Web Developer Technical Assessment

Welcome to the Web developer Technical Assessment. This assessment is designed to evaluate candidate’s technical skills and identify candidate’s strongest points

The assessment consists of two parts:

1.     Reviewing the Assessment: approximately 10 minutes. The candidate will be given the assessment and left alone to review it. The candidate may make notes and start designing the solution during this time. At the end the candidate may ask questions about the assessment to clarify what is required, before starting to implement their solution.

2.     Coding: 2 hours. The candidate will implement web application based on the provided requirements. The task is designed to evaluate the candidate’s expertise as well as assess the time management skills and ability to read and understand technical documentation. If the application is not finished on time, the task will not be considered a failure. If the candidate is able to fully complete the task, any additional functionality will be considered as bonus points.

Agent Application for Flight Booking

* Create a small Single Page Application for Flight Search;
* It can have following modules/ components:
* **Create a Flight Search Options.**

Source City (required)

Destination City (required)

Travel Date (required).

Return Date (Optional).

* **Search Result - Grid should show the below.**

Flight Number.

Airline Name.

Departure and Arrival Time.

Duration.

No. Of Stops.

Price.

Total number of results

**NOTE**:

·        JSON data specific to list of Cities (Source City, Destination City) and Search Results can be mocked either on client/ express server level (preferably);

·        Grid should support pagination and should load data page-wise;

Technical Requirements

You have been asked by your team manager to implement the prototype of the user authentication/ registration system.

The prototype can be implemented as

·        Single Page Application:

Technology Stack:

·        React 16+

·        Redux 3.5+

·        React-Router

·        NodeJS + Express JS/ HAPI JS (for Application Server)

·        HTML5

·        Css3

·        Webpack (Bundling)

·        ES6

·        Jest/Mocha (Unit Testing)

Submitted Exercises will be evaluated on following criterions:

·        Performance and scalability

·        Usage Design principles/patterns

·        Coding best practices/standards

?Acceptance Criteria

* Single Page App to be created using the technology stack mentioned above;
* All Views/Screens should be responsive (you may use Bootstrap or any other framework of your choice);
* All components should manage state through Redux;
* Bundling the application using Webpack;
* Creating all React components using ES6;
* SPA should be served using the Express application server over Node.
* ~~Major UI components should have unit tests developed using the Jest/Mocha framework~~

Deliverables

At the end of the coding test we will require the following artefact’s for evaluation (as a zip file or hosted on an accessible git repository).

·        The full source code of the application

Helpful Tips

We are developers, so we will be looking for good development practices in your solution. Select the technologies you are most comfortable with, aim at showing your best skills, rather than experimenting new technologies.

The time limit is shorter than you think, so focus on getting the application to work first, and then add extras afterwards.

You will be assessed based on your skill level. We would expect a higher standard from senior developer versus a junior.