April 28, 2025

amazon_tests.spec.ts

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

test.beforeEach(async ({ page }) => {
    await page.goto('https://www.amazon.in/gp/bestsellers/?ref_=nav_em_cs_bestseller_0_1_1_2');
});

test('Test01 - Verify page title', async ({ page }) => {
    await expect(page).toHaveTitle('Amazon.in Bestsellers: The most popular items on Amazon');
});

test('Test02 - Verify Amazon banner text', async ({ page }) => {
    const bannerText = await page.locator('#zg_banner_text').textContent();
    expect(bannerText).toContain('Amazon Bestsellers');
});

test('Test03 - Clicking on Baby Products and capturing different screenshots', async ({ page }) => {
    await page.getByRole('link', { name: 'Baby Products' }).first().click();
    // capture screenshot of element
    await page.getByAltText("MamyPoko Pants All Night Absorb Baby Diapers, Large (L), 44 Count 9-14 kg").screenshot({ path: './screenshots/element.png' });
    // capture visible_page screenshot
    await page.screenshot({ path: './screenshots/visible_page.png' });
    // capture full_page screenshot
    await page.screenshot({ path: './screenshots/full_page.png', fullPage: true });
});

test('Test04 - Click on Books category & verify product visibility', async ({ page }) => {
    await page.getByRole('link', { name: 'Books' }).first().click();
    const products = page.getByLabel('Action & Adventure');
    await expect(products).toBeVisible();
});
```

```
test('Test05 - Click on Electronics category & verify Cameras & Photography is visible',
async ({ page }) => {
    await page.getByRole('link', { name: 'Electronics' }).first().click();
    const item = page.getByRole('link', { name: 'Cameras & Photography' });
    await expect(item).toBeVisible();
});

test('Test06 - Performing mouse actions', async ({ page }) => {
    await page.getByRole('link', { name: 'Electronics' }).first().click({ button: 'left' });
    await page.getByRole('link', { name: 'Electronics' }).first().click({ button: 'middle' });
    await page.getByRole('link', { name: 'Electronics' }).first().click({ button: 'right' });
});

test('Test07 - Performing keyboard actions', async ({ page }) => {
    await page.keyboard.press('Tab');
    await page.keyboard.press('Enter');
});
```

==

Github Actions
github/workflows/github-workflow.yml

```
name: GitHub Actions Demo
run-name: ${{ github.actor }} is testing out GitHub Actions 🚀
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "🎉 The job was automatically triggered by a ${{ github.event_name }}
event."
      - run: echo "🐧 This job is now running on a ${{ runner.os }} server hosted by
GitHub!"
      - run: echo "🔎 The name of your branch is ${{ github.ref }} and your repository is
${{ github.repository }}."
      - name: Check out repository code
        uses: actions/checkout@v4
      - run: echo "💡 The ${{ github.repository }} repository has been cloned to the
runner."
```

```
      - run: echo "🖥️ The workflow is now ready to test your code on the runner."
      - name: List files in the repository
        run: |
          ls ${{ github.workspace }}
      - run: echo "🍏 This job's status is ${{ job.status }}."
=================================================================
```

May 6, 2025

portfolio.spec.ts

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

test('Test_01 - has title', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');

  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle(/John Smith - Web Designer/);
});

test('Test_02 - to have URL', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');

  // Expect URL "to contain" a substring.
  await expect(page).toHaveURL('http://localhost:5500/index.html');
});

test('Test_03 - verify More About Me button is visible', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');
  await expect(page.getByText('More About Me')).toBeVisible();
});

test('Test_04 - verify John Smith is visible', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');
  await expect(page.getByText('John Smith').first()).toBeVisible();
});

test('Test_05 - verify after clicking More About Me button', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');
  await page.getByText('More About Me').click();
  await expect(page).toHaveURL('http://localhost:5500/about.html');
});

test('Test_06 - verify Technical Skills & Professional Skills are visible', async ({ page })
=> {
  await page.goto('http://localhost:5500/skills.html');
  await expect(page.getByText('Technical Skills')).toBeVisible();
  await expect(page.getByText('Professional Skills')).toBeVisible();
});

test('Test_07 - verify Communication & Project Management are visible', async ({ page
}) => {
  await page.goto('http://localhost:5500/skills.html');
  await expect(page.getByText('Communication')).toBeVisible();
  await expect(page.getByText('Project Management')).toBeVisible();
});


test('Test_08 - verify JavaScript & UI/UX Design are visible', async ({ page }) => {
  await page.goto('http://localhost:5500/skills.html');
  await expect(page.getByText('JavaScript')).toBeVisible();
  await expect(page.getByText('UI/UX Design')).toBeVisible();
});

test('Test_09 - verify after clicking Skill', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');
  await page.getByText('Skill').click();
  await expect(page).toHaveURL('http://localhost:5500/skills.html');
});

test('Test_10 - verify after clicking Contact', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');
  await page.getByText('Contact').click();
  await expect(page).toHaveURL('http://localhost:5500/contact.html');
});

test('Test_11 - verify Get in Touch & Send Message are visible', async ({ page }) => {
  await page.goto('http://localhost:5500/contact.html');
  await expect(page.getByText('Get in Touch')).toBeVisible();
  await expect(page.getByText('Send Message')).toBeVisible();
});
```

```
test('Test_12 - verify after clicking Home tab', async ({ page }) => {
  await page.goto('http://localhost:5500/contact.html');
  await page.getByText('Home').click();
  await expect(page).toHaveURL('http://localhost:5500/index.html');
});

test('Test_13 - verify after clicking About tab', async ({ page }) => {
  await page.goto('http://localhost:5500/contact.html');
  await page.getByText('About').click();
  await expect(page).toHaveURL('http://localhost:5500/about.html');
});

test('Test_14 - verify after clicking About', async ({ page }) => {
  await page.goto('http://localhost:5500/index.html');
  await page.getByText('About').first().click();
  await expect(page).toHaveURL('http://localhost:5500/about.html');
});

test('Test_15 - verify Download CV btn is present in About page', async ({ page }) => {
  await page.goto('http://localhost:5500/about.html');
  await expect(page.getByText('Download CV')).toBeVisible();
  await expect(page.getByText('Web Designer & Developer')).toBeVisible();
});

test.only('Test_16 - verify Name, Email, Experience, Freelance are present in About page', async ({ page }) => {
  await page.goto('http://localhost:5500/about.html');
  await expect(page.getByText('Name:')).toBeVisible();
  await expect(page.getByText('Email:')).toBeVisible();
  await expect(page.getByText('Experience:')).toBeVisible();
  await expect(page.getByText('Freelance:')).toBeVisible();
});
============================================================
```

May 7, 2025

api.spec.ts

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

```
test.describe.parallel('API Testing', () => {
    const baseUrl = 'https://reqres.in/api';

    test('Simple API Test - Assert Response Status', async ({ request }) => {
        const response = await request.get(`${baseUrl}/users/2`);
        expect(response.status()).toBe(200);

        const responseBody = JSON.parse(await response.text());
        console.log(responseBody);
    });

    test('Simple API Test - Assert Invalid Endpoint', async ({ request }) => {
        const response = await request.get(`${baseUrl}/users/non-existing-endpoint`);
        expect(response.status()).toBe(401);
    });

    test('GET Request - Get User Details', async ({ request }) => {
        const response = await request.get(`${baseUrl}/users/2`);
        const responseBody = JSON.parse(await response.text());
        // console.log(responseBody);

        expect(response.status()).toBe(200);
        expect(responseBody.data.id).toBe(2);

        expect(responseBody.data.first_name).toBe('Janet');
        expect(responseBody.data.last_name).toBe('Weaver');
    });
    // need to check this ===>
    test('POST Request - Create New User', async ({ request }) => {
        const response = await request.post(`${baseUrl}/users`, {
            data: {
                "name": "Abba",
                "job": "QA"
            },
        });
        //expect(response.status()).toBe(200);
        const responseBody = JSON.parse(await response.text());
        console.log(responseBody);
    });
    test('POST Request - Login Successful', async ({ request }) => {
        const response = await request.post(`${baseUrl}/login`, {
            data: {
                "email": "eve.holt@reqres.in",
                "password": "cityslicka"
            },
        });
        const responseBody = JSON.parse(await response.text());
        expect(responseBody.status()).toBe(200);
        expect(responseBody.token).toBeTruthy();
    })
});
```

==

May 8, 2025

api_test.spec.ts

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

test('Test_01 - GET request and store response in a variable', async ({ request }) => {
    const response = await request.get(`https://reqres.in/api/users/2`);
    // verify status code
    // expect(response.status()).toBe(200);

    // validate response text contains a given string
    const text = await response.text();
    expect(text).toContain('Janet');

    // log responseBody to console
    const responseBody = await response.json();
    console.log(responseBody);
    // validate last_name = Weaver
    expect(responseBody.data.last_name).toBe('Weaver');
});

test('Test_02 - GET request - single user not found', async ({ request }) => {
    const response = await request.get(`https://reqres.in/api/users/23`);
    // verify status code
    expect(response.status()).toBe(401);
    console.log('Single user(23) not found')
});
```

```
test('Test_03 - POST request - create a new user', async ({ request }) => {
    const response = await request.post('https://reqres.in/api/users', {
        headers: {
            'X-API-KEY': 'reqres-free-v1',
            'Content-Type': 'application/json'
        },
        data: {
            "name": "Raghav",
            "job": "Teacher"
        }
    })
    expect(await response.status()).toBe(201);

    // validate createdAt is present
    const responseBody = await response.json();
    console.log(responseBody);
    // expect(responseBody.data.createdAt).toBeTruthy();
});

test('Test_04 - POST request - Validate Successful Login', async ({ request }) => {
    const response = await request.post(`https://reqres.in/api/login`, {
        headers: {
            'X-API-KEY': 'reqres-free-v1',
            'Content-Type': 'application/json'
        },
        data: {
            "email": "eve.holt@reqres.in",
            "password": "cityslicka"
        }
    });
    expect(response.status()).toBe(200);
});

test('Test_05 - POST request - Validate unsuccessful Login', async ({ request }) => {
    const response = await request.post(`https://reqres.in/api/login`, {
        headers: {
            'X-API-KEY': 'reqres-free-v1',
            'Content-Type': 'application/json'
        },
        data: {
            "email": "peter@klaven"
        }
    });
    expect(response.status()).toBe(400);
});
```

<=================>

booker_tests.spec.ts

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

const baseUrl = `https://restful-booker.herokuapp.com`;

test('Test_01 - POST request and validate response', async ({ request }) => {
    const response = await request.post(`${baseUrl}/booking`, {
        data: {
            "firstname": "Testers Talk",
            "lastname": "API Tested",
            "totalprice": 1000,
            "despositpaid": true,
            "bookingdates": {
                "checkin": "2018-01-01",
                "checkout": "2019-01-01"
            },
            "additionalneeds": "super bowls"
        }
    });

    // validate status code
    //expect(response.ok()).toBeTruthy();
    expect(response.status()).toBe(200);

    const responseBody = await response.json();
    console.log(responseBody);

    // validate first and last name
    expect(responseBody.data.firstname).toBe('Testers Talk');
    expect(responseBody.data.lastname).toBe('API Tested');
    // validate checkin and checkout dates
    expect(responseBody.data.bookingdates.checkin).toBe('2018-01-01');
    expect(responseBody.data.bookingdates.checkout).toBe('2019-01-01');
});
========================================================================
```