Self-Assessment and Personalized Learning (SAPL) System

Team Members

Alfeshkhi, Hassan

Islam Riad, A B M Kamrul

Islam, Md Saiful

Kommineni, Prathyusha

IT 7933 Capstone project

KSU & Digi Safari

Digi Safari

Digital Expedition for All

KENNESAW STATE UNIVERSITY

11/25/2020

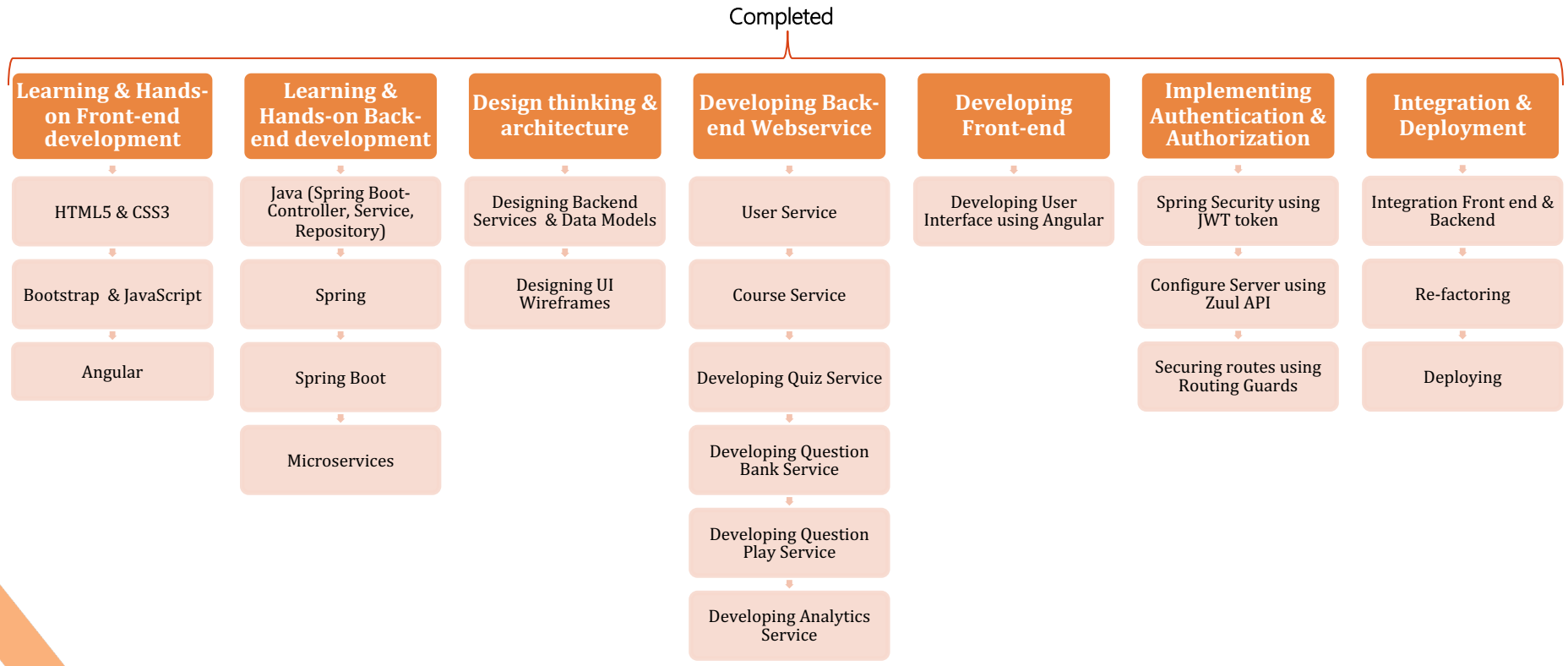
Digi-Safari (DS)

- Digi-Safari (DS) provides software training and services to corporate clients looking to re-skill/ Up-skill existing/ new hires under the B2B (Business-Business) model. DS wants to extend the service for the B2C (Business-Consumer) segment, where individual learners can also directly register for self-paced training.
- They also provide training programs in different universities like us and help students to boost programming knowledge by working in a real-world project. That helps to kick start their career in the IT industry.

Project background

- To better assist learners in assessing their current skill level and pick the best-fit learning path, DS planned to develop a Self-Assessment and Personalized Learning-path (SAPL) System.
- SAPL provide an on-line self-assessment tool through a series of questions, which will auto-select the next question based on response from previous questions equivalent to the GRE type test.
- Based on the performance and candidate's career goal, SAPL automatically come up with Personalized Learning-path. Candidates then have the option of enrolling through Digi-Safari training programs.

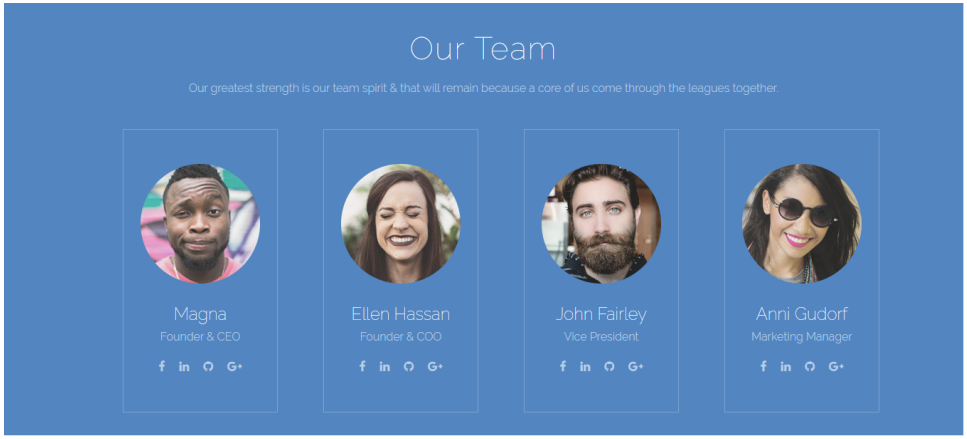
SAPL Project Summary



Learning & Hands-on Front-end development

HTML 5 & CSS3

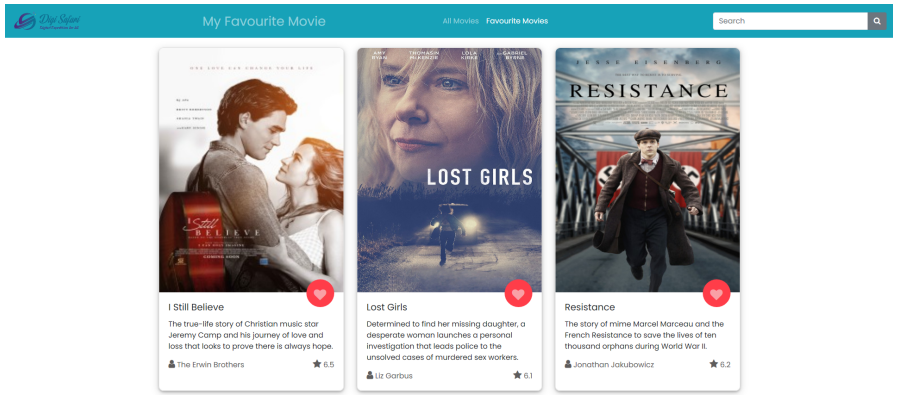
- Web Development fundamentals, Creating Simple web page
- CSS3 Flex Box, Responsive Web Design
- Implement Font-Awesome Icons



Learning & Hands-on Front-end development

Bootstrap & JavaScript

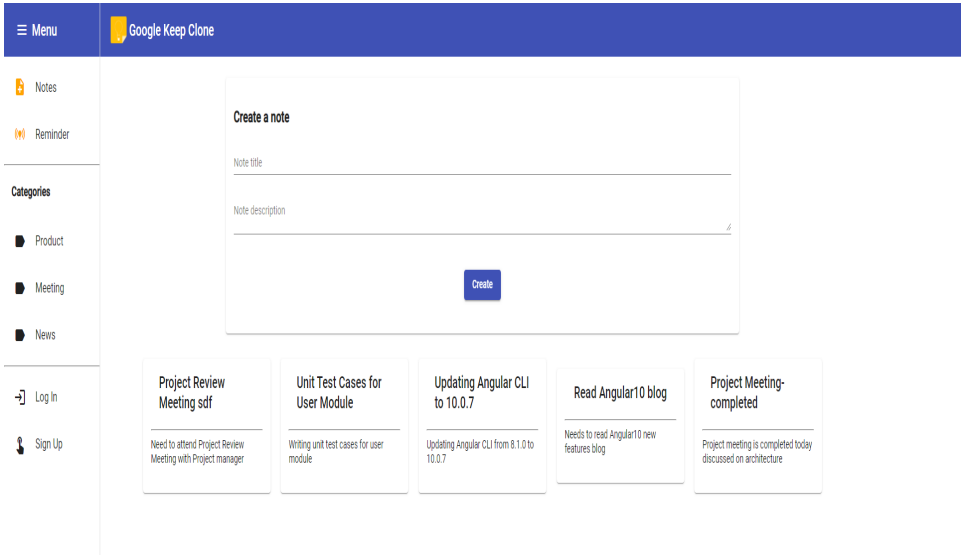
- Bootstrap and its components (NavBar, GridSystem,Form,Card)
- Fetch API with GET, POST, DELETE HTTP requests
- Promise API (usage of resolve, reject, then, catch functions)
- json-server, http-server
- Implement Font-Awesome Icons



Learning & Hands-on Front-end development

Angular

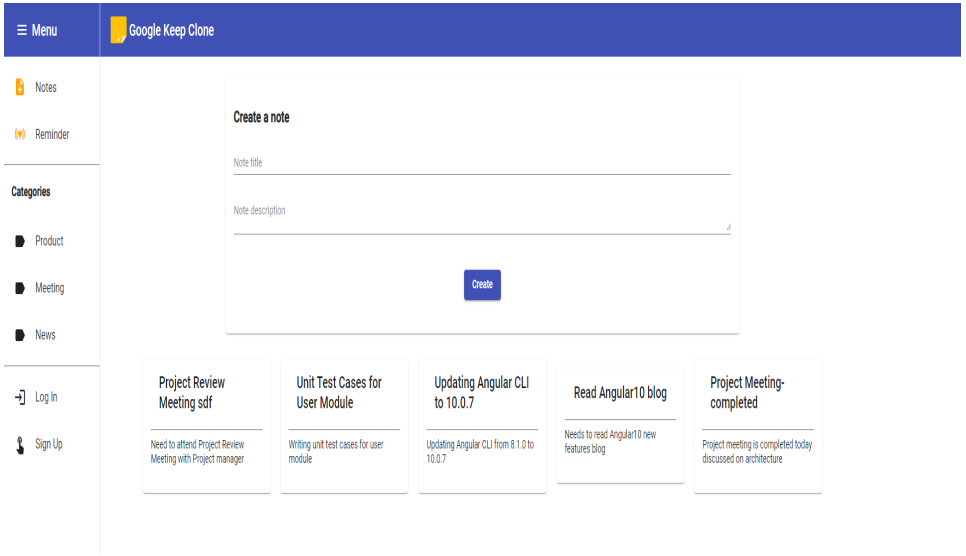
- Working with Angular Architecture, Modules, Components, Template Data Binding, Event Handling, Angular Material
- Working With Services, Working with HTTP API calls, GET, POST,PUT,DELETE,PATCH
- Angular Forms, Template Driven, Reactive Forms, Routing, Guards, JWT Security



Learning & Hands-on Front-end development

Code Review - Refactoring

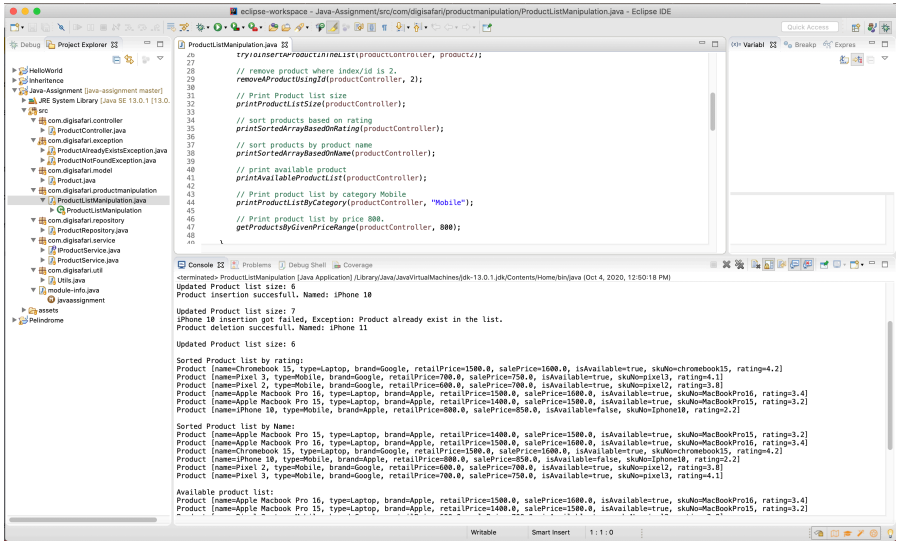
- Code Review on Angular
- Improve readability
- Increase scalability



Learning & Hands-on Front-end development

Java

- Java Fundamentals, Working with Primitives, Control Statements
- Working with classes & Objects, constructors, static members, methods
- Object Oriented Programming Features, Exception Handling
- Working with Collections



Design thinking & architecture of the capstone project

Design thinking & architecture

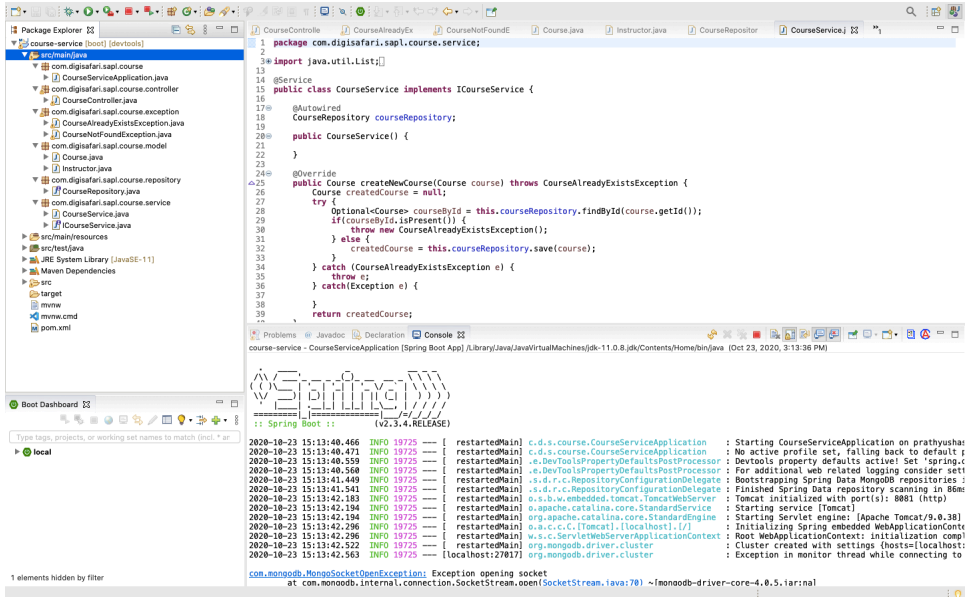
- Design Thinking, Domain Driven Design
- Designing Backend Services, Data Models
- Wireframes

END POINTS	HTTP REQUEST	DESCRIPTION
http://localhost:8084/api/v1/Quiz	GET	Should fetch all the quizzes
http://localhost:8084/api/v1/Quiz/{QuizId}	GET	Should fetch specific quiz with provided id
http://localhost:8084/api/v1/Quiz/{QuizId}/CreateQuiz	POST	Should create a new quiz
http://localhost:8084/api/v1/Quiz/{QuizId}/UpdateQuiz	POST	Should update the existing quiz
http://localhost:8084/api/v1/Quiz/{QuizId}/DeleteQuiz	GET	Should delete the quiz

Learning & Hands-on Back-end development(Tools and Coding)

Spring

- Introduction to Spring Framework, Core Container, Dependency Injection, Spring MVC



Spring Boot

- Testing RESTful Web Services, Introduction to CI, Implementing Gitlab CI pipelines for RESTful Webservices

Learning & Hands-on Back-end development(Tools and Coding)

Spring MVC

- Spring MVC, Developing a monolithic application using Spring – Hibernate

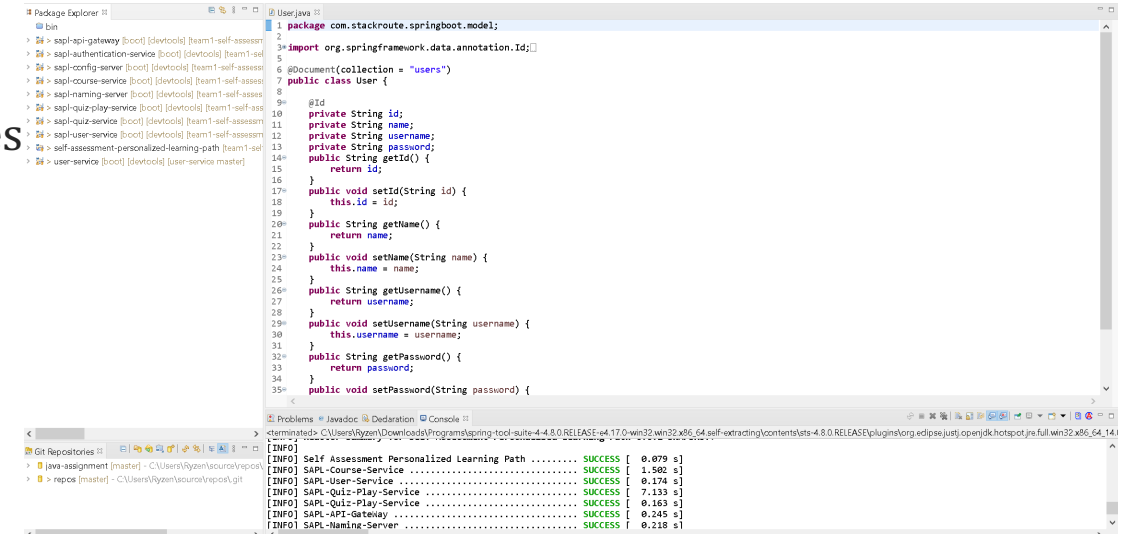
Spring Cloud - Microservices

- Introduction to Microservices, Monolith vs Microservice, Netflix Cloud, Spring Cloud, Eureka Service Registry & Discovery,

Structure of Project Architecture

Developing User and Quiz Service

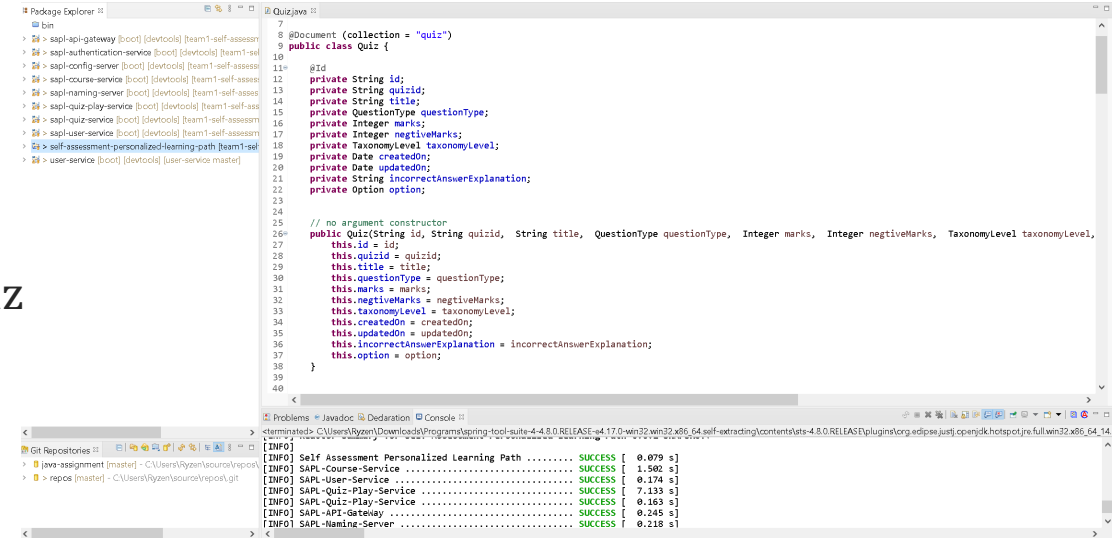
- User Login and Sign Up
- User can browse the courses
- User can see the course topics
- User can attempt the Quiz
- User can see the attempted quizzes
- User can access personalized Learning path based on the quiz result



Structure of Project Architecture

Developing Question Bank Service (Quiz Service)

- Should allow to create quiz
- Should fetch all the created quizzes
- Should allow to update a quiz
- Should inactivate a quiz
- Should fetch the details for the quiz
- Should fetch the statistics of the quiz



Developing Back-end Webservice

Developing Quiz Play Service

- A user should be able to take / attempt quiz for any particular course
- A user should be able to see his score, preview his quiz, and the description for the wrong answer.
- User should fetch the next question based on the previous answered question(It should increase/decrease the bloom's taxonomy level

Authentication Service

- Should implement user authentication & authorization

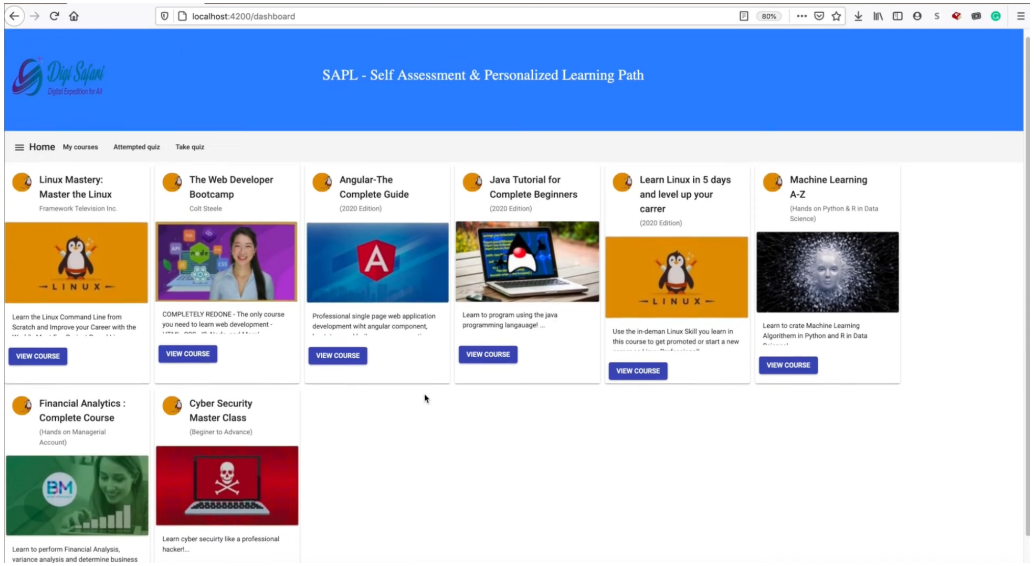
Course Service

- Should allow the CRUD operation on the course

Developing Front-end Interface

Developing UI using Angular part 1

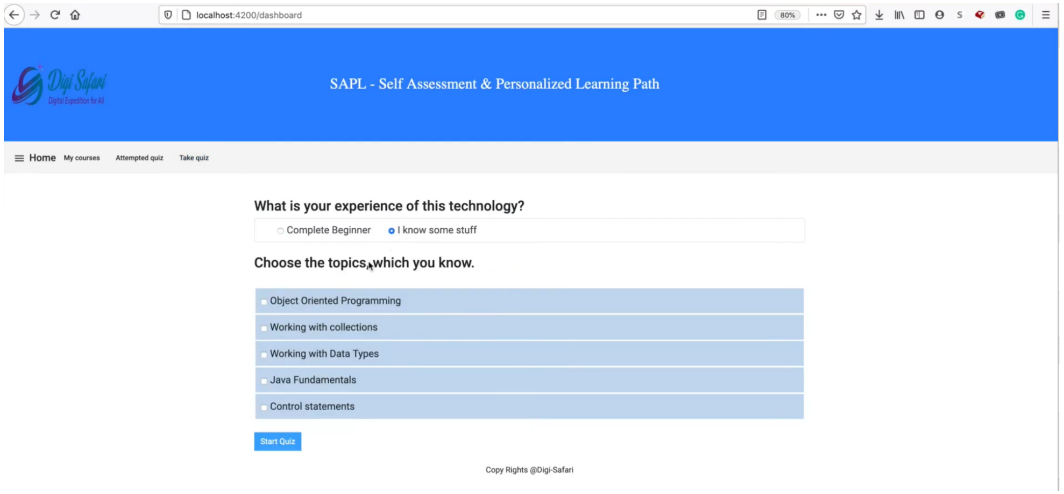
- Developing services to consume backend Quiz Service end points
- Developing services to consume backend Question Bank Service End points
- Developing the Front-end UI screen to replicate the functionality of quiz and quiz play services



Developing Front-end Interface

Developing UI using Angular part 2

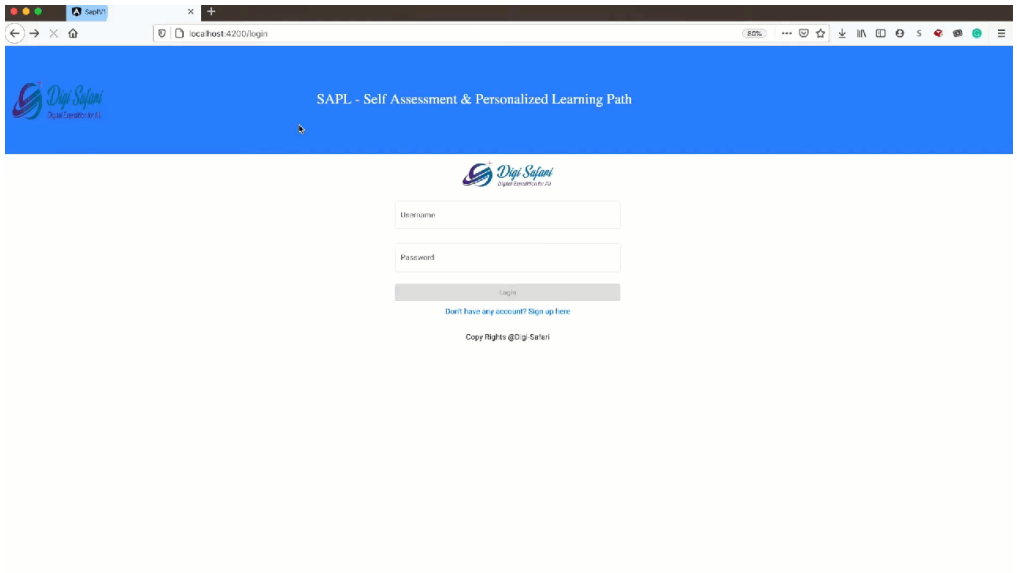
- Developing services to consume backend Quiz Play Service end points
- Developing services to consume backend Analytics Service End points
- Developing the Front-end UI screen to replicate the functionality of Quiz play and Analytics services



Implementation

Implementing Authentication & Authorization

- Implement authentication service using JWT token
- Implementing Authentication & Authorization using Zuul API Gateway
- Securing routes in Angular using Routing Guards

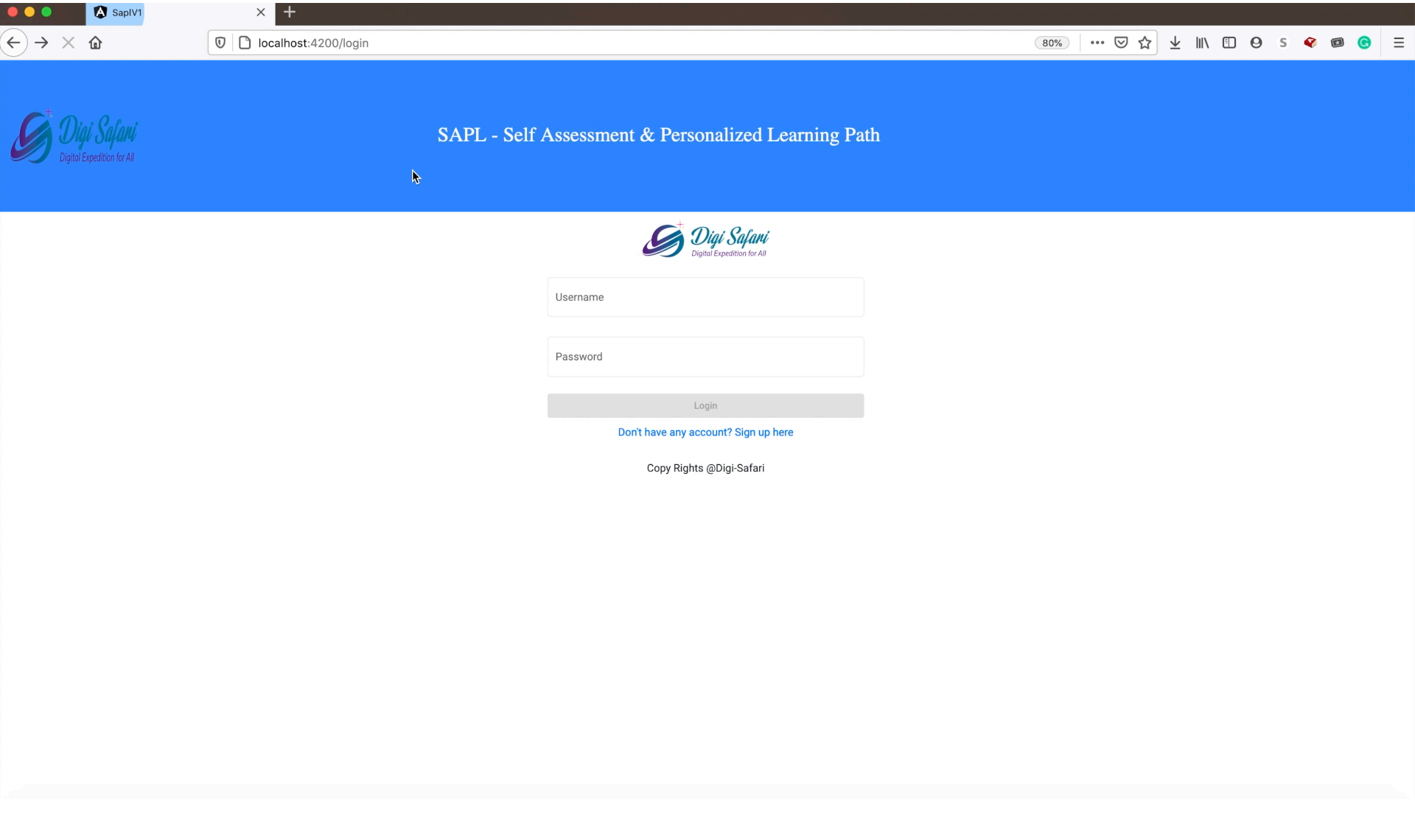


Integration & Deployment

Integration & Deployment

- Implement unit testing and e2e testing for the whole application and automating through Gitlab CI pipelines
- Containerize the application using Docker
- Finally, package and deploy the application to Kubernetes

Project Video Demo



Thank You!