Full Stack coding challenge

Welcome fellow coder!

Introduction

You have chosen to undertake the Full Stack Challenge. This is a challenge and not a test! It is not about passing or failing, but rather about us giving you a clean canvas that you can use to demonstrate how passionate you are about coding. Thus, go forth and prove to us that you have what it takes to survive the everyday life as a code wielding bug slayer!

Getting Started

To get started, please create a new public repo and add commit your code there, remember to make regular commits! Once you have finished the challenge or you have ran out of time, please send us a link to your repo.

Before you start

A few things that you should know before you start:

- You need to write both the Front-End and Back-End code

- Please read through the entire challenge before starting.

- Typical development time is approximate 4 hours. Sooner is better.

- Please do frequent and well commented commits.

- Commenting your code is not mandatory, but will be appreciated.

Full stack requirements:

- Please use frontend framework based on Web components such as React, Angular.

- For the Back-End, please use C# (.Net Framework/.Net Core) or Java (Spring Boot) to create your API.

- Create separate build for Frond end and Back end.

- Each phase below will require you to write Front-end code as well as Back-end code.

- The data provided needs to be stored in a NoSql or any Database of your preference.

Challenge description:

Background

So, I have a friend called Frank. He owns a second hand car garage and needs your help!

He would like you to create a website for his shop. These days Frank only sells his second hand cars online and could use a website of his own. With that said I guess we should do this in phases, because you know...agile! So it would be best if you start with Phase1!

Phase 1:

Start by creating a page that will display all of the cars that Frank has across all of his warehouses. Please sort the list according to _date_added_ asc.

Create one GET API to fetch the list of cars. Mock JSON reference (https://api.jsonbin.io/b/5ebe673947a2266b1478d892)

Phase 2:

Now, allowing the user to click on any of the cars (that are licensed = true) would make it a lot nicer. Once the user clicked on a car, we then show more details such as, the warehouse where it is stored and its location perhaps? It is up to you how detailed you want to make it.

Phase 3:

Wow! Okay, we now have some good functionality! So let's take it one step further! Let's allow the user to add the car he is viewing to some sort of shopping cart so that he can easily checkout once he is done shopping. Oh, and perhaps we should show the user the total amount as well?

Expectation:

- Project should be in working order.

- Unit test coverage for both frontend and backend.

- Application addresses the basic security vulnerabilities.

- Well-structured and clean code.

- Documentation to run project.

Summary:

By no means is the goal to get a solution covering all special cases in a 100% robust way. Requiring this would be naive given the time limit!

It is important that you can explain why you chose your solution instead of another.

At least the functionality that you deliver should be error free.

What you implement and how you do it is subject to your creativity and ambition ☺

Good luck!