

Checkout.com Front End Challenge

Collecting Customer Feedback

We want to build a web application to collect and display customer feedback.

We have put together some initial wireframes below:

Page 1

Feedback Form

Name

Email

Rating (1 - 5 stars)

Comment

Submit

Submit feedback



Go back to submit again



Page 2

Feedback Results

Go back

Chart showing distribution of 1 - 5 ratings submitted

Latest Comments

Email

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua....

Email

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua....

Email

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua....

Detailed Requirements

The application should allow a customer to:

1. Write a review for a product (name, email, rating 1 to 5 stars and leave a comment).
 - a. **All fields are required and must be validated prior to form submission.** Use your judgment on sensible validation.
2. Following submission of feedback, the user should be transitioned to a second page where they can see a chart showing the distribution of ratings and most recent comments (you can use any library available for charting).
3. The user can go back to submit more feedback. Going back should present the user with an empty form.

Guidelines

Kindly create a public repository on GitHub and provide your recruitment manager with a link.

Technologies

The only constraint is that your solution must use React. Otherwise you are free to make use of any technology (toolchains, component libraries, etc.) you are comfortable with.

What we value at Checkout.com

Documentation

The right amount of documentation is the one that explains what a project does, how to operate and why certain choices were made. Also, what's left to improve and why.

Code Architecture

Code should be easy to read and change. Consider folder structure, component breakdown, general separation of concerns, ease of testing and keeping things simple for the reader.

Tests

Software Engineers at Checkout.com are responsible for building, testing, deploying and operating our software. For this exercise, we're not looking for full test coverage, but we are looking for a good demonstration of how you test. You should ensure there are enough tests included to demonstrate your knowledge and ensure the application works as expected.

User Experience

We're looking for a functional user interface, enhanced by thoughtful user experience that adapts gracefully to different screen sizes and devices.