

Deals Jobs - QA Software Engineer / SDET challenge - Reza Paramarta
Feature: Mentoring

Challenge 1: Automation Testing (Test Strategy & Scenarios)

Test Strategy - Types of Testing needed:

1. Functional Testing

- Ensure the mentoring feature works as intended based on requirement.

2. UI Testing

- Validate display of button, form and user flow.

3. Integration Testing

- The Mentoring feature is connected to login, user profile, and so on.

4. Regression Testing

- Ensure new features do not interrupt old features.

5. Exploratory Testing

- Find bugs which are not written in the requirements.

Key Areas of Focus

- User authentication (login and register)
- Mentor listing and search
- Mentor detail page
- Booking / request mentoring
- Error handling (empty data, error network)
- UI behavior (loading, validation message)

Test Environments

- Staging Environment (utama)
- Local / Test Environment (for automation purpose)
- Browser:
 - Chrome (primary)
 - Safari (secondary)
- API test via backend mock

Assumptions

- User can register and login
- Mentor data is available
- There is no payment gateway (or in mock)
- API mentoring is stable

High Level Test Scenarios

Authentication:

- User register with valid data
- User login with registered account
- User login with wrong password

Mentoring Feature

- User see list of mentors
- User search mentor based on:
 - Name
 - Role
- User open mentor detail
- User request / book mentoring
- User see mentoring request status
- User successfully submits mentoring schedule
- User sees scheduled mentoring on dashboard / profile
- User receives status update for scheduled mentoring

Negative Scenarios

- Search mentor with unmatched keywords

- Request mentoring without login
- Submit form mentoring with empty data
- User submits scheduled without selecting date or time
- User tries to double book the same schedule

Testing Schedule

| Days | Activities |
|-------|---|
| Day 1 | Review requirement & create scenario test |
| Day 2 | Manual Testing mentoring feature |
| Day 3 | Create cypress automation (login & search mentor) |
| Day 4 | Extend automation & debugging |
| Day 5 | Github Action (CI/CD) and Documentation |

Challenge 2 - UI Automation:

Framework: Cypress (Javascript)

Automated Scenarios:

1. Register with random fake email to avoid flakiness
2. User login with valid credentials
3. Search mentor
4. View Mentor Detail

Automation Code Script:

e23_mentoring.cy.js

```
describe('Mentoring Feature', () => {
    // This function generate a random string for unique user data
    registration

    function randomString(length = 8) {
        const chars = 'abcdefghijklmnopqrstuvwxyz0123456789';
        let result = '';
        for (let i = 0; i < length; i++) {
            result += chars.charAt(Math.floor(Math.random() *
chars.length));
        }
        return result;
    }

    // This loop hooks before each test case to visit the base URL
    beforeEach(() => {
        cy.visit('/');
    });

    // Test case for user registration
    it('User create new account', () => {
        const randomPart = randomString();
        const timestamp = Date.now();
        const email = `user${randomPart}@${timestamp}.com`;
        const password = 'Secure123!';

        cy.get('#username').type(email);
        cy.get('#password').type(password);
        cy.get('#confirmPassword').type(password);
        cy.get('#register').click();

        cy.get('#username').should('have.value', email);
        cy.get('#password').should('have.value', password);
        cy.get('#confirmPassword').should('have.value', password);
    });
});
```

```

const user = {
  fullName: `Reza_test_${randomPart}`,
  email: `reza.test${randomPart}${timestamp}@gmail.com`,
  waNumber: `628${Math.floor(100000000 + Math.random() * 900000000)}`,
  password: 'Reza12345!',
};

cy.register(user);
cy.xpath("//button[normalize-space()='Unduh Sekarang']").should('be.visible');

});

// Test case for user login and mentoring feature
it('User can login, search mentor, and open mentor detail', () => {
  // Login using custom command
  cy.login(Cypress.env('EMAIL'), Cypress.env('PASSWORD'));
  // Assert login success (mentoring menu visible)
  cy.get(
    'li[id="navigation-item-mentoring"] a[class="inline-block py-1 text-xs !font-semibold text-white !decoration-2 !underline-offset-4 hover:!text-gray-200 hover:underline"]'
  )
    .contains('Mentoring')
    .should('be.visible');
  cy.contains('Mentoring').click();
  // Search for a mentor
  cy.get('#searchMentor').type('QA Engineer');
  // Assert search results
  cy.contains('QA Software Engineer').click();
  // Assert mentor detail page
  cy.contains('About').should('be.visible');
});

});
});

```

Commands.js

```

require('cypress-xpath');

Cypress.Commands.add('login', (email, password) => {
  cy.get('#dealls-navbar-login-btn').click();
  cy.get('#basic_email').type(email);
  cy.get('#basic_password').type(password);
  cy.get('button[type="submit"]').click();
});

Cypress.Commands.add('register', (user) => {

```

```
cy.xpath('//a[@id="dealls-navbar-register-btn"]').click();
cy.xpath(
    '//button[@class="flex items-center gap-3 rounded-lg border p-3 transition-all border-tertiary-violet-50 bg-[#F4F2FF]"'
).click();
cy.xpath('//button[normalize-space()="Continue"]').click();
cy.xpath("//input[@id='fullName']").type(user.fullName);
cy.xpath("//input[@id='waNumber']").type(user.waNumber);
cy.xpath("//input[@id='email']").type(user.email);
cy.xpath("//input[@id='confirmEmail']").type(user.email);
cy.xpath("//input[@id='password']").type(user.password);
cy.xpath("//input[@id='checkPrivacyPolicy']").check({ force: true });
cy.xpath("//button[@type='submit']").click();
});
```

Link for code above: https://github.com/rezaparamarta/Dealls_Mentoring_Feature

Challenge 3: Exploring & Bug Investigation

Objective: Evaluate your analytical thinking, problem-solving approach, and understanding of potential failure points in [dealls](#). The link is n't accessible any more.