[Date]

Three Musketeers

Test Report

FSMA Project

# Table of Contents

[**Table of Contents**](#_h3arz2n0t75b) **2**

[**Web Performance Test**](#_214kbo88ur2x) **3**

[Screenshots](#_c1wgvccshqte) 3

[Performance Tests](#_4lsdkumklyuu) 3

[Load Tests](#_xpuv6bkktean) 7

[**Static Code Analysis**](#_14u390upp2gq) **10**

[Risks and Resolutions](#_95e7aop4jbyc) 10

[Screenshots](#_vse1rvqdx30f) 10

[**Grey Box Tests**](#_dwod6tzaoi2f) **11**

[**White Box Tests**](#_3241gy90v7tz) **11**

[**Black Box Tests**](#_i6ihycnd5b2j) **11**

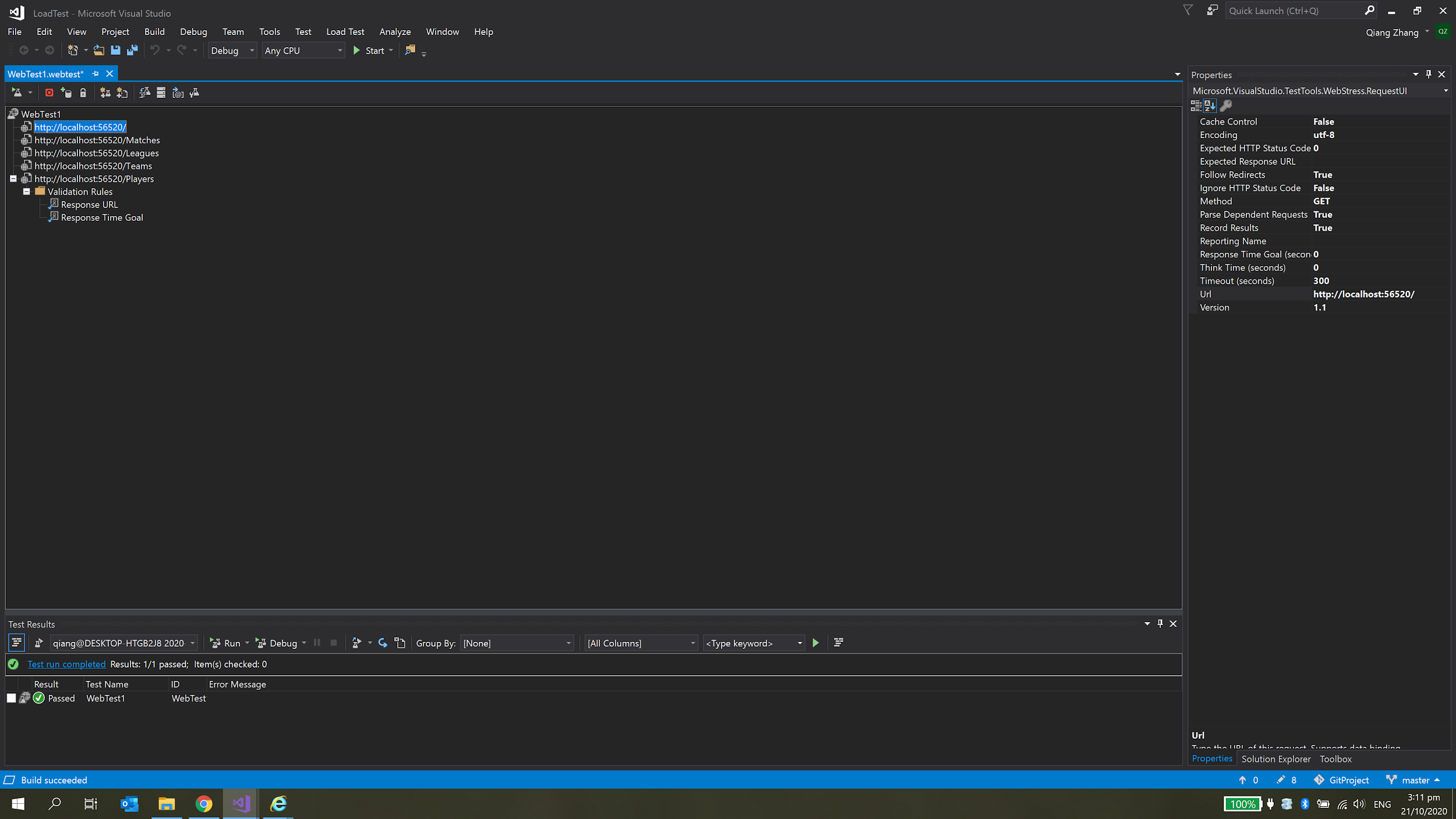
# Web Performance Test

Because this is a web application, sometimes many clients will access it at the same time. Performance testing could check the performance when many clients access the website. And load testing could identify the performance of the application under maximum workload. This is important for web applications.

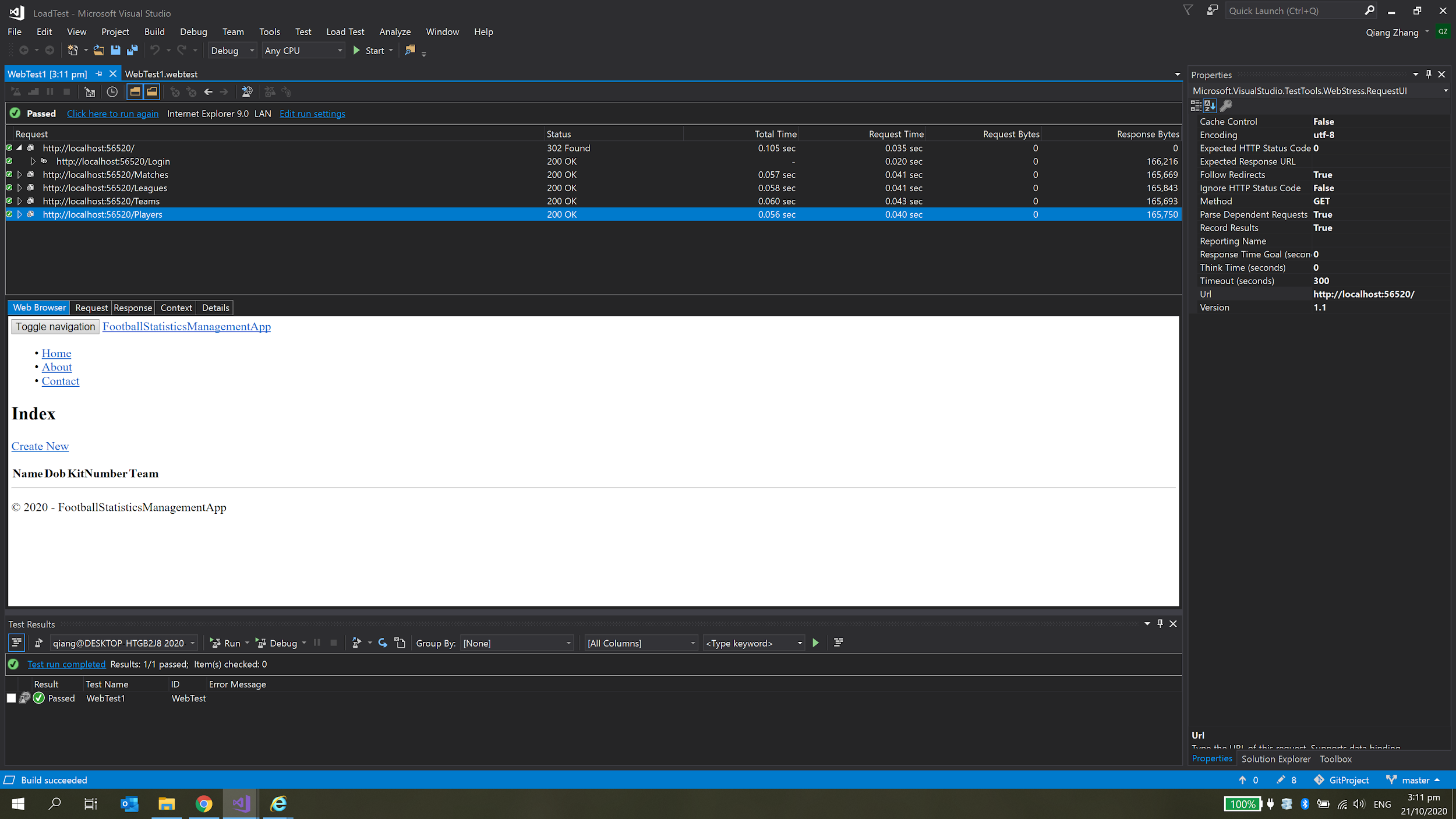
Generated Report: *please see PerformanceLoadTest.xlsx.*

## Screenshots

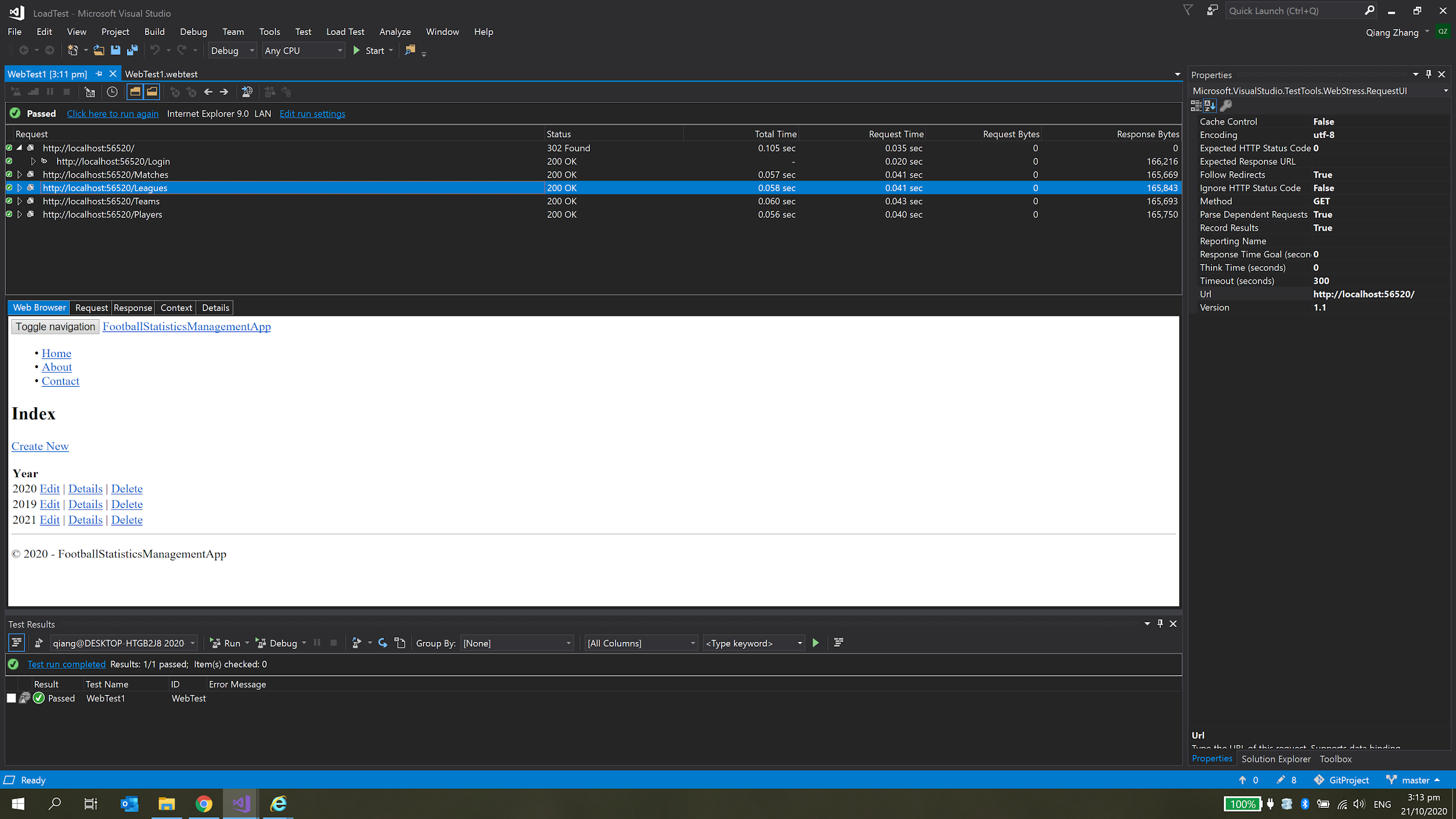
### Performance Tests



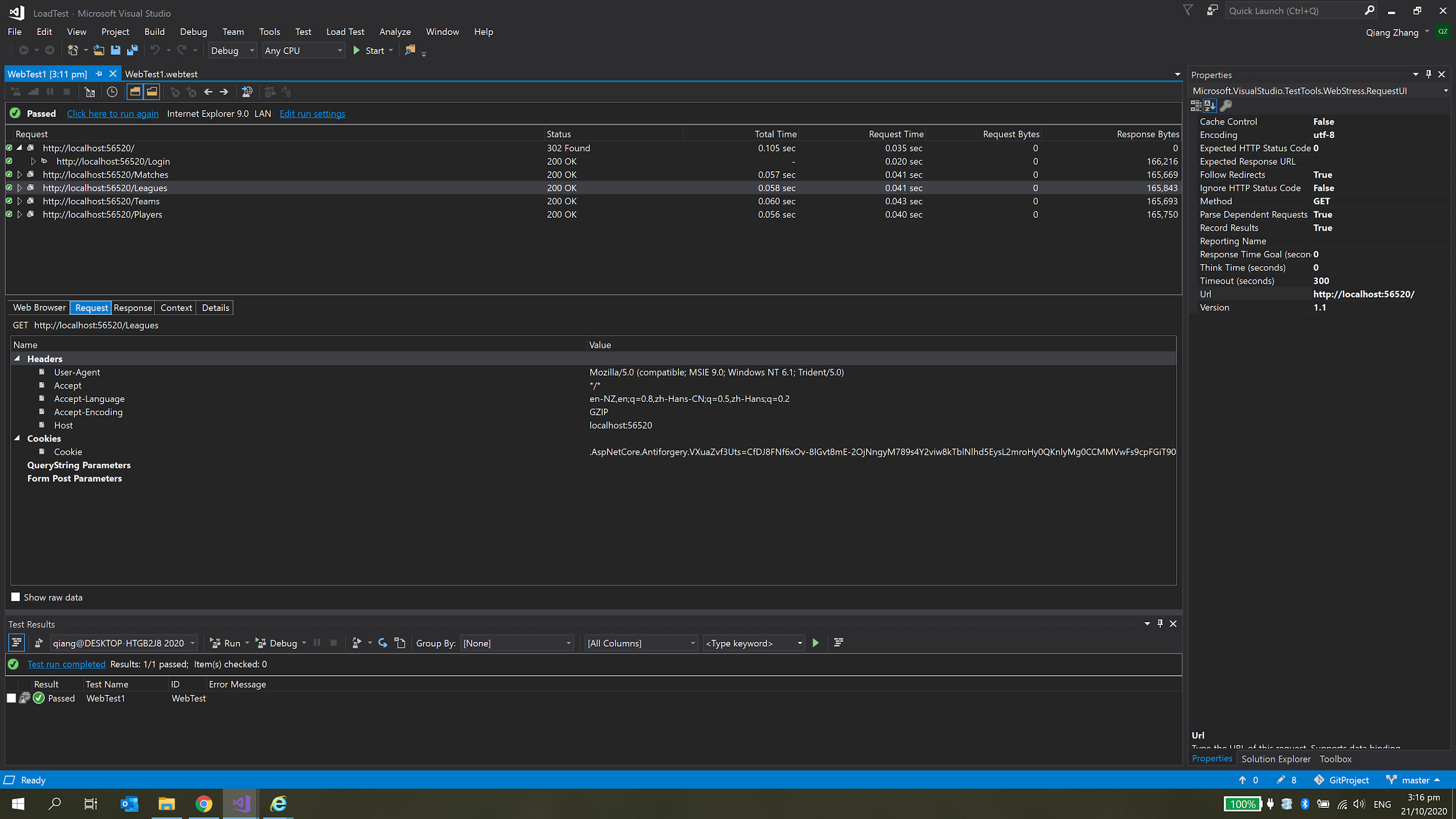
*Figure 1.1: Recording website*



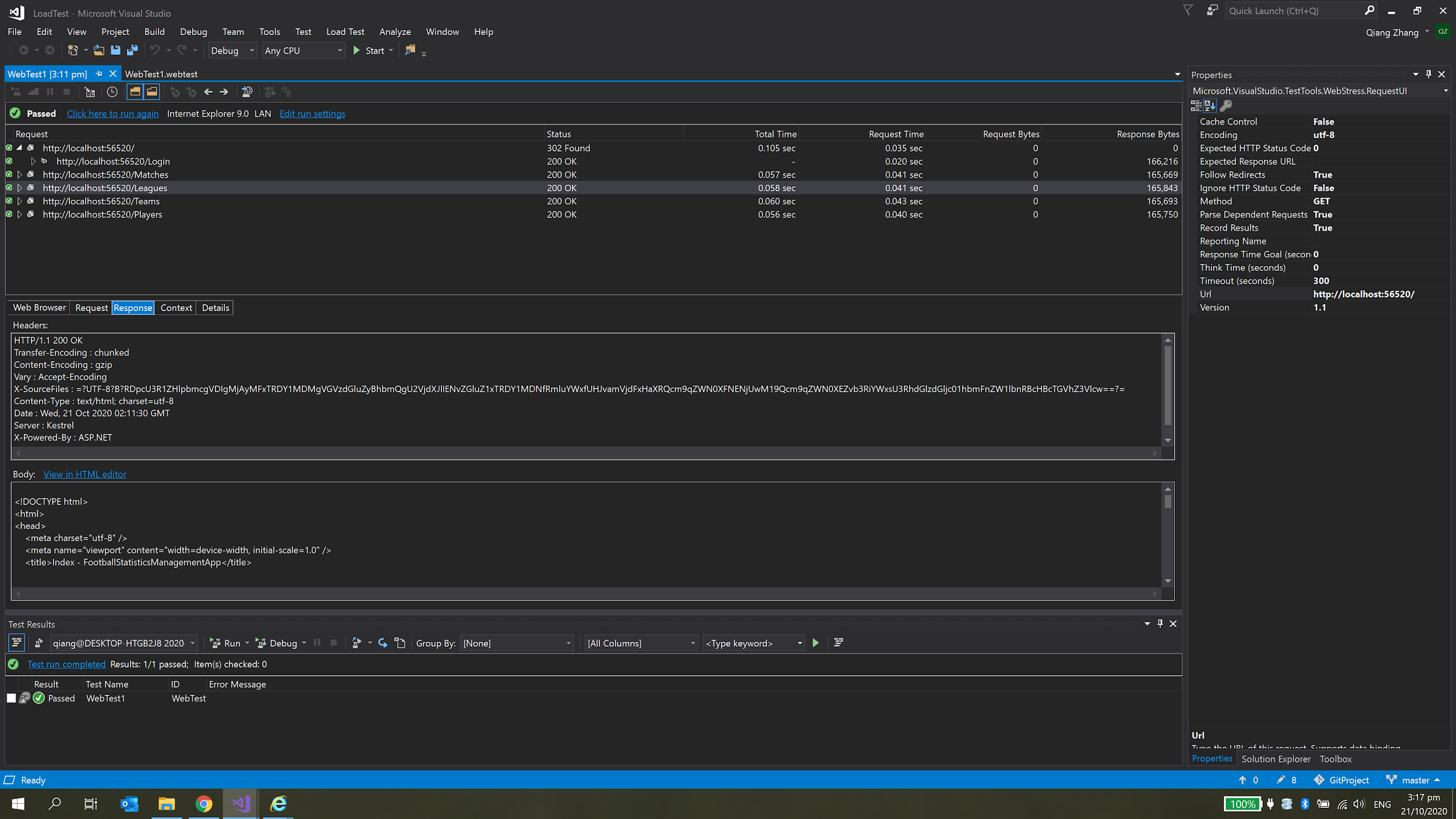
*Figure 1.2 Run the recorded test*



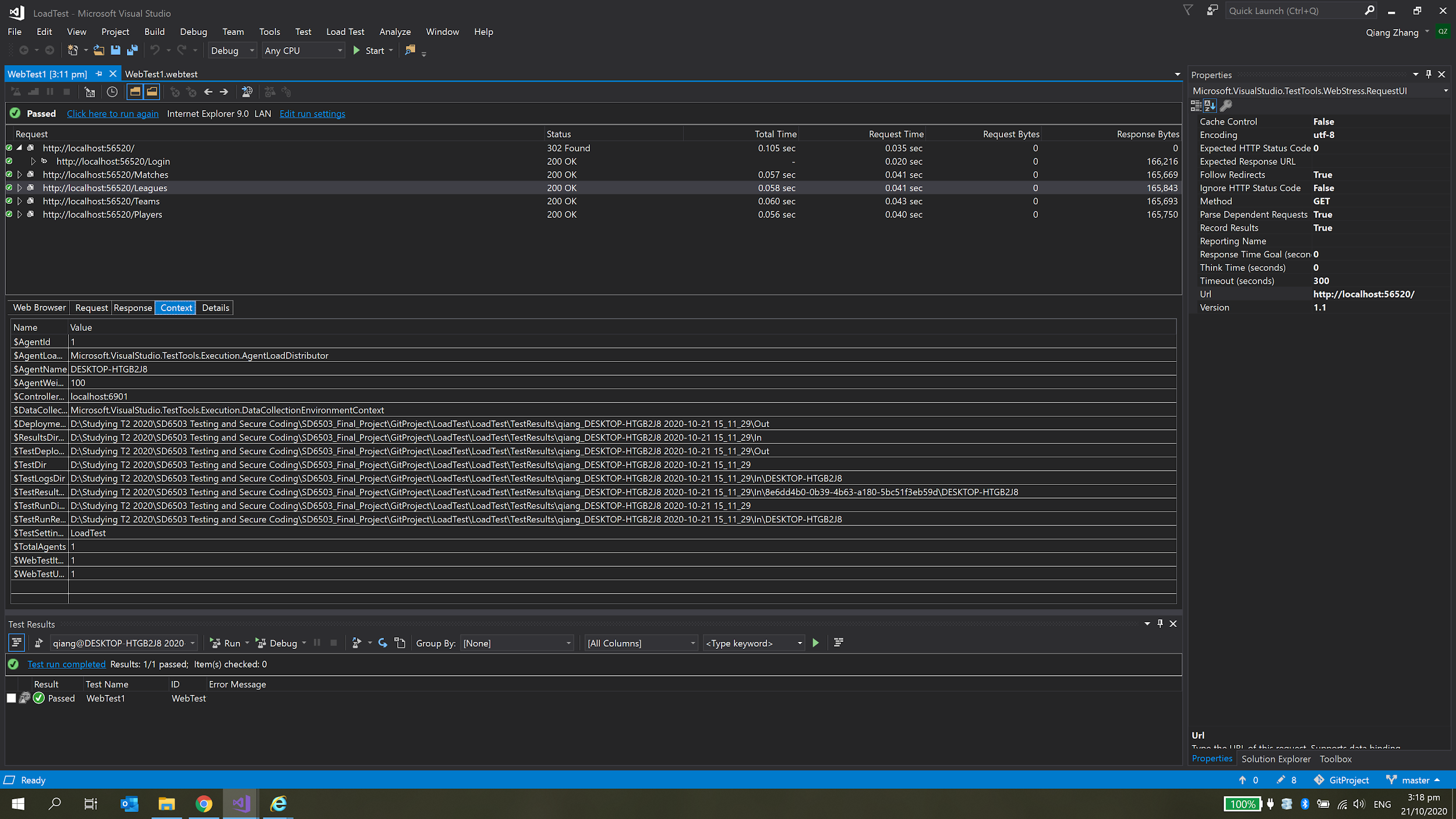
*Figure 1.3 the results about the league page*



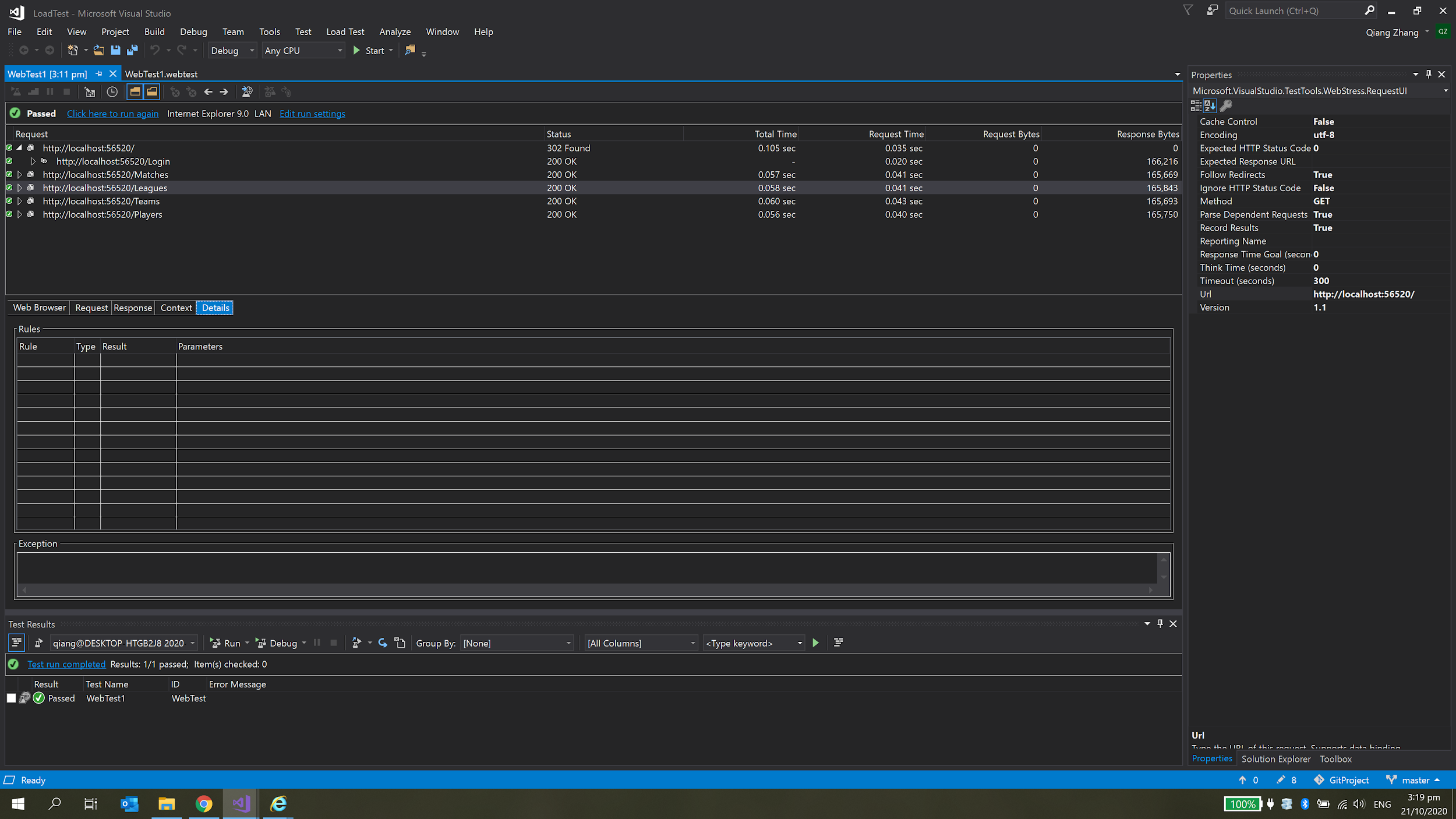
*Figure 1.4 the result about leagues page’s request*



*Figure 1.5 the result about leagues page’s response*

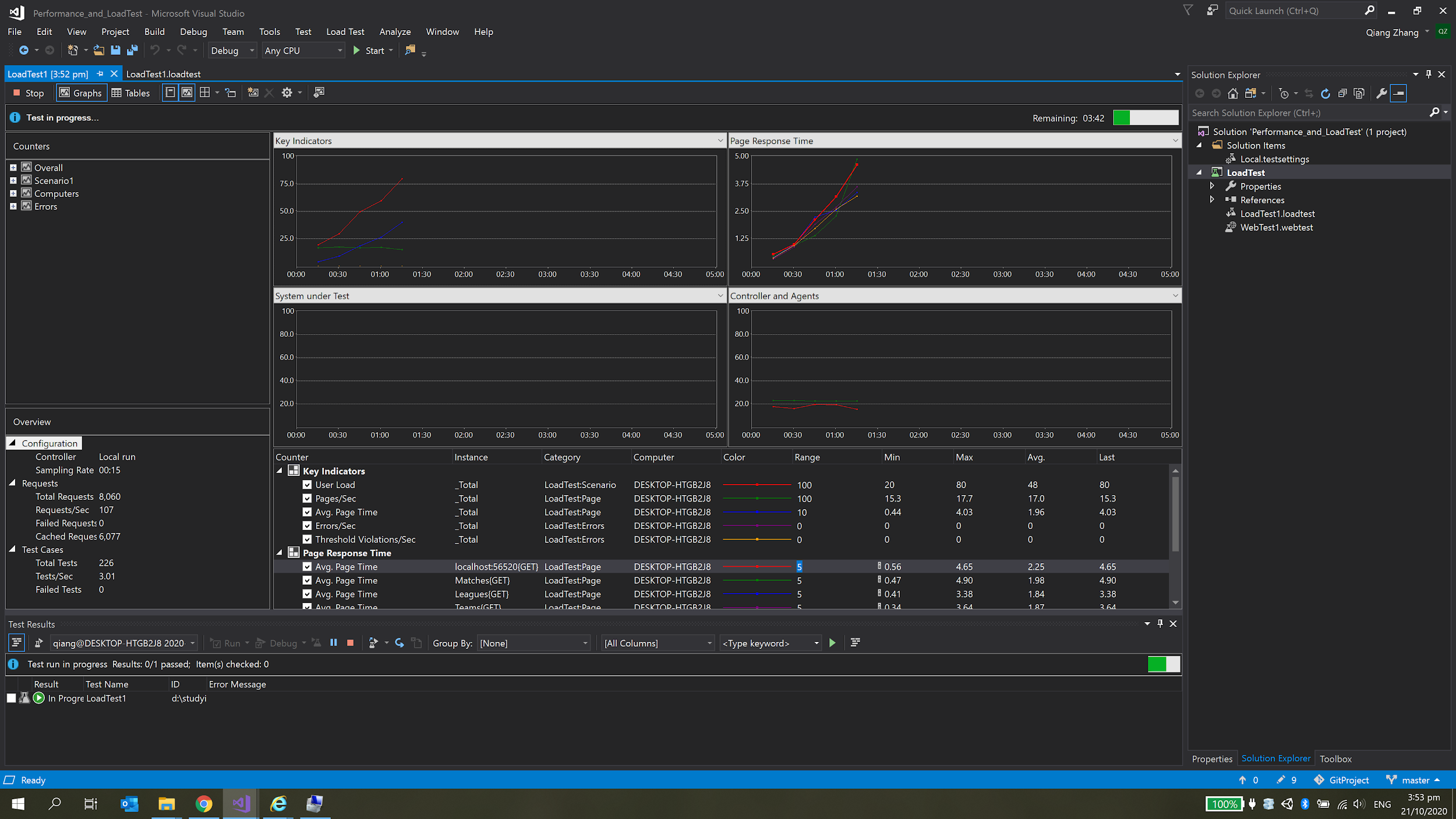


*Figure 1.6 the result about leagues page’s contest*

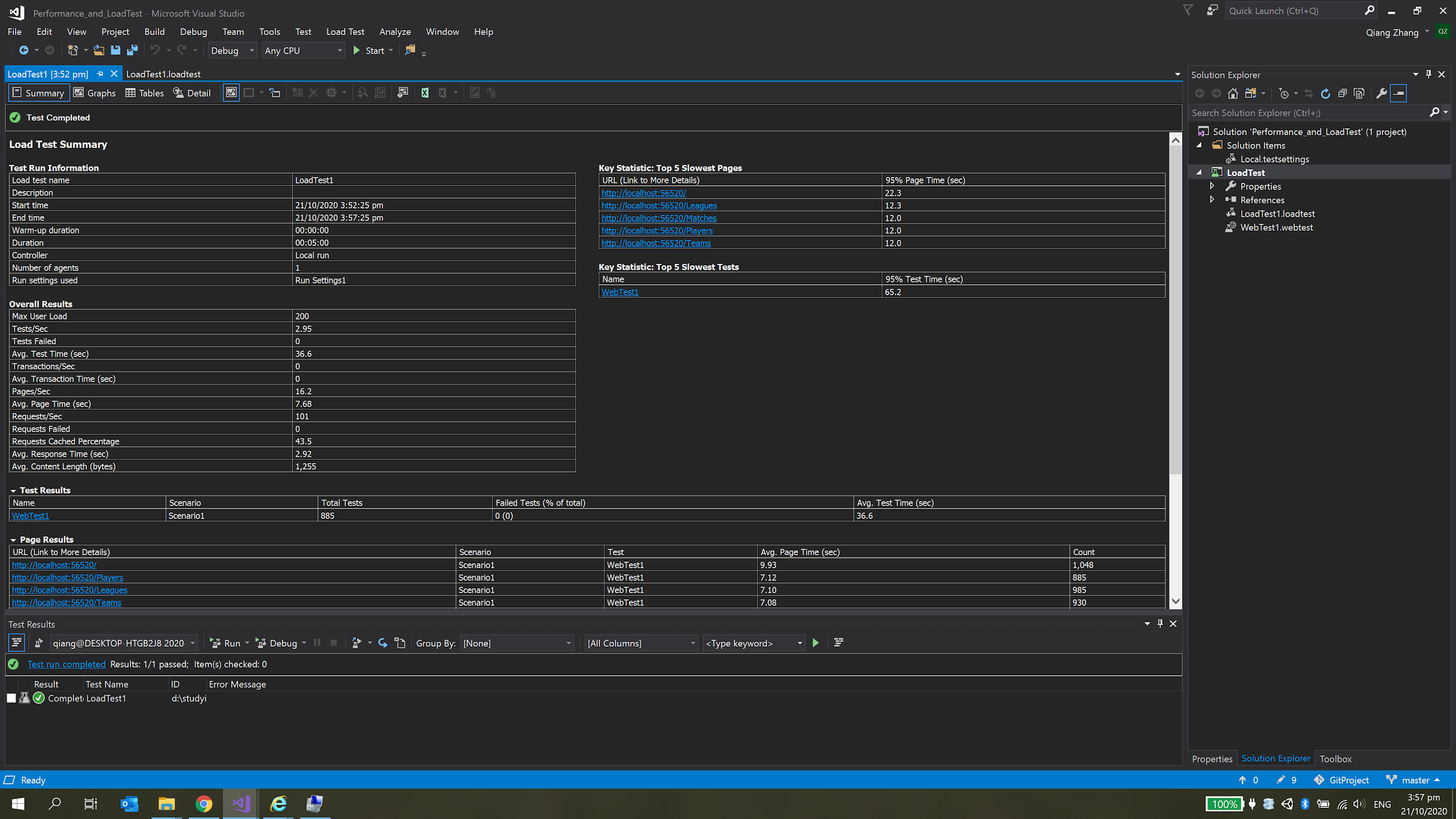


*Figure 1.7 the result about leagues page’s details*

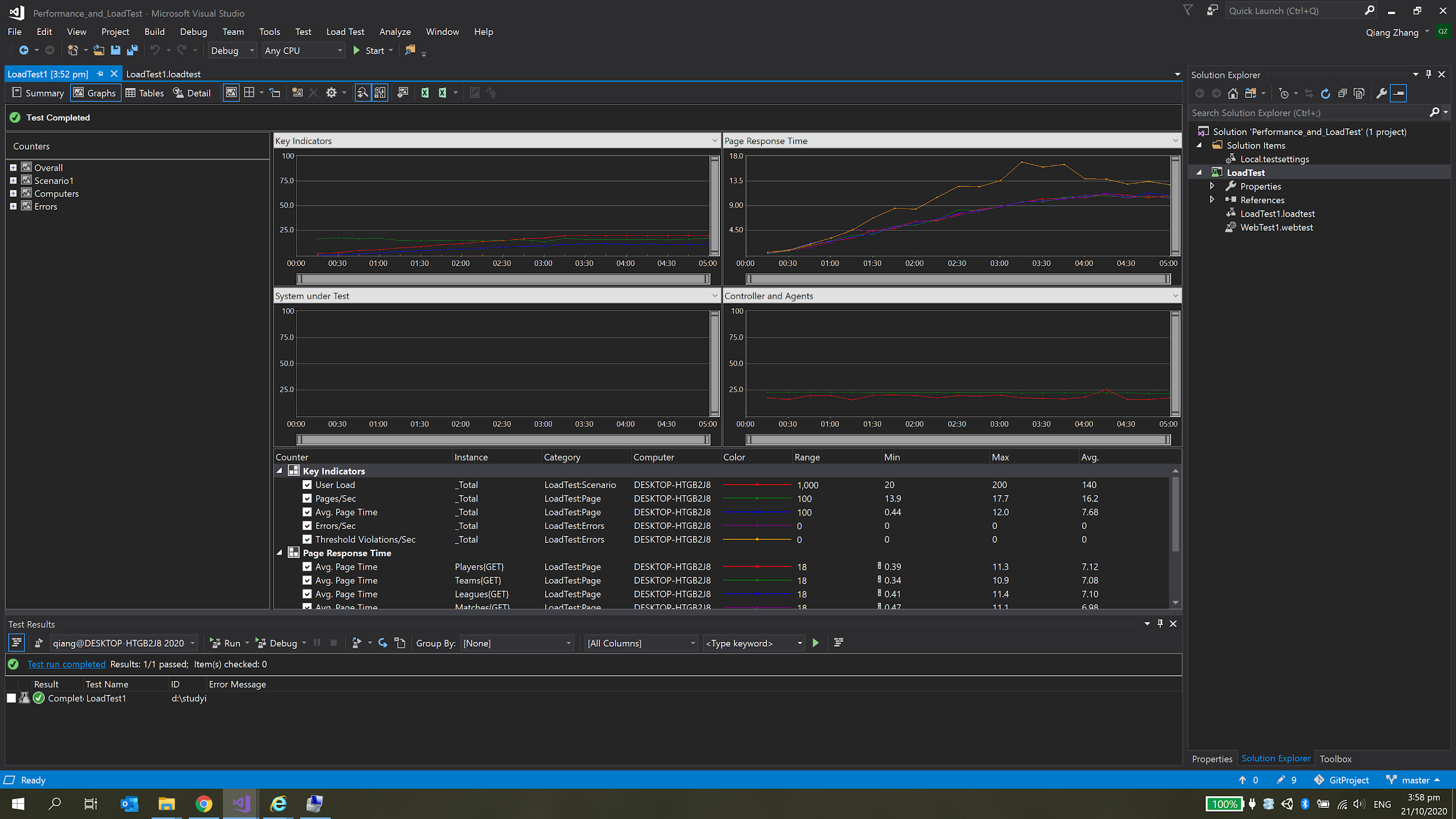
### Load Tests



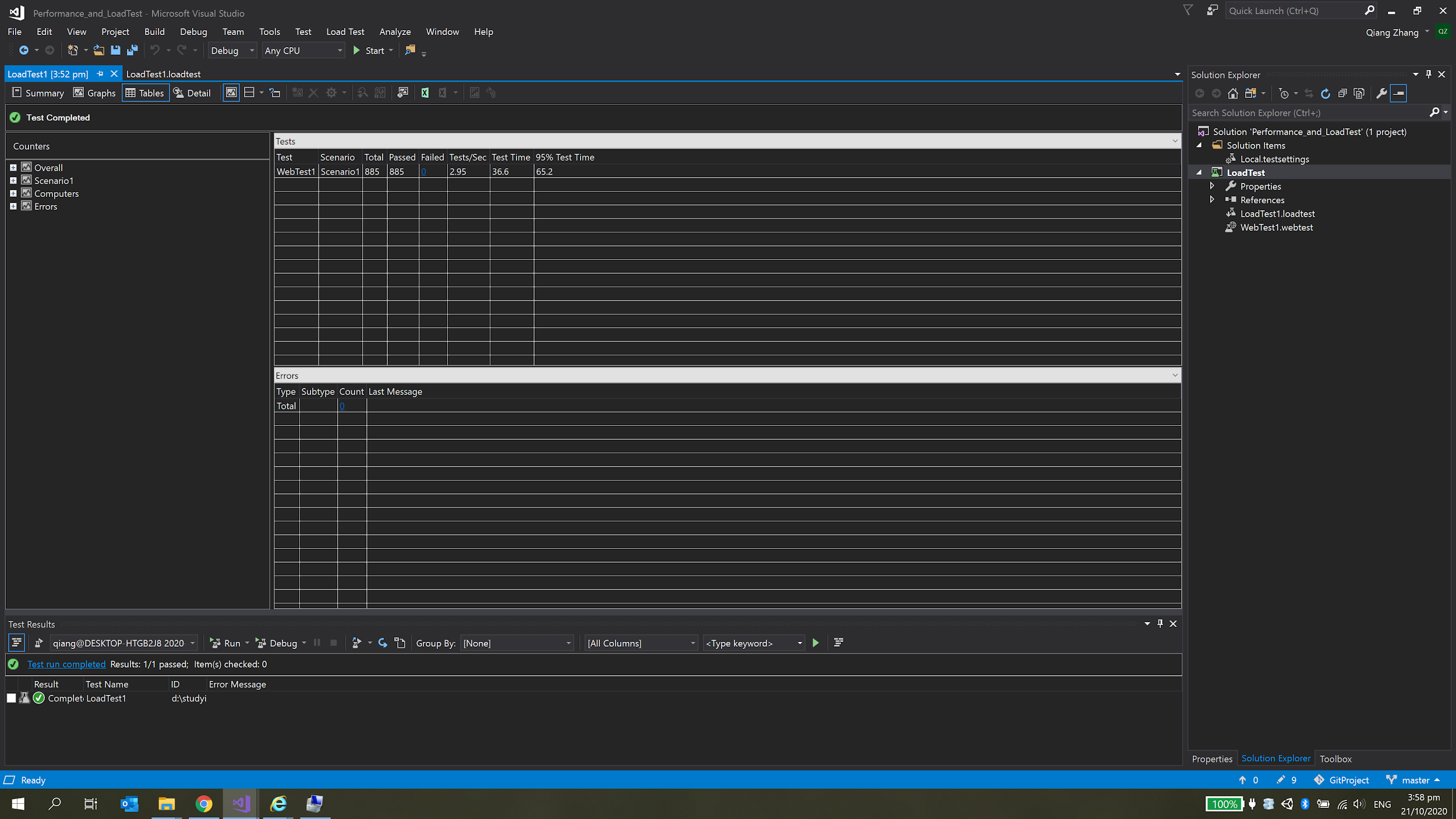
*Figure 2.1 Running Load Test*



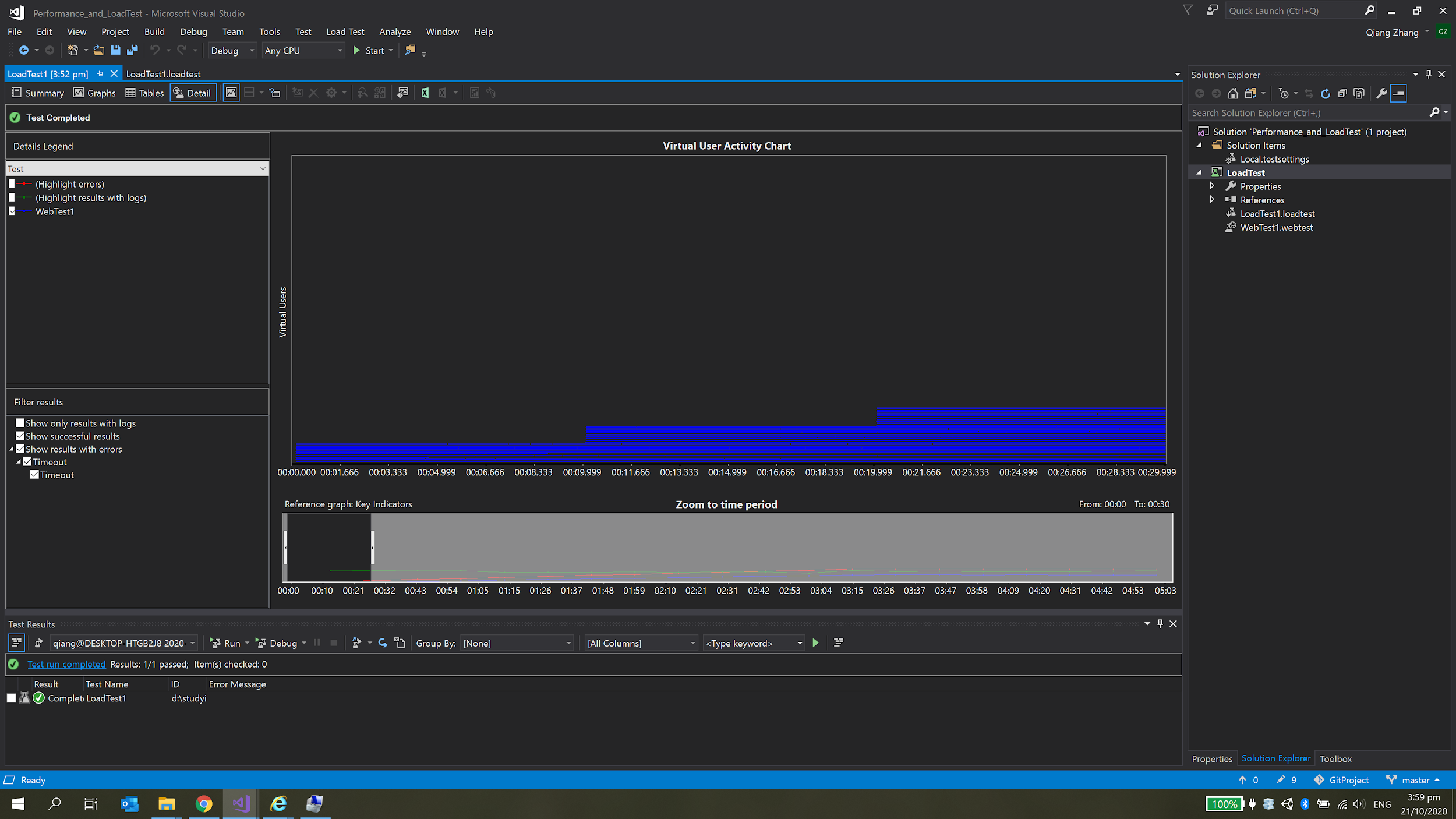
*Figure 2.2 The result of Load Test in Summary*



*Figure 2.3 The result of Load Test in Graphic*



*Figure 2.4 The result of Load Test in Table*



*Figure 2.5 The result of Load Test in Details*

# Static Code Analysis

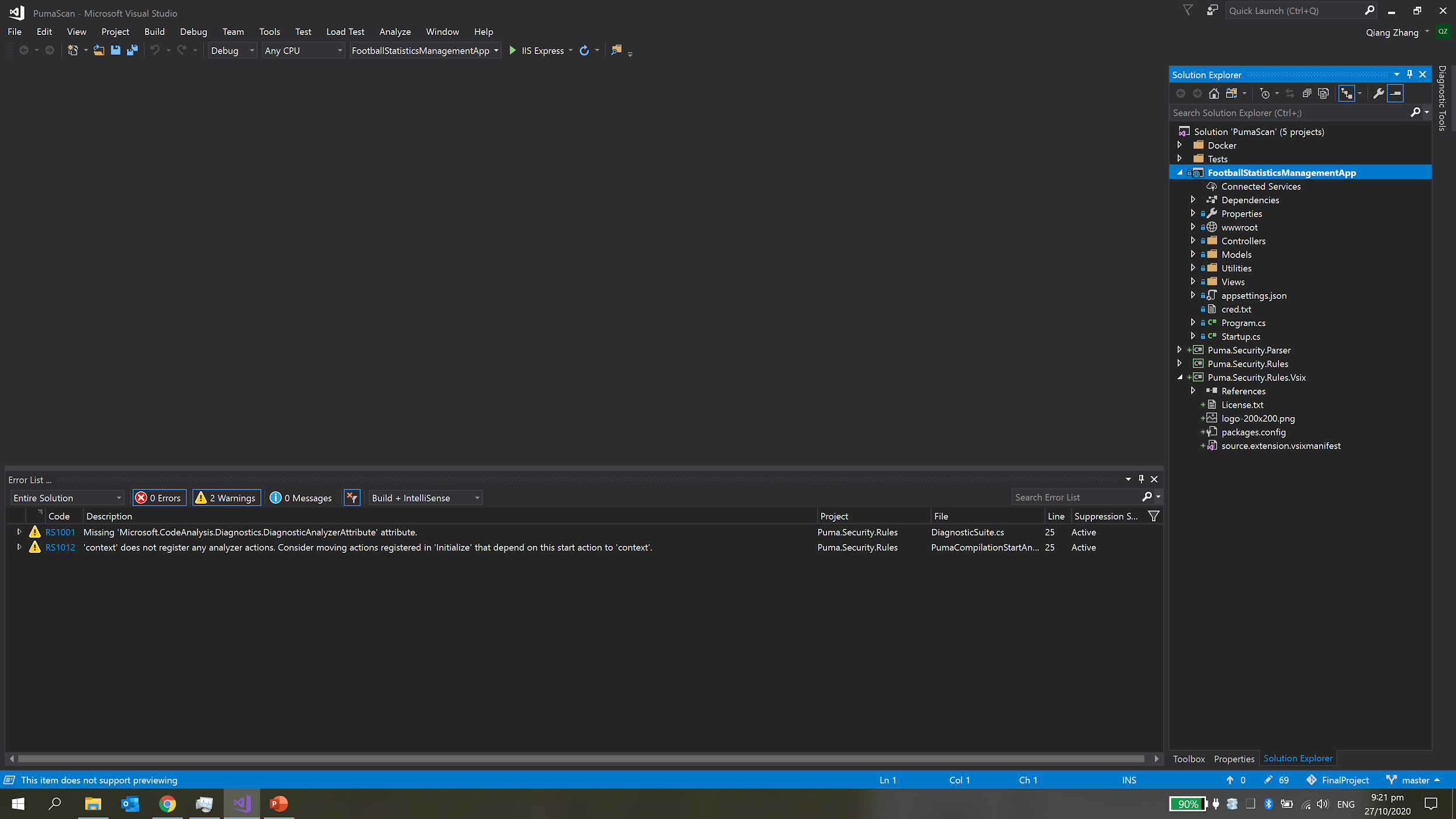
We use Puma Scan to take the static code analysis.

For using this analysis tool, we would be able to avoid vulnerabilities and enhance the security on the application. This is important for security of web applications.

## Risks and Resolutions

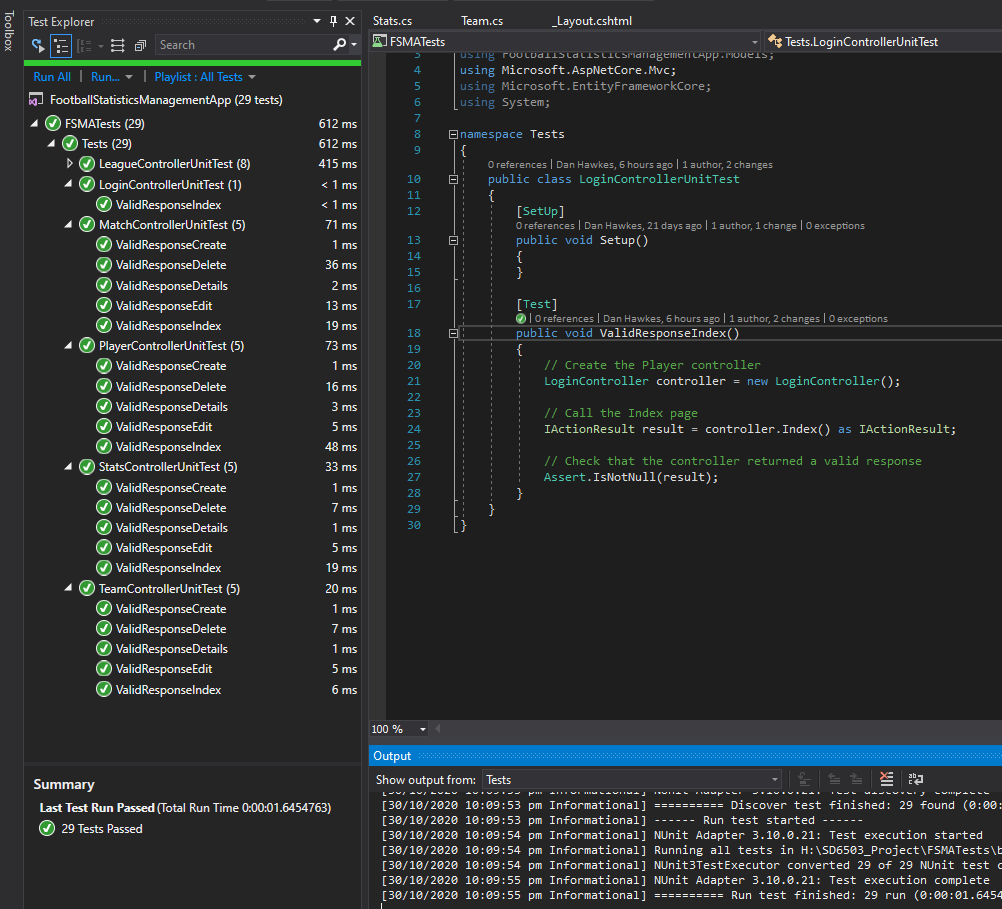
The most important risk for a web application is sensitive data leakage. There are few methods we could prevent this risk. Firstly, we could encrypt information in the database to storage. Such as, we could use a hash algorithm to encrypt player’s personal information. If someone invades our database, he can’t get any information about the player. Secondly, we could avoid displaying sensitive information inadvertently. For example, when we create error messages, we avoid using any useful information on that. As a result, when someone wants to attack the web application, he could not get any useful information on error messages.

## Screenshots



*Figure 3.1 The result of Static Code Analysis*

# Grey Box Tests



*See GreyBoxTesting.xlsx (Inside All Testing folder)*

# White Box Tests

*See WhiteBoxTesting.xlsx*

(Inside all testing folder)

# Black Box Tests

*See BlackBixTesting.xlsx*

*(Inside all testing folder)*