# Project Report

|  |  |
| --- | --- |
| Project Title | Plan, Schedule, Test Community Portal |
| Qualification Name (NICF) | NICF-Advanced Certificate in Infocomm Technology (Software and Applications) |
| Product Name | NICF-Capstone Project using Java |
| Module Name (NICF) | NICF-Capstone Project using Java |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name** | | Assessor name | |
| Syukur Sidiq Nur Alam | |  | |
| Date issued | Completion date | | Submitted on |
|  |  | |  |
|  | |  | |
| Project title | Plan, Schedule, Test Community Portal | | |

|  |
| --- |
| Learner declaration |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: 18 August 2023 |

Content

1. Project background
2. Project Objectives
3. Project Requirement Specifications
4. Task 1
5. Task 2
6. Task 3
7. Task 4
8. Task 5
9. Task 6

1. Project Background

This project aims to develop a community portal for software developers, similar to LinkedIn, for ABC Jobs Pte Ltd. This portal aims to connect programmers and software administrators, allowing them to register, find other users, view profiles, post jobs, and more. This project is part of the NICF-Capstone project using Java and students must complete previous modules, including User Interface Framework (Module 3), Database Design and Implementation Materials (Module 4) and Web Development Platform (Module 5).

This project will be a summative assessment for the students of the "NICF-Capstone Project Using Java" course. Students should design, plan and test the community portal they developed in Module 5. In this project, the students have to develop the necessary functionality in the Capstone project and do the pilot phase.

Community Portal Main Page:

The community portal consists of several main pages, including:

1. Community Portal Home
2. Registration Page
3. Registration Confirmation Page
4. Update Profile Page
5. Find User Pages
6. List Search Results Page
7. Public Profile Page
8. Registration Confirmation Email
9. Login page
10. Forgot Password page
11. Forgot Password Confirmation Page

The following functions are part of the Database Design & Will be improved as features are required in Capstone Project, Depending on time availability.

1. Send messages
2. Read Messages
3. Post on Message Boards
4. Message Board List
5. Read Threads
6. Post Job Opportunities
7. List Job Opportunities & Feedback
8. Project Objective

The goal of this project is to define the scope of the tests and establish a plan for the tests, prepare test scenarios and execute the tests. Therefore, we aim to ensure that the Software Developer Community Portal is efficient, meets specific requirements, and provides a seamless user experience.

2.1 Scope of Testing

**In scope**

**Post and apply for jobs:** Test coverage includes features related to job posting and registration of those jobs on the community portal.

**Thread:**

The tests will include a feature to create threads where users can start discussions and post answers to those threads in the community portal.

**Out of scope:**

**Registration Functionality:** Test the user registration process, including displaying a thank you page and sending a confirmation email.

**Search user profiles:** Authentication feature allows users to search other users based on parameters such as first name, last name, company name, city and country.

**View user profiles:** Make sure users can see other users' public profiles after searching.

**Login and Forgot Password**: Check the login process and the ability to recover forgotten passwords.

**Update user profile:** An experimental feature that allows users to update their profile information.

2.2 Preparation of test scripts

Test scenarios will be prepared to ensure full test coverage for all functions in specified test scope. Each test scenario will include step-by-step instructions, test data, and expected results to guide the tester in the execution phase.

2.3 Test Execution

The test execution phase will include running prepared test scripts on the community portal. The testers will follow the specified steps and enter the test data to evaluate the portal performance and identify any issues or errors. Results will be logged and any deficiencies identified will be flagged for further resolution.

By defining test coverage, preparing test scripts, and running tests, we aim to validate the functionality and reliability of the Software Developer Community Portal, ensuring availability ready for deployment and subsequent improvements in the Capstone project

1. Project Requirements Specifications

High-Level Community Portal Functional Features:

For Software Programmers:

**Login:** Software Programmers can log in to their accounts using their credentials.

**Register**: New Software Programmers can create an account and register on the community portal.

**Logout:** Software Programmers can log out of their accounts to end their session.

**Update Profile:**Software Programmers can edit and update their profile information.

**Search User**: The Software programmer can search for other users by parameters like First Name, Last Name, Company Name, and City.

**View Public Profiles:** Programmers can view the public profiles of other users after searching for them.

**Send Messages:** Programmers can send messages to other users on the portal.

**Create Threads:** Programmers can create threads to start discussions.

**Post Replies:** Programmers can post replies to existing threads.

**Post Job Opportunities:** Programmers can post job opportunities on the portal.

**List of Job Opportunities & Feedback:** Programmers can view the job opportunities and responses received.

For Administrator:

**Login:** Administrators can access their accounts with their login information.

**Logout:**To end their session, administrators can log out of their accounts.

**Manage User Data:** Admins can manage user data, such as user information and access rights.

**Send Bulk Emails:** Admins can send bulk emails to invite programmers to register on the community portal.

**Test method:**

**Functional Testing:**

**User Acceptance Testing (UAT):**

* UAT involves testing the community portal from an end-user perspective.
* UAT ensures that the portal meets the requirements and expectations of the users.
* Real users participate in UAT to validate portal usability, functionality, and overall user experience.

**Unit Tests:**

* Unit testing focuses on testing individual components or portal units separately.
* It verifies that each unit functions properly according to its design and requirements.
* Unit tests improve code quality and reliability by detecting and fixing defects early.

**Non-Functional Testing:**

**Compatibility - Cross-Browser Testing:**

* This test ensures the community portal works smoothly on various web browsers (e.g., Chrome, Firefox, Edge.).
* Compatibility ensures the user interface and functionality are consistent and work as expected across browsers.

**Performance Efficiency - Load Testing:**

* Load testing evaluates the portal's performance under expected user loads.
* It identifies the maximum number of concurrent users that the portal can handle without performance degradation.
* Load testing helps optimize portal performance, ensuring responsiveness and scalability.

**Portability (Different Devices):**

* This test evaluates the functionality and usability of the portal across multiple devices (e.g., desktops, laptops, tablets, smartphones).

We can thoroughly evaluate community portal features by categorizing testing techniques into functional and non-functional categories. Functional testing validates the core functionality of the portal, ensuring it meets user requirements. In contrast, non-functional testing addresses critical aspects such as performance, compatibility, and portability, contributing to a robust and user-friendly portal.

Using these selected test methods, we can systematically assess the functional features of the community portal for Software Programmers and Administrators. Thorough testing will help identify and fix any issues, ensuring that the portal meets high-quality standards and provides a smooth user experience.

1. Task 1

Task Statement: . Write a brief requirement specification for the Community Portal.

4.1 Functional requirements:-

System requirements

a. List of all modules to be developed：

Public Module - Home

**User Registration:** The Community Portal must be able to register users with data such as First Name, Last Name, Email, Password, and City.

**Login:** Registered users must be able to log in using their email and password.

**Forgotten Password:** Users should have the option to request forgotten passwords if they forget them. The system must verify their email and then be given access to change the password.

**Update Profile:**Users can Update Profiles when successfully logged into the system.

**Public User Profile:** The Portal will provide a public profile for each user, which other registered users can view. The profile will contain important information such as Full Name, work experience, education, and City.

**Other User Search:** The user should be able to look up other users using criteria like First Name, Last Name, and City.

**Private Module - Administrator/User:**

**Administrator Role:** The portal will have an Administrator role with special rights to manage user data, send bulk emails, and post jobs.

**User Management:** Administrators can view, edit, or delete user profiles.

**Bulk Email:** Administrators must be able to send bulk emails to software programmers.

**Thread:** Users will see the thread on their homepage, and they can apply for jobs in the thread

**Job Posting:** Users must be able to post job opportunities within the portal.

b. All pages in each module with a brief description and their purpose.

|  |  |
| --- | --- |
| Web page | Description |
| Registration Page | To register as a member of the community portal, one must fill in personal information. |
| Registration Confirm | Users can get a confirmation message after registration. |
| Login | Users can log in with data already in the database. |
| Forget Password | Users can forget passwords to change passwords so users can re-enter the system. |
| Update Profile | Users can Update Profiles when successfully logged into the system. Users can add Education and Experience. |
| Search | Users can search for other users. |
| Search Result | Users can see the search result. |
| Others Profile | Users can view other users’ profiles. |
| Homepage | Users can Read and react to a thread. |

c. All pages in each module which are create, edit and view pages

|  |  |
| --- | --- |
| Landing Page | View Page |
| Sign In Page | Create Page |
| Sign Up Page | Create Page |
| Forgot Password Page | Edit Page |
| Homepage | View Page |
| Update Profile Page | Create and Edit Page |
| Search Others Profile Page | Create Page |
| Others Profile Page | View Page |
| Landing Page | View Page |

d. Sitemap of Public & Private Site

Public

Private

4.2:- Non-Functional Requirements:

|  |  |  |
| --- | --- | --- |
| No | Requirements | Purpose |
| 1 | Performance | Ensure the Community Portal operates efficiently and responds quickly to user actions. |
| 2 | Availability | Ensure the Community Portal is accessible to users with minimal downtime. |
| 3 | Security aspects | Protect user data, prevent unauthorized access, and ensure data privacy. |
| 5 | Database backup | Regularly backup the database to prevent data loss and enable recovery in case of failures. |

4.3 Technical Environment Requirements

Hardware

* Memory: 4 GB or more.
* Graphics Card: Intel HD / Radeon Graphics or more.
* CPU: Dual Core Cpu or more.
* HDD: 60 GB or more.
* OS: Windows 10 / 11
  + - * Database server – MySQL server
      * Firewall settings - Port 9091 -- Http

- Port 3306 – Database

Software

* Eclipse IDE: Download and install Eclipse IDE for Java EE Developers. Ensure a version that supports Java EE development, including the necessary tools and plugins for Spring MVC development. (Eclipse Version: 4.26.0)
* Spring Framework: Download and include the Spring framework libraries in your project. (Spring framework 5.2.10)
* Apache Tomcat: Install a servlet container like Apache Tomcat on your machine. We need it to deploy and run the Spring MVC application. Ensure that the servlet container is compatible with the version of the Spring framework you're using. (Apache Tomcat 9.0)
* Maven or Gradle: A build automation tool like Maven or Gradle can simplify the project setup and manage dependencies effectively.
* Java Development Kit (JDK): Install the latest version of JDK on your system. Eclipse requires a compatible JDK to run. (JDK 18.0.2)

Database

* MySQL WorkBench 8.0. (MySQL Server 8.0.31)

4.4 Testing Requirements

**Functional Testing Requirements:**

**User registration:**

* Verify successful registration with all required fields.
* Confirm receipt of the confirmation email after registration.

**Login and Authentication:**

* Test successful login with valid credentials.
* System verification rejected invalid login attempts.

**Forgotten Password:**

* Test the password reset function via the email link provided.

**User Search:**

* User verification can search for others using different parameters.
* Make sure search results are accurate and relevant.

**Public User Profile:**

* Test that the public user profile displays the correct information.
* Make sure the profile is accessible to other registered users.

**User Management (Admin):**

* Test Administrator's ability to manage user profiles.
* Ensure that the system can edit user information and delete user accounts.

**Bulk Email (Admin):**

* Confirm that Administrators can send bulk emails to Users.

**Post Job(Admin):**

* Ensure the admin can post the jobs.

**Non-Functional Testing Requirements:**

**Compatibility Testing / Cross Browser:**

* Test the portal on different web browsers (Chrome, Firefox, Edge) to ensure compatibility.
* Verify that the portal is working correctly in each supported browser.

**Performance Efficiency - Load Testing:**

* Assess portal performance under various load conditions.
* Determine response time and system stability under heavy user traffic.

**Portability Testing (Different Devices):**

* Verify portal responsiveness across devices, including desktops, tablets, and mobile phones.
* Make sure the layout and functionality of the portal adapt to different screen sizes.

1. Task 2

Task Statement: Briefly explain different types of testing, how they will help to create a better system

Functional Testing:

Unit Tests:

* Unit testing focuses on testing individual components or portal units separately.
* It verifies that each unit functions properly according to its design and requirements.
* By detecting and fixing defects early, unit tests contribute to overall code quality and reliability, resulting in a more robust and stable system.

User Acceptance Testing (UAT):

* UAT involves testing the community portal from an end-user perspective.
* Real users participate in UAT to validate portal usability, functionality and overall user experience.
* By incorporating feedback from real users, UAT helps identify user-centric issues and ensures that the system meets user expectations, resulting in a better user experience.

System Testing:

* System testing evaluates the complete community portal as an integrated system.
* It tests the interactions between different modules and components to ensure they work together seamlessly.
* System Testing validates that the portal meets specified requirements and properly performs its intended function by assessing the system.

Non-Functional Testing:

Cross-Browser Testing:

* Cross-Browser Testing verifies that the community portal functions properly in many web browsers (Chrome, Firefox, and Edge).
* This guarantees that the user interface and functionality are uniform and function properly across browsers.
* The portal's accessibility improves, reaches a bigger audience, and gives a better user experience through testing on multiple browsers.

Load Testing:

* Load Testing evaluates the portal’s performance under expected user loads.
* It identifies the maximum number of concurrent users that the portal can handle without performance degradation.
* Load Testing ensures the portal remains responsive and scalable by optimizing portal performance, providing a smooth and reliable user experience even during peak usage.

Usability Testing:

* Usability Testing assesses the community portal's friendliness and ease of use.
* This involves real users performing certain tasks on the portal to identify usability issues or points of confusion.
* By improving the user interface and navigation, Usability Testing improves overall user experience and satisfaction.

1. Task 3

Task Statement: Propose relevant tests suitable for the Project Scenario

Functional testing:

1. Unit testing
2. User Acceptance testing

Non-Functional testing:

1. Cross-browser testing
2. Load Testing
3. Usability testing

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Scenario** | **Number of test cases** |
| TS001-Unit Testing | Test each component in the application | 4 |
| TS002- UAT testing | Ensure the portal’s features align with course requirements and meet users’ expectations. | 2 |
| TS003-Cross Browser testing | Ensure the application works across multiple major browsers | 4 |
| TS004- Load Testing | Testing the performance of the application | 2 |
| TS005 – Usability Testing | Assess the ease of use for tasks | 2 |

1. Task 4

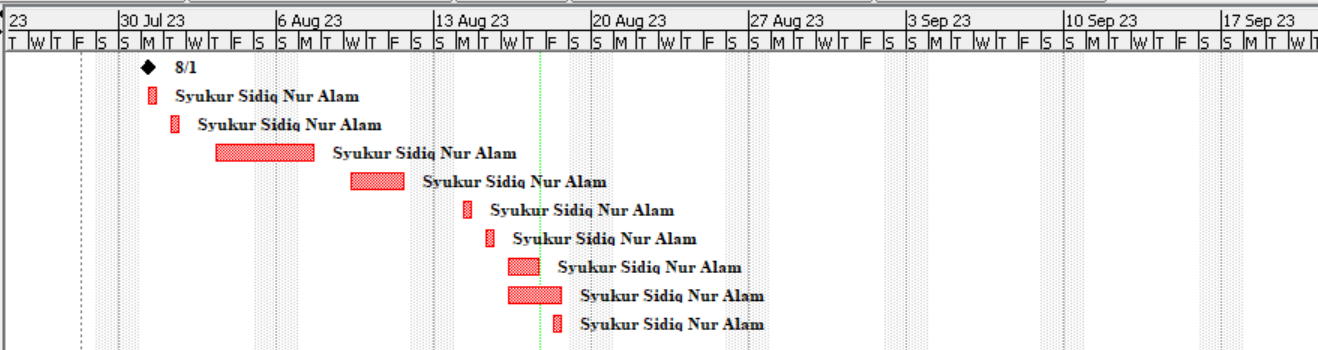
Task Statement:

Write briefly on optimal scheduling times for different tests 2. Prepare a schedule for the execution of tests. Include it as part of the Project Presentation.

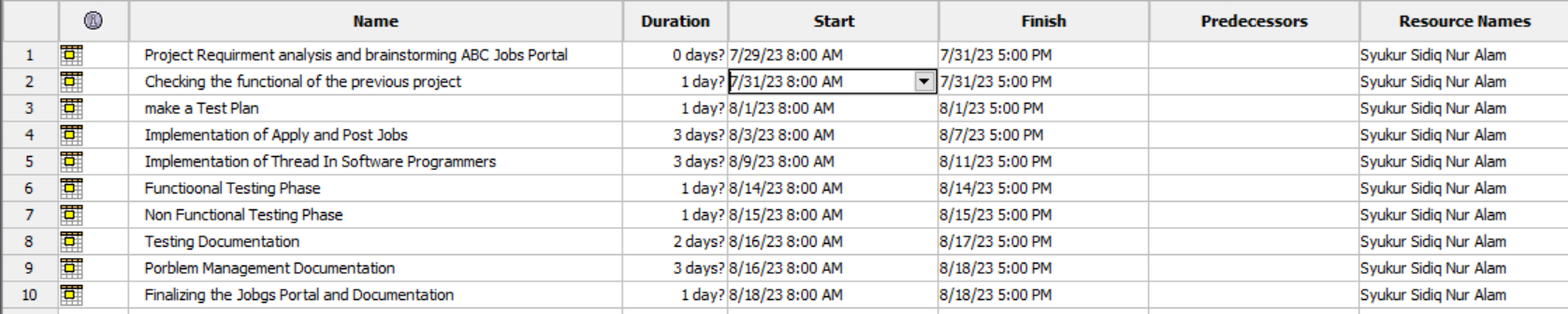
## **Scheduling**

|  |  |  |  |
| --- | --- | --- | --- |
| Web page | Weighting | Functional Points (FP) | Total |
| Apply for Job | 3 | 5 | 15 |
| Post Job | 3 | 5 | 15 |
| Thread | 3 | 5 | 14 |
|  |  | Total functional points | 45 |
|  |  | Estimated person hour per FP | 1.0 |
|  |  | Total hours needed | 45 |
|  |  | Total days needed | 8.5 days |

**Gantt Chart**



**Gantt Table**



**Test Schedule IU6 (Slide 15 and refer to class notes)**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Schedule ID: | TS0001 |  |  |
| Product ID / Name: | ABC Community portal website |  |  |
| Product Version or Build: | 1.0 |  |  |
| Present Owner: | Syukur Sidiq Nur Alam | Software Engineer |  |
| Created On: | 29 July 2023 |  |  |
| Review On: | 18 August 2023 |  |  |
| Review By: | Syukur Sidiq Nur Alam | PM |  |
| Review Comments: | Good |  |  |
| Current Version: | (Make apply/post jobs, adding thread feature) |  |  |
| Change Details: | If your version is 1.0, put “NIL”  If your version is not 1.0, insert the change you made in your Power BI report <Nil> |  |  |
| Current Status: | Approved |  |  |
| Signing Off Authority: | Arvinder Kaur  Syukur Sidiq Nur Alam | QA manager  PM | 18th August 2023 |

1. Task 5 : Phase Test Plan

Task Statement: Develop a Phase Test Plan for testing the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | Start | End | Person Days | Resource |
| Identifying and Brainstorming requirements | 28-7-2023 | 28-7-2023 | 1 | Syukur Sidiq Nur Alam |
| Checking the functional of the previous project | 31-7-2023 | 31-7-2023 | 1 | Syukur Sidiq Nur Alam |
| Make a Test Plan | 01-8-2023 | 01-8-2023 | 1 | Syukur Sidiq Nur Alam |
| Implementation of Apply and Post Jobs | 03-8-2023 | 07-8-2023 | 1 | Syukur Sidiq Nur Alam |
| Implementation of Thread in Software Programmers | 09-8-2023 | 11-8-2023 | 1 | Syukur Sidiq Nur Alam |
| Functional Testing Phase | 14-8-2023 | 14-8-2023 | 1 | Syukur Sidiq Nur Alam |
| Non Functional Testing Phase | 15-8-2023 | 15-8-2023 | 1 | Syukur Sidiq Nur Alam |
| Testing Documentation | 15-8-2023 | 15-8-2023 | 1 | Syukur Sidiq Nur Alam |
| Problem Management Documentation | 16-8-2023 | 16-8-2023 | 1 | Syukur Sidiq Nur Alam |
| Finalizing the Jobs Portal and Documentation | 18-8-2023 | 18-8-2023 | 1 | Syukur Sidiq Nur Alam |

Goal

This test aims to identify and measure the problems/issues in the web application.

Objectives

Validate and verify that system functionality works according to requirements.

**Features not to be tested**

1. Login
2. Logout
3. Registration
4. Update Profile
5. Search for Others
6. Views Others

**Features Be Tested**

1. Post Job
2. Apply for a Job
3. Post Thread
4. Comment on Thread

**Approach**

1. Unit Testing
2. UAT Testing
3. Cross Browser Testing
4. Load Testing
5. Usability Testing

**Suspension Criteria:**

Component tests for thread and task functionality will be paused under the following conditions:

* Discovery of critical errors that prevented the successful completion of the test, such as a severe system crash or data corruption.
* Implemented significant changes to thread and job features that may affect testing.
* Other unforeseen conditions make further testing unfeasible.

**Resumption Criteria:**

Component testing for thread and task functionality will continue once the following criteria are met:

* All issues identified in the suspension criteria have been resolved or appropriately mitigated.
* New software builds containing fixes or updates related to critical errors have been implemented.
* New software build that includes necessary upgrades, addressing identified critical issues, available for testing.

**Exit Criteria (Complete Test):**

Component testing can be considered complete if the following conditions are met:

Scope of Requirements:

* All requirements categorized as High and Medium priority have undergone testing, which includes validation and functional verification.
* The scope of testing includes a thorough evaluation of the functionality, user interaction, and data handling specified by the requirements.

Defect Resolution and Verification:

* Critical and Moderate severity defects identified during testing have been treated with care.
* The resolution of each defect is strictly verified, ensuring that the repair is effective and does not cause unwanted side effects.

**Test Metrics**

|  |  |
| --- | --- |
| **Reported By:** | Syukur Sidiq Nur Alam |
| **Report Date:** | 17/8/2023 |
|  |  |
| **Tested By:** | Syukur Sidiq Nur Alam |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Unit Testing/ TS001 / UT001 / UT002 / UT003 |
| **Total Number of Test Cases** | 4 |
| **No. of Pass:** | 4 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:07:00 |
|  |  |
| **Tested By:** | Syukur Sidiq Nur Alam |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | UAT Testing / TS002 / UAT001 / UAT002 |
| **Total Number of Test Cases** | 2 |
| **No. of Pass:** | 2 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:07:00 |
|  |  |
| **Tested By:** | Syukur Sidiq Nur Alam |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Cross Browser Testing/ TS003 / CB001 / CB002 / CB003 / CB004 |
| **Total Number of Test Cases** | 4 |
| **No. of Pass:** | 4 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:06:00 |
|  |  |
| **Tested By:** | Syukur Sidiq Nur Alam |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Load Testing/ TS004 / LT001 / LT002 |
| **Total Number of Test Cases** | 2 |
| **No. of Pass:** | 2 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:10:00 |
|  |  |
| **Tested By:** | Syukur Sidiq Nur Alam |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Usability Testing / TS005 / UST001 / UST002 |
| **Total Number of Test Cases** | 2 |
| **No. of Pass:** | 2 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:04:00 |
|  |  |
| **SUMMARY LOG STATUS** | |
| **Total Number of Test Cases** | 14 |
| **Total Number of Pass** | 14 |
| **Total Number of Failures** | 0 |
| **Pass %** | 100% |
| **Total Elapsed Time** | 00.34.00 |

**Test Deliverables**

1. Phase Test plan
2. Test schedules
3. Test scripts
4. Test Results
5. Bugs write-up
6. Task 6

Task Statement: Execute the tests and provide the reports as part of the Project Presentation.

**Scenario 1: Unit Testing**

|  |  |
| --- | --- |
| **Test Scenario** | Test each component in the application |
| **TS001** |
| **Test Cases** | Unit Testing |
| **UT001** | Test creating and posting a thread, ensuring the logic is work |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Expected Result | Test Data | Actual Result | Final Result |
| TS001 | UT001 | 5 | All of the code is already developed and ready to test. | The logic in the create\_thread method works as I would expect for handling data and interacting with dependent components. | Image: test\_image  Body: “Test the Body” | As Expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **UT001** |  |
| **UT002** |  |

**Scenario 2: UAT Testing**

|  |  |
| --- | --- |
| **Test Scenario** | Ensure the portal's features align with course requirements and meet users' expectations. |
| **TS002** |
| **Test Cases** | UAT Testing |
| **UAT001** | Identify whether the user can Post Thread in the portal with their account. |
| **UAT002** | Identify whether the user can Comment on another Thread in the portal with their account. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Expected Result | Test Data | Actual Result | Final Result |
| TS002 | UAT001 | 5 | All of the code is already developed and ready to test. | When Log in, they can Post a Thread in The Community Portal and the Thread can see by another users | Image: screenshot (27).png  Body: Test body  Title : test title | As Expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **UAT001** | **Before**      **After** |

**Scenario 3: Cross Browser Testing**

|  |  |
| --- | --- |
| **Test Scenario** | Ensure the application works across multiple major browsers |
| **TS003** |
| **Test Cases** | Cross Browser Testing |
| **CB001** | Ensure Thread compatibility and consistent display on various browsers. |
| **CB002** | Ensure Thread compatibility and consistent display on various browsers. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Expected Result | Test Data | Actual Result | Final Result |
| TS003 | CB001 | 5 | The requirement Browser is ready to use. | User Can Post A thread with Edge Browser and have a consistent display | Image: rodi.png  Body: “ When you have an Independence Day but you still have a task. Be Like: ” | As Expected | Pass |
|  | CB004 | 5 |  | User Can comment on thread with Firefox Browser and have a consistent display | Comment: “Yes you can still use that, and dont install too much extention because it will heavy on your laptop” | As Expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **CB001** | **Before**      **After** |
| **CB004** | **Before**      **After** |

**Scenario 4: Load Testing**

|  |  |
| --- | --- |
| **Test Scenario** | Testing the performance of the application. |
| **TS004** |
| **Test Cases** | Load Testing |
| **LT001** | Identify if the index load time is within 5 Seconds |
| **LT002** | If the Dashboard load time is within 5 Seconds |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Expected Result | Actual Result | Final Result |
| TS004 | LT001 | 5 | Need to use chrome and inspect element to use Dev Tools LightHouse | The index page will load in less than 5 seconds and performance is above 80% | As Expected | Pass |
| LT002 | 5 | The Dashboard page will load in less than 5 seconds and performance is above 80% | As Expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **LT001** |  |
| **LT002** |  |

**Scenario 5: Usability Testing**

|  |  |
| --- | --- |
| **Test Scenario** | To ensure all navigation in header and footer works properly and the content is visible |
| **TS005** |
| **Test Cases** | Usability Testing |
| **UB001** | Test whether the home page is user friendly in design and user can understand the flow of features. |
| **UB002** | Test whether the dashboard is user friendly in design and user can understand the flow of features. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Expected Result | Actual Result | Final Result |
| TS005 | UB001 | 5 | Implementation of ABC Car Portal is complete with all webpages and navigation links and uploaded on localhost server | User will be able to use all navigation links in the home page properly and understand the flow of features available | As Expected | Pass |
| UB002 | 5 | User will be able to use all navigation links in the dashboard page properly and understand the flow of features available | As Expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **UB001** | This is the button to Register. When user want to see the homepage and scroll down, they will see this button  This is the button to Register and user can see it clearly  This is the button to Login and user can see it clearly |
| **UB002** | there is a name and city.  This is the section for posting threads, and below that is a thread had post by users.  This is the buttons to go to dashboard, profile, search another user and logout |