Experiment No. 06

Title: Event handling using JavaScript to explore web browser environment.

Batch: B1 Roll No.: 16010420133 Experiment No.:6

Aim: Event handling using JavaScript to explore web browser environments.

Resources needed: Notepad++, Web Browser

Theory:

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

An HTML web page has finished loading
An HTML input field was changed
An HTML button was clicked
Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

HTML allows event handler attributes, with JavaScript code, to be added to HTML elements.

What can JavaScript Do?
Event handlers can be used to handle, and verify, user input, user actions, and browser actions:

- Things that should be done every time a page loads
- Things that should be done when the page is closed
- Action that should be performed when a user clicks a button
- Content that should be verified when a user inputs data

Many different methods can be used to let JavaScript work with events:

- HTML event attributes can execute JavaScript code directly
- HTML event attributes can call JavaScript functions
- You can assign your own event handler functions to HTML elements

- You can prevent events from being sent or being handled

Syntax:

```
<element  event='some  JavaScript'>
<element  event="some  JavaScript">
```

Example

In the following example, an onclick attribute (with code), is added to a <button> element:

```
<!DOCTYPE html>
<html>
<body>
<button onclick="document.getElementById('demo').innerHTML=Date()">The time
is?</button>

<p id="demo"></p>

</body>
</html>
```

JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events:

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element. To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute: onclick=JavaScript

EXAMPLE

```
<!DOCTYPE html>
<html>
<body>

<h1 onclick="this.innerHTML='Ooops!'">Click on this text!</h1>

</body>
</html>
```

Activity:

Apply following JS events on your web pages

Input Events

- Onblur
- onreset

Mouse Events

- Onmouseover
- Onmousedown

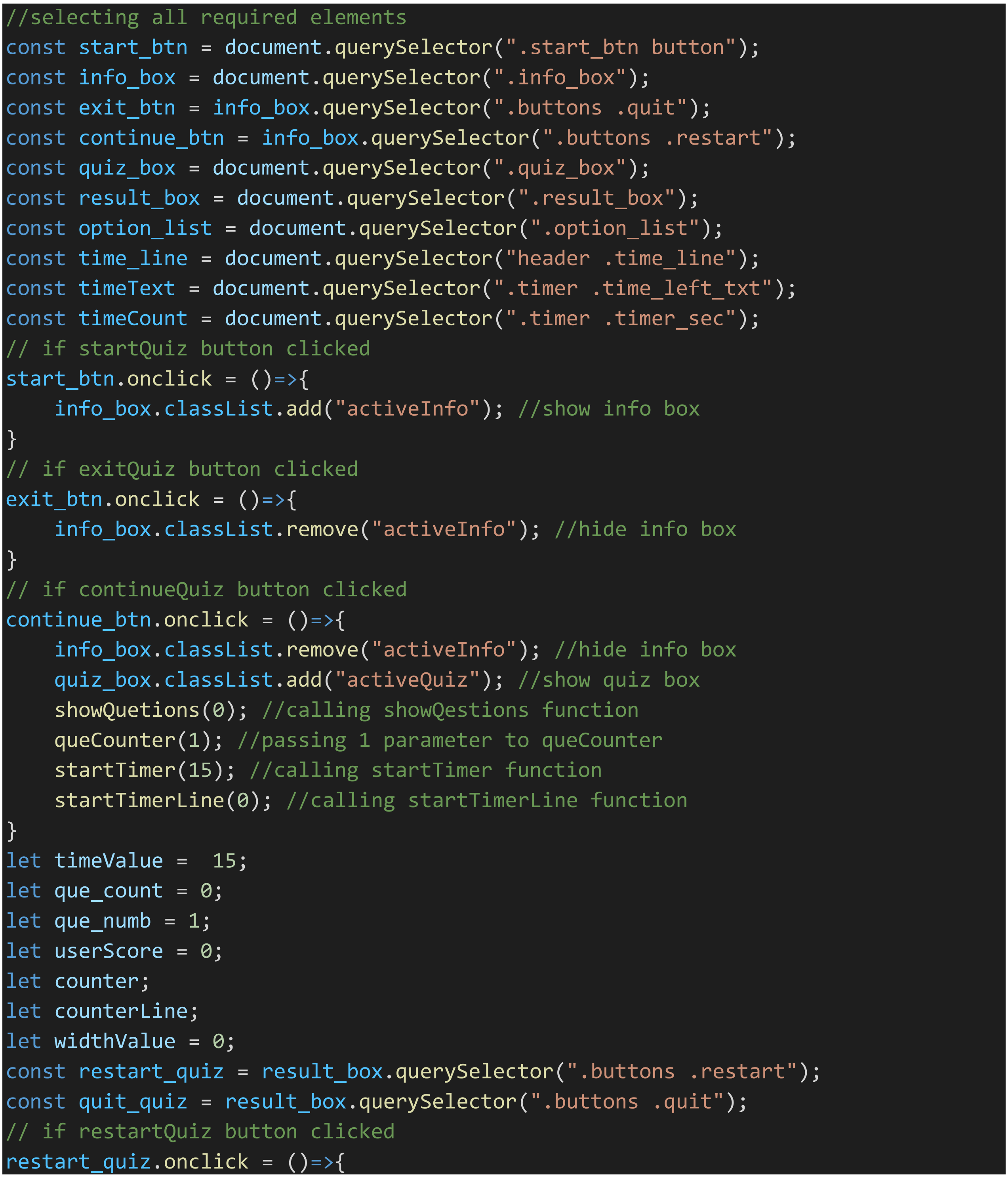
Click Events

- Onclick
- Ondbclick

Apply following DOM events to your webpages

- Onload
- Onchange
- onmouseover

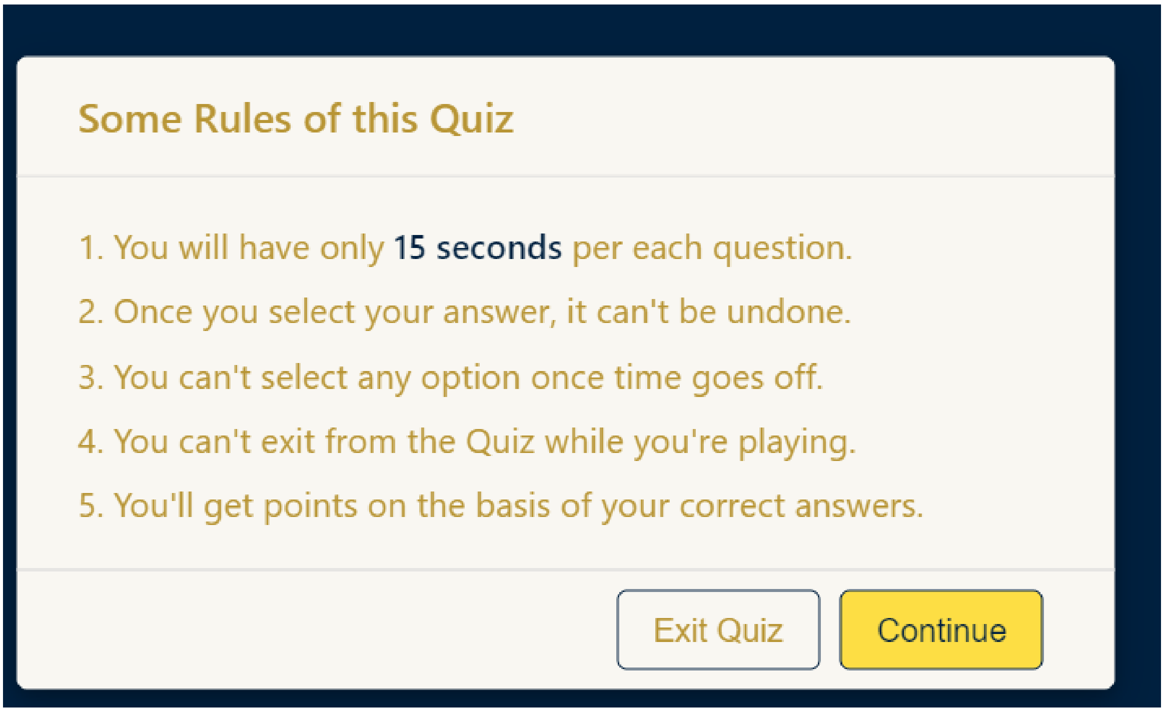
Results: (Program printout with output)



```
//selecting all required elements
const start_btn = document.querySelector(".start_btn button");
const info_box = document.querySelector(".info_box");
const exit_btn = info_box.querySelector(".buttons .quit");
const continue_btn = info_box.querySelector(".buttons .restart");
const quiz_box = document.querySelector(".quiz_box");
const result_box = document.querySelector(".result_box");
const option_list = document.querySelector(".option_list");
const time_line = document.querySelector("header .time_line");
const timeText = document.querySelector(".timer .time_left_txt");
const timeCount = document.querySelector(".timer .timer_sec");
// if startQuiz button clicked
start_btn.onclick = ()=>{
    info_box.classList.add("activeInfo"); //show info box
}
// if exitQuiz button clicked
exit_btn.onclick = ()=>{
    info_box.classList.remove("activeInfo"); //hide info box
}
// if continueQuiz button clicked
continue_btn.onclick = ()=>{
    info_box.classList.remove("activeInfo"); //hide info box
    quiz_box.classList.add("activeQuiz"); //show quiz box
    showQuetions(0); //calling showQestions function
    queCounter(1); //passing 1 parameter to queCounter
    startTimer(15); //calling startTimer function
    startTimerLine(0); //calling startTimerLine function
}
let timeValue =  15;
let que_count = 0;
let que_numb = 1;
let userScore = 0;
let counter;
let counterLine;
let widthValue = 0;
const restart_quiz = result_box.querySelector(".buttons .restart");
const quit_quiz = result_box.querySelector(".buttons .quit");
// if restartQuiz button clicked
restart_quiz.onclick = ()=>{
    quiz_box.classList.add("activeQuiz"); //show quiz box
    result_box.classList.remove("activeResult"); //hide result box
    timeValue = 15;
    que_count = 0;
    que_numb = 1;
    userScore = 0;
    widthValue = 0;
    showQuetions(que_count); //calling showQestions function
    queCounter(que_numb); //passing que_numb value to queCounter
    clearInterval(counter); //clear counter
    clearInterval(counterLine); //clear counterLine
    startTimer(timeValue); //calling startTimer function
    startTimerLine(widthValue); //calling startTimerLine function
    timeText.textContent = "Time Left"; //change the text of timeText to Time
Left
    next_btn.classList.remove("show"); //hide the next button
}
// if quitQuiz button clicked
quit_quiz.onclick = ()=>{
    window.location.reload(); //reload the current window
}
const next_btn = document.querySelector("footer .next_btn");
const bottom_ques_counter = document.querySelector("footer .total_que");
// if Next Que button clicked
next_btn.onclick = ()=>{
    if(que_count < questions.length - 1){ //if question count is less than total
question length
        que_count++; //increment the que_count value
        que_numb++; //increment the que_numb value
        showQuetions(que_count); //calling showQestions function
        queCounter(que_numb); //passing que_numb value to queCounter
        clearInterval(counter); //clear counter
        clearInterval(counterLine); //clear counterLine
        startTimer(timeValue); //calling startTimer function
        startTimerLine(widthValue); //calling startTimerLine function
        timeText.textContent = "Time Left"; //change the timeText to Time Left
        next_btn.classList.remove("show"); //hide the next button
    }else{
        clearInterval(counter); //clear counter
        clearInterval(counterLine); //clear counterLine
        showResult(); //calling showResult function
    }
}
// getting questions and options from array
function showQuetions(index){
    const que_text = document.querySelector(".que_text");
    //creating a new span and div tag for question and option and passing the
value using array index
```

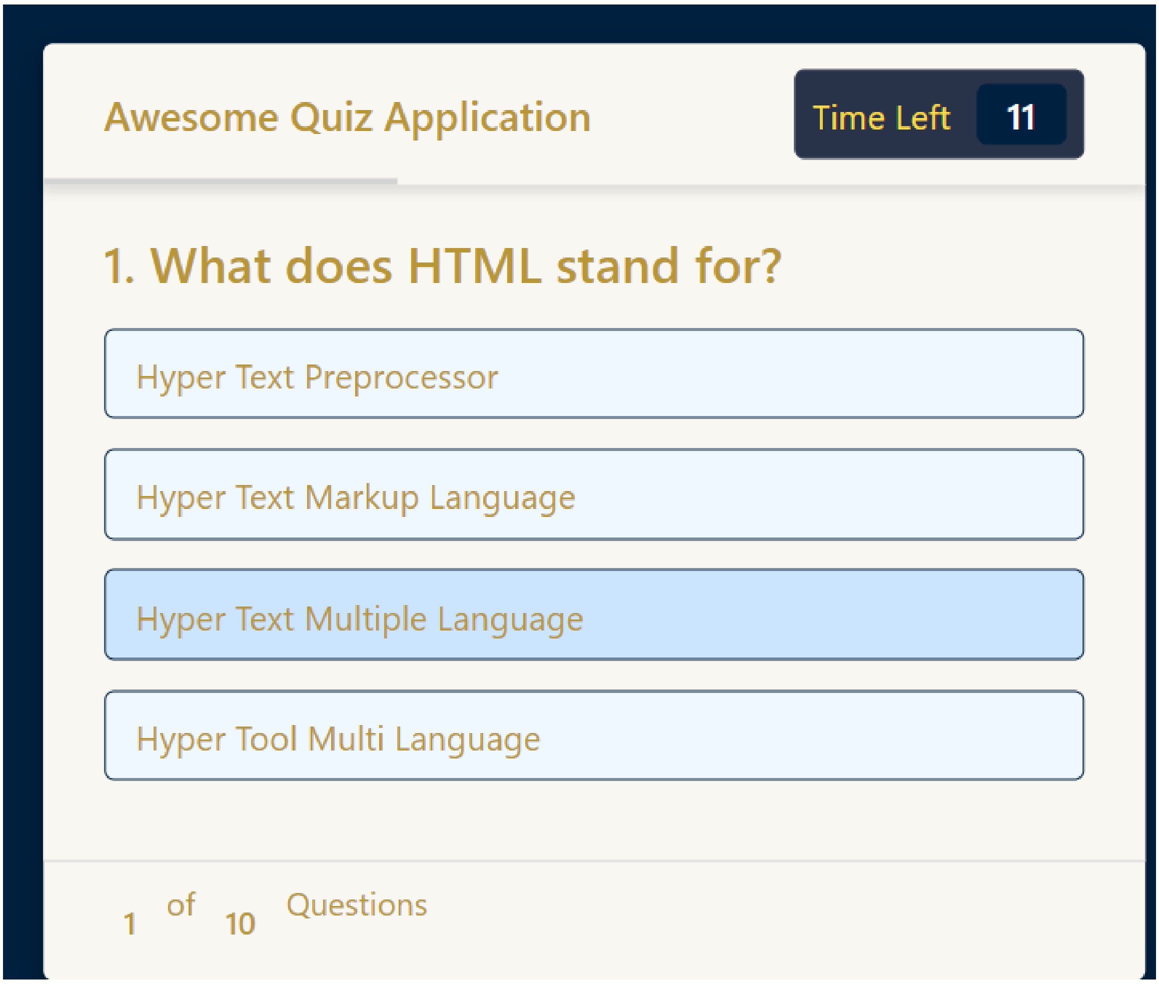
```
    let que_tag = '<span>'+ questions[index].numb + ". " +
questions[index].question +'</span>';
    let option_tag = '<div class="option"><span>'+ questions[index].options[0]
+'</span></div>'
    + '<div class="option"><span>'+ questions[index].options[1] +'</span></div>'
    + '<div class="option"><span>'+ questions[index].options[2] +'</span></div>'
    + '<div class="option"><span>'+ questions[index].options[3] +'</span></div>';
    que_text.innerHTML = que_tag; //adding new span tag inside que_tag
    option_list.innerHTML = option_tag; //adding new div tag inside option_tag

    const option = option_list.querySelectorAll(".option");
    // set onclick attribute to all available options
    for(i=0; i < option.length; i++){
        option[i].setAttribute("onclick", "optionSelected(this)");
    }
}
// creating the new div tags which for icons
let tickIconTag = '<div class="icon tick"><i class="fas fa-check"></i></div>';
let crossIconTag = '<div class="icon cross"><i class="fas fa-times"></i></div>';
//if user clicked on option
function optionSelected(answer){
    clearInterval(counter); //clear counter
    clearInterval(counterLine); //clear counterLine
    let userAns = answer.textContent; //getting user selected option
    let correcAns = questions[que_count].answer; //getting correct answer from
array
    const allOptions = option_list.children.length; //getting all option items

    if(userAns == correcAns){ //if user selected option is equal to array's
correct answer
        userScore += 1; //upgrading score value with 1
        answer.classList.add("correct"); //adding green color to correct selected
option
        answer.insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon
to correct selected option
        console.log("Correct Answer");
        console.log("Your correct answers = " + userScore);
    }else{
        answer.classList.add("incorrect"); //adding red color to correct selected
option
        answer.insertAdjacentHTML("beforeend", crossIconTag); //adding cross icon
to correct selected option
        console.log("Wrong Answer");
        for(i=0; i < allOptions; i++){
            if(option_list.children[i].textContent == correcAns){ //if there is
an option which is matched to an array answer
                option_list.children[i].setAttribute("class", "option correct");
//adding green color to matched option
```

```
                option_list.children[i].insertAdjacentHTML("beforeend", tickIconTag); //adding tick icon to matched option
                console.log("Auto selected correct answer.");
            }
        }
    }
    for(i=0; i < allOptions; i++){
        option_list.children[i].classList.add("disabled"); //once user select an option then disabled all options
    }
    next_btn.classList.add("show"); //show the next button if user selected any option
}
function showResult(){
    info_box.classList.remove("activeInfo"); //hide info box
    quiz_box.classList.remove("activeQuiz"); //hide quiz box
    result_box.classList.add("activeResult"); //show result box
    const scoreText = result_box.querySelector(".score_text");
    if (userScore > 3){ // if user scored more than 3
        //creating a new span tag and passing the user score number and total question number
        let scoreTag = '<span>and congrats! , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;  //adding new span tag inside score_Text
    }
    else if(userScore > 1){ // if user scored more than 1
        let scoreTag = '<span>and nice , You got <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;
    }
    else{ // if user scored less than 1
        let scoreTag = '<span>and sorry , You got only <p>'+ userScore +'</p> out of <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;
    }
}
function startTimer(time){
    counter = setInterval(timer, 1000);
    function timer(){
        timeCount.textContent = time; //changing the value of timeCount with time value
        time--; //decrement the time value
        if(time < 9){ //if timer is less than 9
            let addZero = timeCount.textContent;
            timeCount.textContent = "0" + addZero; //add a 0 before time value
        }
        if(time < 0){ //if timer is less than 0
            clearInterval(counter); //clear counter
            timeText.textContent = "Time Off"; //change the time text to time off
            const allOptions = option_list.children.length; //getting all option
items
            let correcAns = questions[que_count].answer; //getting correct answer
from array
            for(i=0; i < allOptions; i++){
                if(option_list.children[i].textContent == correcAns){ //if there
is an option which is matched to an array answer
                    option_list.children[i].setAttribute("class", "option
correct"); //adding green color to matched option
                    option_list.children[i].insertAdjacentHTML("beforeend",
tickIconTag); //adding tick icon to matched option
                    console.log("Time Off: Auto selected correct answer.");
                }
            }
            for(i=0; i < allOptions; i++){
                option_list.children[i].classList.add("disabled"); //once user
select an option then disabled all options
            }
            next_btn.classList.add("show"); //show the next button if user
selected any option
        }
    }
}
function startTimerLine(time){
    counterLine = setInterval(timer, 29);
    function timer(){
        time += 1; //upgrading time value with 1
        time_line.style.width = time + "px"; //increasing width of time_line with
px by time value
        if(time > 549){ //if time value is greater than 549
            clearInterval(counterLine); //clear counterLine
        }
    }
}
function queCounter(index){
    //creating a new span tag and passing the question number and total question
    let totalQueCounTag = '<span><p>'+ index +'</p> of <p>'+ questions.length
+'</p> Questions</span>';
    bottom_ques_counter.innerHTML = totalQueCounTag;  //adding new span tag
inside bottom_ques_counter
}
```



Some Rules of this Quiz

1. You will have only **15 seconds** per each question.
2. Once you select your answer, it can't be undone.
3. You can't select any option once time goes off.
4. You can't exit from the Quiz while you're playing.
5. You'll get points on the basis of your correct answers.



Awesome Quiz Application

Time Left 11

1. What does HTML stand for?

Hyper Text Preprocessor

Hyper Text Markup Language

Hyper Text Multiple Language

Hyper Tool Multi Language

1 of 10 Questions

Questions:

Q1) What are the different types of load events

- DOMContentLoaded – the browser fully loaded HTML and completed building the DOM tree. However, it hasn’t loaded external resources like stylesheets and images. In this event, you can start selecting DOM nodes or initialize the interface.
- load – the browser fully loaded the HTML and also external resources like images and stylesheets.

When you leave the page, the following events fire in sequence:

- beforeunload – fires before the page and resources are unloaded. You can use this event to show a confirmation dialog to confirm if you really want to leave the page. By doing this, you can prevent data loss in case you are filling out a form and accidentally click a link to navigate to another page.
- unload – fires when the page has completely unloaded. You can use this event to send the analytic data or to clean up resources.

Q2) Explain Onkeypress , onkeyup events

onkeyup event

The onkeyup event executes a JavaScript function when the user releases a key. The onkeyup method allows derived classes to handle the event without attaching the delegate. This is the preferred technique for handling the event in derived classes.

onkeypress event

The keypress event only fires for keys that have a printable representation. The onkeypress event is not fired for all keys, for example for ALT, SHIFT, CTRL, ESC and BACKSPACE in all browsers, this event is fired only when the user presses a key.

Outcomes:

CO 3 Apply Javascript and Json for Web Application development

Conclusion: (Conclusion to be based on the outcomes achieved)

Applied Js on Webpage

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- "Web technologies: Black Book", Dreamtech Publications
- http://www.w3schools.com