ARGO DIGITAL: CODE CHALLENGE

Hello!

Argo Digital is committed to providing engineering candidates with the best possible opportunity to demonstrate their skills and experience.

This code challenge has been designed to mimic the kind of work, and technology stack, that you might be expected to work on and with in our business. It consists of a few progressive components. It is not a hard requirement that you complete all the components and **we do not expect you to spend more than 2 hours on this.** At the same time we would urge you to use this opportunity to demonstrate what you are capable of.

Your code will undergo a thorough review once submitted and you should be prepared to discuss your approach if you are subsequently invited to visit us for an in-person tech interview.

Good luck!
Andrew Breen
SVP Digital
Argo

PROBLEM STATEMENT

This is your first day at XYZ Insurance Inc. The company just set their targets for the year and they are looking to expand their product portfolio.

One of the products your team is tasked with building is called Xavier, an online "quick quoting" system. This will be a place where brokers can get a quick quote for private jet liability insurance on behalf of their clients.

You want to get something to market as quick as possible. The first step is to put up a landing page.

Technical Requirements

1. React SPA client (hint: use a scaffolding tool to save time)
2. State management library of your choice (if necessary)
3. Include relevant unit tests

Delivery Requirements

1. Include a README file with clear instructions for how to build, test, and run your project.
2. Zip file that includes your project's root directory (feel free to remove the node_modules subdirectory before zipping).

TASK 1: Create a homepage component for the Xavier product.
Guidelines:

1. Needs to include some text and images for the intro
2. Needs to have a reusable contact section
3. Needs to be accessible and usable from any device, since brokers are always on the run
4. Note: XYZ Insurance is a stickler for a simple and clear user experience

You did a great job with Xavier's homepage -- it's the hottest thing on the market right now. The management team is super thrilled and want to launch the first version right away. Oren, the Investment Guru at XYZ Insurance Inc. just told your team about this cool new third-party service called Amazing Quote that allows you to compute a private jet liability insurance quote in under 1 min. The service is exposed over a RESTful API. You are tasked to integrate this service on the Xavier website.

TASK 2: Integrate the Amazing Quote API on the Xavier website.
Guidelines:

1. Create a new page on the Xavier website
2. Create a form on the page to capture the jet details. The form should contain the following fields:
    a. "Owner Name" (string)
    b. "Jet Model" (dropdown), options: Gulfstream G650, Cessna A-37 Dragonfly, Cessna Citation Encore
    c. "Jet Seat Capacity" (number)
    d. "Manufacturing Date" (date)
    e. "Purchase Price" (number)
    f. "Broker Email" (email)
3. Submit the jet details to the Amazing Quote API to get a quote
4. Show a loading animation while waiting for the quote
5. Upon a successful quote receipt, display it to the broker
6. In the case when the quote could not be computed, display a message to the broker
7. Use the API documentation below.

Amazing Quote API Documentation

POST https://j950rrlta9.execute-api.us-east-2.amazonaws.com/v1/ArgoChallenge

Request

Headers

- content-type: application/json
- x-api-key: L0Q3GvXCwB9jVSmvaJbw5augw4xHCvMy4Egqim2p

Body

Field	**Data Type**	**Description**
owner_name (required)	string	full name of jet owner
model (required)	enum('Gulfstream G650', 'Cessna A-37 Dragonfly', 'Cessna Citation Encore')	model name of jet
seat_capacity (required)	positive integer	seating capacity of jet
manufactured_date (required)	ISO date	date jet was manufactured
purchase_price (required)	positive integer	purchase price in cent
broker_email (required)	string	broker's email

Sample API Responses

Success

Status Code: 200

Body

```
{
  “quote”: {
        "owner_name": "John Smith",
        "model": "Gulfstream G650",
        "seat_capacity": 12,
        "manufactured_date": "2011-04-13T00:00:00.000Z",
        "purchase_price": 7100000000,
        "broker_email": "jim@broker.com"
  },
  "annual_premium": 100000
}
```

Failure - Not Eligible for Coverage

Status Code: 403

Body

```
{
 “errorMessage”: “Not Eligible for Coverage”,
 "errors": [
  {
   "type": "underwriting",
   "reasons": [
    "Jet is too old.",
    "Jet has too many seats."
   ]
  }
 ]
}
```

Failure - Field Validation

Status Code: 422

Body

```
{
  “errorMessage”: “Failed Validation”,
  "errors": [
    {
      "type": "validation",
      "fields": [
        { "name": "manufactured_date", "reason": "Manufactured date is not properly formatted." },
        { "name": "broker_email", "reason": "Broker e-mail should not be blank." }
      ]
    }
  ]
}
```