The case for backend developers

First impressions are important, second impressions are everything.

So it's time for action. Let's stop talking and get some serious coding done. After all, that's what makes the world go round!

With the following assignment, you can show us what you got. The case tests your competences regarding development and gives us an idea on how well you come up with solutions.

ASSIGNMENT

Create a super simple website on which you can search for movie information. This shows your back-end skills, not your front-end skills, don't sweat about the design. This shouldn't take you more than 3-5 hours to complete. A few things to consider:

- **API selection**: Choose a movie database API that provides free access and suits your needs (e.g. IMDB, Rotten Tomatoes or any other of your choosing);
- **API integration**: Use any preferred server-side language (C#, Node, Java) to integrate with the chosen API;
- **Search functionality**: Implement a search function that takes text user input and calls the API with appropriate parameters or filter down the data;
- **Data processing and filtering**: Filter the retrieved data based on the search criteria if you're not using the API parameters;
- **Data presentation**: Prepare the data for presentation. This might involve extracting specific information (title, release date, plot summary) and formatting it into a suitable structure. This shows your back-end skills, not your front-end skills.

TIPS AND TRICKS

- **Start small and iterate**: Don't try to build everything at once. Start with a basic features and gradually add features;
- **Use online resources**: There are many tutorials, code examples, and documentation available online for both API integrations and web development;

- **Version control**: Use a version control system like Git to track changes in your code and revert to previous versions if needed. Don’t forget what you are building;
- **Pick your favorites**: Choose any language or framework you are confident with. Think about crushing that second interview 💪

BONUS THINGS WE WOULD CONSIDER

- **Error handling**: Implement proper error handling to gracefully handle situations where the API call fails or the data cannot be parsed;
- **Caching**: Consider implementing caching mechanisms to improve performance by storing frequently accessed data;
- **Security**: Ensure proper security measures are implemented, especially if you're handling user input or API keys;
- **Surprise us!** Build something we haven’t thought of or what you think is a valuable addition.

Share your code with us before the second interview and try to remember and briefly document the decisions you made and why you made them. The whole second interview is about **this** case.

Don’t forget to kick some ass!