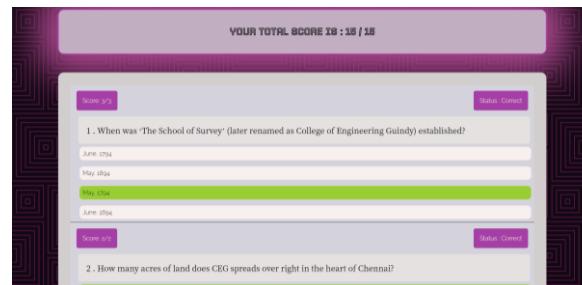
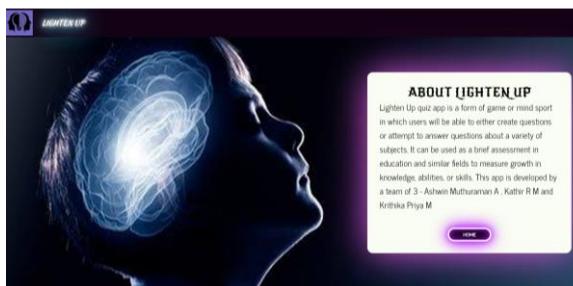
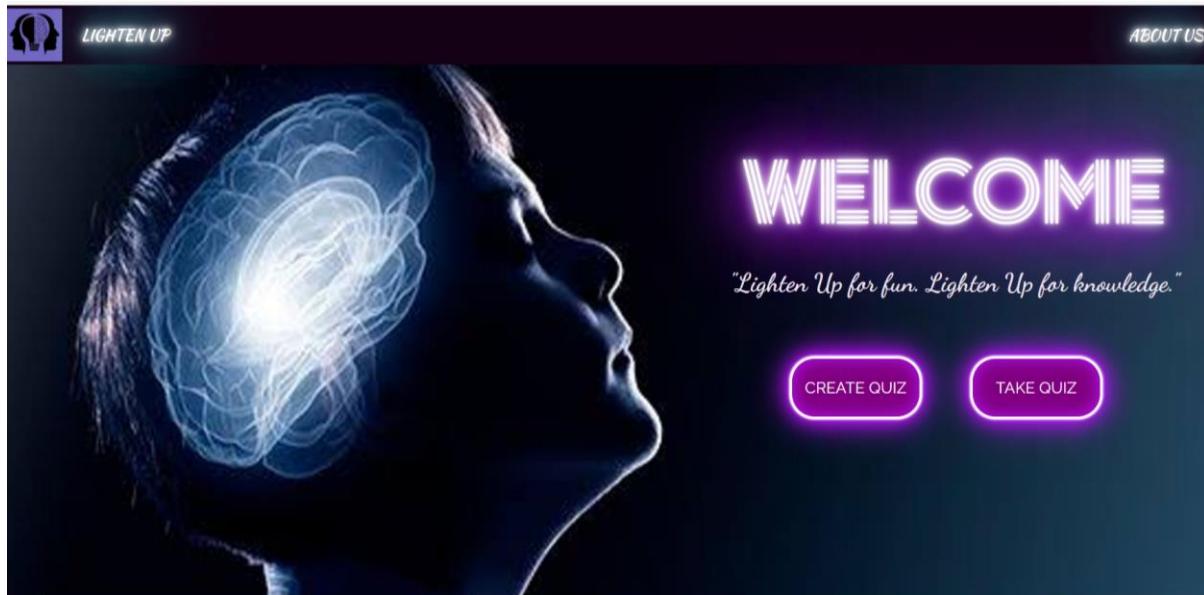


LIGHTEN UP APP



INTRODUCTION

A quiz is a form of game which can also be considered as mind sport, in which players attempt to answer questions correctly about a certain or variety of subjects. It usually refers to a short test of knowledge, with question formats with multiple choice options. Quizzes can be used as a brief assessment in education and similar fields to measure growth in knowledge, abilities, or skills.

Lighten Up is an online quiz app in which users will be able to create quizzes and also be able to attempt the available quizzes. Users can create quizzes on any subject of interest, which need not necessarily come under academics, which makes the app more fun filled and universal. The app also gives an immediate analysis after attempting a quiz. We have developed the app such that it has a user-friendly interface. The main focus of developing this app is to explore various concepts of client-side and server-side scripting, with the help of HTML, Javascript, CSS, PHP and SQL.

Lighten up app is developed by a team of 3 –

 Ashwin Muthuraman A

 Kathir R M

 Krithika Priya M

PROJECT IMPLEMENTATION

TABLES IN DATABASE:

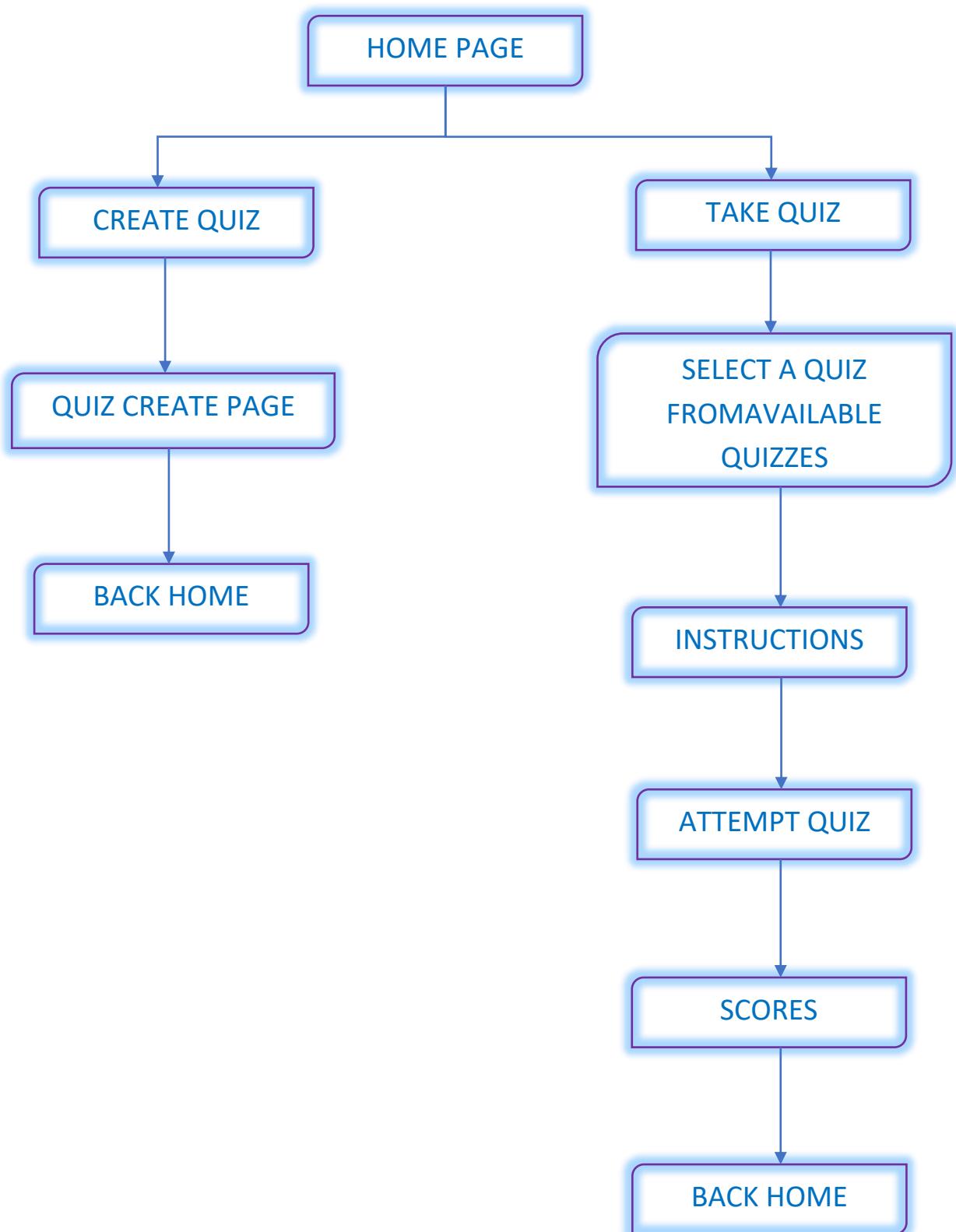
- ✚ quiz_list – contains the quiz_id, quiz_name, author_name and quiz_description
- ✚ question_table – contains the quiz_id, quiz_name, question_id, questions, and the 4 options
- ✚ answer_table – contains the quiz_id, quiz_name, question_id, correct_option
- ✚ user_input – contains the quiz_id, quiz_name, question_id, user_answer

FILES CREATED:

S.No.	Name of the File	Type	Function
1.	index	HTML – Client side	Landing Page / Home Page for the app
2.	abtus	HTML – Client side	Displays information about the app
3.	landingpg	CSS – Client Side	Contains the styling for index page
4.	landingnav	CSS – Client Side	Contains styling for navigation bar for index page
5.	nav	CSS – Client Side	Contains styling for navigational bar for other pages
6.	quiz_create	PHP – Client Side	Displays the form for creating quiz

7.	create	CSS – Client side	Contains the styling for quiz_create page
8.	update_quiz_questions	PHP – Server Side	Receives data from quiz_create page and updates the contents to database
9.	quiz_display	PHP – Server Side	Selects all the available quizzes and displays their quiz name, author name and description to the user in the form of cards
10.	cards	CSS – Client side	Contains the styling for quiz_display page
11.	instructions_page	PHP – Client Side	Displays the instructions for attempting the quiz
12.	instructions	CSS – Client side	Contains the styling for instructions_page page
13.	quiz_config	PHP – Server Side	Configures to database and selects questions of a particular quiz
14.	script	PHP – Client Side	Contains the javascript code for timer and displaying options for each question and stores user inputs in a javascript array, which is then passed to server side
15.	quiz_page	PHP – Client Side	Displays the questions for the selected quiz
16.	quiz_page	CSS – Client Side	Contains the styling for quiz_page page
17.	receive_data	PHP – Server Side	Receives user selected inputs and updates in the database
18.	myscore	PHP – Server Side	Retrieves user input and correct answer from database and also calculates the score
19.	score	CSS – Client Side	Contains the styling for myscore page

CONTROL FLOW OF THE APP



SPECIAL FEATURES

- All pages have been made responsive through CSS.
- All the pages have been made dynamic through javascript and CSS, to enhance the features of the app.
- While creating the quiz, the user can create any number of questions – achieved by dynamism through javascript.
- Also, the mark / score per question can also be decided by the user, who creates the quiz.
- Total number of available questions and total score of the quiz has been made available in the instructions page, for the user to have a clear idea about the quiz.
- The timer has been set such that once the time gets over for each question, the next question will be automatically submitted, and if the time lapses for the last question, the quiz gets auto-submitted.
- For the first 15 seconds, the timer will be displayed in green colour, after that it changes its colour from green to yellowish green for the next 10 seconds. Also, the timer changes to red colour and a pulsating effect is added to the timer, when just 5 seconds is left for a question, to make the user know about the time left.
- Once the user submits the quiz, score and analysis will be displayed immediately. Also, an animation effect is added to the score display, to make the page look attractive.

CODE SNIPPET

myscore.php

Calculating total score of the user by retrieving data from database

```
$result = mysqli_query($conn, $quiz_display);
$arr = array();
$score = 0;
$tot_score = 0;
$full_score = 0;
#calculating the total score
if ($result) {
    while ($arr = mysqli_fetch_assoc($result)) {
        $score = 0;
        if ($arr['user_answer'] == $arr['correct_option']) {
            $score = $arr['score'];
        } else if ($arr['user_answer'] == 0) {
            #do nothing
        }
        $tot_score += $score;
        $full_score += $arr['score'];
    }
}

echo "<div class = 'scorecard'>";
echo "<h3> YOUR TOTAL SCORE IS : " . $tot_score . " / ".$full_score."</h3>". "</div>";
$result = mysqli_query ($conn, $quiz_display);
$other_answers = "<span class = 'others'>";
$correct_answer = "<span class = 'correct'>";
$wrong_answer = "<span class = 'wrong'>";
$span_close = "</span>";
echo "<div class = 'container'>";
$arr = array();
```

Code for displaying correct and wrong answers in green and red colour in analysis

```
while ($arr = mysqli_fetch_assoc($result)) {
    $status = '';
    $score = 0;
    $user_values = array($other_answers, $other_answers, $other_answers, $other_answers);
    $user_values[$arr['correct_option'] - 1] = $correct_answer;
    if ($arr['user_answer'] == $arr['correct_option']) {
        $status = 'Correct';
        $score = $arr['score'];
    } else if ($arr['user_answer'] == 0) {
        $status = 'Unanswered';
    } else {
        $status = 'Wrong';
        $user_values[$arr['user_answer']-1] = $wrong_answer;
    }
    echo "<div class = 'element'><div class = 'info'><p class = 'score'>Score: ".$score."/". $arr['score']. "</p></div'><div class = 'options'>" . $user_values[0] . "<p>" . $arr['option_a'] . "</p>" . $span_close . "</div'>"; 
}
echo "</div>";

?>
```

quiz_create.php

Dynamically creating form elements for adding one more question

```
<script type="text/javascript">
    var count = 1;
    function add_qn() {
        count++;
        var x = "<table class = 'question_details'> <tr> <th> QUESTION " + count + "</th> <td> <textarea name='textarea_" +
            count+"'" rows='10' cols='15' value='' required> </textarea> </td> </tr>";
        x += "<tr> <td class = 'left-col'> <input type='radio' name='radio" + count +
            "' value='1' required/> OPTION A </td> <td> <input type='text' name='option_a" +
            count + "' required autocomplete='off'/'> </td> </tr>";
        x += "<tr> <td class = 'left-col'> <input type='radio' name='radio" + count +
            "' value='2'> OPTION B </td> <td> <input type='text' name='option_b" +
            count + "' required autocomplete='off'/'> </td> </tr>";
        x += "<tr> <td class = 'left-col'> <input type='radio' name='radio" + count +
            "' value='3'> OPTION C </td> <td> <input type='text' name='option_c" +
            count + "' required autocomplete='off'/'> </td> </tr>";
        x += "<tr> <td class = 'left-col'> <input type='radio' name='radio" + count +
            "' value='4'> OPTION D </td> <td> Enter the score </td> <td> <input type='number' name='score_" + count +
            "' required autocomplete='off' min = '1' max = '100'> </td> </tr>";
        x += "</table>";
        var y = "<p id='qn" + count + "'> </p>";
        document.getElementById('questions').insertAdjacentHTML("beforeend", y);
        document.getElementById("qn"+count).innerHTML = x;
        document.getElementById('num_of_qns').value = count;
    }
</script>
</html>
```

quiz_display.php

Dynamically displaying the available quizzes in the form of cards

```
<script type="text/javascript">
    var number_of_quiz = <?php echo $no_of_quiz ?>;
    var quiz_name_array = <?php echo json_encode($quiz_list); ?>;
    let i = 1;
    window.onload = display_buttons();
    function display_buttons() {
        for(i = 1 ; i <= number_of_quiz; i++) {
            var y = "<div class = 'card' onclick='send_id(quiz_name_array[" + i + "-1].quiz_id)'>" + '<h1>' +
                quiz_name_array[i-1].quiz_name + '</h1><p>' + quiz_name_array[i-1].quiz_description +
                "</p><p id = 'author'> Made by " +
                quiz_name_array[i-1].author_name + "</p></div>";
            document.querySelector(".container").insertAdjacentHTML("beforeend",y);
        }
    }

    function send_id(j) {
        document.getElementById('quiz_id_set').value = quiz_name_array[j-1].quiz_id;
        document.getElementById('quiz_name_set').value = quiz_name_array[j-1].quiz_name;
        document.getElementById('author_name_set').value = quiz_name_array[j-1].author_name;
        document.getElementById('form_values').submit();
    }
</script>
```

quiz_page.css

Styling for question and options while attempting questions

```
#question{
    background-color: rgb(209, 200, 200);
    font-family:'Noto Serif TC';
    padding: 1em;
    font-size: 1.2rem;
    border-radius: 10px;
}
.options{
    display: flex;
    font-family: 'Raleway',serif;
    flex-direction: column;
    flex-wrap: nowrap;
}
input[type = 'radio']){
    display: none;
}
.option{
    padding:1em;
    background-color: #AA96DA;
    margin-block:2px;
    border-radius: 10px;
    color: rgb(20, 18, 18);
    transition: all 300ms ease-in-out;
}
.option:hover{
    cursor: pointer;
    padding:1.2em;
    background-color: #CC6CC4;
}
```

Styling for timer and the pulsating effect

```
.timer_container{
    font-family:'Noto Serif TC';
    display: flex;
    align-items: baseline;
    justify-content: space-between;
    width: 100%;
}
.circle{
    width: 50px;
    height: 50px;
    line-height: 50px;
    border-radius: 50%;
    font-size:1.5rem;
    color: black;
    text-align: center;
    background: #5DD206;
    transform: scale(1);
}

@keyframes clockpulse {
0% {
    transform: scale(0.95);
    box-shadow: 0 0 0 0 rgba(0, 0, 0, 0.7);
}

70% {
    transform: scale(1);
    box-shadow: 0 0 0 10px rgba(0, 0, 0, 0);
}

100% {
    transform: scale(0.95);
    box-shadow: 0 0 0 0 rgba(0, 0, 0, 0);
}
}
```

receive_data.php

Updating user answers to database

```
<body>
<?php
session_start();
$answers_string = $_POST['answers_string'];
$answers_array = str_split($answers_string);
$number_of_questions = $_SESSION['number_of_questions'];
$quiz_name = $_SESSION['quiz_name'];
$quiz_id = $_SESSION['quiz_id'];
for ($i = 0; $i < count($answers_array); $i++) {
    $answers_array[$i] = (int)$answers_array[$i];
}
$server = 'localhost';
$user = 'root';
$password = "";
$dbname = 'quick_quiz';
$connection = mysqli_connect($server, $user, $password);
mysqli_select_db($connection, $dbname);
if (!$connection) {
    echo 'Connection to database failed';
}
#inserting into the database
$i = 0;
while ($i < $number_of_questions) {
    $insertion_query = "UPDATE user_input SET user_answer = '$answers_array[$i]' WHERE quiz_id = '$quiz_id' and quiz_name = '$quiz_name' and question_id = $i+1";
    if (!mysqli_query($connection, $insertion_query)) {
        echo "Error in insertion" . mysqli_error($connection);
    }
    $i++;
}
include('myscore.php');
?>
```

script.php

Converting user inputs to javascript array from php array and then updating the questions and options for the next question

```
<script>
const questionsArray = <?php echo json_encode($questions_array); ?>; //converting a php array to a js array
const numberOfQuestions = parseInt(<?php echo $number_of_questions ??>);
var clickCount = 0;
let question = document.querySelector('#question');
let optionLabels = document.querySelectorAll('label');
let options = document.querySelectorAll('input[type = "radio"]');
let label = document.querySelectorAll('.option');
let score = document.getElementById('score');
let ansArr = [];
let nextBtn = document.getElementById('next-btn');
window.onload = (update_questions());
//updating the questions and options
function update_questions() {
    let selected = 0;
    if (clickCount < numberOfQuestions) {
        question.textContent = questionsArray[clickCount].q;
        optionLabels[0].textContent = questionsArray[clickCount].a;
        optionLabels[1].textContent = questionsArray[clickCount].b;
        optionLabels[2].textContent = questionsArray[clickCount].c;
        optionLabels[3].textContent = questionsArray[clickCount].d;
        score.innerHTML = questionsArray[clickCount].sc + ' marks';
    }
    for (let i = 0; i < 4; i++) {
        label[i].className = 'option';
        if (options[i].checked) {
            ansArr.push(options[i].value);
            options[i].checked = false;
            selected = 1;
        }
    }
    if (selected == 0 && clickCount != 0) {
        ansArr.push("0");
    }
    clickCount++;
    if (clickCount === numberOfQuestions + 1) {
        let important = document.querySelector('#important');
        var ansString = ansArr.join("");
        important.value = ansString;
    }
    console.log(ansArr);
}
```

Code for displaying timer

```
window.onload = (clock());
var flag = 0;
var qns = 1;
document.getElementById("qn_no").innerHTML = "1 out of " + numberOfQuestions + " Questions";
document.getElementById("auto_next").style.visibility = "hidden";
document.getElementById("submit").style.visibility = "hidden";

function clock() {
    var seconds = document.getElementById("countdown").innerHTML = 30;
    if (numberOfQuestions == 1) {
        document.getElementById("next-btn").innerHTML = "Submit";
    }
    var countdown = setInterval(function() {
        seconds--;
        document.getElementById("countdown").innerHTML = seconds;
        if((seconds <= 30) && (seconds > 15))
        {
            document.getElementById("countdown").style.backgroundColor = "#5DD206";
        }
        else if((seconds <= 15) && (seconds > 5))
        {
            document.getElementById("countdown").style.backgroundColor = "#C8F740";
        }
        else if(seconds <= 5 )
        {
            document.getElementById("countdown").style.backgroundColor = "#F72721";
            document.getElementById("countdown").style.animation = "clockpulse 1s infinite";
        }

        if ((seconds < 0 || flag == 1)) {
            document.getElementById("countdown").style.webkitAnimationPlayState = 'paused';
            clearInterval(countdown);
            update_questions();
            qns++;
            if (qns <= numberOfQuestions) {
                document.getElementById("qn_no").innerHTML = qns + " out of " + numberOfQuestions + " Questions";
                // document.getElementById("qn_no").innerHTML = qns + " / " + numberOfQuestions;
            }
            flag = 0;
            document.getElementById("auto_next").click();
            document.getElementById("countdown").style.backgroundColor = "#5DD206";
        }
    }, 1000);
}
```

```
function auto_timer() {
    if (qns < numberOfQuestions) {
        clock();
    } else if (qns == numberOfQuestions) {
        document.getElementById("next-btn").innerHTML = "Submit";
        clock();
    } else {
        document.getElementById("submit").click();
        document.getElementById("countdown").innerHTML = "0";
    }
}

function next_timer() {
    flag = 1;
}

label[0].addEventListener('click' , () => {change_options(0)} );
label[1].addEventListener('click' , () => {change_options(1)} );
label[2].addEventListener('click' , () => {change_options(2)} );
label[3].addEventListener('click' , () => {change_options(3)} );

function change_options(i) {
    label[i].className = 'checked';
    for ( let j =0 ; j < 4 ; j++ ) {
        if( j === i)
        continue;
        label[j].className = 'option';
    }
}

</script>
```

cards.css

Styling for displaying available quizzes in the form of cards

```
.container{  
    display: grid;  
    grid-template-columns: repeat(auto-fit,minmax(13rem,1fr));  
    grid-template-rows: auto;  
    grid-gap:2em;  
}  
.card{  
    background-color: rgb(40, 40, 48);  
    color: rgb(150,150,150);  
    border-radius: 10px;  
    padding:1rem;  
    box-shadow: rgb(180, 41, 122) 0 0 90px;  
    border:4px double rgb(224, 130, 224);  
    transition: all 300ms ease-in;  
}  
.card:hover{  
    cursor: pointer;  
    color: white;  
}  
.card > h1{  
    font-weight: 800;  
    font-family: 'Zen loop';  
}  
.card > p{  
    font-family: 'Roboto Condensed';  
}  
#author{  
    font-family: 'Zen loop';  
    font-size: 2rem;  
    text-align: end;  
    margin-bottom: 0px;  

```

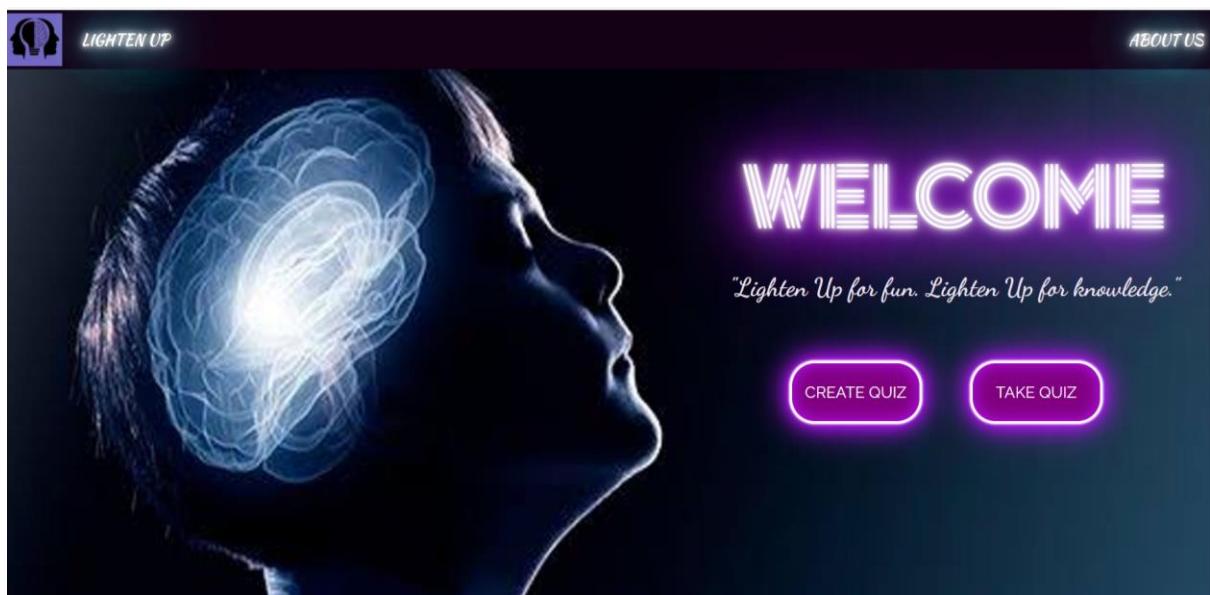
update_quiz_questions.php

Updating the new quiz created by user to database

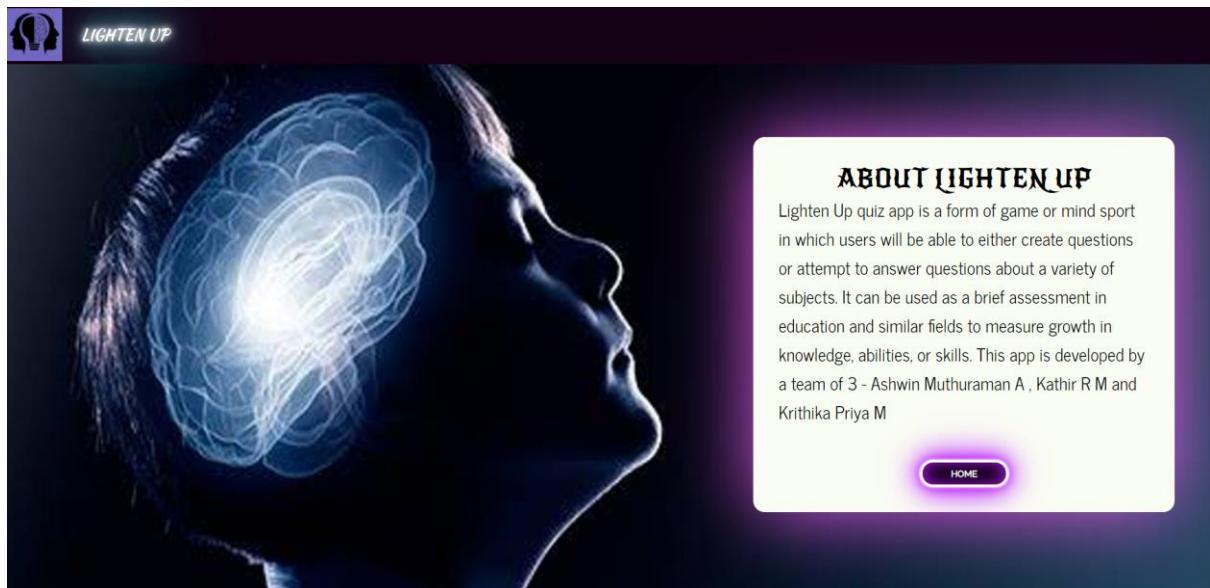
```
$insert_into_quiz_list = "INSERT INTO quiz_list (quiz_name, author_name, quiz_description) VALUES  
('{$quiz_name}', '{$author_name}', '{$quiz_description}')";  
if(!mysqli_query($connection, $insert_into_quiz_list)) {  
    echo "ERROR IN INSERTING" . mysqli_error($connection);  
}  
$select_quiz_id = "SELECT * FROM quiz_list";  
$quiz_id = mysqli_query($connection, $select_quiz_id);  
if(!$quiz_id) {  
    echo "ERROR IN SELECTING QUIZ_ID" . mysqli_error($connection);  
}  
  
$count_list = 0;  
while($row = mysqli_fetch_assoc($quiz_id)) {  
    $count_list++;  
}  
  
for($i = 0 ; $i < $num_of_qns ; $i++) {  
    $insert_questions = "INSERT INTO question_table VALUES ($count_list, '$quiz_name', $i+1, '$questions[$i]',  
    '$option_a[$i]', '$option_b[$i]', '$option_c[$i]', '$option_d[$i]', '$scores[$i]')";  
    if(!mysqli_query($connection, $insert_questions)) {  
        echo "ERROR INSERTING QUESTION " . $i+1;  
    }  
    $insert_user_answers = "INSERT INTO user_input VALUES ($count_list, '$quiz_name', $i+1, 0)";  
    if(!mysqli_query($connection, $insert_user_answers)) {  
        echo "ERROR MAY OCCUR WHILE USER INPUTS ANSWERS AT QUESTION " . $i+1;  
    }  
}  
  
for($i = 0 ; $i < $num_of_qns ; $i++) {  
    $insert_answers = "INSERT INTO answer_table VALUES ($count_list, '$quiz_name', $i+1, '$correct_ans[$i]')";  
    if(!mysqli_query($connection, $insert_answers)) {  
        echo "ERROR INSERTING ANSWER " . $i+1;  
    }  
}  
?>
```

OUTPUT

HOME PAGE:



ABOUT US:



TAKE QUIZ (before creating a new quiz):

The interface shows three quiz cards on a dark background with a circuit board pattern:

- National Freedom Fighters**
A quiz on the great personalities who sacrificed their lives selflessly FOR Indian Independence
Made by Krithika Priya
- College of Engineering Guindy**
A quiz on the oldest technical institute of Asia - College of Engineering Guindy - CEG
Made by Ashwin Muthuraman
- Javascript and PHP**
A quiz on the basics of Javascript and PHP languages
Made by Kathir

CREATE QUIZ:

The interface shows a quiz creation form on a dark background with a circuit board pattern:

QUIZ DETAILS

QUIZ NAME : HTML and CSS
AUTHOR NAME : Michael
DESCRIPTION: A general quiz on the basic of HTML and CSS

INSTRUCTIONS: Fill out the questions and the options below it and also select the option containing the correct answer.

QUESTION 1

Which of the following is an inline element?

OPTION A
 OPTION B
 OPTION C
 OPTION D

Enter the score 2

CREATE **ADD ONE MORE QUESTION**

ADD ONE MORE QUESTION:

The interface shows a quiz creation screen with a circuit board background. It displays two questions:

QUESTION 1

What is the default value of the position property?

QUESTION 2

What is the default value of the position property?

OPTIONS FOR QUESTION 1:

- OPTION A: div tag
- OPTION B: section tag
- OPTION C: anchor tag
- OPTION D: p tag

Enter the score: 2

OPTIONS FOR QUESTION 2:

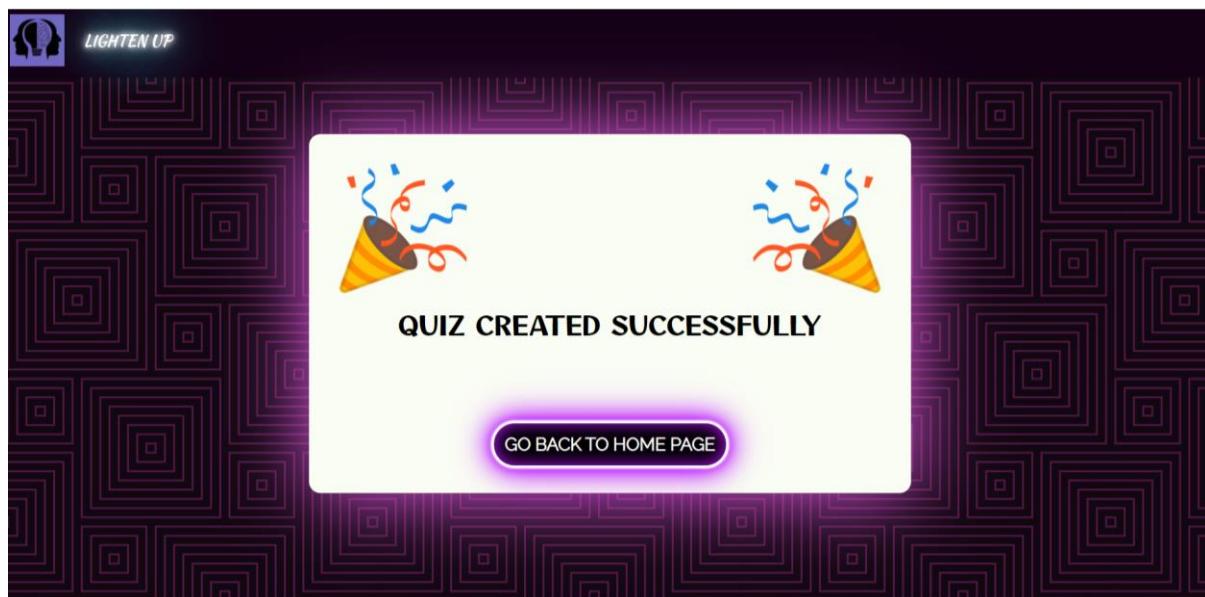
- OPTION A: relative
- OPTION B: static
- OPTION C: fixed
- OPTION D: absolute

Enter the score: 3

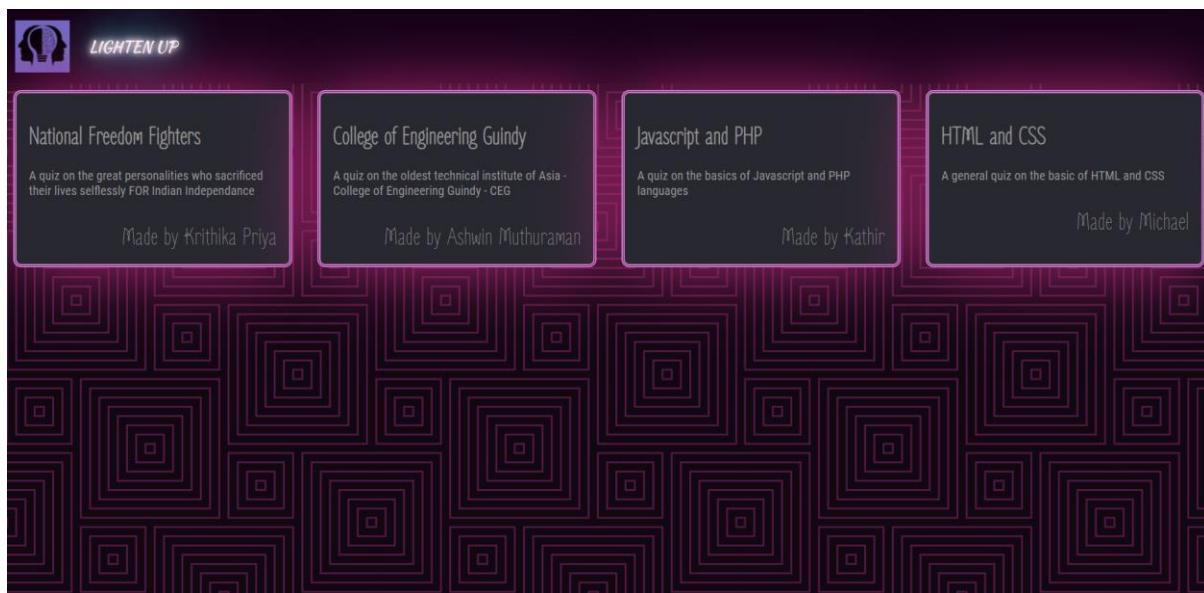
Buttons:

- [CREATE](#)
- [ADD ONE MORE QUESTION](#)

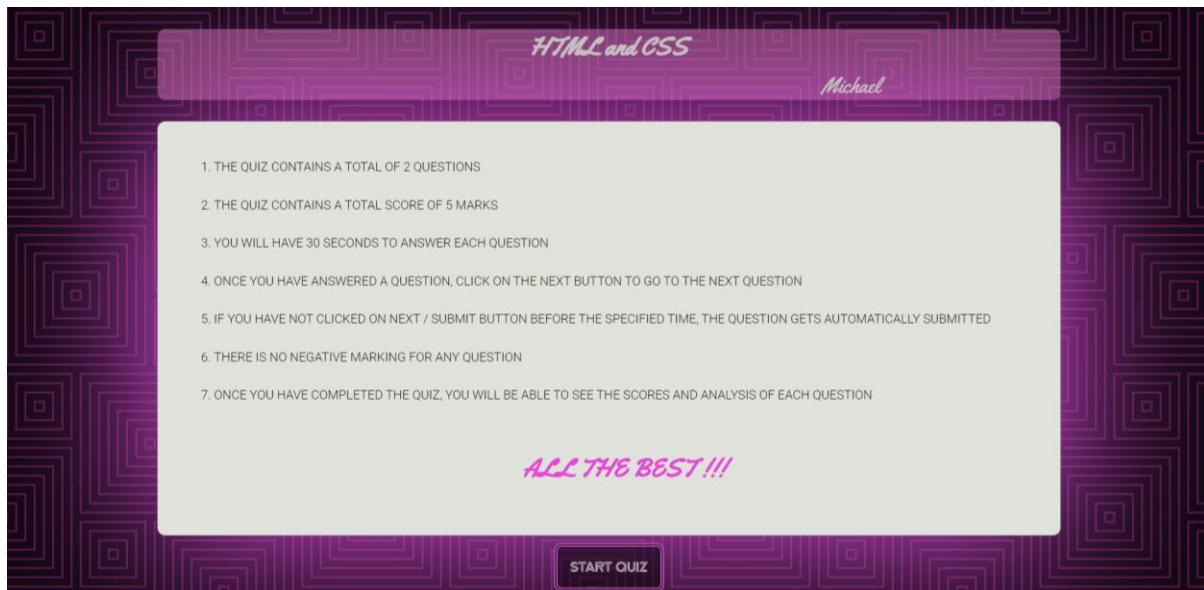
CREATE:



TAKE QUIZ (after creating a new quiz):



AFTER SELECTING A QUIZ:



START QUIZ:

1 out of 2 Questions

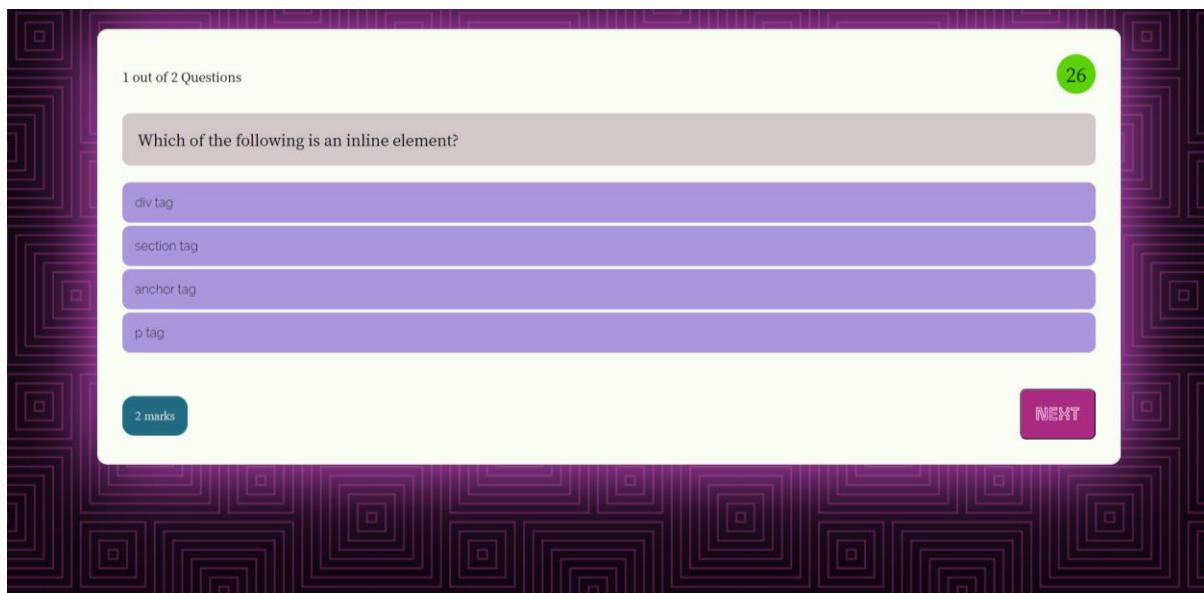
26

Which of the following is an inline element?

- div tag
- section tag
- anchor tag
- p tag

2 marks

NEXT



2 out of 2 Questions

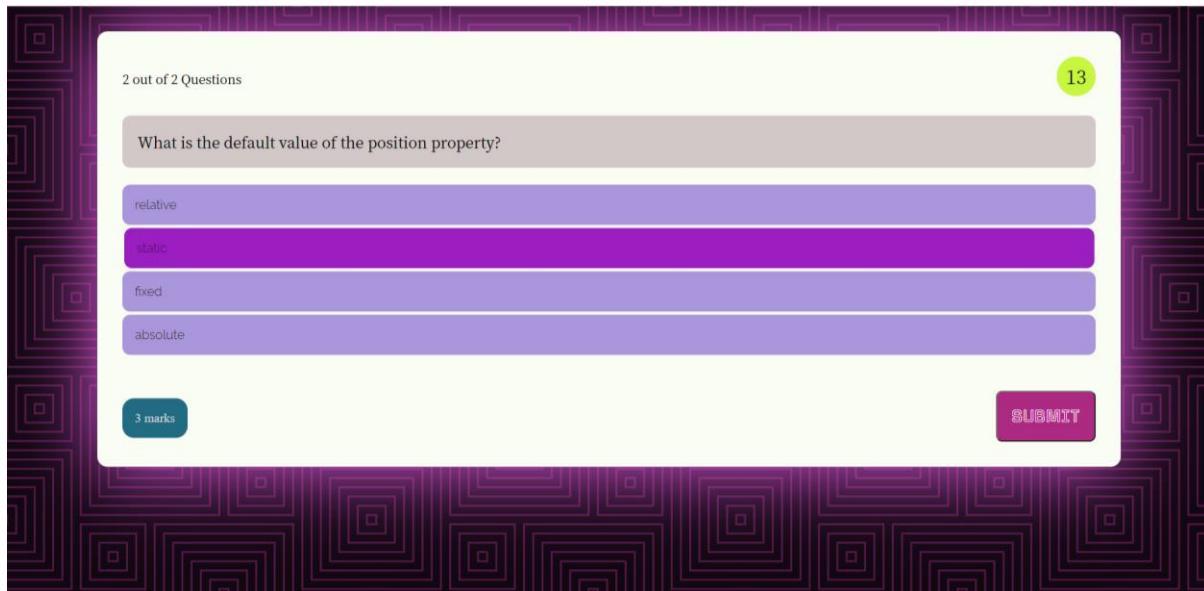
13

What is the default value of the position property?

- relative
- static
- fixed
- absolute

3 marks

SUBMIT



SUBMIT:

YOUR TOTAL SCORE IS : 5 / 5

Score: 2/2

Status: Correct

1 . Which of the following is an inline element?

div tag

section tag

anchor tag

p tag

Score: 3/3

Status: Correct

2 . What is the default value of the position property?

relative

static

fixed

absolute

[HOME](#)

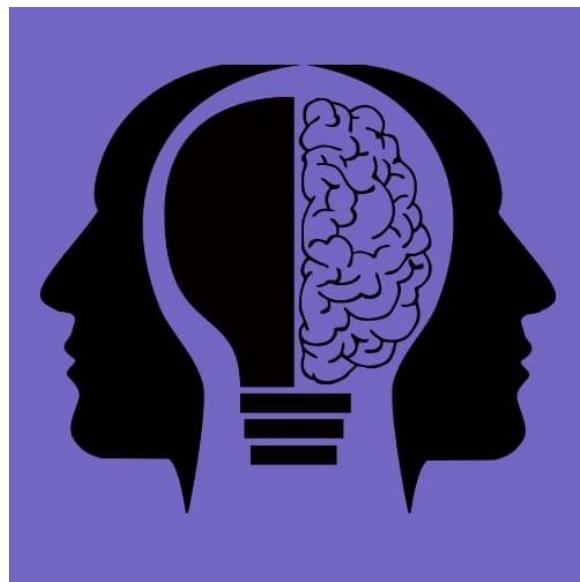
FURTHER SCOPE OF DEVELOPMENT

- + Proper user registration (login / signup feature) can be added for users, both for creating and attempting the quiz which enables to keep a track on the user works.
- + Different levels can be added and the time constraint can be changed accordingly. Also, the app can be further developed such that the quiz can contain different type of questions (fill-ups, one words, etc).
- + Author who creates the quiz can be provided with a link of the quiz that they created so that they can share it with required people.
- + Separate pages for each subject can be set so that one can easily search for a quiz in a particular subject.
- + A search bar can be added if one has to find a particular quiz with either the quiz name or the author's name or even the description.
- + Sound effects can be added to the timer for making the app more attractive.
- + Full screen mode when the quiz is being taken can be activated, so that user cannot switch to any other application, while attempting the quiz.
- + Scheduling calendar can be added, so that the user can schedule to take a quiz on a particular date and time and he/she will receive notification for it.
- + The app can further be developed into a proper online exam app.

CONCLUSION

The app runs successfully and produces satisfactory results living up to the expectations of the user. We have achieved immense satisfaction for developing a successful and a good app in terms of effectiveness and efficiency, flexibility, adaptability, generality, robustness and reusability. We, all three of us, are very happy that we created the app with the basics what we learnt, by contributing equal works towards the development of app.

Lighten Up app, when further developed, can be launched successfully and can be used among all types of users. The app when used amongst growing children will definitely provide joy and excitement to the user. This also develops a competitive spirit among various users.



"Lighten Up for fun. Lighten Up for knowledge."