Flopan Project Report

A report submitted in partial fulfilment of the requirements
for the award of degree of

Bachelor of Technology
In
Computer Science and Engineering

Submitted by:
Bazgha Razi

Submitted to:
Mr. Kapil Singh

Delhi Global Institute of Technology

Bahadurgarh, Jhajjar, Haryana 124201

Affiliated to

Maharshi Dayanand University

CERTIFICATE

This is to certify that the minor project FLOPAN submitted by Bazgha Razi, to Maharishi Dayanand University(MDU), Rohtak-124001, Haryana in partial fulfilment of the requirement for the award of semester 6^{th} of the degree of B.Tech in Computer Science Engineering. This project fulfils the requirement as per the regulation of the institute of university in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this time institute or any other institute or university to the best of my knowledge and belief.

Mentor Signature

ACKNOWLEDGEMENT

It has been great honour and privilege to undergo this project.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all.

I am highly indebted to professors for their guidance and constant supervision as well as for providing necessary information regarding the project and for their support in completing the project. Their constant guidance made me understand this project and its manifestations in great depths helped us to complete the assigned tasks on time.

I am also thankful and grateful to my parents who helped me throughout this project period.

Bazgha Razi

B.Tech(CSE)

DECLARATION

I here by declare that the project report entitled “FLOPAN” submitted by me. Bazgha Razi partially fulfilment the requirements for this project work under the guidance of professors.

I also declare that, the report is only prepared for my academic requirement not for any other purpose.

Bazgha Razi
B.Tech(CSE)

CONTENTS

INTRODUCTION

In this project I have to design an extension/addons. Name of the addon is FLOPAN. The name flopan is derived from floating panel. It is a special type of multi-window mode addons.

This project has been a great learning experience as I have discovered most of the tools and techniques during this project period. Learned about various frameworks and also learned about APIs for making Floapn addon.

Prior to this project period, I've already experience working with Javascript. It is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user. It is also used in making web portal.

During the project period, I used Javascript and APIs for making Flopan.

Also work with canva, figma and wireframes for designing the addon's UI/UX.

I have also met many guides during the building period of this project. They are passionate about technology and innovation. They made this journey truly inspiring from a technological point.

PROJECT DETAILS

Name of the Project

FLOPAN

Frontend Used

HTML

CSS

JavaScript

Backend Used

JavaScript

Picture-in-picture API

Screen Capture API

Version Control/ IDE

Github

VS Code

Web Browsers

Microsoft Edge

Google Chrome

Flopan Logo

FLOPAN VISION AND MISSION

Vision: To become the best multi-window addon provider in industry to fulfill the dreams of every aspirant for shaping her/his future by taking help from Flopan and utilizing its features.

Mission: To help students as well as who are working on the research work.It helps to read two articles simultaneously without switching to tabs while making notes or any research.

FLOPAN

Flopan stands for floating panel. It is a special type of multi-window mode addons. It lets the user watch a video or read an article in a small window pinned to a corner of the screen while navigating between apps or browsing content on the other screen.

Inspiration

FLopan was inspired by picture and picture technique. In picture in picture method we can float the video while browsing to other tabs or applications. So, flopan is an upgraded version of PIP(Picture in Picture). Flopan not only provide floating window for videos but also we float any application or tab while using other. It helps you to read two articles simultaneously without switching to tabs for making your notes or doing any research.

Why Flopan?

- Flopan is an upgraded version of PIP based addons.
- It lets the user watch a video in a small window pinned to a corner of the screen while navigating between apps or browsing content on the main screen.
- It not only provide floating window for videos but also we float any application or tab while using other.
- It helps you to read two articles simultaneously without switching to tabs.
- It helps during any research.
- It’s also very useful for making notes.
- We can resize the flopan screen and also move it according to our requirement.

Features of FLOPAN

Start Button

It is created for starting the addon/ extension.

Screen Select

It is a feature in which we can select any tab, screen and application that we want to see in the floating window.

Floating Window

It is a small window pinned in the bottom right corner of the screen. We can also resize and move floating window as per our requirements.

Close Button

It is required to close the floating window mode.

Weekly Work For Project

Week 1	Defining scope for the project, idea brainstorming and read up about addon privacy research.
Week 2	Design discussions and UI mockups for addon's pop-up page.
Week 3	Create Addon's Web Page UI/UX. Add all the necessary components using some languages and frameworks.
Week 4	Create visualisations and more design discussions. Scope evaluation and revisit the initial plan to make sure minimum viable product is completed on time.
Week 5	Study about the PIP web API as well as Screen Capture API.
Week 6	Evaluation and testing of the FLOPAN addon.

What is an IDE?

An integrated development environment (IDE) is software for building applications that combines common developer tools into a single graphical user interface (GUI). An IDE typically consists of:

- **Source code editor**: A text editor that can assist in writing software code with features such as syntax highlighting with visual cues, providing language specific auto-completion, and checking for bugs as code is being written.
- **Local build automation**: Utilities that automate simple, repeatable tasks as part of creating a local build of the software for use by the developer, like compiling computer source code into binary code, packaging binary code, and running automated tests.
- **Debugger**: A program for testing other programs that can graphically display the location of a bug in the original code.

An IDE allows developers to start programming new applications quickly because multiple utilities don't need to be manually configured and integrated as part of the setup process. Developers also don't need to spend hours individually learning how to use different tools when every utility is represented in the same workbench. This can be especially useful for onboarding new developers who can rely on an IDE to get up to speed on a team's standard tools and workflows. In fact, most features of IDEs are meant to save time, like intelligent code completion and automated code generation, which removes the need to type out full character sequences.

NOTE: During my project I used Visual Studio Code for my Flopan.

HTML and CSS

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.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

In simple terms, HTML is a set of commands that tell web browsers how to display specific parts of websites. For example:

<h1>This is a header</h1>

This is a paragraph of text with a hyperlink.

This is an element of an unordered list
This is another one

The parts in angle brackets are called tags. A tag is a command that tells the browser that the following text needs to be displayed in a certain way. A tag can also inform the browser that there needs to be an image or a button in a certain place.

You put these tags in a text file and show it to the browser. The browser reads these tags and renders the web page without the tags, but minding what they mean. Browsers on computers and smartphones understand these commands in more or less the same way, so you can be sure that your web page will work the same way across most browsers around the world.

For example, the text above would look like this:

This is a header

This is a paragraph text with a hyperlink.

● This is an element of an unordered list

● This is another one

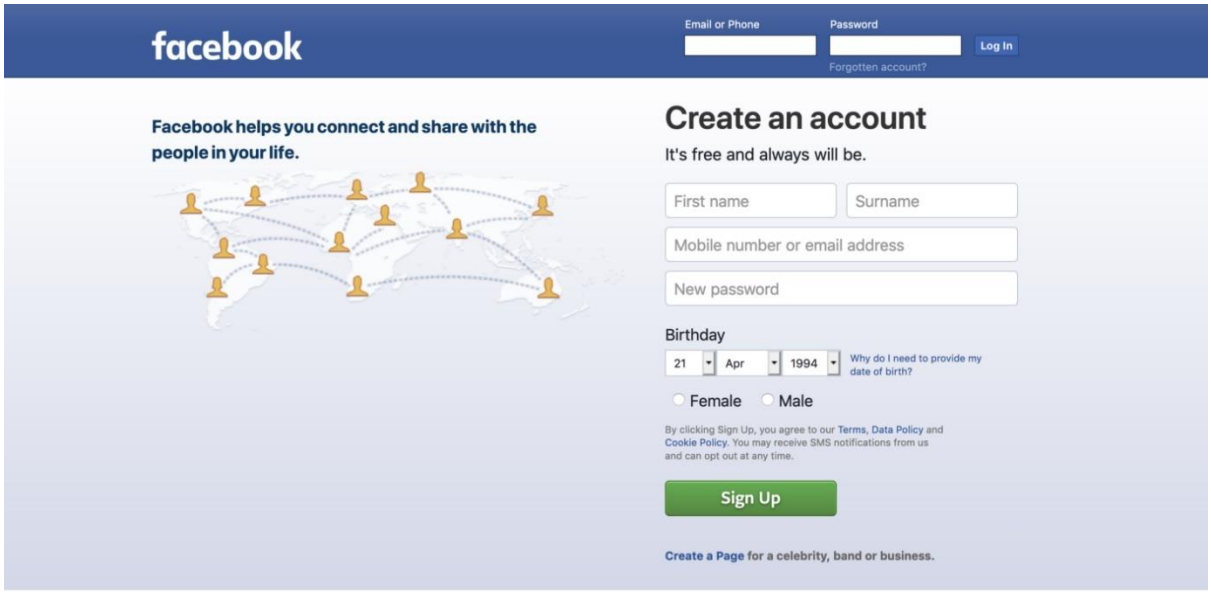
There are many markup languages in the world. For example, both Microsoft Word and Apple Pages use their own proprietary markup languages to store information about your text documents. When you write something in MS Word, there is a lot of design data embedded within your document. Although you may not see it, that data is there in the form of markup commands.

CSS

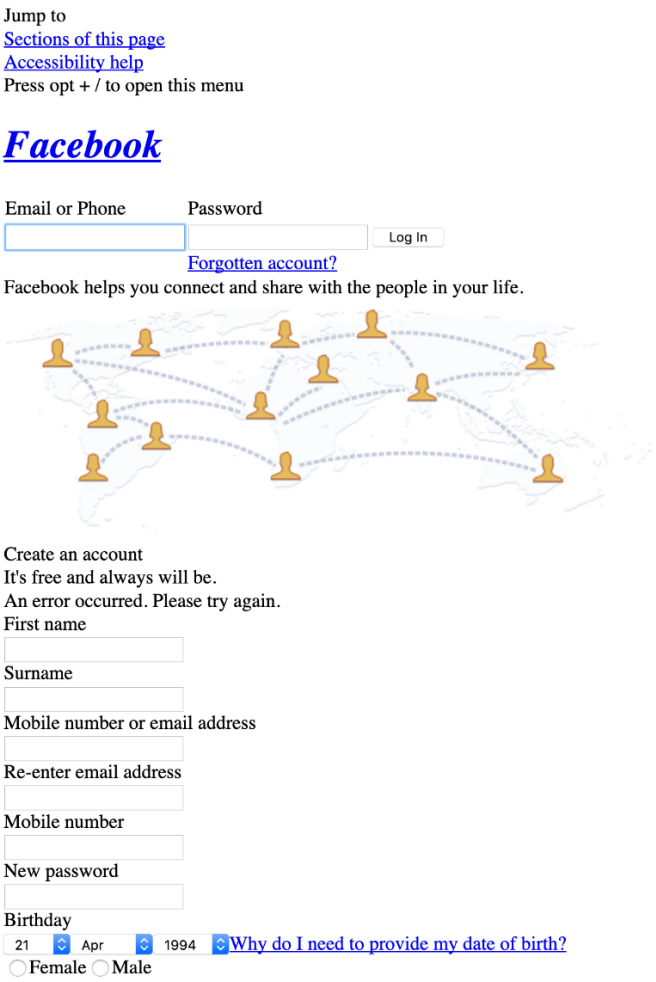
CSS (Cascading Style Sheets) is a language for styling the webpage. We can change the appearance and the layout of the webpage by using CSS. We can also define how a website's view changes in different screens like desktops, tablets, and mobile devices.

CSS is not a programming language, like C++ or JavaScript. However, CSS is not as easy as it seems. If you try to use it without understanding, you will have difficulties in web development. Therefore, learning CSS is as important as learning a programming language.

Let's continue with an example to reveal the effect of CSS on a website. Below you see how the Facebook Webpage looks as usual (with CSS):



And here you see how Facebook looks without CSS:



As we can see, CSS makes a website's visual presentation much better.

How to Add CSS in HTML?

Now you've learned why and how to define a selector and write some CSS code. But that's not enough. We also need to add CSS inside HTML otherwise it doesn't recognize the changes.

We can add CSS in an HTML file in 3 different ways:

1. External CSS File:

Keeping CSS code in a separate file is best practice. In real programming world, we need to keep HTML, CSS and JavaScript code in separate files and later import them where necessary.

We can create a separate CSS file with an extension of `.css` and include it to HTML. For example, we can create a CSS file like this one: `index.css`

Inside `index.css`, we can write our CSS code:

```
p {
  color: red;
}
```

Then we can import `index.css` to HTML with a **<link>** tag like below:

```
<head>
   <link rel="stylesheet" href="index.css">
 </head><body>

 <p> I'm a Text </p>

</body>
```

So now the HTML file has the CSS code and the changes will apply to the elements.

2. Internal CSS with <style> tag:

Another way where you can write CSS code is inside a **<style>** tag in HTML. This will keep the CSS code directly inside the HTML file, rather than in a separate file.

```
<style> p {
    color: red;
  }</style><body>  <p> I'm a Text </p></body>
```

3. Inline Style:

The third way is to write CSS rules directly inside an HTML element, with the **style** attribute. In this method, we define the CSS rules directly inside the tag and don't need to create a class for it.

```
<p style="color: blue; font-size: 22px;"> I'm a Text </p>
```

JavaScript

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.). There are also more advanced server-side versions of JavaScript such as Node.js, which allow you to add more functionality to a website than downloading files (such as realtime collaboration between multiple computers). Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects; for example:

- Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.

- Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

This means that in the browser, JavaScript can change the way the webpage (DOM) looks. And, likewise, Node.js JavaScript on the server can respond to custom requests from code written in the browser.

Uses of JavaScript

1. Adding interactive behaviour to web pages

JavaScript allows users to interact with web pages. There are almost no limits to the things you can do with JavaScript on a web page – these are just a few examples:

- Show or hide more information with the click of a button
- Change the color of a button when the mouse hovers over it
- Slide through a carousel of images on the homepage
- Zooming in or zooming out on an image
- Displaying a timer or count-down on a website
- Playing audio and video in a web page
- Displaying animations
- Using a drop-down hamburger menu

2. Creating web and mobile apps

Developers can use various JavaScript frameworks for developing and building web and mobile apps. JavaScript frameworks are collections of JavaScript code libraries that provide developers with pre-written code to use for routine programming features and tasks—literally a framework to build websites or web applications around.

Popular JavaScript front-end frameworks include React, React Native, Angular, and Vue. Many companies use Node.js, a JavaScript runtime environment built on Google Chrome's JavaScript V8 engine. A few famous examples include Paypal, LinkedIn, Netflix, and Uber!

3. Building web servers and developing server applications

Beyond websites and apps, developers can also use JavaScript to build simple web servers and develop the back-end infrastructure using Node.js.

4. Game development

Of course, you can also use JavaScript to create browser games. These are a great way for beginning developers to practice their JavaScript skills.

Facts About JavaScript

1. Along with HTML and CSS, JavaScript is one of the three main things of the www (World Wide Web). It enables interactive web pages and thus is an essential part of web applications. A majority of websites use it and all major web browsers have a devoted JavaScript engine to execute it.
2. JavaScript is single threaded. This is the reason lots of people who use multi-threaded programming thinks its working is slow as it would not be able to make use of all the cores of the CPU properly.
3. Despite the fact that there are similarities between JavaScript and Java, including language name, respective standard libraries and syntax, these two languages are distinct and differ significantly in design.
4. Like all other scripting languages, arrays and objects can be created with a brief shortcut syntax. These literals structure the basis of JSON data format.

5. JavaScript supports regular expressions in a manner similar to Perl, which provides a concise and powerful syntax for text manipulation that is more sophisticated than the built-in string functions.
6. There is a CSRF attack known as “JavaScript hijacking” in which a tag on an attacker’s site damages a page on the victim’s site that returns private information such as JavaScript or JSON.

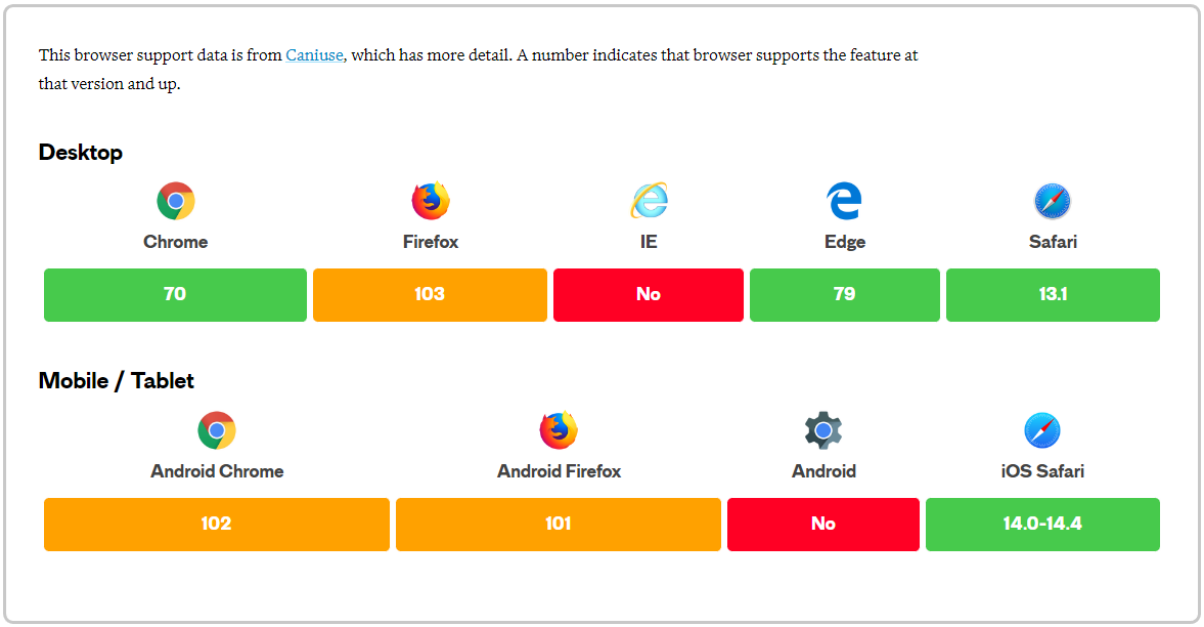
Picture-In-Picture Web API

Picture-in-Picture made its first appearance on the web in the Safari browser with the release of macOS Sierra in 2016. It made it possible for a user to pop a video out into a small floating window that stays above all others, so that they can keep watching while doing other things. It’s an idea that came from TV, where, for example, you might want to keep watching your Popular Sporting Event even as you browse the guide or even other channels.

Not long after that, Android 8.0 was released which included picture-in-picture support via native APIs. Chrome for Android was able to play videos in picture-in-picture mode through this API even though its desktop counterpart was unable to do so.

This lead to the drafting of a standard Picture-in-Picture Web API which makes it possible for websites to initiate and control this behavior.

At the time of writing, only Chrome (version 70+) and Edge (version 76+) support this feature. Firefox, Safari, and Opera all use proprietary APIs for their implementations.



Screen Capture API

The Screen Capture API introduces additions to the existing Media Capture and Streams API to let the user select a screen or portion of a screen (such as a window) to capture as a media stream. This stream can then be recorded or shared with others over the network.

Screen Capture API concepts and usage

The Screen Capture API is relatively simple to use. Its sole method is MediaDevices.getDisplayMedia(), whose job is to ask the user to select a screen or portion of a screen to capture in the form of a MediaStream.

To start capturing video from the screen, you call getDisplayMedia() on the instance of Media navigator.mediaDevices:

```
captureStream = awaitnavigator.mediaDevices.getDisplayMedia(displayMediaOptions);
```

Copy to Clipboard

The Promise returned by getDisplayMedia() resolves to a MediaStream which streams the captured media.

See the article Using the Screen Capture API for a more in-depth look at how to use the API to capture screen contents as a stream.

Additions to existing interfaces

The Screen Capture API doesn't have any interfaces of its own; instead, it adds one method to the existing MediaDevices interface.

MediaDevices interface

MediaDevices.getDisplayMedia()

The getDisplayMedia() method is added to the MediaDevices interface. Similar to getUserMedia(), this method creates a promise that resolves with a MediaStream containing the display area selected by the user, in a format that matches the specified options.

Additions to existing dictionaries

The Screen Capture API adds properties to the following dictionaries defined by other specifications.

MediaTrackConstraints

MediaTrackConstraints.cursor

A ConstrainDOMString indicating whether or not the cursor should be included in the captured display surface's stream, and if it should always be visible or if it should only be visible while the mouse is in motion.

MediaTrackConstraints.displaySurface

A ConstrainDOMString indicating what type of display surface is to be captured. The value is one of application, browser, monitor, or window.

MediaTrackConstraints.logicalSurface

Indicates whether or not the video in the stream represents a logical display surface (that is, one which may not be entirely visible onscreen, or may be completely offscreen). A value of true indicates a logical display surface is to be captured.

MediaTrackSettings

MediaTrackSettings.cursor

A string which indicates whether or not the display surface currently being captured includes the mouse cursor, and if so, whether it's only visible while the mouse is in motion or if it's always visible. The value is one of always, motion, or never.

MediaTrackSettings.displaySurface

A string indicating what type of display surface is currently being captured. The value is one of application, browser, monitor, or window.

MediaTrackSettings.logicalSurface

A Boolean value which is true if the video being captured doesn't directly correspond to a single onscreen display area.

Testing and Implementation

TESTING

Testing is the process of exercising software with the intent of finding errors and ultimately correcting them. The following testing techniques have been used to make this project free of errors.

Content Review

The whole content of the project has been reviewed thoroughly to uncover typographical errors, grammatical error and ambiguous sentences.

Navigation Errors

Different users were allowed to navigate through the project to uncover the navigation errors. The views of the user regarding the navigation flexibility and user friendliness were taken into account and implemented in the project.

Unit Testing

Focuses on individual software units, groups of related units.

- Unit – smallest testable piece of software.

- A unit can be compiled /assembled / linked/loaded; and put under a test harness.

- Unit testing done to show that the unit does not satisfy the application and /or its implemented software does not match the intended designed structure.

Integration Testing

Focuses on combining units to evaluate the interaction among them

- Integration is the process of aggregating components to create larger components.

• Integration testing done to show that even though components were individually satisfactory, the combination is incorrect and inconsistent.

System testing

Focuses on a complete integrated system to evaluate compliance with specified requirements (test characteristics that are only present when entire system is run)

• A system is a big component.

• System testing is aimed at revealing bugs that cannot be attributed to a component as such, to inconsistencies between components or planned interactions between components.

• Concern: issues, behaviours that can only be exposed by testing the entire integrated system (e.g., performance, security, recovery)each form encapsulates (labels, texts, grid etc.). Hence in case of project in V.B. form are the basic units. Each form is tested thoroughly in term of calculation, display etc.

Regression Testing

Each time a new form is added to the project the whole project is tested thoroughly to rectify any side effects. That might have occurred due to the addition of the new form. Thus regression testing has been performed.

White-Box testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing and structural testing) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and

determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Techniques used in white-box testing include:

API testing (application programming interface) – testing of the application using public and private APIs.

Code coverage – creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once).

Fault injection methods – intentionally introducing faults to gauge the efficacy of testing strategies.

Code coverage tools can evaluate the completeness of a test suite that was created with any method, including black-box testing. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. Code coverage as a software metric can be reported as a percentage for:

Function coverage, which reports on functions executed Statement coverage, which reports on the number of lines executed to complete the test 100% statement coverage ensures that all code paths, or branches (in terms of control flow) are executed at least once. This is helpful in ensuring correct functionality, but not sufficient since the same code may process different inputs correctly or incorrectly.

Black-box testing

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The tester is only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing. Specification-based testing aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional.

Specification-based testing may be necessary to assure correct functionality, but it is insufficient to guard against complex or high-risk situations.

One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester likely has a different set and may emphasize different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

This method of test can be applied to all levels of software testing: unit, integration, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

Alpha Testing

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

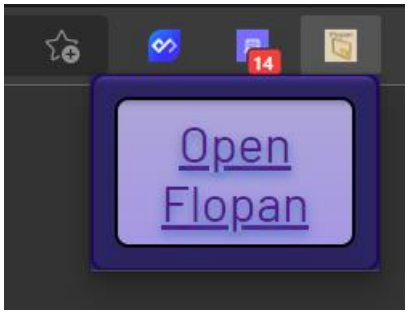
Beta Testing

Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

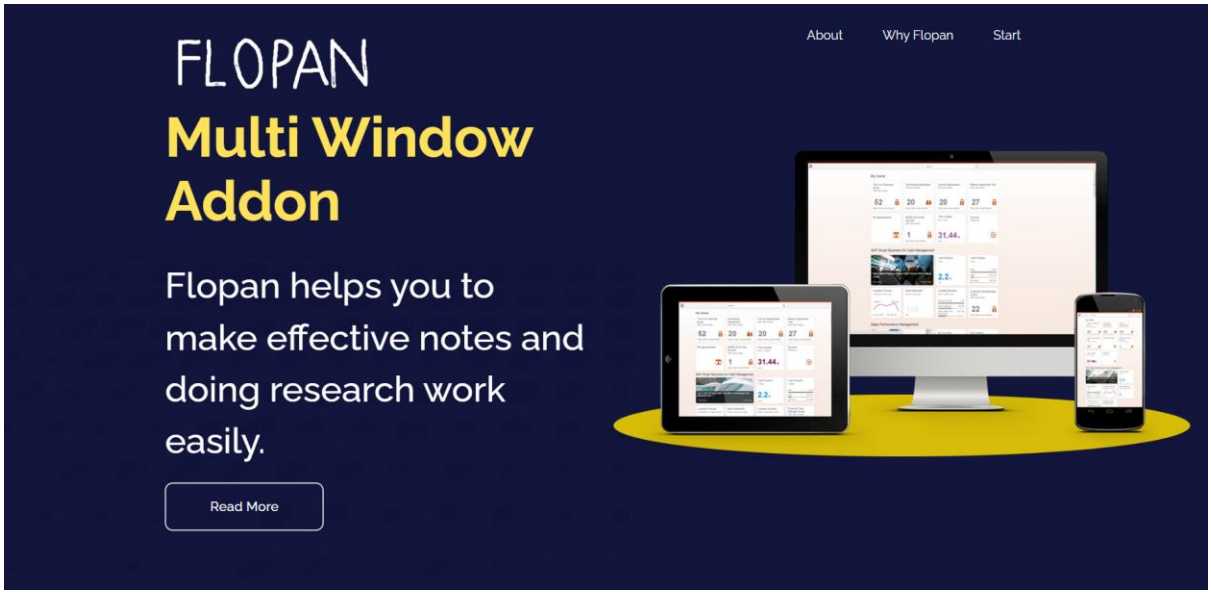
Implementation

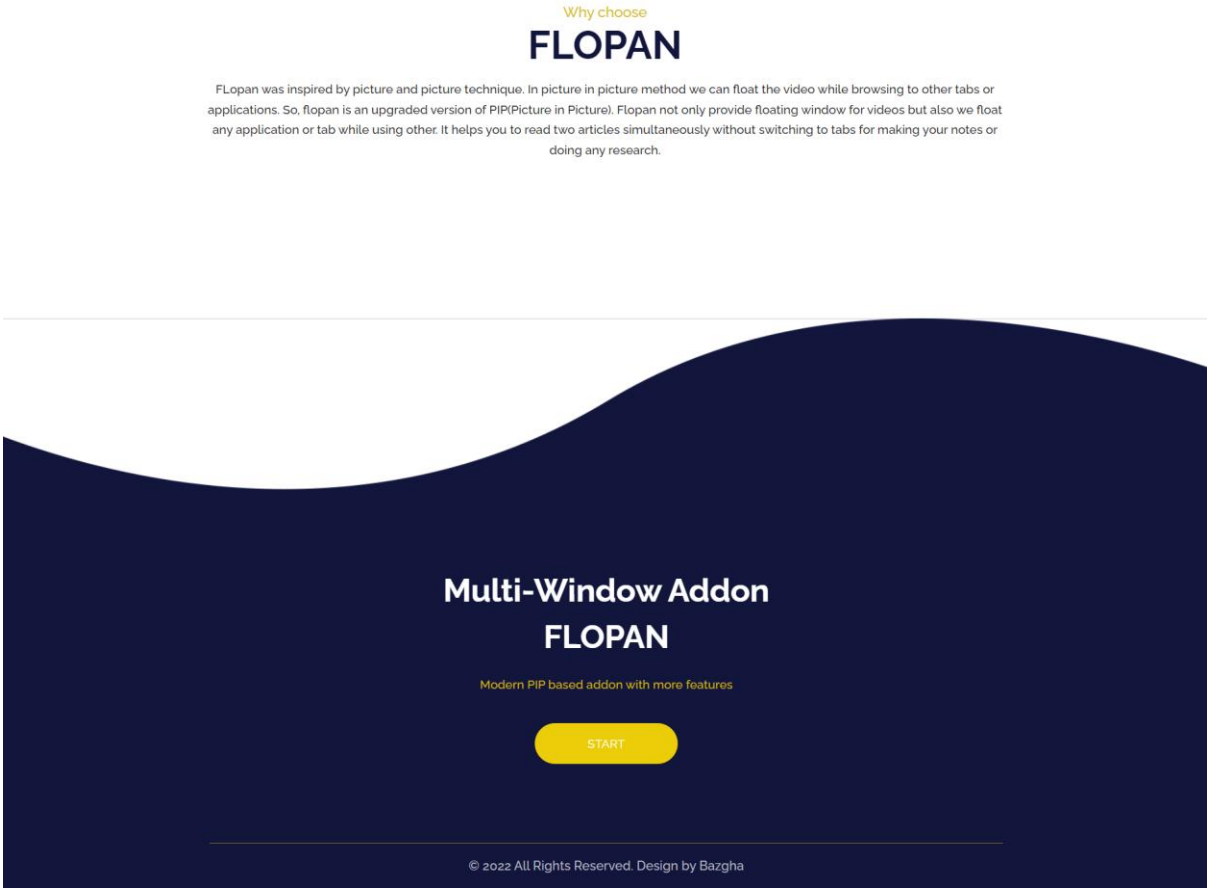
FLOPAN

Popup Page

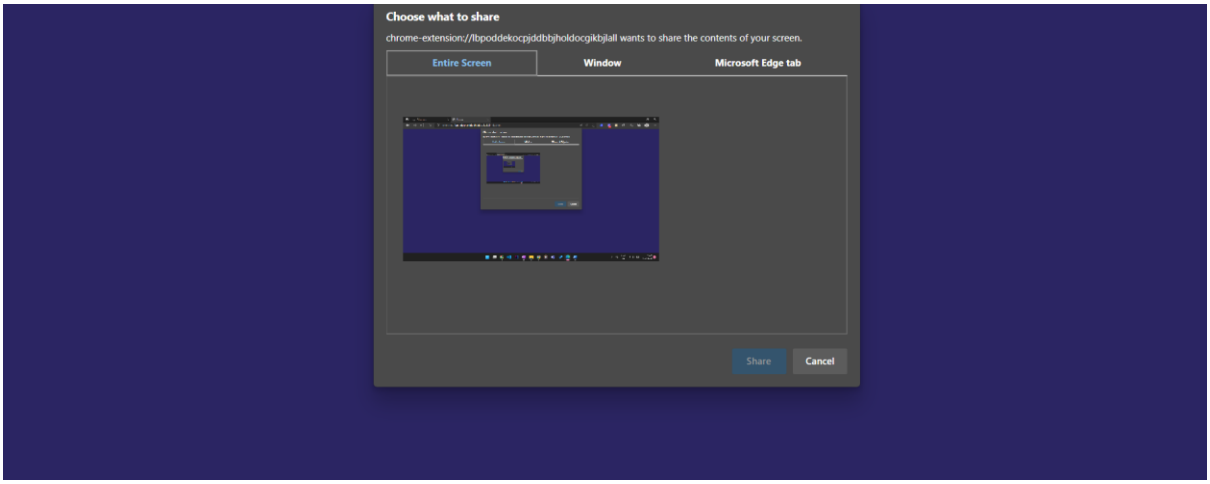


Flopan Page





Screen Select Option



How to use this addon

Fork the repository

Fork this repository by clicking on the fork button on the top of this page. It will create a copy of this project in your github account.

Clone the repository

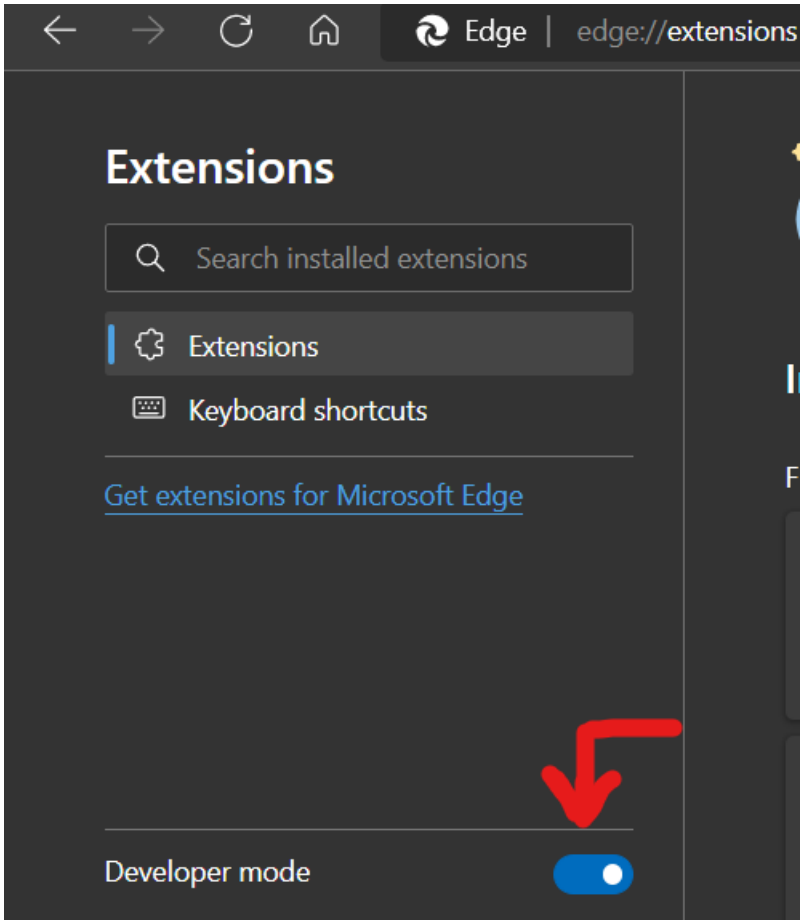
Open the forked repository, click on the code button and copy the url. Now, open the terminal and run the command

https://github.com/Bazgha19/Flopan-Addon.git

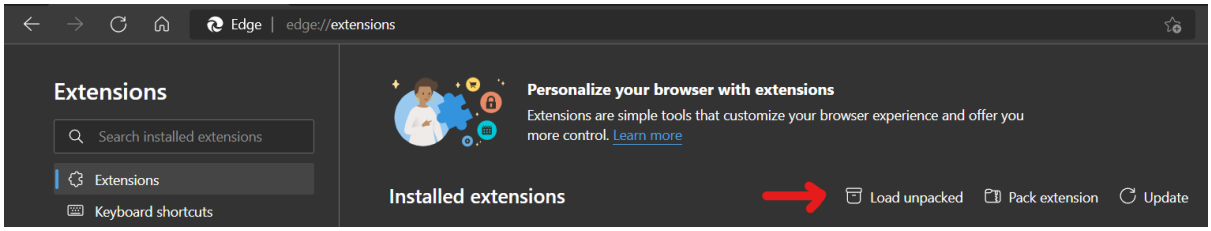
Developer Mode

-> Go to: edge://extensions/

-> Enable developer mode from the bottom left side.



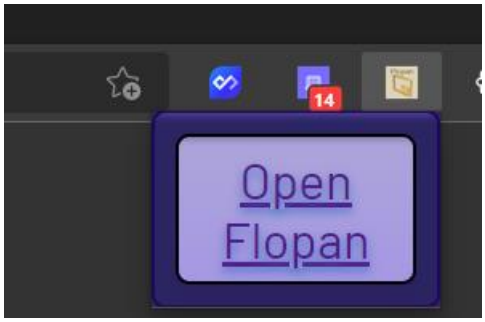
-> Then click on Load Unpacked option and select the folder of your addon.



Done

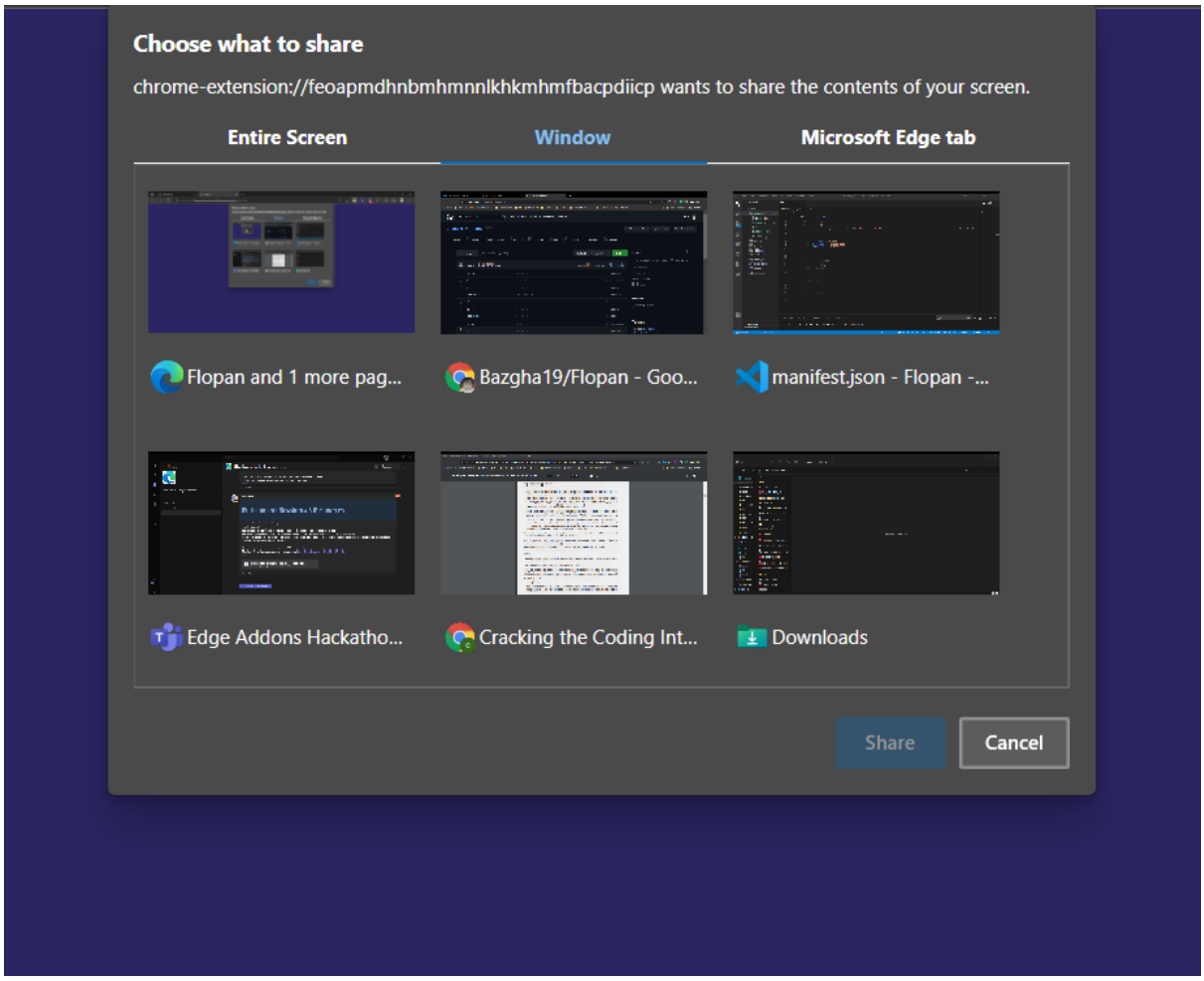
-> It's all set, now you're ready to use this extension by clicking on the icon.

-> After clicking on the icon click on start and run the extension.

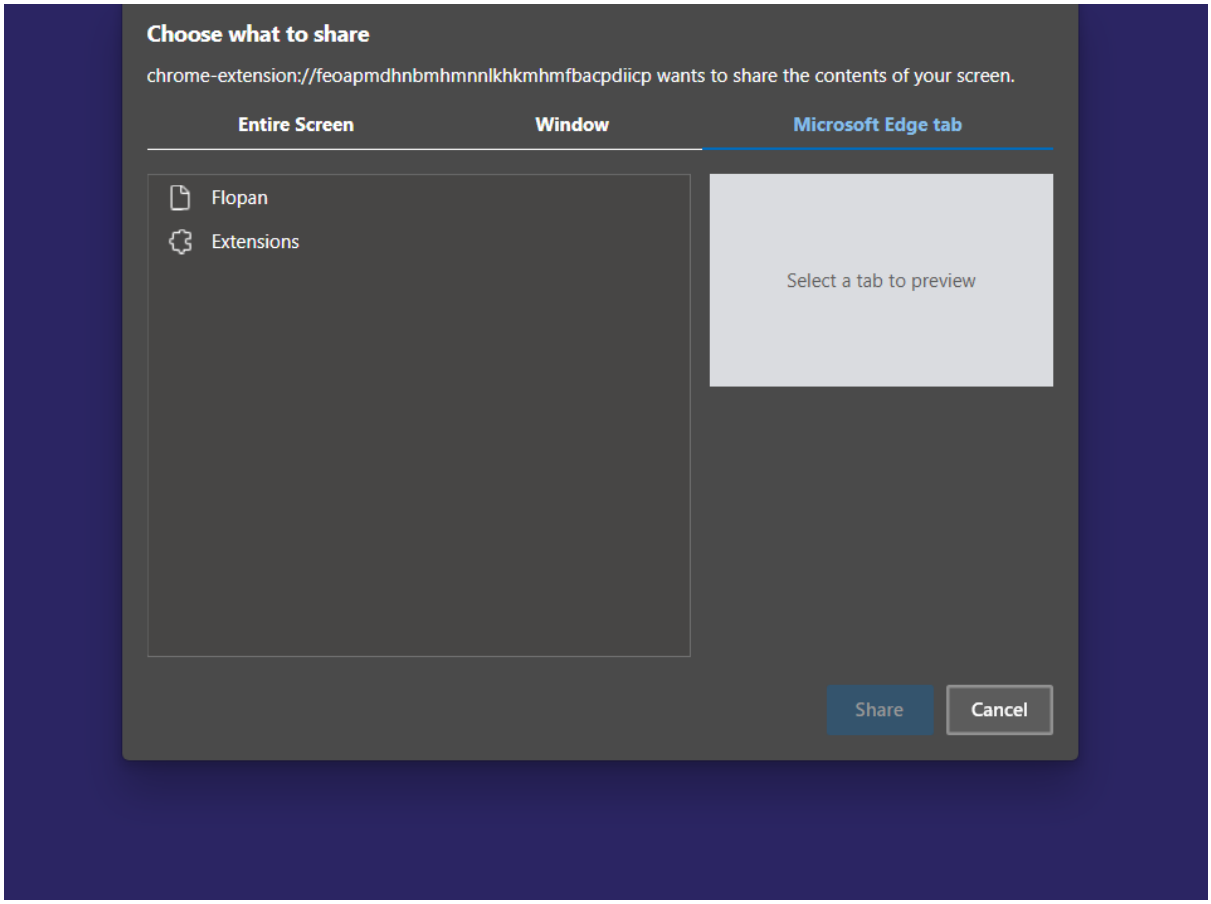


-> Then select an entire screen or window or any tab you want to run simultaneously. Entire Screen:

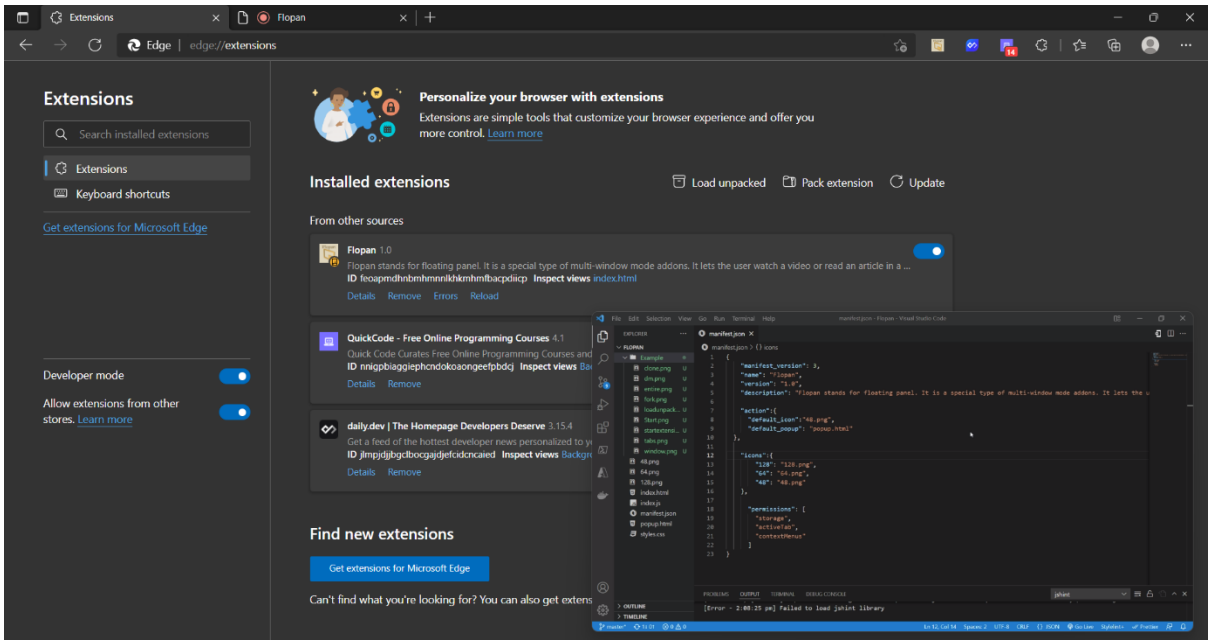
Window:



Microsoft Edge Tabs:



Enjoy Flopan!!



Learning's and Value Addition

A project is a learning experience of its own kind. Neither is it spoon-fed school learning, nor pressure filled workload. It is in between; I not only learn the basics of work life but also the skills required for a brighter professional career.

During my project I had learned lots of thing and few of them I'm mentioning here:

Teamwork: It is important because it enables your team to share ideas and responsibilities, which helps reduce stress on everyone, allowing them to be meticulous and thorough when completing tasks.

Problem Solving Skills: It allow you to find candidates who are cognitively equipped to handle anything their jobs throw at them. Problem solvers can observe, judge, and act quickly when difficulties arise when they inevitably do.

Work Ethics: Workplace ethics ensures positive ambience at the workplace. Workplace ethics leads to happy and satisfied employees who enjoy coming to work rather than treating it as a mere source of burden.

Adaptability Skills: It expands your capacity to handle change, no matter how serious it might be. Instead of throwing away your energy trying to change your circumstance, you will change yourself right from within, thus making you thrive in whatever situation you find yourself.

Communication Skills: It is fundamental to the existence and survival of humans as well as to an organization. It is a process of creating and sharing ideas, information, views, facts, feelings, etc. among the people to reach a common understanding.

Responsibility: Each step we take towards being responsible and productive helps to raise our self-esteem and our relationships with friends, family and co-workers improve ten-fold.

Time Management: It helps you prioritize your tasks so that you ensure you have enough time available to complete every project. The quality of your work increases when you're not rushing to complete it ahead of a fast-approaching deadline.

Theoretical v/s Practical Knowledge

Practical knowledge is knowledge that is acquired by day-to-day hands-on experiences. In other words, practical knowledge is gained through doing things; it is very much based on real-life endeavours and tasks. On the other hand, theoretical knowledge teaches the reasoning, techniques, and theory of knowledge. While practical knowledge is gained by doing things, theoretical knowledge is gained, for example, by reading a manual.

While theoretical knowledge may guarantee that you understand the fundamental concepts and have know-how about how something works and its mechanism, it will only get you so far, as, without practice, one is not able to perform the activity as well as he could.

During this project period, I had only theoretical knowledge about some programming language never build any thing before. But during

this project period mentor assigned me a task for Flopan and then I try to make that first I made mock-up and yes, it is my first mock up I had made ever. Then some research is important to work practically on a project. So, I did that and share that with Saurav and get some review. He told me some of my mistakes and then I correct them accordingly because this flopan addon is used by many people, so some feedback is important, and I really get some positive feedback as well as some negative from him and all that are honest. So, on the negative feedback, I research again for that and end up with good user experience. By doing this I got lots of practical knowledge with theoretical knowledge.

Theoretical and practical knowledge are interconnected and complement each other — if one knows exactly HOW to do something, one must be able to apply these skills and therefore succeed in practical knowledge.

CONCLUSION

The project “FLOPAN” aims to simplify making the notes and to doing research works.

It eases the work of student’s life. It allows you to watch videos/read any article in a floating window (always on top of other windows) so you can keep an eye on what you’re watching while interacting with other sites, or applications.

It lets the user use an application in a small window pinned to a corner of the screen while navigating between apps or browsing content on the main screen. We can also resize the window and move according to our requirements. If you want to close the window, then there is also a close button in the floating window.

Bibliography

- Download an IDE:
  VS Code: https://visualstudio.microsoft.com/downloads/

- For learning HTML:
  https://developer.mozilla.org/en-US/docs/Web/HTML
  https://www.w3schools.com/TAgs/default.asp
  https://docs.microsoft.com/en-us/cpp/mfc/html-basics?view=msvc-170

- For learning CSS:
  https://developer.mozilla.org/en-US/docs/Web/CSS
  https://devdocs.io/css/
  https://www.w3schools.com/cssref/

- For learning JavaScript:
  https://www.javascript.com/
  https://www.w3schools.com/js/
  https://developer.mozilla.org/en-US/docs/Web/JavaScript
  https://github.com/rwaldron/idiomatic.js

- For learning Picture-in-Picture Web API:
  https://css-tricks.com/an-introduction-to-the-picture-in-picture-web-api/
  https://developer.mozilla.org/en-US/docs/Web/API/Picture-inPicture_API

- For Screen Capture Web API:
  https://developer.mozilla.org/en-US/docs/Web/API/Screen_Capture_API