**Testing the web UI by giving the required inputs:**

**Introduction:**

  UI Testing, also known as GUI Testing is basically a mechanism meant

to test the aspects of any software that a user will come into contact with. This

usually means testing the visual elements to verify that they are functioning

according to requirements – in terms of functionality and performance. UI testing

ensures that UI functions are bug-free.

Websites comprise web elements created with CSS, JavaScript, and numerous

other programming languages. UI testing performs tests and assertions of these

elements to validate their efficacy. It is focused on examining visual and structural

parts of the software i.e. parts the user would be concerned with, rather than the

internal logic of the software.

UI Testing covers the gamut of visual indicators and graphic-based icons toolbars,

fonts, menus, text boxes, radio buttons, checkboxes, colors, and more.

**Features:**

* Functionality
* Visual Design
* Performance
* Usability
* Compliance

**Scope of UI testing:**

**• Data type errors:** The test checks that only valid data can be entered for certain data

fields such as dates, currency, etc.

**• Field widths:** The test checks that certain text fields do not allow the user to place

inputs over a specific character limit.

**• Navigational elements:** The test checks that all navigational buttons on a page are

working, and that they redirect users to the right page.

**• Progress bars:** The test checks that when displaying pages or screens that take time to

load completely, a progress bar appears to let the user know that the page is loading.

**• Type-ahead:** If the UI uses drop-down lists, type-ahead is required. In a drop-down

menu with multiple options, the user should be able to find the right one by typing the

first letter. Making the user go through a long list constitutes unfavorable user

experience.

**• Table scrolling:** If the website has data tables, and if the table extends into a

second page, then the user should be able to scroll through all the data while

keeping the headers visible and in place.

**• Error Logging:** This test checks that in case of a system error, the software

records error details to a log file so that it can be reviewed later.

**• Menu Items:** The test checks that the software displays only the menu available

in its particular geographical location (if that is applicable).

**• Working shortcuts:** If the software supports shortcuts, this test validates that

each of them works as expected across multiple browsers, platforms, and

devices.

**Challenges of UI Testing:**

1. **Choosing the Right Automated UI Testing Tool**

Without the right automated UI testing tool, testers would have to

manually test the user interface, which would be time-consuming and effort-

intensive. Not to mention, manual testing is prone to human error. However,

choosing a tool that fits into the testing team’s existing workflow is essential. It

should also have record/playback abilities, support reusable tests, and require

minimal maintenance. It should also have in-built mechanisms for reporting and

tracking bugs. Given the huge number of automation frameworks in the market,

one can see why making a choice would be challenging.

1. **Complications of Testing Web Components**

Most websites now include functionalities that require several complex and unique

web components to run. Testing complex elements such as maps, flowcharts,

diagrams, etc., create another layer of difficulting while UI testing.

1. **Testing Constant UI Upgrades**

Modern websites must be upgraded constantly to adapt to users’ ever-evolving

needs and preferences. Often these upgrades involve integration with third-party

tools or their newer versions. Naturally, this gives rise to a new set of

functionalities that must be tested. Repeat this process with every upgrade, and one

realizes where the challenge lies. However, visual regression testing has solved this

for many testing teams having to go through a repeated cycle.

1. **Handling Multiple Errors**

Since UI Testing is almost always an extensive process, creating UI test scripts

for automated UI testing takes maximum time. Thus, when errors do show up,

handling them becomes problematic. Since much of the time and effort has gone

into creating the test script, testers find themselves short on time and resources

when it comes to error resolution.

.

1. **ROI Calculation for Automated UI testing**

Since the UI of any software keeps changing, evolving, and upgrading, UI

automation testing tools must be updated accordingly to keep up. Testing upgrades

almost always involves more investment of time and money, thus making it hard to

offer a definite estimate of ROI. Naturally, this causes trouble when planning

finances for the testing process.

**Four Benefits of Automated UI Testing:**

* Automated UI Testing Saves Time. ...
* Automated UI Testing Enables Faster Feedback. ...
* Automated UI Testing Improves Accuracy. ...
* Automated UI Testing Enables Bugs to Be Discovered More Quickly.

|  |  |
| --- | --- |
|  |  |