# Fullstack Junior-Level Test

1. Create a full stack web application, composed of a client-side and a back-end.

2. Use whatever frontend framework you feel comfortable with. React is recommended.

3. Create the backend using Python 3.5+ or Node.js 8.0+.

4. For the backend framework, use Flask or Express.

5. Use MySQL, or MongoDB, as the database to store data in.

## Project Definition

1. Create a website (hopefully one that looks nice) that has a basic form which a Marketer can submit (by filling in the data and clicking a Submit button) with the following fields:

○ His first name (optional to fill) ○ His last name (optional to fill) ○ His email address (required to fill, must be a valid email address) ○ His website address (optional to fill) ○ His LinkedIn profile address (optional to fill) ○ A choice answer to the following question: "How many years of experience do you have with Facebook Marketing?", with the following possible answers: \* "no experience" \* "0-1 years" \* "1-2 years" \* "2 or more years" (optional to select) ○ A value range answer: “What was the biggest campaign budget you have managed in a single month?” The values can be from $1000 to $500,000. (optional to fill)

Make sure you do both client-side and server-side validation of the input.   
Of course, add a “Reset Form” button.

2. You will use a database, which will be either MySQL or MongoDB, or any other database you wish - whatever is easiest for you.

3. When a user submits a form, you will store this information in the database. The Marketer will be redirected to a new page, saying "Thank you!"

4. You will prevent double submissions. (If you submit once - you can’t submit again, instead you will see “Already submitted” message. If you force double submission - identified by the email address - the submission will fail.)

5. The website itself will show the number of Marketers that submitted the form so far. e.g. "36 marketers have joined so far!"

## Extra points

6. Add another page, which is password protected (static hard-coded username and password, no need for databases) that will allow you to see all the submitted information in a table. 7. The table should be nice, sortable by date, email and other values.

## Project Submission

The project should be available for download via a ​Gitlab​ repository. Don’t include our name anywhere in the repository.

What we will be looking for:

● Code styling - consistent all through the board, no eslint/pylint warnings. Comments - concise, to the point, where required - no more, no less. ● Clear, simple, robust design of project files, modules, classes, functions ● Ease of deployment / execution locally ● Usage of modern language constructs (e.g. ES6 in JavaScript, generators and comprehensions in Python, etc) ● Attention to detail