

Greetings from Hexad. I would like to inform you that your profile has been shortlisted for Full stack developer's Assessment round.

Please find attached the assignment that you will need to submit for this round of assessments. The assignment in question has been broken down into stories that you will need to attempt in sequential order. This emulates a real-world scenario, where, in an agile setup we build software iteratively with a lean approach in mind. We are not interested in the most futuristic solutions, but are looking for how your thoughts have evolved with the stories in question.

Things to keep in mind:

1. Tests

We love tests! We write them everyday, multiple times in the same day and are especially fond of test driven development (**TDD**). We would love to see your solution have been driven using tests. You can refer to the resource below if you need some guidelines/hints about how to do TDD.

<https://medium.com/javascript-scene/tdd-changed-my-life-5af0ce099f80>

2. Clean code

We love Uncle Bob to the T! But what we love more is beautifully written clean code that is a delight to read. And nothing excites us more than a well modelled solution. We would love to see your thoughts (in code) around object modelling and clean code. You could also choose to include your architectural decisions/thoughts/assumptions in a README so that we can understand your thought process. You can refer to the resource below for some guidelines/hints about how to go about this.

<https://medium.com/mindorks/how-to-write-clean-code-lessons-learnt-from-the-clean-code-robert-c-martin-9ffc7aef870c>

3. Principles principles

We love **SOLID**, **DRY** and **KISS** principles. We are looking for a SOLID principle driven, DRY following, KISS inspired code that is maintainable and testable to the end of the world. You can refer to the resources below to find out more about these principles.

<https://www.baeldung.com/solid-principles>
<https://dzone.com/articles/software-design-principles-dry-and-kiss>

4. Commit regularly

What we value most in our code is the fact that we iteratively build the solution over a period of time handling a variety of use cases and changes that come our way. And the only way we do that is by making sure we commit regularly. We expect you to commit your code regularly, especially after each story completion, so that we can see how the solution evolved with the progression of stories. **IN THE INTEREST OF FAIRNESS, DON'T PUSH THE CODE TO A PUBLIC REPOSITORY.**

Timelines:

You have a maximum of 5 to 7 days to revert to us with the completed assignment, although we always like candidates who can get it done in 3. Once submitted, our panel will evaluate the assignment based on the above-mentioned principles and get back to you.

Submitting your solution:

Once you are happy with your solution, create a zip file containing your source code, any configuration files and your version control folder (e.g. .git) and send it to the HR. If you have any custom build tools, mention that in a README specifically with details on how to run your tests and compile your code