

Acme Widget Co

Acme Widget Co are the leading provider of made up widgets and they've contracted you to create a proof of concept for their new sales system.

They sell three products –

Product	Code	Price
Red Widget	R01	\$32.95
Green Widget	G01	\$24.95
Blue Widget	B01	\$7.95

To incentivise customers to spend more, delivery costs are reduced based on the amount spent. Orders under \$50 cost \$4.95. For orders under \$90, delivery costs \$2.95. Orders of \$90 or more have free delivery.

They are also experimenting with special offers. The initial offer will be “buy one red widget, get the second half price”.

Your job is to implement the basket which needs to have the following interface –

- It is initialised with the product catalogue, delivery charge rules, and offers (the format of how these are passed it up to you)
- It has an add method that takes the product code as a parameter.
- It has a total method that returns the total cost of the basket, taking into account the delivery and offer rules.

Here are some example baskets and expected totals to help you check your implementation.

Products	Total
B01, G01	\$37.85
R01, R01	\$54.37
R01, G01	\$60.85
B01, B01, R01, R01, R01	\$98.27

What we expect to see –

- A solution written in easy to understand and update TypeScript
- Simple UI in React. It doesn't have to be super polished, but it can't be ugly
- A README file explaining how it works and any assumptions you've made
- Pushed to a public Github repo

Don't want to do a homework assignment? We get it.

We respect your time. If you already have code you are proud of, you can skip the test project above. However, to ensure a fair comparison with other candidates, your submission **must** follow these 3 rules:

1. **It must be technically comparable:** If the test below asks for a complex app with state management, don't send a static landing page. Your project needs to demonstrate the same technical depth and skills we are asking for from others OR above.
2. **Guide us:** You must include a brief document (or README) telling us exactly **where to look**. Explain what the project does and highlight the specific files or architectural decisions you are proud of so we know what to focus on.
3. **Show the history:** We need to see *how* you build software, not just the final result. Please share a repository link (GitHub/GitLab) with **visible commit history**.
 - *Note: If the repo is private, please invite us. In the absolute worst case, we can accept a Zip file containing the `.git` folder, but a live repo is strongly preferred.*