JSS MAHA VIDYAPEETHA, MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Uttrahalli-Kengeri Main Road, Bengaluru-560060

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEEERING

CERTIFICATE

This is to certify that the project work entitled *"*TYPING GAME - WEBSITE*"* is a bonafide work carried out by **Mr. Deepak Naidu (1JS19CS049), Mr. Girish Kumar DV (1JS19CS057) in fulfillment of the requirements for Web Technology of VI Semester Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi** during the academic year 2021-22. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department Library. The project report has been approved as it satisfies the academies requirements in respect of project work prescribed for the said degree

Mrs. Savita N
Assistant Professor Dept. of CSE
JSSATE, Bengaluru

Dr. Naveen N.C
Professor and Head Dept of CSE
JSSATE, Bengaluru

Name of the Examiners **Signature with Date**

1)…… ……

2)…… ……

ABSTRACT

The development of technology has increased very rapidly and can be found in almost all areas of human life, one can find one of them in the field of education. Nowadays, education learning is using technology in order to perform the delivery of material will become more interesting, easy and memorable.

Here we have developed a TYPING GAME, which allows the user to type in a selected set of strings to improve their speed and accuracy in typing.
This application uses HTML5, CSS3 and JAVASCRIPT.

We have used various CSS styling features to make the page more appealing to the user and also implemented a toggle button to make the user chose the difficulty of the game, as per their expertise.

ACKNOWLEDGEMENT

We express our humble pranams to his holiness **Jagadguru Sri Sri Sri Shivaratri Deshikendra Mahaswamiji** for showering his blessings onus to receive good education and have a successful career.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of support from all ends during the course of this project's take the opportunity to express our gratitude to all whose guidance and encouragement helped us emerge successfully.

We are thankful for the resourceful guidance, timely assistance and graceful gesture of our guide **Mrs. Savita N**, Assistant Professor, Department of Computer Science and Engineering who has helped us in every aspect of our project work.

We are also forever grateful to **Dr. Naveen N.C**, Head of the Department, Computer Science and Engineering, for his unending support, guidance and encouragement in all our ventures.

We express our sincere thanks to our beloved principal, **Dr. Bhimasen Soragaon** for having supported us in all academic endeavors.

Last but not the least we would be immensely pleased to express our heartfelt thanks to all the teaching and non-teaching staff of the Department of CSE and our friends for their timely help, support and guidance.

Deepak Naidu T (1JS19CS049)

Girish Kumar DV (1JS19CS057)

Table of Contents

Title **Page No.**

Chapter 1

1.1 Introduction

This website is a basic typing game based website which allows the user to choose the level of difficulty from easy or medium or hard as per their expertise and would start playing the game.

Once the user starts playing the game, a set of preset words are randomly picked from the database and the user needs to type in the same word in the space provided for input. Meanwhile time is being reduced.

If the user gets the word right, based on his difficulty, he can get a +4s(easy), +2s(medium), +1(difficult) time boost to his time reserve.

For every correct word input, the user score is incremented by one point.

The game persists until the rate of decrement of time is less than or equal to the rate increment due to the right word input by user.

Once the user runs out of the reserve time, his score is displayed along with a reload button, to replay the game with the desired difficulty.

Web Technology

Web development is the work involved in developing a website for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex web applications, electronic businesses, and social network services. A more comprehensive list of tasks to which Web development commonly refers, may include Web engineering, Web design, Web content development, client liaison, client-side/server-side scripting, Web server and network security configuration, and e-commerce development.

Among Web professionals, "Web development" usually refers to the main non-design aspects of building Web sites: writing markup and coding. Web development may use content management systems (CMS) to make content changes easier and available with basic technical skills.

For larger organizations and businesses, Web development teams can consist of hundreds of people (Web developers) and follow standard methods like Agile methodologies while developing Web sites. Smaller organizations may only require a single permanent or contracting developer, or secondary assignment to related job positions such as a graphic designer or information systems technician.

Web development may be a collaborative effort between departments rather than the domain of a designated department. There are three kinds of Web developer specialization: front-end developer, back-end developer, and full-stack developer. Front- end developers are responsible for behavior and visuals that run in the user browser, while back-end developers deal with the servers.

Since the commercialization of the Web with Tim Berners-Lee developing the World Wide Web at CERN, the industry has boomed and has become one of the most used technologies ever.

Web development tools (often called dev-tools or inspect element) allow web developers to test and debug their code. They are different from website builders and integrated development environments (IDEs) in that they do not assist in the direct creation of a webpage, rather they are tools used for testing the user interface of a website or web application.

Web development tools come as browser add-ons or built-in features in web browsers. Most popular web browsers, such as Google Chrome, Firefox, Internet Explorer, Safari, Microsoft Edge and Opera, have built-in tools to help web developers, and many additional add-ons can be found in their respective plugin download centers.

Web development tools allow developers to work with a variety of web technologies, including HTML, CSS, the DOM, JavaScript, and other components that are handled by the web browser. Due to increasing demand from web browsers to do more, popular web browsers have included more features geared for developers.

Several notable web browsers have support for web developer tools that allow web designers and developers to look at the make-up of their pages. These are all tools that are built into the browser and do not require additional modules or configuration.

- Firefox – F12 opens the Web Console / Browser Console (since Firefox 4). The Web Console applies to a single content tab; the Browser Console applies to the whole browser. Many addons also exist, including Firebug.
- Google Chrome – Chrome Developer Tools (DevTools)
- Internet Explorer and Microsoft Edge – version 8) opens F12 Web Developer Tools (as
- Opera – Opera Dragonfly
- Safari – Safari Web Development Tools (as of version 3)

1.1.1 HTML and DOM

- The Hyper-Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.
- It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.
- Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages.
- HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.
- HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets.
- Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements.

- The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense.

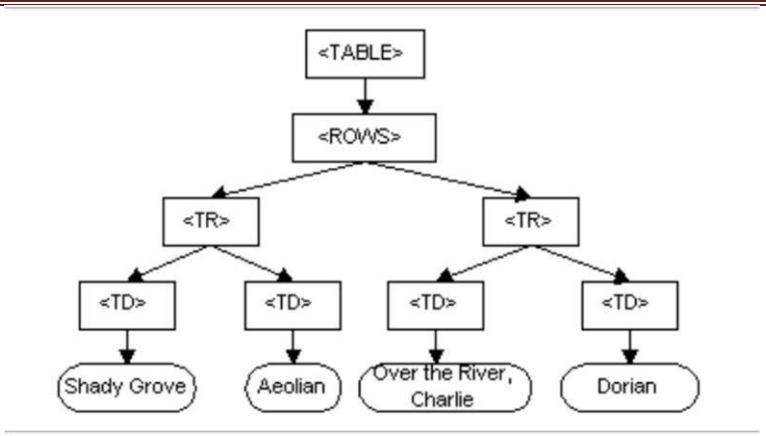
- XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

- With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model.

- The Document Object Model is a programming API for documents. The object model itself closely resembles the structure of the documents it models. For instance, consider this table, taken from an HTML document:

```
<TABLE>
  <ROWS>
   <TR>
    <TD>Shady Grove</TD>
    <TD>Aeolian</TD>
   </TR>
   <TR>
    <TD>Over the River, Charlie</TD>
    <TD>Dorian</TD>
   </TR>
  </ROWS>
</TABLE>
```

- The Document Object Model represents this table like this:



1.1.2 CSS

- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

- CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.

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

- Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille- based tactical devices.

- CSS also has rules for alternate formatting if the content is accessed on a mobile device.

- The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

1.1.3 JAVASCRIPT

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

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

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

1.2 Objectives

- The main objective of the **TYPING GAME** is to make the user be able to type faster and with greater accuracy, along with a gist of competitiveness.
- It makes it really easy to for the user to adapt to the API and wouldn't take a lot of time to be good at the game.
- It creates a perfect place for the user to make use of the keyboard and skill up to be a faster and accurate typist.

1.3 Summary

The chapter discussed before is an overview about the **HTML, DOM, CSS and JAVASCRIPT.** The scope of objectives of the project are mentioned clearly.

Chapter 2

Requirement Specifications

2.1 Software Specification

 Operating System: Windows 10, Mac OS, Linux
HTML, CSS and JAVASCRIPT

2.2 Hardware Specification

- Processor: x86 compatible processor with 1.7 GHz Clock Speed
- RAM: 512 or greater
- Hard Disk:20 GB or greater
- Monitor: VGA/SVGA
- Keyboard: 104 keys standard Mouse: 2/3 button. Optical/Mechanical

2.3 User Characteristics

- Should be comfortable with basic working of the computer.
- Must have basic knowledge of English.

Chapter 3

Implementation

3.1 Source Code

Source Code is an informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading.

In our webapp all the required files are stored in the same directory containing the HTML ,CSS and JS files.

HTML Source Code

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.11.2/css/all.min.css"
      integrity="sha256-+N4/V/SbAFiW1MPBCXnfnP9QSN3+Keu+NlB+0ev/YKQ="
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="styles.css" />
    <title>Speed Typer</title>
  </head>
  <body>
    <button id="settings-btn" class="settings-btn">
      <i class="fas fa-cog"></i>
    </button>

    <div id="settings" class="settings">
      <form id="settings-form">
```

```
        <div>
          <label for="difficulty">Difficulty</label>
          <select id="difficulty">
            <option value="easy">Easy</option>
            <option value="medium">Medium</option>
            <option value="hard">Hard</option>
          </select>
        </div>
      </form>
    </div>

    <div class="container">
      <h2>👩‍💻 Speed Typer 👨‍💻</h2>
      <small>Type the following:</small>

      <h1 id="word"></h1>

      <input
        type="text"
        id="text"
        autocomplete="off"
        placeholder="Type the word here..."
        autofocus
      />

      <p class="time-container">Time left: <span id="time">10s</span></p>

      <p class="score-container">Score: <span id="score">0</span></p>

      <div id="end-game-container" class="end-game-container"></div>
    </div>

    <script src="script.js"></script>
  </body>
</html>
```

CSS Source Code

```css
* {
  box-sizing: border-box;
}

body {
  background-color: #2c3e50;
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  margin: 0;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
}

button {
  cursor: pointer;
  font-size: 14px;
  border-radius: 4px;
  padding: 5px 15px;
}

select {
  width: 200px;
  padding: 5px;
  appearance: none;
  -webkit-appearance: none;
  -moz-appearance: none;
  border-radius: 0;
  background-color: #a7c5e3;
}

select:focus,
button:focus {
  outline: 0;
}

.settings-btn {
  position: absolute;
  bottom: 30px;
  left: 30px;
```

```
}

.settings {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  background-color: rgba(0, 0, 0, 0.3);
  height: 70px;
  color: #fff;
  display: flex;
  align-items: center;
  justify-content: center;
  transform: translateY(0);
  transition: transform 0.3s ease-in-out;
}

.settings.hide {
  transform: translateY(-100%);
}

.container {
  background-color: #34495e;
  padding: 20px;
  border-radius: 4px;
  box-shadow: 0 3px 5px rgba(0, 0, 0, 0.3);
  color: #fff;
  position: relative;
  text-align: center;
  width: 500px;
}

h2 {
  background-color: rgba(0, 0, 0, 0.3);
  padding: 8px;
  border-radius: 4px;
```

```
  margin: 0 0 40px;
}

h1 {
  margin: 0;
}

input {
  border: 0;
  border-radius: 4px;
  font-size: 14px;
  width: 300px;
  padding: 12px 20px;
  margin-top: 10px;
}

.score-container {
  position: absolute;
  top: 60px;
  right: 20px;
}

.time-container {
  position: absolute;
  top: 60px;
  left: 20px;
}

.end-game-container {
  background-color: inherit;
  display: none;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  z-index: 1;
}
```

JAVASCRIPT Source Code

```
const word = document.getElementById('word');
const text = document.getElementById('text');
const scoreEl = document.getElementById('score');
const timeEl = document.getElementById('time');
const endgameEl = document.getElementById('end-game-container');
const settingsBtn = document.getElementById('settings-btn');
const settings = document.getElementById('settings');
const settingsForm = document.getElementById('settings-form');
const difficultySelect = document.getElementById('difficulty');

// List of words for game
const words = [
  'sigh',
  'tense',
  'airplane',
  'ball',
  'pies',
  'juice',
  'warlike',
  'bad',
  'north',
  'dependent',
  'steer',
  'silver',
  'highfalutin',
  'superficial',
  'quince',
  'eight',
  'feeble',
  'admit',
  'drag',
  'loving'
];
```

```
// Init word
let randomWord;

// Init score
let score = 0;

// Init time
let time = 10;

// Set difficulty to value in ls or medium
let difficulty =
  localStorage.getItem('difficulty') !== null
    ? localStorage.getItem('difficulty')
    : 'medium';

// Set difficulty select value
difficultySelect.value =
  localStorage.getItem('difficulty') !== null
    ? localStorage.getItem('difficulty')
    : 'medium';

// Focus on text on start
text.focus();

// Start counting down
const timeInterval = setInterval(updateTime, 1000);

// Generate random word from array
function getRandomWord() {
  return words[Math.floor(Math.random() * words.length)];
}

// Add word to DOM
function addWordToDOM() {
  randomWord = getRandomWord();
  word.innerHTML = randomWord;
}

// Update score
function updateScore() {
  score++;
  scoreEl.innerHTML = score;
}
```

```
// Update time
function updateTime() {
  time--;
  timeEl.innerHTML = time + 's';

  if (time === 0) {
    clearInterval(timeInterval);
    // end game
    gameOver();
  }
}

// Game over, show end screen
function gameOver() {
  endgameEl.innerHTML = `
    <h1>Time ran out</h1>
    <p>Your final score is ${score}</p>
    <button onclick="location.reload()">Reload</button>
  `;

  endgameEl.style.display = 'flex';
}

addWordToDOM();

// Event listeners

// Typing
text.addEventListener('input', e => {
  const insertedText = e.target.value;

  if (insertedText === randomWord) {
    addWordToDOM();
    updateScore();
```

```
    // Clear
    e.target.value = '';

    if (difficulty === 'hard') {
      time += 2;
    } else if (difficulty === 'medium') {
      time += 3;
    } else {
      time += 5;
    }

    updateTime();
  }
});

// Settings btn click
settingsBtn.addEventListener('click', () =>
settings.classList.toggle('hide'));

// Settings select
settingsForm.addEventListener('change', e => {
  difficulty = e.target.value;
  localStorage.setItem('difficulty', difficulty);
});
```

Chapter 4 Results

The project is compiled and executed using **Visual Studio Code.** We have put in few screenshots here to show the working of Typing Game Website. Dark Mode

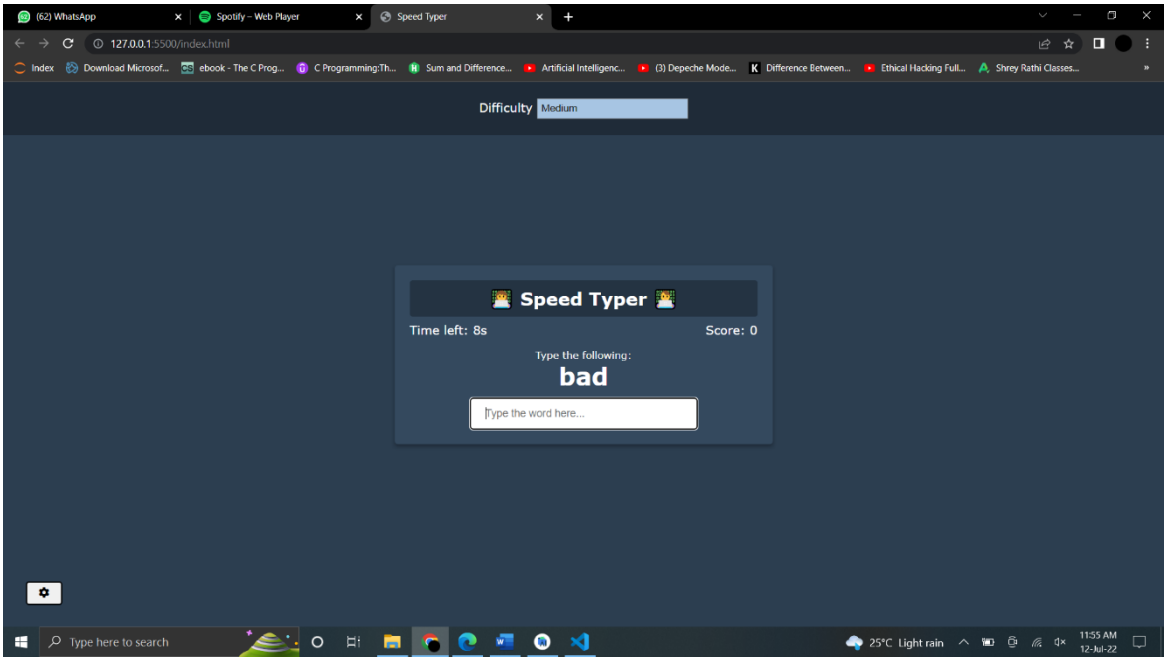


Fig 4(a) : Starting page when we run the website server.

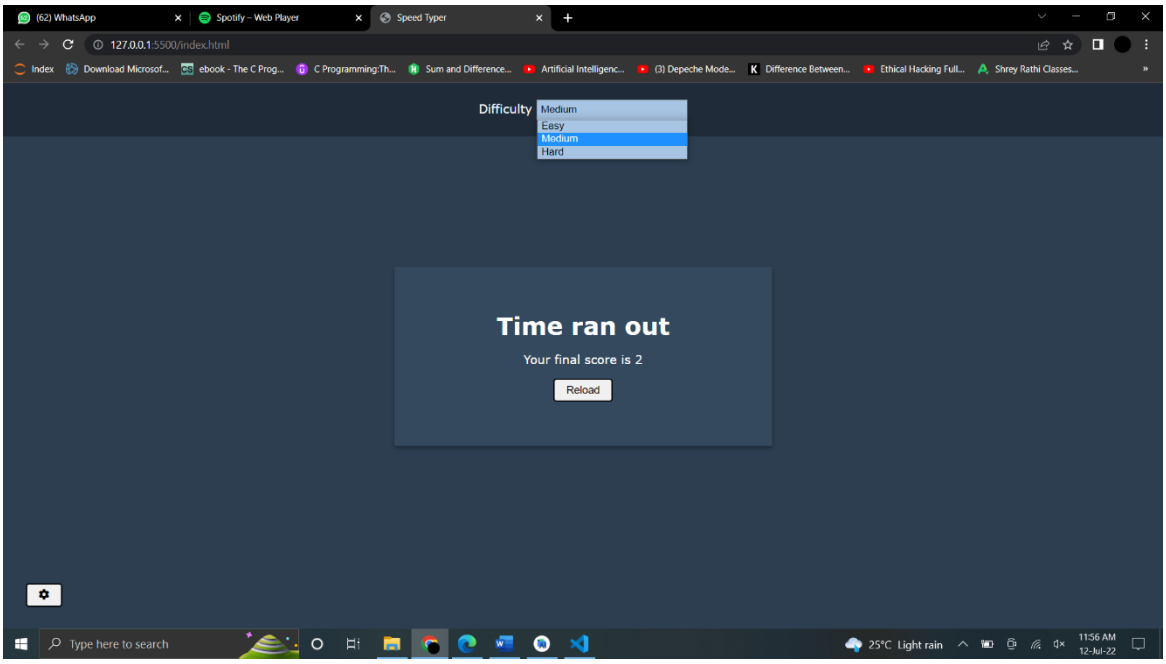


Fig 4(b) : User getting an option to choose the level of difficulty.

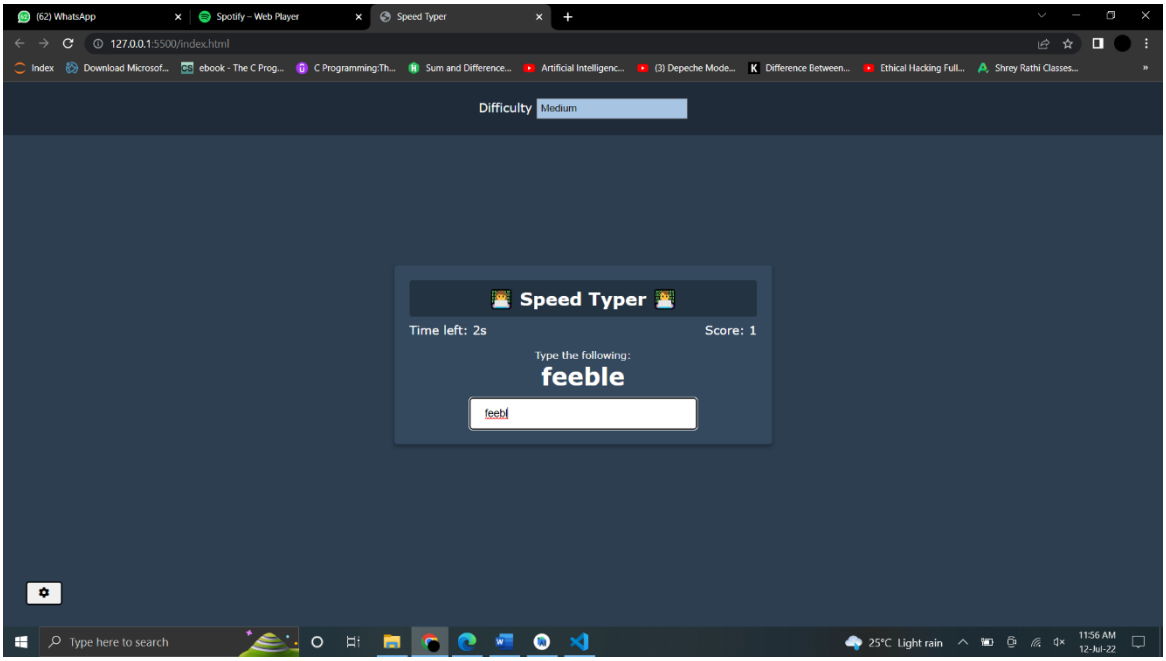


Fig 4(c): User typing in the required word to get a time boost.

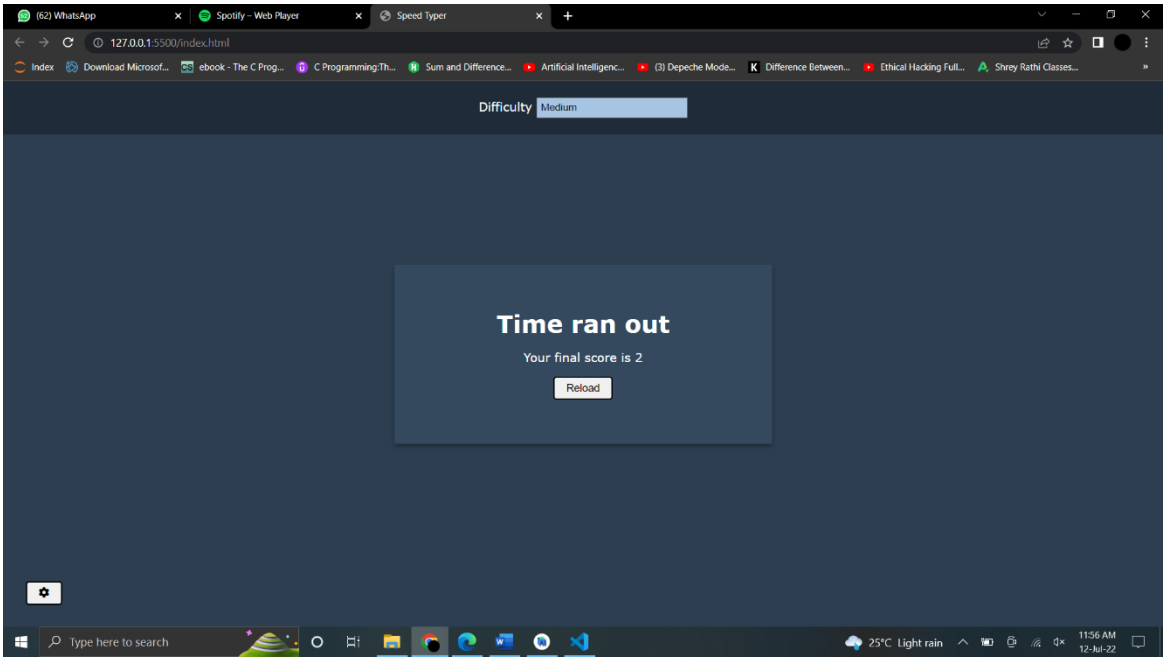


Fig 4(d) : Once user runs out of time, score is displayed and button to restart game is provided. The number of correct words typed by user is their score.

Conclusion

In today's Web development, a good page design is essential. A bad design will lead to the loss of visitors and that can lead to a loss of business. In general, a good page layout has to satisfy the basic elements of a good page design. This includes color contrast, text organization, font selection, style of a page, page size, graphics used, and consistency.

In order to create a well-designed page for a specific audience. The developer needs to organized and analyze the user's statistics and the background of the users. Although it can be hard to come up with a design that is well suited to all of the users, there will be a design that is appropriate for most of the audience. The better the page design, the more hits a page will get. That implies an increase in accessibility and a possible increase in business.

If we had to pick one theme for this year's survey, it would be new beginnings. A new survey in a brand new domain for us; new features for CSS and a newfound respect for CSS's ever-improving capabilities from the community.

To be fair, CSS has seen its share of doubters as well. The growing importance given to JavaScript over the past few years can sometimes leave some feeling like HTML & CSS are becoming second-tier players, and like their own skills in these fields aren't as valued anymore.

But the data tells another story. While JavaScript isn't going anywhere, it also seems certain that CSS is not going to be replaced anytime soon. And the more powerful CSS becomes, the more valuable mastering it will be.

And make no mistake, CSS is definitely evolving. Most of the features covered in the survey are far from mainstream yet, and upcoming features such as subgrids and aspect ratios are also sure to keep us busy.