Full-Stack Coding Test

Thanks for taking the time to do our coding test. Your challenge for today would be to build a basic hotel search web app. The app needs to integrate with an API provider to fetch hotels' availability.

General Requirements

- This test consists of both client & server-side work. Please use React.js for the client and Node.js for your backend. You're welcome to use a boilerplate to get up & and running faster.
- **We suggest leaving the design part for the end** and concentrating on the architecture and on getting the app up and running first.
- The design, with all the accurate specs and assets, is available here. It is designed for desktop with 1920x1080 resolution. Support of any other breakpoints or resolutions is not required.

App Functionality

The functionality of the website is quite straightforward - the user lands on a blank page with just a search bar on top. The search bar should have the following filters (all fields are mandatory):

- Destination - the full list of ski resorts can be found here. Simply copy & paste the JSON into your app.
- Group size - 1-10.
- Trip start & end dates - the exact dates of the desired trip.

On submitting the search, the results should be shown on the same page, below the search bar, displaying all the available hotels for the selected dates, destination, and group size.

- The results should be retrieved from an external “API provider” (this is actually our internal mock server that will provide you with mock data). All the required information regarding the API is described in the next section below.
- **Although you’re integrating with just one API provider, your architecture should support easy integration of additional API providers in the future.**
- The aggregated results should be sorted by price, ascending.

API reference

IMPORTANT - This API provider only returns hotel rooms for the exact amount of people requested. To provide the customers with more options, your app should also include the results of larger rooms. For example, a couple looking for rooms for 2, should also get rooms for 3 and 4 people. This limitation requires you to run multiple requests to the external API and aggregate the results.

The performance of the search is crucial in terms of user experience. You should avoid any unnecessary waiting times and make sure the user gets the initial and follow-up results as soon as they’re retrieved from the API provider.

- Endpoint - **(POST) -** https://gya7b1xubh.execute-api.eu-west-2.amazonaws.com/default/HotelsSimulator
- Request body (example) -

```
{
 "query": {
  "ski_site": 1,
  "from_date": "03/04/2022",
  "to_date": "03/11/2022",
  "group_size": 4
 }
}
```

Submission Guidelines

- Push your code to a GitHub repository
- Include clear instructions for building and running your application
- Share the repository with pavel@weski.com