

Welcome to your

CODING CHALLENGE

We can't wait to see, what you will create!

General information

Task: Build a search engine

Tech Stack: Typescript, React, Node.js

Time frame: 3 days, starting the day after you received this pdf.

Please note: what we would like to see more than a 100% completed solution, is your approach.

This includes clean code practices, history of git commits, code organization and reasonable security practices.

Upon completion of this task, please send the details of the **Github repo**, you used to commit the code to, by **answering the email with which you received this pdf** – the same goes for any questions you might have **before** you start the challenge.

Okay, let's GO!

Requirements specification

1. We have a search field with a button
 - a. the search gets triggered when the search button is clicked
 - b. the search gets triggered when you press Enter/Return
2. The backend should be built using a Node.js framework of your choice
 - a. You can use a hard coded array of objects for the search result
 - i. Bonus: you can have an actual database of your choice (SQL or NoSQL) with a sample data set and describe how to use it in the project
 - b. The objects should have following structure: title, photo, description, short Description
3. Once the search is triggered, it should search objects on the server by title
4. The search results should be displayed on the page (no pagination needed)
5. The result should be styled as a flexbox (max width is free of choice but not full screen width)
 - a. the left column should contain a resized photo (thumbnail)
 - b. the right column should contain a title and shortDescription
 - c. Columns should have proportions of 20-80%
6. The title should link to a page which contains a full-sized photo and description, the page should also have a link to go back to the search results
7. Setup a basic code testing structure as a placeholder where real tests would go, using any testing utility of your choice.

Routing should also be done on the client side using React.

You're free to choose a routing library or implement your own router.

You're free to choose how you'll implement paths both on the frontend and the backend.

Code should be written using Typescript. You can also use **any** or **unknown** types.