Automation Testing Approach:

Scope:

The main goal of test automation is to increase efficiency and reliability in the software testing process. By automating repetitive tasks like regression testing, developers and testers can save time and resources.

This Test automation works by simulating user interactions with the software under test. This can include clicking buttons, entering text, and navigating through different screens or pages to validate the application's behavior from the end user's perspective

For this application, I would choose End-to-End testing, and Regression testing for automation. Each level focuses on different aspects of the software, from individual components to the entire system.

One key benefit of test automation is its repeatability. Once a test script is written, it can be executed multiple times with consistent results. This helps ensure that bugs are identified and fixed reliably.

Tools and Technologies

- **WebDriverIO**: Utilized for browser automation and interaction.
- **TypeScript**: Enhances code quality and scalability with static typing.
- **BDD Framework (e.g., Cucumber)**: Enables behavior-driven development with Given-When-Then syntax.

Project Structure

- src/: Contains TypeScript source files.
- src/pageObjects/: Directory for Page Objects that encapsulate web elements and interactions.
- src/feature/: Directory containing BDD test scenarios written in feature files (e.g., .feature files for Cucumber)
- src/stepDefinitions/: Directory for BDD step definitions written in TypeScript.
- config/: Configuration files for WebDriverIO and TypeScript.
- reports/: Directory for storing test reports.
- utils/: Utility functions and helpers.

Execution and Integration

- **Continuous Integration (CI)**: Integrate with CI/CD pipelines (e.g., Jenkins, GitLab CI) to automate test execution on code changes.
- **Reporting**: Utilize reporting tools to generate and visualize test results, aiding in identifying failures and tracking test coverage over time.

Reason to Automate and Types of testing needed for each test case:

TC #1: Clicking on maintainX image directs to blank home page

Automatability: Yes

Reason: Users often expect intuitive navigation elements like images to lead them to relevant sections of a website, such as the home page. This is particularly important for critical paths that are commonly used by users, such as returning to the home page from any section of the site. The action of clicking an image and verifying the page content is repetitive and suitable for automation.

Types of tests: Functional Testing, UI Testing, Regression and End-to-End Testing testing

TC #2: In 'Create Work Order', Name field is accepting numeric, special characters and infinite number of characters

Automatability: Yes

Reason: Input validation tests are repetitive and time-consuming when performed manually. Automation ensures these tests are executed consistently and efficiently every time. Automation allows testing various input scenarios and edge cases at scale, ensuring comprehensive coverage.

Types of tests: Positive Testing, **Negative Testing, Boundary Testing,** Regression testing

TC #3: Info Icons beside 'Name' and 'Assignee' is missing

Automatability: Yes

Reason: Info Icons provide users with contextual information or tooltips about the fields they are interacting with, enhancing usability and reducing user errors.

Automated tests can quickly verify the presence and functionality of Info Icons across various browsers and platforms.

Types of tests: Functional Testing **, UI/UX Testing,** Regression Testing, Cross-Browser Testing

TC #4: The word "Retourner" is not translated to its English equivalent "Return"

Automatability: Yes

Reason: The translation feature will likely be used in multiple scenarios and languages. Automating this ensures consistency and saves time.

Types of tests: Functional Testing**, UI Testing,** Regression Testing

TC #5: Close icon (x) next to the assignee's name is not functioning

Automatability: Yes

Reason: Clicking the close icon and verifying the UI changes are repetitive tasks that can be efficiently handled by automation.

Types of tests: Functional Testing**, UI Testing,** Regression Testing, Usability testing

TC #6: 'Reopen the work order' button is not helpful

Automatability: Yes

Reason: The scenario involves repetitive actions that are ideal for automation (e.g., navigating to a page, clicking a button).

Types of tests: Functional Testing**, UI Testing,** Regression Testing, Negative testing

TC #7: Work order is created without assignee

Automatability: Yes

Reason: By including this test in automated regression suites, issues related to work order creation without assignees can be identified early in the development cycle.

Types of tests: Functional Testing**,** Regression Testing

TC #8: Duplicate Assignees are selectable while creating work order

Automatability: Yes

Reason: Manually testing to prevent user for selecting same assignee can be time-consuming and error prone. Automation ensures consistent and thorough testing.

Types of tests: Functional Testing**,** Regression Testing

TC #9: A new work order is created with '#' and 'numeric' at the beginning of the name. For example: #9 GPK

Automatability: Yes

Reason: Automated tests ensure that the application continues to enforce the correct format for work orders after changes or updates are made.

Types of tests: Regression Testing

TC #10: 'Name' is defined while creating work order, but this attributes as 'Title' in Work order

Automatability: Yes

Reason: This test case involves verifying a specific attribute (name) consistently across multiple work orders. Automating it ensures that the validation is consistently applied without manual effort.

Types of tests: Regression Testing

TC #11: Functionality of sorting by Id and Status is not applied

Automatability: Yes

Reason: Automated tests cover various scenarios (ascending/descending order) more thoroughly than manual testing, thereby increasing test coverage.

Types of tests: Functional Testing**, UI Testing,** Regression Testing

TC #12: Search Functionality for Work Orders list is missing

Automatability: Yes

Reason: Automated tests provide faster feedback compared to manual testing, especially when validating search results across different datasets and scenarios.

Types of tests: Functional Testing**, Usability Testing,** Regression Testing

Key Features and Functionalities of MaintainX :

1. Create New Work Order
    - Feature: Test the functionality of ‘Create’ button by ensuring that user could be able to create work order successfully with ‘Name’ and ‘Assignee’
    - Type of Tests:

        - Positive Testing: Verify that valid letters and certain range of characters are accepted in Name field by picking an assignee.
        - Negative Testing: Ensure that invalid inputs (e.g., incorrect formats, special characters) are rejected with appropriate error messages.
        - **Usability Testing:** usability testing should be considered to ensure error messages are user-friendly and guide the user to provide valid input.
2. View Work Orders
    - Feature: Test the Work Orders list is arranged in a tabular format indicating ID, Name and Status
    - Types of Tests:
        - **Functional Testing:** Ensures basic functionality like data display, sorting, filtering, Pagination and row selection work correctly from a user's perspective.

3. Work Orders
    - Feature: Test the Functionality of Work Order and ensure that proper assignees are assigned.
    - Types of Tests:
        - Functional Testing
        - UI Testing
4. Close the Work order
    - Feature: Test the functionality of 'Close the work order' button and validate the status of Work order from 'Open' to 'Done'
    - Types of Tests:
        - Functional Testing
5. **Reopen the Work order**
    - **Feature: Test the functionality of 'Reopen the Work order' button and validate the status of work order from 'Done' to 'Open'**
    - **Types of Tests:**
        - **Functional Testing**