
# Full Stack Developer Test

## Information

This test is for us to see how you approach a problem and your way of solving it. This test needs to be done using Node and the interface using React.

The project needs to contain a readme file with all the instructions to download and run the project.

We will review this code as though it were to be used for production ergo avoid leaving in any unnecessary console.logs and comments.

Project structure and pattern use is very important, do not take it lightly!

Please do not rush through it, take your time. This is not about how fast you do it, it's about how you do it.

**Your project will be reviewed as though it were to be used for production therefore any unnecessary comments or console logs will be noted.**

## Test

### Requirements

- Use Typescript on Backend
- Use eslint and prettifier
- Use Exceptions on Backend
- Validate the request body
- React with state management Redux, Flux or MobX.
- Organised folder structure
- Comment your code
- Deploy the front-end and back-end using Heroku, Amazon AWS or any cloud service.
- Tests using Jest and Enzyme **(Bonus Points)**

# Questions

## Question 1

Create a page that will contain a **list of all the games** provided in the *game-data.json* file. For this task you can use the *thumb.url* property to display the game thumbnail.

## Question 2

In this task you are required to create a **search functionality**:

- When the user types inside the search bar, the game list created in Question 3 should be updated accordingly.

## Question 3

Consider a **Slot machine** defined like this:

- Reel1: [“cherry”, ”lemon”, “apple”, ”lemon”, “banana”, “banana”, ”lemon”, ”lemon”]
- Reel2: [”lemon”, “apple”, ”lemon”, “lemon”, “cherry”, “apple”, ”banana”, ”lemon”]
- Reel3: [”lemon”, “apple”, ”lemon”, “apple”, “cherry”, “lemon”, ”banana”, ”lemon”]

The user starts with 20 coins. Each spin will cost the user 1 coin.

**Please note that slot machines only consider pairs a match if they are in order from left to right.**
**Eg:**

**Apple, Cherry, Apple - no win**
**Apple, Apple, Cherry - win**

**Rewards**

- 3 cherries in a row: 50 coins, 2 cherries in a row: 40 coins
- 3 Apples in a row: 20 coins, 2 Apples in a row: 10 coins
- 3 Bananas in a row: 15 coins, 2 Bananas in a row: 5 coins
- 3 lemons in a row: 3 coins

Create an endpoint on the backend that when called by the frontend will return the result of the spin and the coins the player won.

## Question 4

Use the sentences below to draw a schema of a database you would create to store this information:


You are working in a casino. A casino has games. Each game has a unique type. Each game has one or more countries where players are allowed to bet from. A player may or may not have a favourite game.

Send the image of the schema and also the sql to create the database and tables.

## Question 5

Based on the above, write a SQL query to get all players that have games of type “SLOT” as their favourite games.