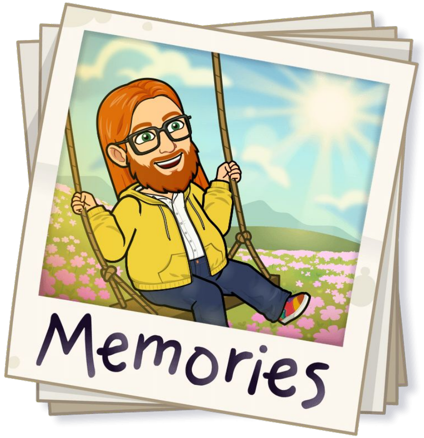
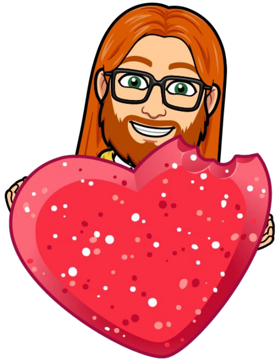
Coding Challenges

Prepare for Success in Technical Interviews

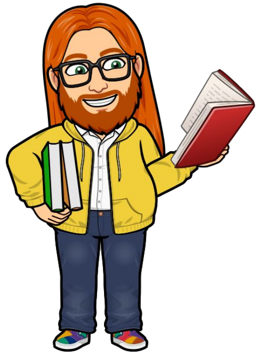
Who is Ben, the Full Snack Tester?



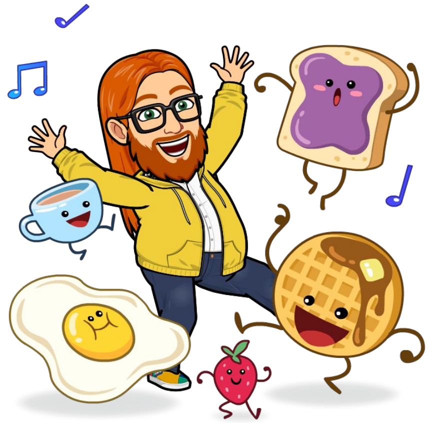
12+ Years



Love testing



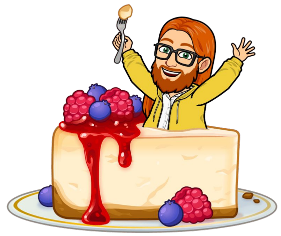
Here to learn



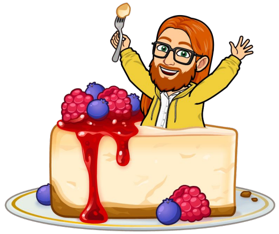
Follow

Agenda

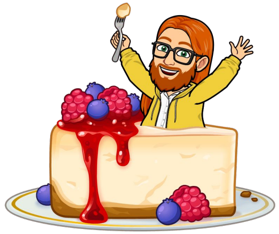
- Setup the Snack Shop on your laptops
- Warm up with snack based ice breaker
- Welcome to Make Believe Labs
- Challenge 1: Update an existing test
- Challenge 2: Expand our coverage with a new test
- Challenge 3: Review some code and provide feedback
- Reflections and feedback
- Resources: More ways to practice



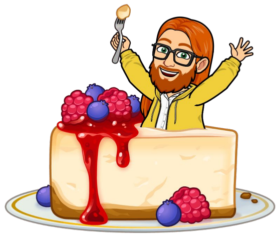
Let’s get setup



⏳ 10 minute timebox

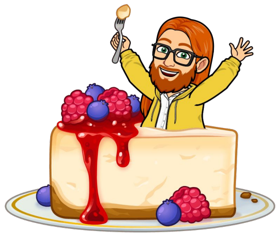


https://github.com/make-believe-labs/snack-shop-sit/blob/main/testbash-2024/workshop-setup.md



Where is the Snack Shop?

https://lab.fullsnacktester.com/



Setup 2: Get system tests running

Continue to follow instructions in workshop-setup.md

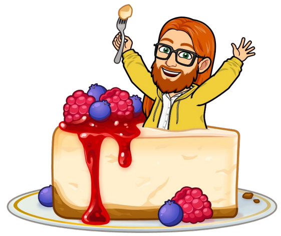
cd snack-shop-sit

nvm install

nvm use

npm ci

npm run



Setup 3: NPM scripts in the SIT

Continue to follow instructions in workshop-setup.md

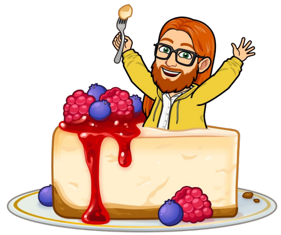
npm run test:api

npm run test:ui

npm run report

npm run update

npmr run record



How much sugar are in your Tic Tacs?

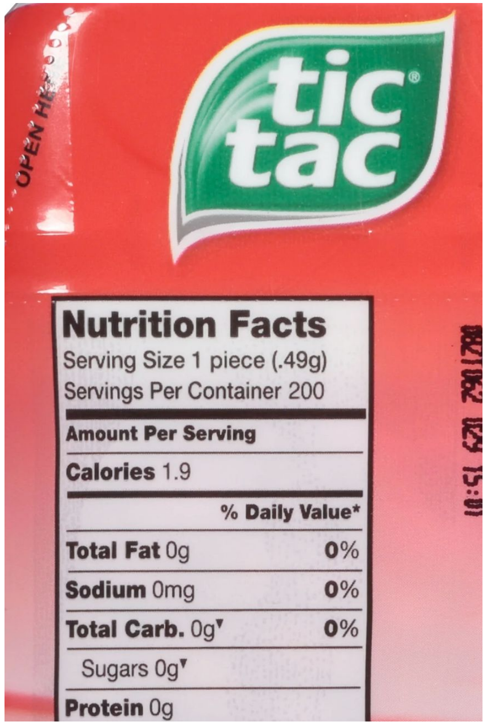
Take the next three mins to do some quick research,

How much sugar is in Tic Tacs?

There are a few packs on your table,

share them around.

⌛ 3 minute timebox



Nutrition Facts
Serving Size 1 piece (.49g)
Servings Per Container 200

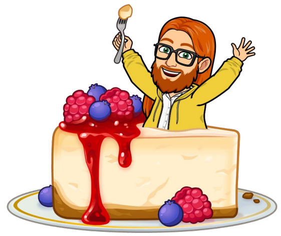
Amount Per Serving	
Calories 1.9	
	% Daily Value*
Total Fat 0g	**0%**
Sodium 0mg	**0%**
Total Carb. 0g▼	**0%**
Sugars 0g▼	
Protein 0g	

Welcome to Make Believe Labs

Started in 2024 by founders Ben Dowen, Lisi Hocke and Vernon Richards, Make Believe Labs is the most awesome pretend digital agency of all time.

Our developers have been rushing to get The Snack Shop mvp ready for a critical client demo, and almost every shortcut has been taken and every corner cut in our pursuit of speed.

Now we need help upping our testing game, and we think you're the perfect candidate for the job. Before we sign on the dotted line, there are a few code challenges we need you to complete so we can be sure you're up to the task.

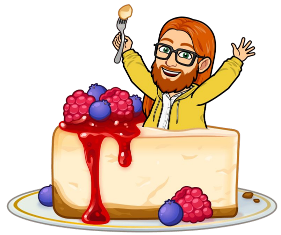


What are our interviewers looking for?

Like many employers, at Make Believe Labs we appreciate technical skills go far beyond writing code. So throughout the day we will be paying close attention to these other important aspects.

How well can you:

- Grasp the fundamental concepts
- Think about and approach the problem
- Communicate and collaborate
- Use your time and understand what else could be done
- Zoom in and out to see the big picture and the detail



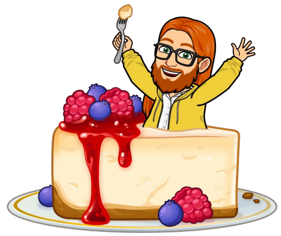
Don’t panic! Do work in pairs or small groups

At Make Believe, we are big fans of collaborative working. So feel free to work together during today's challenges.

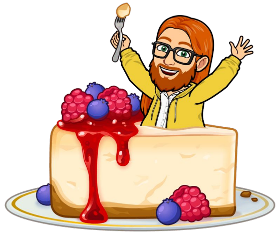
We recommend small groups, of 2-3, although you may work individually if you prefer.

You can either gather round a single laptop, and use driver/navigator strong style pairing, or each follow along on your own laptop, and share tips, tricks and ask questions as you go.

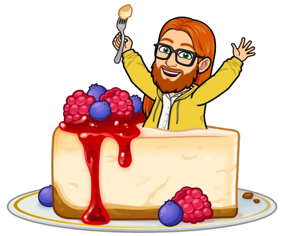
For further support, please raise your hand, and me or one of our lovely ambassadors will come and help.



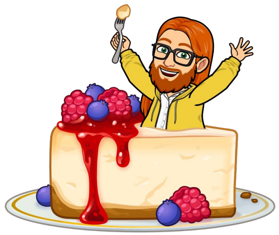
Take notes as you go



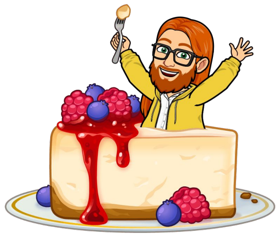
Challenge 1: Update an existing test



⌛ 15 minute timebox



testbash-2024/challenge1-improve-test.md



Update a test

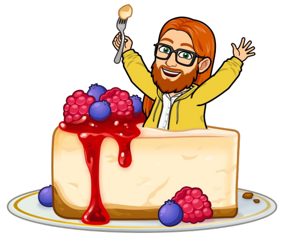
The test in **snacks.spec.ts** was created last week using the **Playwright record option**, it worked great **first time round**, and it passed code review with a **demo done on the developers machine**.

Since then, it's been **failing for everyone else,** can you figure out why and fix it?

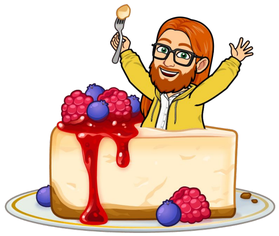
Follow instructions in challenge1-improve-test.md and fix it.

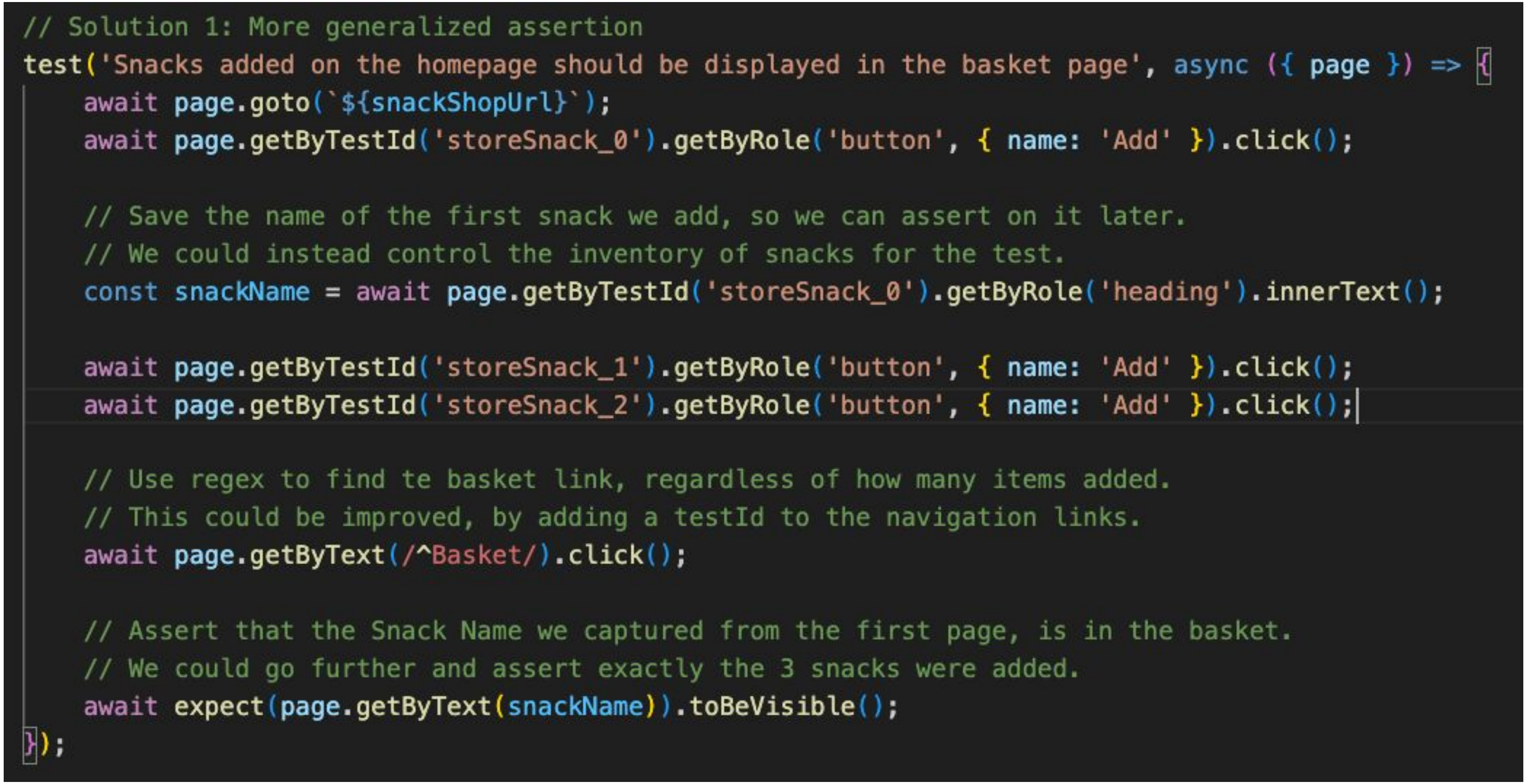
Start by doing the minimal fix, if you have time, pick a side quest.

Remember you don't need to finish the code, take notes, and be ready to discuss.



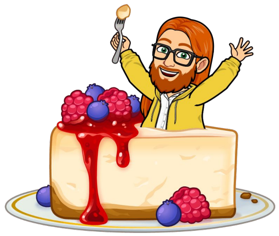
Update a test: How did you get on?



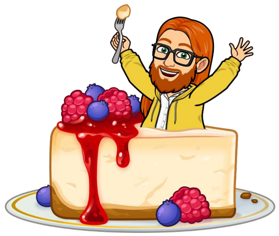


```
// Solution 1: More generalized assertion
test('Snacks added on the homepage should be displayed in the basket page', async ({ page }) => {
    await page.goto(`${snackShopUrl}`);
    await page.getByTestId('storeSnack_0').getByRole('button', { name: 'Add' }).click();

    // Save the name of the first snack we add, so we can assert on it later.
    // We could instead control the inventory of snacks for the test.
    const snackName = await page.getByTestId('storeSnack_0').getByRole('heading').innerText();

    await page.getByTestId('storeSnack_1').getByRole('button', { name: 'Add' }).click();
    await page.getByTestId('storeSnack_2').getByRole('button', { name: 'Add' }).click();

    // Use regex to find te basket link, regardless of how many items added.
    // This could be improved, by adding a testId to the navigation links.
    await page.getByText(/^Basket/).click();

    // Assert that the Snack Name we captured from the first page, is in the basket.
    // We could go further and assert exactly the 3 snacks were added.
    await expect(page.getByText(snackName)).toBeVisible();
});
```

Challenge 2: New test



⏳ 20 minute timebox



Challenge 2: New test

Follow instructions in:
workshops/testbash-2024/challenge2-new-test.md

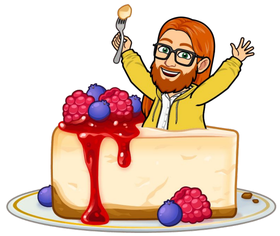
Summary:

You're tasked with writing a new test

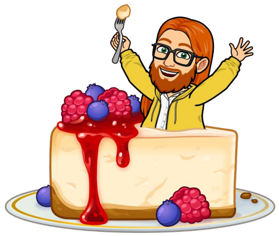
Option 1: New UI Test, mocking orders API in Playwright

Option 2: API Test for the orders API

If you get more time, follow a side quest.



New test: How did you get on?



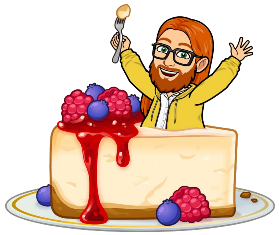
New test: An example solution

Solutions at the bottom of the instructions, found in:
workshops/testbash-2024/challenge2-new-test.md

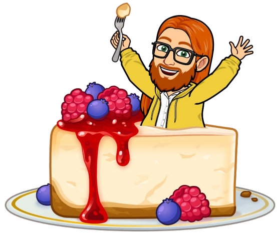
Or see git:

https://github.com/make-believe-labs/snack-shop/blob/test-bash-24-solutions/snack-shop-sit/api-tests/orders.spec.ts

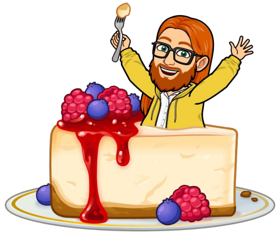
https://github.com/make-believe-labs/snack-shop/blob/test-bash-24-solutions/snack-shop-sit/ui-tests/orders.spec.ts



Challenge 3: Code review



⌛ 15 minute timebox



Exercise 3: Reading code

Follow instructions in:
[workshops/testbash-2024/challenge3-read-code.md]

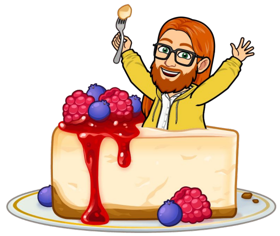
Reading code and giving feedback is important

Option 1: Read the Snack Shop FE Basket Component

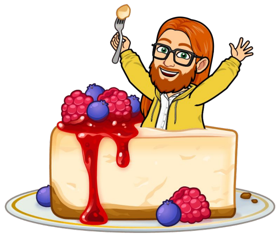
Option 2: Swap with another person/pair and read their test from challenge 2

Make notes and give feedback, discuss with others on your table

Optional: Use a LLM chatbot like Phind to help you review the code



Read code: How did you get on?

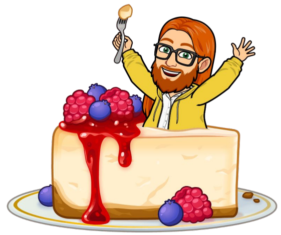


Reflection and feedback

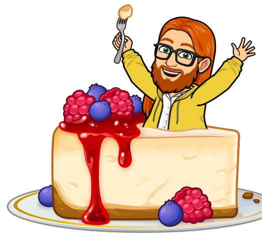
You all did flipping amazing!

Recap of learning outcomes:

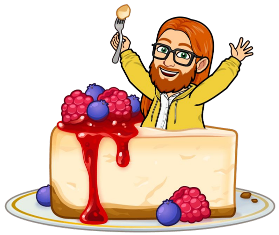
- Identify what is expected from you in a coding exercise
- Recommend improvements to existing automation
- Use Jest to add a test to an existing automation framework
- Discuss what else you could do with the exercise, given more time



Please give me some feedback



Lean, Practice, Succeed!



What next: Test Automation focus

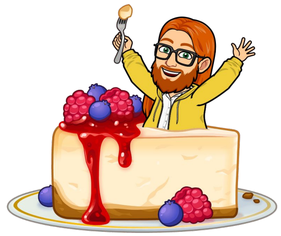
Bas Dijkstra’s guide to building a test automation project
https://github.com/basdijkstra/a-test-automation-project

Angie Jones 10 Portfolio Projects for automation engineers
https://angiejones.tech/10-portfolio-projects-for-automation-engineers/

Andrew Knight, the Automation Panda, list of practice sites:
https://automationpanda.com/2021/12/29/want-to-practice-test-automation-try-these-demo-sites/

Alan “Evil Tester” Richardsons API Challenges
https://www.eviltester.com/page/tools/apichallenges/

Test Automation University
https://testautomationu.applitools.com/



What next: Learning to code focus

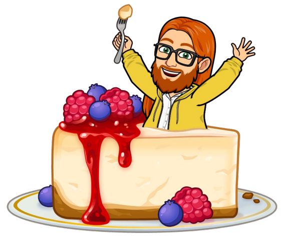
Free Code Camp:
https://www.freecodecamp.org/

Code Academy
https://www.codecademy.com/

Codewars:
https://www.codewars.com/

React Tic-Tac-Toe tutorial
https://react.dev/learn/tutorial-tic-tac-toe

LeetCode
https://leetcode.com/



What next: Code and Computer Science Fundamentals

Base CS
https://medium.com/basecs
https://www.codenewbie.org/basecs

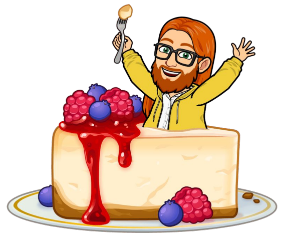
CS50: Introduction to Computer Science
https://pll.harvard.edu/course/cs50-introduction-computer-science

Free short courses from Open University
https://www.open.edu/openlearn/free-courses/full-catalogue

SOLID Principles
https://solidprinciples.org/

Object-oriented programming
https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming

Functional Programming
https://www.freecodecamp.org/news/intro-to-functional-programming-basics/



Thank you!

Go do a 99s talk 😍