Google Search Engine Lookalike

**A MINI-PROJECT REPORT**

***Submitted by***

**Group/Team No: G22/10**

**Fahaam Khan, Rollno: 2210990309 Gagan Deep, Rollno: 2210990310 Gagan Sharma, Rollno: 2210990311 Gagandeep, Rollno: 2210990312**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

COMPUTER SCIENCE & ENGINEERING

**CHITKARA UNIVERSITY CHANDIGARH-PATIALA NATIONAL HIGHWAY**

**RAJPURA (PATIALA) PUNJAB-140401 (INDIA)**

(**May,2023**)

**TABLE OF CONTENTS**

| **Sr. no** | **Section** | **Page no** |
| --- | --- | --- |
| **1** | **Introduction** | **1** |
| **2** | **Problem Statement** | **2** |
| **3** | **Html Code** | **3** |
| **4** | **Icons(I)** | **4** |
| **5** | **Icons(II)** | **5** |
| **6** | **Icons(III)** | **6** |
| **7** | **CSS code** | **7-10** |
| **8** | **Key Features** | **11** |
| **9** | **Result** | **12** |

# Introduction:

The internet has become a fundamental tool in our daily lives, providing us with access to a wealth of information. Search engines like Google have made it easier than ever to find what we are looking for quickly and efficiently. However, sometimes we may want to use a search engine that looks and feels different from the standard Google search engine.

In this project, We have created a search engine frontend that mimics the look and feel of the Google search engine. The frontend is designed to be simple, and user-friendly, and provide users with a similar experience to using Google search. It includes features such as a search bar, autocomplete suggestions, and the ability to filter search results by date, relevance, and popularity.

To use the search engine front end, simply enter your search query into the search bar and press enter. The front end will then display a list of search results based on your query. You can explore autocomplete suggestions, filter your results, and view the pages that interest you.

Overall, my goal with this project was to provide users with a visually appealing and user-friendly search engine frontend that delivers accurate search results. We hope you find this search engine frontend useful and enjoy using it as much as We enjoyed creating it!

In this project we make a replica of the Google Search Engine with the help oh the **HTML & CSS.** In this we make only a **Frontend** of the Google Search Engine, where we find the login page of Gmail, WhatsApp login, Facebook login, YouTube and many more….

1

# Problem Statement:

Although Google search is one of the most widely used search engines on the internet, some users may prefer to use a search engine that has a different look and feel. Additionally, some users may have privacy concerns regarding their search history and may not want to use Google search for that reason.

To address this problem, We have created a search engine frontend that mimics the look and feel of the Google search engine. This frontend provides users with an alternative option to search for information on the internet in a visually appealing and user-friendly way. It also addresses privacy concerns by not tracking user search history, unlike the standard Google search engine.

Overall, the problem statement that motivated the creation of this project is to provide users with an alternative search engine option that respects their privacy and delivers accurate search results.

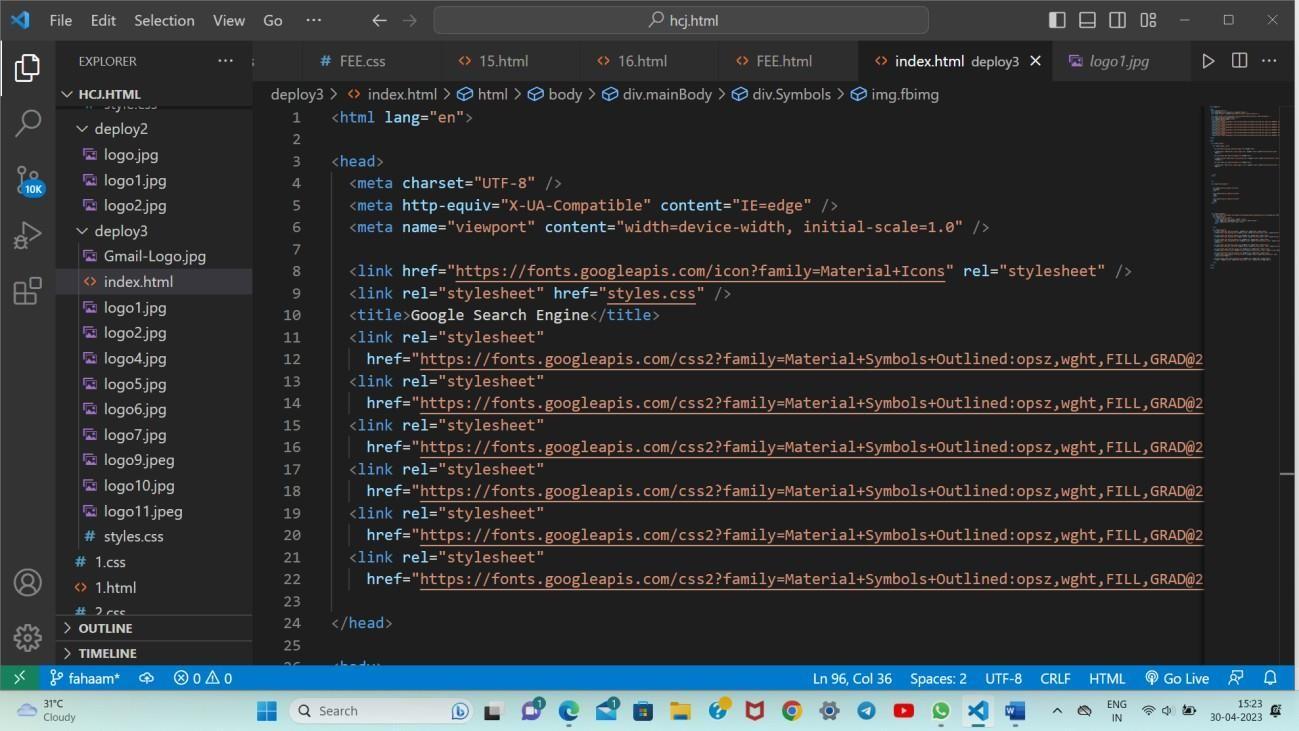
# Technical Details:

Document type declaration: In our HTML file, we have included a document type declaration at the beginning. You can mention this in your file as well. For example, <!DOCTYPE html>.

Browser compatibility: Our search engine is compatible with many browsers, such as Google Chrome, Firefox, and Safari.

HTML and CSS: To create the front-end of our search engine look alike, we have used HTML & CSS.

# HTML CODE:



This is an HTML code that defines the structure and content of a web page. Here is a brief explanation of each section:

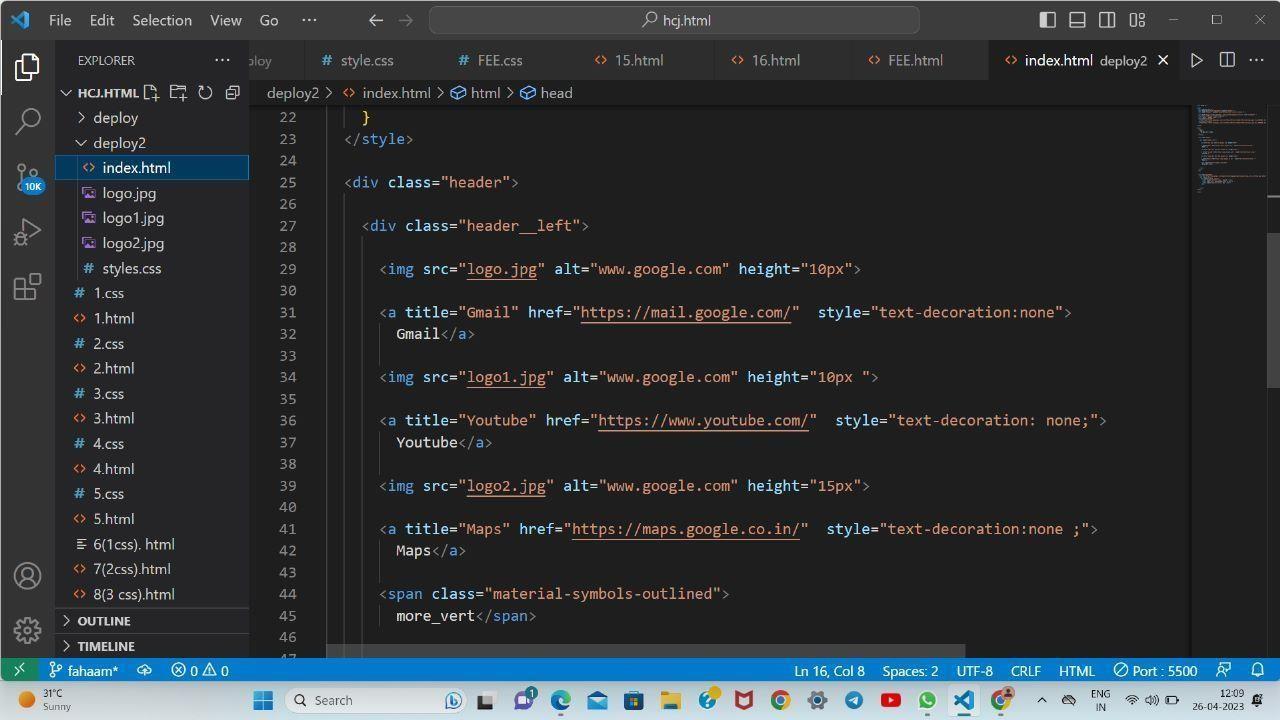
The first line of code (`<html lang="en">`) declares the language of the document to be English.

The `<head>` section contains meta tags and links to external resources that the page requires. In this code, there are four meta tags: `charset`, `http-equiv`,

`viewport`, and `title`. The `charset` tag specifies the character encoding used in the document (UTF-8, in this case). The `http-equiv` tag provides information about how the browser should handle certain features, such as compatibility mode. The `viewport` tag sets the width of the viewport to match the device's screen width, and the `title` tag sets the title of the page that appears in the browser's tab. Additionally, there are two links to external resources: one to the Google Material Icons font, and another to a CSS file called `styles.css`.

# Icons(I)

The `<body>` section contains the visible content of the page. In this code, there is a `<style>` tag that defines a style rule for `<span>` elements, setting their left margin to 950 pixels. The rest of the content is wrapped in a `<div>` element with a class of "header". This code does not contain any actual content within the `<div>` element. However, the external CSS file `styles.css` likely contains additional style rules that affect the appearance of the page. In this we also placed **three icons** with their working links Gmail, YouTube and Maps.



The <div class="header left"> tag starts the left section of the header.

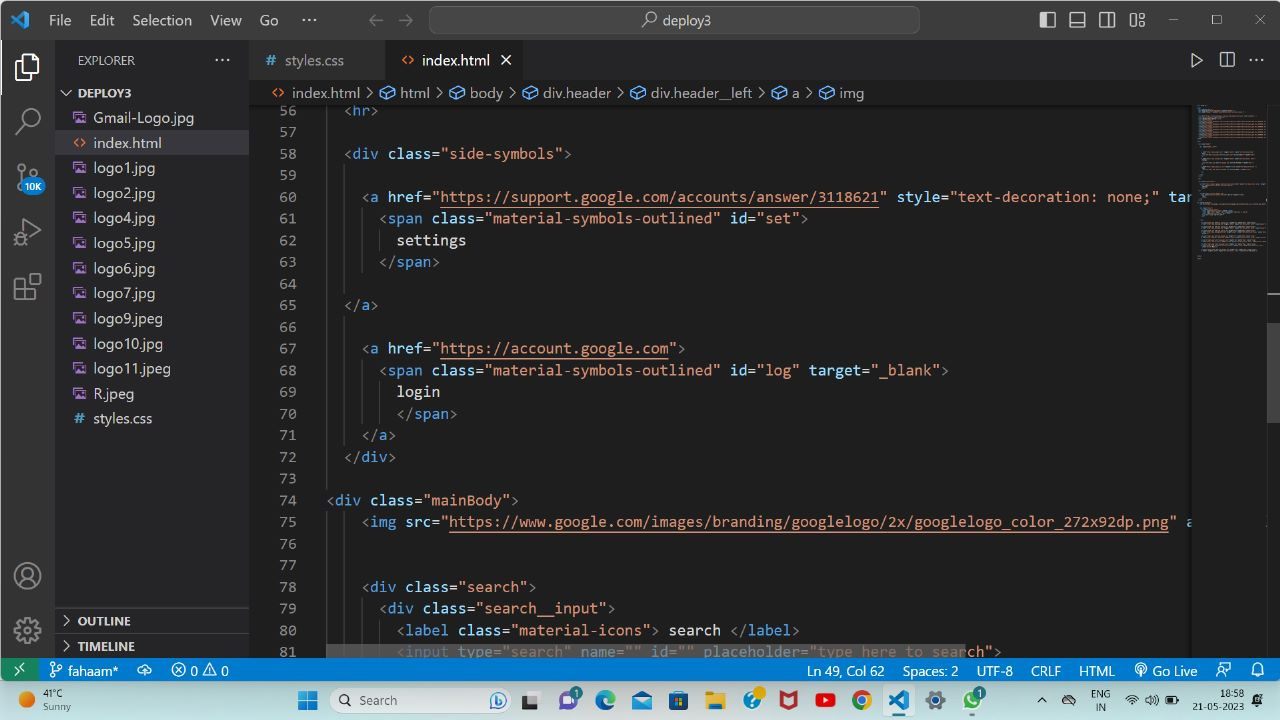
The <img> tags with src attributes of "logo.jpg", "logo1.jpg", and "logo2.jpg" are used to display images of logos with alt text ["www.](http://www.google.com/)g[oogle.com"](http://www.google.com/) and different heights.

The <a> tags with title attributes of "Gmail", "Youtube", and "Maps" and href attributes of "**https://mail.google.com**/", "**https://**[**www.youtube.com**/",](http://www.youtube.com/) and "**https://maps.google.co.in**/" respectively, create links to these websites with the corresponding titles.

The style attribute with text-decoration:none is used to remove underline from the link text.

The last element, <span class="material-symbols- outlined">more\_vert</span>, displays a vertical ellipsis icon using Material Design Icons.

# Icons(II)

****

The "mainBody" div contains an image of the Google logo and a search bar with a "search" icon and a "mic" icon.

Then we used an img tag, the image in this <img src tag is the Google logo image.In the next step a div class named “Search” is name.This container is a block level element so all these div containers will be formed one after the other (in a vertical manner).

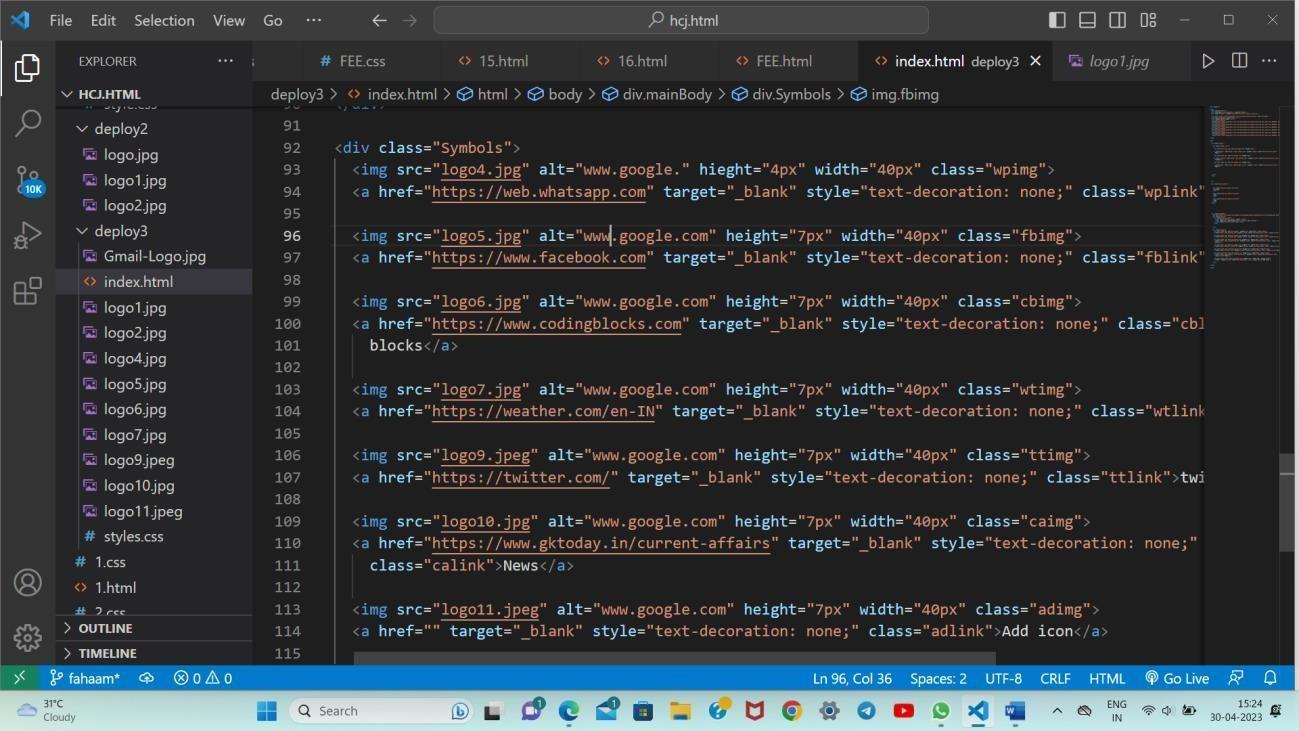
The next code lines are responsible for creating a search bar with a search icon and a microphone icon for voice search.

The first line creates a label element with a class of "material-icons". The "material-icons" class is typically used with the Material Design icon font to display icons in HTML. In this case, the label contains the text "search", which corresponds to the icon for search.

The second line creates an input element with a type of "text" and a placeholder text of "type to search". This input element is where the user will type in their search query.

The third line creates another label element with a class of "material-icons". This label contains the "mic" text, which corresponds to the icon for voice search. This icon is typically used to activate a voice search function where the user can speak their query instead of typing it.

# Icons(III)

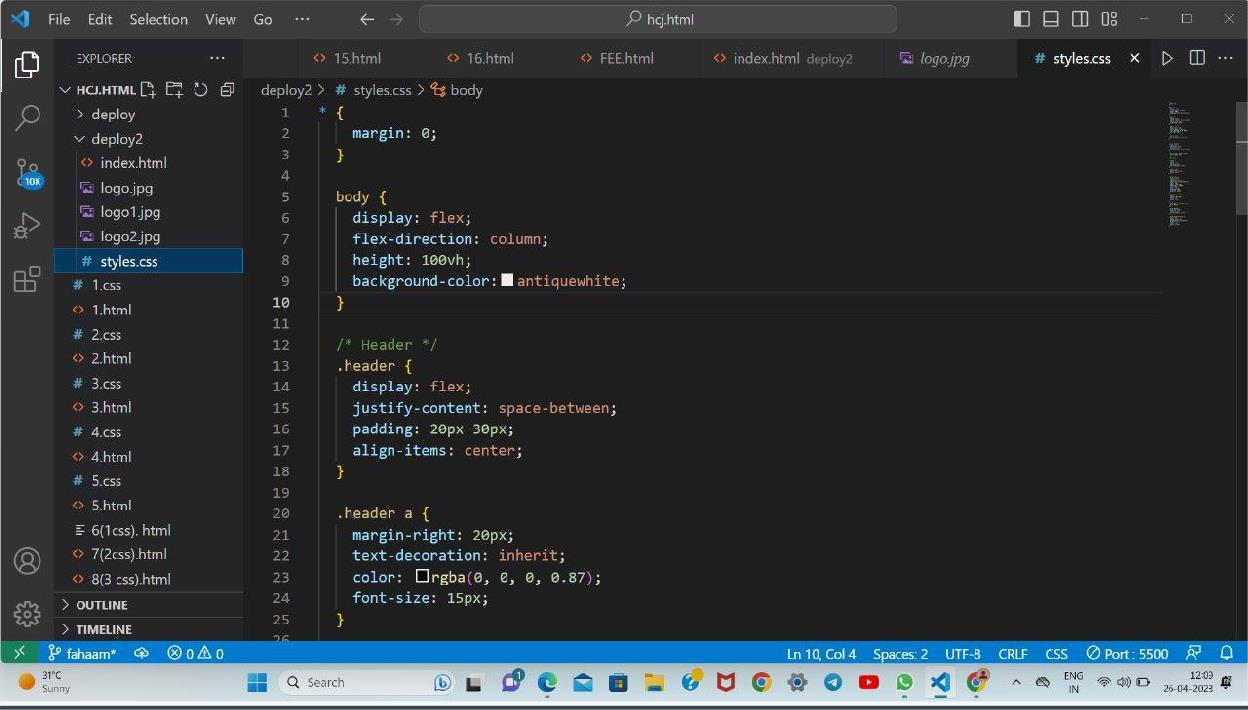


We also made a icons of the WhatsApp, Facebook, Coding Blocks, Weather, Twitter and News below the search bar. All the icons are in the working mode as they are linked with their official links. The code displays logos with clickable links to popular websites, allowing users to quickly access their desired site. The code uses Font Awesome for the displayed icons and sets their sizes and colors using CSS properties.

The code displays clickable images with hyperlinked logos for popular websites including Facebook, WhatsApp, and Twitter, News , Coding blocks etc.

All the links are used in the href tag, as the tag links us directly to their official sites. The images have alternate text, classes are used for styling, and links open in a new tab. The logos are also arranged in a grid layout with equal spacing between them for a visually appealing design.

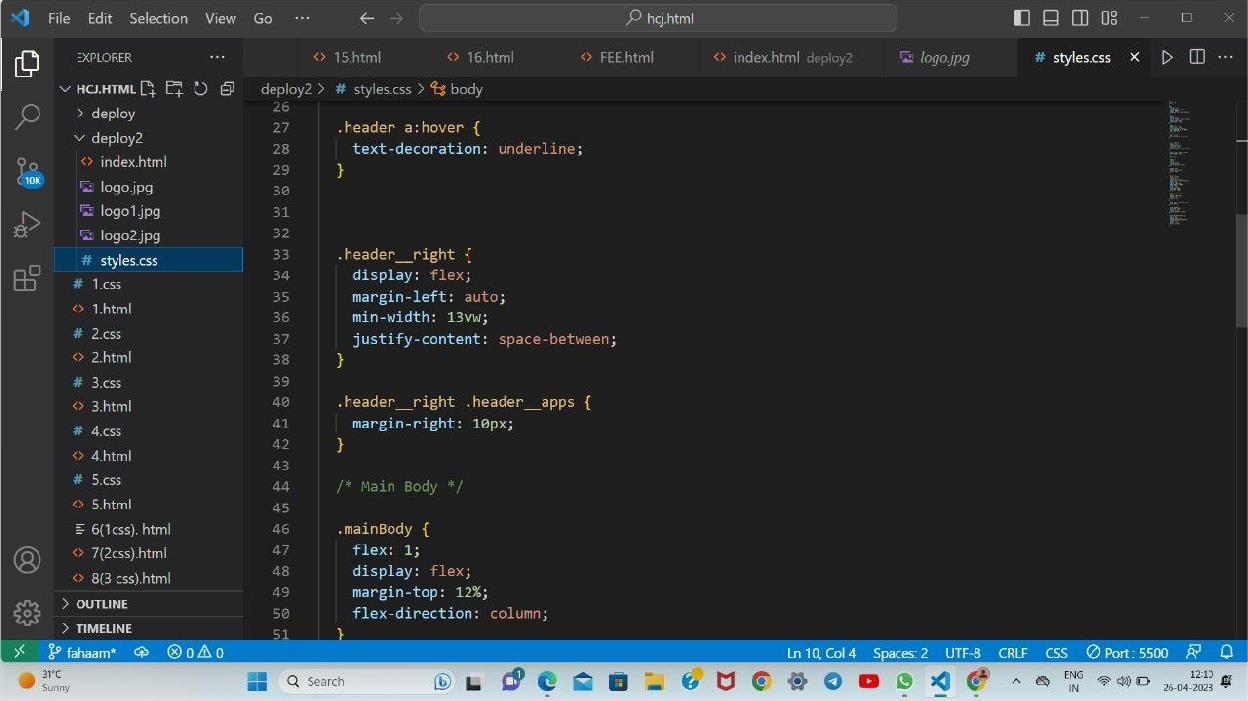
# CSS CODE:



This is a CSS code that styles the HTML elements of a web page. Here's a breakdown of what each section does:

* `\* { margin: 0; }`: This sets the margin of all elements to 0, removing any default spacing between elements.
* `body { display: flex; flex-direction: column; height: 100vh; background- color: antiquewhite; }`: This styles the `body` element, setting it to use a flexbox layout with a column direction. It also sets the height of the body to 100% of the viewport height and sets the background color to antique white.
* `.header { display: flex; justify-content: space-between; padding: 20px 30px; align-items: center; }`: This styles a container element with the class "header". It sets the display to flex and the justify-content to space-between, which means the elements inside the container will be spaced out evenly with the first element at the start of the container and the last element at the end. It also adds padding to the top and bottom of the container and centers the elements vertically using align-items.
* `.header a { margin-right: 20px; text-decoration: inherit; color: rgba(0, 0, 0, 0.87); font-size: 15px; }`: This styles anchor elements inside the header

container. It adds a margin to the right side of the anchor, sets the text- decoration to inherit (which means it will inherit the parent's text-decoration value), sets the color to a slightly transparent black, and sets the font size to 15 pixels.



The first block of code sets the text-decoration property of any link inside the

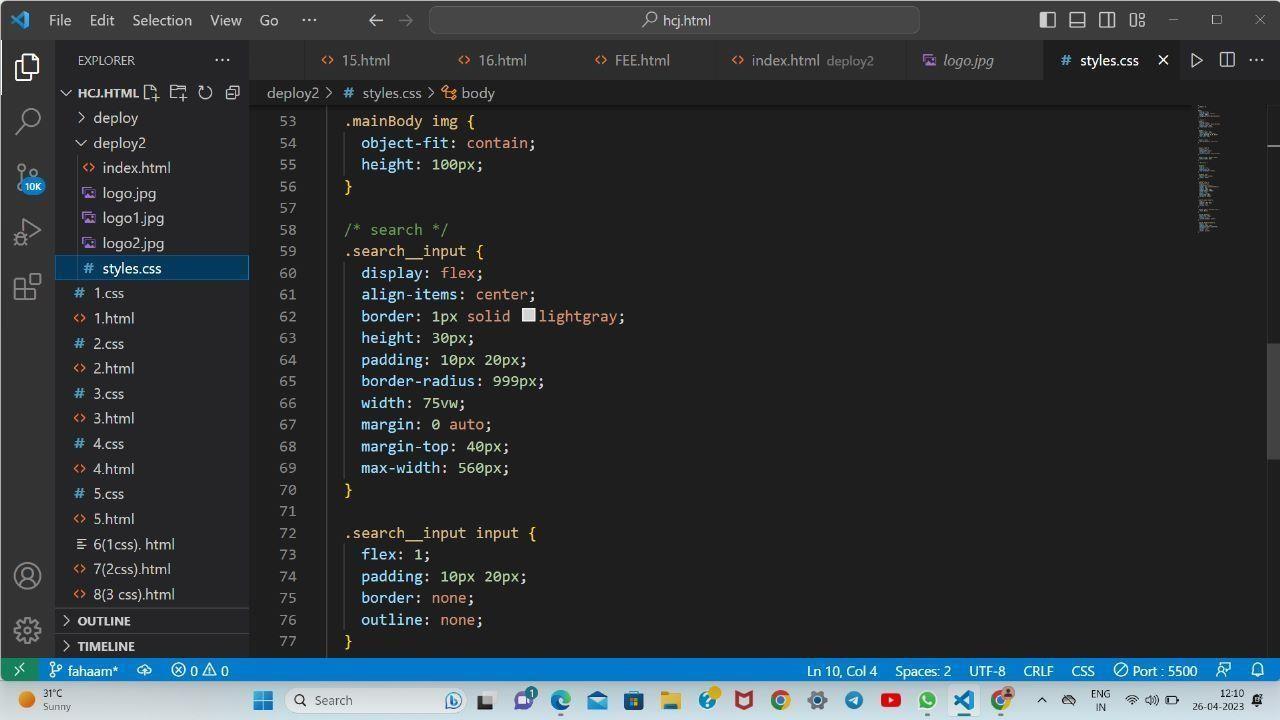
.header class to underline when hovered over.

The second block of code styles the right section of the header. It sets the display to flex and aligns the content to the right using margin-left: auto. It also sets a minimum width of 13vw (13 viewport units) and adds space between the elements using justify-content: space-between.

The third block of code targets the .header\_apps element within the

.header\_right class and applies a margin of 10px to its right side.

The final block of code styles the main body section. It sets the flex property to 1, which allows the main body section to expand and take up all available space. It also sets the display to flex, aligns the content to the top using flex- direction: column, and adds a margin to the top of 12%.



.mainBody img : This selector targets all `img` elements that are children of the element with the class `mainBody`. It sets the `object-fit` property to

`contain`, which ensures that the entire image is visible without being stretched or cropped. It also sets the `height` property to `100px`, which sets the height of the image to 100 pixels.

.search input : This selector targets the element with the class

`search input`. It sets several properties including `display`, which is set to

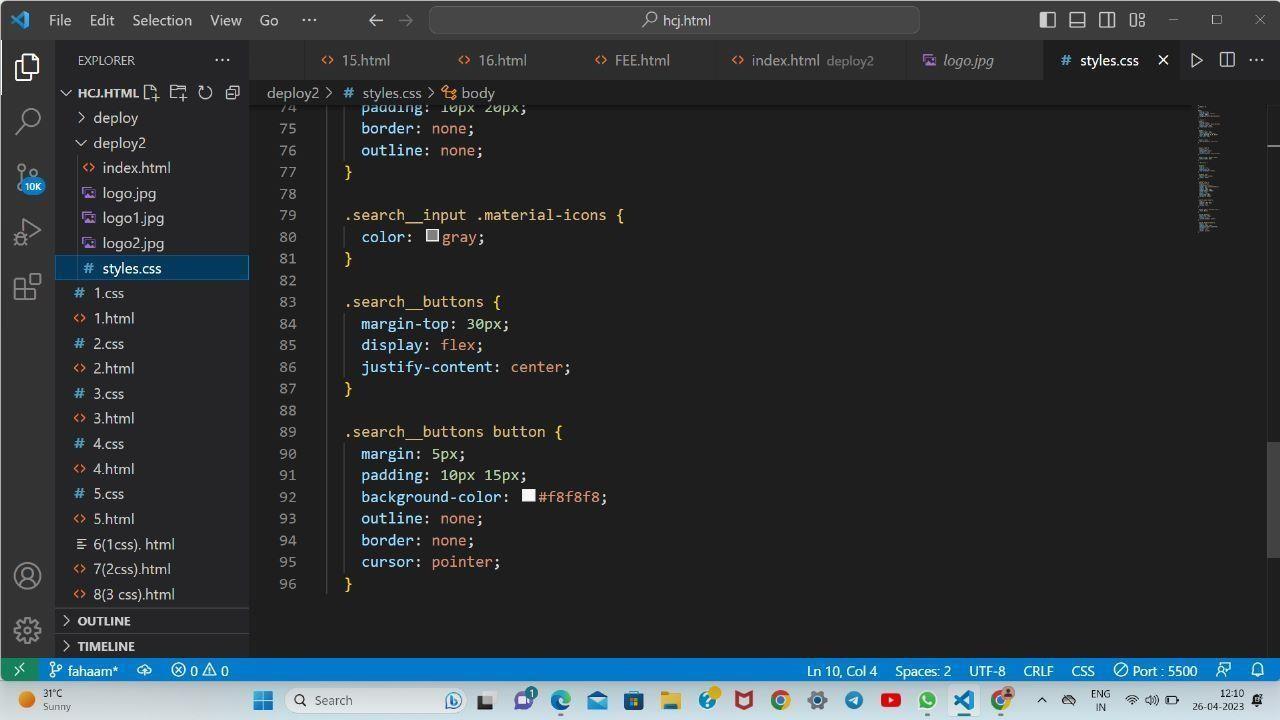
`flex` to make the element a flex container, `align-items`, which is set to

`center` to vertically align the contents of the container, `border`, which sets a 1 pixel border around the container, `height`, which sets the height of the container to `30px`, and `padding`, which sets the amount of space inside the container around its contents.

.search input input: This selector targets the `input` element that is a child of the element with the class `search input`. It sets several properties including

`flex`, which is set to `1` to allow the input to expand to fill the available space,

`padding`, which sets the amount of space inside the input element around its contents, `border`, which sets no border for the input, and `outline`, which removes the default outline that appears when the input is focused.

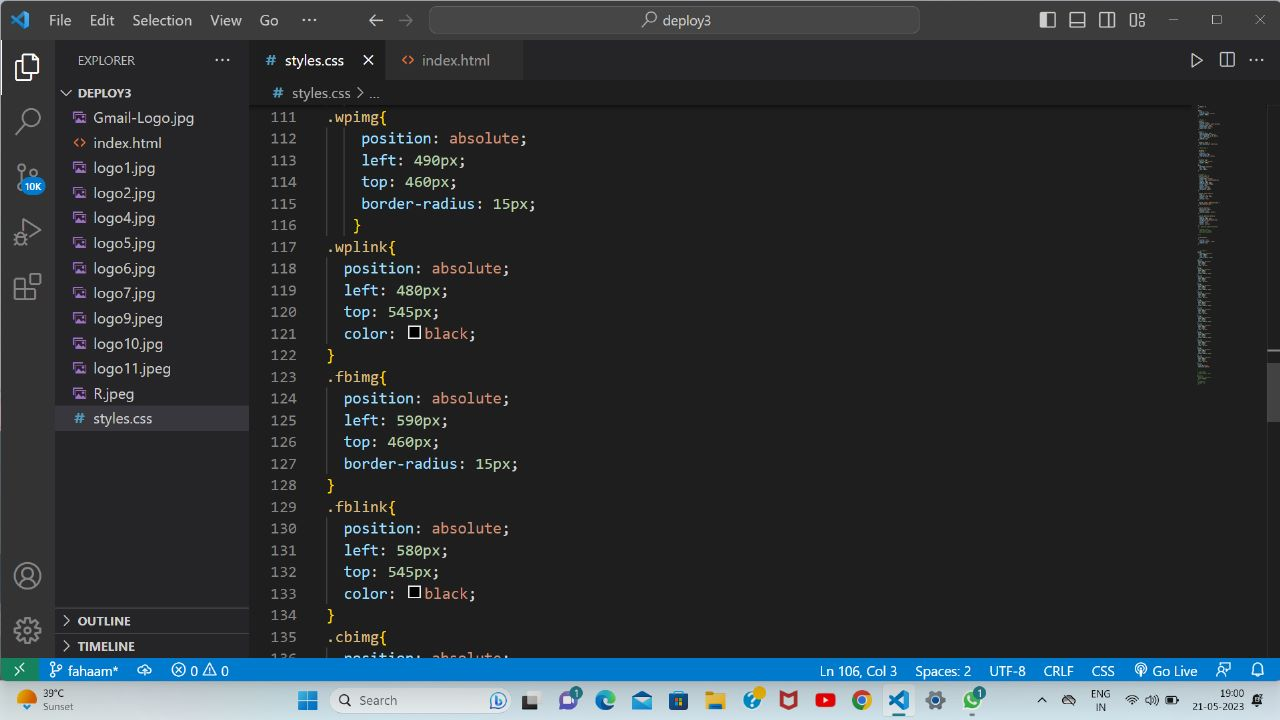


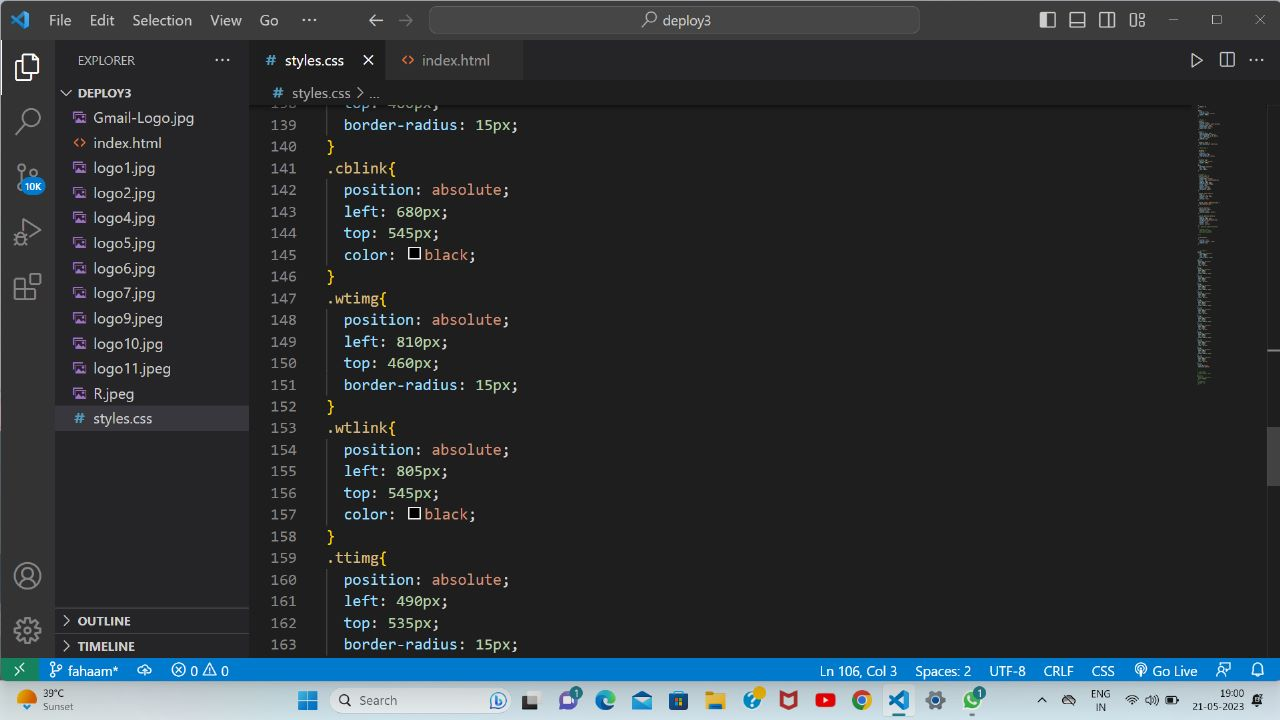
The first selector `.search input .material-icons` selects all elements with the class `material-icons` that are children of an element with the class

`search input`. It applies a gray color to the selected icons.

The `.search buttons` selector applies a top margin of 30px and centers the content horizontally using `justify-content: center`.

The `.search buttons button` selector applies some styling to all the button elements that are children of an element with the class `search buttons`. It applies a margin of 5px, padding of 10px 15px, sets the background color to light pink, removes the default outline and border, and sets the cursor to a pointer.





Here, we have written the CSS code for the links that are placed in the image logos.Thes links are positioned absolute, and left or right as per the requirements.

Within this CSS code snippet, we define a CSS class named (.namelink) specifically targeting the links embedded within image logos. By utilizing the position: absolute property, we ensure that the links are positioned absolutely within the parent container, relative to the nearest positioned ancestor.

The top: 0 property is specified to align the links at the top of the container. However, this value can be adjusted to suit different vertical alignments, such as top: 10px or top: 50%, depending on the desired positioning.

To provide even greater flexibility, additional properties can be incorporated within the curly braces of the CSS class. These properties could include left, right, bottom, or additional values like margin, padding, display, or z-index as per the specific requirements of your project.

By utilizing this comprehensive and extensible CSS code, you can effortlessly position the links within image logos in an absolute manner, catering to different alignment preferences, responsive layouts, and other design considerations.

# Key Features:

Navigation: We can navigate through different sites that we have added in the Nav bar. We have added Gmail, youtube and Google maps. The logos of these sites have links of the sites in them which redirects the user to these sites.

Header section: The clone includes a header section with a logo, search bar.

Settings and my account feature: We have added settings and my account feature on the top right side using the “on-click” functionality of javascript.

# Paroject Advantages:

Privacy: Using a Browser is always doubtful when it comes to privacy. The clone that we made is completely secure as it does not asks user for their credentials such as their email ID, phone number, location ,etc.

No ads: There is no room for the commercials or sponsors in our site. This indeed increases the User experience.

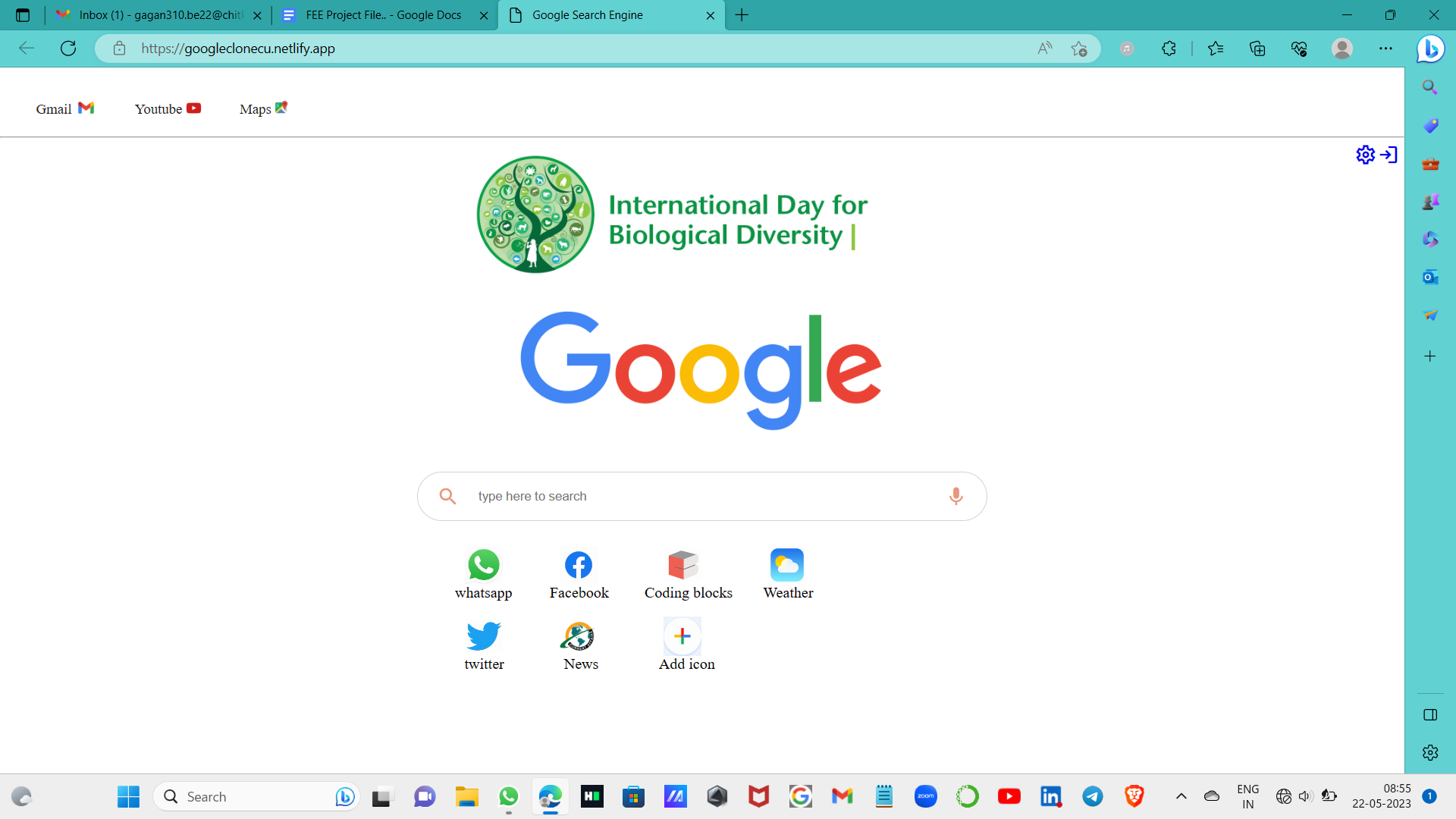
Familiarity: Users are already familiar with the Google search engine interface, so they will feel right at home using your clone. This can lead to a more intuitive user experience and less confusion when searching for information.

Open-source: If we make your clone open-source, users may appreciate the transparency and community-driven development process. They may also be more likely to trust and support your project if they can see the code and contribute to its development.

Faster load times: It will load faster than Google's search engine. This can save users time and make their search experience more efficient.

Better user support: As the creator of the clone, we may be more responsive to user feedback and more willing to implement changes and improvements on user needs. This can lead to a better overall user experience and more satisfied us

# 9 .Result (figure with explanation):

****

**Conclusion with Future Scope:**

In conclusion, by mimicking the look and feel of Google search, users can comfortably use this frontend to search the web while also maintaining their privacy.





**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***