

Assignment Code: SE01

Last updated: 28-11-2023

Introduction

Welcome to the technical assignment with Microservices using .NET/Node and RDBMS. This assignment is designed to assess your skills in building microservices architecture, and database integration. Your task is to develop a user authentication and registration system for a web application using a modern web framework.

Assignment Overview

Your goal is to create a fully functional Web application with microservices. The app should provide the following key features.

- ❖ **User Authentication:**
 - Allow users to register and log in securely.
 - Implement JWT-based authentication for user sessions.
- ❖ **User Registration form:**
 - Create a simple web based registration form that accepts username and password, atleast one document upload field, and date field.

Technical Stack

You are required to use the following technologies and tools:

- ✓ Utilize .NET with ASP.NET Core or Node JS for building Microservices.
- ✓ An RDBMS of your choice (e.g., SQL Server, My SQL or Maria DB) for data storage.
- ✓ Docker for containerization (recommended).
- ✓ Any frontend framework of your choice (HTML, CSS, JavaScript, jQuery, Angular, React, or Vue.js) for the user interface.

Instructions

- ❖ **Set up the project:**
 - Organize the solution into separate projects for microservices (Authentication, User Registration, and update).
- ❖ **Design the Microservices Architecture:**
 - Identify the core functionalities of the assignment, such as user Registration, authentication, login)
 - We are expecting atleast 2 microservices here.
 - Registration web service–
 - RegisterUser() :Front end application will send all the user entered registration details, validates the data, send to Data process webservice.
 - AuthenticateUser(): Login validation to check whether given username and password is matching. Connect to Data process web service and check whether user exists in the database.
 - Data process web service: This will connect to any of the RDBMS and store/retrieve data from databased.
 - Design individual Microservices for each functionality, ensuring clear separation of concerns.
 - Define clear API contracts (RESTful APIs or gRPC) for communication between microservices.
- ❖ **Front-end Development:**

- Build the front-end app using any frontend framework of your choice (HTML, CSS, JavaScript, jQuery, Angular, React, or Vue.js)
- ❖ **Connect Front-end with Microservices:**
 - Implement client-side code to interact with the backend Microservices via RESTful
- ❖ **Security:**
 - Implement secure communication between the client and Microservices using HTTPS
 - Implement authentication and authorization mechanisms to protect sensitive operations.
- ❖ **Testing and Quality Assurance:**
 - Write comprehensive unit tests (using unit testing framework) and integration tests for each Microservice and the front-end.
 - Perform end-to-end testing to verify the complete functionality of the chat app.
- ❖ **Monitoring and Logging:**
 - Set up monitoring and logging to track the performance and health of the Microservices.
 - Implement error handling and logging to facilitate troubleshooting.
- ❖ **Deployment:**
 - Deploy the Microservices and front-end in local docker
- ❖ **Documentation:**
 - Create detailed documentation on how to build, deploy, and use the app with Microservices, and RDBMS.
 - Document your code, APIs, and any important decisions or assumptions made during development.
 - Provide clear instructions on how to build and run the application locally.

Add-ons (Bonus)

- ❖ **Testing and Quality Assurance:**
 - Perform load testing to ensure the app can handle many concurrent users.

Submission

Please submit your completed assignment within **seven** days. You can share your code repository link or a zip file with the necessary code and documentation.

Evaluation Criteria

Your assignment will be evaluated based on the following criteria:

- ❖ **Functionality:** Does the application meet all the required features and functionalities?
- ❖ **Code Quality:** Is the code well-structured, maintainable, and follows best practices?
- ❖ **Microservices Architecture:** Is the microservices architecture properly designed and implemented?
- ❖ **Database Integration:** Is data stored and retrieved from the RDBMS accurately?
- ❖ **User Experience:** Is the frontend user interface intuitive and responsive?
- ❖ **Unit Testing:** Are there sufficient test cases, and does the application pass them?
- ❖ **Documentation:** Is proper required documentations provided?
- ❖ **Add-ons (Bonus):** Any "add-ons" attempted, creativity, innovation, and implementation of any additional features are considered as extra bonus points.

Conclusion

Thank you for taking the time to complete this technical assignment. We look forward to reviewing your work and discussing your solution in the next interview round. If you have any questions or need clarifications, please reach out to **techarch@tetherfi.com**.