|  |
| --- |
|  |

|  |
| --- |
| **Technical Report** |
| ICTWEB503 – Create Web-based Programs |
|  |
|  |
|  |
| **Sebastian Vowels** |
| **26/10/2021** |
|  |

Table of Contents

[Task 1 - Email 2](#_Toc12456988)

[Task 2 – Report 2](#_Toc12456989)

[Hypertext Transfer Protocol (HTTP) 2](#_Toc12456990)

[Limitations of HTTP 2](#_Toc12456991)

[Advantages of HTTP 2](#_Toc12456992)

[Web Service 2](#_Toc12456993)

[Web Application 2](#_Toc12456994)

[Task 3 – Development 2](#_Toc12456995)

[Task 4 – Perform Test 2](#_Toc12456996)

[Task 5 – Sign-off Sheet 3](#_Toc12456997)

# 

# Task 1 – Email

Graphical user interface, text, application, email

Description automatically generated

# Task 2 – Report

### Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is a communication system utilised widely throughout the world, primarily on the internet. There have been different iterations of the standard of HTTP, namely HTTP 1 and HTTP 2. HTTP 1 was created in 1997 and its creators were unable to recognise the far stretching uses of the internet at the time. As such, standard provided in HTTP 1 was inefficient for what modern internet users required. HTTP 2 was created in 2015 and aimed to rectify a significant amount of these issues.

One of the biggest issues that HTTP 2 rectified related to the prioritisation of the resources being delivered. In HTTP 1, after a server receives a request from a client, they would send one resource at a time and await a response from the client confirming they had received the sent resource. Understandably, this process was highly inefficient as websites grew in size and complexity. When HTTP 1 was created a website consisted of a HTML page. Modern websites require HTML pages, CSS styling, JavaScript code, and multimedia assets galore. By forcing wait times for confirmation messages to be received, HTTP 1 was highly inefficient. HTTP 2 rectified this by allowing web servers to send all resources at once and provide context to the browser as to what should be rendered first as well as if any resources are missing. Should something be missing, the browser is then able to request just what they need. This concept is referred to as multiplexing and it substantially increased the efficiencies of the internet.

In regards to the FF web application, HTTP 2 should be utilised. Whilst in its current state there isn’t a great deal of information being transferred to the browser, HTTP 2 is the international standard and provides FF the opportunity to add functionality later on that might take better use of the improved standard.

### Limitations of HTTP

HTTP isn’t without its issues. Whilst being adopted world wide as the standard for internet communication there are some limitations of the protocol that should be kept in mind:

* Security
  + HTTP is sent across the internet as a plain text message. There is no encryption and anyone in between the client and the server is able to intercept the message.

### Advantages of HTTP

However, taking all of the limitations as previously mentioned into account, there is good reason as to why HTTP has been adopted worldwide:

* Connectionless
  + HTTP operates by sending messages, not having a face to face conversation. As a result, direct connections are not required between the client and server.
* Media Independent
  + Any type of data can be sent through HTTP as long as both the client and server have instructions as to how to unpack it.

### Web Service (API)

### Provide a description of any other applications that the primary web application will connect to or use in the process of normal operation – and whether modifications were required to any external applications / services.

### Web Application (Front End)

* List all the functionality of the web application.  In other words, what are the functions included in your front-end.
* Provide a description of the functionality that needed to be added to the existing web application as per the Client request
  + Provide a description of how this functionality has changed how the web application is used.
* Provide a description of the benefits of the Session management system implemented
* How did you implement sessions management? Describe the methods/tools/frameworks used to allow for use of session variables and the storage of user data
* Screenshot of the code showing session variables

# Task 3 – Development

* Screenshot of the Solution or Project Explorer of your front-end showing all the Folders included in your project – See Task Sheet for what screenshots to take

# Task 4 – Perform Test

Perform testing of the features implemented in the application making sure to test the features in multiple environments and against multiple use-cases – 10+ Testing with some session management

|  |  |  |
| --- | --- | --- |
| **Test Performed** | **Expected Results** | **Actual Results** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Analyse the results and provide a test summary about the errors that occurred during testing and the possible solutions or fixes

# Task 5 – Sign-off Sheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** |  | | |
| **Project Manager:** | Shaun O’Sullivan | | |
| **Start Date:** |  | | |
| **Completion Date:** |  | | |
| **Project Deliverables:** | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| ***I acknowledge that I have submitted all the stated deliverables as per the Project Requirements and your instructions.*** | | | |
|  | |  |  |
| **Programmer’s Name** | | **Signature** | **Date** |
| Shaun O’Sullivan | |  |  |
| **Project Manager** | | **Signature** | **Date** |