

TODO Project

Pick your thing, show your skills

OVERVIEW

The TODO project ideally should be considered as distributed system solution that enables end-users to register TODO items and keeping track on what is left to be done.

Your task is to pick the area of the whole system that you feel most familiar with and provide as close as possible to a working solution in a time-box manner that demonstrates your skills, knowledge and ambition in the best possible way.

It is recommended to pay attention to the production-readiness of the solution rather than its completeness. For example having similar, aka copy/paste, implementation for `add` and `remove` REST API calls will be less important than having HTTP params validation for only one of the REST API calls.

REQUIREMENTS

Functional

1. User has at least one list, where:
 - a. List has a title
 - b. List has zero, one or more items
 - c. Item has a description
 - d. Item can be either new or done
2. Users must be able to add, delete or update items
3. Users must be able to share an item

Non functional

A good solution will cover at least one of the following non-functional requirements:

1. Simplicity and Predictability.

-
2. Reliability and Scalability.
 3. Security and Privacy.
 4. Excellent UX, such as cross device, offline and etc. support.

Deliverables

1. Working code in a git or similar source repository.
2. What is implemented?
 - a. What is included in the deliverables?
 - b. Why have you decided to focus on what you've delivered?
3. What has to finished to complete the aspect of the project you delivered?
 - a. Including functionality, tools, actions, activities and etc.
4. How to deliver the full-blown solution to end-users, aka go live?
 - a. (similar to the previous point)

FAQ

How much time I need to complete the project?

For an experienced candidate this task should take a couple of hours to complete.

In case you are not experienced, you will need more time, but it should not take more than few days.

I need more time to complete the task than I'm comfortable with. What to do?

Try to reduce the deliverables to a working sub-set and explain what is left to be done.

For example focus to deliver best in class add an item to a list UX and explain what is left and how you intent to implement it.

I can code it, but I need designs?

Don't worry about the UX if this is not your thing. Follow the default platform design patterns.

For example, if you are C++ post school graduate you can deliver your solution as CLI program.

I can implement multiple-aspects of the project, what should I do?

Focus on areas that will better demonstrate your skills.

For example instead of doing a REST API, you can focus on delivering a well thought set of tests or an infrastructure as code solution and describe the rest in the accompanying notes.

I still cannot decide what part of the solution to deliver. What do you recommend?

In that case you have a very broad set of skills and instead of code you can deliver architecture which contains enough details to be considered implementation ready.