A

PROJECT REPORT

ON

"EYE VIEW TRACKER FOR ONLINE EXAMINATION"

Submitted By

Ayush P. Thaware **Lokesh K. Rahangdale**

Krunal N. Patle **Aarti G. Dandhare**

Bhuvan T. Thakre

Guide

Prof. Vaishnavi Ganesh

in partial fulfilment for the award of the degree of

B. TECH.
IN
COMPUTER SCIENCE AND ENGINEERING

PRIYADARSHINI COLLEGE OF ENGINEERING

RASHTRASANT TUKDOJI MAHARAJ NAGPUR UNIVERSITY

Session 2022-23

PRIYADARSHINI COLLEGE OF ENGINEERING, NAGPUR

Department of Computer Science and Engineering

CERTIFICATE

Certified that this project report **"EYE VIEW TRACKER FOR ONLINE EXAMINATION"** is the bonafide work of "**AYUSH P. THAWARE, LOKESH K. RAHANGDALE, KRUNAL N. PATLE, AARTI G. DANDHARE, BHUVAN T. THAKRE"** from 6th Semester carried out the project work under my supervision.

Mrs. Vaishnavi Ganesh
Asst. Prof. & Guide

Mrs. Rajshri Pote
Asst. Prof. & Project In-charge

Dr. Leena Patil
Assoc. Prof. & Head of the Department

Dr. S.A. Dhale
Principal

DECLARATION

I hereby declare that the project entitled "**EYE VIEW TRACKER FOR ONLINE EXAMINATION**" submitted for the B. TECH. (Computer Science and Engineering) degree is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the Students

Ayush P. Thaware

Semester: 6th

Lokesh K. Rahangdale

Place: Nagpur

Date:

Krunal N. Patle

Aarti G. Dandhare

Bhuvan T. Thakre

AKNOWLEDGEMENT

I would like to take this opportunity to express my deep appreciation and gratitude to everyone who has contributed to the successful completion of my thesis report on **"Eye View Tracker for Online Examination".**

I am extremely grateful to my advisor "**Prof. Vaishnavi Ganesh**", for her guidance, encouragement, and support throughout the project. Her vast knowledge, constructive feedback, and timely advice were invaluable in shaping my research work and helping me achieve my objectives.

I would also like to extend my sincere thanks to **Priyadarshini College of Engineering, Nagpur** for providing me with the necessary resources and facilities to carry out my research work. I am grateful for the opportunity to work with cutting-edge technologies and gain valuable insights.

My heartfelt thanks go out to my friends and colleagues, who provided me with support, motivation, and feedback throughout my research work. I am grateful for the discussions, brainstorming sessions, and the exchange of ideas, which helped me to broaden my horizons and enhance my understanding of the topic.

ABSTRACT

This project proposes an "Eye View Tracker for Online Examination" that makes use of the webGazer library to track students' eye movements during online exams. The webGazer library is a JavaScript-based eye-tracking library that tracks the user's eye movements using the computer's webcam. The Eye View Tracker for Online Examination is a novel instrument designed to improve the integrity of online exams and provide a fair and just evaluation procedure. The webGazer library is integrated into an online examination platform by the Eye View Tracker for Online Examination, which is developed as a web-based application. During an online exam, the Eye View Tracker monitors the student's eye movements and detects any odd behaviour that could indicate academic cheating. The technical aspects of the project involve the incorporation of the webGazer library into the web-based application and the development of algorithms to detect suspicious behaviour based on eye tracking data. The research also explores the merits and drawbacks of using eye tracking technology in online exams. Overall, the Eye View Tracker for Online Examinations is a useful tool for maintaining the integrity of online tests and ensuring that students areevaluated fairly and justly.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -1

INTRODUCTION

1.1 INTRODUCTION

Online exams have become more common in recent years, particularly with the shift towards remote learning and remote work. However, one of the most difficult aspects of online exams is ensuring that students are not cheating or engaging in any other type of academic misconduct. One possible solution to this problem is to utilise eye tracking equipment to trackthe students' eye movements during the exam.

We present an "Eye View Tracker for Online Examination" that leverages the webGazer library to track students' eye movements during online exams. The webGazer library is a JavaScript-based eye-tracking library that tracks the user's eye movements using the computer's webcam. By incorporating this library into an online service exam platform, we can monitor the student's eye movements on the examination platform and identify any unusual activities that may indicate academic misconduct.

The Eye View Tracker for Online Examination is a novel instrument designed to improve the integrity of online exams and provide a fair and justevaluation procedure. In the following parts, we will go through the project's execution and technical aspects, as well as the potential benefits and drawbacks of using eye tracking technology in online exams.

1.2 LITERATURE SURVEY

Eye tracking has been used in various fields, such as psychology, marketing, and education. In the context of online exams, eye tracking can be used to monitor test-takers' gaze behavior to prevent cheating, improve test design, and enhance test-takers' experience.[1]

Eye tracking technology has been gaining popularity in recent years due to its ability to capture users' gaze patterns, which can provide insights into cognitive processes and behaviors (Bulling et al., 2011).[2]

Additionally, a study by Duchowski et al. (2018) investigated the use of eye tracking to assess the difficulty of online exams. The study found that eye tracking metrics, such as fixation duration and number of fixations, can be used to predict the difficulty of exam questions. This information can be used to adjust the difficulty level of questions and ensure that the exam accurately measures students' knowledge and skills. [3]

Another study by Fleischer et al. (2018) explored the use of eye tracking to evaluate the usability of online exams. The study found that eye tracking data can be used to identify areas of the test interface that cause difficulties for test-takers, such as unclear instructions or distracting elements. This information can be used to improve the design of online exams and enhance test-takers' experience. [4]

One study by Krasich et al. (2021) investigated the use of eye tracking technology for online exams in higher education. The study found that eye tracking can effectively detect cheating behaviors, such as looking at external sources, and that students who were aware of the eye tracking system were less likely to cheat. However, the study also noted that there are ethical concerns surrounding the use of eye tracking for surveillance purposes, and that the system should be transparent and secure to ensure students' privacy.[5]

A study by Dhawan, R., & Jha, S. (2020) , provides a comprehensive review of eye-tracking technology for online exam proctoring. The authors discuss the advantages and limitations of eye-tracking technology and describe the existing approaches for detecting cheating behaviors using eye-tracking technology. [6]

CHAPTER -2

ANALYSIS AND DESIGN

2.1 REQUIREMENT ANALYSIS

Before we begin implementing the Eye View Tracker for Online Examination using the webGazer library, we must first assess the project's needs. The following are the requirements that we must meet:

1. Web-based Application: The Eye View Tracker should be web-based and easily accessible from any internet-connected device. It should be linked to an online examination platform, allowing students to take tests online.
2. The Eye View Tracker should make use of an eye tracking system that can precisely identify the user's eye movements. The webGazer library, a JavaScript-based eye-tracking library that makes use of the computer's webcamto track the user's eye movements, will be used for this project.
3. Real-time tracking: During the test, the Eye View Tracker should keep tabs on the user's eye movements. It ought to be able to recognize any shifts in the user's sight, such as when they turn their head or look away from the screen.
4. Integration with Examination Platform: To track the user's eye movements throughout the exam, the Eye View Tracker should be integrated with the online examination platform. It needs to be able to spot any unusual behaviors that might point to academic dishonesty.
5. To guarantee that it may be used by a variety of users, the Eye View Tracker should be compatible with various web browsers and operating systems.
6. An easy-to-use, understandable user interface is essential for the Eye View Tracker. It must to come with thorough usage guidelines. the eye tracking technology and should provide the data obtained from the eye tracking in a clear and understandable manner.

These conditions must be met in order for us to create an efficient Eye View Tracker for Online Examination that will protect students from academic dishonesty and improve the integrity of online assessments.

2.2 TECHNOLOGY AND SOFTWARE DETAILS

Software and Technologies

In order to create the Eye View Tracker for Online Examination, several software tools and technologies were used. The following are the main technologies and software tools utilized in this project:

1. WebGazer Library
2. HTML
3. CSS
4. JavaScript
5. Vs Code (IDE)
6. Web Browser(any)

Hardware Requirements

In order to create the Eye View Tracker for Online Examination, several hardware tools were used. The following are the main hardware tools with minimum specification utilized in this project:

1. **Device:** Any PC, Laptop having webcam.
2. **Ram Capacity:** 4.00 / 8.00 GB.
3. **Processor:** Intel Pentium.
4. **OS Architecture:** 64-Bit, x86 based processor.

2.2.1 WebGazer Library:

WebGazer is a JavaScript-based library that allows for real-time eye tracking through the computer's webcam. The Human-Computer Interaction Lab at the University of Washington created the library, which was released as an open-source project in 2016.

WebGazer detects the position of the user's eyes on the screen using a machine learning model. When a user visits a WebGazer-integrated website, the library takes the video stream from the user's webcam and analyses the photos to estimate the user's eye position. To accurately track the user's eye movements, the library uses a combination of face detection and gaze estimation techniques.

Once the user's eye movements are observed, the data can be utilized for a variety of purposes, including improving the website's user experience or, in the instance of the Eye View Tracker for Online Examination, tracking the user's behavior during an online exam. Typically, eye-tracking data is saved in a database and analyzed with machine learning algorithms to find trends and anomalies in the user's behavior.

WebGazer is superior to typical eye-tracking devices in various ways. For starters, no specialized hardware is required, such as eye-tracking cameras, which may be costly and complex to set up. Second, it is simple to incorporate into existing websites using JavaScript, allowing it to be used by a broader spectrum of developers. Finally, because it does not require the user to wear any additional equipment or devices, it is less invasive than standard eye-tracking systems.

Overall, WebGazer is a creative and adaptable library with the potential to transform the field of eye-tracking. Its simplicity, accessibility, and accuracy make it a perfect tool for a variety of applications, ranging from online examination monitoring touser experience research.

To use WebGazer.js you need to add the webgazer.js file as a script in your website:

```
/* WebGazer.js library */
<script src="webgazer.js" type="text/javascript" >
```

The webgazer object is added to the global namespace after the script is included. WebGazer offers ways for controlling the operation of WebGazer.js, such as starting and stopping it, adding call-backs, and changing out modules. Web gazer's two most crucial methods are webgazer.begin() and webgazer.setGazeListener(). webgazer.begin() starts the data collection that allows the predictions to work, so it's critical to call it early on. WebGazer.js is ready to start making predictions after calling webgazer.begin(). The webgazer.setGazeListener() method makes it easy to get these forecasts.

```
webgazer.setGazeListener(function(data, elapsedTime) {
if (data == null) {
        return;
}
var xprediction = data.x; //these x coordinates are relative to the viewport
var yprediction = data.y; //these y coordinates are relative to the viewport
console.log(elapsedTime); //elapsed time is based on time since begin was called
}).begin();
```

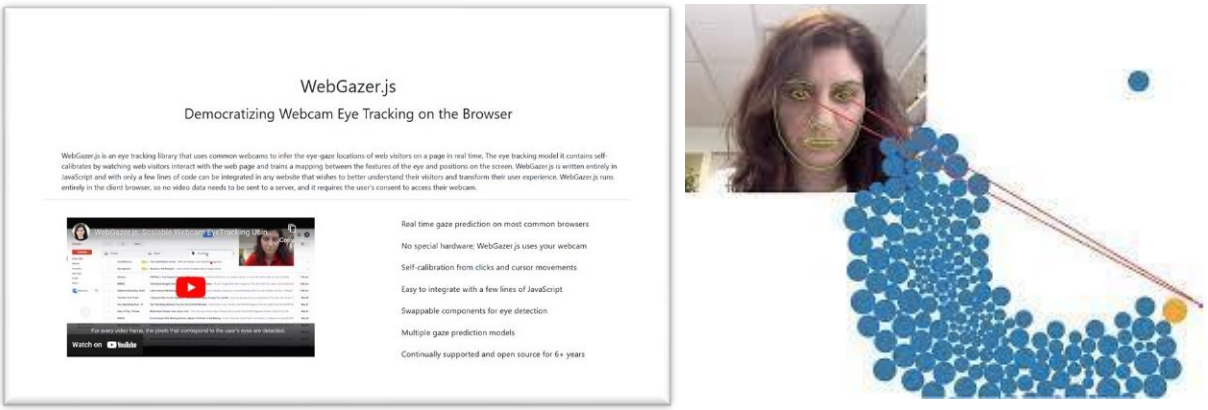


Fig 2.2.1: Web Gazer Library

2.2.2 HTML

HTML (Hyper-Text Markup Language) is a markup language that defines the structure of your content. HTML (Hyper-Text Markup Language) is one of the most important foundations on the web. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on. HTML is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

HTML Architecture:

The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e., doctype declaration, HTML, head, title, and body elements) upon which all web pages are created

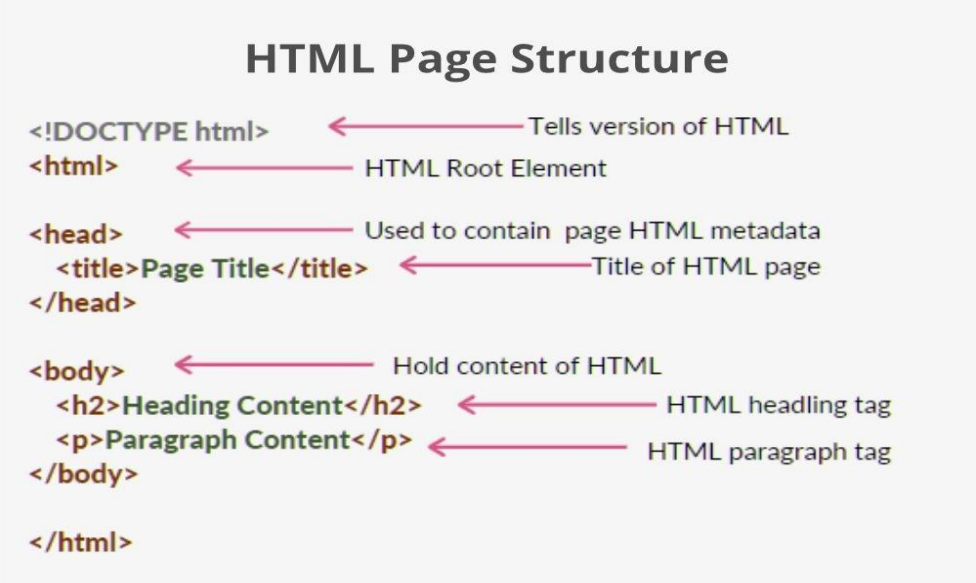


Fig 2.2.2: Architecture Of HTML

<!DOCTYPE HTML>: This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declarationis not case-sensitive.

<html>: This is called the HTML root element. All other elements are contained within it.

<head>: The head tag contains the "behind the scenes" elements for a webpage. Elements within the head aren't visible on the front-end of a webpage. HTML elements used inside the <head> element include:

- <style>
- <title>
- <script>
- <meta>
- <link>

<body>: The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end. An HTML document can be created using any text editor. Save the text file using .html or .htm. Once saved as an HTML document, the file can be opened as a webpage in the browser.

> **Note:** Basic/built-in text editors are Notepad (Windows) and TextEdit (Macs). Basic text editors are entirely sufficient for when you're just getting started. As you progress, there are many feature-rich text editors available which allow for greater function and flexibility.

Features of HTML:

• It is easy to learn and easy to use.

• It is platform-independent.

• Images, videos, and audio can be added to a web page.

•Multilanguage support

• Hypertext can be added to the text.

• It is a markup language

Advantages:

- HTML is used to build websites.
- It is supported by all browsers.
- It can be integrated with other languages like CSS, JavaScript, etc.

Disadvantages:

- HTML can only create static web pages. For dynamic web pages, other languages have to be used.
- A large amount of code has to be written to create a simple web page.
- The security feature is not good.

2.2.3 INTRODUCTION TO CSS

CSS stands for Cascading Style Sheets, CSS describes how HTML elements are to be displayed on screen, paper, or in other media.CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

There are three types of CSS which are given below:

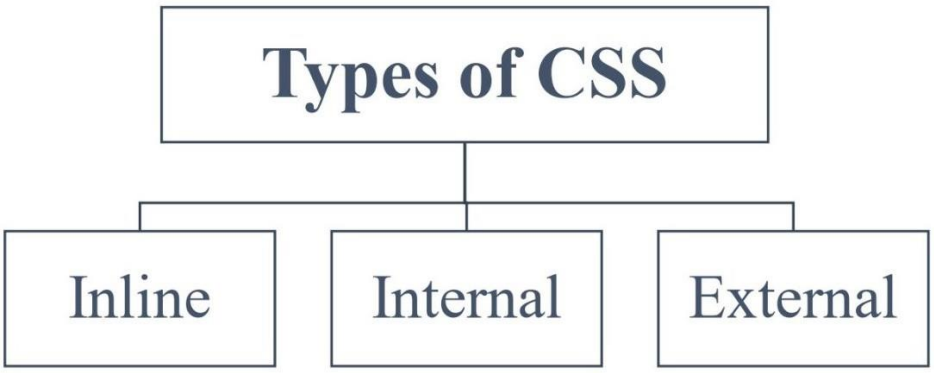


Fig 2.2.3(a): Types Of CSS

1. **External style sheet (Using HTML <link> Tag)**: External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc.). CSS property written in a separate file with .CSS extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages. The link tag is used to link the external style sheet with the html webpage.

```
<link rel="stylesheet" href="style.css">

</style>
```

2. **Internal style sheet (Using the <style> Element):** This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e., the CSS is embedded within the HTML file.

```
<style>
element {
// CSS property
}
</style>
```

3. **Inline Style:** Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using style attribute. It is used to apply a unique style fora single element.

```
<h1 style="style-property">Hello World</h1>
```

CSS Architecture: -

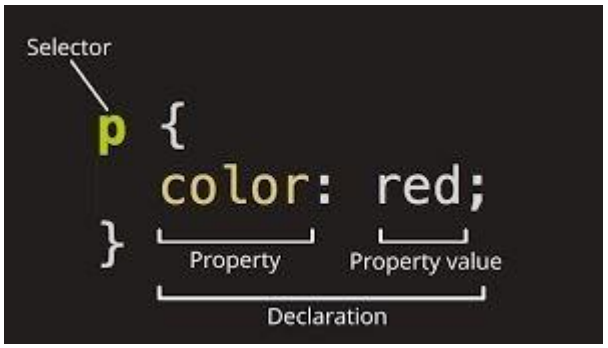


Fig 2.2.3(b): CSS Architecture

CSS Syntax and Selectors

A CSS Syntax rule consists of a selector, property, and its value. The selector points to the HTML element where CSS style is to be applied. The CSS property is separated by semicolons. It is a combination of selector name followed by the property: value pair that is defined for the specific selector.

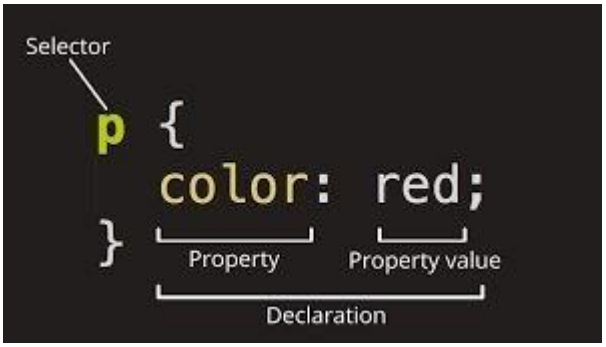


Fig 2.2.3(b): Syntax of CSS

A CSS Selectors are used to selecting HTML elements based on their element name, id, attributes, etc. It can select one or more elements simultaneously.

The Simple Selector can be categorized in 3 ways:

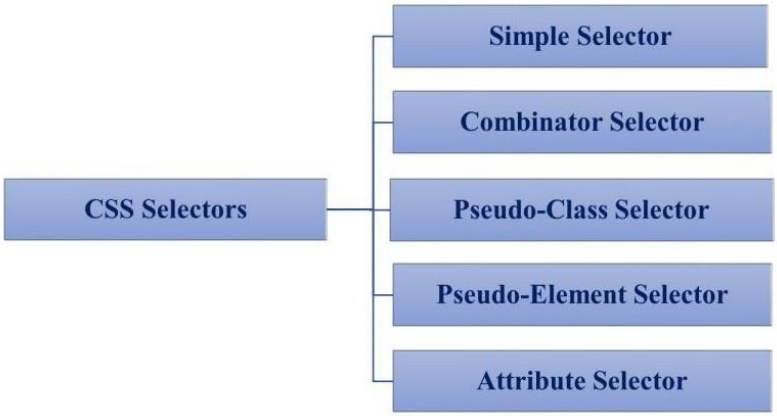


Fig 2.2.3 (c): Types of CSS Selector

CSS element Selector

The element selector selects HTML elements based on the element name.

```
Syntax:
element_name {
// CSS Property
}
```

CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```
Syntax:
#id_name {
// CSS Property
}
```

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.
To select elements with a specific class, write a period (.) character, followed by the class name.

```
Syntax:
.class_name {
// CSS Property
}
```

CSS Universal (*) Selector

The * selector in CSS is used to select all the elements in an HTML document. It also selects all elements which are inside under another element. It is also called the universal selector.

```
Syntax:
* {
// CSS Property
}
```

CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions. Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
Syntax:
h1, h2,h3 {
// CSS Property
}
```

CSS Comments:

The Comments in CSS, are the statements in your code that are ignored by the compiler and are not executed. Comments are used to explain the code. They make the program more readable and understandable. Comment can be single-line or multi-line. The /* */ comment syntax can be used

for both single and multiline comments.

```
Syntax:        /*Content*/
```

Features of CSS

Opportunity in Web Design: CSS and HTML knowledge are a requirement for anyone who wants to start a professional career in web design.

1. Website Design: By using CSS, we can control a wide range of styles, including text color, font style, paragraph spacing, column size and layout, background color, and image color. We can also alter how the layout appears on various screens and device sizes, as well as a wide range of other effects.
2. Web control: CSS is simple to learn and has the ability to control HTML documents. The markup languages XHTML and HTML are integrated with it. Other Languages: Once we understand the fundamentals of CSS and HTML, it will be easier to comprehend other related technologies like Angular, PHP, and JavaScript.
3. Other Languages: Once we have knowledge of some basics of CSS and HTML, other associated technologies like Angular, PHP, and JavaScript are become clearer to understand.

2.2.4 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.

Adding JavaScript can be added to your HTML file in two ways:

1. **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
2. **External JS:** We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

Syntax: <script>//JS Code</script>

Features of JavaScript

- Object-Centered Script Language
- Client-Side Technology
- Validation Of User's Input.
- Light-Weighted Language
- Interpreted Language
- Case Sensitive
- Used as Front-End as well as Back-End Language too.
- Platform Independent.

ADVANTAGES:

- Easy to Learn: JavaScript is easy to learn and has a relatively simple syntax, making it accessible to beginners.
- Wide Range of Applications: JavaScript can be used for a wide range of applications, including web development, mobile app development, game development, and server-side programming.
- Interactivity: JavaScript enables you to add interactivity and dynamic behavior to web pages, making them more engaging and user-friendly.

DISADVANTAGES:

- Browser Compatibility: Different browsers may interpret JavaScript differently, which can lead to inconsistencies in the behavior of web pages across different browsers.
- Security Risks: JavaScript can be vulnerable to security risks such as cross-site scripting (XSS) and code injection if not used correctly.
- Performance: JavaScript can impact the performance of web pages if not optimized correctly, leading to slower load times and a poorer user experience.

2.2.5. Vs Code (IDE)

Introduction to Visual Studio Code

In simple words, Visual Studio Code is a code editor. "A free editor that helps the programmer write code, aids in debugging, and corrects the code using the intelli-sense method," is what Visual Studio Code is described as. In plain English, it makes it easier for people to develop code. Although many claims that it is only partially an editor and an IDE, the choice is ultimately up to the coders. Any program or piece of software that we can see or use relies on background code to function.

Some of them were so elementary that it was quite challenging to write elementary English level programs in them. As time went on, many programming languages required a unique structure and support in order to be coded and developed further, which was not achievable with these editors. One of the various types of editors that have been developed is VI Editor, often known as Sublime Text Editor. Because of its features, users can customize the editor to suit their needs. For example, they can download libraries from the internet and incorporate them into the code as needed.

What can Visual Studio Code do?

Visual Studio Code has some very unique features. They are listed as below:

1. **Multiple programming languages are supported**: Supports several programming languages. Programmers formerly required Web-Capability, which has built-in multi-language support but requires a different editor for each language. This also means that it will be able to quickly identify any errors or cross-language references.

2. **Cross-Platform Support**: In the past, editors supported Windows, Linux, or Mac operating systems. But Visual Studio Code is platform-independent. Consequently, all three platforms can use it. Additionally, the code is compatible with all three platforms; previously, the open-source and proprietary software codes were distinct.
3. **Extensions and Support:** Usually supports all the programming languages but, if the user/programmer wants to use the programming language which is not supported then, he can download the extension and use it. And performance-wise, the extension doesn't slow down the editor as its rums as a different process.
4. **Repository:** With the ever-increasing demand for the code, secure and timely storage is equally important. It is connected with Git or can be connected with any other repository for pulling or saving the instances.
5. **Web-Support:** Comes with built-in support for Web applications. So, web applications can be built and supported in VSC.
6. **Hierarchy Structure:** The code files are located in files and folders. The required code files also have some files, which may be required for other complex projects. These files can be deleted as per convenience.
7. **Improving Code:** Some code snippets can be declared a bit differently, which might help the user in the code. This function prompts the user, wherever necessary, to change it to the suggested option.
8. **Terminal Support:** Many of the times, the user needs to start from the root of the directory to start with a particular action, in-built terminal or console provides user support to not to switch in-between two screens for the same.

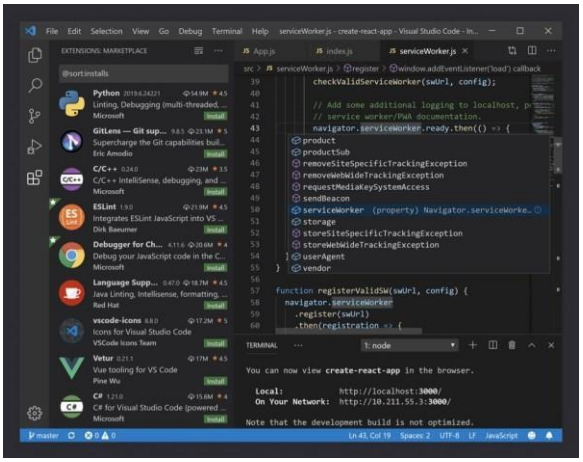


Fig 2.2.5: Visual Code IDE

CHAPTER-3

PROJECT DESCRIPTION

3.1 DESIGN OF PROPOSED WORK

To provide a precise and trustworthy tool for monitoring and tracking test-takers' eye movements during online exams, an eye view tracker for onlineexamination system using the WebGazer library was developed.

WebGazer is an effective open-source framework that may be used to buildeye view trackers for online tests because it uses machine learning methodsto track and predict eye movements in real-time.

The WebGazer library can be integrated into an online exam system to tracktest-takers' eye movements and identify any suspicious activity or unauthorised resource access. This can lessen the possibility of academic dishonesty and cheating, ensuring that the exam results are accurate and trueto the test-knowledge takers and skills.

The WebGazer library can help to streamline the exam administration procedure by automating the monitoring and surveillance process. As aresult, administrators' and proctors' workloads may be reduced, and theprocess of administering exams may be more effectively and efficientlyoverall.

Overall, the purpose of developing an eye view tracker for online examination system using the WebGazer library is to provide a powerful and reliable tool for enhancing the security and reliability of online exams,while also improving the efficiency and effectiveness of the exam administration process.

3.2 DATA FLOW DIAGRAM

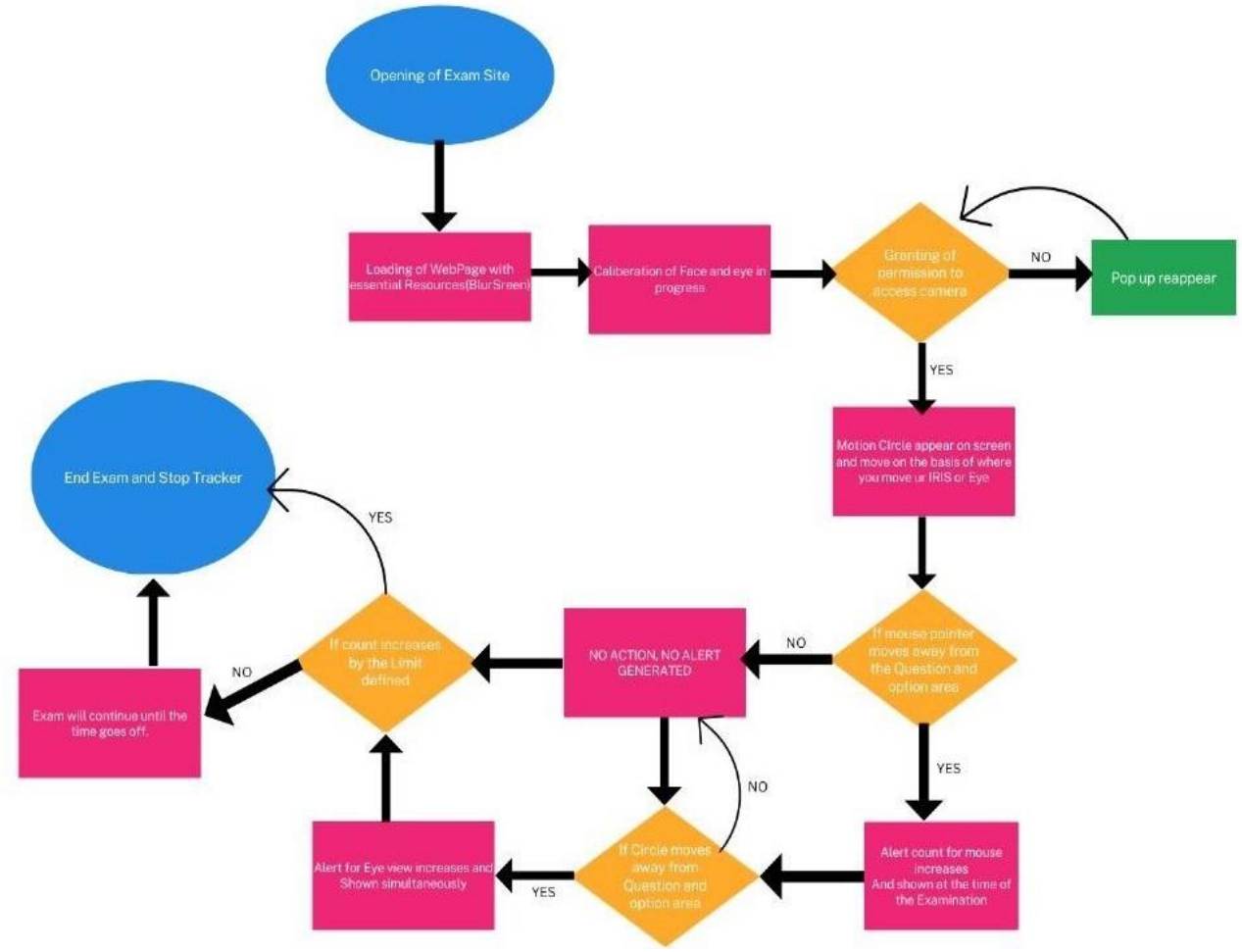


Fig 3.2: Data Flow Diagram

3.3 WORKING MODULE

The functioning module for creating an eye view tracker for an online examination system using the WebGazer library consists of the steps below:

1. **Library integration for WebGazer:** The WebGazer library must first be integrated into the online testing system. In order for the system to track the test taker's eye movements in real time, the necessary code must be added.

2. **Procedure for calibration:** Before the exam, the system must adjust the WebGazer library to each test-particular taker's eye movements. To recognise and calibrate the test-eye taker's movements, the system instructs them to follow a specific pattern on the screen.

3. **Eye movement tracking during the exam:** The system will provide the proctor with a detailed report of the test taker's eye movements during theexam after the exam. This can be used to validate the exam results and ensure that they are accurate and representative of the test taker's actual knowledge and abilities.

4. **Data review:** Following the exam, the technology will provide the proctorwith a detailed report of the test taker's eye movements. This can be used to ensure the integrity of the exam findings and that they are true and representative of the test taker's actual knowledge and abilities.

5. **Additional analysis of eye movement data**: Following the exam, the system will provide the proctor with a detailed report of the test taker's eyemovements during the exam. This can be used to confirm the exam findings' integrity, as well as their accuracy and relevance to the test taker's actual knowledge and abilities.

CHAPTER: - 4

PROJECT IMPLEMENTATION

4.1 MODULE IMPLEMENTATION

4.1.1 HTML CODE

index.html (file name)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      rel="stylesheet"
     href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.14.0/css/all.min.css"
integrity="sha512-
1PKOgIY59xJ8Co8+NE6FZ+LOAZKjy+KY8iq0G4B3CyeY6wYHN3yt9PW0XpS
riVlkMXe40PTKnXrLnZ9+fkDaog=="
crossorigin="anonymous" />
    <link rel="stylesheet" href="style.css" />
    <!-- <link rel="stylesheet" href="/WebGazer/www/css/style.css"> -->
    <script src="https://webgazer.cs.brown.edu/webgazer.js?"></script>
    <!-- used online webgazer library source-->
    <!-- <script src="/WebGazer/www/webgazer.js"></script> -->
    <title>Eye View Tracker for Online Examination</title>
  </head>
  <body>
    <div id="count">
      <div id="alertCount"></div>
      <p>
        Red Dot Alert Count :
```

```
      <span id="alert-count">0</span>
    </p>
    <p class="note">Note: As the Alert Count of The Red Dot Exceeds 100 Then the Exam will be Closed/Terminated</p>
  </div>
  <div class="quiz-container" id="myDiv">
    <!-- <div class="quiz-header"> -->
    <h2 id="question">Question is loading...</h2>
    <ul>
      <li>
        <input type="radio" name="answer" id="a" class="answer" />
        <label for="a" id="a_text">Answer...</label>
      </li>
      <li>
        <input type="radio" name="answer" id="b" class="answer" />
        <label for="b" id="b_text">Answer...</label>
      </li>
      <li>
        <input type="radio" name="answer" id="c" class="answer" />
        <label for="c" id="c_text">Answer...</label>
      </li>
      <li>
        <input type="radio" name="answer" id="d" class="answer" />
        <label for="d" id="d_text">Answer...</label>
      </li>
    </ul>
    <button id="submit">Submit</button>
    <!-- </div> -->
  </div>
// Internal JS given below
  <script>
    var alertCount = 0
    var divElement = document.getElementById('myDiv')
    divElement.addEventListener('mouseout', function (event) {
```

```
    // Check if the mouse is outside the div element and not inside any child elements
    if (!divElement.contains(event.relatedTarget)) {
      setTimeout(() => {
        alertCount++
      }, 5000)
      document.getElementById('alertCount').innerText =
        'Mouse Out Alert Counts :' + alertCount
        if (alertCount >= 10) {
        alert('Your exam is ended.')
        window.close()
      }
    }
  })
  // initialize WebGazer
  webgazer.setRegression('ridge') // set regression type
  webgazer
    .setGazeListener(function (data, elapsedTime) {
      if (!data) return
      var x = data.x
      var y = data.y
      // Get the position and dimensions of the container div
      var container = document.getElementById('myDiv')
      var containerRect = container.getBoundingClientRect()
      // Check if the gaze point is outside of the container
      if (
        x < containerRect.left ||
        x > containerRect.right ||
        y < containerRect.top ||
        y > containerRect.bottom
      ) { // Increment the alert count
        var alertCount = parseInt(
          document.getElementById('alert-count').textContent,
        )
        document.getElementById('alert-count').textContent = alertCount + 1
```

```
        } //end exam on number of alerts
        if (alertCount >= 100) {
        alert('Your exam is ended.')
        window.close()
        }
       }) .begin()
   </script>
   <script src="script.js"> // external JS file </script>
  </body>
</html>
```

4.1.2 CSS CODE

style.css (file name)

```
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@200
;400&display=swap");
* {
   box-sizing: border-box;
}
body {
   background-color: #b8c6db;
   background-image: linear-gradient(315deg, #b8c6db 0%,
#f5f7fa 100%);
   font-family: "Poppins", sans-serif;
   display: flex;
   align-items: center;
   justify-content: center;
   height: 100vh;
   overflow: hidden;
   margin: 0;
   filter: blur(5px);
   pointer-events: none;
}
```

```
.quiz-container {
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 10px 2px rgba(100, 100, 100, 0.1);
    width: 80%;
    margin-left: 50px;
    margin-right: 50px;
    margin-top: 50px;
    margin-bottom: 50px;
    height: 95%;
    max-width: 95vw;
    overflow: hidden;
}

.quiz-header {
    padding: 0.5rem;
}

h2 {
    padding: 0.5rem;
    text-align: left;
    margin-top: 10%;
    margin-left: 22%;
}

ul {
    list-style-type: none;
    padding: 5px;
    padding-top: 2rem;
    margin-left: 25%;
}

ul li {
    font-size: 1.2rem;
    margin: 1rem 0;
```

```
}

ul li label {
    cursor: pointer;
}

button {
    background-color: #8e44ad;
    color: #fff;
    border: none;
    margin-left: 25%;
    margin-top: 1.5rem;
    display: inline;
    width:  15vh;
    cursor: pointer;
    font-size: 1.1rem;
    font-family: inherit;
    padding: 1.3rem;
    border-radius: 10px;
}

button:hover {
    background-color: #732d91;
}

button:focus {
    outline: none;
    background-color: #5e3370;
}
#count {
    border: 5px solid green;
    border-radius: 5px;
    margin-top: 5rem;
    margin-left: 50px;
    text-align: center;
```

```
    width: 20rem;
}
.note {
    color: red;
    font-weight: 700;
    border: 5px solid red;
    border-radius: 5px;
    width:  85%;
    padding: 10px;
    margin-left: 20px;
}
```

4.1.3 JAVASCRIPT CODE

script.js(file name)

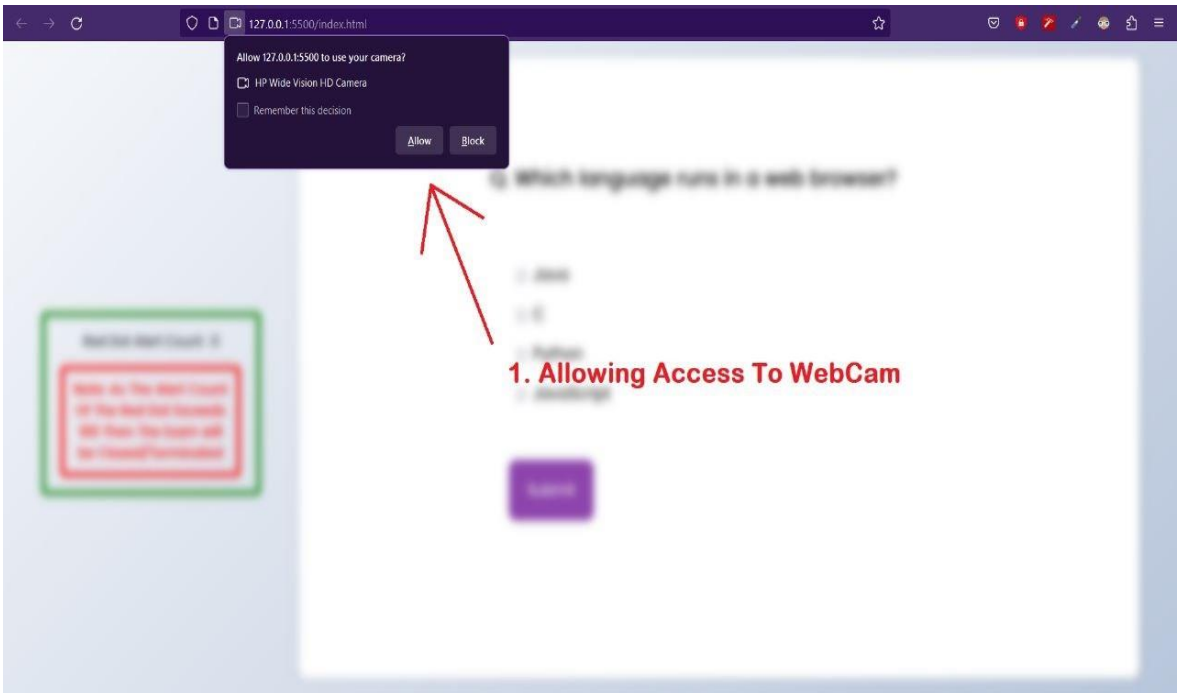
```
window.onload = function() {
    // set timer to remove blur effect after 20 seconds
    setTimeout(removeBlur, 10000);
  };
  function removeBlur() {
    // remove blur effect by setting the filter to "none"
    document.body.style.filter = "none";
    // remove restriction on mouse by setting the pointerEvents  to "auto"
    document.body.style.pointerEvents = "auto";
  }
//quiz questions
const quizData = [{
    question: "Q. Which language runs in a web browser?",
    a: "Java",
    b: "C",
    c: "Python",
    d: "JavaScript",
    correct: "d",
}, {
```

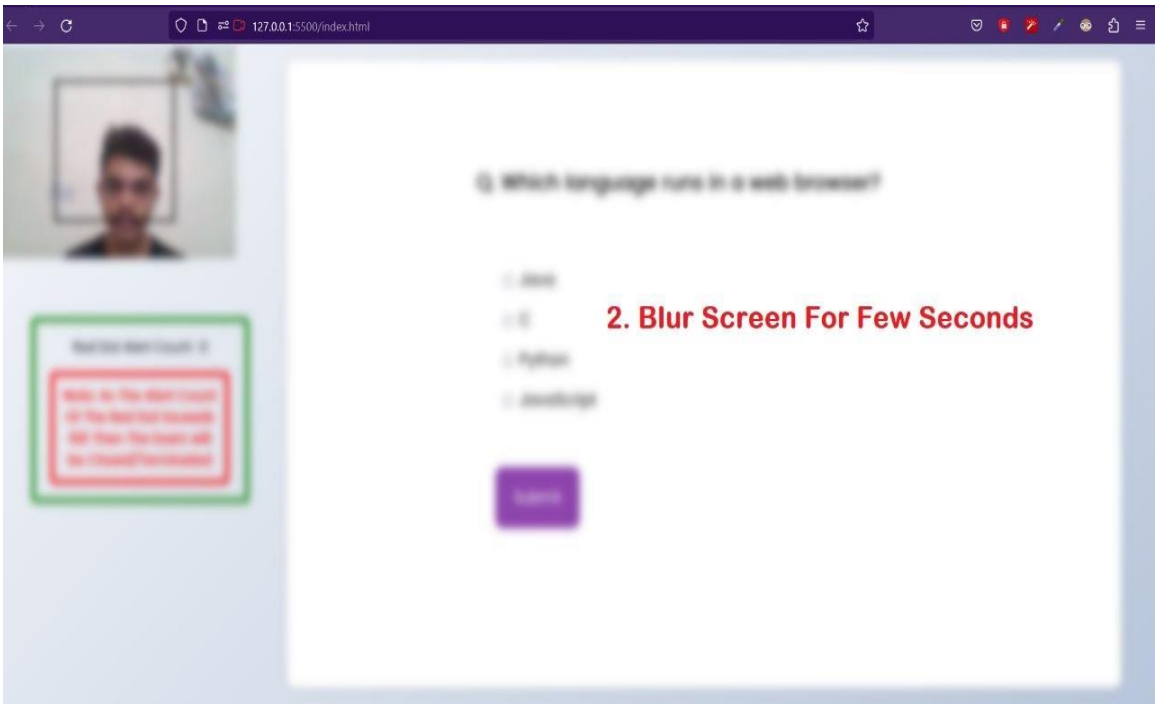
```
    question: " Q. What does CSS stand for?",
    a: "Central Style Sheets",
    b: "Cascading Style Sheets",
    c: "Cascading Simple Sheets",
    d: "Cars SUVs Sailboats",
    correct: "b",
}, {
    question: "Q. What does HTML stand for?",
    a: "Hypertext Markup Language",
    b: "Hypertext Markdown Language",
    c: "Hyperloop Machine Language",
    d: "Helicopters Terminals Motorboats Lamborginis",
    correct: "a",
}, {
    question: "Q. What year was JavaScript launched?",
    a: "1996",
    b: "1995",
    c: "1994",
    d: "none of the above",
    correct: "b",
}, ];
const quiz = document.getElementById("myDiv");
const answerElements = document.querySelectorAll(".answer");
const questionElement = document.getElementById("question");
const a_text = document.getElementById("a_text");
const b_text = document.getElementById("b_text");
const c_text = document.getElementById("c_text");
const d_text = document.getElementById("d_text");
const submitButton = document.getElementById("submit");
let currentQuiz = 0;
let score = 0;
const deselectAnswers = () => {
    answerElements.forEach((answer) => (answer.checked = false));
};
const getSelected = () => {
```

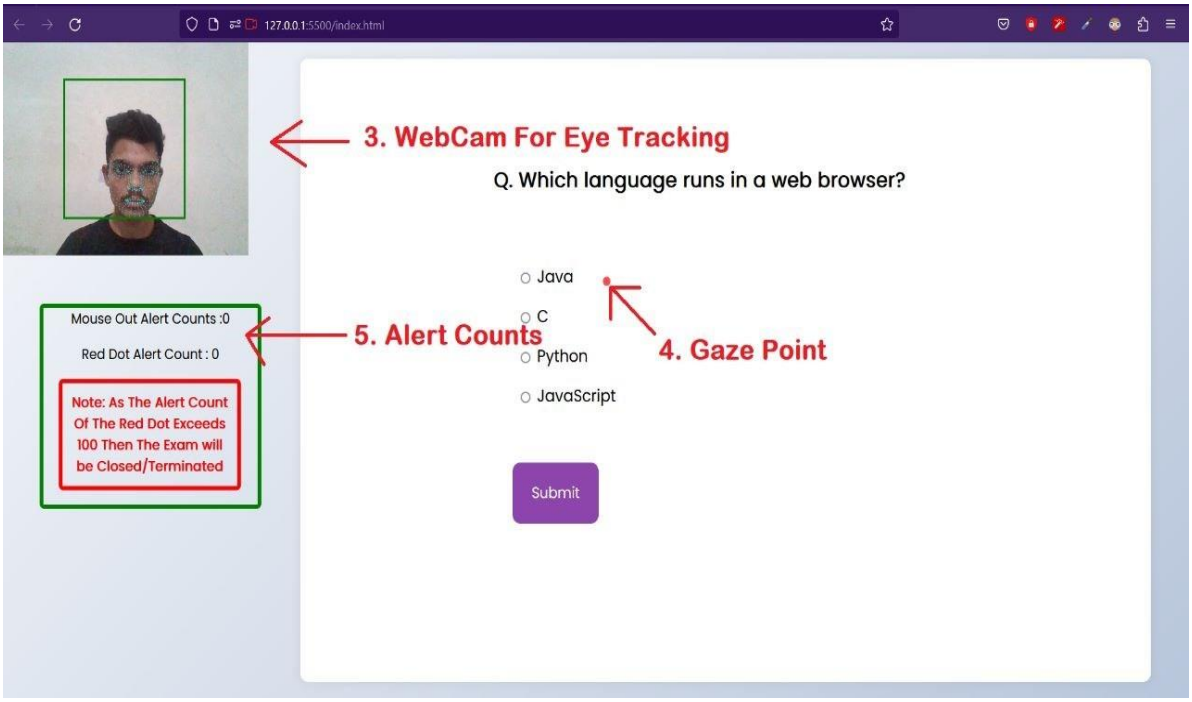
```
    let answer;
    answerElements.forEach((answerElement) => {
        if (answerElement.checked) answer = answerElement.id;
    });
    return answer;
};
const loadQuiz = () => {
    deselectAnswers();
    const currentQuizData = quizData[currentQuiz];
    questionElement.innerText = currentQuizData.question;
    a_text.innerText = currentQuizData.a;
    b_text.innerText = currentQuizData.b;
    c_text.innerText = currentQuizData.c;
    d_text.innerText = currentQuizData.d;
};
loadQuiz(); // loading the quiz questions on window
//on clicking on submit button, submitting the answer and moving to another
question
submitButton.addEventListener("click", () => {
    const answer = getSelected();
    if (answer) {
        if (answer === quizData[currentQuiz].correct) score++;
        currentQuiz++;
        if (currentQuiz < quizData.length) loadQuiz();
        else {
            //printing the score and closing the window on clicking on end exam
button
            quiz.innerHTML = ` <h2>You answered ${score}/${quizData.length}
questions correctly</h2>
                <button onclick="window.close()">End Exam</button>`
        }
    }
});
```

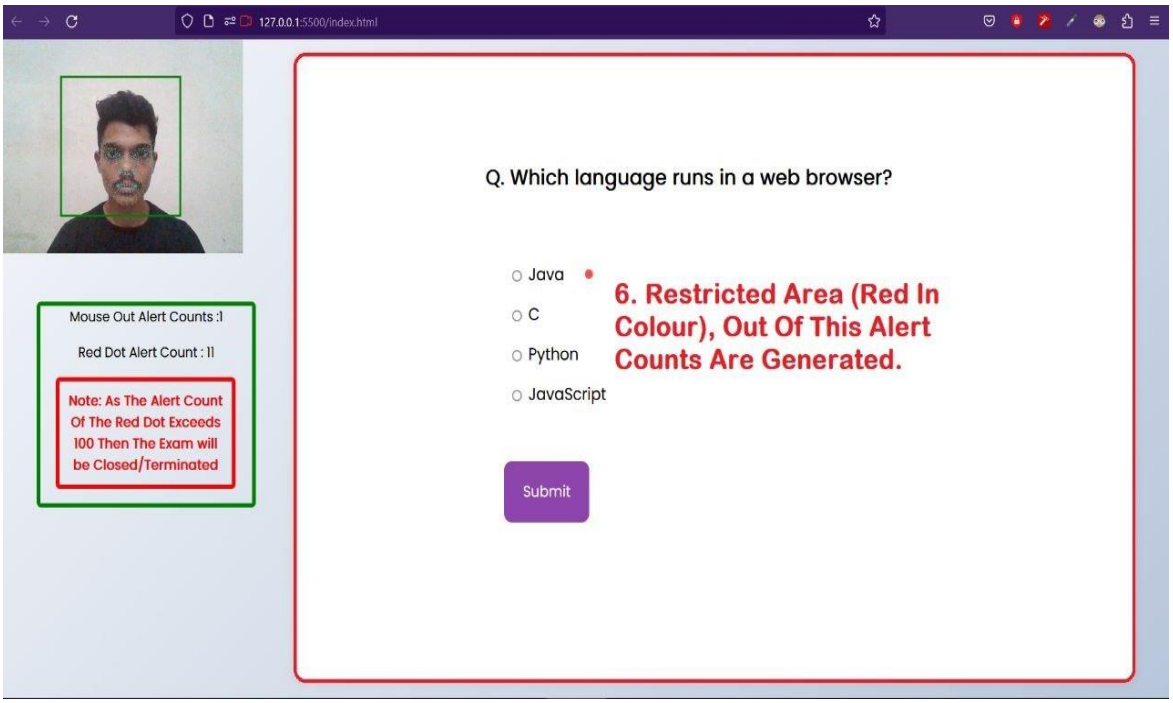
CHAPTER: -5

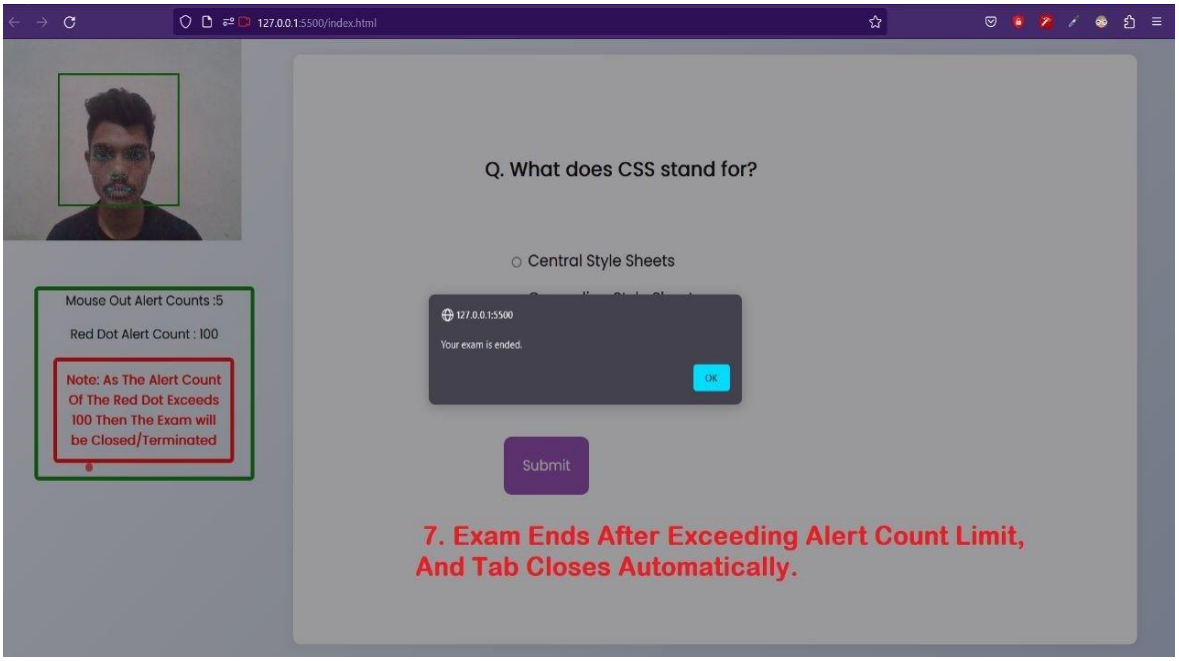
SCREENSHOTS

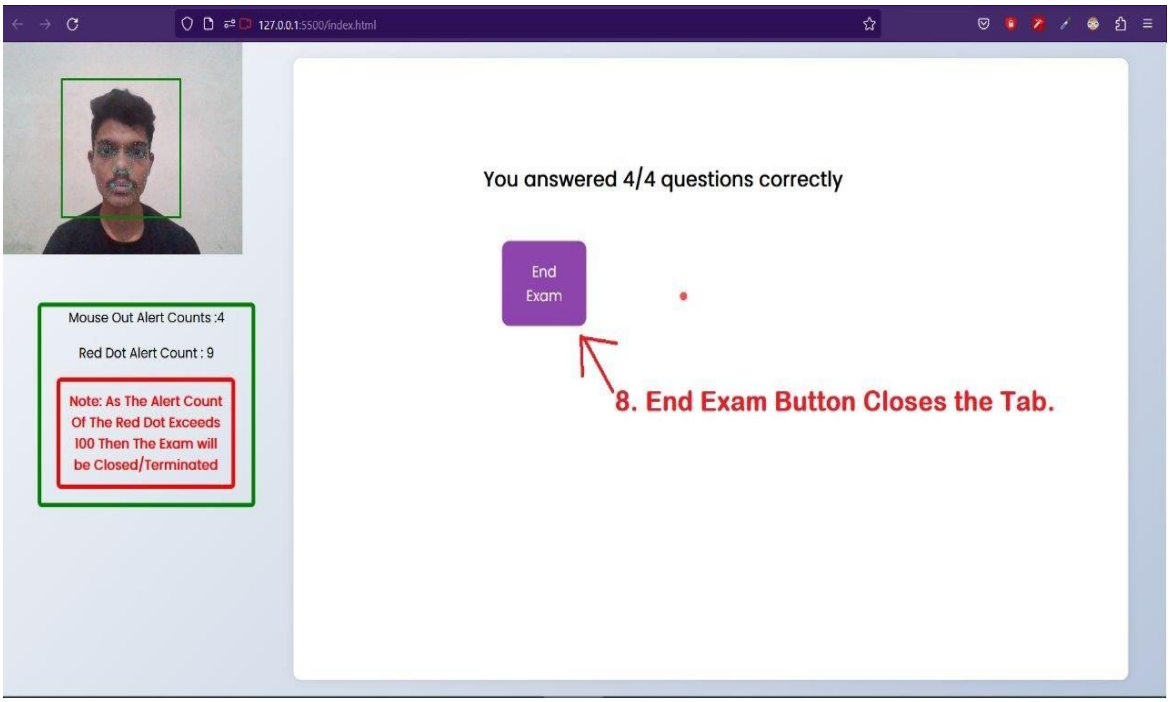





127.0.0.1:5500/index.html
3. WebCam For Eye Tracking
Q. Which language runs in a web browser?
Java
C
Python
JavaScript
4. Gaze Point
Mouse Out Alert Counts :0
Red Dot Alert Count : 0
5. Alert Counts
Note: As The Alert Count Of The Red Dot Exceeds 100 Then The Exam will be Closed/Terminated
Submit


127.0.0.1:5500/index.html
Q. Which language runs in a web browser?
Java
C
Python
JavaScript
6. Restricted Area (Red In Colour), Out Of This Alert Counts Are Generated.
Mouse Out Alert Counts :1
Red Dot Alert Count : 11
Note: As The Alert Count Of The Red Dot Exceeds 100 Then The Exam will be Closed/Terminated
Submit


127.0.0.1:5500/index.html
Mouse Out Alert Counts :5
Red Dot Alert Count : 100
Note: As The Alert Count Of The Red Dot Exceeds 100 Then The Exam will be Closed/Terminated
Q. What does CSS stand for?
Central Style Sheets
127.0.0.1:5500
Your exam is ended.
OK
Submit
7. Exam Ends After Exceeding Alert Count Limit, And Tab Closes Automatically.


127.0.0.1:5500/index.html
Mouse Out Alert Counts :4
Red Dot Alert Count : 9
Note: As The Alert Count Of The Red Dot Exceeds 100 Then The Exam will be Closed/Terminated
You answered 4/4 questions correctly
End Exam
8. End Exam Button Closes the Tab.

IMPLEMENTED PROJECT

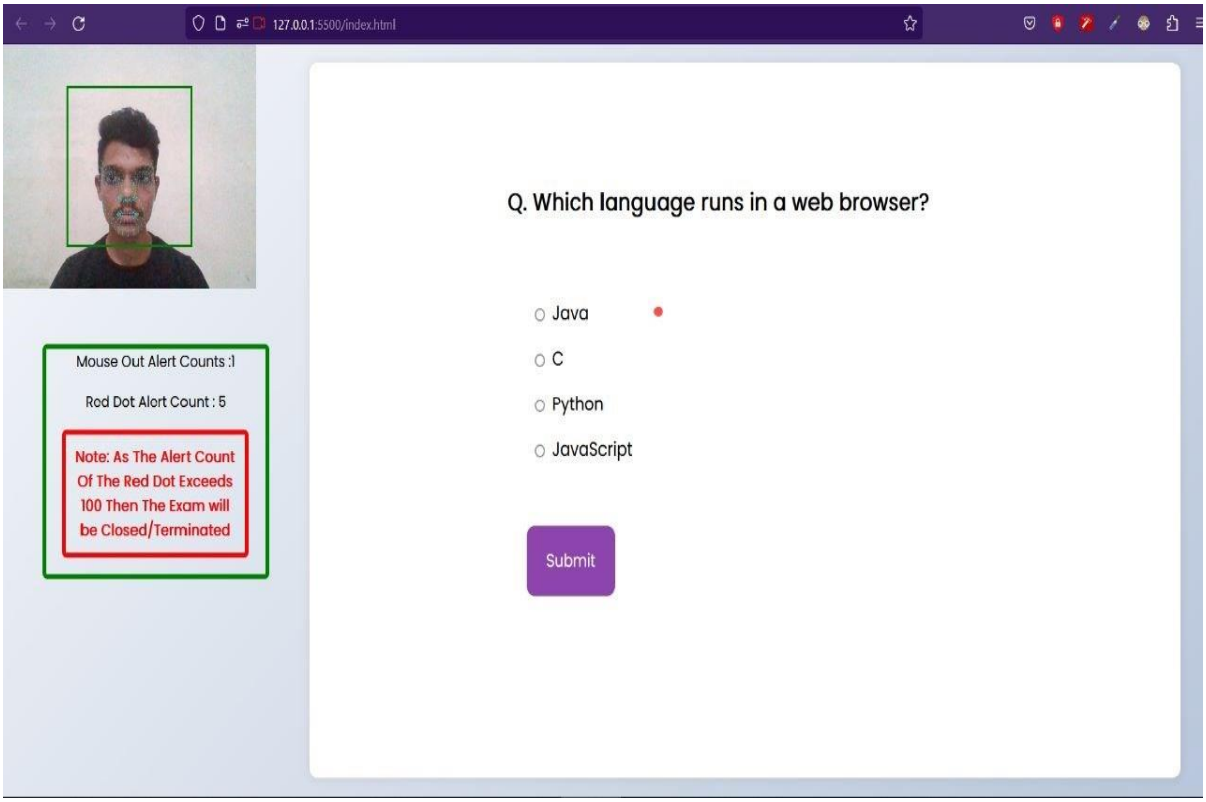


Fig 5.1: Implemented Project

CHAPTER:- 6

ADVANTAGES AND APPLICATION

6.1 ADVANTAGES

1. **Increased Exam Security:** Exam security can be improved by using eye view tracking technology to detect any suspicious behaviour, suchas looking away from the screen or accessing unauthorised resources. This can assist in preventing cheating and ensuring that exam results are accurate and reflect the test taker's actual knowledge and abilities.

2. **Real-time Monitoring:** Eye view tracking technology can monitor the test taker's eye movements in real time during the exam. This can aid in the detection of any anomalies or unusual behaviour, allowing the proctor to intervene and take appropriate action.

3. **Automated Monitoring:** Eye view tracking technology can monitorthe test taker's eye movements in real time during the exam. This canaid in the detection of any anomalies or unusual behaviour, allowingthe proctor to intervene and take appropriate action.

4. **Reliable Results:** Eye view tracking technology can provide reliable and accurate results, as it tracks the test taker's eye movements in real-time. This can help to ensure that exam results are accurate and reflective of the test taker's actual knowledge and abilities.

6.2 APPLICATIONS

1. **Educational Institutions:** Educational institutions can use eye view tracking technology to monitor online exams and prevent cheating. This can help to ensure that exam results are accurate and reflect the test taker'sactual knowledge and abilities.

2. **Employers:** Employers can use eye tracking technology to conduct online assessments and verify job applicants' skills and abilities. This canaid in ensuring that job candidates are qualified for the position.

3. **Certification Programs:** Certification programmes can use eye view tracking technology to validate the skills and abilities of individuals seeking certification. This can aid in ensuring that certified individuals arequalified for their roles.

4. **Government Exams:** Government agencies can use eye tracking technology to administer online exams for various positions, such as civilservice jobs. This can help to ensure that the hiring process is fair and objective, and that qualified candidates are chosen for the position.

Overall, eye view tracker technology has a wide range of benefits and applications in online examination systems. It can help to improve examsecurity, provide real-time monitoring, automate the monitoring process, and deliver consistent results. Educational institutions, employers, certification programmes, and government agencies can all use this technology to conduct fair and unbiased exams and assessments.

CHAPTER: -7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

An original solution that can successfully address the problem of cheating in online exams is the Eye View Tracker for Online Examination System using the WebGazer library. In order to track test-takers' eye movements during an exam, this system employs computer vision techniques. It is able to recognise and flag any suspicious behaviour, such as looking away from the screen, accessing unauthorised resources, or interacting with other people.

The monitoring procedure can also be automated with the help of the WebGazer library, which lessens the workload for administrators and proctors. This can lessen the possibility of mistakes or inconsistencies while also increasing the efficiency and effectiveness of the exam administration process.

The implementation of this system will increase the validity and fairness of online exams and give students a safe and dependable testing environment. Additionally,this system can lessen the workload of human proctors, improving the effectiveness and efficiency of the procedure.

7.2 FUTURE SCOPE

The Eye View Tracker for online examination system has enormous potential for adoption in many educational institutions, online learning platforms, and corporate training programs. As technology and artificial intelligence advance, this method's accuracy and efficiency in detecting cheating and other suspicious behaviors can be improved.

One potential future application is to integrate the Eye View Tracker with other biometric systems such as voice recognition and facial recognition, which would improve the system's accuracy and dependability. Another area of future researchcould be the development of machine learning algorithms that can analyze and understand the eye movement data collected during the tests, generating insights that can be used to improve the exam design and content.

CHAPTER: -8

REFERENCES

8.1 REFERENCES

[1] E. Cutrell and Z. Guan, “What are you looking for? an eye-tracking study of information usage in web search” In Proc. CHI, pages 407–416, 2017.

[2] Bulling, A., Ward, J. A., Gellersen, H., & Tröster, G. (2011). “Eye movements for human-computer interaction”. ACM Computing Surveys (CSUR), 43(3), 1-41.

[3] Duchowski, A. T., Betke, M., Vargas-Martin, M., & Payeur, A. (2018). “Using eye tracking to predict cognitive difficulty in reading.” ACM Transactions on Applied Perception (TAP), 15(1), 1-18.

[4] Fleischer, J., Kuusinen, K., & Tirri, H. (2018). “Using eye-tracking to evaluate the usability of online exams”. Journal of Educational Technology & Society, 21(1), 147-157.

[5] Krasich, K., Donnelly, M., & Burns, M. (2021). Using eye tracking to prevent cheating in online exams. International Journal for Educational Integrity, 17(1), 1-16.

[6] Dhawan, R., & Jha, S. (2020). Exploring eye-tracking technology for online examination: A review. International Journal of Advanced Science and Technology, 29(3), 2837-2846.