KOMPARE full-stack TASK

Introduction

Your objective is to develop a basic full stack application in which users can enter customers data and calculate the insurance price.
Additionally, users of the application must be able to select additional options and discounts.
The insurance price is calculated on the backend.

Example use case

The sales agent is currently working in their office when a customer walks in, seeking information about insurance prices for their vehicle. The agent proceeds to input the necessary data into a simple form. Once the data is submitted, the sidebar and header sections of the application needs to be updated.

- in the sidebar section, the available coverages are displayed
- in the header section, the available discounts are displayed
    - additionally, the header section also presents the total price
- below the main form, the price details are displayed

Implementation

Application mockup is at the end of file.

1. Frontend

In the given scenario, the form data will be applied upon clicking the "Save" button. However, any changes made to the checkboxes in the header and sidebar sections should be applied immediately upon clicking them.

Moreover, it's important to ensure that any changes made in one section of the application are reflected in the other sections as well. This means that modifying a checkbox in the header or sidebar should update the linked information in the other sections, ensuring consistency throughout the application.

1.1. Main form

The main form must contain a form with the following fields (all fields except "Voucher" and "Price match" are required):

- Name
- Birthdate
- City
- Vehicle power
- Voucher
- Price match

Price details are displayed below the form, and include the following:

- Basic price (without all discounts and coverages)
- List of discounts (if any are applied)
  - for every discount, amount in EUR must be displayed
- List of coverages (if any are applied)
  - for every coverage, amount in EUR must be displayed
- Total price

1.2. Header

The header must contain a list of available discounts which can be selected, along with the display of the total price.

1.3. Sidebar

The sidebar must contain a list of available coverages which can be selected.

2. Backend

To ensure the necessary functionality, implement the following for the back-end:

- Store all configurations, including coverages, discounts, and additional options, in the database.
- Create an endpoint that receives the necessary customer data and the selected coverages, discounts, and additional options. This endpoint will calculate the insurance price based on a combination of the base price, selected additional coverage prices and selected discounts/surcharges.
  - The base price is calculated based on the city and age of the customer.

The coverages and discounts are typically accessible to all customers, with a few exceptions, and they can impact the price in various ways. The specific rules regarding their availability and application are outlined as follows:

Additional coverages:

Bonus Protection - 12% of the base price
AO+ - Fixed price:

- 55 EUR for users younger than 30 years
- 105 EUR for users that are 30 years old or older

Glass protection - 80% of the vehicle power

Discounts/surcharges:

Commercial discount - 10% on the base price
Adviser discount - 20% on all coverages - if at least 2 coverages are selected
VIP discount - 5% on the total price - show only if vehicle power over 80
Strong car surcharge - + 10% if vehicle power over 100 - applied automatically (can't be turned off)
Voucher - a user entered price that discounts the total price

Price matching:

The price matching field accepts a numerical value and ensures that the new total price matches it. The total price is adjusted by modifying the base price, disregarding the vehicle power and user age parameters. Instead, the new base price should be calculated by factoring in the currently active coverages and discounts, ensuring the new total is equal to the price specified in the input field.

Deadline for submission

The completed task must be submitted within 10 days from receiving.

Further hints

It's not necessary to create an authentication or authorization system. Users can just be anonymous.

Required technologies

- Typescript (strict type checking: no any and no ts-ignores)
- React (hooks, function components, state management of choice)

- Css framework of choice
- NodeJs
- MongoDB
- REST api or Graphql

Extra points

While these things are not mandatory, their completion would bring us pleasure. Feel free to undertake them at your own pace without any pressure.

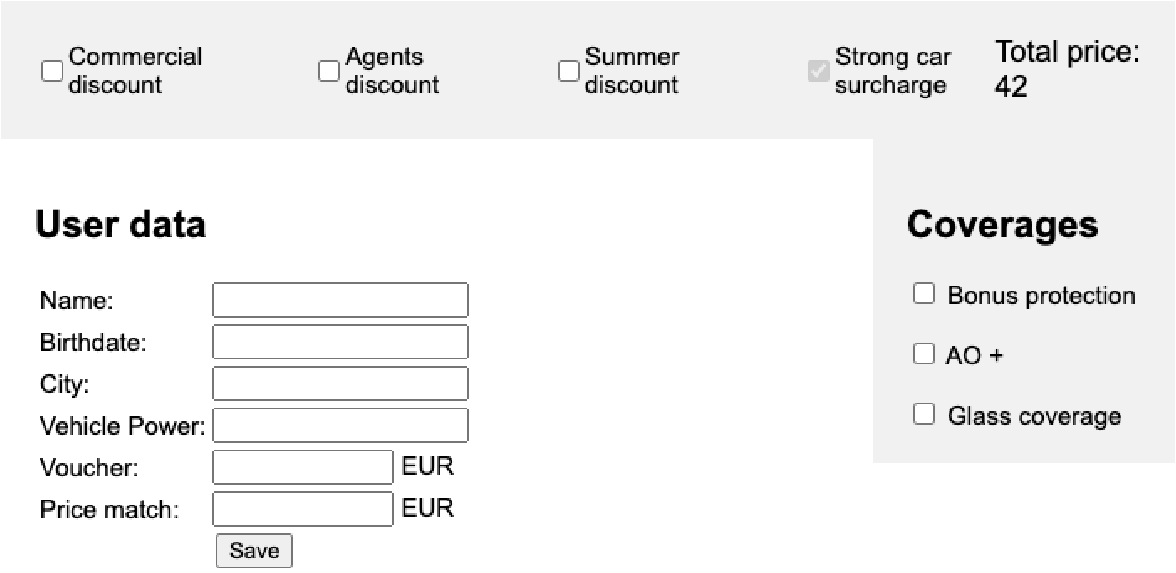
- Unit tests (library of choice)
- Log the data to database and present it on a separate route for display

Evaluating Your Solution

The following criteria serve as our evaluation guidelines for assessing the quality of your solution:

- The application behaves correctly (as specified above)
- The application shall be easy to install & run in development mode. Prefer an embedded database & provide instructions on how to run your application.
- Deliver production-quality code

Application mockup



☐ Commercial discount ☐ Agents discount ☐ Summer discount ☑ Strong car surcharge

Total price: 42

User data

Name:	
Birthdate:	
City:	
Vehicle Power:	
Voucher:	EUR
Price match:	EUR

Coverages

☐ Bonus protection

☐ AO +

☐ Glass coverage