

Contact Form E2E Testing with Playwright

Create a contact form (contact.html) and test it using Playwright to ensure:

- The user can type in phone, email, and zip
- The form validates input
- It shows success or error messages

Folder Structure

```
playwright-e2e-demo/  
├── public/  
│   ├── contact.html    ← HTML page with the form  
│   └── contact.js      ← Validation logic  
├── tests/  
│   └── contact.spec.ts ← Playwright test  
└── package.json
```

HTML: public/contact.html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Contact Form</title>  
  <link  
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/  
bootstrap.min.css"  
    rel="stylesheet"  
  />  
</head>  
<body>  
  <div class="container mt-5">  
    <h1>Contact Form</h1>  
    <form id="myForm" onsubmit="return false;">  
      <div class="mb-3">  
        <label for="phone" class="form-label">Phone</label>  
        <input type="text" class="form-control" id="phone"  
name="phone" />  
      </div>  
      <div class="mb-3">  
        <label for="email" class="form-label">Email</label>  
        <input type="email" class="form-control" id="email"  
name="email" />  
      </div>  
      <div class="mb-3">  
        <label for="zip" class="form-label">Zip Code</label>  
        <input type="text" class="form-control" id="zip" name="zip" />  
      </div>  
      <button type="submit" class="btn btn-primary">Submit</button>  
    </form>
```

```
    <div id="response" class="mt-3"></div>
  </div>
```

```
    <script src="/js/contact.js"></script>
  </body>
</html>
```

JS: public/js/contact.js

```
document.addEventListener('DOMContentLoaded', () => {
  const form = document.getElementById('myForm');
  const responseDiv = document.getElementById('response');

  form.addEventListener('submit', (e) => {
    e.preventDefault();

    const phone = document.getElementById('phone').value.trim();
    const email = document.getElementById('email').value.trim();
    const zip = document.getElementById('zip').value.trim();

    const errors = [];

    if (!/^\\d{10}$/.test(phone)) {
      errors.push('Phone must be 10 digits.');
```

```
    }
    if (!/^\\w[-.]+@[\\w.-]+\\.\\[a-zA-Z]{2,}$/.test(email)) {
      errors.push('Invalid email address.');
```

```
    }
    if (!/^\\d{5}$/.test(zip)) {
      errors.push('Zip must be 5 digits.');
```

```
    }

    if (errors.length) {
      responseDiv.innerHTML = '<div class="text-danger">' +
errors.join('<br>') + '</div>';
    } else {
      responseDiv.innerHTML = '<div class="text-success">Thank you! We
will contact you soon.</div>';
    }
  });
});
```

Playwright Test: tests/contact.spec.ts

```
import { test, expect } from '@playwright/test';

test('contact form submits with valid data', async ({ page }) => {
  await page.goto('http://localhost:3000/contact.html');

  await page.fill('#phone', '5121234567');
```

```
await page.fill('#email', 'test@example.com');
await page.fill('#zip', '78701');

await page.click('button[type="submit"]');

await expect(page.locator('#response')).toContainText('Thank you');
});
```

How to Run It

1. Install & Set Up

```
npm install -D @playwright/test
npx playwright install
npm install -D serve # If not already
```

2. Serve Your App

```
npx serve public
```

Visit: <http://localhost:3000/contact.html>

3. Run the Test

```
npx playwright test tests/login.spec.ts
```

Or: `npx playwright test tests`

Also try without form filled out.

Optional for debugging:

```
npx playwright test tests/login.spec.ts --headed
```