

MINI PROJECT

(2021-22)

Candy

Project Report



Institute of Engineering & Technology

Submitted By –

Prachi Agrawal(191500663)

Neha(191500489)

Ritika Brewal(191500663)

Under the Supervision of

Mr. Mandeep Singh

Technical Trainer

Department of Computer Engineering &
Applications



Department of Computer
Engineering and Applications
GLA University, 17 km. StoneNH#2,
Mathura-Delhi Road, Chaumuha,
Mathura – 281406 U.P(India)

Declaration

I hereby declare that the work which is being presented in the Bachelor of technology. Project “**Candy**”, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my own work carried under the supervision of **Mr. Mandeep Singh, Technical Trainer, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: *prachiagrawal, neha, ritikabrewal*

Name of Candidate: Prachi Agrawal, Neha, Ritika Brewal

University Roll No.: 191500552, 191500489, 191500663



Department of Computer Engineering
and Applications GLA University, 17
km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P
(India)

Certificate

This is to certify that the project entitled “Candy”, carried out in Mini Project – II Lab, is a Bonafide work by Prachi Agrawal, Neha, Ritika Brewal and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Mr. Mandeep Singh

Technical TrainerCEA

Department

Date: 25-05-2022



Department of Computer Engineering
and Applications GLA University, 17
km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P
(India)

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, I would like to place my deep gratitude to GLA University for providing me the instructor Mr. Mandeep Singh, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, I wouldn't have been able to complete this project.

So, I would like to thank my dear parents for helping us to grab this opportunity to get trained and our colleagues who helped us find resources during the training.

Thanking You

Sign: *prachiagrawal, neha, ritikabrewal*

Name of Candidate: Prachi Agrawal, Neha, Ritika Brewal

University Roll No.: 191500552, 191500489, 191500663



Department of Computer Engineering
and Applications GLA University, 17
km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P
(India)

Table of Contents

Declaration	2
Certificate	3
Acknowledgments	4
Table of Contents	5
1. Introduction	7
1.1 Aim	
2. Application Infrastructure	
Java Script	
Html	
CSS	
Bootstrap	
front end development	
3. Definitions	
Functionalities	
Motivation	

Objective	
Existing system	
4. IDE	23
Visual Studio Code	
5. Programming Language Used	25
6. Requirements	25
Software	
hardware	
7. Screen Shot	26
8. Code	54
9. Bibliography	116



**Department of Computer Engineering and
Applications GLA University, 17 km. Stone NH#2,
Mathura-Delhi Road, Chaumuha, Mathura – 281406
U.P (India)**

ABSTRACT

A Candy is a software program that allows you to access the internet and all the web pages in it. Most common browsers are Google Chrome, Microsoft Edge, Brave, Firefox, and Safari, and you are most definitely reading this article in one of these.

This project is a tutorial demonstrating the capabilities of the web platform on Windows 10. The browser is a sample app built around the using primarily python PYQt5 module. Built using Visual Studio 2015.

Python is an object-oriented programming language. The Qt library written in C++ is used for developing native desktop GUI applications and produces cross-platform code, so it's a good tool to develop multi-platform applications. We can easily create our own web browser in Python with the help of the PyQT5 library and the version of Python 3 will suit well for this tutorial, though Python 2.7 is still in use in many organizations and in my environment also.

PyQT is a widely used module which is used to make GUI applications with much ease. We can develop many complex GUI applications using PyQT very easily. It has a modern look and light user interface.

Additionally, we're taking advantage of the new ECMAScript 2015 (ES2015) support in Chakra, the JavaScript engine behind Microsoft Edge and the WebView control. ES2015 allows us to remove much of the scaffolding and

boilerplate code, simplifying our implementation significantly. The following ES2015 features were used in the creation of this app: `Array.from()`, `Array.prototype.find()`, arrow functions, method properties, `const`, `for-of`, `let`, `Map`, `Object.assign()`, Promises, property shorthands, Proxies, spread operator, `String.prototype.includes()`, `String.prototype.startsWith()`, Symbols, template strings, and Unicode code point escapes.



**Department of Computer Engineering and
Applications GLA University, 17 km. Stone NH#2,
Mathura-Delhi Road, Chaumuha, Mathura – 281406
U.P (India)**

Aim

Candy's main aim is to provide easy and customizable web browser for the user, so that user can modify it according its conveniency.



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

APPLICATION INFRASTRUCTURE

HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation ([CSS](#)) or functionality/behavior ([JavaScript](#)).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

At its heart, [HTML](#) is a language made up of [elements](#), which can be applied to pieces of text to give them different meaning in a document (Is it a paragraph? Is it a bulleted list? Is it part of a table?), structure a document into logical sections (Does it have a header? Three columns of content? A navigation menu?), and embed content such as images and videos into a page. This module will introduce the first two of these and introduce fundamental concepts and syntax you need to know to understand HTML.



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

JavaScript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled and multi-paradigm. It has dynamic typing, prototype-based object-orientation and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behaviour, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but they are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

Frontend Development

We used HTML, CSS, JavaScript and Bootstrap in our website for the front-end development.

DEFINITION

1.1 MOTIVATION

Browsers which are generally used by the all of are getting slower day by day and it takes a lot of space in the memory and save history and also save cache memory, but in our browser we are providing user a fast browsing and our browser is small in size. It will not take too much space in the memory and it will not save any history and any cache memory, it will help in fast browsing.



Department of Computer Engineering and Applications
GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

1.2 OBJECTIVE

The main objective of “Candy” is that it is a software program that allows you to access the internet and all the web pages in it. Most common browsers are Google Chrome, Microsoft Edge, Brave, Firefox, and Safari, and you are most definitely reading this article in one of these.



**Department of Computer Engineering and
Applications GLA University, 17 km. Stone NH#2,
Mathura-Delhi Road, Chaumuha, Mathura – 281406**

U.P (India)

1.3 EXISTING SYSTEM

1. Candy is a customized web browser which is similar to other browsers like Mozilla Firefox, Google Chrome, Microsoft Edge etc. with some differing functionalities.
2. It has Python for the backend part which is used to connect with the network so that whenever the user search for anything it will redirect it to the google search engine.
3. It has most visiting sites on the home page itself, which makes it more convenient for the user.



Department of Computer Engineering and
Applications GLA University, 17 km. Stone NH#2,
Mathura-Delhi Road, Chaumuha, Mathura – 281406
U.P (India)

SOURCE

The source of project (including all the project work, documentation) will is available at the following link-

mailto:https://github.com/prachi0011/candyBrowser_miniProject2

Function:

- 1.2.1 User will be able to search anything they want.
- 1.2.2 User experience fast browsing.
- 1.2.3 User can customize the browser accordingly.
- 1.2.4 User can easily search with the URL address.



**Department of Computer Engineering and
Applications GLA University, 17 km. Stone NH#2,
Mathura-Delhi Road, Chaumuha, Mathura – 281406
U.P (India)**

IDE

VISUAL STUDIO CODE:

Visual Studio Code is a free source code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, Code refactoring, and Embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Microsoft has released Visual Studio Code's source code on the VS Code repository of Github.com, under the permissive MIT License, while the compiled binaries are freeware.

In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. It is based on the Electron framework, which is used to develop Node.js Web Applications that run on the blink layout system. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure Devops (formerly called Visual Studio Online and Visual Studio Team Services).

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports several programming languages and a set of features that differs per language.

Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or

the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using language server Protocol.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.



**Department of Computer Engineering
and Applications GLA University, 17
km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P
(India)**

REQUIREMENTS:

Following are the hardware and the software requirements for our project:

a) Hardware:

- Laptop/Desktop
- 1.8 GHz or faster processor. Quad-core or better recommended
- 2 GB of RAM
- Hard disk space: Minimum of 800MB up to 210GB of available space
- Video card that supports a minimum display resolution of 720p(1280 by 720)

b) Software:

- Windows 8.1 and above
- Visual Studio Code

c) Languages used:

- Python
- Html
- CSS

PYTHON CODE:

```
# importing required libraries
# import subprocess
# subprocess.run("pip install PyQt5")

from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from PyQt5 import QtGui
from PyQt5.QtWebEngineWidgets import *
from PyQt5.QtPrintSupport import *
import os
import sys

# main window
class MainWindow(QMainWindow):

    # constructor
    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)

        # creating a tab widget
        self.tabs = QTabWidget()
        self.tabs.setStyleSheet("QTabBar::tab { height: 35px; width:
100px}")
        # making document mode true
        self.tabs.setDocumentMode(True)

        # adding action when double clicked

        self.tabs.tabBarDoubleClicked.connect(self.tab_open_doubleclick)

        # adding action when tab is changed
        self.tabs.currentChanged.connect(self.current_tab_changed)

        # making tabs closeable
        self.tabs.setTabsClosable(True)
```

```

# adding action when tab close is requested
self.tabs.tabCloseRequested.connect(self.close_current_tab)

# making tabs as central widget
self.setCentralWidget(self.tabs)

# creating a status bar
self.status = QStatusBar()

# setting status bar to the main window
self.setStatusBar(self.status)
#self.status.setGeometry(10, 10, 680, 480)

# creating a tool bar for navigation
navtb = QToolBar("Navigation")

# adding tool bar to the main window
self.addToolBar(navtb)

# creating back action
back_btn = QAction("◀", self)
#back_btn.setGeometry(100, 50, 50, 20)

# setting status tip
back_btn.setStatusTip("Back to previous page")

# adding action to back button
# making current tab to go back

back_btn.triggered.connect(lambda:self.tabs.currentWidget().back())

# adding this to the navigation tool bar
navtb.addAction(back_btn)

# similarly adding next button
next_btn = QAction("▶", self)
next_btn.setStatusTip("Forward to next page")

```

```

        next_btn.triggered.connect(lambda:
self.tabs.currentWidget().forward())
        navtb.addAction(next_btn)

        # similarly adding reload button
        reload_btn = QAction("↺", self)
        reload_btn.setStatusTip("Reload page")
        reload_btn.triggered.connect(lambda:
self.tabs.currentWidget().reload())
        navtb.addAction(reload_btn)

        # creating home action
        home_btn = QAction("△", self)
        home_btn.setStatusTip("Go home")

        # adding action to home button
        home_btn.triggered.connect(self.navigate_home)
        navtb.addAction(home_btn)

        # adding a separator
        navtb.addSeparator()

        # creating a line edit widget for URL
        self.urlbar = QLineEdit()

        # adding action to line edit when return key is pressed
        self.urlbar.returnPressed.connect(self.navigate_to_url)

        # adding line edit to tool bar
        navtb.addWidget(self.urlbar)

        # similarly adding stop action
        stop_btn = QAction("Stop", self)
        stop_btn.setStatusTip("Stop loading current page")
        stop_btn.triggered.connect(lambda:
self.tabs.currentWidget().stop())
        navtb.addAction(stop_btn)

        # creating first tab

```

```
self.add_new_tab(QUrl("C:/Users/hp/OneDrive/Documents/Candy/candyBrowser_miniProject2/homepage/home.html"), 'Homepage')
```

```
# showing all the components  
self.show()
```

```
# setting window title  
self.setWindowTitle("Candy")
```

```
# method for adding new tab  
def add_new_tab(self, qurl = None, label = "Blank"):
```

```
    # if url is blank  
    if qurl is None:  
        # creating a google url  
        qurl = QUrl('http://www.google.com')
```

```
    # creating a QWebEngineView object  
    browser = QWebEngineView()
```

```
    # setting url to browser  
    browser.setUrl(qurl)
```

```
    # setting tab index  
    i = self.tabs.addTab(browser, label)  
    self.tabs.setCurrentIndex(i)
```

```
    # adding action to the browser when url is changed  
    # update the url  
    browser.urlChanged.connect(lambda qurl, browser = browser:  
                                self.update_urlbar(qurl,  
browser))
```

```
    # adding action to the browser when loading is finished  
    # set the tab title  
    browser.loadFinished.connect(lambda _, i = i, browser =
```

browser:

```
browser.page().title()))
```

```
self.tabs.setTabText(i,
```

```
# when double clicked is pressed on tabs
```

```
def tab_open_doubleclick(self, i):
```

```
    # checking index i.e
```

```
    # No tab under the click
```

```
    if i == -1:
```

```
        # creating a new tab
```

```
        self.add_new_tab()
```

```
# when tab is changed
```

```
def current_tab_changed(self, i):
```

```
    # get the curl
```

```
    qurl = self.tabs.currentWidget().url()
```

```
    # update the url
```

```
    self.update_urlbar(qurl, self.tabs.currentWidget())
```

```
    # update the title
```

```
    self.update_title(self.tabs.currentWidget())
```

```
# when tab is closed
```

```
def close_current_tab(self, i):
```

```
    # if there is only one tab
```

```
    if self.tabs.count() < 2:
```

```
        # do nothing
```

```
        return
```

```
    # else remove the tab
```

```
    self.tabs.removeTab(i)
```

```
# method for updating the title
```

```
def update_title(self, browser):
```



```

# if signal is not from the current tab
if browser != self.tabs.currentWidget():
    # do nothing
    return

# get the page title
title = self.tabs.currentWidget().page().title()

# set the window title
self.setWindowTitle("Candy")

#to specify browser icon
QApplication.setWindowIcon(QtGui.QIcon("icon.jpg"))

# action to go to home
def navigate_home(self):

    # go to google
    self.tabs.currentWidget().setUrl(QUrl("http://www.google.com"))

# method for navigate to url
def navigate_to_url(self):

    # get the line edit text
    # convert it to QUrl object
    q = QUrl(self.urlbar.text())

    # if scheme is blank
    if q.scheme() == "":
        # set scheme
        q.setScheme("http")

    # set the url
    self.tabs.currentWidget().setUrl(q)

# method to update the url
def update_urlbar(self, q, browser = None):

```

```
# If this signal is not from the current tab, ignore
if browser != self.tabs.currentWidget():

    return

# set text to the url bar
self.urlbar.setText(q.toString())

# set cursor position
self.urlbar.setCursorPosition(0)

# creating a PyQt5 application
app = QApplication(sys.argv)

# setting name to the application
app.setApplicationName("Candy")

# creating MainWindow object
window = MainWindow()

# loop
app.exec_()
```

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Candy Web Browser</title>
  <link rel="stylesheet" href="/CSS/Candy.css">
</head>
<body>
  <nav>
    <a href="https://mail.google.com/mail/u/0/?ogbl#inbox"
target="_blank">Gmail</a>
    <a href="https://www.google.co.in/imghp?hl=en&authuser=0&ogbl"
target="_blank">Images</a>

  </nav>

  <section class="section-1">
    
    <form>
      <br><br>
      <div class="s-box">
        
        <input type="text" class="s-input">
        
        <input type="submit" class="s-btn" value="Search">
      </div>
    </form>
  </section>
  <div class="recents">
    <div class="icon">
      <a href="https://www.facebook.com/" target="_blank"> </a>
      <div class="txt">
        <a href="https://www.facebook.com/"
target="_blank">Facebook</a>
      </div>
    </div>
  </div>
</body>
</html>
```

```

</div>
<div class="icon">
  <a href="https://web.whatsapp.com/" target="_blank"></a>
  <div class="txt">
    <a href="https://web.whatsapp.com/"
target="_blank">Whatsapp</a>
  </div>
</div>
<div class="icon">
  <a href="https://twitter.com/" target="_blank"></a>
  <div class="txt">
    <a href="https://twitter.com/" target="_blank">Twitter</a>
  </div>
</div>
<div class="icon">
  <a href="https://www.youtube.com/" target="_blank"></a>
  <div class="txt">
    <a href="https://www.youtube.com/"
target="_blank">Youtube</a>
  </div>
</div>

</div>
<footer>
  <h4>India</h4>
</footer>
</body>
</html>

```

CSS CODE:

```
* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
  text-decoration: none;
  font-family: arial, sans-serif;
}
body {
  background-color: rgb(175, 213, 238);
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}
nav {
  display: flex;
  align-items: center;
  justify-content: flex-end;
  margin-right: 30px;
  margin-top: 10px;
  margin-left: 14px;
}
nav a {
  font-size: 17px;
  display: inline-block;
  line-height: 24px;
  font-weight: 400;
  color: rgb(0, 0, 0, 0.87);
  cursor: pointer;
  padding: 5px 6.7px;
}
nav a:hover {
  text-decoration: underline;
  color: purple;
  opacity: 0.85;
}
.section-1 img {
  width: 35%;
  margin: auto 7px;
  cursor: pointer;
}
.section-1 img:hover {
```

```

    opacity: 0.8;
}
.section-1 {
    flex: 1;
}
.section-1 .logo {
    display: block;
    margin: 0px auto;
    margin-top: 10px;
}
.s-box {
    width: 580px;
    margin: 0px auto;
    position: relative;
    margin-top: -8px;
    text-align: center;
}
.s-box .s-input {
    display: block;
    border: 1px solid rgb(194, 190, 190);
    color: rgba(0, 0, 0, 0.87);
    font-size: 16px;
    padding: 13px;
    padding-left: 45px;
    border-radius: 25px;
    width: 100%;
    transition: box-shadow 100ms;
    outline: none;
}
.s-box .s-input:hover {
    border: 1px solid white;
    box-shadow: 0 0 2px rgba(0, 0, 0, 0.05), 0 0 2px rgba(0, 0, 0, 0.05),
        0 0 3px rgba(0, 0, 0, 0.05), 0 0 4px rgba(0, 0, 0, 0.05),
        0 0 5px rgba(0, 0, 0, 0.05), 0 0 4px rgba(0, 0, 0, 0.05),
        0 0 5px rgba(0, 0, 0, 0.05);
}
.s-box .search-icon {
    width: 35px;
    opacity: 0.4;
    position: absolute;
    top: 13px;
    left: 14px;
}
.s-box .vs-icon {

```

```

width: 18px;
position: absolute;
top: 13px;
right: 24px;
}
.s-btn {
border: none;
padding: 9px 15px;
color: #666;
font-size: 14px;
border-radius: 4px;
display: inline-block;
margin-right: 10px;
margin-top: 28px;
outline: none;
cursor: pointer;
transition: border-color 100ms;
}
.s-btn:hover {
border: 1px solid #aaaa;
}
.recents {
width: 50%;
display: inline-flex;
padding: 10px;
margin-right: 20px;
margin-bottom: 20px;
margin-top: 10px;
margin-left: 360px;
}
.recents .icon {
border-radius: 5px;
width: 20%;
height: 120px;
margin-right: 20px;
background-color: white;
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
}
.recents .icon img {
width: 70px;
}

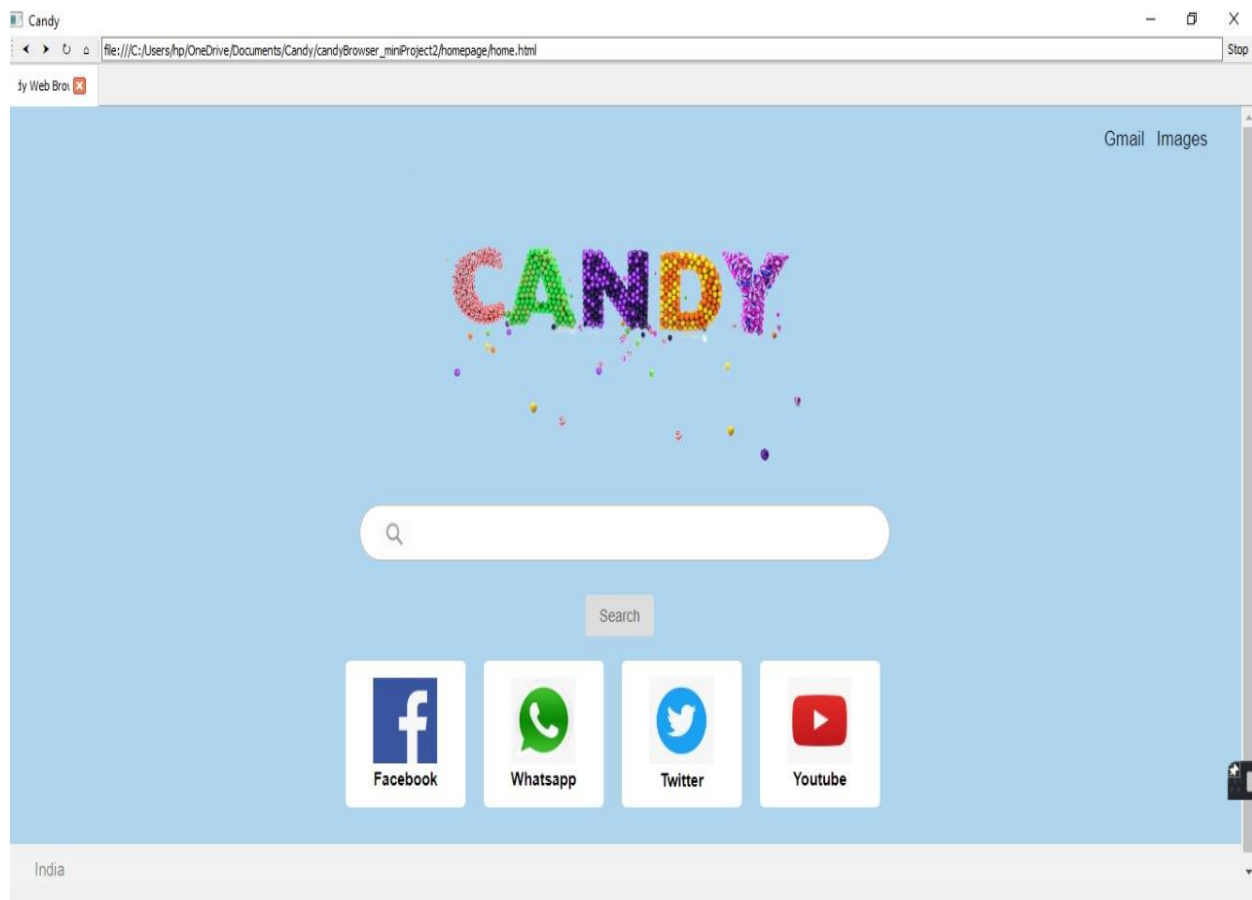
```

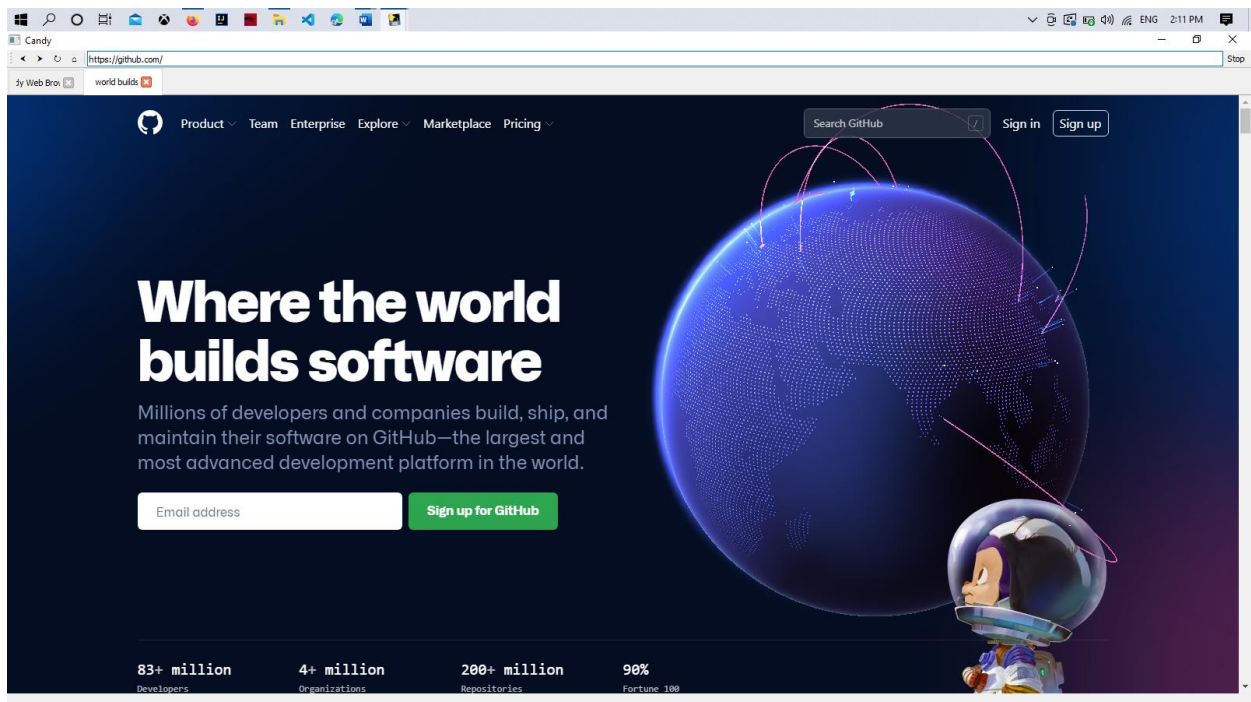
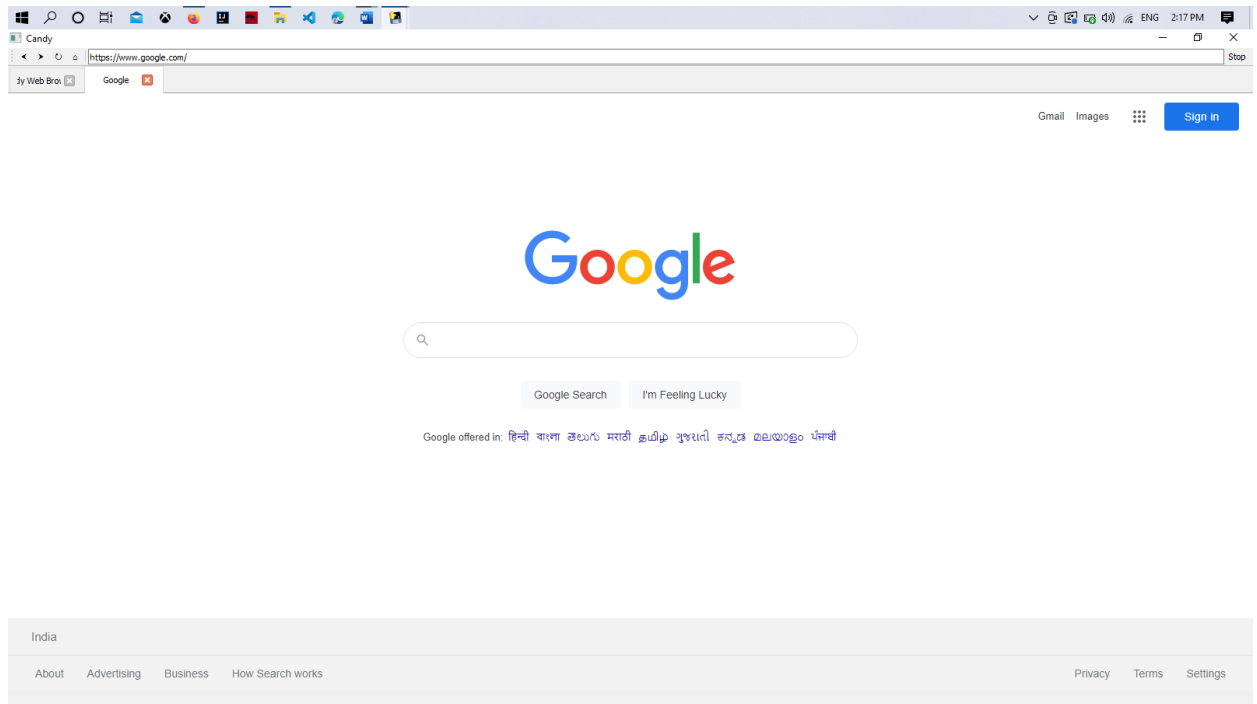
```

.recents .icon:hover {
  border: 2px solid white;
  box-shadow: 0 0 2px rgba(0, 0, 0, 0.05), 0 0 2px rgba(0, 0, 0, 0.05),
    0 0 3px rgba(0, 0, 0, 0.05), 0 0 4px rgba(0, 0, 0, 0.05),
    0 0 5px rgba(0, 0, 0, 0.05), 0 0 4px rgba(0, 0, 0, 0.05),
    0 0 5px rgba(0, 0, 0, 0.05);
}
.recents .icon .txt a {
  text-decoration: none;
  font-size: 15px;
  font: arial;
  color: black;
  font-weight: bold;
}
.recents .icon .txt a:hover {
  color: purple;
}
footer {
  background: #f2f2f2;
}
footer h4 {
  color: rgba(0, 0, 0, 0.54);
  font-size: 15px;
  line-height: 39.5px;
  font-weight: 400;
  padding-left: 29px;
  border: 1px solid #eee;
}
footer a {
  color: #5f6368;
  display: inline-block;
  padding: 12px 11.3px;
  font-size: 14px;
}
footer a:hover {
  text-decoration: underline;
}

```


RESULT:





CONCLUSION:

A web browser opens a connection to a web server based on the URL entered by the user. The URL specifies the protocol to use, typically http for web servers. The last entry in the URL includes the directory and file names that the browsers is looking for. Web browsers are of different kinds but all are used to search what we want from web, the browsers differ each and every one, the action they perform was same. Some browsers are more effective compare to others.

BIBLIOGRAPHY

1. <https://www.beta-labs.in/>
2. <https://www.w3schools.com/>
3. <mailto:https://pypi.org/project/PyQt5/>
4. <https://html.com/>

