DEPARTMENT OF COMPUTER SCIENCE

GROUP ID:15 COMPUTER NETWORKS LABORATORY(CS1311)

PROJECT REPORT ON BUILDING A SIMPLE WEB BROWSER

SUBMITTED BY:

PARTHA PRITAM PAUL(14-1-5-059)

DEBASISH DUTTA(14-1-5-067)

RITURAJ DAS(14-1-5-061)

GUIDED BY:

PRASENJIT DAS

(Assistant Professor, CSE Dept.)

CONTENTS

OBJECTIVE OF THE PROJECT:

The main objective of the project is to understand the working of a simple web browser and also the necessary requirements and features that are in a good browser. Also, the various terms related are mentioned and described.

INTRODUCTION:

A web browser (commonly referred to as a browser) is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. An *information resource* is identified by a Uniform Resource Identifier (URI/URL) and may be a web page, image, video or other piece of content. **Hyperlinks** present in resources enable users easily to navigate their browsers to related resources.

Although browsers are primarily intended to use the **World Wide Web**, they can also be used to access information provided by web servers in private networks or files in file systems.

FUNCTIONS OF A WEB BROWSER:

The primary purpose of a web browser is to bring information resources to the user ("retrieval" or "fetching"), allowing them to view the information ("display", "rendering"), and then access other information ("navigation", "following links").

This process begins when the user inputs a Uniform Resource Locator (URL), for example http://en.wikipedia.org/, into the browser. The prefix of the URL, the **Uniform Resource Identifier** or URI, determines how the URL will be interpreted. The most commonly used kind of URI starts with http: and identifies a resource to be retrieved over the **Hypertext Transfer Protocol (HTTP)**. Many browsers also support a variety of other prefixes, such as https: for HTTPS, ftp: for the File Transfer Protocol, and file: for local files. Prefixes that the web browser cannot directly handle are often handed off to another application entirely. For example, mailto: URIs are usually passed to the user's default e-mail application, and news: URIs are passed to the user's default newsgroup reader.

In the case of http, https, file, and others, once the resource has been retrieved the web browser will display it. HTML and associated content (image files, formatting information such as CSS, etc.) is passed to the browser's layout engine to be transformed from mark up to an interactive document, a process known as "rendering". Aside from HTML, web browsers can generally display any kind of content that can be part of a web page. Most browsers can display images, audio, video, and XML files, and often have plug-ins to support Flash applications and Java applets. Upon encountering a file of an unsupported type or a file that is set up to be downloaded rather than displayed, the browser prompts the user to save the file to disk.

Information resources may contain hyperlinks to other information resources. Each link contains the URI of a resource to go to. When a link is clicked, the browser navigates to the resource indicated by the link's target URI, and the process of bringing content to the user begins again.

FEATURES:

User Interface

Most major web browsers have these user interface elements in common:

- *Back* and *forward* buttons to go back to the previous resource and forward respectively.
- A *refresh* or *reload* button to reload the current resource.
- A *stop* button to cancel loading the resource. In some browsers, the stop button is merged with the reload button.
- A *home* button to return to the user's home page.
- An address bar to input the Uniform Resource Identifier (URI) of the desired resource and display it.

- A search bar to input terms into a web search engine. In some browsers, the search bar is merged with the address bar.
- A status bar to display progress in loading the resource and also the URI of links when the cursor hovers over them, and page zooming capability.
- The *viewport*, the visible area of the webpage within the browser window.
- The ability to view the HTML source for a page.

Major browsers also possess incremental find features to search within a web page.

Privacy and Security

Most browsers support HTTP Secure and offer quick and easy ways to delete the web cache, download history, form and search history, cookies, and browsing history. For a comparison of the current security vulnerabilities of browsers, see comparison of web browsers.

Standards Support

Early web browsers supported only a very simple version of HTML. The rapid development of proprietary web browsers led to the development of non-standard dialects of HTML, leading to problems with interoperability. Modern web browsers support a combination of standards-based and *de facto* HTML and XHTML, which should be rendered in the same way by all browsers.

Extensibility

A browser extension is a computer program that extends the functionality of a web browser. Every major web browser supports the development of browser extensions.

SOME ELEMENTS REALATED TO WEB BROWSER:

(I) URI (Uniform Resource Identifier)

In information technology, a **Uniform Resource Identifier** (**URI**) is a string of characters used to identify a resource. Such identification enables interaction with representations of the resource over a network, typically the World Wide Web, using specific protocols. Schemes specifying a concrete syntax and associated protocols define each URI. The most common form of URI is the Uniform Resource Locator (URL), frequently referred to informally as a *web address.* More rarely seen in usage is the Uniform Resource Name (URN), which was designed to complement URLs by providing a mechanism for the identification of resources in particular namespaces.

(II)Hyperlink

In computing, a **hyperlink**, or simply a **link**, is a reference to data that the reader can directly follow either by clicking, tapping, or hovering. A hyperlink points to a whole document or to a specific element within a document. Hypertext is text with hyperlinks. The text that is linked is called anchor text. A software system that is

used for viewing and creating hypertext is a *hypertext system*, and to create a hyperlink is *to hyperlink* (or simply *to link*). A user following hyperlinks is said to *navigate* or *browse* the hypertext.

(III). HTTP (Hypertext Transfer Protocol):

The Hypertext Transfer Protocol (**HTTP**) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.

Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

HTTP functions as a **request–response protocol** in the **client–server** computing model. A web browser, for example, may be the *client* and an application running on a computer hosting a website may be the *server*. The client submits an HTTP *request* message to the server. The server, which provides *resources* such as HTML files and other content, or performs other functions on behalf of the client, returns a *response* message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

(IV).URL (Uniform Resource Locator):

A Uniform Resource Locator (**URL**), commonly informally termed a web address (a term which is not defined identically) is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of **Uniform Resource Identifier (URI),** although many people use the two terms interchangeably. A URL implies the means to access an indicated resource, which is not true of every URI. URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications.

Most web browsers display the URL of a web page above the page in an address bar. A typical URL could have the form `http://www.example.com/index.html`, which indicates a protocol (`http`), a hostname (`www.example.com`), and a file name (`index.html`).

3.BUILDING A BASIC WEB BROWSER:

Tools Used: NetBeans IDE

Language Used: Java with JFrame

A basic web browser is built in java using JFrame to design the user interface.

The Program is divided into some subparts which are explained in detail.

It contains an address bar where a user can type the address and a go button which when clicked loads the web page. For displaying the web pages, we require JEditor Pane and Hyperlink Listener is required to respond when the user clicks on a link in the document. Following are the steps for creating the web browser.

Step 1: Setting up the constructor of the class. In the constructor, we first create the address bar for taking URLs and also set up browser to use a proxy server.

```
System.getProperties().put("http.proxyHost", "172.16.30.20")

System.getProperties().put("http.proxyPort", "8080");

System.getProperties().put("http.proxyUser", "NITS");

System.getProperties().put("http.proxyPassword", "abcde");

addressBar= new JTextField("http://");
```

Step 2: Implementing Hyperlink Listener

Hyperlink Listener contains a single function as **hyperlinkUpdate()** which is defined as follows. If hyperlink is clicked then URL of that hyperlink is loaded in the edit pane by calling **loadURL()** method.

Step 3: Creating the loadURL() method.

The **loadURL()** method takes a string as an input and takes that input to display the webpage denoted by the string. The string here in generated when the user inputs data into the addressbar or clicks on a hyperlink which are converted into strings and sent to the method. It uses the **setpage()** method to display the webpage.

Step 4: Action Listener for Enter.

Basically, when enter is pressed it takes the URL in the address bar and converts into string and passes it to the **loadURL()** method for it to display the web page.

Step 5: Creating the Main Method.

These are the step that are required for us to build this simple web browser that takes an url as input and displays the web page.

4.REFERENCES:

- Wikipedia.org
- Computer Networks by B.A. Forouzan
- sourcecodeworld.com
- Youtube.com

5.CONCLUSION:

A basic web browser with minimal functionality was created where we could demonstrate the basic guidelines for building a full-fledged web browser. The basic features were implemented and gave us a good overview of elements such as URL, http, Hyperlink etc. and their relations.