Exercício: Desenvolvimento e Teste de “MyWebApp” com Cypress

Objetivo: Neste exercício, você desenvolverá uma aplicação web simples chamada “MyWebApp” e, em seguida, criará testes end-to-end para essa aplicação usando o Cypress.

Parte 1: Desenvolvimento da Aplicação “MyWebApp”

Requisitos da Aplicação “MyWebApp”:

1. **Frontend:**
    - Utilize HTML, CSS e JavaScript para criar uma interface amigável.
    - **Página de Login:**
        - Campos para inserção de e-mail e senha.
        - Botão “Entrar”.
    - **Dashboard:**
        - Exiba uma lista de usuários registrados.
        - Inclua opções para adicionar, editar e excluir usuários.
        - Um formulário deve surgir ao optar por adicionar ou editar, contendo campos para nome, e-mail e um seletor para upload de foto de perfil.
2. **Backend:**
    - Recomenda-se o uso de Node.js com Express, mas sinta-se livre para escolher a tecnologia que preferir.
    - **Autenticação:**
        - Implemente um sistema de login que valide e-mails e senhas predefinidos (sugestão: `admin@example.com` / `admin123`).
    - **Gerenciamento de Usuários:**
        - Crie endpoints CRUD para adicionar, recuperar, atualizar e excluir usuários.
        - Utilize um banco de dados de sua escolha para persistir os dados dos usuários. Pode ser algo simples como SQLite ou até mesmo um array temporário.
    - **Upload de Imagens:**
        - Adicione a capacidade de fazer upload de imagens. As fotos podem ser armazenadas localmente ou em um serviço externo de sua escolha.

Parte 2: Testando “MyWebApp” com Cypress

Tarefas:

1. **Configuração:**
    - Configure o Cypress no projeto.
2. **Página de Login:**
    - Escreva testes para verificar a presença dos campos de e-mail e senha.
    - Teste o fluxo de autenticação, tanto para login bem-sucedido quanto para falhas.
3. **Dashboard:**
    - Após autenticar-se, elabore testes para assegurar que a lista de usuários é corretamente exibida.
    - Implemente testes para a adição, edição e exclusão de usuários.
4. **Requisições de API:**
    - Com o auxílio do `cy.intercept()` , capture e simule respostas das chamadas de API relacionadas ao gerenciamento de usuários.
    - Emule cenários adversos (ex.: falhas na obtenção da lista de usuários) e assegure que a aplicação reage de maneira adequada.
5. **Upload de Imagens:**

- Desenvolva testes para verificar a funcionalidade de upload de fotos de perfil, garantindo que a imagem é corretamente exibida após sua inclusão.

6. **Execução dos Testes:**
    - Rode seus testes tanto no modo visual quanto no modo headless e certifique-se de que todos sejam executados com sucesso em ambos os cenários.

Material de Referência Rápida: Funcionalidades do Cypress

1. Instalação e Configuração

- **Instalando o Cypress:**

```
npm install cypress --save-dev
```

- **Abrindo o Cypress pela primeira vez:**
    - Isso criará uma estrutura básica de diretórios e alguns arquivos de exemplo.

```
npx cypress open
```

2. Navegação e Interação com a Aplicação

- **Visitar uma página:**

```
cy.visit('URL_DA_PÁGINA')
```

- **Interagir com elementos:**
    - Digitar em um campo de input:

```
cy.get('SELETOR_DO_ELEMENTO').type('TEXTO_A_SER_DIGITADO')
```

    - Clicar em um botão ou link:

```
cy.get('SELETOR_DO_ELEMENTO').click()
```

3. Assertions (Afirmações)

- **Verificar se um elemento está presente:**

```
cy.get('SELETOR_DO_ELEMENTO').should('exist')
```

- **Verificar se um elemento tem uma classe específica:**

```
cy.get('SELETOR_DO_ELEMENTO').should('have.class', 'NOME_DA_CLASSE')
```

- **Verificar se um elemento contém um texto específico:**

```
cy.get('SELETOR_DO_ELEMENTO').should('contain', 'TEXTO_ESPERADO')
```

4. Trabalhando com Requisições de Rede

- **Interceptar uma chamada de API:**

```
cy.intercept('MÉTODO_HTTP', 'URL_DA_API').as('aliasDaChamada')
```

- **Aguardar uma chamada de API e fazer uma verificação:**

```
cy.wait('@aliasDaChamada').its('response.statusCode').should('eq', CÓDIGO_DE_STATUS_ESPERADO)
```

5. Autenticação Programática

- **Realizar um request diretamente para a API de login:**

```
cy.request({
  method: 'MÉTODO_HTTP',
  url: 'URL_DO_ENDPOINT_DE_LOGIN',
  body: { ...DADOS_DO_USUÁRIO }
})
```

- **Definindo um token no `localStorage` após login:**

```
window.localStorage.setItem('NOME_DO_ITEM', 'VALOR_DO_ITEM')
```

6. Execução dos Testes

- **Executar testes no modo headless:**

```
npx cypress run
```

- **Executar testes no modo visual:**

```
npx cypress open
```

Esse material de referência oferece uma visão geral das funcionalidades básicas do Cypress usadas na atividade proposta. Para um entendimento mais profundo e para explorar outras funcionalidades avançadas, é sempre recomendado consultar a documentação oficial do Cypress.