Sana Tech

Developer Technical Test

sana

Introduction

Welcome to our Sana Developer recruitment process. As part of this phase, you will be developing a practical assessment that will help us to have an overview of the best of your experience as developer.

The assessment is simple in terms of business logic, but please keep in mind that this is on purpose! The specific business requirements define for you the conditions to accomplish, but beyond seeing your code work, we look forward to evidence the best of your developer skills in the implementation of each on them.

Enjoy coding and good luck!

Instructions

- Carefully read the assessment and develop the indicated requirements.
- For any doubt or question, please contact the Sana recruiter, we will answer asap.
- For submitting your results use GitHub, One Drive or Google Drive. Avoid sending it in temporal link or WeTransfer.

Tips

- Read carefully.
- Please start from scratch with the assessment, so we can see all the choices you made concerning the assignment. Nonetheless, if you use a template, please explain which one you used.
- If you must do any concessions, please explain in code comments what you would do in a real-life situation.
- Keep in mind that the main goal of this exercise is to show the best of your knowledge and experience with the indicated technologies.

Specifications

Estimated Time for implementation: **6-8H**

- The Frontend side of the application should be implemented using React (Recommended to use the last version, but not required). You can use JavaScript or Typescript.
- The Backend side of the application should be implemented using .Net Core (Recommended to use the last version, but not required)
- The connection between backend and frontend can be built using a common API. However, it will be a plus if you can use a GraphQL implementation (optional, not required).
- It is required to use Visual Studio or VS Code as dev environment.
- It is required to use as data storage a SQL Server Database.
- Please use English for all code-named variables.

Assessment

Part 1: Database Design

Design a database to store data for a web shop with product categories, products, customers, and orders. Please make sure it contains the following requirements:

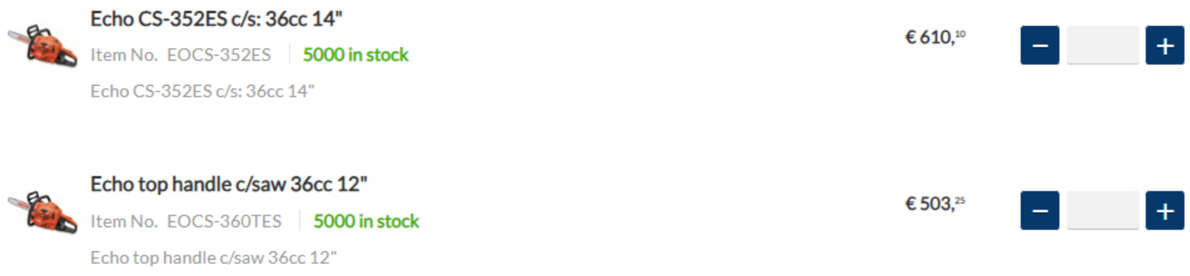
- Products can be related to multiple categories.
- Customers can have multiple orders.
- Orders can have multiple products.

For this part of the assessment, we expect as result a database schema including table relations and indexes. It can be a screenshot of the database schema, or a database diagram built in a tool of your choice.

Part 2: Application Development

Using the database that you designed in Part 1, create a website using React (Frontend) and .Net Core (Backend) to simulate the basic functionality of a web shop. The following sections should be implemented:

1. **Catalog:** A web page where a list of products is displayed showing basic information of each one and enabling the user to add a defined quantity of the product to a shopping cart. The following operations should be supported:
   - Listing of products: Displaying title, Product Code, small description, price, and available stock.
   - Shopping Cart: The user should be able to add a specific quantity of the product (determined by the user through a numeric text box) to a virtual Shopping Cart.
   - Paging: A maximum of 10 products per page.
   - Shopping Cart storage: You should store the products added by the user, however you should use a different type of storage than Database, like session, memory, etc.
   - Stock Validation: Before adding the defined quantity to the shopping cart, you should validate if there is stock available.



Echo CS-352ES c/s: 36cc 14"

Item No. EOCS-352ES | 5000 in stock

Echo CS-352ES c/s: 36cc 14"

€ 610,^{10}

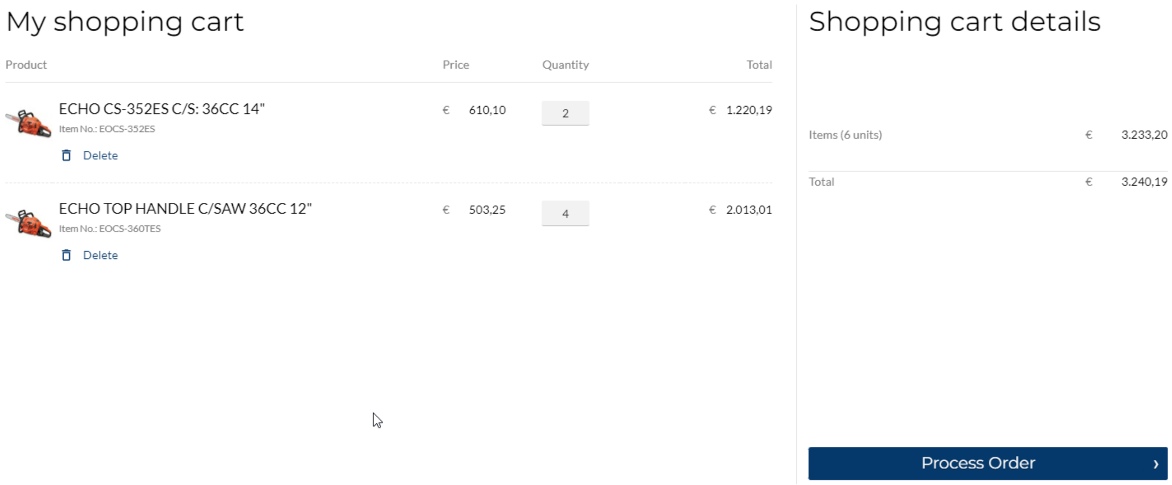
Echo top handle c/saw 36cc 12"

Item No. EOCS-360TES | 5000 in stock

Echo top handle c/saw 36cc 12"

€ 503,^{25}

2. **Shopping Cart**: A web page where the list of products added for purchase is displayed. The list should show basic information of the product and being able of editing quantity and processing an order. The following operations should be supported:
    - Listing of products: Displaying title, Product Code, Quantity, Price.
    - Totals: Displaying the total price by line (product x quantity), Displaying a summary of the total of the entire order.
    - Quantity Editing: The user should be able of editing the quantity of each product line.
    - Deleting: The user should be able of deleting a product from the list.
    - Process Order: The user should be able to create an order with the list of products that are part of the shopping cart.



My shopping cart

Product	Price	Quantity	Total
ECHO CS-352ES C/S: 36CC 14" Item No.: EOCS-352ES Delete	€ 610,10	2	€ 1.220,19
ECHO TOP HANDLE C/SAW 36CC 12" Item No.: EOCS-360TES Delete	€ 503,25	4	€ 2.013,01

Shopping cart details

Items (6 units)	€	3.233,20
Total	€	3.240,19

For this part of the assessment, we expect as result the next assets:

- **Source Code:** Appreciated if you can upload it in GitHub and share it from there.
- **Database:** Scripts with the table schema and data generated from SQL Server.
- **Notes:** Additional notes from your side for making easier the setup of your test for our experts.

Prioritize relationships, not just transactions.

E-commerce for SAP and Microsoft Dynamics.

Visit **www.sana-commerce.com**

Microsoft Partner

Microsoft

Gold Data Analytics
Gold Enterprise Resource Planning
Gold Application Development
Gold Cloud Platform
Silver Datacenter

SAP Partner

sana