PL3 assignment 3

Team members :-

1. 2018BTECS00019 Sarthak Patil

2. 2018BTECS00067 Parth Navale

3. 2018BTECS00068 Sahil Reddy

**Q1 & Q3. Header information of the page:**

The HTTP headers identify the specific URL the browser accessed when you opened a web page. Google Chrome has a built-in developer tool that allows you to capture HTTP headers and save them to a file.

To capture headers in chrome:

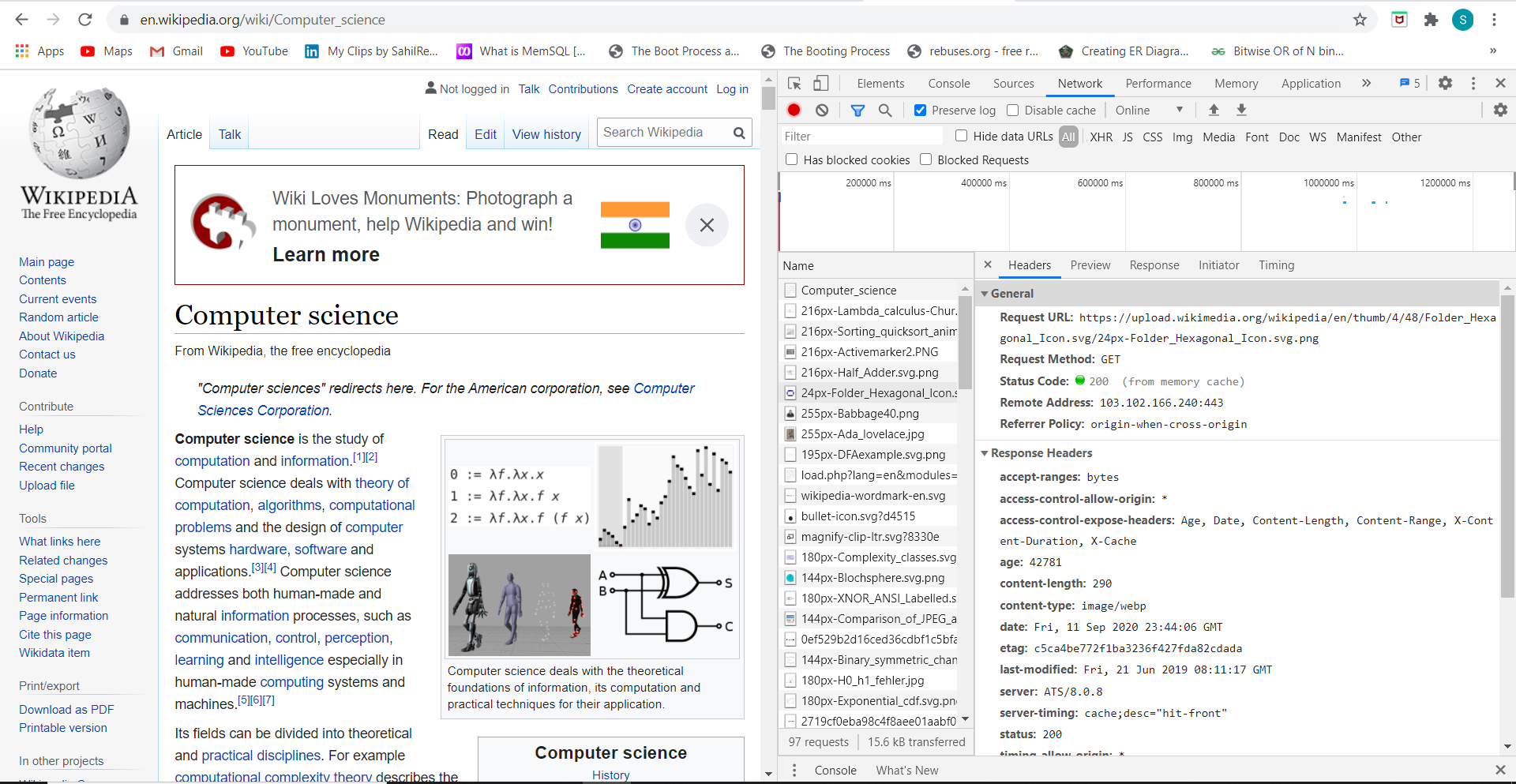
Open the developer tools window by pressing CTRL + SHIFT + i

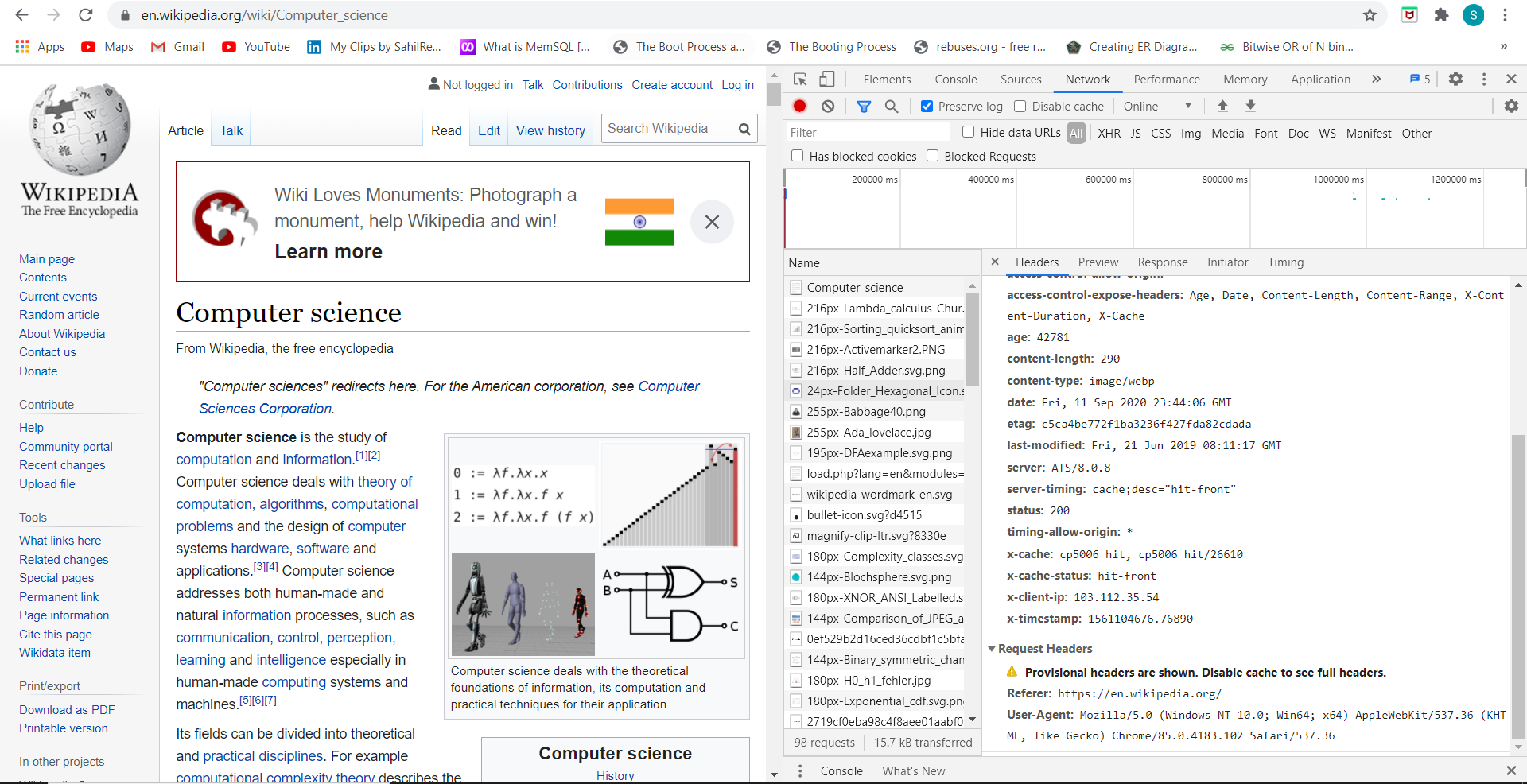
1. Click the Network tab.

2.Make sure the record button is red and the Preserve log option is checked.

3.Once the test is completed, right click on one transaction and select Save all as HAR(http archive format) with content.

**CHROME:**





Data collection starts when you hit **[F5]** or browse to something within the current tab.

Click on the Cookies tab/panel for well formatted Cookie details.

Once you have some output simply click on the name of any object to view the HTTP headers (as well as Request Method, Response Status Code and HTTP version) related to it

Information in **response headers** include:.

* Response status; 200 is a valid response from the server.
* Date of request.
* Server details; type, configuration and version numbers. For example the php version.
* Cookies; cookies set on your system for the domain.
* Last-Modified; this is only available if set on the server and is usually the time the requested file was last modified
* Content-Type; text/html is a html web page, text/xml an xml file

**Request headers** contain more information about the resource to be fetched, or about the client requesting the resource.

**MOZILLA:**

Launch Firefox’s built-in developer tools using [F12] or [Ctrl]+[Shift]+I.

You can save time using [Ctrl]+[Shift]+Q instead; this will take you straight to the Network tool. You can also use [Ctrl]+[ or [Ctrl]+] to move between tool tabs

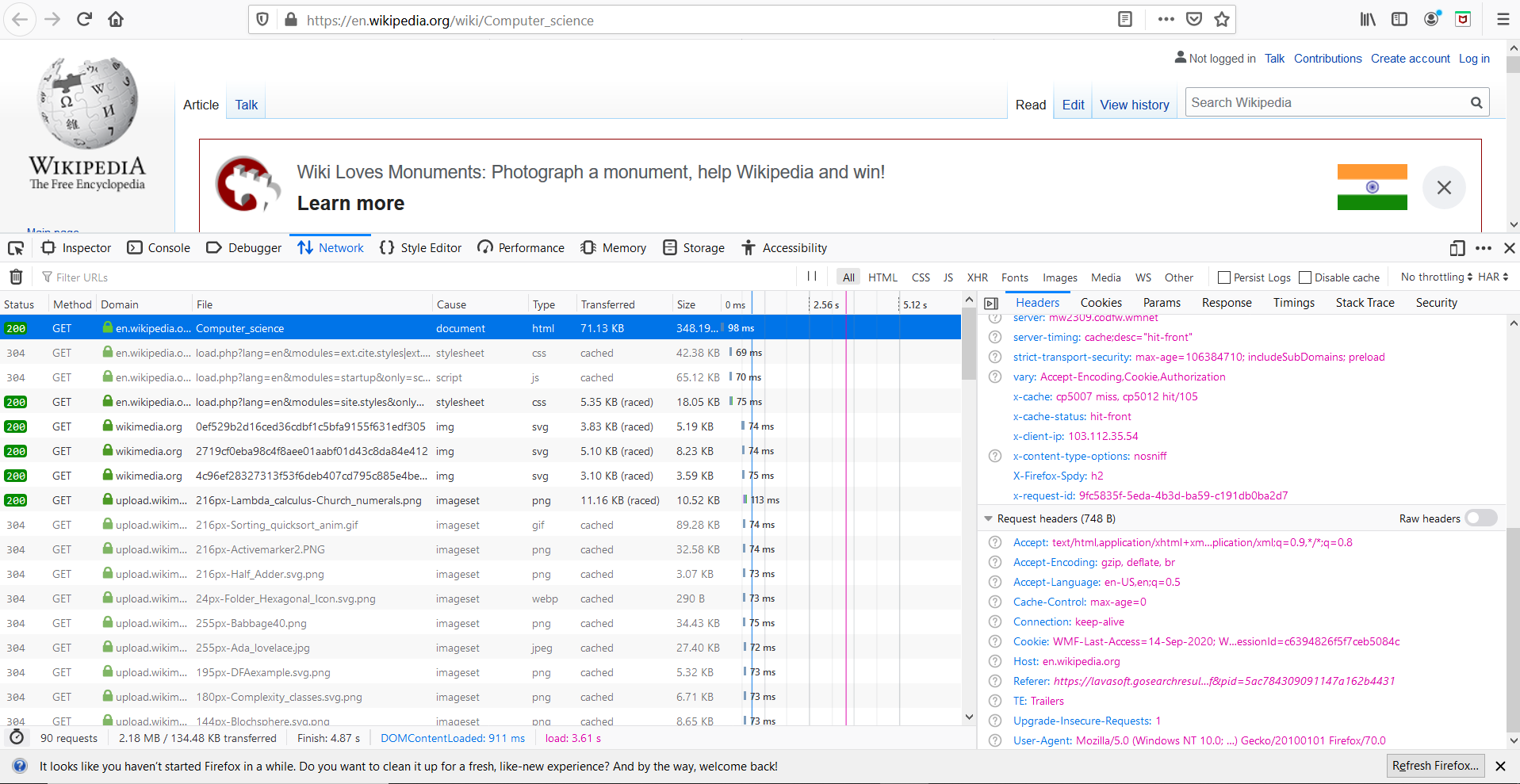
Data collection starts when you hit [F5] or browse to something ki within that tab.

Once you have some output simply click on the name of any object to view the HTTP headers (as well as Request Method and Response Status Code) related to it.

Click on the Cookies tab/panel for well formatted Cookie details.

Browsing to a new page or refreshing the page clears the current data.

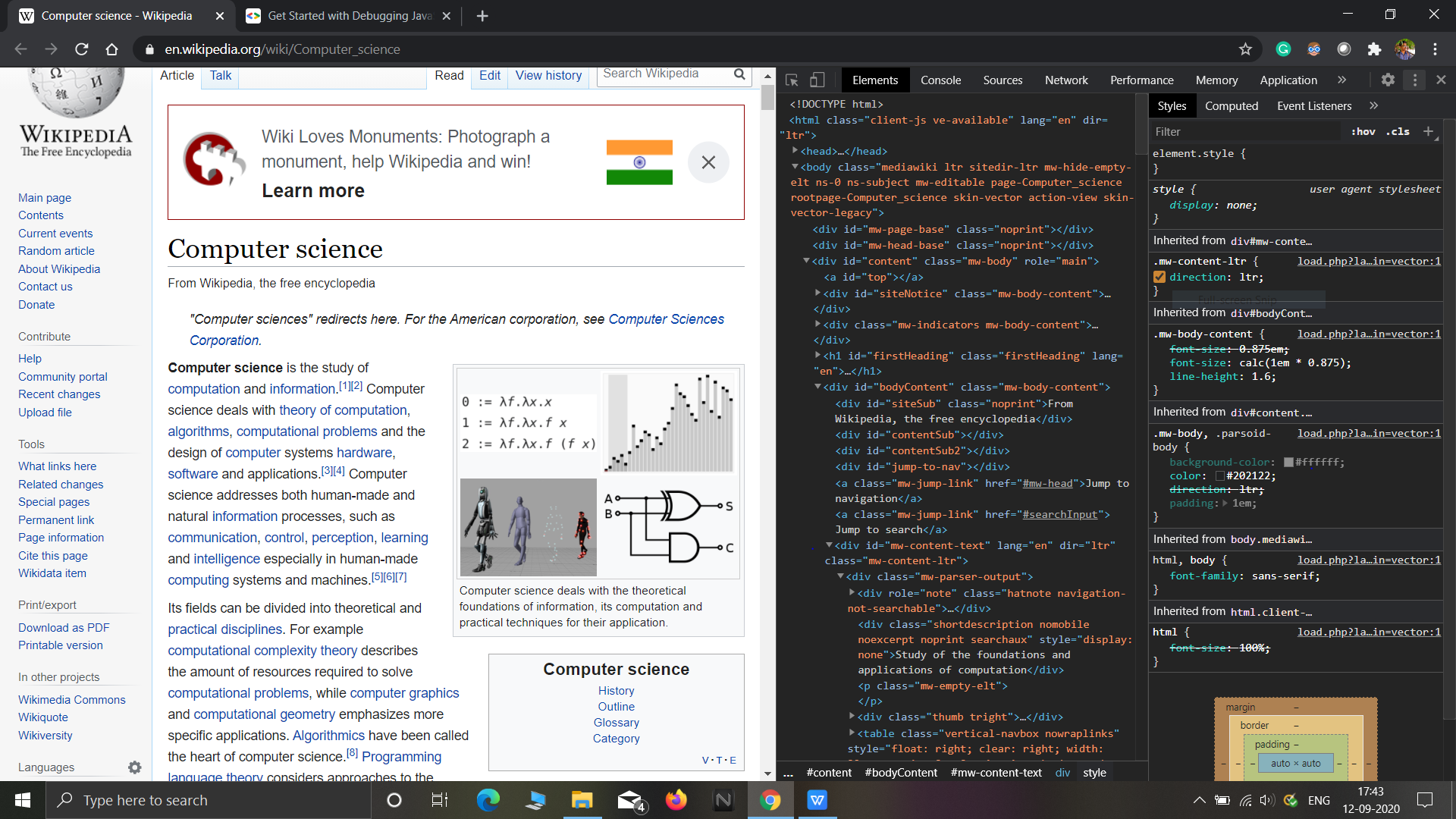
The HTTP version can only be viewed using the Console tab (make sure Net is selected) – click on the object name to view the response version.



**Problem statement 2:**

**4.**

**DOM:-**



* When you're interested in a particular DOM node, **Inspect** is a fast way to open DevTools and investigate that node
* To inspect right click on a particular element
* Click the **Inspect** Inspect icon in the top-left corner of DevTools.
* Once you've selected a node in the DOM Tree, you can navigate the DOM Tree with your keyboard.
* **Search for the nodes:** You can search the DOM Tree by string, CSS selector, or XPath selector.

-Focus your cursor on the **Elements** panel.

-Press Control+F or Command+F (Mac). The Search bar opens at the bottom of the DOM Tree.

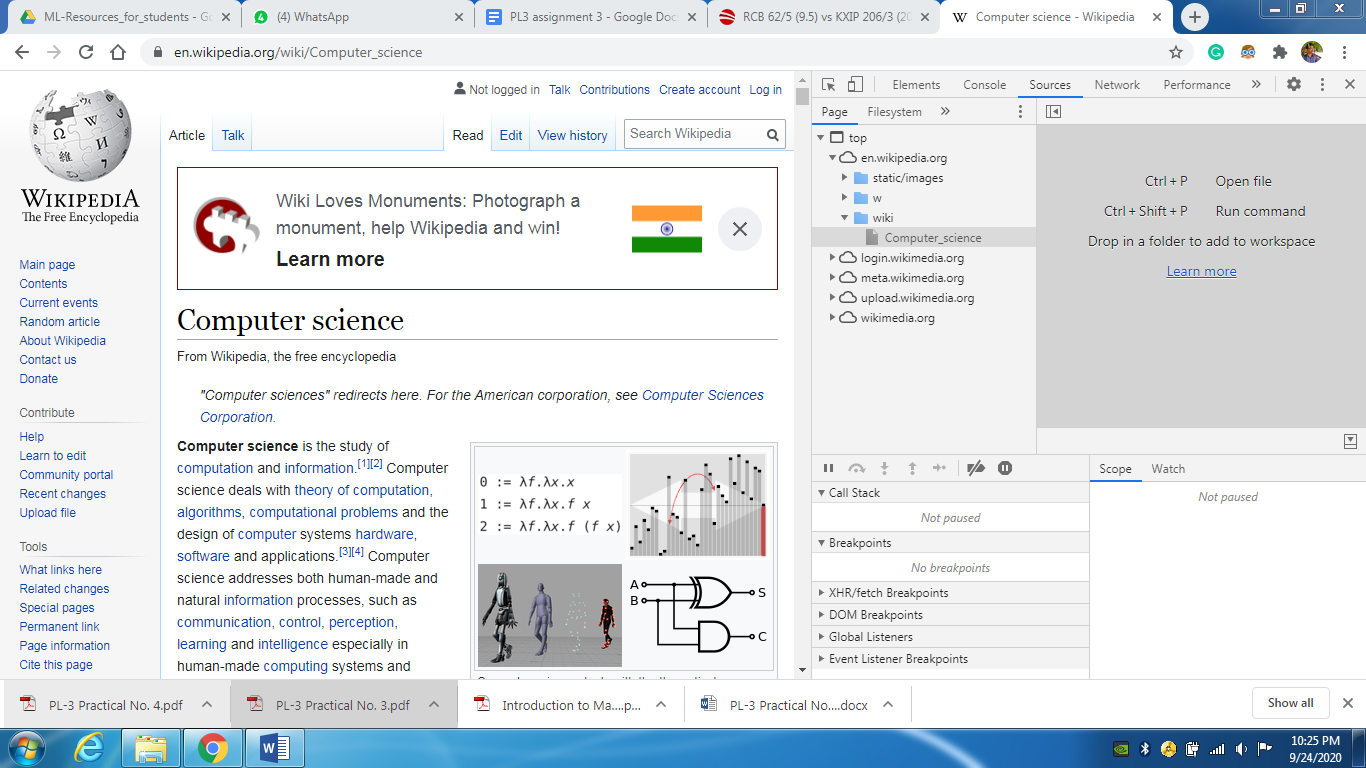
### **Edit content:**

To edit a node's content, double-click the content in the DOM Tree.

**CSS EDITOR:-**

* The Styles tab on the Elements panel lists the CSS rules being applied to whatever element is currently selected in the **DOM Tree**
* Use the Styles tab when you want to change or add CSS declarations to an element.
* You can add class to an element
* Click **.cls**. DevTools reveals a text box where you can add classes to the selected element.
* Type your class in the **Add new class** text box and then press Enter. A check box appears below the **Add new class** text box, where you can toggle the class on and off.
* use the **Styles** tab to permanently apply a CSS pseudostate to an element. DevTools supports :active, :focus, :hover, and :visited.
* Use the **Box Model** interactive diagram in the **Styles** tab to change the width, height, padding, margin, or border length of an element.

**JAVASCRIPT DEBUGGER:-**

****

* The **Sources** panel is where you debug JavaScript.
* Open DevTools by pressing Command+Option+I (Mac) or Control+Shift+I (Windows, Linux). This shortcut opens the **Console** panel.
* Click the **Sources** tab.
* The **Sources** panel UI has 3 parts:

1. The **File Navigator** pane. Every file that the page requests is listed here.
2. The **Code Editor** pane. After selecting a file in the **File Navigator** pane, the contents of that file are displayed here.
3. The **JavaScript Debugging** pane. Various tools for inspecting the page's JavaScript. If your DevTools window is wide, this pane is displayed to the right of the **Code Editor** pane.

* In the **JavaScript Debugging** pane, click **Event Listener Breakpoints** to expand the section. DevTools reveals a list of expandable event categories, such as **Animation** and **Clipboard**.
* Next to the **Mouse** event category, click **Expand** ![Expand
  icon](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA4AAAAQCAIAAACp9tltAAAAWUlEQVR4AZWSMRXAMAgFo7gzBvCACEwgAg8YYG77ytB2u9x8Qy78ZWZVdQLW8eDu3Y3UGxGJCKQOqpqZSB0mAKnfAKK+AUD9B+yr/AE8i38WPwE/LJ8LH+EFt1bfIDGeHDIAAAAASUVORK5CYII=). DevTools reveals a list of mouse events, such as **click** and **mousedown**. Each event has a checkbox next to it.
* One common cause of bugs is when a script executes in the wrong order. Stepping through your code enables you to walk through your code's execution, one line at a time, and figure out exactly where it's executing in a different order than you expected.

**Q2. Request Response Cycle**

## **HOW IT WORKS**

**Request/response cycle**

1. A user opens his browser, types in a URL, and presses Enter.
2. When a user presses Enter, the browser makes a request for that URL.
3. The request hits the Rails router (**config/routes.rb**). The router maps the URL to the correct controller and action to handle the request.
4. The action receives the request and passes it on to the view.
5. The view renders the page as HTML.
6. The controller sends the HTML back to the browser. The page loads and the user sees it.

Request-Response Cycle

The **request**/**response cycle** traces how a user's **request** flows through the app. Understanding the **request**/**response cycle** is helpful to figure out which files to edit when developing an app

Member contributions:

2018BTECS00019:

Using browsers developers tool to go through the DOM, CSS editor, and Javascript debugger options

2018BTECS00068:

Using the browser’s Developer Tools option get the header information of the page.

2018BTECS00067:

Using the browser developer Tool studied the request-response cycle