API AND UI TEST AUTOMATION

Presented by:
Abdallah Osman – 2106433
Nayef Talib - 2359957
Yaseen Yousif - 1906399
Shaded Baaj – 2019451

Introduction to API and UI Testing

API Testing: Focuses on checking the functionalities of APIs directly. It involves sending requests and verifying responses.

UI Testing: Verifies user interactions with the graphical interface. It ensures the application behaves as expected from a user perspective.

- Objective: To automate these tests to enhance efficiency, reliability, and coverage of testing processes, ensuring application quality.

Importance of Automating Tests

- Efficiency: Automation reduces the time required for repetitive testing tasks.
- Consistency: Automated tests produce consistent results, minimizing human errors.
- Coverage: Enables extensive testing of various scenarios, including edge cases.

- Validation of CRUD Operations: Automation ensures that Create, Read, Update, and Delete functionalities work correctly across different scenarios.-
- User Interactions: Validates the overall user experience by simulating real user scenarios.

Overview of CRUD Operations

- Create: Adding new data or records.
- Read: Retrieving existing data or records.
- Update: Modifying existing records.
- Delete: Removing records from the system

- Importance of CRUD in Applications
- Ensures all aspects of data management are covered.
- Vital for the functionality of many applications, particularly those dealing with data entry and management.

API Testing – Methodology

- Tools Used: Postman for sending requests and validating responses.

- - Base URL: https://a_leibpaqkgx3.v7.demo.nocobase.com/api/order:list- Headers:

- - AUTHORIZATION: BEARER [Token]- ACCEPT: APPLICATION/JSON- CONTENT-TYPE: APPLICATION/JSON

- Testing Scenarios: Focused on GET, POST, PUT, and DELETE methods to cover both positive and negative cases.

Case Studies of API Testing

API Testing Scenarios1.

1. Create Order Scenario (200): Validates successful order creation.
2. Create Order Scenario (403): Tests for forbidden access without authorization.
3. Get Order Scenario (200): Retrieves an existing order by ID.
4. Get Order Scenario (400): Checks behavior with a non-existent order ID.
5. Update Order Scenario (200): Confirms updating order information.
6. Delete Order Scenario (200): Ensures successful deletion of an order.

Each Scenario Checks:

- Status codes
- - Response body fields
- - Headers- Performance metrics

UI Testing – Methodology

- Tools Used: Selenium for automating user interactions.
- - Key Focus Areas:
- - Login functionality
- - Navigation between pages
- - User interface elements and interaction validations
- Test Cases Overview:
- 1. Successful Login: Ensures correct credentials log the user in.
- 2. Unsuccessful Login: Validates error handling for incorrect credentials.
- 3. Password Masking: Tests if password input is masked.
- 4. Navigation to Users Page: Checks successful page navigation.
- 5. Searching for a User: Validates search functionality.

Case Studies of UI Testing

UI Testing Scenarios

1. Deactivate Existing Lead: Tests deactivation functionality.

2. Convert Lead to User: Validates successful conversion of a lead.

3. Output Expectations for Each Case:- Successful outcomes reflecting expected behavior.

4. Error messages for negative cases ensuring system robustness.

Example Code Snippets:

-Python

-Example for Successful Login

-Ensure the login function returns True for valid credentials

conclusion

The project successfully demonstrated the application of comprehensive software testing techniques, combining API and UI automation to enhance application reliability and quality. Using Postman for API testing and Selenium for UI testing, the team validated core functionalities and user interactions across multiple scenarios, including both positive and negative cases. This approach ensured robust coverage of CRUD operations and user interface workflows