GUJARAT TECHNOLOGICAL UNIVERSITY

Integrated MCA

Year – 1 (Semester – I) (W.E.F. JULY 2018)

Subject Name: Fundamentals of Web (FoW)
Subject Code: 2618602

1. **Learning Objectives:**

- Students will learn about the opportunities, challenges and techniques for developing websites built with the new resources provided by HTML5, CSS and JavaScript.
- Students will learn about the evolving principles and standards for constructing accessible websites; will understand different classes of disabilities and the available techniques for rendering websites useful to those with disabilities

2. **Pre-requisites:** Working Knowledge of internet and HTML

3. **Course Contents:**

Sr. No.	CourseContent	Weightage Percentage
1	**Introduction to Internet** Evolution of teh internet and World wide Web, Web Basics(hyperlinks, URI,URL, Ports of a URL, Http headers, Http Get and Post requests), Multitier Application Architecture, Client-Side versus Server-side Scripting, Word Wide Web Consortium (W3C), Web 2.0	**5%**
2	**HTML** **HTML Basics:** Introduction, Editing HTML5, First HTML5 Example, W3C HTML validation service, Headings, Linking, Images Special Characters and Horizontal Rules, Lists, Tables, Forms, Internal Linking, meta Elements **HTML5:** New HTML5 input Types, input and data list ELements and auto complete Attributes, Page Structure ELements	**25%**
3	Cascading Style Sheets (CSS) **CSS Basics:** Introduction, inline Styles, Embedded Style Sheets, Conflicting sheets, Linking External Style Sheets, Positioning Elements, Backgrounds, Element Dimensions, Box Model and Text flow, Media Types and Media Queries, Drop-Down Menus, Optional: User Style Sheets **CSS advance:** Text Shadows, Rounded corners, Color, Box Shadows, Linear Gradients, Radial Gradients**,** Multiple background Images	**25%**

4	**JavaScript Basics** Introduction, Your first Script, Modifying Script, Obtaining User Input with prompt Dialogs, Memory Concepts, Arithmetic, Equality and relational operators, Control structures (if statements, while statement, for statement, switch statement, do..while statements), Logical Operators, Assignment Operators, Break/Continue statements, Functions, Arrays	**20%**
5	**JavaScript Advanced** **JavaScript Objects:** Introduction, Math Object, String Object, Date Object, Boolean and Number Objects, **Document Object Model:** Introduction, DOM Nodes and trees, Traversing and modifying a DOM tree, DOM collections, Dynamic Styles, Using a TImer and Dynamic Styles to create Animated Effects **Java Script Event Handling:** Introduction, Reviewing the load event, Event the mousemove and the event object, Rollover with mouseover and mouseout, From Processing with focus, blur, submit and reset, Event bubbling	**25%**

4. Text Books:

i. Paul Deital Harvey Deitel, Abbey Deitel, Internet & World Wide Web: How to Program, 5th Edition, Pearson

5. Reference Books:

i. Rob Larsen, Beginning HTML and CSS, WROX

ii. Ivan Bayross, “*Web Enabled Commercial Application Development using HTML, DHTML, JavaScript, Perl CGI*”, Third revised edition, BPB Publication

iii. Faithe Wempen, “Step by Step HTML 5”, South Asian Edition, Microsoft Press and PHI Learning

iv. Wendy Willard, “HTML: A Beginner's Guide 5/E”, 5th Edition, McGraw Hill

v. Teach yourself Java Script in 24 by Michael Moncur Publisher: Pearson Education

vi. HTML Black Book by Steven Holzner, Publisher: Dreamtech Press

6. Chapter wise Coverage from Textbook(s):

Unit #	**Chapters**
Unit 1	Chapter 1 (1.5, 1.6,1.7,1.8, 1.9,1.10
Unit 2	Chapter 2, 3
Unit 3	Chapter 4, 5 (5.2 to 5.7)
Unit 4	Chapter 6,7,8,9,10
Unit 5	Chapter 11,12,13

Additional suggested Tutorial/ Case studies:

1) Tutorial on Location Awareness:

Geolocations Overview: determining and mapping your location with getcurrentposition,, Determining distance with Position, Options, Following a moving location with watchPosition

2) Tutorial/Seminar/Workshop on Front end framework:

Overview of any front end framework for project like BootStrap

3) Case study:

1. Using all the new elements to Build a news Page
2. Using all the new elements to Build a search results Page
3. Marking up an article page with comments
4. Create a web form using all HTML5 input types
5. Using HTML5 and Bootstrap prepare your website

7. Accomplishments of the student after completing the course:

- gain an insight into web site development using HTML, CSS and JavaScript

Indicative Practical List

A. List of Practical Related to HTML:

1	Create an HTML file which makes use of various Heading tags.
2	Create an HTML file which makes use of paragraph and line break.
3	Create an HTML file which is having following Text Formatting: a. Bold b. Strong c. Big d. Italic e. Small f. Subscript g. Superscript h. Underline
5	Create an HTML file which makes use of different types of links.
6	Create an HTML page which Apply at least 6 different string styles to single text and display them.
7	Create an HTML file which Creates different types of Menu.
8	**Lists:** Create a web page which makes use of Numbered, Letters, Lowercase Letters, Roman Numbers, Lowercase Roman numbers List
9	**Link:** Create a web page showing an ordered list of names of the subjects, with nested list if any subject has been selected it should display the content of each subject
10	**Tables and Lists:** Write HTML code to display your bio-data using different types of lists and tables
11	**List and Tables :** Write HTML code to display Restaurant Menu using different types of lists and tables
12	Suppose your company has three employees and you want to create a company “directory page” listing some information about each of them. Write the HTML for that page and link 1 employee to another employee
13	Create a specimen of a corporate web page. Divide the Browser Screen into two frames. The frame on the left will be the menu consisting of Hyperlinks. Clicking on any of this link will lead to a new page which must open in the target frame which is on the right side.
14	Create Student Registration form with all form elements.

15	Write Html code which gives information of different cities in Gujarat (like Gujarat Tourism web site), when user click on any of the cities from left frame, the information about respective cities should appear on right frame a. Use frame tag Split web page with frames taking the left 25% of browser window, If clicking each name corresponding web page in right 75% of web browser window b. Also display the tourist spot of Gujarat state
16	Create a web page with appropriate content and insert an image towards the left hand side of the page when user clicks on the image, it should open another web page with enlarged image
17	Create Web Page for yourself which display following section (Apply all HTML features learned). 1) Personal Information (Name, Dob, Gender, COntact Number, Email Address) 2) Education Details (Board, School Name, Pass Year, Percentage) 3) Hobbies, Languages Known 4) Awards / Certification 5) Others
18	Write an Html application implementing Website navigation similar to Flipkart.com,
19	Write an Html application similar to photo gallery for any city

Part II: CSS

1	Employ different types of Cascading Style Sheet.
2	Employ Cascading Style Sheet in HTML tags.
3	Employ Cascading Style Sheet using class.
4	Employ Cascading Style Sheet using ID.
5	Employ Cascading Style Sheet on Text, Text Box, Background color
6	Apply CSS on Web Page prepared (Practical #13 of Part I)
7	Create a CSS rule that can make the following changes to the Text. i) Font size : 16 ii) Font color : Red iii) Font family : Arial
8	Create a CSS Library (external library file) and use in webpage.

Part III: JavaScript

1	Write a program in Java Script which accepts names in a text box, if a button is clicked names should be sorted and added in another one text area
2	Write Java Script code to represent Document object
3	Count the number of elements in a form’s element array. Display no of elements residing in a From & display name of each.(Include two Forms in your HTML file) Include Radio and Checkbox Button in html file & show their status (i.e., checked or unchecked depending on the user‘s event) in a msgbox.. Also, clears all the selection using Reset.
4	Write a JavaScript to generate two random numbers and find out maximum and minimum out of it.
5	Write a program in Java Script which contains 3 functions which are invoked on clicking the Red, Blue and green buttons. The function should contain changing the

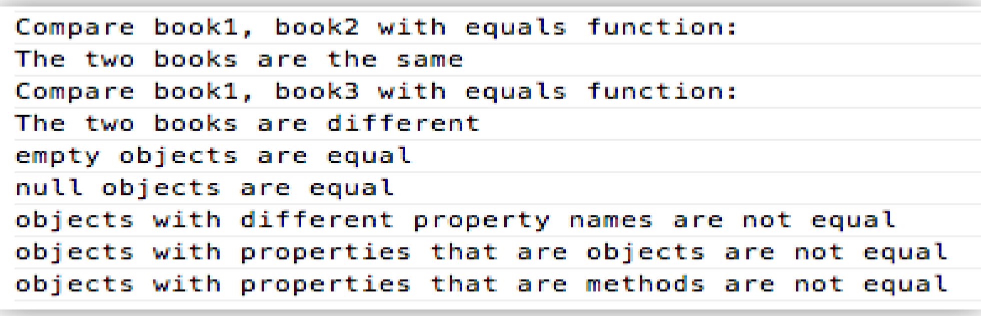
	background, foreground to respective color and to display corresponding status messages
6	Write a JavaScript code block using arrays and generate the current Date in words, this should include the Day, the Month and the Year. O/P should be as: Monday, August 16, 2010
7	Write a program in Java Script that allows user to enter the text. It also allows the user to accept size and font name that has to be applied on the text entered by the user.
8	Represent all properties and methods of Location object in Java Script (desirable)
9	Display a digital clock on your web page (Not in Textbox)
10	Create a Form in HTML with two fields, minimum and maximum, write JavaScript to validate that only numeric value is entered in both, and the value entered in minimum is less than the value entered in maximum.
11	Get the username from prompt after 2 seconds display the welcome name in msgbox after 5 seconds.
12	Write a code which checks the contents entered in a Form's Text element. If the text entered is in the Lowercase, convert to Upper case.
13	Make a function that squares a number. Test it.
14	Create Simple Calculator using JavaScript.
15	Create Scientific Calculator using JavaScript.
16	Develop a HTML page which accepts: a)Any Mathematical Expression b) Evaluates the Expression c) Display the Result of the Evaluation
17	Create a Web Page using two image files, which switch between one another as the mouse pointer moves over the images.
18	Write a program in Java Script which allows certain fields like First Name, Last name, email address, comments. Perform certain validations like first name and last name should not be empty and email should be valid. If user clicks the submit button it should open a new window and contents should be displayed. If reset contents should be cleared

Part IV: All

1	Develop a webpage using java Script which has following fields like Source, Destination; train no, Date and Number of tickets. a. Source and destination should allow only place code in 3 character b. Date should be in the format DD/ MM / YY c. Number of tickets should allow only numeric Apply CSS on various components of webpage.
2	Write a program in Java Script which allows certain fields like Name, Age, Gender, Age, Cite, State and Country. Perform certain validations like name should accept only alphabets, Gender should accept only 1 character, Age should be only in numeric between 1 and 100.
3	Write a HTML to create a "guestbook" from that asks someone for his/her name, sex, age, email address
4	Create a Guess Number Page. The user is given five chances to guess the number between 1 to 100. Computer guess the number (range 1 to 100) and prompt for User to enter number. If user entered number is on higher side it will prompt a hint. Similarly gives hint message for lower side entered number.

5	Write a webpage for Youth Festival registration (Apply validation usign JavaScript and apply internal , inline and external CSS).
6	Create a multiplication table asking the user the number of rows and columns he/she wants.
7	Create a form that collects the first name, last name, email, user id, password and confirm password from the user. All the inputs are mandatory and email address entered should be in correct format. Also, the values entered in the password and confirm password textboxes should be the same. After validating using JavaScript, display proper error messages in red color just next to the textbox where there is an error.
8	Create Java Script Project you'll write a function**, trim (),** to remove the leading and trailing spaces from the string. So if you pass the string “I’m funny " to trim (), you'll get back the string "I'm funny". o Your implementation of **trim**() should handle the following input as follows: o If you pass a string like “I’m funny ", the function returns "I'm funny". o If you pass a string like "I'm funny", the function returns "I'm funny" (that is, the original string). o If you pass the string "", the function returns "". o If you pass any other value (null, 3, [1, 2], etc.), the function returns "". Create a file in your /**AdvJS** folder named trim.html that contains bare-bones HTML and a script with the function, and test cases that show your function works on the test cases described above. You can use **console.log ()** to display the output. You might want to add something like brackets so we know where the string starts and stops:
9	In this project, you will write a function, objectContainsOnlyStrings that takes one object, and determines if that object contains only properties that have string values. If so, then the function returns true; otherwise it returns false. Save the file in your **/AdvJS** folder as **onlyStrings.html**. Write the code for the **objectContainsOnlyStrings()** function, and test using the given test examples (and feel free to add more objects and tests if you like). When the project is working to your satisfaction, hand in **onlyStrings.html**.
10	In this project, you'll write a function, **equals ()**, to compare two objects and determine if they are equal. This will be a "simple" version of equality, meaning that we'll assume that both objects contain only properties with primitive values (numbers, strings, booleans, null, or undefined). The **equals ()** function will take two arguments, the two objects, and compare each property in each object to see if they are the same. Make sure you handle the following cases: • If you pass **null** or **undefined** for one or both of the objects, the result is false. • If any of the properties of either object is an object or a method, the result is false. • If the objects have the same property names but different property values, the result is false. • If the objects have the same property values, but different property names, the result is false. Create a file in your **/AdvJS** folder, **objectsEqual1.html**, that contains bare-bones HTML and a script with the function, and test cases that show your function works on the test

cases described above. You can use console.log () to display the output. Here is the expected output using the test cases in the skeleton code below:



```
Compare book1, book2 with equals function:
The two books are the same
Compare book1, book3 with equals function:
The two books are different
empty objects are equal
null objects are equal
objects with different property names are not equal
objects with properties that are objects are not equal
objects with properties that are methods are not equal
```

Feel free to test other objects, and add test cases of your own.
Write a program that uses one constructor, **Book (),** to create book objects with the following properties: genre, title, author, published (the year it was published), and hasMovie (a boolean indicating if the book has been made into a movie or tv show), **display ()** (a method that displays the object in the console), and **toString()** (a method that returns a string describing the book).
Create a **window.onload** handler function that creates an array of book objects, and then adds each book to the page using the addToPage() function (below). Create a new file, books1.html, in your /AdvJSfolder and use the following HTML and JavaScript to get started:
Save the file in your **/AdvJS** folder as **books1.html**. When the project is working to your satisfaction, hand in **books1.html**
Feel free to add more types of books other than mystery and biography!
Something to think about as you work on this project: Which of the properties did you create using the **Book ()** constructor and which did you add to the **Book.prototype**? Think carefully about which properties belong where and why.

11 In this project, you'll create a function that generates functions to add to the web page below. Notice in the HTML, we have two container <div>s: "container1" and "container2." Both these <div> elements are the full page wide, and 300px tall. (Imagine you are building a game with two players, one in the top part of the screen and one in the bottom part):
Save the file in your **/AdvJS** folder as **gameDivs.html**.
First, create a function, **addDivTo()** that takes an id for a container, and returns a function. The function returned should take one argument, a string representing a CSS class. (**Hint:** don't name your parameter "class": this is a reserved word in JavaScript, so use another name!). The function will create a new <div> element and add it to the <div> with the id **id** that was passed into the call to **addDivTo()**.
When you call the function returned from **addDivTo()**, you'll pass in the name of the class you want the <div> to have. The function will create the <div>, give it the class, position the <div> randomly in the correct container <div>, append the new <div> to the container <div> and return the new <div>.
Then write an event handler function for the window load event to add several <div> elements to the page in the two containers. Feel free to create more classes for different color <div> elements if you want (or modify the CSS to make circles too if you like).
Comment your code liberally! When the project is working to your satisfaction, hand in **gameDivs.html**.
More hints:

	• Create a new element with **document.createElement("div")**. • Add a div to a container div using **appendChild()**. • Create a random position using Math.random() and Math.floor(): **newDiv.style.left = Math.floor(Math.random() * (div.offsetWidth - 175)) + "px";** **newDiv.style.top = Math.floor(Math.random() * (div.offsetHeight - 175)) + "px";** where **div** is a container <div>.
12	In this project, you'll build a very simple music game. We've begun the code for you below; you just need to finish it up (and perhaps, clean it up a bit too!). The game works like this: you click on note buttons to create a chord. Each time you click on a note, add the note to the "notesDisplay" (an input text area). After three notes, the code should check to see if the string you've entered is a valid major chord (we won't use sharps and flats for this project, so the only valid chords for our purposes are C, F, and G, though you could add more). After checking, it should display a message in the "message" <div> (replacing the existing text), and reset the form for the next chord. The goal for this project is to think about how to organize your code, and as you build it, to understand the scope of each function and variable you use. Comment your code to briefly describe your organization decisions, as well as the scope of the key variables and functions.