Publicis Sapient - Angular_SDE II

Test Summary

- No. of Sections: 2
- No. of Questions: 2
- Total Duration: 60 min

Section 1 - SWOT

Section Summary

- No. of Questions: 1
- Duration: 10 min

Additional Instructions:

None

Q1. Fill it in with your SWOT to fill up this role in the organization. (Eg. Your strength could be your experience, Threats could be moving to a new organizational culture, and so on.) Upload your filled-in SWOT.
Download the following sheet: https://bit.ly/doc-swot

File Count: 1

Allowed file format(s): .pdf .jpeg .jpg .doc .docx .zip

Section 2 - Free Form Coding

Section Summary

- No. of Questions: 1
- Duration: 50 min

Additional Instructions:

None

Q1. **Problem Statement 1**
If you have to build an application similar to **www.bookmyshow.com** for the theatres within the city, and each theatre would have at least 2 screens what would be the approach?

1. Please define the use cases considered
2. API's to be defined as GET, POST, PUT, Delete
3. Theatre contains id, no_of_screen, Movie Class, location.
4. Movie class contains id, name, cast, date.
5. An average load of 2 screens in each theatre.
6. Consider the worst case of 10000 users trying to book the seats for each theatre
7. High-level performance-oriented architecture which you will design and deploy

Problem Statement 2
What are the components you would configure if the above application has to be hosted in Google Cloud with autoscaling and cost in mind?

Method – Routes – Details

- GET - /allmovies – Array of movies list
- GET - /alltheatres – Array of theatre list
- GET - /theatre/{id} – Theatre along with movie or Theatre not found
- POST - /addtheatre – Add Theatre details with movies
- PUT - /updatetheatre/{id} – Update theatre details
- DELETE - /deletetheatre/{id} – Remove theatre by id or Theatre not found.

Problem Statement 2
What are the components you would configure if the above application has to be hosted in Google Cloud with autoscaling and cost in mind?

Front End

- Angular 9+
- Google Material Design
- Bootstrap / Bulma

Server Side
- Spring Boot
- Spring Web (Rest Controller)
- Spring Hibernate

Core Platform
- OpenJDK

Database
- MySQL

DB Access
username: root
password: examly
cmd: mysql -u root --protocol=tcp -p
Create Database test.

NFRs
- Demonstrate SOLID and HATEOAS principles, Design Patterns in the design and implementation
- Demonstrate Performance, Optimization & Security aspects
- Demonstrate Production readiness of the code
- Demonstrate TDD & BDD & Quality aspects
- Demonstrate sensitive information used in the Micro Services such as API keys are protected/encrypted

Documentation
- Include the open-API spec to be part of the code. (Documentation to explain the purpose of the API along with Error codes that the service consumers & client must be aware of!)
- Create a README.md file in the project folder and explain the design and implementation approach

Instructions regarding IDE
Navigate to Book Store react folder
Start working on the front-end in the angularapp folder and back-end in the springapp folder.
Click port 8080 and use the URL as an Endpoint.
In the Front-end, Instead of localhost:8080 click port 8080 inside the platform and use that URL.
To connect to MySQL database:
- Inside the IDE, click on the Hamburger menu on the left top corner and click file new terminal.
- In the terminal, use the following command to connect to the database
- **mysql -u root --protocol=tcp –p**
- **Password: examly**
- Create Database bookstore.

To run and execute the project:
After completing the entire source code without any errors. Use the following command in the terminal to run your project
- **mvn install –DskipTests**
- **mvn spring-boot:run**
- To check the project, click **port 8080** inside the IDE and use the URL as an Endpoint.
  Open another Terminal and navigate to react folder.
- Run **npm install**
- Run **npm start**
- To check the project, click **port 8081** inside the IDE.

To run the testcases and submit the project:
- After the above step, click the run testcase button inside the IDE.
- Once the testcase returns **SUCCESS** status, you can submit the project.

Answer Key & Solution

Section 1 - SWOT

Q1 Solution cannot be displayed

Section 2 - Free Form Coding

Q1 Solution cannot be displayed