



TITLE:

**QUIZ Application Using HTML, CSS, and
JavaScript**

BY:

Neelam Vazirani
Batch- EON(2021-5736)

UNDER THE GUIDANCE OF:

Mrs. Amruta Deore
(Technical Trainer)
Edubridge India Pvt Ltd

Contents

1.0 Aim /Benefits of the Project

2.0 Methodology

3.0 Resources Required

4.0 Rationale

5.0 Literature Review

5.1 JavaScript

- Features
- Variables
- Comments
- Operators
- Conditionals
- Functions

5.2 HTML

5.3 CSS

6.0 Flow Chart

7.0 Algorithm

8.0 DFD Diagram

9.0 ER Diagram

10.0 Code

11.0 Output

12.0 Conclusion

13.0 Reference

1.0 Aim /Benefits of the Project:

This Project aims at:

1. To give dynamism and interactive website using JavaScript.
2. Add styles using Cascading Style Sheet (CSS).
3. To give semantics to the program using HTML.

2.0 Methodology:

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) I have covered in much more detail in other parts of the Learning Area.

- HTML is the markup language that we use to structure and give meaning to our web content, for example defining paragraphs, headings, and data tables, or embedding images and videos in the page.
- CSS is a language of style rules that we use to apply styling to our HTML content, for example setting background colors and fonts, and laying out our content in multiple columns.
- JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

3.0 Resources Required

S. No.	Name of Resource/material	Specifications
1	Hardware computer system	Computer (i3-i5 preferable),RAM minimum 8 GB and onwards but not limited to
2	Operating system	Windows XP/windows 7/ Linux version 5 or later
3	Software	Notepad++ and any browser

5.0 Rationale

Importance: The importance of JavaScript as a web technology can be determined from the fact that it is currently used by 94.5% of all websites. As a client-side programming language, JavaScript helps web developers to make web pages dynamic and interactive by implementing custom client-side scripts. At the same time, the developers can also use cross-platform runtime engines like Node.js to write server-side code in JavaScript. They can even combine JavaScript, HTML5 and CSS3 to create web pages that look good across browsers, platforms, and devices. There are also a number of reasons why each modern web developer must know how to leverage all benefits of JavaScript.

5.0 Literature Review

❖ JavaScript:

JavaScript is a programming language that adds interactivity to your website. This happens in games, in the behavior of responses when buttons are pressed or with data entry on forms; with dynamic styling; with animation, etc. This article helps you get started with JavaScript and furthers your understanding of what is possible.

JavaScript ("JS" for short) is a full-fledged dynamic programming language that can add interactivity to a website. It was invented by Brendan Eich (co-founder of the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation).

JavaScript is versatile and beginner-friendly. With more experience, you'll be able to create games, animated 2D and 3D graphics, comprehensive database-driven apps, and much more!

JavaScript itself is relatively compact, yet very flexible. Developers have written a variety of tools on top of the core JavaScript language, unlocking a vast amount of functionality with minimum effort. These include:

- Browser Application Programming Interfaces (APIs) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.
- Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as Twitter or Facebook.
- Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications.

❖ Features of Java:

The primary objective of Java programming language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. A list of the most important features of the Java language is given below.

- **Light Weight Scripting language:** JavaScript is a lightweight scripting language because it is made for data handling at the browser only. Since it is not a general-purpose language so it has a limited set of libraries. Also, as it is only meant for client-side execution and that too for web applications, hence the lightweight nature of JavaScript is a great feature.
- **Dynamic Typing:** JavaScript supports dynamic typing which means types of the variable are defined based on the stored value. For example, if you declare a variable `x` then you can store either a string or a Number type value or an array or an object. This is known as dynamic typing.
- Object-oriented programming support
- **Functional Style:** Starting from ES6, the concept of class and OOPs has been more refined. Also, in JavaScript, two important principles with OOP in JavaScript are Object Creation patterns (Encapsulation) and Code Reuse patterns (Inheritance). Although JavaScript developers rarely use this feature but its there for everyone to explore.
- **Platform Independent:** This implies that JavaScript is platform-independent or we can say it is portable; which simply means that you can simply write the script once and run it anywhere and anytime. In general, you can write your JavaScript applications and run them on any platform or any browser without affecting the output of the Script.
- **Prototype-based:** JavaScript is a prototype-based scripting Language. This means javascript uses prototypes instead of classes or inheritance. In languages like Java, we create a class and then we create objects for those classes. But in JavaScript, we define object prototype and then more objects can be created using this object prototype.
- **Interpreted Language:** JavaScript is an interpreted language which means the script written inside JavaScript is processed line by line. These Scripts are interpreted by JavaScript interpreter which is a built-in component of the Web browser. But these days many JavaScript engines in browsers like the V8 engine in chrome uses just in time compilation for JavaScript code.

- **Async Processing:** JavaScript supports Promise which enables asynchronous requests wherein a request is initiated and JavaScript doesn't have to wait for the response, which at times blocks the request processing. Also starting from ES8, Async functions are also supported in JavaScript, these functions don't execute one by one, rather they are processed parallelly which has a positive effect on the processing time, reducing it to a great extent.
- **Client-Side Validation:** This is a feature which is available in JavaScript since forever and is still widely used because every website has a form in which users enter values, and to make sure that users enter the correct value, we must put proper validations in place, both on the client-side and on the server-side. JavaScript is used for implementing client-side validations.

❖ Variables:

Variables are containers that store values. You start by declaring a variable with the [var](#) (less recommended, dive deeper for the explanation) or the [let](#) keyword, followed by the name you give to the variable:

Syntax:

```
let myVariable;
```

❖ Comments:

Comments are snippets of text that can be added along with code. The browser ignores text marked as comments. You can write comments in JavaScript just as you can in CSS:

Syntax:

```
/*
```

Everything in between is a comment.

```
*/
```

❖ Operators:

An operator is a mathematical symbol that produces a result based on two values (or variables).

1. Addition: Add two numbers together or combine two strings.
2. Subtraction, Multiplication, Division: These do what you'd expect them to do in basic math.
3. Assignment: This assigns a value to a variable.
4. Equality: This performs to see if two values are equal. It returns Boolean result.

❖ Conditionals:

Conditionals are code structures used to test if an expression returns true or not. A very common form of conditionals is the if ... else statement.

❖ Functions:

Functions are a way of packaging functionality that you wish to reuse. It's possible to define a body of code as a function that executes when you call the function name in your code. This is a good alternative to repeatedly writing the same code.

1. `let myVariable = document.querySelector('h1');`
2. `alert('hello!');`

These functions, `document.querySelector` and `alert`, are built into the browser.

❖ Events

Real interactivity on a website requires event handlers. These are code structures that listen for activity in the browser, and run code in response. The most obvious example is handling the [click event](#), which is fired by the browser when you click on something with your mouse.

❖ HTML

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

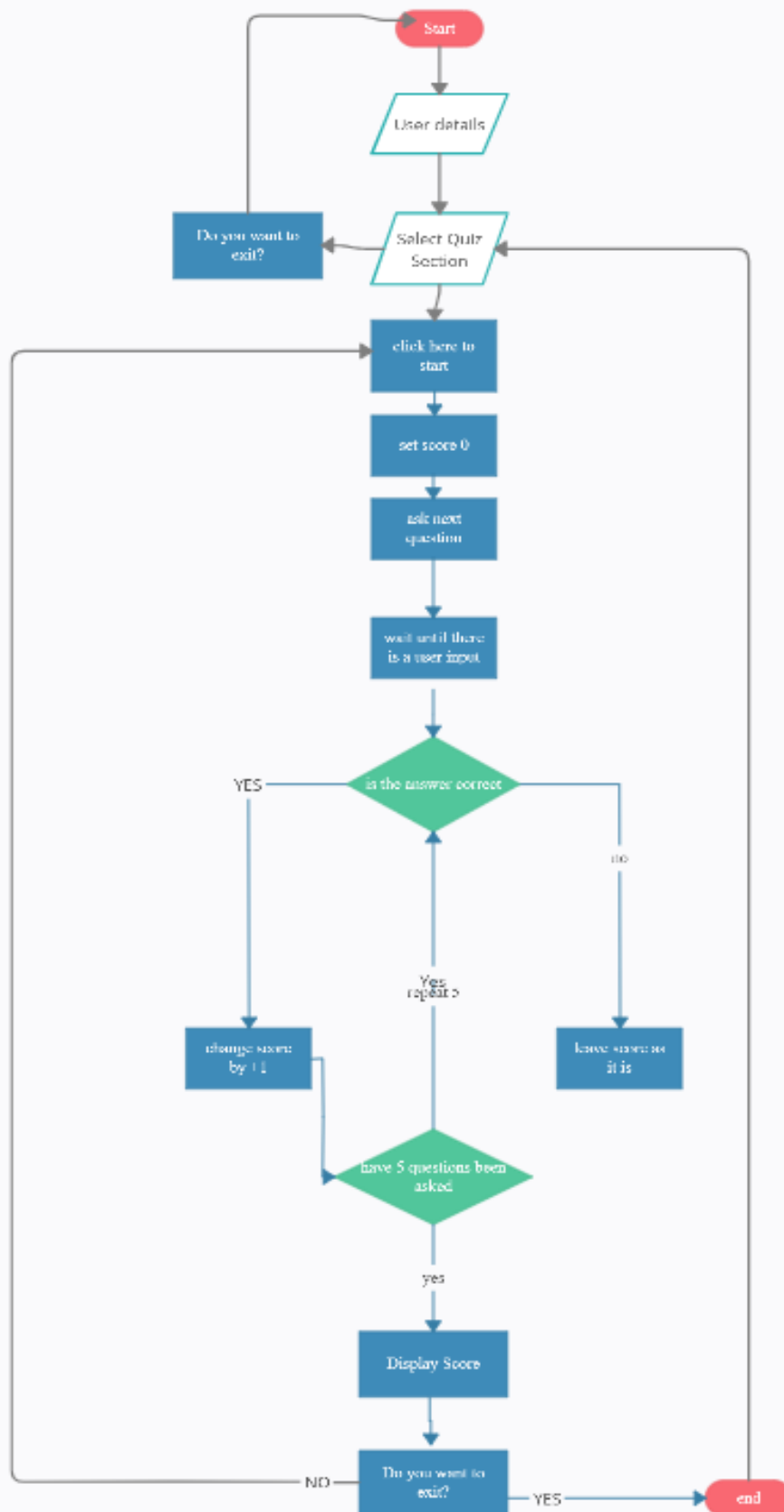
❖ CSS

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

- 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

6.0 Flow Chart:



7.0 Algorithm:

Step 1: Start

Step 2: Display employee full name

Step 3: Select the quiz section

Step 3.1: If the user selects HTML then questions related to this topic will be displayed.

Step 3.2: If the user selects CSS then questions related to this topic will be displayed.

Step 3.3 If the user selects JAVA then questions related to this topic will be displayed.

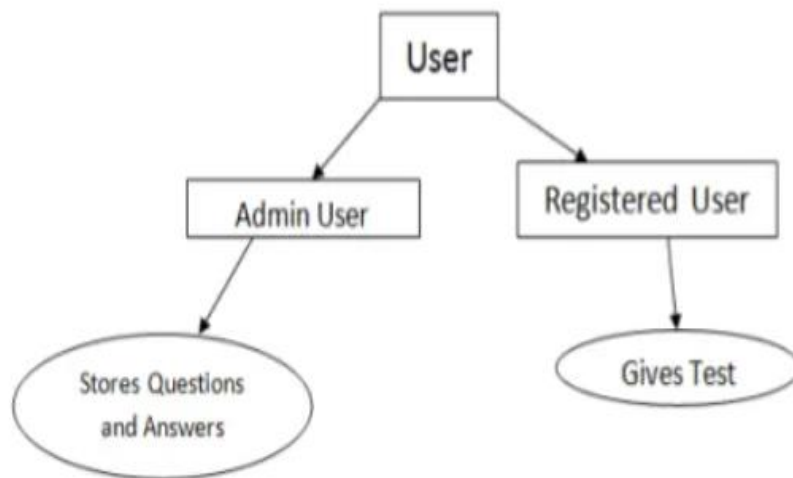
Step 3.4: If the user selects SQL then questions related to this topic will be displayed.

Step 4: If the user selects to go back then the quiz page will be displayed.

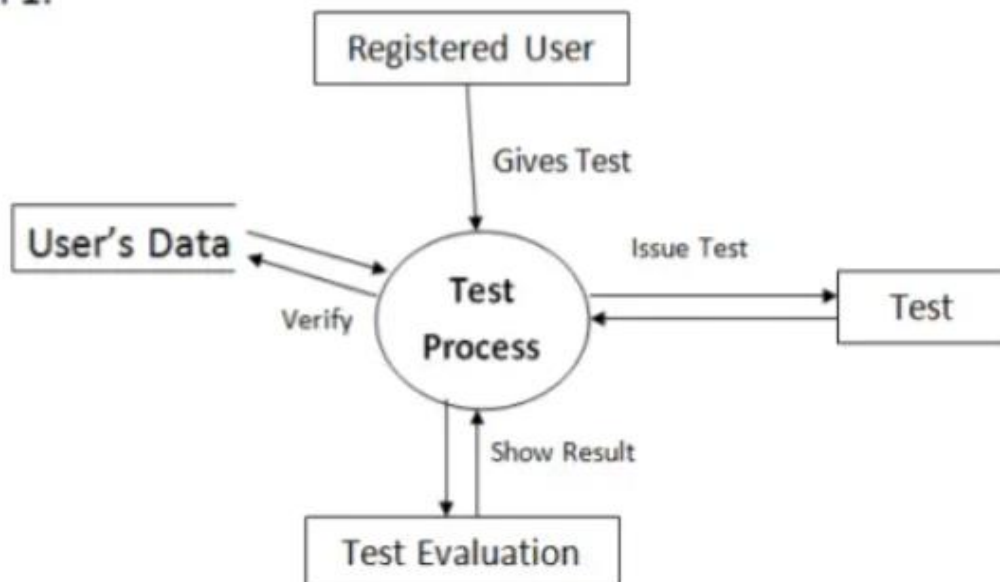
Step 5: Stop

8. DFD diagram

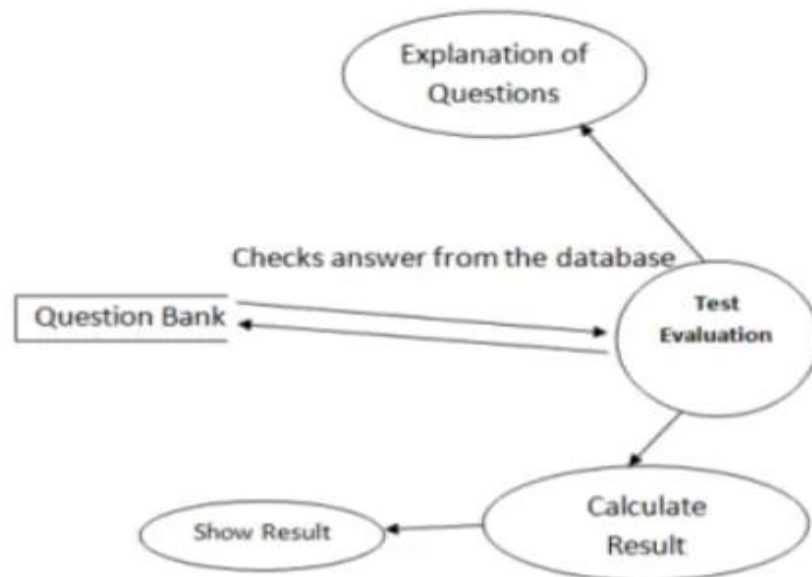
Level 0:



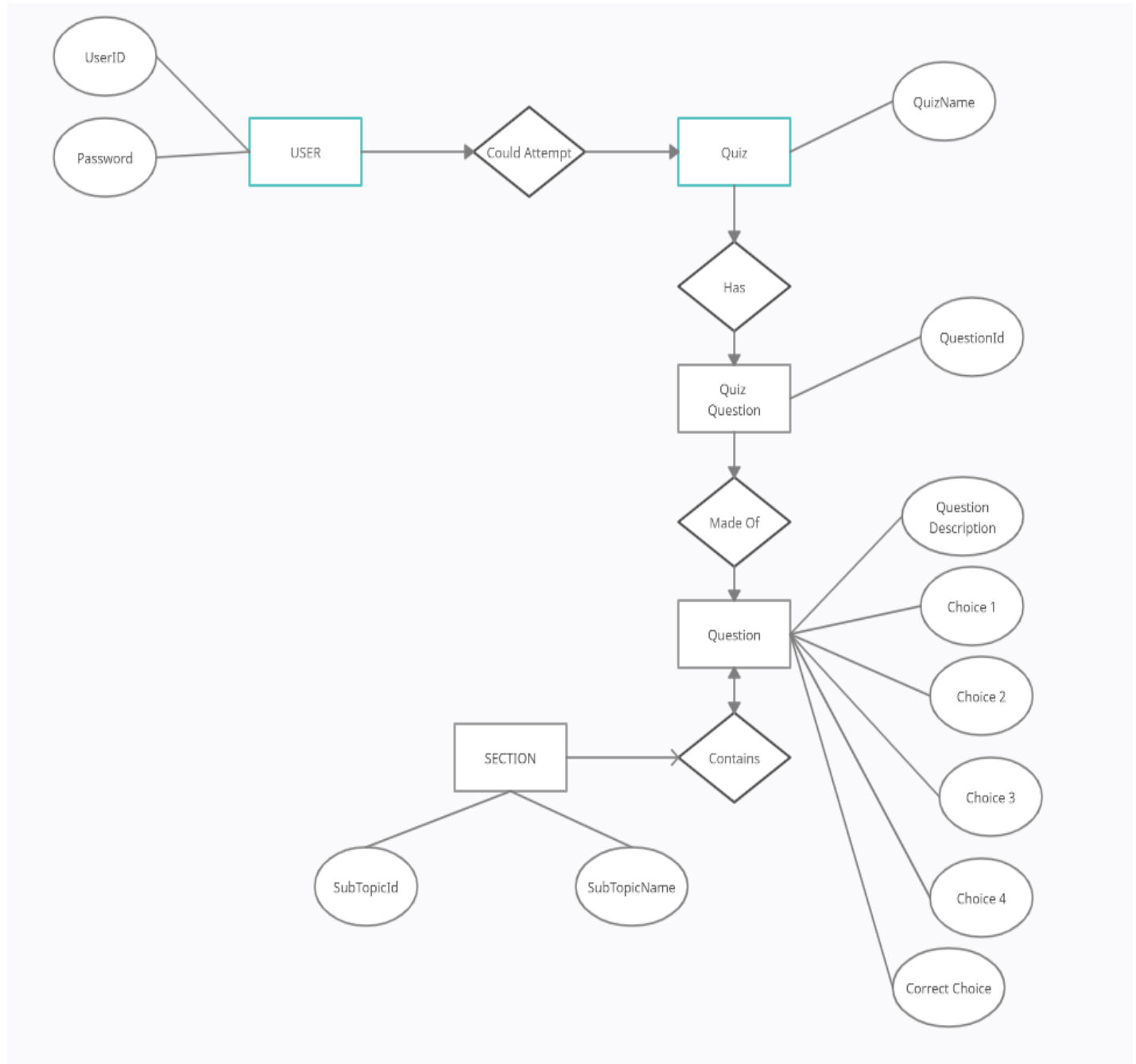
Level 1:



Level 2:



9.ER Diagram



10 Code:

Module1: First Page

1. Firstpage.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Quiz Game!</title>
<link rel="stylesheet" href="firstpage.css">
</head>
<body>
<div class="container">

<a href="Login.html" </a>
<button class="btn"> Let's Go!</button>
</div>
</body>
</html>
```

2. Firstpage.css File

```
/* Container needed to position the button. Adjust the width as needed */
.container {
    position: center;
    width: 100%;
}

/* Make the image responsive */
.container img {
    width: 100%;
}

/* Style the button and place it in the middle of the container/image */
.container .btn {
    position: absolute;
    top: 40%;
    left: 50%;
    transform: translate(-50%, -50%);
    -ms-transform: translate(-50%, -50%);
    background-color: #ff52ff;

    font-size: 46px;
    padding: 12px 24px;
    border: none;
    cursor: pointer;
    border-radius: 15px;
}

.container .btn:hover {
    background-color: #ff1a8c;
}
```

Module2: Login Page

1. Login.html

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome To Login Form</title>
<link rel="stylesheet" href="Login.css">
</head>

<body>
<!-- Main div code -->
<div id="main">
<div class="h-tag">
<h2>Welcome To Login Page</h2>
</div>
<!-- Login box -->
<div class="login">
<table cellspacing="2" align="center" cellpadding="8" border="0">
<tr>
<td>Enter User Name :</td>
<td><input type="text" placeholder="Enter user name here" id="email" class="tb"
/></td>
</tr>
<tr>
<td>Enter Password :</td>
<td><input type="password" placeholder="Enter Password here" id="pwd1" class="tb"
/></td>
</tr>
<tr>
<td></td>
<td>
<input type="submit" value="Reset" onclick="clearFunc()" class="btn" />
<input type="submit" value="Login" class="btn" onClick="login()" /></td>
</tr>
</table>
</div>
<!-- login box div ending here.. -->
</div>
<!-- Main div ending here... -->
<script src="Login.js"></script>
</body>
</html>
```


2. Login.css

```
body
{
    margin:0px;
    background-color: #f0ffff;
    font-family:Arial, Helvetica, sans-serif;
}
#main
{
    width:600px;
    height:260px;
    margin-left:auto;
    margin-right:auto;
    border-radius:5px;
    padding-left:10px;
    margin-top:100px;
    border-top:3px double #f1f1f1;
    border-bottom:3px double #f1f1f1;
    padding-top:20px;
    background-color: #7fffd4;
}
#main table
{
    font-family:"Comic Sans MS", cursive;
}
/* css code for textbox */
#main .tb
{
    height:28px;
    width:230px;
    border:1px solid #27a465;
    background-color:#f0ffff;
    font-weight:bold;
    border-left:5px solid #f7f7f7;
    opacity:0.9;
}

/* css code for button*/
#main .btn
{
    width:80px;
    height:32px;
    font-weight:bold;
    border:0px solid #27a465;
    text-shadow: 0px 0.5px 0.5px #fff;
    border-radius: 2px;
    font-weight: 600;
    color: #27a465;
    letter-spacing: 1px;
    font-size:14px;

    transition: 1s;
```

```

}

#main .btn:hover
{
  background-color:#27a465;
  border-radius: 2px;
  color:#f1f1f1;
  border:1px solid #f1f1f1;
}

```

3. Login.js

```

function login()
{
  var uname = document.getElementById("email").value;
  var pwd = document.getElementById("pwd1").value;
  var filter = /^[a-zA-Z0-9_\.\-]+\@(([a-zA-Z0-9\-\])+\.)+([a-zA-Z0-9]{2,4})+$/;
  if(uname == '')
  {
    alert("Please enter user name");
  }
  else if(pwd=='')
  {
    alert("enter the password");
  }
  else if(!filter.test(uname))
  {
    alert("Enter valid email id.");
  }
  else if(pwd.length < 6 || pwd.length > 6)
  {
    alert("Password range is equal to 6.");
  }
  else
  {
    alert('Thank You for your details!\n Successfull Login');
    //Redirecting to other page or webste code or you can set your own html page.
    window.location = "file:///D:/HTML%20PROGRAMS/index.html";
  }
}

//Reset Inputfield code.
function clearFunc()
{
  document.getElementById("email").value="";
  document.getElementById("pwd1").value="";
}

```

Module3: Select Section

1. index.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>home page</title>
  <link rel="stylesheet" href="index.css">

</head>
<body>
<div class="container">
<div id="home" class="flex-column flex-center">
<h1>Pick a topic</h1>
<a href="rulesHTML.html" </a>
<button id="html-btn" class="btn">1. HTML QUIZ SECTION</button>

<a href="rulesCSS.html" </a>
<button id="css-btn" class="btn">2. CSS QUIZ SECTION</button>

<a href="rulesJAVA.html" </a>
<button id="java-btn" class="btn">3. JAVA QUIZ SECTION</button>

<a href="rulesSQL.html" </a>
<button id="sql-btn" class="btn">4. SQL QUIZ SECTION</button>

<a href="firstpage.html" </a>
<button id="sql-btn" class="btn">Go back</button>
</div>
</div>
</body>
</html>
```

2. index.css File

```
*{
  box-sizing: border-box;
  margin:0;

  width: 50vw;
  background-color: #7fffd4;
  border-radius: 10px;
  align-items:center;
}
.container {
  width:100vw;
  min-height:150vh;
  display:grid;
  place-items: center;
  background-color: #f0ffff;
}
.flex-column , h1{
  display:flex;
  flex-direction: column;
  align-items:center;
}
.flex-center, h1{
  justify-content:center;
  align-items: center;
}
.btn{
  font-size:20px;
  padding: 20px;
  text-align:center;
  margin-top:20px;
  cursor: pointer;
  background-color:#00ff00;
}
.btn:hover{
  cursor: hover;
  background-color:#ffa07a;
  transform:scale(1.03) ;
}
.hidden{
  display:none;
}
.btn[disabled]:hover{
  cursor: not-allowed;
  transform:none;
}
```

Module4: HTML

1. rulesHTML.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>HTML Rules Page</title>
<link rel="stylesheet" href="rulesHTML.css">
</head>

<body>
<div class="container">
<div id="home" class="flex-column flex-center">

<div class="info-title"><h1>Welcome to HTML Quiz</h1>
<h2>Here are some rules:</h2>
</div>
    <div class="info-list">
        <div class="info">1. The quiz contains multiple choice questions
with four options each.</div>
        <div class="info">2. You cannot change the selected option once
clicked on 'submit' button.</div>
        <div class="info">3. Click on 'submit' to submit answer of each
question</div>
        <div class="info">4. You'll get points on the basis of your correct
answers.</div>
    </div>
<a href="HTML.html" </a>
<button id="btn" class="btn">Continue</button>
</div>
</div>
</body>
</html>
```

2. rulesHTML.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;7
00&display=swap');
*{
    width: 50vw;
    background-color: #7fffd4;
    border-radius: 10px;
    align-items:center;
}
.container {
width:100vw;
    min-height:100vh;
    display:grid;
    place-items: center;
    background-color: #f0ffff;
}
.flex-column , h1{
    display:flex;
    flex-direction: column;
    align-items:center;
}
.flex-center, h1{
justify-content:center;
align-items: center;
}
.btn{
    font-size:20px;
    padding: 20px;
    text-align:center;
    margin-top:20px;
    cursor: pointer;
    background-color:#00ff00;
}

.btn:hover{
    cursor: hover;
    background-color:#ffa07a;
}
.hidden{
    display:none;
}

html{
    font-size: 120%;
}
.start_btn,
.info_box,
.quiz_box,
.result_box{
    position: absolute;
    top: 50%;
    left: 50%;
```

```
    transform: translate(-50%, -50%);
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),
                0 6px 20px 0 rgba(0, 0, 0, 0.19);
}
.info_box{
    width: 540px;
    background: #fff;
    border-radius: 5px;
    transform: translate(-50%, -50%) scale(0.9);
    opacity: 0;
    pointer-events: none;
    transition: all 0.3s ease;
}
.info_box .info-title{
    height: 60px;
    width: 100%;

    display: flex;
    align-items: center;
    padding: 20 30px;
    border-radius: 5px 5px 0 0;
    font-size: 20px;
    font-weight: 600;
}
.info_box .info-list{
    padding: 15px 30px;
}
.info_box .info-list .info{
    margin: 5px 5px;
    font-size: 17px;
}
.info_box .info-list .info span{
    font-weight: 600;
    color: #007bff;
}
.info_box .buttons{
    height: 60px;
    display: flex;
    align-items: center;
    justify-content: flex-end;
    padding: 0 30px;
    border-top: 1px solid lightgrey;
}
.info_box .buttons button{
    margin: 0 5px;
    height: 40px;
    width: 100px;
    font-size: 16px;
    font-weight: 500;
    cursor: pointer;
    border: none;
    outline: none;
    border-radius: 5px;
    border: 1px solid #007bff;
    transition: all 0.3s ease;
}
```

3. HTML.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Quiz Game!</title>
<link rel="stylesheet" href="HTML.css">
</head>
<body>
<div class="main-div">
  <div class="inner-div">
    <h2 class="question">Questions come here</h2>
    <ul>
      <li>
        <input type="radio" name="answer" id="ans1" class="answer">
        <!--the name should be same for all the lists since when we click on
the radio button then the other button should get deselected.-->
        <label for="ans1" id="option1">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans2" class="answer">
        <label for="ans2" id="option2">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans3" class="answer">
        <label for="ans3" id="option3">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans4" class="answer">
        <label for="ans4" id="option4">Answer option</label>
      </li>
    </ul>
    <button id="submit" >submit</button>
    <div id="showScore" class="scoreArea"></div>
    <button id="exit" onclick="openWin()">Exit</button>
  </div>
</div>
<script src="HTML.js"></script>
</body>
</html>
```


4. HTML.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
    margin: 0;
    padding:0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
    font-weight: 300;
}
html{
    font-size: 120%;
}

.main-div{
    width: 100vw;
    min-height:100vh;
    display:grid;
    place-items: center;
    background-color: #f0ffff;
}

.inner-div{
    width: 50vw;
    background-color: #7fffd4;
    padding: 22px 22px 30px 30px;
    border-radius: 10px;
    box-shadow: 5px 10px 10px 5px #dcdcdc;
}

.inner-div h2{
    font-size:35px;
    font-weight: 400;
    margin: 1px 0 4px 0;
}

.inner-div li{
    font-size:15px;
    margin-top:10px;
    list-style: none;
}

input{
    cursor: pointer;
}

#submit, .btn{
    padding: 10px 15px;
    outline:none;
    font-size:15px;
    margin:auto;
    display: block;
    text-transform:uppercase;
    margin-top:20px;
    cursor: pointer;
}

#submit:hover{
```

```

        background-color:#ffa07a;
    }
    #exit, .btn{
        padding: 10px 15px;
        outline:none;
        font-size:15px;
        margin:auto;
        display: block;
        text-transform:uppercase;
        margin-top:20px;
        cursor: pointer;
    }
    #exit:hover{
        background-color:#ffa07a;
    }
    #showScore{
        background-color:#00ff00;
        margin-top:20px;
        padding:10px 15px;
    }
    #showScore h3{
        font: 30px;
        text-align: center;
    }
    #showScore .btn{
        margin-top: 20px;
        background-color: #adff2f;
    }
    #showScore .btn:hover{
        background-color:#ff69b4;
    }
    .scoreArea{
        display: none;
    }

```

5. HTML.js File

```

// creating an array and passing the number, questions, options, and answers
const quizDB = [
    {
        question: "Q1) What does HTML stand for?",
        a: "Hyper Text Preprocessor",
        b: "Hyper Text Markup Language",
        c: "Hyper Text Multiple Language",
        d: "Hyper Tool Multi Language",
        ans:"ans2"
    },

    {
        question: "Q2) Which of the following element is responsible for making the
text italic in HTML?",
        a: "i tag",
        b: "italic tag",
        c: "it tag",
        d: "pre tag",
    }

```

```

    ans:"ans1"
  },

  {
    question: "Q3) The correct sequence of HTML tags for starting a webpage is -",
    a: "Head, Title, HTML, body",
    b: "HTML, Body, Title, Head",
    c: "HTML, Title, Head, Body",
    d: "HTML, Head, Title, Body",
    ans:"ans4"
  },

  {
    question: "Q4) Which of the following element is responsible for making the text bold in HTML?",
    a: "pre tag",
    b: "a tag",
    c: "b tag",
    d: "br tag",
    ans:"ans3"
  },

  {
    question: "Q5) Which of the following tag is used for inserting the largest heading in HTML?",
    a: 'h3 tag',
    b: "h1 tag",
    c: "h5 tag",
    d: "h6 tag",
    ans:"ans2"
  }
];

const question = document.querySelector('.question'); //for class we use .
const option1 = document.querySelector('#option1'); //for ids we use #
const option2 = document.querySelector('#option2');
const option3 = document.querySelector('#option3');
const option4 = document.querySelector('#option4');
const submit = document.querySelector('#submit');
const answers = document.querySelectorAll('.answer'); //taking reference from input
const showScore = document.querySelector('#showScore');
let questionCount = 0;
let score=0;
//defining function loadQuestion()
const loadQuestion=()=> {
  const questionList = quizDB[questionCount];
  question.innerHTML = questionList.question;
  option1.innerHTML = questionList.a;
  option2.innerHTML = questionList.b;
  option3.innerHTML = questionList.c;
  option4.innerHTML = questionList.d;
}

```

```

//calling function loadQuestion()
loadQuestion();
//defining function getCheckAnswer()
const getCheckAnswer={() => {
let answer;
  answers.forEach((curAnsElem) => {
    if(curAnsElem.checked){
      answer = curAnsElem.id;
    }
  });
  return answer;
};
const deselectAll = () =>{
  answers.forEach((curAnsElem) => curAnsElem.checked = false);
};
var myWindow;

function openWin() {
  myWindow = window.open("index.html");
}
submit.addEventListener('click', ()=>{
  const checkedAnswer = getCheckAnswer();
  console.log(checkedAnswer);

  if(checkedAnswer== quizDB[questionCount].ans)
  {
    score++;
  };
  questionCount++;
  deselectAll();
  if(questionCount<quizDB.length){
    loadQuestion();
  }
  else {
    showScore.innerHTML = `
<h3> Your Score is ${score}/${quizDB.length}</h3>
<button class="btn" onclick="location.reload()">Play Again</button>
`;
    showScore.classList.remove('scoreArea');
  }
});

```

Module5: CSS

1. rulesCSS.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>CSS Rules Page</title>
<link rel="stylesheet" href="rulesCSS.css">
</head>
<body>
<div class="container">
<div id="home" class="flex-column flex-center">
<div class="info-title"><h1>Welcome to CSS Quiz</h1>
<h2>Here are some rules:</h2>
</div>
    <div class="info-list">
        <div class="info">1. The quiz contains multiple choice questions
with four options each.</div>
        <div class="info">2. You cannot change the selected option once
clicked on 'submit' button.</div>
        <div class="info">3. Click on 'submit' to submit answer of each
question</div>
        <div class="info">4. You'll get points on the basis of your correct
answers.</div>
    </div>
<a href="CSS.html" </a>
<button id="btn" class="btn">Continue</button>
</div>
</div>
</body>
</html>
```

2. rulesCSS.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{ width: 50vw;
background-color: #7fffd4;
border-radius: 10px;
align-items:center;
}
.container {
width:100vw;
min-height:100vh;
display:grid;
place-items: center;
background-color: #f0ffff;
}
.flex-column , h1{
display:flex;
flex-direction: column;
align-items:center;
}
.flex-center, h1{
justify-content:center;
align-items: center;
}
.btn{
font-size:20px;
padding: 20px;
text-align:center;
margin-top:20px;
cursor: pointer;
background-color:#00ff00;
}
.btn:hover{
cursor: hover;
background-color:#ffa07a;
}
.hidden{
display:none;
}
html{
font-size: 120%;
}
.start_btn,
.info_box,
.quiz_box,
.result_box{
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
}
```

```
.info_box{
  width: 540px;
  background: #fff;
  border-radius: 5px;
  transform: translate(-50%, -50%) scale(0.9);
  opacity: 0;
  pointer-events: none;
  transition: all 0.3s ease;
}
.info_box .info-title{
  height: 60px;
  width: 100%;

  display: flex;
  align-items: center;
  padding: 20 30px;
  border-radius: 5px 5px 0 0;
  font-size: 20px;
  font-weight: 600;
}
.info_box .info-list{
  padding: 15px 30px;
}
.info_box .info-list .info{
  margin: 5px 5px;
  font-size: 17px;
}
.info_box .info-list .info span{
  font-weight: 600;
  color: #007bff;
}
.info_box .buttons{
  height: 60px;
  display: flex;
  align-items: center;
  justify-content: flex-end;
  padding: 0 30px;
  border-top: 1px solid lightgrey;
}
.info_box .buttons button{
  margin: 0 5px;
  height: 40px;
  width: 100px;
  font-size: 16px;
  font-weight: 500;
  cursor: pointer;
  border: none;
  outline: none;
  border-radius: 5px;
  border: 1px solid #007bff;
  transition: all 0.3s ease;
}
```

3. CSS.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Quiz Game!</title>
<link rel="stylesheet" href="CSS.css">
</head>
<body>
<div class="main-div">
  <div class="inner-div">
    <h2 class="question">Questions come here</h2>
    <ul>
      <li>
        <input type="radio" name="answer" id="ans1" class="answer">
        <!--the name should be same for all the lists since when we click on
the radio button then the other button should get deselected.-->
        <label for="ans1" id="option1">Answer option</label>
      </li>

      <li>
        <input type="radio" name="answer" id="ans2" class="answer">
        <label for="ans2" id="option2">Answer option</label>
      </li>

      <li>
        <input type="radio" name="answer" id="ans3" class="answer">
        <label for="ans3" id="option3">Answer option</label>
      </li>

      <li>
        <input type="radio" name="answer" id="ans4" class="answer">
        <label for="ans4" id="option4">Answer option</label>
      </li>
    </ul>
    <button id="submit" >submit</button>
    <div id="showScore" class="scoreArea"></div>
    <button id="exit" onclick="openWin()">Exit</button>
  </div>
</div>
<script src="CSS.js"></script>
</body>
</html>
```


4. CSS.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
    margin: 0;
    padding:0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
    font-weight: 300;
}
html{
    font-size: 120%;
}
.main-div{
    width: 100vw;
    min-height:100vh;
    display:grid;
    place-items: center;
    background-color: #f0ffff;
}
.inner-div{
    width: 50vw;
    background-color: #7fffd4;
    padding: 22px 22px 30px 30px;
    border-radius: 10px;
    box-shadow: 5px 10px 10px 5px #dcdcdc;
}
.inner-div h2{
    font-size:35px;
    font-weight: 400;
    margin: 1px 0 4px 0;
}
.inner-div li{
    font-size:15px;
    margin-top:10px;
    list-style: none;
}
input{
    cursor: pointer;
}
#submit, .btn{
    padding: 10px 15px;
    outline:none;
    font-size:15px;
    margin:auto;
    display: block;
    text-transform:uppercase;
    margin-top:20px;
    cursor: pointer;
}
```

```
#submit:hover{
    background-color:#ffa07a;
}
#exit, .btn{
    padding: 10px 15px;
    outline:none;
    font-size:15px;
    margin:auto;
    display: block;
    text-transform:uppercase;
    margin-top:20px;
    cursor: pointer;
}
#exit:hover{
    background-color:#ffa07a;
}
#showScore{

    background-color:#00ff00;
    margin-top:20px;
    padding:10px 15px;

}

#showScore h3{
    font: 30px;
    text-align: center;
}

#showScore .btn{
    margin-top: 20px;
    background-color: #adff2f;

}

#showScore .btn:hover{
    background-color:#ff69b4;
}
.scoreArea{
    display: none;
}
```

5. CSS.js File

```
// creating an array and passing the number, questions, options, and answers
const quizDB = [
  {
    question: "Q1) What does CSS stand for?",
    a: "Cascade style sheets",
    b: "Color and style sheets",
    c: "Cascading style sheets",
    d: "None of the above",
    ans:"ans3"
  },

  {
    question: "Q2) The property in CSS used to change the background color of an element is -",
    a: "bgcolor",
    b: "color",
    c: "background-color",
    d: "All of the above",
    ans:"ans3"
  },

  {
    question: "Q3) The CSS property used to control the element's font-size is -",
    a: "text-style",
    b: "font-size",
    c: "text-size",
    d: "None of the above",
    ans:"ans2"
  },

  {
    question: "Q4) Which of the HTML attribute used to define the internal stylesheet is?",
    a: "style tag",
    b: "link tag",
    c: "b tag",
    d: "script tag",
    ans:"ans1"
  },

  {
    question: "Q5) Which CSS property used to specify the transparency of an element is?",
    a: 'opacity',
    b: "filter",
    c: "visibility",
    d: "overlay",
    ans:"ans1"
  }
];
```

```

const question = document.querySelector('.question'); //for class we use .
const option1 = document.querySelector('#option1'); //for ids we use #
const option2 = document.querySelector('#option2');
const option3 = document.querySelector('#option3');
const option4 = document.querySelector('#option4');
const submit = document.querySelector('#submit');
const answers = document.querySelectorAll('.answer'); //taking reference from
input
const showScore = document.querySelector('#showScore');
let questionCount = 0;
let score=0;
//defining function loadQuestion()
const loadQuestion()=> {
const questionList = quizDB[questionCount];
question.innerHTML = questionList.question;
option1.innerHTML = questionList.a;
option2.innerHTML = questionList.b;
option3.innerHTML = questionList.c;
option4.innerHTML = questionList.d;
}
//calling function loadQuestion()
loadQuestion();
//defining function getCheckAnswer()
const getCheckAnswer()=> {
let answer;
answers.forEach((curAnsElem)=> {
    if(curAnsElem.checked){
        answer = curAnsElem.id;
    }
});
return answer;
};
const deselectAll = () =>{
    answers.forEach((curAnsElem)=> curAnsElem.checked = false);
};
var myWindow;
function openWin() {
    myWindow = window.open("index.html");
}
submit.addEventListener('click', ()=>{
    const checkedAnswer = getCheckAnswer();
    console.log(checkedAnswer);

    if(checkedAnswer== quizDB[questionCount].ans)
    {
        score++;
    };
    questionCount++;
    deselectAll();
    if(questionCount<quizDB.length){
        loadQuestion();
    }
}

```

```

else {
showScore.innerHTML = `
<h3> Your Score is ${score}/${quizDB.length}</h3>
<button class="btn" onclick="location.reload()">Play Again</button>
`;
showScore.classList.remove('scoreArea');
}
});

```

Module 6: JAVA

1. rulesJAVA.html File

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Rules Page</title>
<link rel="stylesheet" href="rulesJAVA.css">
</head>
<body>
<div class="container">
<div id="home" class="flex-column flex-center">

<div class="info-title"><h1>Welcome to JAVA Quiz</h1>
<h2>Here are some rules:</h2>
</div>
  <div class="info-list">
    <div class="info">1. The quiz contains multiple choice questions with
four options each.</div>
    <div class="info">2. You cannot change the selected option once clicked
on 'submit' button.</div>
    <div class="info">3. Click on 'submit' to submit answer of each
question</div>
    <div class="info">4. You'll get points on the basis of your correct
answers.</div>
  </div>
  <a href="JAVA.html" </a>
<button id="btn" class="btn">Continue</button>
</div>
</div>
</body>
</html>

```

2. rulesJAVA.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
    width: 50vw;
    background-color: #7fffd4;
    border-radius: 10px;
    align-items:center;
}
.container {
width:100vw;
    min-height:100vh;
    display:grid;
    place-items: center;
    background-color: #f0ffff;
}
.flex-column , h1{
    display:flex;
    flex-direction: column;
    align-items:center;
}
.flex-center, h1{
justify-content:center;
align-items: center;
}
.btn{
    font-size:20px;
    padding: 20px;
    text-align:center;
    margin-top:20px;
    cursor: pointer;
    background-color:#00ff00;
}

.btn:hover{
    cursor: hover;
    background-color:#ffa07a;
}
.hidden{
    display:none;
}
html{
    font-size: 120%;
}
.start_btn,
.info_box,
.quiz_box,
.result_box{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
```

```
        box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),
                    0 6px 20px 0 rgba(0, 0, 0, 0.19);
    }
    .info_box{
        width: 540px;
        background: #fff;
        border-radius: 5px;
        transform: translate(-50%, -50%) scale(0.9);
        opacity: 0;
        pointer-events: none;
        transition: all 0.3s ease;
    }
    .info_box .info-title{
        height: 60px;
        width: 100%;

        display: flex;
        align-items: center;
        padding: 20 30px;
        border-radius: 5px 5px 0 0;
        font-size: 20px;
        font-weight: 600;
    }
    .info_box .info-list{
        padding: 15px 30px;
    }
    .info_box .info-list .info{
        margin: 5px 5px;
        font-size: 17px;
    }
    .info_box .info-list .info span{
        font-weight: 600;
        color: #007bff;
    }
    .info_box .buttons{
        height: 60px;
        display: flex;
        align-items: center;
        justify-content: flex-end;
        padding: 0 30px;
        border-top: 1px solid lightgrey;
    }
    .info_box .buttons button{
        margin: 0 5px;
        height: 40px;
        width: 100px;
        font-size: 16px;
        font-weight: 500;
        cursor: pointer;
        border: none;
        outline: none;
        border-radius: 5px;
        border: 1px solid #007bff;
        transition: all 0.3s ease;
    }
}
```

3. JAVA.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Quiz Game!</title>
<link rel="stylesheet" href="JAVA.css">
</head>
<body>
<div class="main-div">
  <div class="inner-div">
    <h2 class="question">Questions come here</h2>
    <ul>
      <li>
        <input type="radio" name="answer" id="ans1" class="answer">
        <!--the name should be same for all the lists since when we click on
the radio button then the other button should get deselected.-->
        <label for="ans1" id="option1">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans2" class="answer">
        <label for="ans2" id="option2">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans3" class="answer">
        <label for="ans3" id="option3">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans4" class="answer">
        <label for="ans4" id="option4">Answer option</label>
      </li>
    </ul>
    <button id="submit" >submit</button>
    <div id="showScore" class="scoreArea"></div>
    <button id="exit" onclick="openWin()">Exit</button>
  </div>
</div>
<script src="JAVA.js"></script>
</body>
</html>
```


4. JAVA.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
    margin: 0;
    padding:0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
    font-weight: 300;
}
html{
    font-size: 120%;
}
.main-div{
    width: 100vw;
    min-height:100vh;
    display:grid;
    place-items: center;
    background-color: #f0ffff;
}
.inner-div{
    width: 50vw;
    background-color: #7fffd4;
    padding: 22px 22px 30px 30px;
    border-radius: 10px;
    box-shadow: 5px 10px 10px 5px #dcdcdc;
}
.inner-div h2{
    font-size:35px;
    font-weight: 400;
    margin: 1px 0 4px 0;
}
.inner-div li{

    font-size:15px;
    margin-top:10px;
    list-style: none;
}
input{
    cursor: pointer;
}
#submit, .btn{
    padding: 10px 15px;
    outline:none;
    font-size:15px;
    margin:auto;
    display: block;
    text-transform:uppercase;
    margin-top:20px;
    cursor: pointer;
}
#submit:hover{
    background-color:#ffa07a;
```

```
}
#exit, .btn{
    padding: 10px 15px;
    outline:none;
    font-size:15px;
    margin:auto;
    display: block;
    text-transform:uppercase;
    margin-top:20px;
    cursor: pointer;
}
#exit:hover{
    background-color:#ffa07a;
}
#showScore{
    background-color:#00ff00;
    margin-top:20px;
    padding:10px 15px;
}
#showScore h3{
    font: 30px;
    text-align: center;
}
#showScore .btn{
    margin-top: 20px;
    background-color: #adff2f;
}
#showScore .btn:hover{
    background-color:#ff69b4;
}
.scoreArea{
    display: none;
}
```

5. JAVA.js File

```
// creating an array and passing the number, questions, options, and answers
const quizDB = [
  {
    question: "Q1) Which of the following option leads to the portability and security of Java?",
    a: "Bytecode is executed by JVM",
    b: "The applet makes the Java code secure and portable",
    c: "Use of exception handling",
    d: "Dynamic binding between objects",
    ans:"ans1"
  },
  {
    question: "Q2) Which of the following is not a Java features?",
    a: "Dynamic",
    b: "Architecture Neutral",
    c: "Use of pointers",
    d: "Object-oriented",
    ans:"ans3"
  },
  {
    question: "Q3) _____ is used to find and fix bugs in the Java programs.",
    a: "JVM",
    b: "JRE",
    c: "JDK",
    d: "JDB",
    ans:"ans4"
  },
  {
    question: "Q4) What is the return type of the hashCode() method in the Object class?",
    a: "Object",
    b: "int",
    c: "long",
    d: "void",
    ans:"ans2"
  },
  {
    question: "Q5) Evaluate the following Java expression, if x=3, y=5, and z=10: ++z + y - y + z + x++",
    a: '24',
    b: "23",
    c: "20",
    d: "25",
    ans:"ans4"
  }
];
```

```

const question = document.querySelector('.question'); //for class we use .
const option1 = document.querySelector('#option1'); //for ids we use #
const option2 = document.querySelector('#option2');
const option3 = document.querySelector('#option3');
const option4 = document.querySelector('#option4');
const submit = document.querySelector('#submit');
const answers = document.querySelectorAll('.answer'); //taking reference from
input
const showScore = document.querySelector('#showScore');
let questionCount = 0;
let score=0;
//defining function loadQuestion()
const loadQuestion=()=> {
    const questionList = quizDB[questionCount];
    question.innerHTML = questionList.question;
    option1.innerHTML = questionList.a;
    option2.innerHTML = questionList.b;
    option3.innerHTML = questionList.c;
    option4.innerHTML = questionList.d;
}
//calling function loadQuestion()
loadQuestion();
//defining function getCheckAnswer()
const getCheckAnswer=()=> {
    let answer;
    answers.forEach((curAnsElem)=> {
        if(curAnsElem.checked){
            answer = curAnsElem.id;
        }
    });
    return answer;
};
const deselectAll = () =>{
    answers.forEach((curAnsElem)=> curAnsElem.checked = false);
};
var myWindow;
function openWin() {
    myWindow = window.open("index.html");
}
submit.addEventListener('click', ()=>{
    const checkedAnswer = getCheckAnswer();
    console.log(checkedAnswer);

    if(checkedAnswer== quizDB[questionCount].ans)
    {
        score++;
    };
    questionCount++;
    deselectAll();
    if(questionCount<quizDB.length){
        loadQuestion();
    }
    else {
        showScore.innerHTML =

```

```

<h3> Your Score is ${score}/${quizDB.length}</h3>
<button class="btn" onclick="location.reload()">Play Again</button>
`;
showScore.classList.remove('scoreArea');
}
});

```

Module 6: SQL

1. rulesSQL.html File

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Rules Page</title>
<link rel="stylesheet" href="rulesSQL.css">
</head>
<body>
<div class="container">
<div id="home" class="flex-column flex-center">
<div class="info-title"><h1>Welcome to SQL Quiz</h1>
<h2>Here are some rules:</h2>
</div>
<div class="info-list">
<div class="info">1. The quiz contains multiple choice questions
with four options each.</div>
<div class="info">2. You cannot change the selected option once
clicked on 'submit' button.</div>
<div class="info">3. Click on 'submit' to submit answer of each
question</div>
<div class="info">4. You'll get points on the basis of your correct
answers.</div>
</div>
<a href="SQL.html" </a>
<button id="btn" class="btn">Continue</button>
</div>
</div>
</body>
</html>

```

2. rulesSQL.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
    width: 50vw;
    background-color: #7fffd4;
    border-radius: 10px;
    align-items:center;
}
.container {
    width:100vw;
    min-height:100vh;
    display:grid;
    place-items: center;
    background-color: #f0ffff;
}
.flex-column , h1{
    display:flex;
    flex-direction: column;
    align-items:center;
}
.flex-center, h1{
    justify-content:center;
    align-items: center;
}
.btn{
    font-size:20px;
    padding: 20px;
    text-align:center;
    margin-top:20px;
    cursor: pointer;
    background-color:#00ff00;
}
.btn:hover{
    cursor: hover;
    background-color:#ffa07a;
}
.hidden{
    display:none;
}
html{
    font-size: 120%;
}
.start_btn,
.info_box,
.quiz_box,
.result_box{
    position: absolute;
    top: 50%;
    left: 50%;
```

```
    transform: translate(-50%, -50%);
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),
                0 6px 20px 0 rgba(0, 0, 0, 0.19);
}
.info_box{
    width: 540px;
    background: #fff;
    border-radius: 5px;
    transform: translate(-50%, -50%) scale(0.9);
    opacity: 0;
    pointer-events: none;
    transition: all 0.3s ease;
}
.info_box .info-title{
    height: 60px;
    width: 100%;

    display: flex;
    align-items: center;
    padding: 20 30px;
    border-radius: 5px 5px 0 0;
    font-size: 20px;
    font-weight: 600;
}
.info_box .info-list{
    padding: 15px 30px;
}
.info_box .info-list .info{
    margin: 5px 5px;
    font-size: 17px;
}
.info_box .info-list .info span{
    font-weight: 600;
    color: #007bff;
}
.info_box .buttons{
    height: 60px;
    display: flex;
    align-items: center;
    justify-content: flex-end;
    padding: 0 30px;
    border-top: 1px solid lightgrey;
}
.info_box .buttons button{
    margin: 0 5px;
    height: 40px;
    width: 100px;
    font-size: 16px;
    font-weight: 500;
    cursor: pointer;
    border: none;
    outline: none;
    border-radius: 5px;
    border: 1px solid #007bff;
    transition: all 0.3s ease;
}
```

3. SQL.html File

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Quiz Game!</title>
<link rel="stylesheet" href="SQL.css">
</head>
<body>
<div class="main-div">
  <div class="inner-div">
    <h2 class="question">Questions come here</h2>
    <ul>
      <li>
        <input type="radio" name="answer" id="ans1" class="answer">
        <!--the name should be same for all the lists since when we click on
the radio button then the other button should get deselected.-->
        <label for="ans1" id="option1">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans2" class="answer">
        <label for="ans2" id="option2">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans3" class="answer">
        <label for="ans3" id="option3">Answer option</label>
      </li>
      <li>
        <input type="radio" name="answer" id="ans4" class="answer">
        <label for="ans4" id="option4">Answer option</label>
      </li>
    </ul>
    <button id="submit" >submit</button>
    <div id="showScore" class="scoreArea"></div>
    <button id="exit" onclick="openWin()">Exit</button>
  </div>
</div>
<script src="SQL.js"></script>
</body>
</html>
```


4. SQL.css File

```
/* importing google fonts */
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
    margin: 0;
    padding:0;
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
    font-weight: 300;
}
html{
    font-size: 120%;
}
.main-div{
    width: 100vw;
    min-height:100vh;
    display:grid;
    place-items: center;
    background-color: #f0ffff;
}
.inner-div{
    width: 50vw;
    background-color: #7fffd4;
    padding: 22px 22px 30px 30px;
    border-radius: 10px;
    box-shadow: 5px 10px 10px 5px #dcdcdc;
}
.inner-div h2{
    font-size:35px;
    font-weight: 400;
    margin: 1px 0 4px 0;
}
.inner-div li{
    font-size:15px;
    margin-top:10px;
    list-style: none;
}
input{
    cursor: pointer;
}
#submit, .btn{
    padding: 10px 15px;
    outline:none;
    font-size:15px;
    margin:auto;
    display: block;
    text-transform:uppercase;
    margin-top:20px;
    cursor: pointer;
}
```

```
#submit:hover{
    background-color:#ffa07a;
}
#exit, .btn{
    padding: 10px 15px;
    outline:none;
    font-size:15px;
    margin:auto;
    display: block;
    text-transform:uppercase;
    margin-top:20px;
    cursor: pointer;
}
#exit:hover{
    background-color:#ffa07a;
}
#showScore{
    background-color:#00ff00;
    margin-top:20px;
    padding:10px 15px;
}
#showScore h3{
    font: 30px;
    text-align: center;
}
#showScore .btn{
    margin-top: 20px;
    background-color: #adff2f;
}
#showScore .btn:hover{
    background-color:#ff69b4;
}
.scoreArea{
    display: none;
}
```

5. SQL.js File

```
// creating an array and passing the number, questions, options, and answers
const quizDB = [
  {
    question: "Q1) What is the full form of SQL?",
    a: "Structured Query List",
    b: "Structure Query Language",
    c: "Sample Query Language",
    d: "None of the above",
    ans: "ans2"
  },
  {
    question: "Q2) Which of the following is not a valid SQL type?",
    a: "FLOAT",
    b: "NUMERIC",
    c: "DECIMAL",
    d: "CHARACTER",
    ans: "ans3"
  },
  {
    question: "Q3) Which of the following is not a DDL command?",
    a: "TRUNCATE",
    b: "ALTER",
    c: "CREATE",
    d: "UPDATE",
    ans: "ans4"
  },
  {
    question: "Q4) Which statement is used to delete all rows in a table without having the action logged?",
    a: "DELETE",
    b: "REMOVE",
    c: "DROP",
    d: "TRUNCATE",
    ans: "ans4"
  },
  {
    question: "Q5) How many Primary keys can have in a table?",
    a: 'Only 1',
    b: "Only 2",
    c: "Depends on no of Columns",
    d: "Depends on DBA",
    ans: "ans1"
  }
];

const question = document.querySelector('.question'); //for class we use .
const option1 = document.querySelector('#option1'); //for ids we use #
const option2 = document.querySelector('#option2');
const option3 = document.querySelector('#option3');
const option4 = document.querySelector('#option4');
const submit = document.querySelector('#submit');
```

```

const answers = document.querySelectorAll('.answer'); //taking reference from
input
const showScore = document.querySelector('#showScore');
let questionCount = 0;
let score=0;
//defining function loadQuestion()
const loadQuestion=()=> {
    const questionList = quizDB[questionCount];
    question.innerHTML = questionList.question;
    option1.innerHTML = questionList.a;
    option2.innerHTML = questionList.b;
    option3.innerHTML = questionList.c;
    option4.innerHTML = questionList.d;
}
//calling function loadQuestion()
loadQuestion();
//defining function getCheckAnswer()
const getCheckAnswer=()=> {
    let answer;
    answers.forEach((curAnsElem)=> {
        if(curAnsElem.checked){
            answer = curAnsElem.id;
        }
    });
    return answer;
};
const deselectAll = () =>{
    answers.forEach((curAnsElem)=> curAnsElem.checked = false);
};
var myWindow;
function openWin() {
    myWindow = window.open("index.html");
}
submit.addEventListener('click', ()=>{
    const checkedAnswer = getCheckAnswer();
    console.log(checkedAnswer);
    if(checkedAnswer== quizDB[questionCount].ans)
    {
        score++;
    };
    questionCount++;
    deselectAll();

    if(questionCount<quizDB.length){
        loadQuestion();
    }
    else {
        showScore.innerHTML = `
<h3> Your Score is ${score}/${quizDB.length}</h3>
<button class="btn" onclick="location.reload()">Play Again</button>
`;
        showScore.classList.remove('scoreArea');
    }
});

```

11.0 Output

11.1 First Page Module:

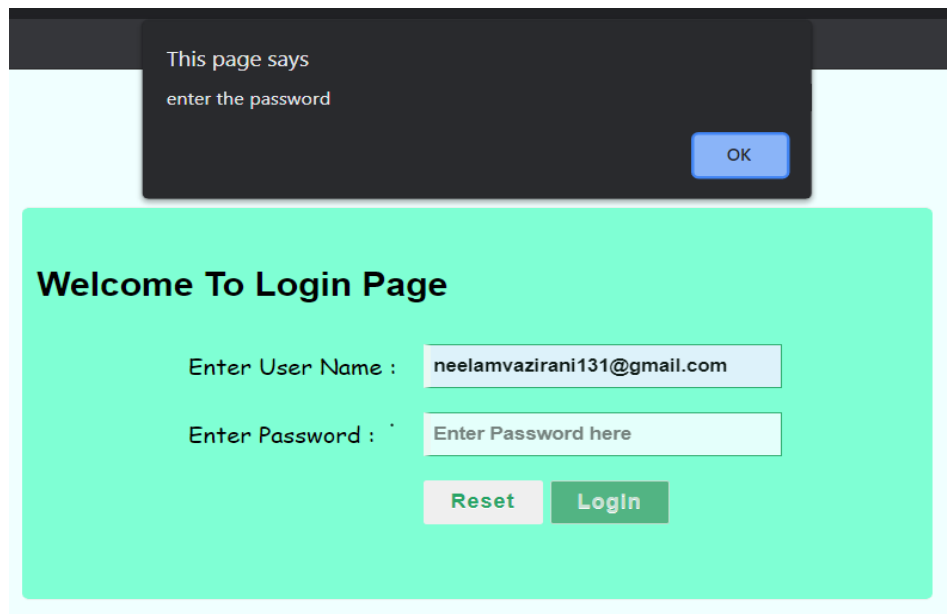


11.2 Login Module:

(i) When the user presses 'Login' button without entering user name and password

A screenshot of a web application's login page. At the top, a dark grey banner contains the text 'This page says' and 'Please enter user name' in white. Below this, a blue 'OK' button is visible. The main content area has a light green background. It features the heading 'Welcome To Login Page' in bold black text. Below the heading are two input fields: 'Enter User Name : Enter user name here' and 'Enter Password : Enter Password here'. At the bottom of the form are two buttons: a grey 'Reset' button and a green 'Login' button.

(ii) When the user presses 'Login' button without entering password



This page says
enter the password

OK

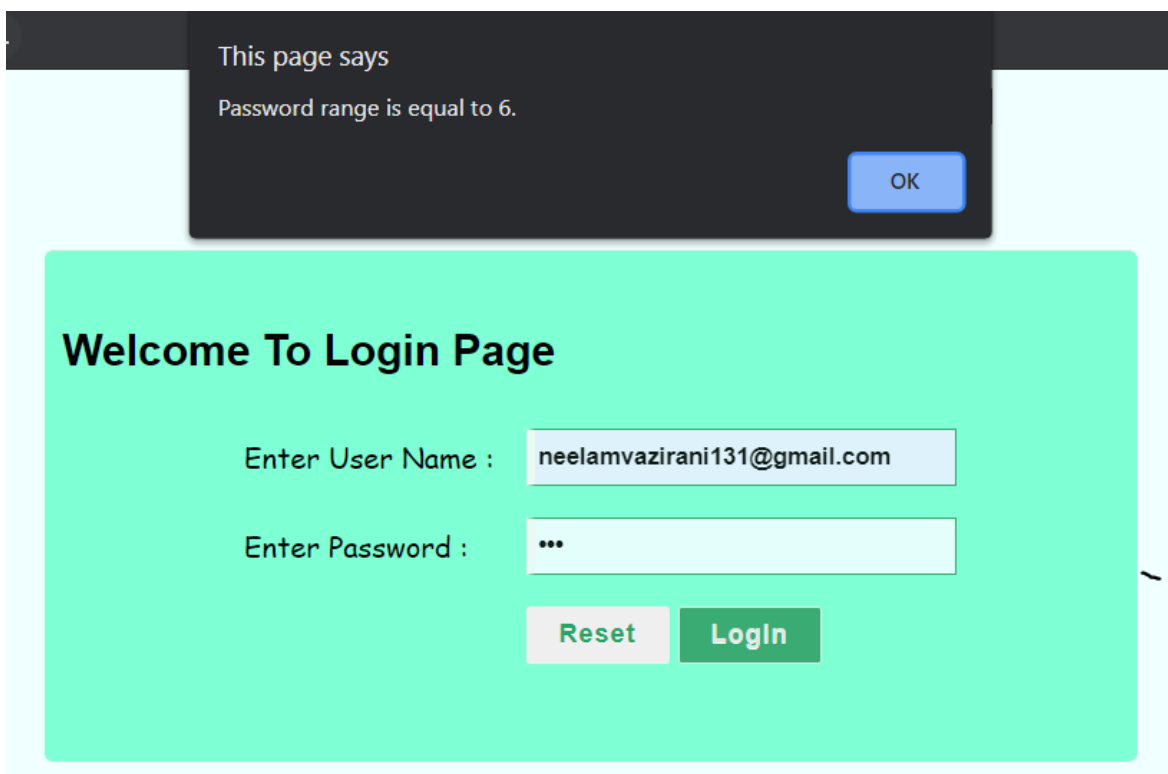
Welcome To Login Page

Enter User Name :

Enter Password :

The screenshot shows a login page with a light blue background. At the top, there is a dark grey header bar. Below the header, a dark grey message box with rounded corners contains the text "This page says" and "enter the password" in white, with a blue "OK" button. The main content area is a light blue rectangle. It contains the heading "Welcome To Login Page" in bold black text. Below the heading, there are two input fields. The first is labeled "Enter User Name :" and contains the email "neelamvazirani131@gmail.com". The second is labeled "Enter Password :" and is empty. Below the input fields are two buttons: a light blue "Reset" button and a dark blue "Login" button.

(iii) When the user presses 'Login' button with range of password ≤ 6



This page says
Password range is equal to 6.

OK

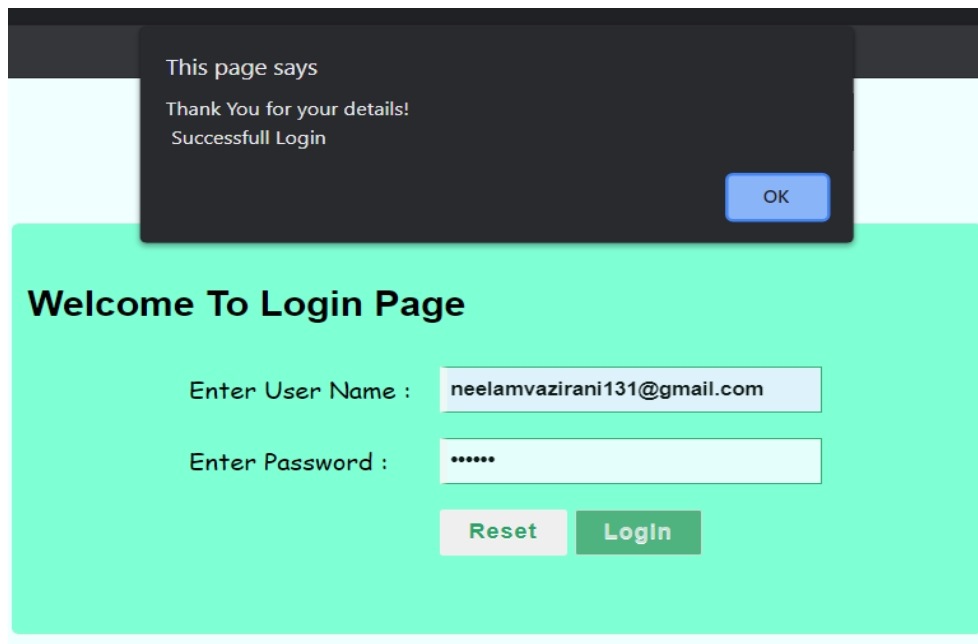
Welcome To Login Page

Enter User Name :

Enter Password :

The screenshot shows the same login page as in the previous image. The dark grey message box now contains the text "This page says" and "Password range is equal to 6." in white, with a blue "OK" button. The main content area is the same light blue rectangle with the heading "Welcome To Login Page". The "Enter User Name" field still contains the email "neelamvazirani131@gmail.com". The "Enter Password" field now contains three dots "...". The "Reset" and "Login" buttons are still present at the bottom.

(iv) When the user presses 'Login' button after filling the correct data



This page says
Thank You for your details!
Successfull Login

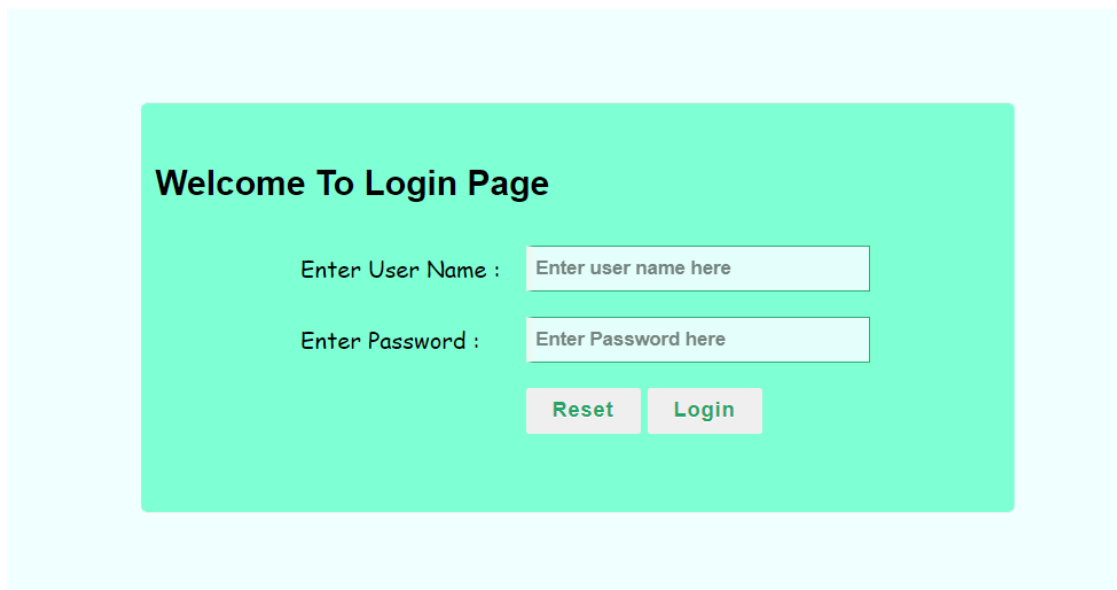
OK

Welcome To Login Page

Enter User Name :

Enter Password :

(v) When the user presses 'Reset' button



Welcome To Login Page

Enter User Name :

Enter Password :

11.3 Selection of topic Module:



The screenshot shows a web interface for selecting a topic. It features a light blue background with a central white rounded rectangle. Inside this rectangle, the title "Pick a topic" is centered at the top. Below the title are five red buttons with black text, stacked vertically. The buttons are labeled: "1. HTML QUIZ SECTION", "2. CSS QUIZ SECTION", "3. JAVA QUIZ SECTION", "4. SQL QUIZ SECTION", and "Go back".

Pick a topic

1. HTML QUIZ SECTION

2. CSS QUIZ SECTION

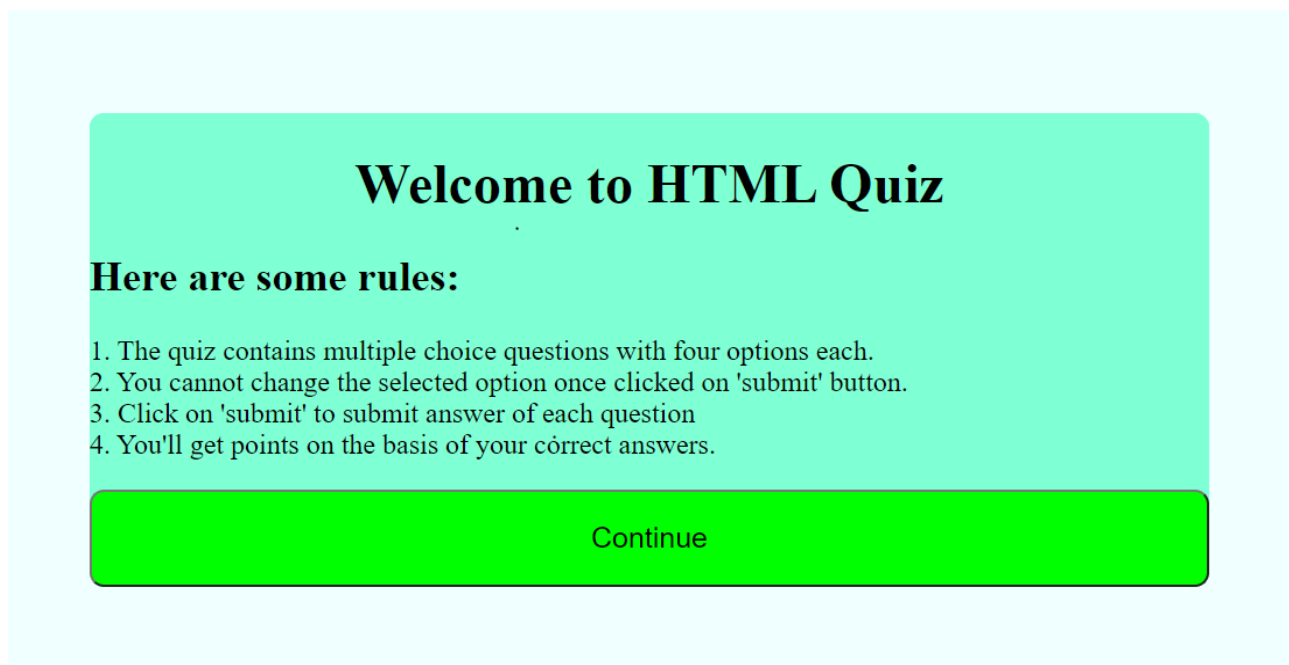
3. JAVA QUIZ SECTION

4. SQL QUIZ SECTION

Go back

11.4 HTML Quiz Section Module:

(i) HTML Rules



The screenshot shows a web interface for the HTML Quiz. It features a light blue background with a central white rounded rectangle. Inside this rectangle, the title "Welcome to HTML Quiz" is centered at the top. Below the title, the text "Here are some rules:" is followed by a list of four rules. At the bottom of the rectangle is a red button with black text labeled "Continue".

Welcome to HTML Quiz

Here are some rules:

1. The quiz contains multiple choice questions with four options each.
2. You cannot change the selected option once clicked on 'submit' button.
3. Click on 'submit' to submit answer of each question
4. You'll get points on the basis of your correct answers.

Continue

(ii) **Displaying HTML questions one by one when the user selects 'submit' button**

Q1) What does HTML stand for?

- ☐ Hyper Text Preprocessor
- ☒ Hyper Text Markup Language
- ☐ Hyper Text Multiple Language
- ☐ Hyper Tool Multi Language

SUBMIT

EXIT

Q2) Which of the following element is responsible for making the text italic in HTML?

- ☒ i tag
- ☐ italic tag
- ☐ it tag
- ☐ pre tag

SUBMIT

EXIT

Q3) The correct sequence of HTML tags for starting a webpage is -

- ☐ Head, Title, HTML, body
- ☒ HTML, Body, Title, Head
- ☐ HTML, Head, Title, Body
- ☐ HTML, Head, Title, Body

SUBMIT

EXIT

Q4) Which of the following element is responsible for making the text bold in HTML?

- ☐ pre tag
- ☐ a tag
- ☒ b tag
- ☐ br tag

SUBMIT

EXIT

Q5) Which of the following tag is used for inserting the largest heading in HTML?

- ☐ h3 tag
- ☒ h1 tag
- ☐ h5 tag
- ☐ h6 tag

SUBMIT

EXIT

(iii) Displaying Score

Your Score is 5/5

PLAY AGAIN

EXIT

11.5 CSS Quiz Section Module:

(i) CSS Rules

Welcome to CSS Quiz

Here are some rules:

1. The quiz contains multiple choice questions with four options each.
2. You cannot change the selected option once clicked on 'submit' button.
3. Click on 'submit' to submit answer of each question
4. You'll get points on the basis of your correct answers.

Continue

(ii) Displaying CSS questions one by one when the user selects 'submit' button

Q1) What does CSS stand for?

- ☐ Cascade style sheets
- ☐ Color and style sheets
- ☒ Cascading style sheets
- ☐ None of the above

SUBMIT

EXIT

Q2) The property in CSS used to change the background color of an element is -

- ☐ bgcolor
- ☐ color
- ☒ background-color
- ☐ All of the above

SUBMIT

EXIT

Q3) The CSS property used to control the element's font-size is -

- ☐ text-style
- ☒ font-size
- ☐ text-size
- ☐ None of the above

SUBMIT

EXIT

Q4) Which of the HTML attribute used to define the internal stylesheet is?

- ☒ style tag
- ☐ link tag
- ☐ b tag
- ☐ script tag

SUBMIT

EXIT

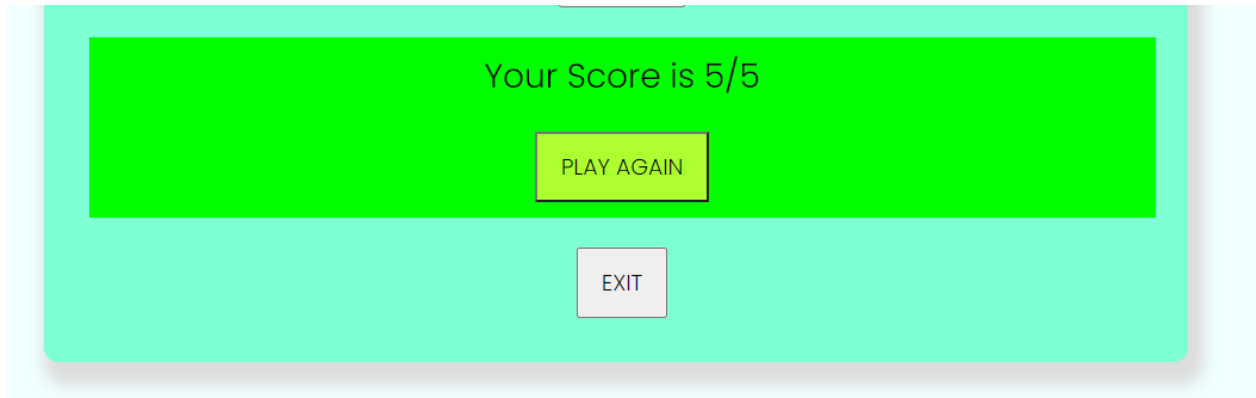
Q5) Which CSS property used to specify the transparency of an element is?

- ☒ opacity
- ☐ filter
- ☐ visibility
- ☐ overlay

SUBMIT

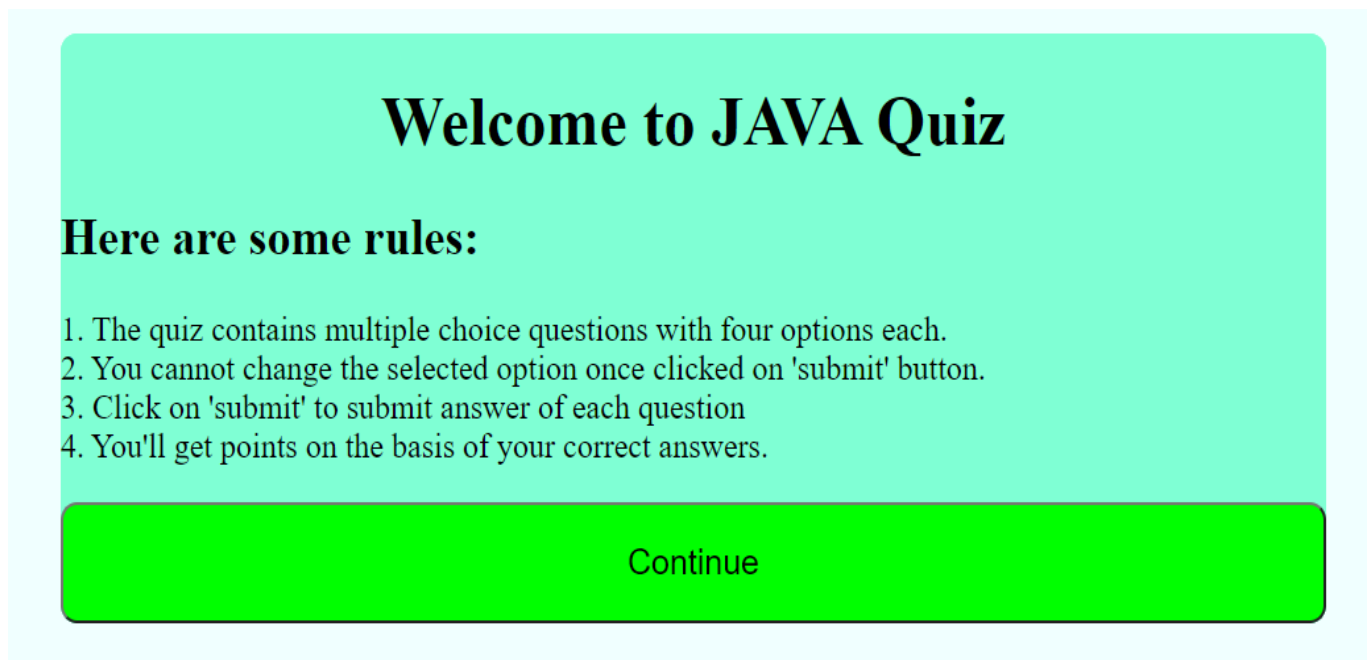
EXIT

(iii) Displaying Score



11.6 JAVA Quiz Section Module:

(i) JAVA Rules



(ii) **Displaying HTML questions one by one when the user selects 'submit' button**

Q1) Which of the following option leads to the portability and security of Java?

- ☒ Bytecode is executed by JVM
- ☐ The applet makes the Java code secure and portable
- ☐ Use of exception handling
- ☐ Dynamic binding between objects

SUBMIT

EXIT

Q2) Which of the following is not a Java features?

- ☐ Dynamic
- ☐ Architecture Neutral
- ☒ Use of pointers
- ☐ Object-oriented

SUBMIT

EXIT

Q3) _____ is used to find and fix bugs in the Java programs.

- ☐ JVM
- ☐ JRE
- ☐ JDK
- ☒ JDB

SUBMIT

EXIT

Q4) What is the return type of the hashCode() method in the Object class?

- ☐ Object
- ☒ int
- ☐ long
- ☐ void

SUBMIT

EXIT

Q5) Evaluate the following Java expression, if x=3, y=5, and z=10: ++z + y - y + z + x++

- ☐ 24
- ☐ 23
- ☐ 20
- ☒ 25

SUBMIT

EXIT

(iii) Displaying Score

Q5) Evaluate the following Java expression, if x=3, y=5, and z=10: ++z + y - y + z + x++

- ☐ 24
- ☐ 23
- ☐ 20
- ☐ 25

SUBMIT

Your Score is 5/5

PLAY AGAIN

EXIT

11.7 SQL Quiz Section Module:

(i) SQL Rules

Welcome to SQL Quiz

Here are some rules:

1. The quiz contains multiple choice questions with four options each.
2. You cannot change the selected option once clicked on 'submit' button.
3. Click on 'submit' to submit answer of each question
4. You'll get points on the basis of your correct answers.

Continue

(ii) Displaying SQL questions one by one when the user selects 'submit' button

Q1) What is the full form of SQL?

- ☐ Structured Query List
- ☒ Structure Query Language
- ☐ Sample Query Language
- ☐ None of the above

SUBMIT

EXIT

Q2) Which of the following is not a valid SQL type?

- ☐ FLOAT
- ☐ NUMERIC
- ☒ DECIMAL
- ☐ CHARACTER

SUBMIT

EXIT

Q3) Which of the following is not a DDL command?

- ☐ TRUNCATE
- ☐ ALTER
- ☐ CREATE
- ☒ UPDATE

SUBMIT

EXIT

Q4) Which statement is used to delete all rows in a table without having the action logged?

- ☐ DELETE
- ☐ REMOVE
- ☐ DROP
- ☒ TRUNCATE

SUBMIT

EXIT

Q5) How many Primary keys can have in a table?

- ☒ Only 1
- ☐ Only 2
- ☐ Depends on no of Columns
- ☐ Depends on DBA

SUBMIT

EXIT

(iii) Displaying Score

Q5) How many Primary keys can have in a table?

- ☐ Only 1
- ☐ Only 2
- ☐ Depends on no of Columns
- ☐ Depends on DBA

SUBMIT

Your Score is 5/5

PLAY AGAIN

EXIT

12.0 CONCLUSION

The Quiz Application is based on front technologies like HTML, CSS and JavaScript. With the help of html and CSS the quiz is dynamic and interactive. With the help of JavaScript, we can check whether the right option was clicked or not. Also, it is used to display score, take user input, etc.

The quiz contains welcome page, login page and different sections containing different mcqs based on different topics. The user can select any topic and can start the test. There is no time limit so the user can take their own time and answer the questions.

Various programming languages like HTML, CSS, JavaScript and SQL are available in this application. This application will help develop soft skills and improve user's knowledge on the subject. It is user friendly, so the user can understand the use of the application. Still there is a page just before the quiz starts which states rules. To conclude, this is a simple Online MCQ Quiz which is used to test the user and based on their answers the scores will be displayed.

13.0 REFERENCE

www.edubridgeindia.com