**TEST PLAN**

**FORD WEBSITE**

****

**Table of Contents:**

**● Change Log……………………………………………………………..1 ● Document Revision History……………………………………………1 ●** [**1. Introduction**](https://www.guru99.com/test-plan-for-project.html#1)**………………………………………………………….2 ● 2.** [**Scope**](https://www.guru99.com/test-plan-for-project.html#11)**.…………………………………………………………………2 ● *2***[***.1 In Scope***](https://www.guru99.com/test-plan-for-project.html#111)**…………………………………………………………...2 ● *2***[***.2 Out of Scope***](https://www.guru99.com/test-plan-for-project.html#112)**……………………………………………………….3 ● 3.** [**Quality Objective**](https://www.guru99.com/test-plan-for-project.html#12)**.……………………………………………………3 ● *3.1 Primary Objectives..*……………………………………………….3 ● *3.2 Secondary Objectives.* ………………………………………………....3 ● 4.** [**Roles and Responsibilities**](https://www.guru99.com/test-plan-for-project.html#13)**...…………………………………………4 ● 5**[**.**](https://www.guru99.com/test-plan-for-project.html#2) **Test Approaches...……………………………………………………4 ● *5***[***.1* Overview**](https://www.guru99.com/test-plan-for-project.html#21)**…………………………………………………………….5 ● 6. Test Types……………………………………………………………..8**

**● 7. Test Strategy…………………………………………………………10 ●** ***7.1 QArole in test process*…………………………………………………..11 ● *7.2 Bug Triage.……………………………………………………...*11 ● 8**[**.**](https://www.guru99.com/test-plan-for-project.html#3) **Entry and Exit Criteria.……………………………………………..13 ● *8.1 Entry Criteria*……………………………………………………………13 ●** ***8.2 Exit Criteria…………………………………………………….*14 ● 9.** [**Suspension Criteria and Resumption Requirements**](https://www.guru99.com/test-plan-for-project.html#24)**……………....14**

**● 10**[**. Resource & Environment Needs**](https://www.guru99.com/test-plan-for-project.html#4)**…………………………………...14 ● 11.** [**Testing**](https://www.guru99.com/test-plan-for-project.html#41) **Schedule…………………………………………………….15 ● 12.Approvals…………………………………………………………….15 ● 13. Terms/Acronyms…………………………………………………….16**

**Change Log: Version**

Version number

**Change Date**

Date of Change

**By**

Name of person who made changes

**Description**

Description

of the changes were made

1 10/10/2022 1 created

**Document Revision History:**

**Date** **Version** **Author** **Description** **Reviewer** **Approver**

1

**1. INTRODUCTION**

The **Test Plan** is designed to prescribe the *scope, approach, resources,* and *schedule* of all testing activities of the project FORD website **https://www.ford.com**

The plan identifies the *items* to be tested, the *features* to be tested, the *types of testing* to be performed, the *personnel responsible* for testing, the *resources* and *schedule* required to complete testing, and the *risks* associated with the plan.

Customers want a perfect website, which has passed the full cycle of manual and automation testing. Given the specificity of the site it is very important to have the same quality and the site.

**2. SCOPE**

**2.1 In Scope:**

All the features of website FORD which were defined in software requirement [*specs need*](https://docs.google.com/document/d/1rPW5DV82VJT6vtA1VDSrfxaCBuAduxW0zb1yfTh_VMk/edit?pli=1) *to be tested.* The document mainly targets to check the GUI of the FORD main website, Registration form for FORD personal account, Ford F150 Raptor web page, Ford Mustang web page, Ford Bronco web page, check official Ford Dealerships search and verify location, FORD’s API, FORD Website Performance and validates data in report output as per *Requirements Specifications* provided by Client.

**Functions to be tested:**

1. GUI (FORD main web page, F150 web page, Mustang web page, Bronco web page). Manual and automation tests (Selenium IDE)
2. Registration form FORD personal account
3. check official Ford Dealerships search and verify location
4. FORD API collections
5. FORD website Performance Automation test with Lighthouse, GTMetrix and SpeedLab tools
6. Primary Website Security

**2.2 Out of Scope:**

These features are *not to be tested* because they are not included in the software requirement specs.

**Functions *not* to be tested:**

1. Hardware Interfaces
2. Software Interfaces
3. Database logical
4. Communications Interfaces
5. Website Security

**3. QUALITY OBJECTIVES**

The *test objectives* are to verify the functionality of website FORD, the project should focus on testing the Website’s GUI, Registration form,API, Performance and guarantee all these operations can work normally in a real business environment.

**3.1 Primary Objectives:**

A *primary objective* of testing is to: assure that the *system meets the full requirements*, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product. At the end of the project development cycle, the user should find that the project has m*et or exceeded all of their expectations as detailed in the requirements.*

Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification *will be documented* and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

**3.2 Secondary Objectives:**

The *secondary objectives* of testing will be to: identify and expose all *issues* and *associated risks*, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate manner before release. As an objective, this requires *careful and methodical testing* of the website to first ensure all areas of the system are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

**4. ROLESAND RESPONSIBILITIES**

The project should use outsource members as the testers also to save the project cost.

**No.** **Member**

1. Test Manager

2. Testers

3. Developer in Test

**Tasks**

Manage the whole project Define project directions Acquire appropriate resources

Identifying and describing appropriate test techniques/tools/automation.Verify and assess the Test Approach

Execute the tests, Log results, Report the defects.

Implement the test cases, test program, test suite etc.

4. TestAdministrator

5. SQAmembers

Builds up and ensures test environment and assets are managed and maintained

Support Tester to use the test environment for test execution

Take in charge of quality assurance

Check to confirm whether the testing process is meeting

specified requirements

**5. TESTAPPROACHES**

In the project FORD, there are 4 types of testing that should be conducted:

1. Manual Tests
2. Automation Tests
3. API Tests
4. Performance Tests

The project is using an ***Agile approach*,** with weekly iterations (every 14 days).At the end of the *second week* the requirements identified for that iteration will be delivered to the team and will be tested.

Team also must use experience-based testing and error guessing to utilize testers' skills and intuition, along with their experience with similar applications or technologies.

**5.1 Overview:**

**1. T**he **UI/UX *Smoke testing***

will be performed for FORD’s website for verification UI response to Design requirements. Use manual and automation (Selenium IDE) methods.

Sub Plan for UI/UX testing:

1. Verify that all images’links on Homepage are working correctly according to Business requirements.

2. Verify that Logo leads at the website's Homepage after clicking on it.

3. Verify that that Logos: “BRONCO”, “F 150”, “MUSTANG” leads to the corresponding web sites.

4. Verify element present for search local dealership.

5. Verify element present for sign up and create account

6. Verify that at the website's Homepage after clicking on it the “Search” field is working correctly according to Business requirements.

● Positive/Negative/Ad hoc testing should be used *also* if necessary.

**2 Website Manual Tests:**

1. **The FORD Website's** Module **Menu “Dealership Location”** will be tested In this part.

*Check the ability of user to find Ford dealership using users ZIP code :*

1. Valid datas (valid ZIP code) - Positive testing

2. Invalid datas (invalid ZIP code) - Negative testing

User should: find nearest dealership on list and map, choose dealership, follow to dealership web page using link

2. **The FORD Website's** Module **Menu “My Account”** will be tested In this part.

**Submenu “Sign Up”**

*Check the ability of user to create a personal account on FORD 's web page using “@GMAIL.COM” mail domain with:*

1. Valid datas (valid email, according Google requirements) - Positive testing

2. Invalid datas (invalid email) - Negative testing

3. **FORD Website's** Module **Menu “BRONCO”** will be tested In this part.

Submenu **“**EXPLORE BRONCO 2022”

Submenu “BUILD & PRICE”

*Check the ability of user to build a vehicle using configurator and proceed to vehicle order on a local dealership.*

4. **FORD Website's** Module **Menu “F 150”** will be tested In this part.

Submenu **“**EXPLORE F 150 2022”

Submenu “DISCOVER F150 Raptor”

*Check the ability of user to choose a model and trim and proceed to vehicle order on a local dealership.*

5. **FORD Website's** Module **Menu “MUSTANG”** will be tested In this part.

Submenu **“**EXPLORE MUSTANG 2022”

Submenu “DISCOVER 2022 Mustang Shelby® GT500®”

*Check the ability of user to choose a model and trim and proceed to vehicle order on a local dealership.*

Automation tests with Selenium IDE will be *also* created for this testing part.

**Environments for this part:**

**1. OS:**

Edition Windows 10 Pro

Version 21H2

Installed on ‎3/‎30/‎2021

OS build 19044.2130

Experience Windows Feature Experience Pack 120.2212.4180.0Windows 10

**2. Devises:**

Device name DESKTOP-7TO4JMF

Processor Intel(R) Core (TM) i3-7020U CPU @ 2.30GHz 2.30 GHz

Installed RAM 4.00 GB

System type 64-bit operating system, x64-based processor

**3. Browsers:**

● Google Chrome Version 106.0.5249.119 (Official Build) (64-bit)

● Opera (for Windows OS 91.0.4516.77)

**3. Website Automation Tests:**

Tests which are indicated in this part will be automated:

1. Tests for **UI/UX Smoke testing** (to verify the response of UI/UX to design requirements).
2. Test **FORD Website's** Module **Menu “BRONCO”**
3. Test **FORD Website's** Module **Menu “F 150”**
4. Test **FORD Website's** Module **Menu “MUSTANG”**

2. Tests for UI/UX Smoke testing (to verify the response of UI/UX to design requirements).

3. Test for “Search” field (to verify the correct functionality of this field).

**Tools for this part:** ● Selenium IDE

**4. Website API Tests:**

FORDS'sAPI Collections will be tested through the Server response in this part.

API Collections for testing:

**● Asteroids - NeoWs**

NeoWs (Near Earth Object Web Service) is a RESTful web service for near earthAsteroid information. With NeoWs a user can: search forAsteroids based on their closest approach date to Earth, lookup a specificAsteroid with its ----JPL small body id, as well as browse the overall data-set.

**● Earth**

Landsat imagery is provided to the public as a joint project between -------and USGS. ThisAPI is powered by Google Earth EngineAPI, and currently only supports

pan-sharpened Landsat 8 imagery.

**● TLEAPI**

The TLEAPI provides up to date two line element set records, the data is updated daily from [CelesTrak](https://celestrak.com/) and served in JSON format.Atwo-line element set (TLE) is a data format encoding a list of orbital elements of an Earth-orbiting object for a given point in time.

**Tools for this part:**

● Postman 2

● JSON Formatter & Validator

**5. Website Performance Tests:**

Tests for evaluation of the website’s *speed, responsiveness and stability* under a workload will be done in this part*.*

Sub Plan for Performance testing:

1. Check FORD’s website: ● Performance ● Accessibility ● Best practice

● SEO

by using performance testing Tools: 1. Lighthouse

2. GT Metrix 3. SpeedLab

**Tools for this part:** ● Lighthouse ● GT Metrix ● SpeedLab

**6. TEST TYPES**

In the FORD project there are 9 types of testing that should be conducted:

1. Exploratory testing 2. Smoke Testing

3. GUI Testing

4. Functional Testing 5. Positive testing 6. Negative testing 7. ADHOC testing 8. API testing

9. Performance testing

**Exploratory testing :**

Exploratory testing will include a type of software testing where Test cases are not created in advance but QAcheck system on the fly. QAmay note down ideas about what to test before test execution.

**Smoke Testing:**

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for the QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as “Build Verification Testing” or “Confidence Testing.”

In simple terms, we are verifying whether the important features are working and there are no showstoppers in the build that is under testing.

**GUI Testing**:

GUI testing will include testing the UI/UX part of the report. It covers users' Report format, look and feel, error messages, spelling mistakes, GUI guideline violations.

**Functional Testing:**

Functional testing is carried out in order to find out *unexpected behavior* of the report. The characteristics of functional testing are to provide correctness, reliability, testability and accuracy of the report output/data.

**Positive testing:**

Positive testing will includes the type of testing that can be performed on the system

by providing the *valid data* as input. It checks whether an application behaves a*s expected* with positive inputs.

**Negative testing:**

Negative testing will include a method of testing an application or system that ensures that the application is according to the requirements and can handle

the unwanted input and user behavior. *Invalid data* is inserted to compare the output against the given input. Negative testing is also known as *failure testing* or error path testing. When performing negative testing *errors messages* are expected.

**ADHOC testing:**

ADHOC testing will include an informal testing type with an aim to **“break”** the system.

www.linkedin.com/in/kristina-tablev 9

**API testing:**

API testing is a type of software testing that analyzes an application program interface (API) to verify it fulfills its expected functionality, security, performance and reliability.AnAPI test is generally performed by making requests to one or moreAPI endpoints and comparing the response with expected results.

**Performance testing:**

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

**7. TEST STRATEGY**

**7.1 QArole in test process:**

● Understanding *Requirements.*

● Requirement *specifications* will be sent by client.

● Understanding of requirements will be done by QA. ● Preparing *Test Cases*:

QA will be preparing test cases based on the exploratory testing. This will cover all scenarios for requirements.

● Preparing *Test Matrix*:

QA will prepare a RTM which maps test cases to respective requirements. This will ensure the coverage for requirements.

● Reviewing test cases and matrix.

● Peer review will be conducted for test cases and test matrix by QA Lead.

● Any comments or suggestions on test cases and test coverage will be provided by the reviewer respective Author of Test Case and Test Matrix.

● Suggestions or improvements will be re-worked by the author and will be sent for approval.

● Re-worked improvements will be reviewed and approved by the reviewer.

*● Creating Test Data:*

Test data will be created by respective QA on client's developments/test site based on scenarios and Test cases.

● *Executing Test Cases*:

www.linkedin.com/in/kristina-tablev 10

Test cases will be executed by respective QAon the client's development/test site based on designed scenarios, test cases and Test data.

● Test result (Pass/Fail) will be updated in test case document *Defect Logging and Reporting*: QAwill be logging the defect/bugs in Word document and JIRA, found during execution of test cases.

● QAwill inform the respective developer about the defect/bugs.

● QAwill perform Retesting and Regression Testing:

Retesting for fixed bugs will be done by respective QA once it is resolved by the respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

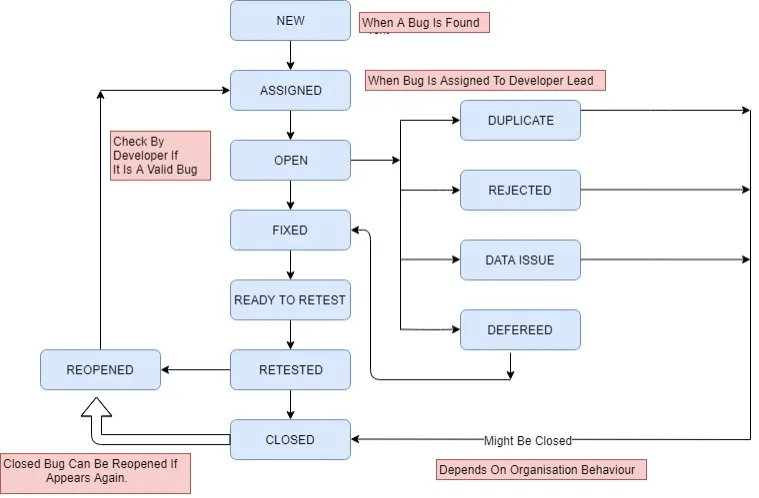
● Deployment/Delivery:

Once all bugs/defects reported after complete testing are fixed and no other bugs are found,the report will be deployed to the client's test site.

Once a round of testing will be done by QA on the client's test site if required Report will be delivered along with sample output by email to the respective lead and Report group.

**7.2 Bug Triage:**

All the issues found while testing will be logged into JIRA

www.linkedin.com/in/kristina-tablev 11

**7.2.1 Bug life cycle:**

**7.2.2 Bug Severity and Priority Definition:**

Bug **Severity** and **Priority** fields are both very i*mportant for categorizing bugs* and *prioritizing* if and when the bugs will be fixed.

The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The Test Lead will be responsible to see that a correct severity level is assigned to each bug.

The QALead, Development Lead and Project Manager will participate in bug review meetings to assign the priority of all currently active bugs.

This meeting will be known as “*Bug Triage Meetings*”. The QALead is responsible for setting up these meetings.

**7.2.3 Bug Severity List:**

**Severity ID Severity** **Severity Description**

The module/product *crashes* or the bug causes *non-*

1 **Highest**

2 **High**

3 Medium

*recoverable* *conditions*. System crashes or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a **Severity 1 bug**.

*Major system components unusable due to failure* or incorrect functionality. **Severity2** bugs cause serious problems such as a *lack of functionality*, or insufficient or unclear error messages that can have a major impact to the user, prevents other areas

of the app from being tested, etc. Severity 2 bugs can have a work around, but the work around is inconvenient or difficult.

*Incorrect functiona*lity of component or process. There is a

simple work around for the bug if it is **Severity 3.**

Documentation errors or signed off **Severity 4** bugs.

Low severity bug occurs when there is *almost no impact* on 4 Low the functionality but it is still a valid defect that should be

corrected.

www.linkedin.com/in/kristina-tablev 12

**7.2.4 Bug Priority List:**

**Priority**

**Priority**

**Level**

**Priority Description**

1 **Highest**

2 **High**

3 Medium

This bug *must be fixed immediately*; the product cannot ship with this bug.

These are important problems that s*hould be fixed as soon*

*as possible*. It would be an embarrassment to the company if this bug shipped.

The problem *should be fixed within the time available*. If

the bug does not delay the shipping date, then fix it.

It is not important (at this time) that these bugs be

4 Low

5 Lowest

addressed. *Fix these bugs after all other bugs* have been fixed.

Documentation errors.

**8. ENTRYAND EXIT CRITERIA**

**8.1 Entry Criteria:**

● All test *hardware platforms* must have been successfully installed, configured, and functioning properly.

● All the necessary d*ocumentation, design, and requirements* information should be available that will allow testers to operate the system and judge the correct behavior.

● All the standard *software tools* including the testing tools must have been successfully installed and functioning properly.

● Proper *test data* is available.

● The *test environment* such as lab, hardware, software, and system administration support should be ready.

● QAresources have completely *understood* the requirements. ● QAresources have strong *knowledge* of functionality.

● *Reviewed* test scenarios, test cases and RTM.

www.linkedin.com/in/kristina-tablev 13

**8.2 Exit Criteria:**

● Acertain level of requirements coverage has been achieved. ● No high priority or severe bugs are left outstanding.

● All high-risk areas have been fully tested, with only minor residual risks left outstanding.

● Cost – when the budget has been spent. ● The schedule has been achieved.

**9. SUSPENSION CRITERIAAND RESUMPTION REQUIREMENTS**

**9.1 Suspension criteria:**

● The *build* contains many serious defects which seriously or limit testing progress.

● Significant *change* in requirements suggested by client. ● Software/Hardware problems.

● Assigned resources are not available when needed by test team

**9.2 Resumption criteria:**

● Resumption will only occur when the problem(s) that caused the suspension have been resolved.

**10. RESOURCEAND ENVIRONMENT NEEDS**

**10.1 Testing Tools:**

**Process**

Test case creation Test case tracking Test case execution Test case management Defect management

Test reporting

**Tool**

Microsoft Word, Microsoft Excel, JIRA JIRA, Confluence

Manual, Selenium IDE

Microsoft Excel, JIRA, Confluence Microsoft Word, JIRA, Confluence

JIRA

www.linkedin.com/in/kristina-tablev 14

API Testing Performance Testing

Automation Testing

Postman

Lighthouse, GT Metrix, SpeedLab

Selenium IDE, XPath (for locators searching)

**10.2 Test Environment x Support level 1 (browsers):**

Test Environment to be setup as per figure below:

● Windows 10: Chrome (latest), Firefox (latest) ● Mac OS : Chrome (latest), Safari (latest)

● Android: Chrome (latest), Firefox (latest)

**Support level 1 (devices):** ● iPhone 12 Pro Max

**Support level 1 (browsers):** ● Safari (latest)

**11. TEST SCHEDULE**

**Task Name** **Start** **Finish** **Effort** **Comments**

**12. APPROVALS**

Project Manager QALead

Name

Signature

www.linkedin.com/in/kristina-tablev 15

**13. TERMS/ACRONYMS**

The below terms are used as examples, please add/remove any terms relevant to the document.

**TERM/ACRONYM** API

GUI

PM

UAT

CM

QA

RTM

**DEFINITION**

Application Program Interface

Graphical user interface

Project manager

User acceptance testing

Configuration Management

QualityAssurance

Requirements Traceability Matrix

www.linkedin.com/in/kristina-tablev 16