

SolaVieve

Full-stack Developer Challenge

Submitted By: Neeraj Anturkar

GitHub Link: [Interview Submission - Fullstack Developer \(Neeraj Anturkar\)](#)

Question 1:

When you start your new job at SolaVieve and receive a summary of all the projects in progress, how would you organize yourself to work on them? Consider priorities, deadlines, Software Requirements Specifications, teams involved.

Answer:

- After getting acquainted with all the ongoing projects at SolaVieve, I would go through the Sprint board for all planned User Stories / Bugs for the Sprint. (Assuming Agile-Scrum project management methodologies are in practice).
- If there are any such entities that can be picked for development, and are aligned with the Sprint Goal set by the team, I would pick simple task first and complex later, just to get comfortable with the existing code base.
- If there are no such planned User Stories / Bugs, I would go through the backlog, and consult the Product Owner's and Development Manager / Team Managers to discuss the priorities, and pickup task accordingly.

Question 2:

Please design the perfect template in which you would like to receive the specifications of a new IT project you are in charge of.

Answer:

Following are sections of template, I would bring up to receive specifications of a new project:

- Project Context - Purpose description of the project
- Project Scope - List of Epics / User Stories that are scoped in the project
- Project Goal - Goal description, Deadlines
- Project Resources - Number of people required, estimated effort, time.
- Deliverables - List of artefacts to be produced with delivery date.

The above template would further be used to derive a project plan.

Question 3:

You receive this screenshot of an error in the app. What steps would you take from the moment you receive it until you fix the bug? Is there any step you would take after the bug is fixed?

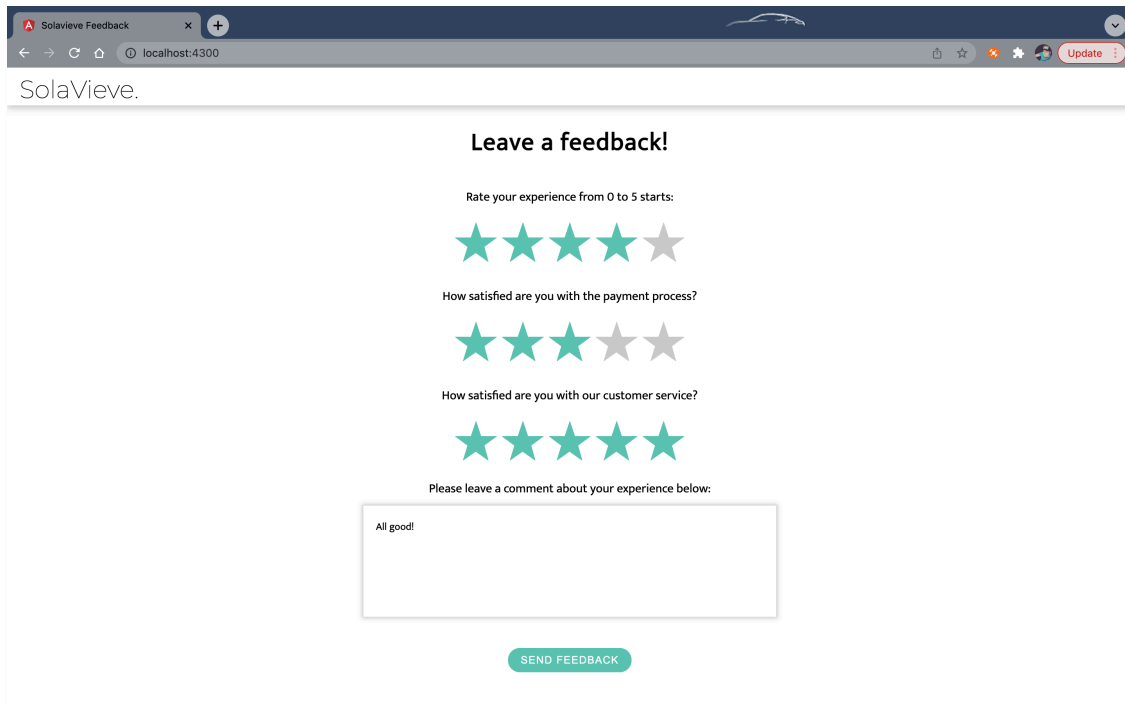
Answer

Upon receiving the screenshot of an app in the error, I would approach the fix in the following steps:

1. Go through the error stack trace to the point where the error has occurred.
2. Check for missing null / undefined checks.
3. Go up the trace with the expected inputs.
4. Reproduce the bug (in debug mode) to understand the behaviour.
5. Fix all detected places in code where the checks are needed.
6. Reproduce the same application behaviour to validate the fix.
7. Write behavioural unit and end to end tests, wherever possible to detect such errors earlier in development phase.

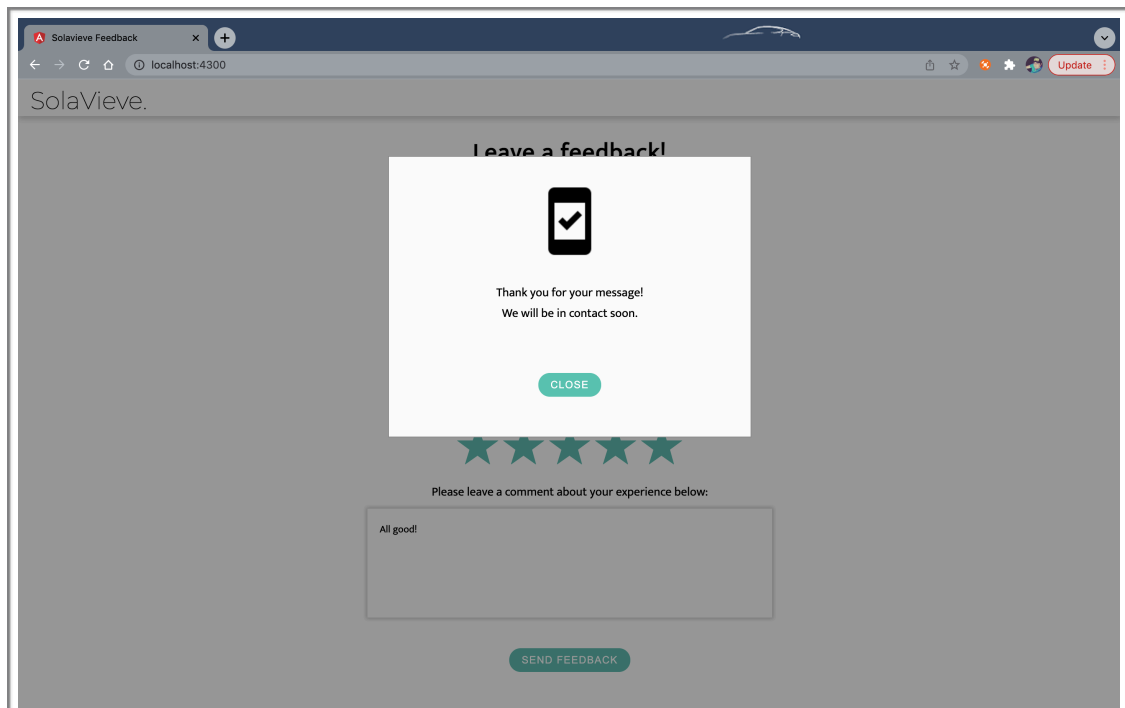
Question 4:

In your first week of work at SolaVieve, you are asked to design a feedback page to add to one of SolaVieve's products. The UX designer sends you the Figma design. Develop the feedback page following the design, with Angular or React. Note: the pop-up is necessary. Any other additional feature that you want to add will provide extra points, as long as it is relevant in the design.



A screenshot of a web browser showing the 'SolaVieve Feedback' page. The page has a dark blue header with the SolaVieve logo. The main content area is white and contains the following elements:

- Leave a feedback!** (Section Header)
- Rate your experience from 0 to 5 stars:** (Text)
- Star Rating:** Five teal stars, with the first four filled and the fifth empty.
- How satisfied are you with the payment process?** (Text)
- Star Rating:** Five teal stars, with the first three filled and the last two empty.
- How satisfied are you with our customer service?** (Text)
- Star Rating:** Five teal stars, all filled.
- Please leave a comment about your experience below:** (Text)
- Text Input:** A white text box containing the text 'All good!'.
- SEND FEEDBACK** (Button)



A screenshot of the same web browser showing the 'SolaVieve Feedback' page, but with a confirmation pop-up displayed. The pop-up is a white box with a dark blue border and contains the following elements:

- Checkmark Icon:** A black square icon with a white checkmark.
- Thank you for your message!** (Text)
- We will be in contact soon.** (Text)
- CLOSE** (Button)

The background of the page is dimmed (grayed out) to show the pop-up. The feedback form elements are still visible behind the pop-up.

GitHub Link: [SolaVieve - Feedback Application - Angular \(Neeraj Anturkar\)](#)

Question 5:

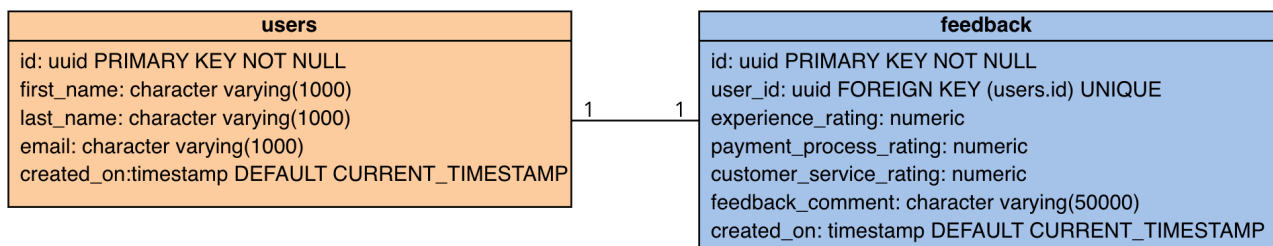
Backend:

As a full stack you will need to assist in some back end development. Could you please build a backend NodeJS server for persisting the feedback to Postgresql.

Connect the frontend to the backend. Discuss your API design & table design decisions in less than 400 words”.

Answer:

Data Model



- This data model assumes user management (creating / authentication) is in place.
- Only users of the system should be allowed to submit feedback.
- All users should be allowed to submit feedback only once.
- There for following constraints are put in place:
 - `user_id.fk` (Foreign key reference user table in feedback table)
 - `user_id.uk` (Unique `user_id` in feedback)

API Design

POST	<code>/api/v1/feedback</code>
Headers	<code>Content-Type: application/json</code> <code>Authorization: Bearer {token}</code>
Body	<pre>{ "experienceRating": 4, "paymentProcessRating": 2, "customerServiceRating": 3, "feedbackComment": "All Good!" }</pre>
Responses	201 - Created 200 - Feedback already submitted 400 - Bad Request (Missing Parameter) 500 - Internal Server Error

Question 6:

Please complete the first three days of Advent of Code
(<https://adventofcode.com/>) .

We will be interested in hearing your implementation strategy in
a follow-up interview.

Answer:

GitHub Link: [SolaVieve - Advent of Code \(Neeraj Anturkar\)](#)

```
Advent of Code [About] [Events] [Shop] [Settings] [Log Out] Neeraj Anturkar 6*
@xxffff2021 [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

~~~~~ 1 **
      2 **
      3 **
      4
      5
      6
      7
      8
      9
     10
     11
     12
     13
     14
     15
     16
     17
     18
     19
     20
     21
     22
     23
     24
     25

Our sponsors help
make Advent of
Code possible:

McGraw Hill -
Join us in
transforming
education. We are
looking for
talented,
passionate,
mission-driven
software
engineers and
leaders looking
to make a
difference
globally. COVID
has provided the
inflection point,
come set the
direction.
```

Question 7:

In layman's words explain what is a Server-Side Request Forgery attack, and how would you prevent

Answer:

Server-Side Request Forgery

- In Server side request forgery attack, attacker can exploit publicly hosted application's vulnerability to read / update internal resources.
- By manipulating the URL the attacker can also read server configurations, connect to internal databases and services that are http enabled.
- The attacker can also use SSRF attack to import untrusted data into the application that expects to read data only from trusted sources.

Mitigation

Following measures can be taken to mitigate SSRF attack:

- Sanitizing and validating data coming from the client.
- Setting up a whitelist of allowed URLs.
- Making sure that a raw response is never sent by the clients.
- Practicing "deny by default" policies for firewall, to restrict non essential intranet traffic.