

F21SC – Industrial Programming

Coursework 1: Web Browser

Due on Tuesday 25 Oct 2022

Submitted By: Dayanandan Natarajan

School of Mathematical and Computer Sciences

Heriot-Watt University

Table of Contents

| | |
|---|-----------|
| 1. Introduction..... | 3 |
| 2. Requirements Checklist..... | 4 |
| 3. Design Considerations | 5 |
| 4. User Guide..... | 6 |
| 1. Navigating to a web page..... | 6 |
| 2. Add a Home Page | 6 |
| 3. Edit Home Page | 6 |
| 4. Add to Favourites | 6 |
| 5. Navigate to a page from Favourites | 7 |
| 6. Edit Favourites..... | 7 |
| 7. History List..... | 8 |
| 8. Bulk Download | 8 |
| 5. Developer Guide | 9 |
| 6. Testing | 10 |
| 7. Programming Language, Implementation, Conclusion and References. | 12 |
| a) Programming Language and Implementation | 12 |
| b) Conclusion | 12 |
| c) References | 12 |
| 8. Appendix | 13 |
| 9. Student Declaration of Authorship..... | 23 |

1. Introduction

The aim of this coursework is to develop a simple web browser and the objective is to develop proficiency in advanced programming concepts. This coursework is about developing a simple web browser using object oriented and functional programming paradigms.

2. Requirements Checklist

Here are the requirements for the web browser and also the status of the delivery.

| Req # | Req Information | Status |
|-------|---|-----------|
| 1 | User able to send HTTP Request and receive HTTP Response messages for the URL entered. | Delivered |
| 2 | User able to see the success and error status codes and messages upon entering the URL. | Delivered |
| 3 | User able to see the 'Title' of the web page. | Delivered |
| 4 | User able to see the web page contents. | Delivered |
| 5 | User able to see the status code, message, 'Title' and contents upon reload. | Delivered |
| 6 | User able to create a 'Home' page with the desired URL. | Delivered |
| 7 | User able to edit the 'Home' page with the desired URL. | Delivered |
| 8 | The browser should load the 'Home' page upon startup. (Defaulted to university home page) | Delivered |
| 9 | User able to add a web page URL to the Favourites list. | Delivered |
| 10 | User able to associate a name to the URL while adding to Favourites list. | Delivered |
| 11 | User able to edit to modify the name and URL for a web page in Favourites. | Delivered |
| 12 | User able to edit to delete a web page from the Favourites. | Delivered |
| 13 | User able to navigate to a web page by clicking the name from the Favourites list. | Delivered |
| 14 | User able to see the list of web pages in the Favourites list which are added in the previous sessions, upon startup. | Delivered |
| 15 | User able to see the history, the list of URL's navigated. | Delivered |
| 16 | User able to navigate to the previous and next web pages. | Delivered |
| 17 | User able to navigate to the web page by selecting the URL in the history list. | Delivered |
| 18 | User able to see the history from previous sessions upon startup. | Delivered |
| 19 | User able to perform bulk download of list of URL's given in a text file. | Delivered |
| 20 | User able to specify the file name for bulk download. (Defaulted to bulk.txt) | Delivered |
| 21 | User able to see only the return status code, no of bytes received and URL on the contents section of the browser. | Delivered |
| 22 | User should be provided with a GUI to perform the basic web browser functionalities. | Delivered |

3. Design Considerations

The Web Browser developed as part of this coursework is based on the following class diagram (Figure 1). The basic functionality is kept in the main class which also builds the browser form. TextBox, RichTextBox, MenuStrip, Button, StatusStrip and openFileDialog were used in the forms to build the GUI. Some of the key features and complex part are spanned off into separate class.

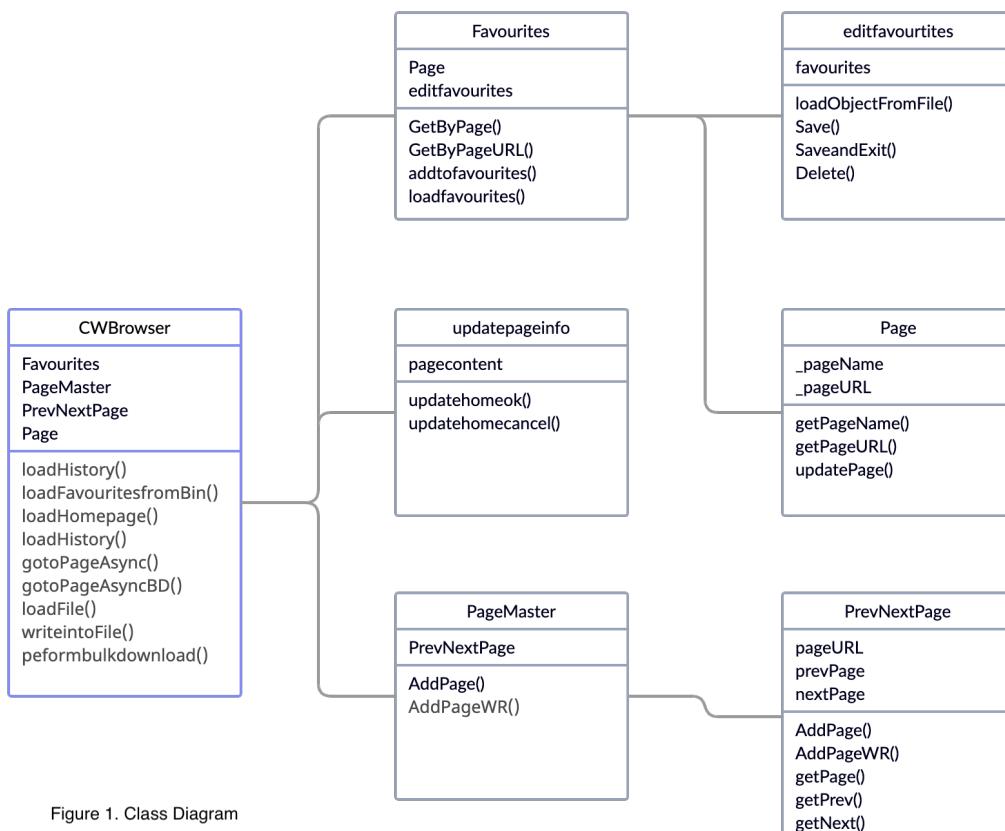


Figure 1. Class Diagram

Some of the key design items,

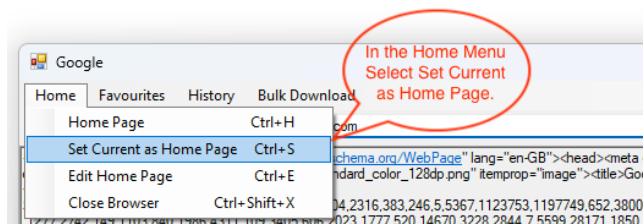
- Built in data structure List is used for the text file operations.
- Custom-built linked list is used to implement the previous and next page operations (Class `PrevNextPage`).
- Serialization is used to store in and retrieve the favourites from the binary files in the form of objects (Class `Favourites`).
- Data binding is used to associate objects stored in binary file to the data grid added to the forms (Class `editFavourites`).
- GUI designed to handle auto rendering whenever there is a resizing.
- Shortcut cuts are included for each of the menu items and there is tool tip on each menu item and button to tell what it can fulfil.
- GUI designed to support navigations without the need for mouse.
- Http request and responses are performed using asynchronous calls to avoid the waiting time until it gets the response.

4. User Guide

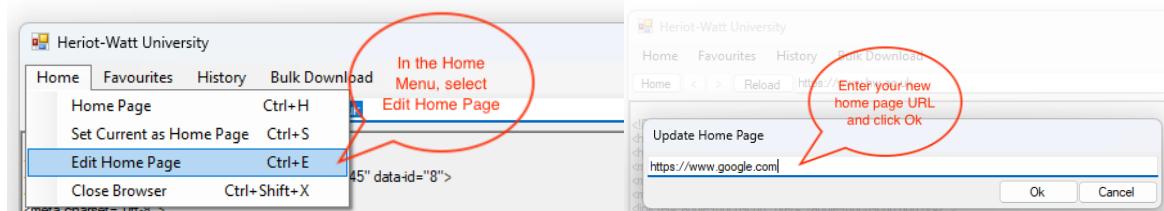
1. Navigating to a web page



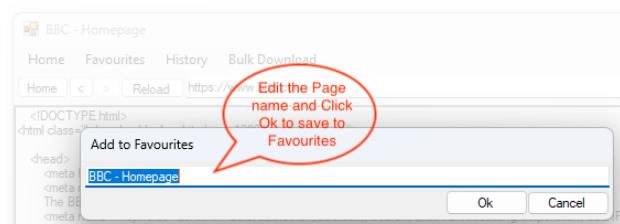
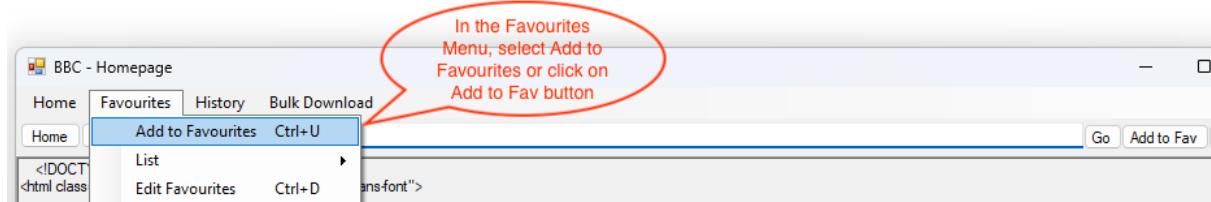
2. Add a Home Page



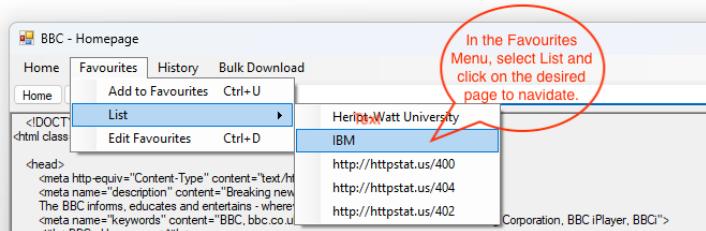
3. Edit Home Page



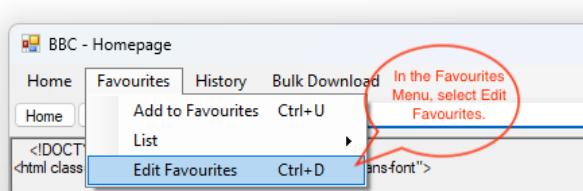
4. Add to Favourites



5. Navigate to a page from Favourites



6. Edit Favourites



Click on the particular page name or URL to edit. Upon completion, click on 'Save' button to save and return to further edit, or click on 'Save and Exit' to save and exit the edit favourites.

| pageName | pageURL |
|---|---|
| Heriot-Watt University UK | https://www.hw.ac.uk |
| IBM | https://www.ibm.com |
| http://httpstat.us/400 | http://httpstat.us/400 |
| http://httpstat.us/404 | http://httpstat.us/404 |
| http://httpstat.us/402 | http://httpstat.us/402 |

Select the row or multiple rows to be deleted, click on the 'Delete Selected' and then 'Save' to commit the delete. Rows will not be deleted if you exit without saving.

| pageName | pageURL |
|---|---|
| Heriot-Watt University | https://www.hw.ac.uk |
| IBM | https://www.ibm.com |
| http://httpstat.us/400 | http://httpstat.us/400 |
| http://httpstat.us/404 | http://httpstat.us/404 |
| http://httpstat.us/402 | http://httpstat.us/402 |

7. History List

The first screenshot shows the History menu open with the 'List' option selected. A red callout bubble points to the 'List' option with the text: 'In the History Menu, select List to see the History list and click on the desired page to navigate.'

The second screenshot shows the History menu open with the 'Clear History' option selected. A red callout bubble points to the 'Clear History' option with the text: 'In the History Menu, select Clear History.'

The third screenshot shows a confirmation dialog box asking 'Are you sure to delete your browsing history?' with 'Yes' and 'No' buttons. A red callout bubble points to the 'Yes' button with the text: 'Click on Yes to confirm to delete history.'

8. Bulk Download

The first screenshot shows the Bulk Download menu open with the 'Download Multiple Pages' option selected. A red callout bubble points to the 'Download Multiple Pages' option with the text: 'In the Bulk Download Menu select on the "Download Multiple Pages" or click on Download button.'

The second screenshot shows a file selection dialog box titled 'Bulk Download - Select a file'. It displays a list of files in a folder structure. A red callout bubble points to the 'bulk' file in the list with the text: 'Select the bulk download file which consists of the list of URL's to be downloaded and click "Open" to proceed.'

5. Developer Guide

The browser is developed as a win form application using the object-oriented concepts and data structures. Key functionalities are moved into separate class from the main browser window handling.

The main class is CWBrowser, which handles the loading of the GUI and events happened and handled over the window. Along with it also handles HTTP connections using inbuilt class HttpClient and text file I/O operations using Streamreader and Streamwriter.

Class Favourites acts a bridge between CWBrowser and Class Page (which handles the use of page name and page URL). Page objects are used to traverse the data between multiple classes to perform data movement between multiple forms. Favourites class methods, addtovavourites() and loadfavourites() gives the capability to store into and retrieve the objects from a binary file using serialization. The class is highly reusable and pluggable where there is a need to store data as objects in binary file. Favourite class can inherit the parent and store the parent contents in binary.

Class editfavourites is used to handle the data movement and binding between the objects and the form. The form uses data grid to visualize and gives the capability to perform maintenance on the screen. Bindingsource is used to bind the objects from binary file to the datagrid which ensured a smooth data management over the form. This class is highly reusable where there is a need for managing data stored as objects and visualized as table structure in the form.

Class PageMaster and PrevNextPage are used to perform the navigation to previous page and next page. Data structure list is used to build this function. Custom built linked list is used to store the objects associated with each other. The AddPage() and getPage() functions help to store and retrieve the page details in the form of objects. getPrev() and getNext() functions help to identify the associated objects linked to each other. The class can be used in any implementation by inheriting them to perform the functions.

Class updatepageinfo is associated with a form to take inputs from user and pass it back to parent form to perform further processing, in this case the form is used to take Home page URL from user to set it as new home page.

The main form also has the capability to take user input by allowing the user to select a text file consisting of list of URLs for bulk download functionality.

6. Testing

Manual testing is performed to test the browser and the following test cases were executed and passed.

| Test case | Test Result | Test Evidence |
|---|---------------------|-----------------|
| To verify the user can send HTTP Request and receive HTTP Response messages for the URL entered. | Passed as expected. | Appendix 1 |
| To verify the user can see the success and error status codes and messages upon entering the URL. | Passed as expected. | Appendix 1 |
| To verify the user can see the 'Title' of the web page. | Passed as expected. | Appendix 1 |
| To verify the user can see the web page contents. | Passed as expected. | Appendix 1 |
| To verify the user can see the status code, message, 'Title' and contents upon reload. | Passed as expected. | Appendix 2 |
| To verify the user can create a 'Home' page with the desired URL. | Passed as expected. | Appendix 3 |
| To verify the user can edit the 'Home' page with the desired URL. | Passed as expected. | Appendix 4, 5,6 |
| To verify the browser loaded with the 'Home' page upon startup. (Defaulted to university home page) | Passed as expected. | Appendix 1 |
| To verify the user can add a web page URL to the Favourites list. | Passed as expected. | Appendix 7 |
| To verify the user can associate a name to the URL while adding to Favourites list. | Passed as expected. | Appendix 8 |
| To verify the user can edit to modify the name and URL for a web page in Favourites. | Passed as expected. | Appendix 9, 10 |
| To verify the user can edit to delete a web page from the Favourites. | Passed as expected. | Appendix 9,11 |
| To verify the user can navigate to a web page by clicking the name from the Favourites list. | Passed as expected. | Appendix 12, 13 |
| To verify the user can see the list of web pages in the Favourites list which are added in the previous sessions, upon startup. | Passed as expected. | Appendix 14 |
| To verify the user can see the history, the list of URLs navigated. | Passed as expected. | Appendix 15, 16 |
| To verify the user can navigate to the previous and next web pages. | Passed as expected. | Appendix 19-21 |
| To verify the user can navigate to the web page by selecting the URL in the history list. | Passed as expected. | Appendix 16 |
| To verify the user can see the history from previous sessions upon startup. | Passed as expected. | Appendix 16 |
| To verify the user can perform bulk download of list of URLs given in a text file. | Passed as expected. | Appendix 22-24 |
| To verify the user can specify the file name for bulk download. (Defaulted to bulk.txt) | Passed as expected. | Appendix 23 |
| To verify the user can see only the return status code, no of bytes received and URL on the contents section of the browser. | Passed as expected. | Appendix 24 |

| | | |
|--|---------------------|-------------|
| To verify the user can use the GUI to perform the basic web browser functionalities. | Passed as expected. | Appendix 25 |
| To verify the user, see appropriate response when entering blank URL. | Passed as expected. | Appendix 26 |
| To verify the user, see appropriate response when entering invalid URL. | Passed as expected. | Appendix 27 |
| To verify the user, see appropriate response when the expected HTTP response code is other than 200. | Passed as expected. | Appendix 28 |
| To verify the user, see appropriate response when the URL times out. (Test failed and the response comes very late and overwriting further navigation) | Failed | |

7. Programming Language, Implementation, Conclusion and References.

a) Programming Language and Implementation

The browser was developed in C# using Visual Studio 2019 as a Win Form application with .Net framework 4.7.2. C# with .Net framework helps to build a robust standalone application which is very portable, and the application can run on any windows machine without a need for any additional files or libraries to be supplied along with. The Common Language Runtime helps to run the C# code much easier and eased the development process.

The object-oriented capability makes it highly efficient and flexible which in turn made it easier for the developers to adopt quickly. The use of Class and objects helped to spin-off the key functionalities outside the core functions and helped to reuse the modules and call them with reference wherever they are needed. There were few challenges in associating the data stored in binary files to the form, which was mitigated by using the data binding and data grid. The I/O operation was seamless with this data binding. The inbuilt library modules made it easier to perform HTTP connections and the existing functions helped to handle the data easily.

b) Conclusion

C# is a better choice of selection when it comes to developing a desktop application over other programming languages, because of its easy to adapt, code and make use of the advanced features such as cross language exception handling, integration, advanced security and deployment.

c) References

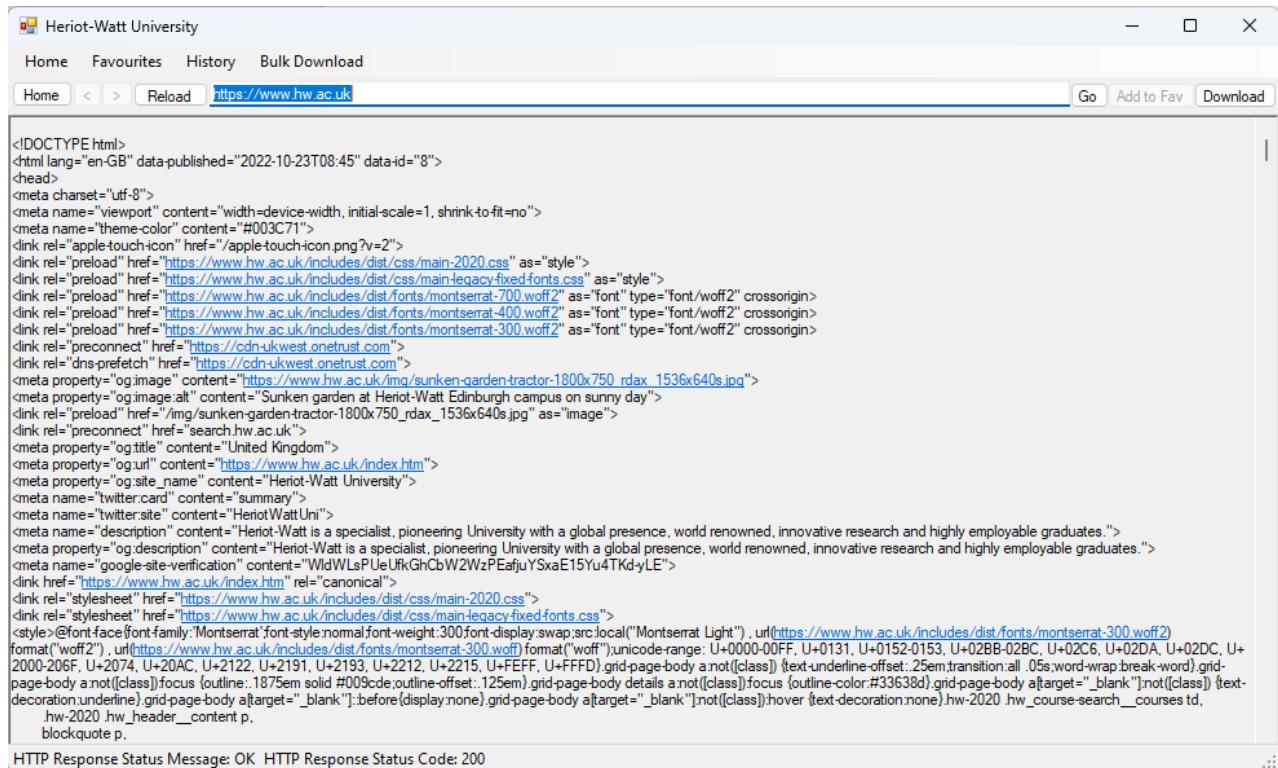
<https://learn.microsoft.com/en-us/dotnet/standard/clr>

<https://www.codeguru.com/csharp/benefits-of-c/>

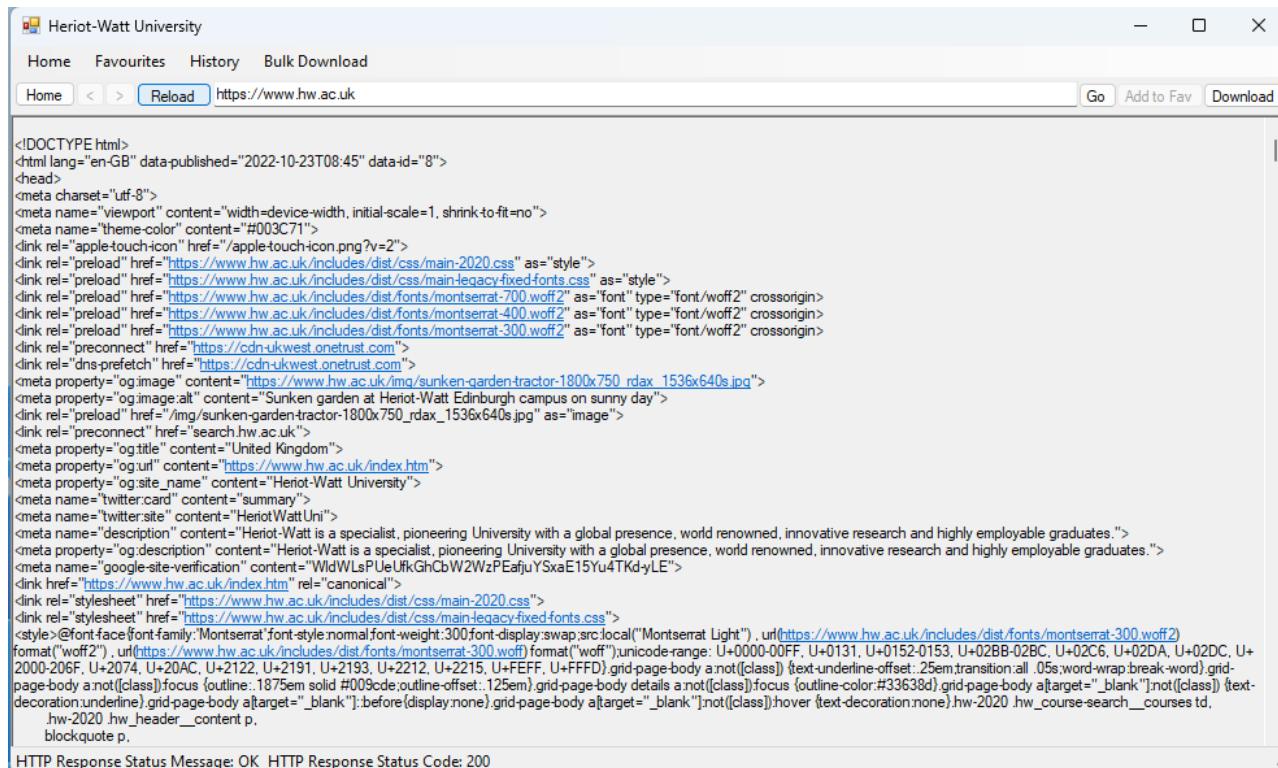
<https://www.bairesdev.com/technologies/csharp/>

8. Appendix

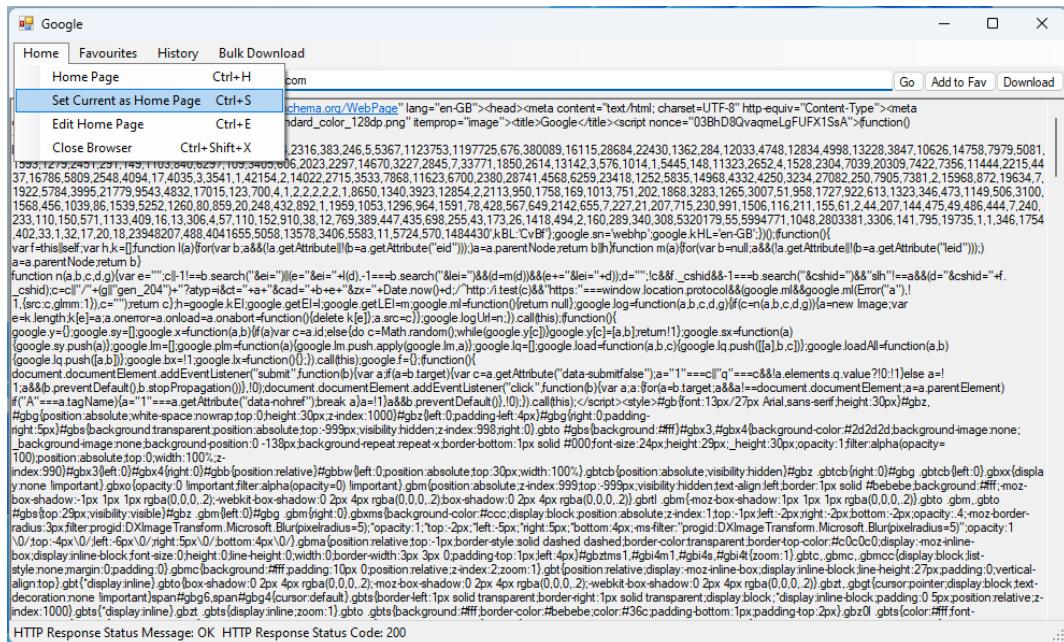
1. Requirements 1_2_3_4_8 Browser Loaded with Home page



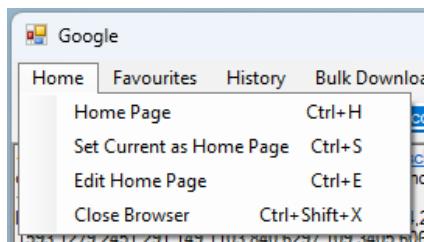
2. Requirement 5 Reload



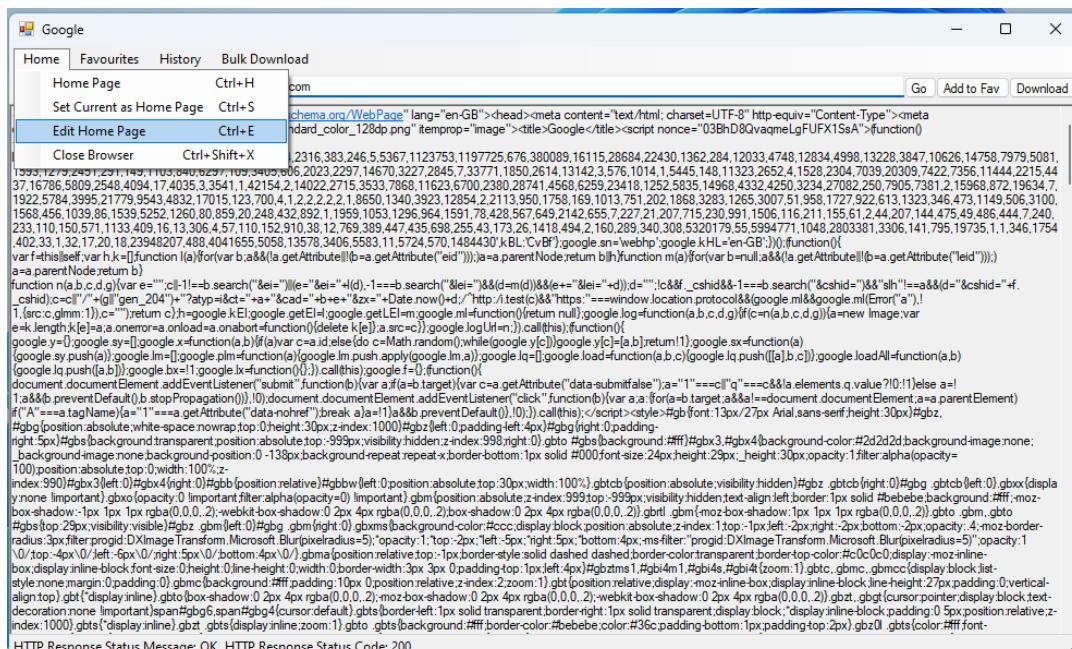
3. Requirement 6 Setting up Home Page



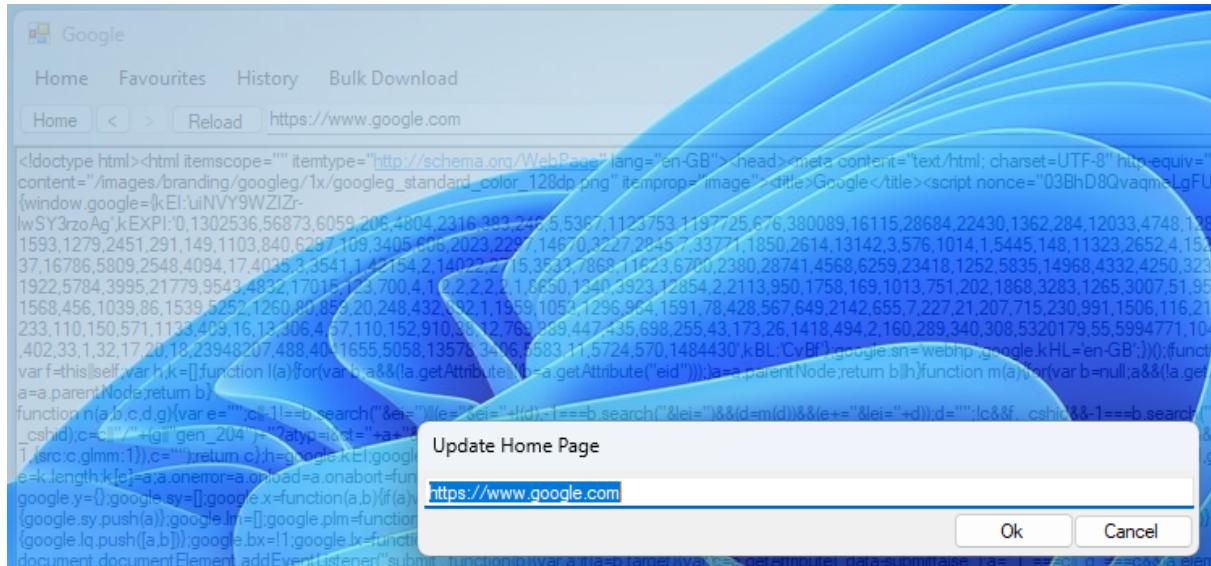
4. Requirement 7 Selecting Home Menu



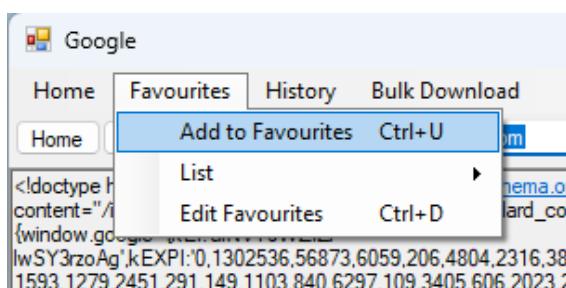
5. Requirement 7 Selecting edit Home Page from Menu



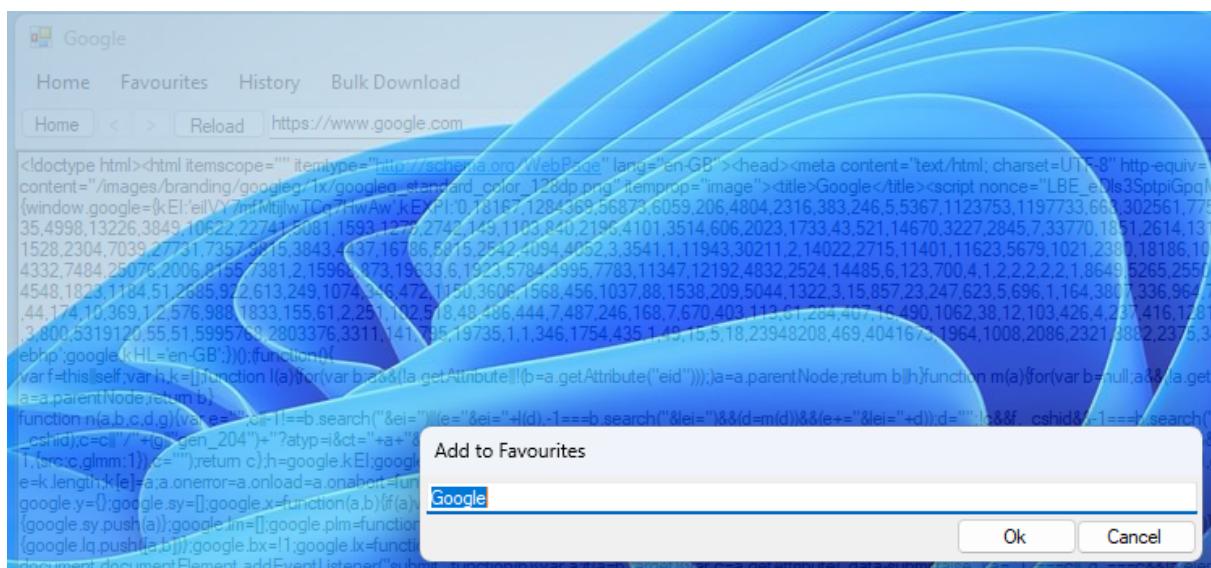
6. Requirement 7 Update Home Page



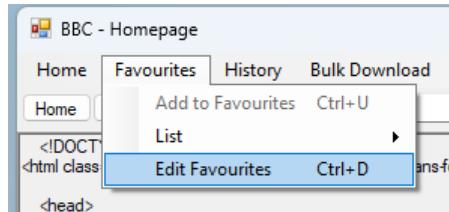
7. Requirement 9 Selecting Add to Favourites from Menu



8. Requirement 10 Associating name while adding to Fav



9. Requirement 11 Selecting Edit Favourites from Favourites Menu



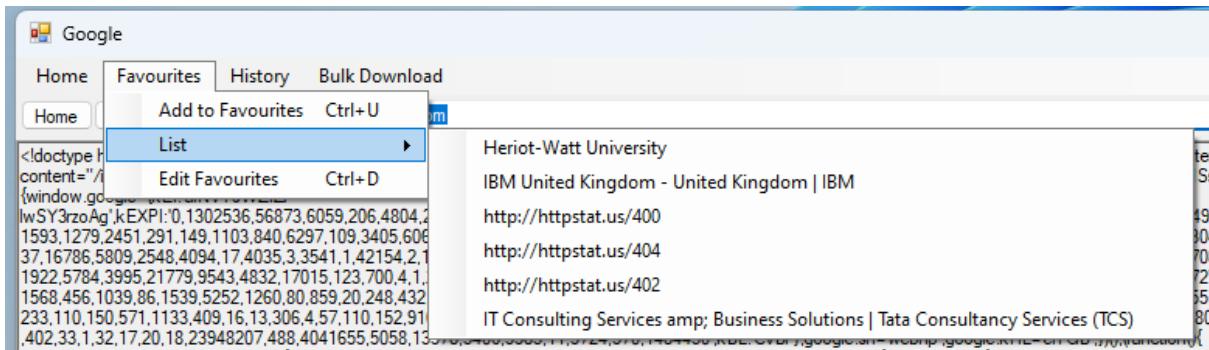
10. Requirement 11 Edit page name in Favourites

A screenshot of a browser window showing the "Edit Favourites" dialog box. The "pageName" column contains "IBM". The "pageURL" column shows the corresponding URLs for each entry. At the bottom of the dialog box are buttons for "Delete Selected", "Save", "Save and Exit", and "Cancel".

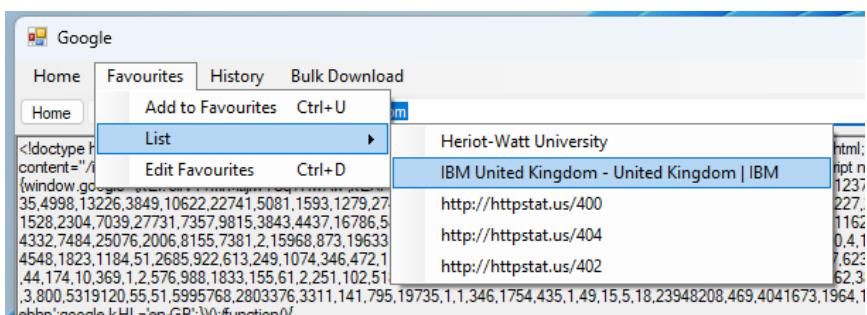
11. Requirement 12 Deleting from Favourites

A screenshot of a browser window showing the "Edit Favourites" dialog box. The entry "IT Consulting Services & Business Solutions | Tat..." is highlighted with a blue selection bar. The "pageName" column contains "IT Consulting Services & Business Solutions | Tat...". The "pageURL" column shows the corresponding URL. At the bottom of the dialog box are buttons for "Delete Selected", "Save", "Save and Exit", and "Cancel".

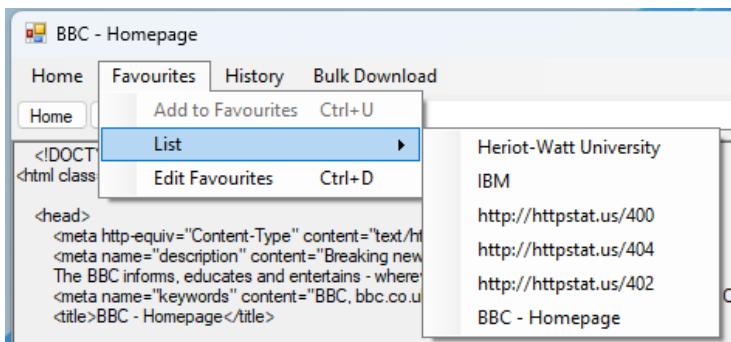
12. Requirement 13 Selecting List from Favourites Menu



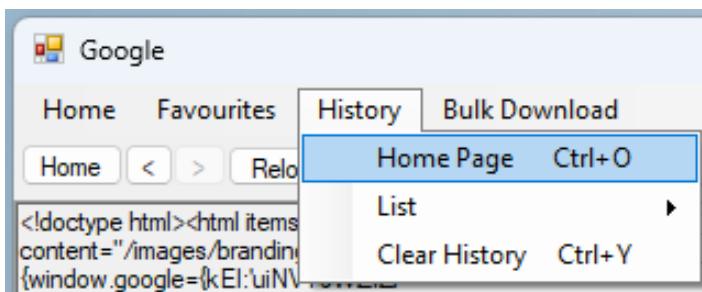
13. Requirement 13 Navigating to a page from Favourites list



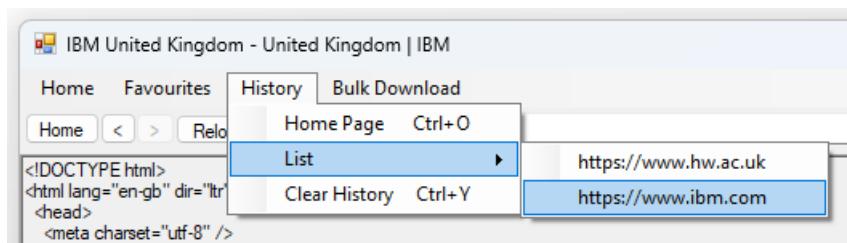
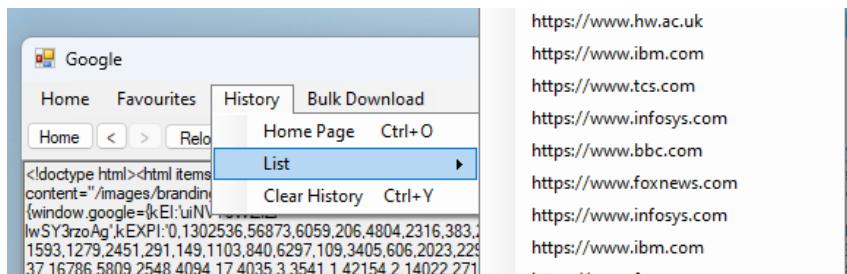
14. Requirement 14 Favourites list from previous session



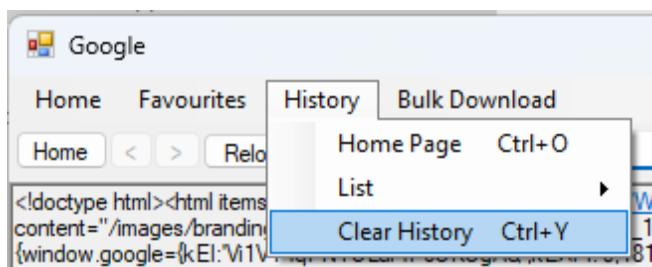
15. Requirement 15 History Menu



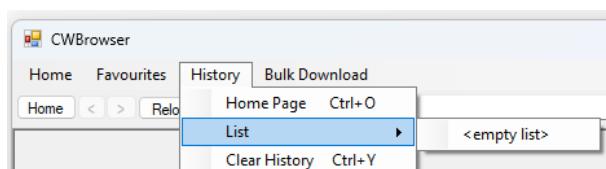
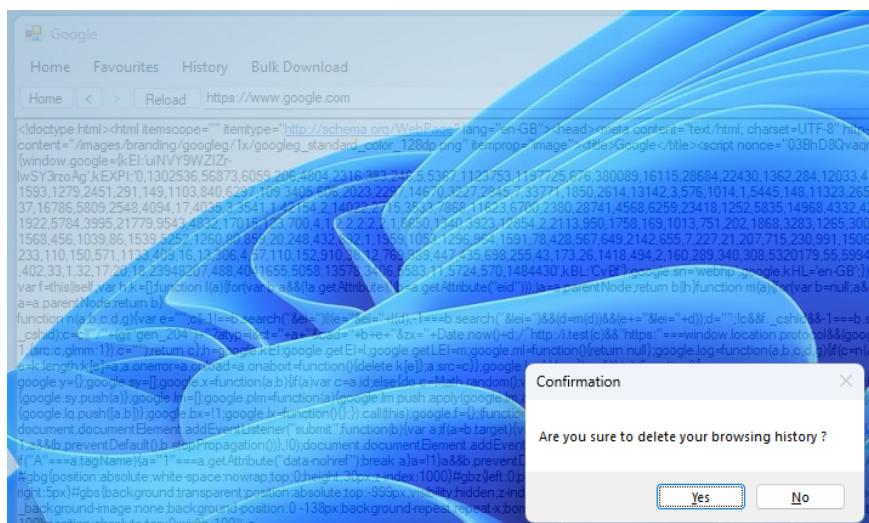
16. Requirement 15_17_18 Selecting from History List



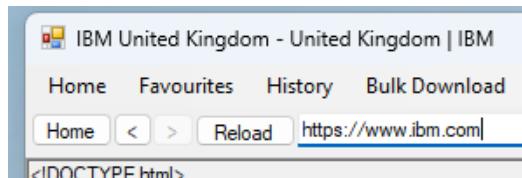
17. Requirement 15a Selecting Clear History from History Menu



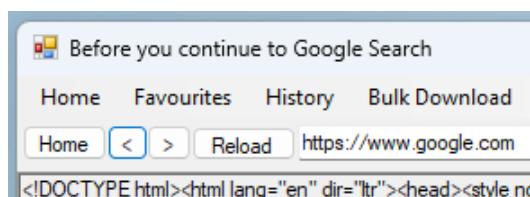
18. Requirement 15a Asking confirmation to clear history



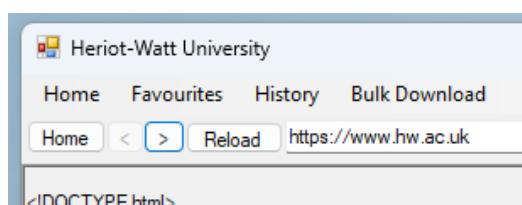
19. Requirement 16 Previous Page with Next disabled



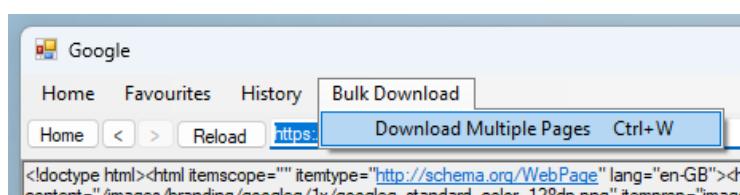
20. Requirement 16 Previous and Next available



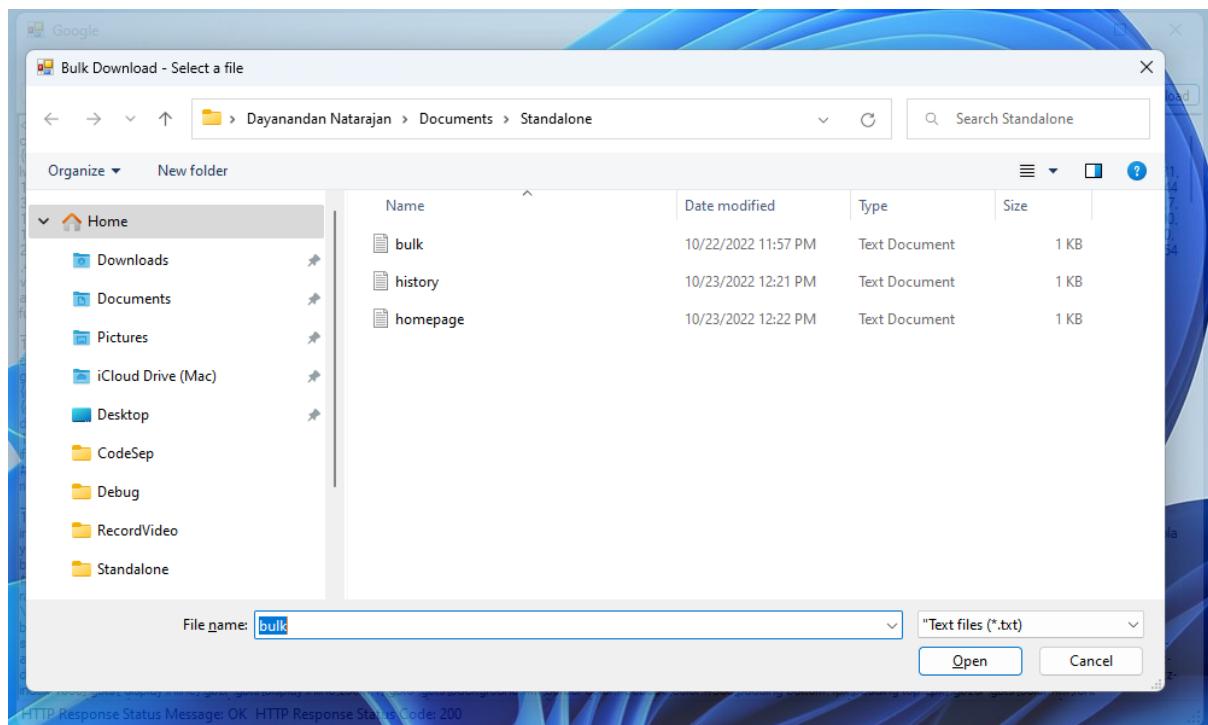
21. Requirement 16 Next Page with Previous disabled



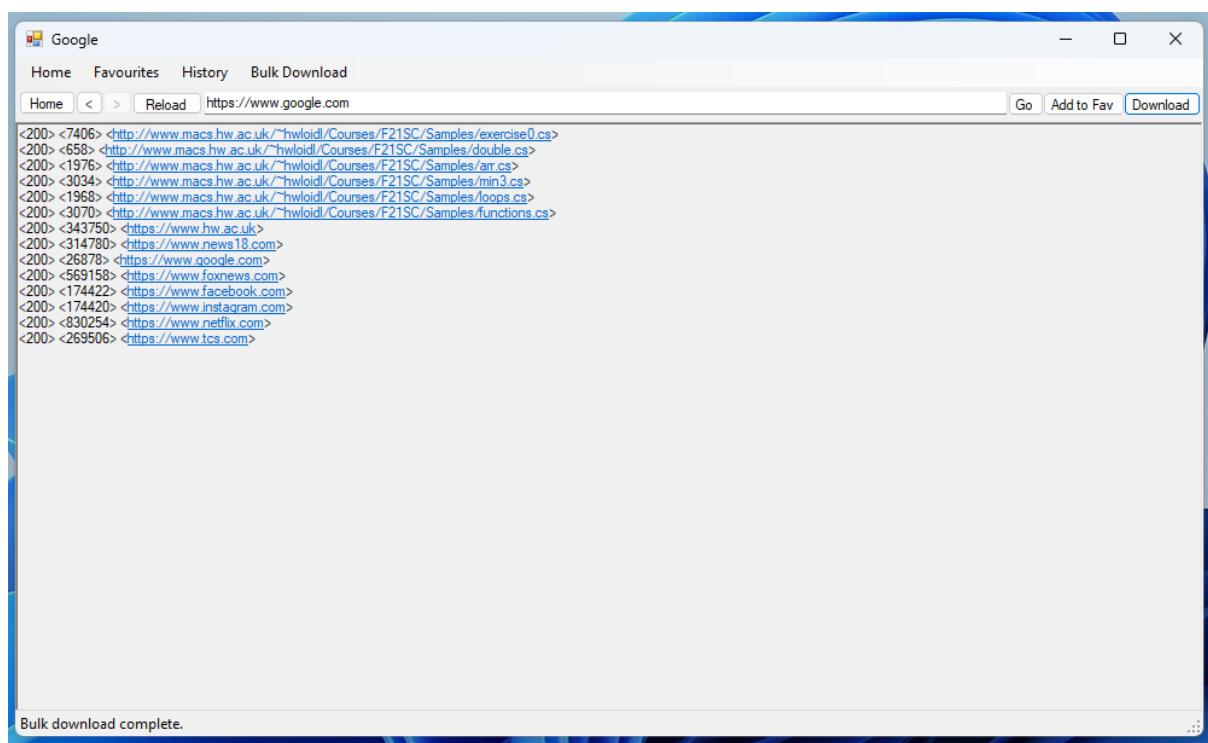
22. Requirement 19 Selecting Bulk Download from Menu



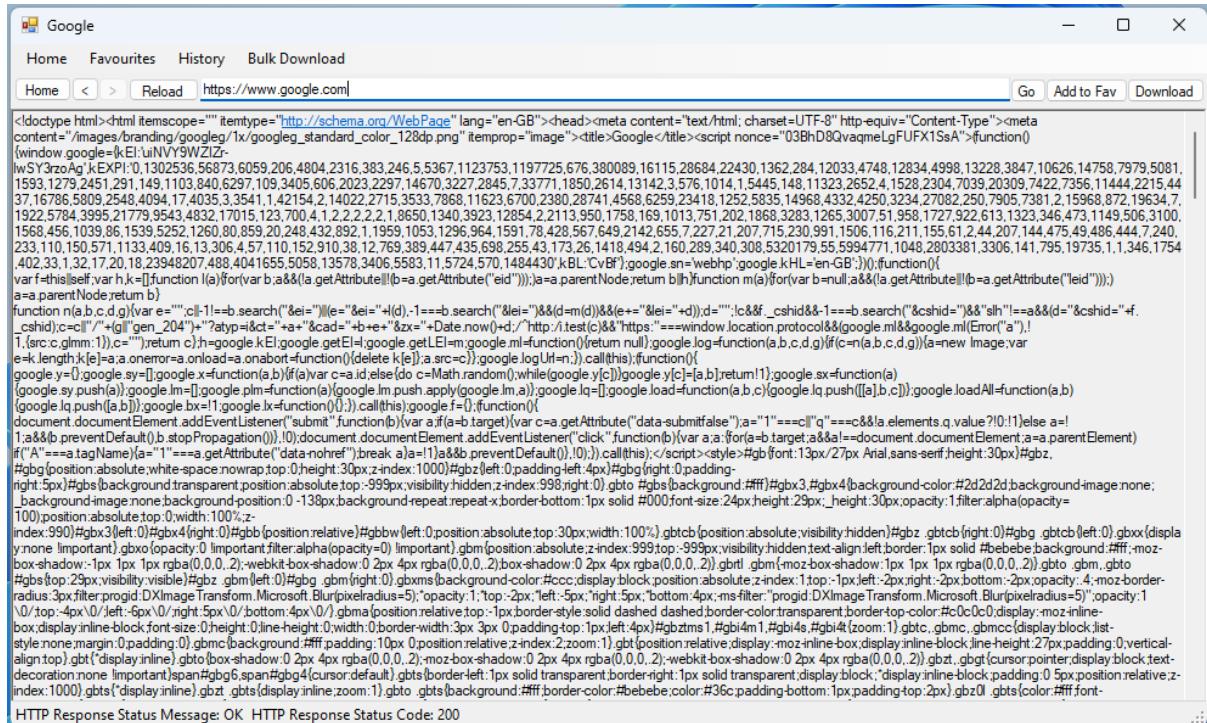
23. Requirement 20 Selecting file for Bulk Download



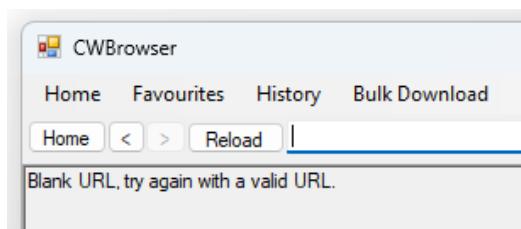
24. Requirement 21 Bulk Download result



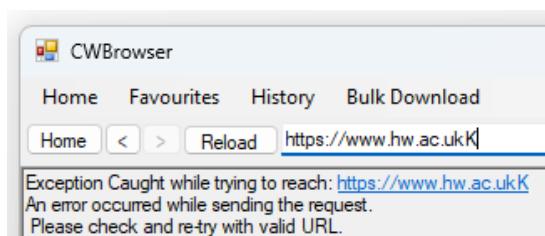
25. Requirement 22 GUI Loaded with different page



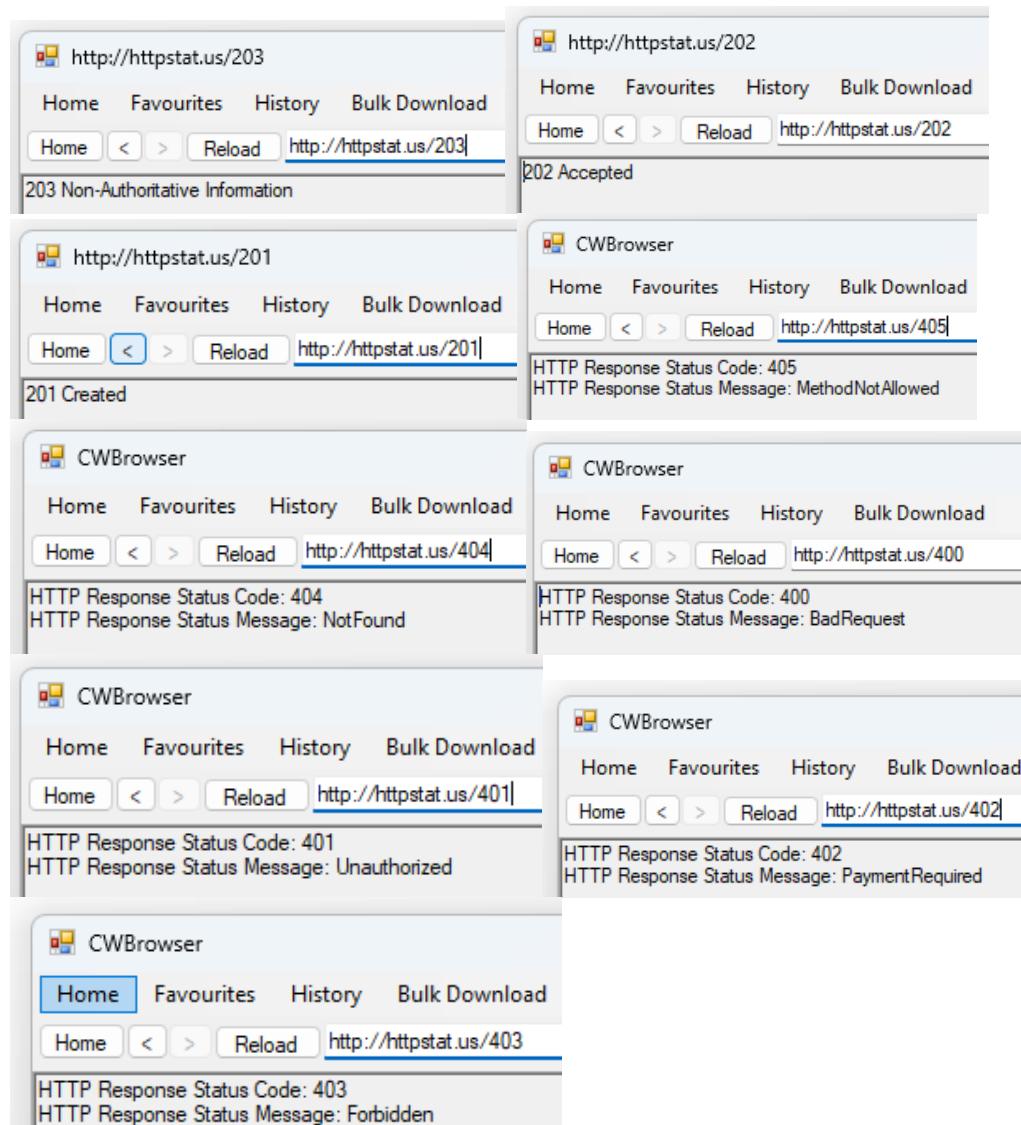
26. Browser response for blank URL



27. Browser response for invalid URL



28. Browser response when status other than 200



9. Student Declaration of Authorship

| | |
|------------------------------|--------------------------------|
| Course code and name: | F21SC – Industrial Programming |
| Type of assessment: | Individual |
| Coursework Title: | Coursework 1: Web Browser |
| Student Name: | Dayanandan Natarajan |
| Student ID Number: | H00393941 |

Declaration of authorship. By signing this form:

- I declare that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the [University's website](#), and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on [Academic Integrity and Plagiarism](#)

Student Signature (type your name): *N. Dayanandan*

Date: 25/10/2022

Copy this page and insert it into your coursework file in front of your title page. For group assessment each group member must sign a separate form and all forms must be included with the group submission.