Technical Assessment

Objective

Using the starter application provided in this link, design and build a containerized Ruby on Rails application that calculates and displays premiums for 3 products:

The primary competency we are assessing is **attention to detail.** The role involves working with tasks that may have incomplete or conflicting requirements. As such, it's crucial to:

- Deliver a **well-tested**, reliable solution.
- Use RSpec and Playwright (or alternatives if preferred) to demonstrate basic test coverage of core functionality — full coverage isn't required.
- Focus on writing **clean, reusable, maintainable**, and well-documented code.
- Focus on optimizing the **user experience** based on the target audience's needs and interactions. You're welcome to suggest additional features that enhance overall usability.
- Please provide **clear and concise instructions** for running your application.
- AI detection tools will be used so this assessment should **not be done with AI**.
- If you are running out of time with completing this assessment, please add in **pseudo code** to demonstrate your intentions.
- We expect the quality of the technical assignment to be delivered at a **business production level**.
- If you are unable to complete certain aspects of the assessment, please provide comments and pseudo code of what you intended to do.

Requirements

1. User Input Form

- Create a user interface to collect the mandatory travel details from the customers trip that is relevant to the policy pricing factors.

2. Form Validation

- An Adult age is recognized as **21 years old**. A child is recognized as under 16 years old requires an adult to travel with them on the policy. **Children travel for free.**
- Trip start and end dates must fall within a maximum total period of 2 years, calculated from the trip start date. Trips may be booked up to **18 months in advance**.
- For single trip policies, the maximum trip duration is 365 days, with the start date counted as day one.
- Implement appropriate validation related to all fields if you feel they are necessary.

Final Premium:

Final Premium = Base Premium × Cruise × Snow

3. Premium Display and Additional Add-ons

- After form submission, redirect to a page showing:
  - Selectable add-ons for:
    - **Snow & Cruise**: Should be able to apply a fixed amount onto the base premium on a **per traveller basis premium**
- Prices for all three cover levels should update instantly when relevant pricing factors have been changed.

4. Styling

- Apply CSS styling for forms and content display. Focus on usability and clarity.

Database

For the database, the test data is already in the starter application provided. Please use suitable commands and approach to build these database tables locally to progress with the test.

Test data is in **.**yml file at **test/fixtures**

Policy Calculation Logic

To generate the premium price for each product, use the following formula:

Base Premium: (Base X Excess × Age × Duration × Destination × Trip Type × Level of Cover)

Final Premium: (Base Premium + Cruise + Snow)

Pricing Factors properties to use:

Field	Description
Trip Type	The options will be either **One Way** or **Return**
Age	Max age we allow for travellers is **84 years old**.
Trip Start Date	-
Trip End Date	-
Duration	-
Destination	The application should support multiple destination selections mapped to the destinations table. Invalid or unknown countries

	must be handled gracefully with appropriate errors. Where multiple destinations apply, the highest zone number determines the multiplier used in pricing.
Excess	-

Additional Premium

The Snow and Cruise additional premiums should be displayed as selectable add-ons on the **screen after the form is submitted**, where the three levels of cover are shown.

Cruise	An additional charge must be applied to the base premium based on the destination with the highest zone number selected from the destinations table. The cruise add-on amount should be applied **per traveller** on the quote.
Snow	When this add-on is selected, users must specify a skiing date range. The additional premium is calculated by multiplying the number of ski days by the per-day ski rate for the highest destination zone and is applied **per traveller**.

Submission

Once you have completed your technical assessment, please clone the starter template from the provided link, create a GitHub repository containing your solution, and send it too john@goinsurance.com.au and simon@goinsurance.com.au as collaborators for review.

Alternatively, you may share your completed assessment as a ZIP file via OneDrive or another file-sharing service of your choice.