

# Web-App: QA Harbor Limited

**Link: [qaharbor.com](http://qaharbor.com)**

**Group Presented By:**

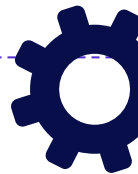
**Md. Musfiq (Lead)**

**Aktaruzzaman Kowshik (Member)**

**[Software Testing & Quality Assurance ]**



# TABLE OF CONTENTS



**01**

**Project Overview**

**02**

**Project  
Methodology**

**03**

**Manual Testing**

**04**

**Performance Testing**

**05**

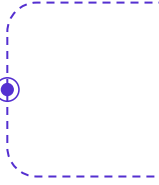
**Automation  
Testing**

**05**

**Conclusion**



# **Web Application : QA Harbor Limited**



# Test Cases : Register Page Functionality

		Project Name	QA Harbor Limited				
		Module Name	Register Page				
		Developed By	Aktaruzzaman Kowshik				
Feature	Test Case Descrption	Pre-condition	Next Steps	Test Data	Expected result	Actual result	Status
Registration	Verify that the design of <b>Register page</b> should be as per as Actual working	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>login</b> button 3: You have tapped on <b>Register Now</b> Button 4: Start your journey with us <b>Candidate</b>	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on login button 3: You have tapped on Register Now Button 4: Start your journey with us Candidate	N/A	Ui should be as per the design	As Expected	Pass
Registration	Verify That user should not be able to Register with blank <b>UserName, email , FirstName, LastName</b>	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>login</b> button 3: You have tapped on <b>Register Now</b> Button 4: Start your journey with us <b>Candidate</b>	Tab on Register Button Without Entering <b>UserName, email , FirstName, LastName</b>	N/A	Below every field there should front-end validation error message should be displayed	As Expected	Pass
Registration	Verify that user should not be able to register by entering invalid <b>email id</b>	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on login button 3: You have tapped on Register Now Button 4: Start your journey with us Candidate	1: enter invalid email id 2: Tab on Register Button	Invalid email used: y3bdhdhyyy	front-end error message should be displayed	As Expected	Pass
Registration	Verify that user should be able to register by entering valid <b>email id and username , Firstname, lastname</b>	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>login</b> button 3: You have tapped on <b>Register Now</b> Button 4: Start your journey with us <b>Candidate</b>	1: enter <b>Valid email id</b> 2: <b>Username, firstname,lastname</b> 3: Tab on Register Button	email used: <b>abcdojana@gmail.com</b> username: <b>AktaruzzamanKowshik</b> Firstname: <b>Aktaruzzaman</b> Lastname: <b>Kowshik</b>	A verification mail has been sent to your email address. Please check your mail and verify to get access of your account.	As Expected	Pass
Registration	Verify that user should be find sign in password in register email address	1: You have launched your <b>email address</b> 2: now check email address and find password	1: You have launched your email address 2: now check email address and find password	N/A	A verification mail has been sent to your email address.	As Expected	Pass

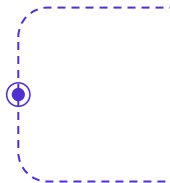


# Automation Testing : Register Functionality

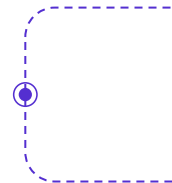
```
import { test, expect } from '@playwright/test';

test('test', async ({ page }) => {
  await page.goto('https://qajobs.qaharbor.com/candidate-registration/');
  await page.getByPlaceholder('Username').click();
  await page.getByPlaceholder('Username').fill('Aktar');
  await page.getByPlaceholder('Email Address', { exact: true }).click();
  await page.getByPlaceholder('Email Address', { exact: true }).fill('abcdojana1@gmail.com');
  await page.getByPlaceholder('First Name').click();
  await page.getByPlaceholder('First Name').fill('aktar');
  await page.getByPlaceholder('Last Name').click();
  await page.getByPlaceholder('Last Name').fill('kowshik')|

  await page.locator('span').filter({ hasText: 'Yes, I agreed the terms of' }).click();
  const page1 = await page1Promise;
  await page.getByRole('button', { name: 'Register Now' }).click();
});
```



# Test Cases: Sign In Page Functionality



		Project Name	QA Harbor Limited				
		Module Name	Sign In Page				
		Developed By	Aktaruzzaman Kowshik				
Feature	Test Case Decsrption	Pre-condition	Next Steps	Test Data	Expected result	Actucal result	Status
Sign In	Verify that the design of Register page should be as per as Actual working	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>Sign In</b> button	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>Sign In</b> button	N/A	Ui should be as per the design	As Expected	Pass
Sign In	Verify That user should not be able to log in with blank <b>UserName</b> or <b>email</b> , <b>password</b>	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>Sign In</b> button	Tab on <b>Sign In</b> Button Without Entering <b>UserName</b> or <b>email</b> , <b>Password</b>	N/A	Below every field there should front-end validation error message should be displayed	As Expected	Pass
Sign In	Verify that user should not be able to <b>Sign In</b> by entering invalid <b>email id</b> should be as per as Actual working	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>Sign In</b> button	1: enter invalid <b>email id</b> 2: Tab on <b>Sign In</b> Button	Invalid email used: pk1@gmail.com	front-end error message should be displayed	As Expected	Pass
Sign In	Verify that user should not be able to <b>Sign In</b> by entering invalid <b>username</b> and <b>Password</b> should be as per as Actual working	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>Sign In</b> button	1: enter invalid <b>username</b> 2: Enter invalid <b>Password</b> 2: Tab on <b>Sign In</b> Button	Invalid username used: aktaruzzaman password : e587f57867c1cb1	front-end error message should be displayed	As Expected	Pass
Sign In	Verify that user should be able to <b>Sign In</b> by entering valid <b>email id</b> and <b>password</b> should be as per as Actual working	1: You have launched <a href="https://qajobs.qaharbor.com/">https:// qajobs.qaharbor.com/</a> 2: User have tapped on <b>Sign In</b> button	1: enter valid <b>email id</b> 2: Enter valid <b>password</b> 3: Tab on <b>Sign In</b> Button	Valid email used: abcdojana@gmail.com used: e587f57867c1cb19ee64671e1c9	Direct go to account Portal	As Expected	Pass



# Automation Testing : Sign in Functionality

```
1 import { test, expect } from '@playwright/test';
2
3 test('test', async ({ page }) => {
4   await page.goto('https://qajobs.qaharbor.com/signin/');
5   await page.getByPlaceholder('Email Address or Username').click();
6   await page.getByPlaceholder('Email Address or Username').fill('abcdojana@gmail.com');
7   await page.getByPlaceholder('Password').click();
8   await page.getByPlaceholder('Password').fill('aktaruzzaman');
9   await page.getByText('Remember Me').click();
10  await page.getByRole('button', { name: 'Sign In' }).click();
11 });
```





# Test Cases: Password Change Functionality



		Project Name	QA Harbor Limited				
		Module Name	Password Change				
		Developed By	Aktaruzzaman Kowshik				
Feature	Test Case Descrption	Pre-condition	Next Steps	Test Data	Expected result	Actual result	Status
Password Change	Verify that the design of <b>Change Password</b> page should be as per as Actual working	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and <b>Click Here</b> Button Tapped	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and <b>Click Here</b> Button Tapped	N/A	Ui should be as per the design	As Expected	Pass
Password Change	Verify That user should not be able to Change password with blank <b>New Password</b> and <b>Confirm password</b>	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: 3: Then change password and <b>Click Here</b> Button Tapped	Tab on <b>Save Changes</b> Button Without Entering <b>New password</b> and <b>Confirm password</b>	N/A	Below every field there should be "Please fill out this field" Message	As Expected	Pass
Password Change	Verify that user should not be able to <b>change password</b> correctlywhen text length not to at least 8 characters	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: 3: Then change password and <b>Click Here</b> Button Tapped	1: enter invalid <b>new password</b> 2: Enter invalid <b>Confirm password</b>	used: aktar used: aktar	Below message "Please increase the text length to at least 8 characters"	As Expected	Pass
Password Change	Verify that user should be able to <b>change password</b> correctly when text length to at least 8 characters	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: 3: Then change password and <b>Click Here</b> Button Tapped	1: enter valid <b>new password</b> 2: Enter valid <b>Confirm password</b>	used: aktaruzzaman used: aktaruzzaman	Direct go to account Portal	As Expected	Pass



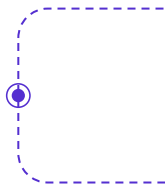


# Automation Testing : Password Change

## Functionality:

```
import { test, expect } from '@playwright/test';

test('test', async ({ page }) => {
  await page.goto('https://qajobs.qaharbor.com/reset-password/');
  await page.getByLabel('Email Address').click();
  await page.getByLabel('Email Address').fill('abcdojana@gmail.com');
  await page.getByRole('button', { name: 'Reset Password' }).click();
  await page.getByRole('link', { name: 'Jobs', exact: true }).click();
  await page.getByRole('link', { name: 'Post A Job Post A Job' }).click();
  await page.getByPlaceholder('Email Address or Username').click();
  await page.getByPlaceholder('Password').click();
  await page.getByRole('button', { name: 'Sign In' }).click();
  await page.getByRole('link', { name: 'Lost Password?' }).click();
  await page.getByLabel('Email Address').click();
  await page.getByRole('button', { name: 'Reset Password' }).click();
});
```



# Test Cases: Jobs Search Functionality



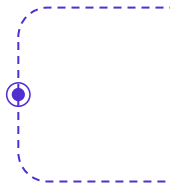
		Project Name	QA Harbor Limited				
		Module Name	Filter Jobs				
		Developed By	Aktaruzzaman Kowshik				
Feature	Test Case Decsrption	Pre-condition	Next Steps	Test Data	Expected result	Actucal result	Status
Filters Job	Verify that the design of <b>Job Filter</b> page should be as per as Actual working	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and <b>Job</b> Button Tapped	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and Job Button Tapped	N/A	Ui should be as per the design	As Expected	Pass
Filters Job	Verify That user should able to <b>Category</b> Selection	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and <b>Job</b> Button Tapped	1: User have Select <b>Manual Tester</b> 2: User have Select <b>Automation Tester</b>	N/A	All <b>Automation Tetster</b> and <b>Manual Tester</b> Job showing the portal	As Expected	Pass
Filters Job	Verify That user should able to <b>Vacancy Type</b> Selection	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and <b>Job</b> Button Tapped	1: User have Select <b>Hybrid</b> 2: User have Select <b>On Site</b> 3: User have select <b>Remote</b>	N/A	All <b>Vacancy Type</b> Job showing the portal	As Expected	Pass
Filters Job	Verify That user should able to <b>Vacancy Status</b> Selection	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and <b>Job</b> Button Tapped	1: User have Select <b>Full Time</b> 2: User have Select <b>Intern</b> 3: User have select <b>Part Time</b>	N/A	All <b>Vacancy Status</b> Job showing the portal	As Expected	Pass
Filters Job	Verify That user should able to <b>Reset</b> Selection Button and <b>Reset</b> all document Filter	1: User have tapped on <b>Sign In</b> button 2: correct information then tapped <b>Sign in</b> , automatic go to <b>account</b> 3: Then change password and <b>Job</b> Button Tapped	1: User have Select <b>Reset</b> Button	N/A	All <b>Job Filter</b> are Reset	As Expected	Pass



# Automation Testing : Find Job Functionality

```
import { test, expect } from '@playwright/test';

test('test', async ({ page }) => {
  await page.goto('https://qajobs.qaharbor.com/find-jobs/');
  await page.getByPlaceholder('Search...').click();
  await page.getByPlaceholder('Search...').fill('dhaka');
  await page.getByLabel('Division').selectOption('77');
  await page.getByRole('button', { name: 'Search' }).click();
  await page.getByPlaceholder('From').click();
  await page.getByRole('link', { name: '17' }).click();
  await page.getByPlaceholder('To').click();
  await page.getByRole('link', { name: '25' }).click();
  await page.getByLabel('Category').selectOption('59');
  await page.getByLabel('Category').selectOption('60');
  await page.locator('label').filter({ hasText: 'Hybrid' }).locator('i').click();
  await page.locator('label').filter({ hasText: 'On Site' }).locator('i').click();
  await page.locator('label').filter({ hasText: 'Remote' }).locator('i').click();
  await page.locator('label').filter({ hasText: 'Full Time' }).locator('i').click();
  await page.locator('label').filter({ hasText: 'Intern' }).locator('i').click();
  await page.locator('label').filter({ hasText: 'Part Time' }).locator('i').click();
  await page.locator('label').filter({ hasText: 'On Site' }).locator('div').click();
  await page.locator('label').filter({ hasText: 'Remote' }).locator('i').click();
  await page.locator('label').filter({ hasText: 'Hybrid' }).locator('i').click();
});
```



# Test Cases: Blog Functionality



Test ID	Test Case Title	Pre-condition	Description	Test Steps	Expected Result	Actual Result	Status	Priority
T_1.0	Verify Blog Page Navigation	User is on the homepage	Check if the blog link navigates correctly	1. Open homepage and Observe if the page content is displayed without errors. 2. Click on the "Blog" URL 'https://qaharbor.com/blog/'	User should be navigated to https://qaharbor.com/blog/ page with all blog posts displayed.	Blog page should load successfully with all content visible.	Pass	High
T_2.0	Verify Blog Post Content Visibility	Blog page loaded successfully	Ensure all blog titles and excerpts are visible	1. Navigate to the Blog page. 2. Verify that titles and short descriptions for each blog post are displayed.	All blog titles and excerpts should be properly visible.	All blog content visible successfully.	Pass	High
T_3.0	Check Pagination on Blog Page	Blog page contains multiple posts (pagination)	Ensure blog pagination functions correctly	1. Scroll to the bottom of the Blog page. 2. Click on the "Next Page or 1, 2, 3" button. 3. Verify that additional blog posts load.	Next set of blog posts should load when clicking "Next Page".	Next page blog page loaded successfully.	Pass	Medium
T_4.0	Verify Blog Image Loading	Blog page loaded successfully	Validate that images for each blog post load	1. Navigate to the Blog page. 2. Verify that all images associated with blog posts load properly without errors.	Images for all blog posts should load and display correctly.	All images of all blog posts display correctly.	Pass	Medium
T_5.0	Test Search Functionality	Blog page contains a search feature (if any)	Validate search accuracy	1. Locate search bar (if available). 2. Enter a keyword (e.g., "API Testing"). 3. Verify the relevant posts are listed in search results.	Search results should accurately reflect the entered keyword, displaying relevant posts.	Search results displaying accurately.	Pass	High



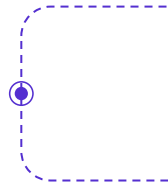


# Test Cases: Blog Post Comment Functionality

Test ID	Test Case Title	Pre-condition	Description	Test Steps	Expected Result	Actual Result	Status	Priority
T_1.0	Verify valid comment submission	User is on the blog post page and comment section is loaded.	Ensure user can submit a valid comment.	1. Enter a comment of at least 20 characters. 2. Fill in valid name and email. 3. (Optional) Add website URL. 4. Click 'Post Comment'.	Comment is submitted successfully with confirmation: 'Your comment is awaiting moderation.'	Submitted successfully.	Pass	High
T_2.0	Verify name field with valid input	User is on the blog post page and comment section is loaded.	Ensure valid name input is accepted.	1. Fill in a valid comment. 2. Enter a valid name (e.g., 'Md. Musfiq'). 3. Fill in valid email and click 'Post Comment'.	Comment is submitted successfully with confirmation.	Submitted successfully.	Pass	Medium
T_3.0	Verify email field with valid input	User is on the blog post page and comment section is loaded.	Ensure valid email input is accepted.	1. Fill in a valid comment. 2. Enter a valid email (e.g., 'musfiq.cse@gmail.com'). 3. Fill in valid name and click 'Post Comment'.	Comment is submitted successfully with confirmation.	Submitted successfully.	Pass	Medium
T_4.0	Verify optional website field with valid input	User is on the blog post page and comment section is loaded.	Ensure valid website URL is accepted.	1. Fill in a valid comment. 2. Fill in valid name and email. 3. Enter a valid URL in the website field. 4. Click 'Post Comment'.	Comment is submitted successfully with confirmation.	Submitted successfully.	Pass	Low
T_5.0	Verify checkbox saves details for next session	User is on the blog post page and comment section is loaded.	Ensure checkbox saves user details for future comments.	1. Fill in all fields with valid input. 2. Check the 'Save my name, email, and website' checkbox. 3. Click 'Post Comment'. 4. Verify user data is retained on next visit.	Comment is submitted successfully, and data is saved for the next session.	Successfully done.	Pass	Low
T_6.0	Verify comment with less than 20 characters	User is on the blog post page and comment section is loaded.	Ensure error is shown for comments under 20 characters.	1. Enter a comment with fewer than 20 characters. 2. Fill in valid name and email. 3. Click 'Post Comment'.	Error message: 'Minimum 20 characters are required to submit the comment.'	Show error message.	Pass	High



# Automation Testing : Blog Comment Functionality



```
import { test, expect } from '@playwright/test';

test('test', async ({ page }) => {
  await page.goto('https://qaharbor.com/');
  await page.getByRole('navigation').getByRole('link', { name: 'Blog' }).click();
  await page.getByRole('link', { name: 'Load Testing: Distributed JMeter Testing and Analyze Results', exact: true }).click();
  await page.getByLabel('Comment *').click();
  await page.getByLabel('Comment *').fill('good article');
  await page.getByLabel('Name *').click();
  await page.getByLabel('Name *').fill('aktaruzzaman Kowshik');
  await page.getByLabel('Email *').click();
  await page.getByLabel('Email *').fill('akowshik1@gmail.com');
  await page.getByLabel('Website', { exact: true }).click();
  await page.getByLabel('Website', { exact: true }).fill('www.aktaruzzaman.com');
  await page.getByLabel('Save my name, email, and').check();
  await page.getByRole('link', { name: 'See more posts' }).click();
  await page.getByRole('link', { name: 'Emerging QA Metrics Every' }).first().click();
});
```



# Performance Testing in JMeter [QA Harbor

Test Plan

- QA Harbor Limited
  - Find Jobs Page
  - Register
  - Home Page
  - View Results Tree
  - Summary Report
  - Graph Results

Thread Group

Name: QA Harbor Limited

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread

Thread Properties

Number of Threads (users): 30

Ramp-up period (seconds): 1

Loop Count: ☐ Infinite 2

Text

- Register
  - Register-0
  - Register-1
  - Find Jobs Page
  - Find Jobs Page
  - Find Jobs Page
  - Register
  - Register
  - Find Jobs Page
  - Register

Sampler result Request Response data

Thread Name:QA Harbor Limited 1-11

Sample Start:2024-12-18 15:02:19 BDT

Load time:3887

Connect Time:0

Latency:63

Size in bytes:198121

Sent bytes:290

Headers size in bytes:1826

Body size in bytes:196295

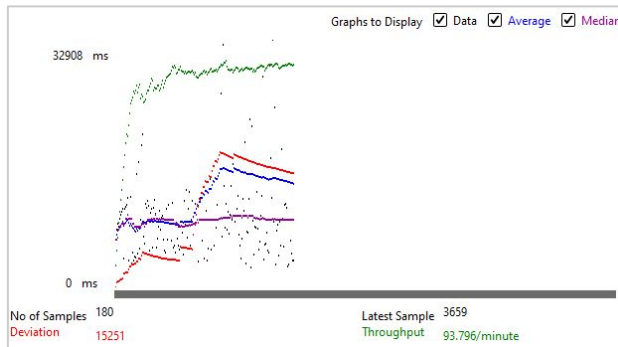
Sample Count:1

Error Count:0

Data type ("text"|"bin"|""):text

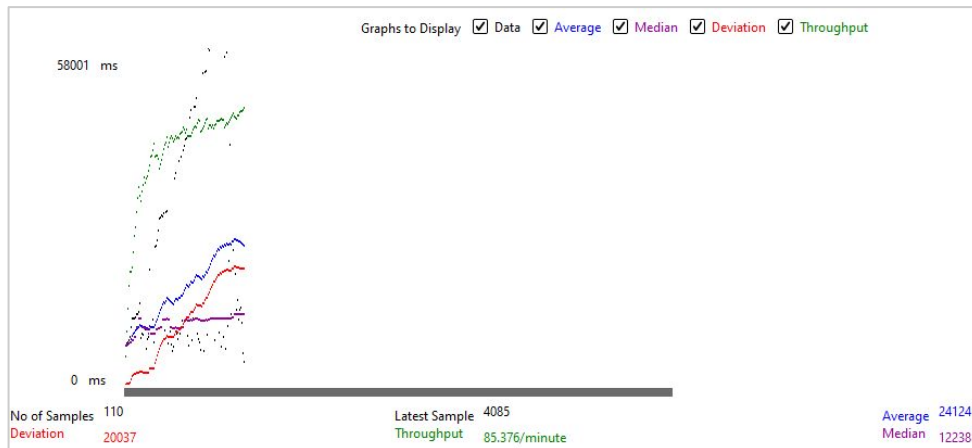
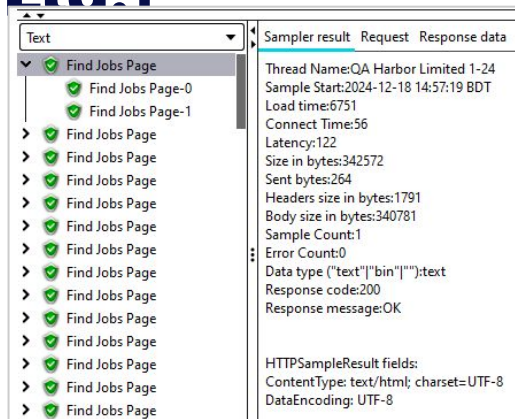
Response code:200

Response message:OK



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Find Jobs Page	60	20728	5368	65636	19935.68	0.00%	33.2/min	185.12	0.14	342589.2
Register	60	8763	2756	60244	12399.67	0.00%	34.4/min	110.81	0.16	198123.9
Home Page	60	12022	3659	65437	8373.67	0.00%	34.4/min	230.20	0.14	410708.2
TOTAL	180	13838	2756	65636	15251.62	0.00%	1.6/sec	484.16	0.41	317140.4

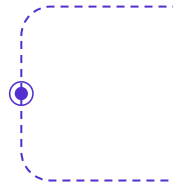




Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Find Jobs Page	110	24124	4085	72246	20037.46	0.00%	1.4/sec	476.05	0.37	342587.7
TOTAL	110	24124	4085	72246	20037.46	0.00%	1.4/sec	476.05	0.37	342587.7



# Conclusion



The testing phase involves evaluating specific aspects of a project to ensure it meets predefined objectives. We will do some more testing later .

- **Functionality Testing:** Verifying that all features and components perform their intended tasks.
- **Performance Testing:** Assessing the system's efficiency under various workloads.
- **Component Testing:** Testing individual modules to ensure they operate correctly within the system.

