INDUSTRIAL TRAINING

REPORT ON

CAM-GALLERY

A PROJECT IN WEB DEVELOPMENT

Submitted in partial fulfilment of the

Requirements for the award of the degree

of

Bachelor of Technology

In

Information Technology

By:

RUHEE JAIN

(00613203119)

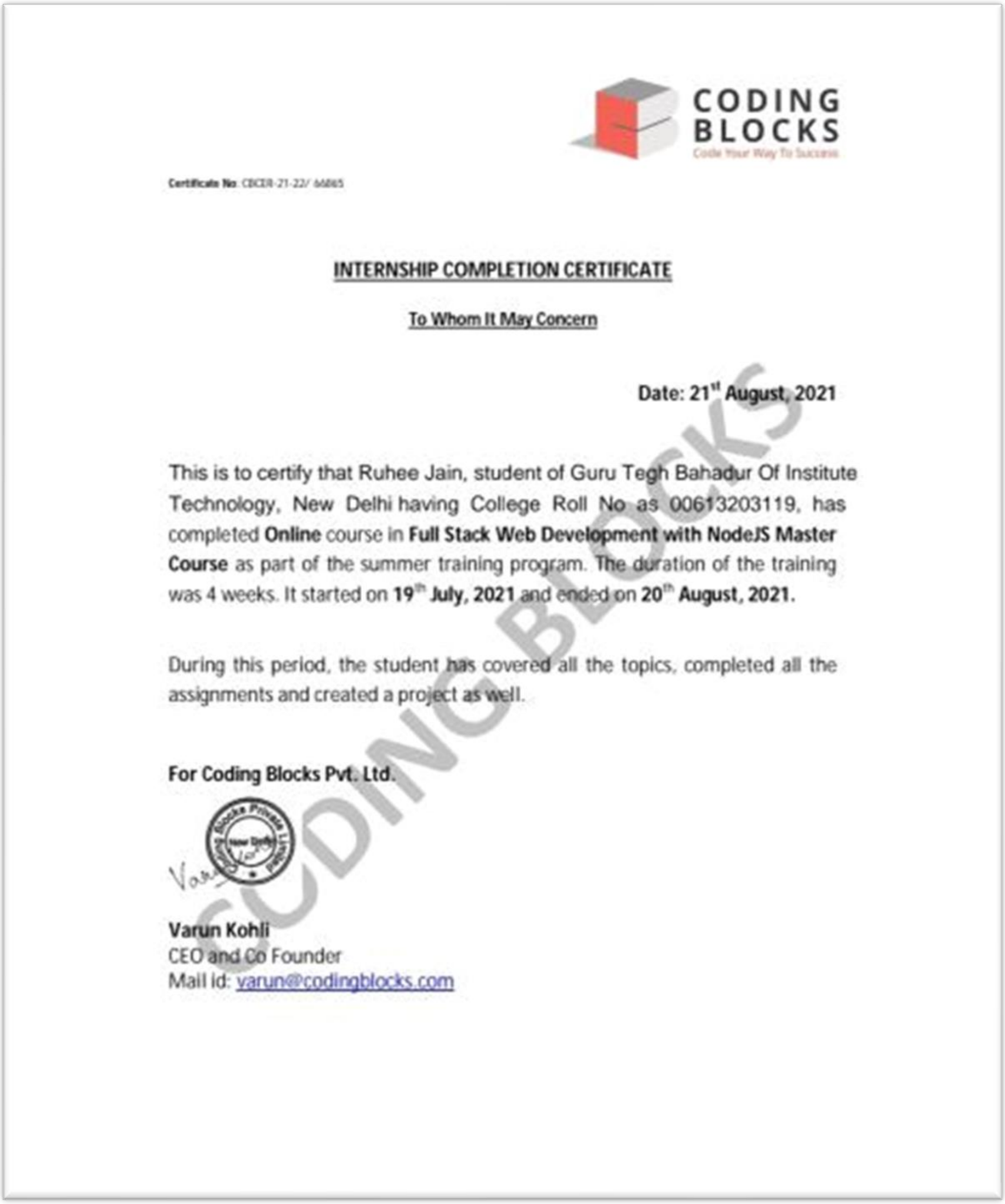
Department of Information Technology

Guru Tegh Bahadur Institute of Technology

Guru Gobind Singh Indraprastha University, Dwarka, New Delhi

BATCH 2019-2023

CERTIFICATE



CODING
BLOCKS
Code Your Way To Success

Certificate No: CBCER-21-22/ 66865

INTERNSHIP COMPLETION CERTIFICATE

To Whom It May Concern

Date: 21st August, 2021

This is to certify that Ruhee Jain, student of Guru Tegh Bahadur Of Institute Technology, New Delhi having College Roll No as 00613203119, has completed **Online** course in **Full Stack Web Development with NodeJS Master Course** as part of the summer training program. The duration of the training was 4 weeks. It started on **19th July, 2021** and ended on **20th August, 2021.**

During this period, the student has covered all the topics, completed all the assignments and created a project as well.

For Coding Blocks Pvt. Ltd.

Coding Blocks Private Limited
New Delhi

Varun Kohli
CEO and Co Founder
Mail id: varun@codingblocks.com

DECLARATION

I hereby declare that all the work presented in this project report entitled "**CAM-GALLERY**" in the partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Information Technology, Guru Tegh Bahadur Institute of Technology, Guru Govind Singh Indraprastha University, New Delhi** is an authentic record of my own work carried out during the period of my industrial training.

The work reported in this has not been submitted by me for award of any other degree of this or any other institute.

DATE: 15.01.2022

RUHEE JAIN
(00613203119)
ruheej1@gmail.com

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Mr. Arnav Gupta sir**, **Teaching Associate, Coding Blocks**, for his valuable guidance and support throughout my training. The summer training helped me in increasing my knowledge and skills in Web Development.

I would also like to express my great gratitude towards **Mr. P.S. Bedi sir (H.O.D IT Department)** and the **teachers in-charge**, **Ms. Bhavneet Kaur ma'am and Ms. Debleena Mitra ma'am** who have given indispensable support and suggestions towards my project. Without their help and guidance, I would not have been able to present this project up to the present standard. I also take this opportunity to give thanks to all my teachers who gave support for the project or in other aspects of my study at Guru Tegh Bahadur Institute of Technology.

DATE: 15.01.2022

RUHEE JAIN

(00613203119)

ruheej1@gmail.com

ABSTRACT

Web programming, also known as web development, is the creation of dynamic web applications. It is the work involved in developing a website for the Internet. Web development can range from developing a simple single static page of plain text to complex web applications

There are two broad divisions of web development – front-end development (also called client-side development) and back-end development (also called server-side development).

The project: Cam-Gallery is just like a smartphone’s Camera. It is capable of capturing photos as well as videos, and not only that but it could even store them and show a gallery view, from where the photos and videos could be deleted or saved locally! And to add to the greatness, it also supports some color filters which one can apply and take photos or videos!

I created it using HTML, CSS And JavaScript and with the help of Browser Web APIs like Mediastream, Navigator, MediaDevices, MediaRecorder, Canvas and for storage I used IndexedDB

During building this project, I tried to implement all that I have learnt in web development, specifically, front-end web development.

CONTENTS TABLE

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e., websites.

Web development can range from developing a simple single static page of plain text to complex web applications

The word Web Development is made up of two words, that is:

- Web: It refers to websites, web pages or anything that works over the internet.
- Development: Building the application from scratch.

1.1 Classification

Web Development can be classified into two ways:

- Frontend Development
- Backend Development

1.2 Frontend Development

The part of a website that the user interacts directly is termed as front end. It is also referred to as the ‘client side’ of the application.

Front-end development refers to constructing what a user sees when they load a web application – the content, design and how you interact with it.
The basic toolset for front-end is well-defined, It includes HTML, CSS AND JavaScript

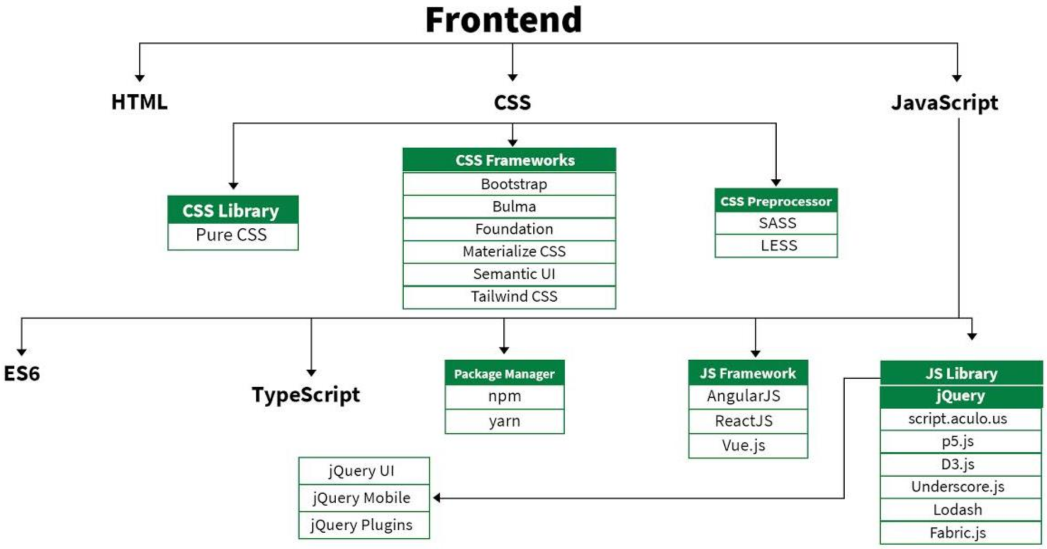


Figure 1: Front-end development

[Ref: https://www.geeksforgeeks.org/web-development/]

- **HTML**: HTML stands for HyperText Markup Language. It is used to design the front-end portion of web pages using markup language. It acts as a skeleton for a website since it is used to make the structure of a website.
- **CSS:** Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. It is used to style our website.

- **JavaScript:** JavaScript is a scripting language used to provide a dynamic behaviour to our website.
- **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular CSS framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones).

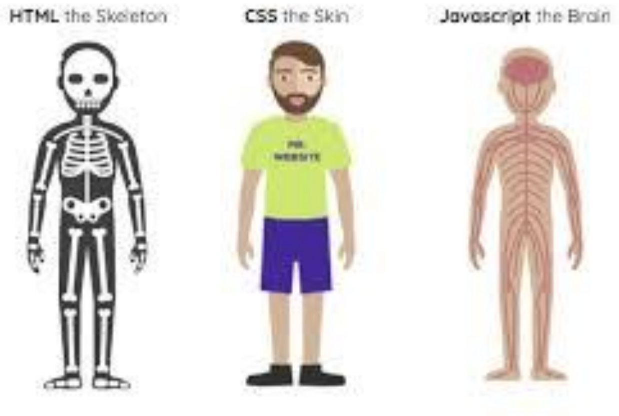


Figure 2: Components of front-end development

[Ref: https://images.app.goo.gl/49VLuCH1zd1mLQX4A]

1.3 Back-end web development

Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

It contains behind-the-scenes activities that occur when performing any action on a website. There are a wide range of languages that could be used for Backend Web Development, few of them include JavaScript, Python, PHP, Java etc.

Nodejs is a popular JavaScript framework for Backend Development

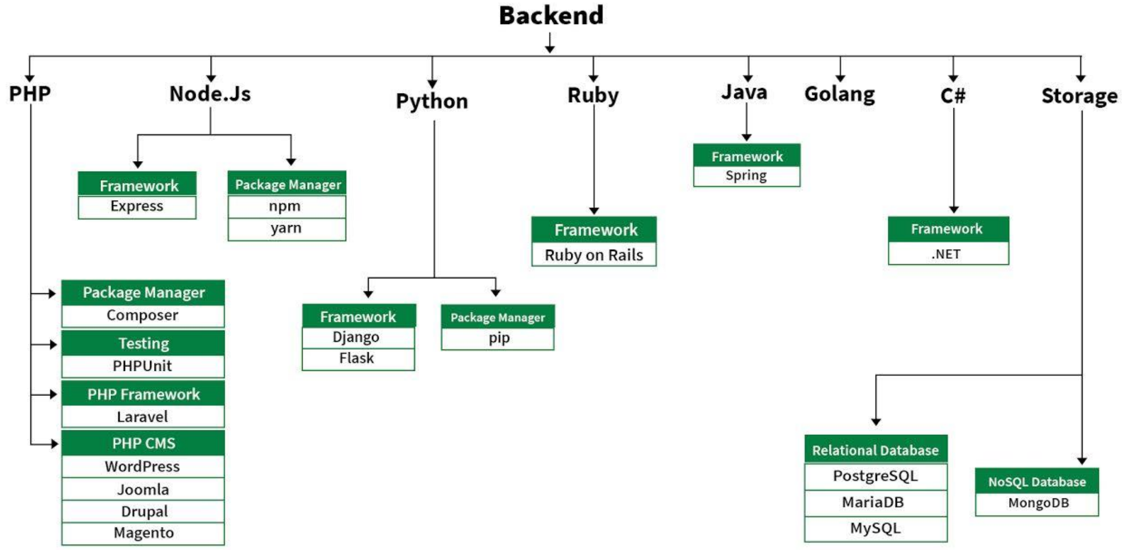


Figure 3: Back-end development

[Ref: https://www.geeksforgeeks.org/web-development/]

- **PHP:** PHP is a server-side scripting language designed specifically for web development.
- **Java:** Java is one of the most popular and widely used programming language. It is highly scalable.
- **Python:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside a browser.
- **Back End Frameworks:** The list of back end frameworks are: Express, Django, Rails, Laravel, Spring, etc.

1.4 WEB APIs

Application Programming Interfaces (APIs) are constructs made available in programming languages to allow developers to create complex functionality more easily. They abstract more complex code away from you, providing some easier syntax to use in its place.

Client-side JavaScript, in particular, has many APIs available to it — these are not part of the JavaScript language itself, rather they are built on top of the core JavaScript language

They generally fall into two categories:

- **Browser APIs -** Browser APIs are built into your web browser and are able to expose data from the browser and surrounding computer environment and do useful complex things with it. For example, the Web Audio API provides JavaScript constructs for manipulating audio in the browser — taking an audio track, altering its volume, applying effects to it, etc.

- **Third party APIs -** Third-party APIs are not built into the browser by default, and you generally have to retrieve their code and information from somewhere on the Web. For example, the Twitter API allows you to do things like displaying your latest tweets on your website. It provides a special set of constructs you can use to query the Twitter service and return specific information

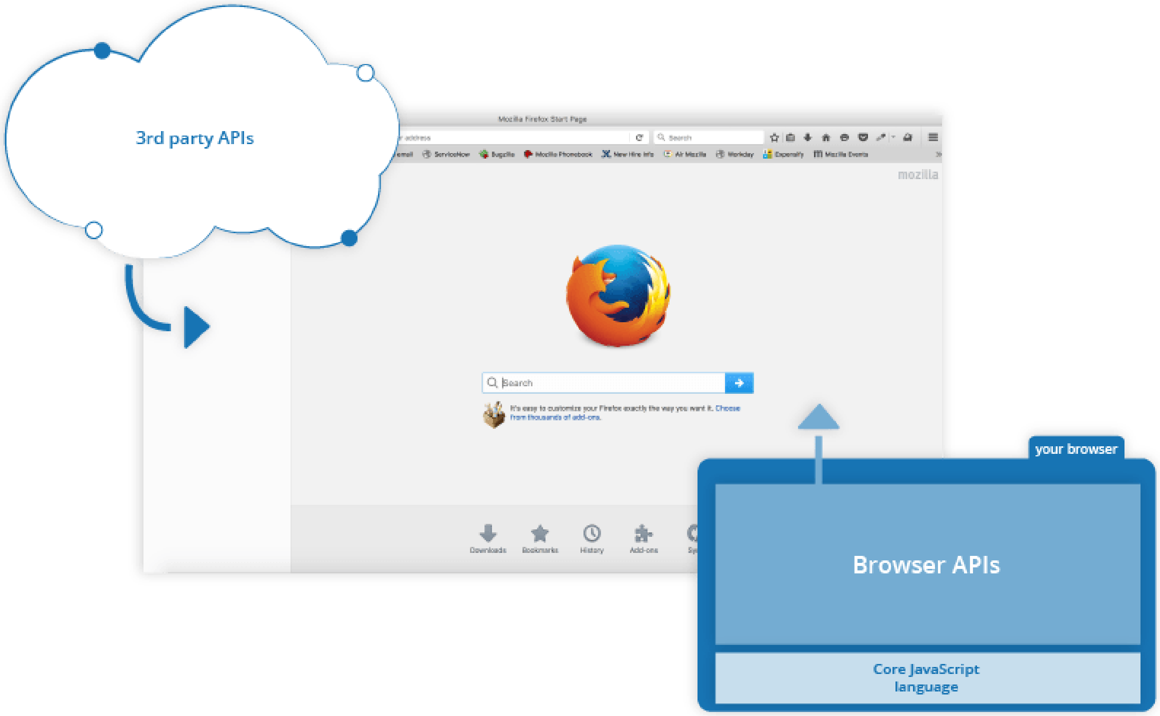


Figure 4: Web APIs

[Ref: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction]

1.5 Web Hosting

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet. A web host, or web hosting service provider, is a business that provides the technologies and services needed for the website or webpage to be viewed in the Internet. Websites are hosted, or stored, on special computers called servers. When Internet users want to view your website, all they need to do is type your website address or domain into their browser. Their computer will then connect to your server and your webpages will be delivered to them through the browser.

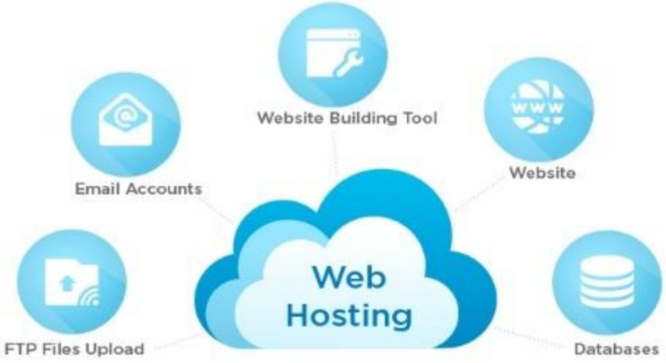


Figure 5: Web hosting

[Ref: https://computerhindinotes.com/wp-content/uploads/2019/11/web-hosting.jpg]

CHAPTER 2: ABOUT THE PROJECT

Cam-Gallery is a website just like a smartphone's Camera, capable of capturing photos as well as videos, and not only that but it could even store them and show a gallery view, from where the photos and videos could be deleted or saved locally.

To add to the greatness, it also supports some color filters which one can apply and take photos or videos!

The gallery view similar to the gallery app of our phones, allows one to view all the clicked photos and recorded videos on a bigger screen view and the download option saves the photo or video in high quality directly to one's local setup.

2.1 WHY THIS PROJECT?

In a world of increasing online presence, the need of more and more new applications and different platforms is on the rise but I wanted to create something useful yet familiar, and what's more useful than a video conferencing platform in these tough times of coronavirus? But then I thought of all the gazillion meeting platforms already in town and the concept of self-love and care, the need to connect virtually with people, the constant desire of humans to be heard and appreciated, to look good and interact with others, all these ideas which have been doing the rounds now made me realise that one needs a camera application on a bigger scale. And who wouldn't want to look perfect just before an important interview and have a look right at the computer screen? And since everything now is getting better and grander, then why not have a web camera application? So, I built a fully functional camera and gallery app just like in our smartphones but this one for the web.

2.2 LANGUAGES USED, TECHSTACK INVOLVED

This project has been created using HTML, CSS And JavaScript and with the help of Browser Web APIs like Mediastream, Navigator, MediaDevices, MediaRecorder, Canvas and for storage IndexedDB was used.
In this project, the following tech stack has been used:

HTML: HTML helps to structure our website well. The way a skeleton system gives a structure to the human body in a similar manner HTML acts as a skeleton for a website, without it a website cannot be made. If you want to work as a Software Developer especially in the Web Development domain, then learning HTML is a must, because without knowledge of it you cannot build a website.

Base for creating websites: HTML is the basic necessity a developer should know while building a website from scratch.

CSS: CSS (Cascading Style Sheets) is a stylesheet language used to design the webpage to make it attractive. The reason of using CSS is to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. Styling has been an essential property for any website since many decades. It increases the standards and overall look of the website which makes it easier for the user to interact with it. A website cannot be made without CSS, as styling is MUST since no user would want to interact with a dull and shabby website.

Figure 6: Front End Web Development

[Ref: https://courses.wscubetech.com/s/store/courses/5f5338c60cf2d7d974471146/cover.jpg?v=1]

Base for web development**:** HTML and CSS is the basic skill that every web developer should know. It is the basic skill that is required for building a website.

Makes your website look attractive**:** A website that’s dull and plain will not attract the user most probably, so adding some style would surely make your website presentable to the user. Makes the design come live**:** A web developer is responsible in making the design given to him as a live product. CSS is used for styling to develop the design of the website.

Increases user experience of website**:** A website with a simple yet beautiful UI would help the users to go through the website easily. CSS is used to make the user interface better.

There are three types of CSS which are given below:

- Inline CSS
- Internal or Embedded CSS
- External CSS

JavaScript: JavaScript is the world most popular lightweight, interpreted compiled programming language. It is also known as scripting language for web pages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments.

As a newbie in this field, there were a few challenges in the way, however with proper guidance from my mentors and the research that I did on my own, I was able to clear a lot of hurdles and build this website on my own.

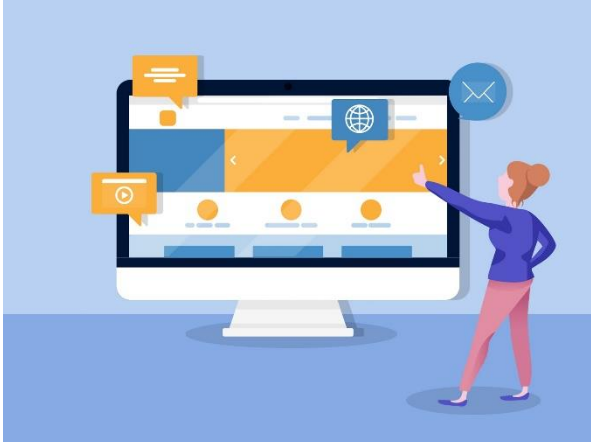


Figure 7: Web Designing

[Ref: https://99designs-blog.imgix.net/blog/wp-content/uploads/2018/09/WHAT-IS-WEB-DESIGN.jpg?auto=format&q=60&w=1860&h=1395&fit=crop&crop=faces]

2.3 SOFTWARE REQUIREMENTS

This project was built on a Windows10 operating system and the source code was written and debugged through Visual Studio Code - a 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.

The CAM-GALLERY APPLICATION is compatible to run on any kind of OS, it is a browser dependent application and not OS-specific. No specific software needs to be installed before running this project

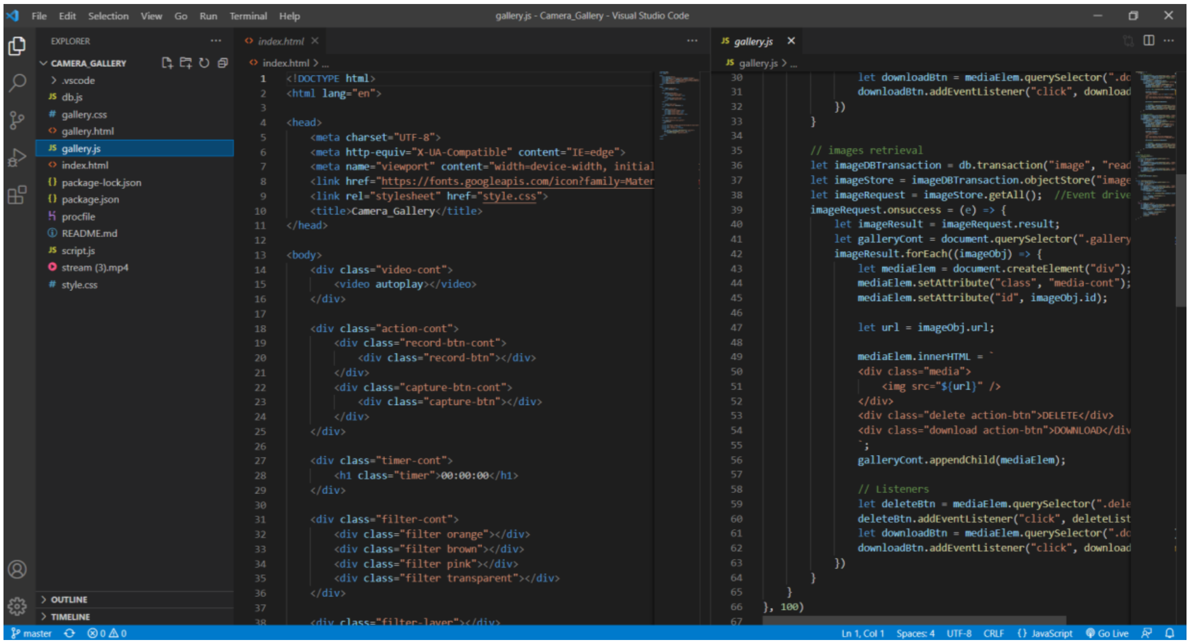
It is supported by all popular web browsers like:

- Google Chrome.
- Mozilla Firefox.
- Microsoft Edge.
- Internet Explorer.
- Safari.

CHAPTER 3: SOURCE CODE

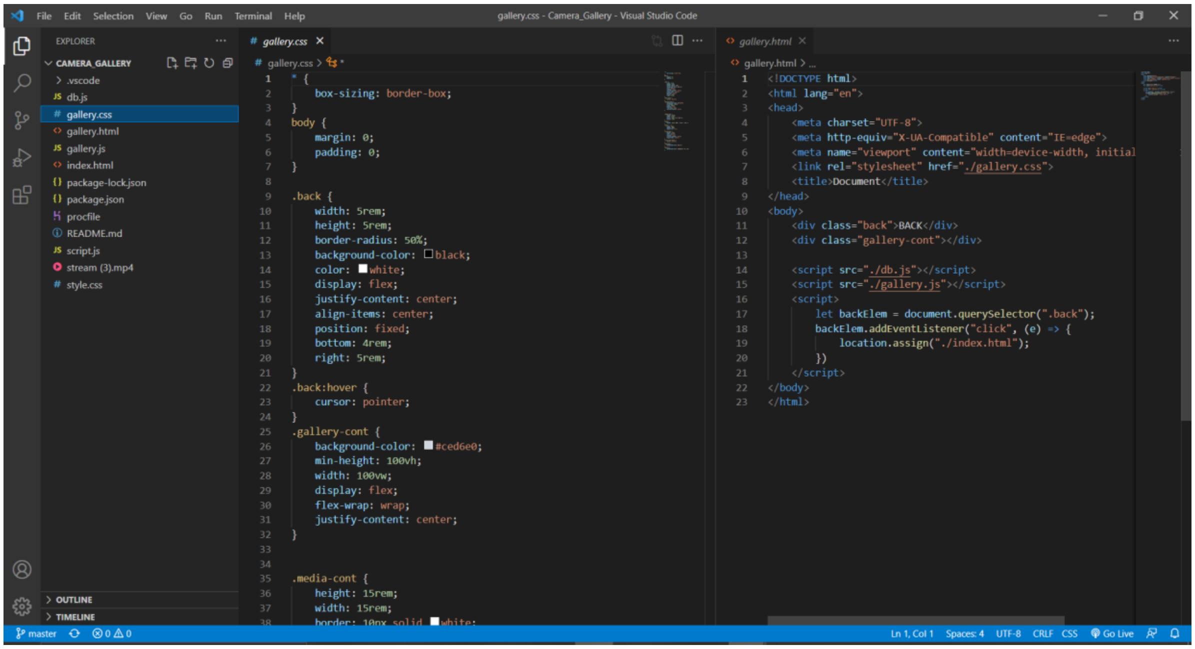
The Source has been uploaded for reference purpose here: GITHUB

SOME CODE SNIPPETS



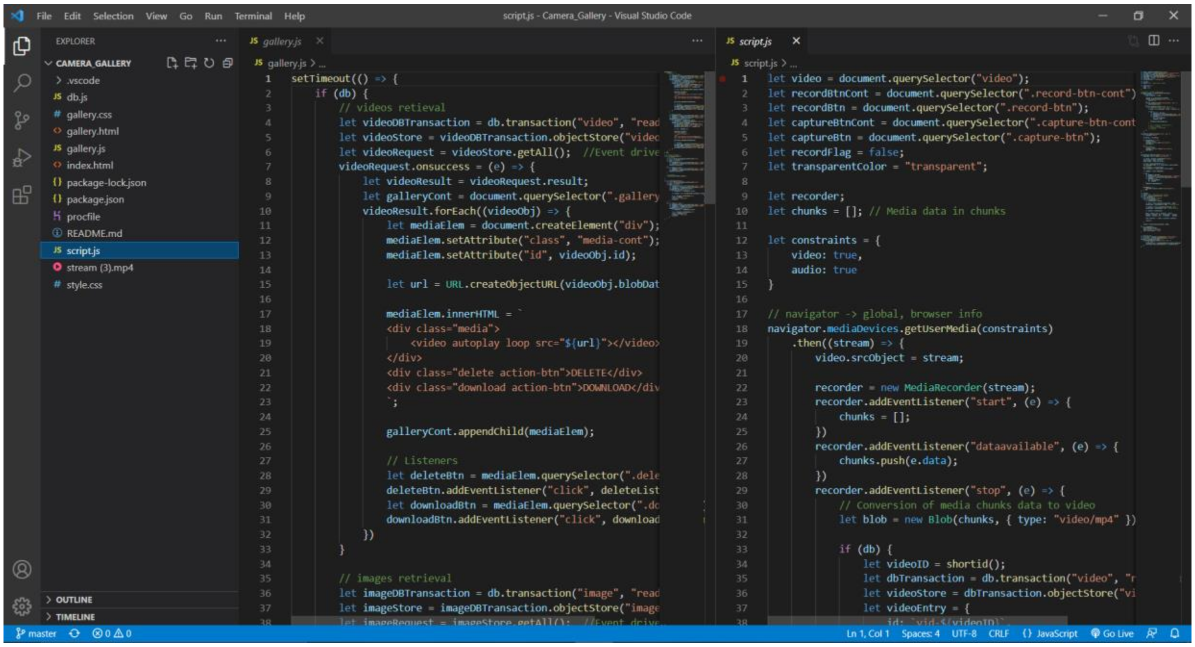
```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial
    <link href="https://fonts.googleapis.com/icon?family=Mater
    <link rel="stylesheet" href="style.css">
    <title>Camera_Gallery</title>
</head>

<body>
    <div class="video-cont">
        <video autoplay></video>
    </div>

    <div class="action-cont">
        <div class="record-btn-cont">
            <div class="record-btn"></div>
        </div>
        <div class="capture-btn-cont">
            <div class="capture-btn"></div>
        </div>
    </div>

    <div class="timer-cont">
        <h1 class="timer">00:00:00</h1>
    </div>

    <div class="filter-cont">
        <div class="filter orange"></div>
        <div class="filter brown"></div>
        <div class="filter pink"></div>
        <div class="filter transparent"></div>
    </div>

    <div class="filter-layer"></div>
```

```javascript
                let downloadBtn = mediaElem.querySelector(".do
                downloadBtn.addEventListener("click", download
            })
        }

        // images retrieval
        let imageDBTransaction = db.transaction("image", "read
        let imageStore = imageDBTransaction.objectStore("image
        let imageRequest = imageStore.getAll();  //Event drive
        imageRequest.onsuccess = (e) => {
            let imageResult = imageRequest.result;
            let galleryCont = document.querySelector(".gallery
            imageResult.forEach((imageObj) => {
                let mediaElem = document.createElement("div");
                mediaElem.setAttribute("class", "media-cont");
                mediaElem.setAttribute("id", imageObj.id);

                let url = imageObj.url;

                mediaElem.innerHTML = `
                <div class="media">
                    <img src="${url}" />
                </div>
                <div class="delete action-btn">DELETE</div>
                <div class="download action-btn">DOWNLOAD</div
                `;
                galleryCont.appendChild(mediaElem);

                // Listeners
                let deleteBtn = mediaElem.querySelector(".dele
                deleteBtn.addEventListener("click", deleteList
                let downloadBtn = mediaElem.querySelector(".do
                downloadBtn.addEventListener("click", download
            })
        }
    }
}, 100)
```



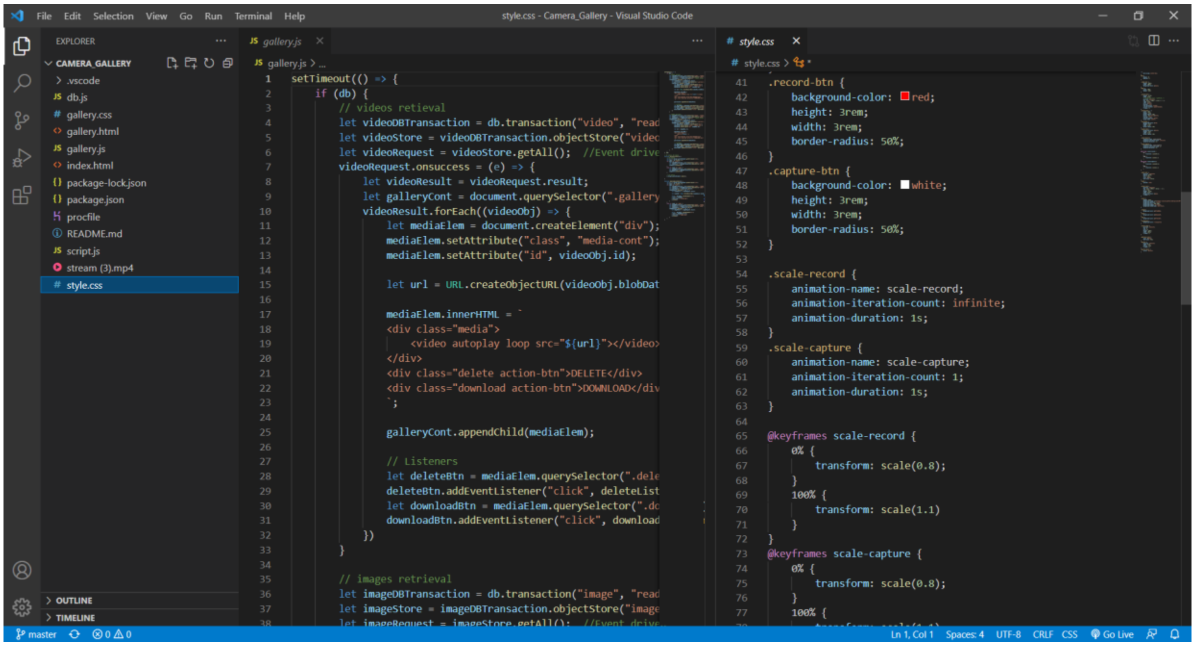
```css
* {
    box-sizing: border-box;
}
body {
    margin: 0;
    padding: 0;
}

.back {
    width: 5rem;
    height: 5rem;
    border-radius: 50%;
    background-color: black;
    color: white;
    display: flex;
    justify-content: center;
    align-items: center;
    position: fixed;
    bottom: 4rem;
    right: 5rem;
}
.back:hover {
    cursor: pointer;
}
.gallery-cont {
    background-color: #ced6e0;
    min-height: 100vh;
    width: 100vw;
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
}


.media-cont {
    height: 15rem;
    width: 15rem;
    border: 10px solid white;
```

```html
<!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
    <link rel="stylesheet" href="./gallery.css">
    <title>Document</title>
</head>
<body>
    <div class="back">BACK</div>
    <div class="gallery-cont"></div>

    <script src="./db.js"></script>
    <script src="./gallery.js"></script>
    <script>
        let backElem = document.querySelector(".back");
        backElem.addEventListener("click", (e) => {
            location.assign("./index.html");
        })
    </script>
</body>
</html>
```



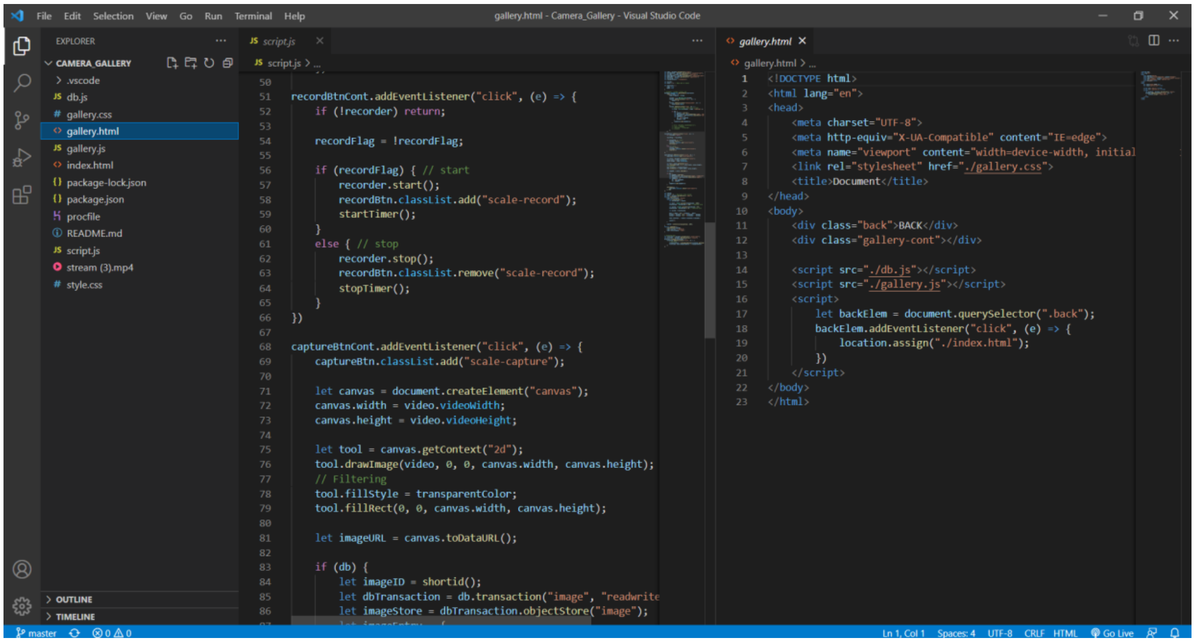
```
setTimeout(() => {
    if (db) {
        // videos retieval
        let videoDBTransaction = db.transaction("video", "read
        let videoStore = videoDBTransaction.objectStore("videc
        let videoRequest = videoStore.getAll();  //Event drive
        videoRequest.onsuccess = (e) => {
            let videoResult = videoRequest.result;
            let galleryCont = document.querySelector(".gallery
            videoResult.forEach((videoObj) => {
                let mediaElem = document.createElement("div");
                mediaElem.setAttribute("class", "media-cont");
                mediaElem.setAttribute("id", videoObj.id);

                let url = URL.createObjectURL(videoObj.blobDat

                mediaElem.innerHTML = `
                <div class="media">
                    <video autoplay loop src="${url}"></video>
                </div>
                <div class="delete action-btn">DELETE</div>
                <div class="download action-btn">DOWNLOAD</div
                `;

                galleryCont.appendChild(mediaElem);

                // Listeners
                let deleteBtn = mediaElem.querySelector(".dele
                deleteBtn.addEventListener("click", deleteList
                let downloadBtn = mediaElem.querySelector(".do
                downloadBtn.addEventListener("click", download
            })
        }

        // images retrieval
        let imageDBTransaction = db.transaction("image", "read
        let imageStore = imageDBTransaction.objectStore("image
        let imageRequest = imageStore.getAll();  //Event drive
```

```
let video = document.querySelector("video");
let recordBtnCont = document.querySelector(".record-btn-cont")
let recordBtn = document.querySelector(".record-btn");
let captureBtnCont = document.querySelector(".capture-btn-cont
let captureBtn = document.querySelector(".capture-btn");
let recordFlag = false;
let transparentColor = "transparent";

let recorder;
let chunks = []; // Media data in chunks

let constraints = {
    video: true,
    audio: true
}

// navigator -> global, browser info
navigator.mediaDevices.getUserMedia(constraints)
    .then((stream) => {
        video.srcObject = stream;

        recorder = new MediaRecorder(stream);
        recorder.addEventListener("start", (e) => {
            chunks = [];
        })
        recorder.addEventListener("dataavailable", (e) => {
            chunks.push(e.data);
        })
        recorder.addEventListener("stop", (e) => {
            // Conversion of media chunks data to video
            let blob = new Blob(chunks, { type: "video/mp4" })

            if (db) {
                let videoID = shortid();
                let dbTransaction = db.transaction("video", "r
                let videoStore = dbTransaction.objectStore("vi
                let videoEntry = {
                    id: `vid-${videoID}`,
```



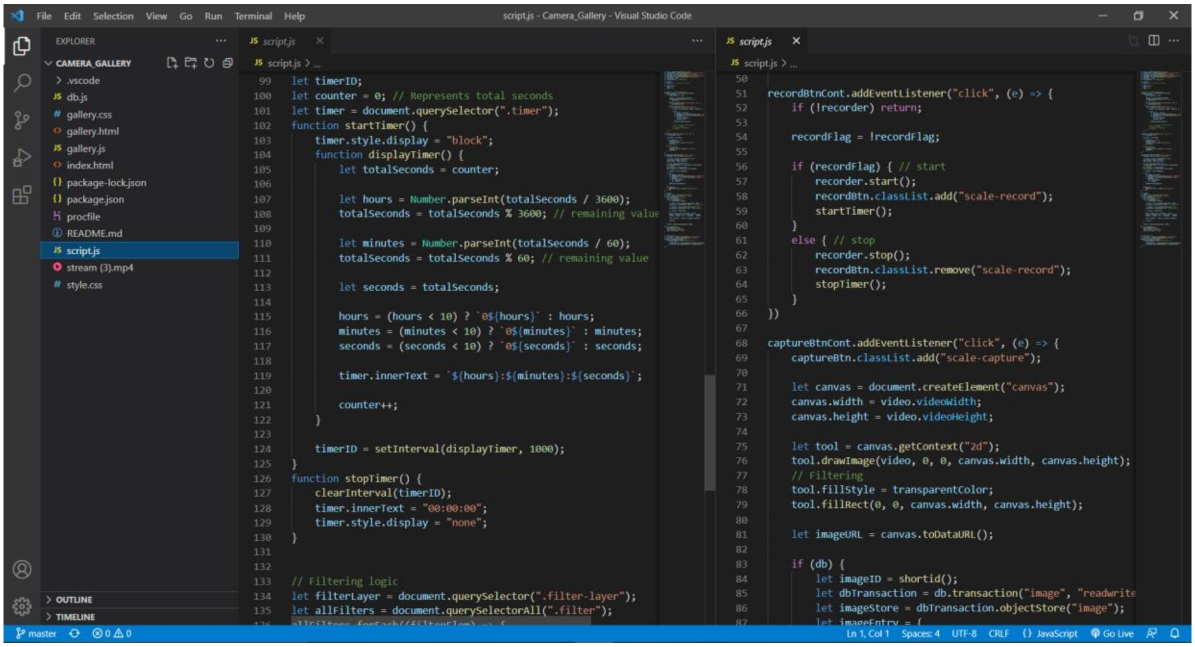
```
setTimeout(() => {
    if (db) {
        // videos retieval
        let videoDBTransaction = db.transaction("video", "read
        let videoStore = videoDBTransaction.objectStore("videc
        let videoRequest = videoStore.getAll();  //Event drive
        videoRequest.onsuccess = (e) => {
            let videoResult = videoRequest.result;
            let galleryCont = document.querySelector(".gallery
            videoResult.forEach((videoObj) => {
                let mediaElem = document.createElement("div");
                mediaElem.setAttribute("class", "media-cont");
                mediaElem.setAttribute("id", videoObj.id);

                let url = URL.createObjectURL(videoObj.blobDat

                mediaElem.innerHTML = `
                <div class="media">
                    <video autoplay loop src="${url}"></video>
                </div>
                <div class="delete action-btn">DELETE</div>
                <div class="download action-btn">DOWNLOAD</div
                `;

                galleryCont.appendChild(mediaElem);

                // Listeners
                let deleteBtn = mediaElem.querySelector(".dele
                deleteBtn.addEventListener("click", deleteList
                let downloadBtn = mediaElem.querySelector(".do
                downloadBtn.addEventListener("click", download
            })
        }

        // images retrieval
        let imageDBTransaction = db.transaction("image", "read
        let imageStore = imageDBTransaction.objectStore("image
        let imageRequest = imageStore.getAll();  //Event drive
```

```
.record-btn {
    background-color: 🟥red;
    height: 3rem;
    width: 3rem;
    border-radius: 50%;
}
.capture-btn {
    background-color: ⬜white;
    height: 3rem;
    width: 3rem;
    border-radius: 50%;
}

.scale-record {
    animation-name: scale-record;
    animation-iteration-count: infinite;
    animation-duration: 1s;
}
.scale-capture {
    animation-name: scale-capture;
    animation-iteration-count: 1;
    animation-duration: 1s;
}

@keyframes scale-record {
    0% {
        transform: scale(0.8);
    }
    100% {
        transform: scale(1.1)
    }
}
@keyframes scale-capture {
    0% {
        transform: scale(0.8);
    }
    100% {
```



```javascript

recordBtnCont.addEventListener("click", (e) => {
    if (!recorder) return;

    recordFlag = !recordFlag;

    if (recordFlag) { // start
        recorder.start();
        recordBtn.classList.add("scale-record");
        startTimer();
    }
    else { // stop
        recorder.stop();
        recordBtn.classList.remove("scale-record");
        stopTimer();
    }
})

captureBtnCont.addEventListener("click", (e) => {
    captureBtn.classList.add("scale-capture");

    let canvas = document.createElement("canvas");
    canvas.width = video.videoWidth;
    canvas.height = video.videoHeight;

    let tool = canvas.getContext("2d");
    tool.drawImage(video, 0, 0, canvas.width, canvas.height);
    // Filtering
    tool.fillStyle = transparentColor;
    tool.fillRect(0, 0, canvas.width, canvas.height);

    let imageURL = canvas.toDataURL();

    if (db) {
        let imageID = shortid();
        let dbTransaction = db.transaction("image", "readwrite
        let imageStore = dbTransaction.objectStore("image");
        let imageEntry = {
```

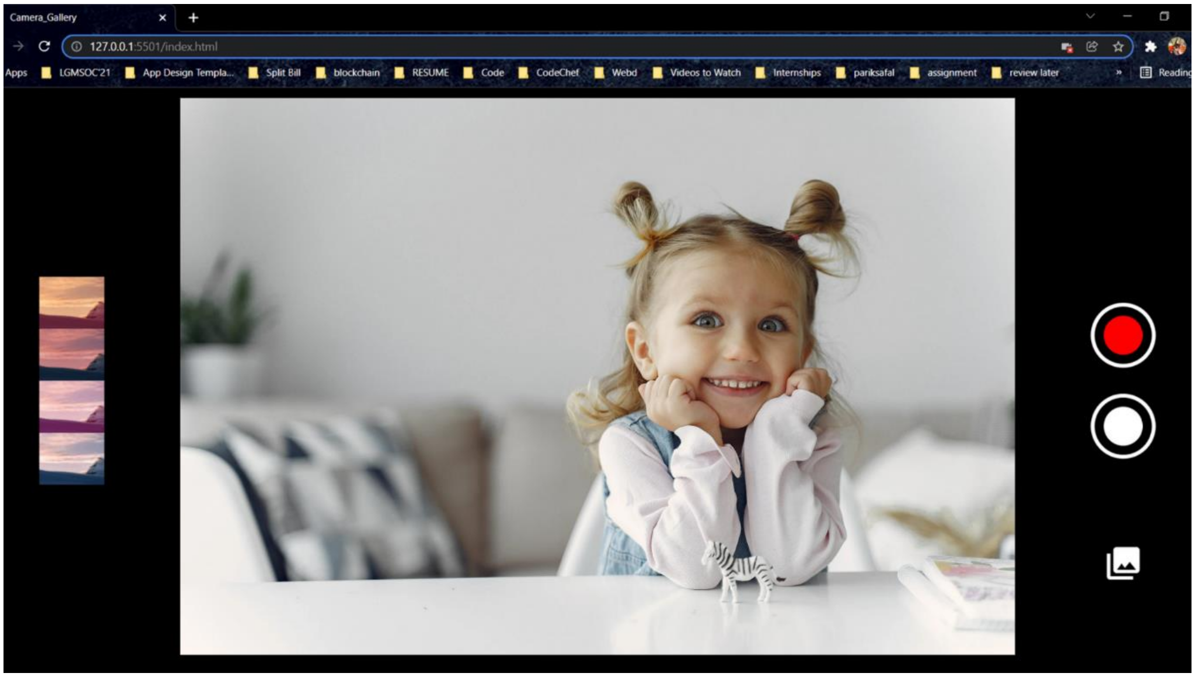
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial
    <link rel="stylesheet" href="./gallery.css">
    <title>Document</title>
</head>
<body>
    <div class="back">BACK</div>
    <div class="gallery-cont"></div>

    <script src="./db.js"></script>
    <script src="./gallery.js"></script>
    <script>
        let backElem = document.querySelector(".back");
        backElem.addEventListener("click", (e) => {
            location.assign("./index.html");
        })
    </script>
</body>
</html>
```

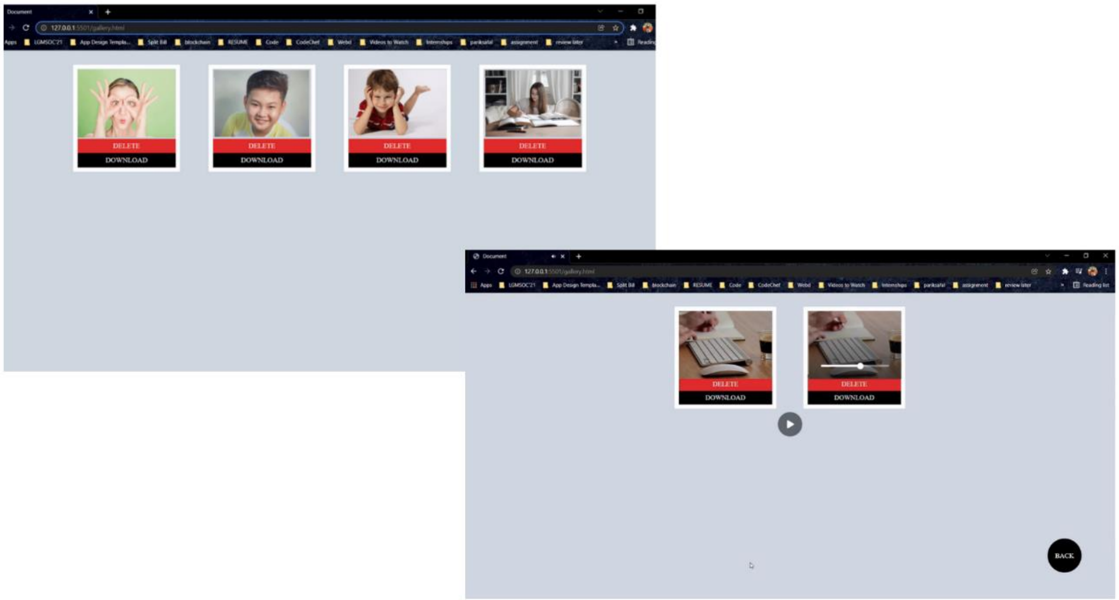


```javascript
let timerID;
let counter = 0; // Represents total seconds
let timer = document.querySelector(".timer");
function startTimer() {
    timer.style.display = "block";
    function displayTimer() {
        let totalSeconds = counter;

        let hours = Number.parseInt(totalSeconds / 3600);
        totalSeconds = totalSeconds % 3600; // remaining value

        let minutes = Number.parseInt(totalSeconds / 60);
        totalSeconds = totalSeconds % 60; // remaining value

        let seconds = totalSeconds;

        hours = (hours < 10) ? `0${hours}` : hours;
        minutes = (minutes < 10) ? `0${minutes}` : minutes;
        seconds = (seconds < 10) ? `0${seconds}` : seconds;

        timer.innerText = `${hours}:${minutes}:${seconds}`;

        counter++;
    }

    timerID = setInterval(displayTimer, 1000);
}
function stopTimer() {
    clearInterval(timerID);
    timer.innerText = "00:00:00";
    timer.style.display = "none";
}


// Filtering logic
let filterLayer = document.querySelector(".filter-layer");
let allFilters = document.querySelectorAll(".filter");
allFilters.forEach((filterElem) => {
```

```javascript

recordBtnCont.addEventListener("click", (e) => {
    if (!recorder) return;

    recordFlag = !recordFlag;

    if (recordFlag) { // start
        recorder.start();
        recordBtn.classList.add("scale-record");
        startTimer();
    }
    else { // stop
        recorder.stop();
        recordBtn.classList.remove("scale-record");
        stopTimer();
    }
})

captureBtnCont.addEventListener("click", (e) => {
    captureBtn.classList.add("scale-capture");

    let canvas = document.createElement("canvas");
    canvas.width = video.videoWidth;
    canvas.height = video.videoHeight;

    let tool = canvas.getContext("2d");
    tool.drawImage(video, 0, 0, canvas.width, canvas.height);
    // Filtering
    tool.fillStyle = transparentColor;
    tool.fillRect(0, 0, canvas.width, canvas.height);

    let imageURL = canvas.toDataURL();

    if (db) {
        let imageID = shortid();
        let dbTransaction = db.transaction("image", "readwrite
        let imageStore = dbTransaction.objectStore("image");
        let imageEntry = {
```

3.1 PROJECT SCREENSHOTS





PROJECT IS LIVE: CHECK HERE

CHAPTER 4: CONCLUSION

This project built on HTML, CSS and JavaScript with all the frontend development skills learnt and thoroughly researched upon. The project is a perfect example of all the concepts that are there to be explored in frontend web development and not only just front end, this project also utilizes WEB APIs. The understanding of WEB APIs is a must to understand and create this project which was successfully done and implemented.

As a newbie in this field, there were a few challenges in the way, however with proper guidance from our mentors and the research that I did on my own, I was able to clear a lot of hurdles and build this website on my own. This summer training along with this project proved to be extremely helpful in the growth of my future. Some of the key takeaways were:

1. Developed a New Skill

2. Gained knowledge about how the industry works in real-Life

3. Improved Research Skills

4. Realised the Potential inside and implemented it to successfully submit this project.

4.1 PROJECT APPLICATION AND FUTURE ASPECT

CAM-GALLERY has wide usage and can be used as a basic tool to record videos and upload them to YouTube, other social networking sites for mass propagation of ideas and stories.

I could also incorporate a chat system and extend this project into a video-based chat website where people could click photos and record videos and share it with friends at the same time!

It could also be turned into a social media platform with inclusion of some basic features. I plan to integrate backend with this project then all the working would be handled by server and the website would work much faster and smoother.

REFRENCES

1.https://www.geeksforgeeks.org/html-tutorials/?ref=lbp

2.https://www.geeksforgeeks.org/css-tutorials/?ref=lbp

3.https://developer.mozilla.org/en-US/docs/Web

4.https://www.w3schools.com/w3css/defaulT.asp

5. https://developer.mozilla.org/en-US/docs/Web/API

6.https://www.w3schools.com/js/

7. https://www.geeksforgeeks.org/web-development/

8. https://www.geeksforgeeks.org/working-with-apis-in-javascript/

9. https://www.geeksforgeeks.org/html-tutorials/?ref=lbp

10.https://www.geeksforgeeks.org/css-tutorials/?ref=lbp