



Arbetstest

FULLSTACKUTVECKLARE

PACOM GROUP AB

Table of Content

Arbetstest för Fullstackutvecklare med C# och Blazor	2
<i>Uppgift: Bygg en webbapplikation för att hantera en Modbus-simulator.....</i>	<i>2</i>
<i>Krav:</i>	<i>2</i>
Backend:	2
Frontend:	2
Modbus Integration:.....	2
Funktioner:	2
Bonus:.....	2
<i>Bedömningskriterier:</i>	<i>3</i>
<i>Dokumentationsuppgifter:</i>	<i>3</i>
XML-kommentarer:	3
Kodkommentarer:	3
README-fil:	3
<i>Modbus Protokoll och Simulator:</i>	<i>3</i>
<i>Referenser:</i>	<i>3</i>

Arbetstest för Fullstackutvecklare med C# och Blazor

Uppgift: Bygg en webbapplikation för att hantera en Modbus-simulator

Krav:

Backend:

- Använd ASP.NET för att skapa en RESTful API.
- API:et ska kunna hantera CRUD-operationer (Create, Read, Update, Delete) för att spara värden i en SQL-databas.
- Använd Entity Framework Core för att interagera med databasen.
- Dokumentation: Lägg till XML-kommentarer i koden för att beskriva metoder, klasser och egenskaper.

Frontend:

- Använd Blazor för att skapa en interaktiv och användarvänlig gränssnitt.
- Skapa komponenter för att visa en på/av-knapp och en indikator som visar om värdet är på eller av.
- Använd HttpClient för att kommunicera med backend-API:et.
- Dokumentation: Lägg till kommentarer i koden för att beskriva komponenter och deras funktioner.

Modbus Integration:

- Integrera med en Modbus-simulator för att kunna ändra och läsa värden.
- Använd NModbus-biblioteket för att hantera Modbus-kommunikation.
- Dokumentation: Skapa en README-fil som beskriver hur man använder applikationen, inklusive hur man startar den, hur man interagerar med gränssnittet och hur man använder Modbus-simulatorens.

Funktioner:

- Användare ska kunna ändra ett värde via en på/av-knapp i frontend.
- En indikator i frontend ska visa om värdet är på eller av, baserat på data från databasen.
- Värdet ska sparas i databasen och hämtas därifrån för att visa indikatorns status.
- Dokumentation: Beskriv hur man använder applikationen i README-filen.

Bonus:

- Implementera autentisering och auktorisering med JWT (JSON Web Tokens).
- **Lägg till enhetstester för både backend och frontend.**

Bedömningskriterier:

- Kodkvalitet och struktur.
- Användning av bästa praxis för både backend och frontend.
- **Funktionalitet** och användarupplevelse.
- Dokumentation och instruktioner för hur man använder applikationen.

Dokumentationsuppgifter:

XML-kommentarer:

- Lägg till XML-kommentarer i backend-koden för att beskriva metoder, klasser och egenskaper.

Kodkommentarer:

- Lägg till kommentarer i frontend-koden för att beskriva komponenter och deras funktioner.

README-fil:

- Skapa en README-fil som beskriver hur man använder applikationen, inklusive hur man startar den, hur man interagerar med gränssnittet och hur man använder Modbus-simulatorn.

Modbus Protokoll och Simulator:

Modbus Protokoll: Modbus är ett kommunikationsprotokoll som används för att ansluta elektroniska enheter. Det finns olika varianter som Modbus RTU och Modbus TCP.

Modbus Simulator: Använd ModRSim2 eller Modbus Poll för att simulera Modbus-enheter.

Referenser:

- [1] Modbus Organization. "About Modbus." Available at: <https://modbus.org>
- [2] NModbus Library. "NModbus - A C# implementation of the Modbus protocol." Available at: <https://github.com/NModbus/NModbus>
- [3] Microsoft Docs. "ASP.NET Core." Available at: <https://docs.microsoft.com/en-us/aspnet/core/>
- [4] Microsoft Docs. "Blazor." Available at: <https://docs.microsoft.com/en-us/aspnet/core/blazor/>