Acceptance Tests

In this lesson, let's discuss what acceptance tests are and see one in practice

WE'LL COVER THE FOLLOWING

- Software Testing

Acceptance tests look a lot more different. Instead of having a set of inputs map to a set of expected outputs, acceptance tests typically look at the state of the world after a set of actions have been taken.

For example, to test the form's password guidance modal, we can do something like:

```
function TestPasswordGuide() {
  const passwordInput = document.getElementsByClassName('form__input--password')[0];
  passwordInput.focus();

  const passwordGuideMessage = document.getElementsByClassName('password__guide__message')[0]

  const badPassword = 'foo';
  const goodPassword = badPassword + 'bar123';

  for (const character of badPassword) {
    const keypress = new Event('keydown');
    keypress.key = character;
    document.dispatchEvent(keypress);
  }

  assert(passwordGuideMessage.innerHTML === 'Bad');

  for (const character of badPassword) {
    const keypress = new Event('keydown');
    keypress.key = character;
    document.dispatchEvent(keypress);
  }

  assert(passwordGuideMessage.innerHTML === 'Good');
}
```

This test focuses on the password input (simulate the act of a user clicking on that input), sends keypress events (simulate a user typing), and checks what

that input), sends keypress events (simulate a user typing), and checks what the password guide message is set to.

Software Testing

The more mature your software/application is, and the more people come to rely on it, the more important tests become. Frontend testing typically gets less code coverage, and one of the reasons for that is because it's often hard to write tests for web apps. If your app has an animation, do you include a delay for it to finish? How do you test that a message shows up if you hover over something?

It can be tempting to handwave that a feature works because you tested it on your machine and saw that it works, or assume your change didn't cause a bug because you don't see one. However, if you include good tests that evolve with your source code, they'll likely help catch regressions and force you to think rigorously about edge cases.

Now, let's take a look at the unique errors for each field in our form.