 8 course with simple CRUD Operations by Abhishek with Spring, Spring Boot and Hibernate.	2	Java	https://test.com	✎ 🗑
2	This is Java 11 course with simple CRUD Operations by Abhishek	This is Java 11 course with simple CRUD Operations by Abhishek with Spring, Spring Boot and Hibernate.	10	Java	https://test1.com	✎ 🗑

Items per page: 3 1 – 2 of 2

©2023 Abhishek Nautiyal | All Rights Reserved

RDS database is created in AWS – Refer below snippet –



abhishekn2109328mysqldb

Summary

DB identifier: abhishekn2109328mysqldb

CPU: 2.06%

Status: Available

Class: db.t3.micro

Role: Instance

Current activity: 20 Connections

Engine: MySQL Community

Region & AZ: eu-north-1c

Connectivity & security

Endpoint & port

Endpoint: abhishekn2109328mysqldb.crnxj3rmz4w2.eu-north-1.rds.amazonaws.com

Port

Networking

Availability Zone: eu-north-1c

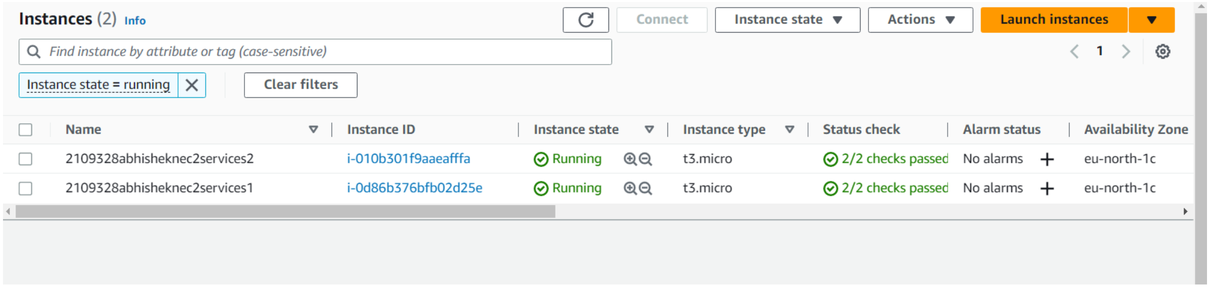
VPC

Security

VPC security groups: default (sg-05ac9155ae6316d88) Active

Publicly accessible

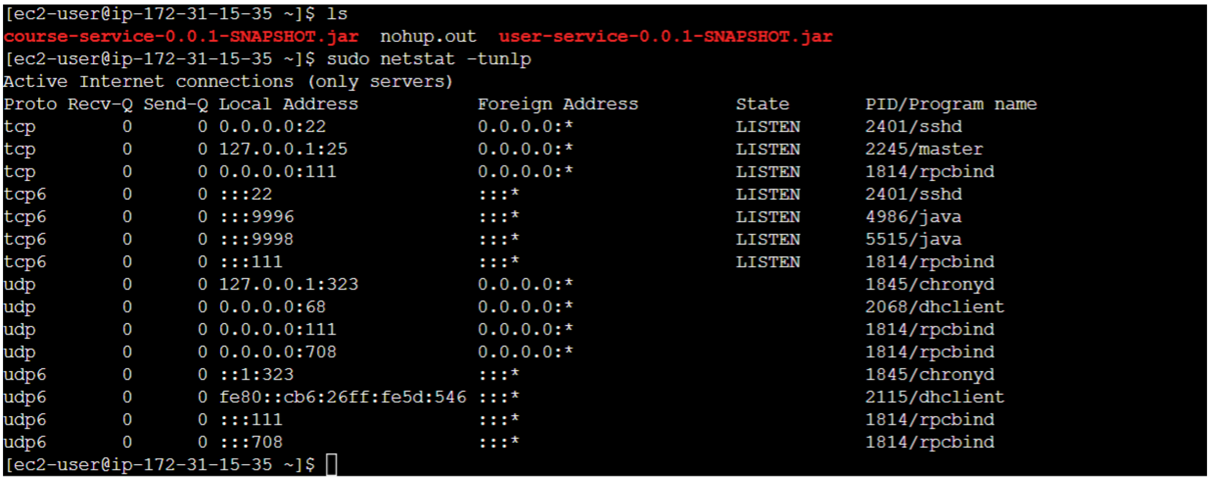
Two EC2 instances are created and in one instance 3 jar files are deployed and in other instance 2 jar files are deployed –



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	2109328abhisheknec2services2	i-010b301f9aaeafffa	✓ Running	t3.micro	✓ 2/2 checks passed	No alarms +	eu-north-1c
□	2109328abhisheknec2services1	i-0d86b376bfb02d25e	✓ Running	t3.micro	✓ 2/2 checks passed	No alarms +	eu-north-1c

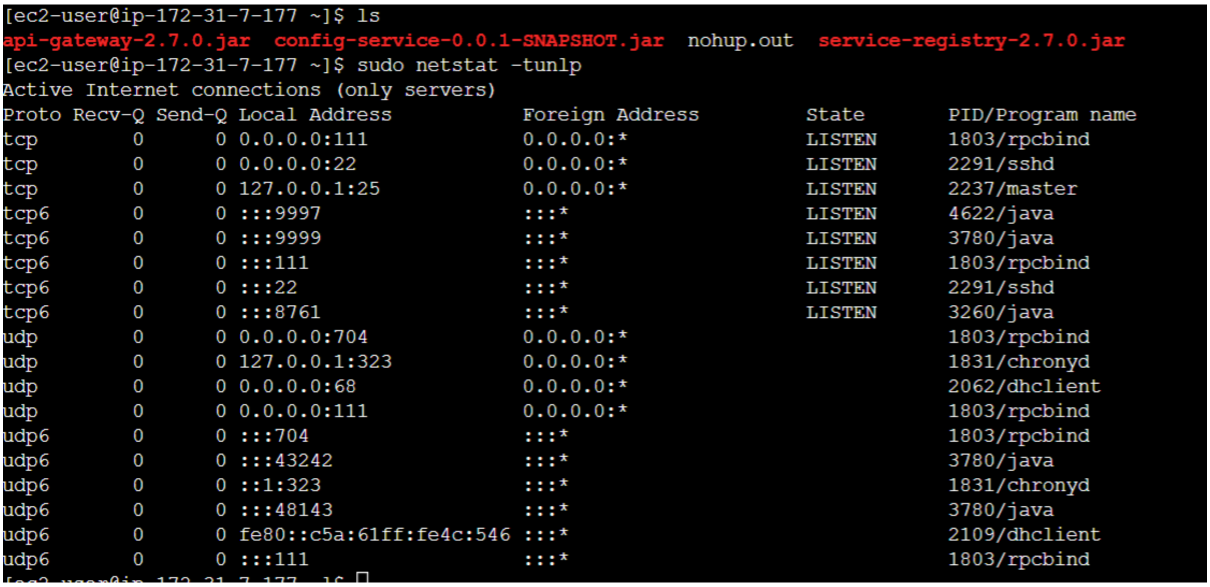
Note- Reason for creating 2 EC2 instance was VM was getting freeze when all jar file deployed in one instance and started.

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



```
[ec2-user@ip-172-31-15-35 ~]$ ls
course-service-0.0.1-SNAPSHOT.jar  nohup.out  user-service-0.0.1-SNAPSHOT.jar
[ec2-user@ip-172-31-15-35 ~]$ sudo netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2401/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      2245/master
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1814/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      2401/sshd
tcp6       0      0 :::9996                 :::*                    LISTEN      4986/java
tcp6       0      0 :::9998                 :::*                    LISTEN      5515/java
tcp6       0      0 :::111                  :::*                    LISTEN      1814/rpcbind
udp        0      0 127.0.0.1:323           0.0.0.0:*                           1845/chronyd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           2068/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1814/rpcbind
udp        0      0 0.0.0.0:708             0.0.0.0:*                           1814/rpcbind
udp6       0      0 ::1:323                 :::*                                1845/chronyd
udp6       0      0 fe80::cb6:26ff:fe5d:546 :::*                                2115/dhclient
udp6       0      0 :::111                  :::*                                1814/rpcbind
udp6       0      0 :::708                  :::*                                1814/rpcbind
[ec2-user@ip-172-31-15-35 ~]$ 
```

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



```
[ec2-user@ip-172-31-7-177 ~]$ ls
api-gateway-2.7.0.jar  config-service-0.0.1-SNAPSHOT.jar  nohup.out  service-registry-2.7.0.jar
[ec2-user@ip-172-31-7-177 ~]$ sudo netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1803/rpcbind
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2291/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      2237/master
tcp6       0      0 :::9997                 :::*                    LISTEN      4622/java
tcp6       0      0 :::9999                 :::*                    LISTEN      3780/java
tcp6       0      0 :::111                  :::*                    LISTEN      1803/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      2291/sshd
tcp6       0      0 :::8761                 :::*                    LISTEN      3260/java
udp        0      0 0.0.0.0:704             0.0.0.0:*                           1803/rpcbind
udp        0      0 127.0.0.1:323           0.0.0.0:*                           1831/chronyd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           2062/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1803/rpcbind
udp6       0      0 :::704                  :::*                                1803/rpcbind
udp6       0      0 :::43242                :::*                                3780/java
udp6       0      0 ::1:323                 :::*                                1831/chronyd
udp6       0      0 :::48143                :::*                                3780/java
udp6       0      0 fe80::c5a:61ff:fe4c:546 :::*                                2109/dhclient
udp6       0      0 :::111                  :::*                                1803/rpcbind
[ec2-user@ip-172-31-7-177 ~]$
```

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/