Department of MCA

Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Web Technologies
2.	Course Code	MCA-302
3.	Academic Year	2024-2025
4.	Semester	MCA-3rd
5.	Number of Lesson plans	36
6.	Faculty Assigned	Mr.Vishal Gupta

Faculty Signature

Lesson Plan No. 1	Course Name: Web Technologies Topic: Text formatting tags	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of text formatting tags. • Learn common text formatting tags and their uses. • Apply text formatting tags to create basic HTML documents
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Begin by asking students if they have ever seen a webpage. - Explain that web pages are created using a language called HTML. - Introduce the concept of tags and how they structure content on a webpage. - Briefly mention that text formatting tags are used to style text on a webpage. 2. **Development** (30 minutes) a. **Common Formatting Tags:** - Introduce the following tags with examples- - `<h1>` to `<h6>` for headings - `<p>` for paragraphs - `<b>` and `<strong>` for bold text - `<i>` and `<em>` for italic text<br>- `<br>` for line breaks b. **Tag Structure:** - Explain the basic structure of HTML tags, including opening and closing tags. c. **Practical Exercise:** - Provide a simple HTML code snippet for students to practice using the learned tags. 3. Exercise (5 minutes) – - What tag is used to define a paragraph? - How do you make text bold in HTML? - What tag is used to insert a line break?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Video of Formatting Tags in HTML https://youtu.be/4ZunanytE88 3. Homework - Create your video log highlighting cloud computing concepts and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1a	Course Name: Web Technologies Topic: Introduction to HTML	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the basics of HTML and its role in creating web pages. Learn the structure of an HTML document. Recognize and use common HTML elements.
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Explain what HTML is and its purpose in building websites. - Briefly discuss the history of HTML. - Provide examples of websites and how they are constructed using HTML. 2. **Development** (30 minutes) a. **Common Formatting Tags:** - Introduce the following tags with examples- - `<h1>` to `<h6>` for headings - `<p>` for paragraphs - `<b>` and `<strong>` for bold text - `<i>` and `<em>` for italic text<br>- `<br>` for line breaks b. **Tag Structure:** - Explain the basic structure of HTML tags, including opening and closing tags. c. **Practical Exercise:** - Provide a simple HTML code snippet for students to practice using the learned tags. 1. Exercise (5 minutes) – - What tag is used to define a paragraph? - How do you make text bold in HTML? - What tag is used to insert a line break?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Video of Formatting Tags in HTML https://youtu.be/4ZunanytE88 3. Homework - Create your video log highlighting cloud computing concepts and submit on Google classroom. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.

	Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2	**Course Name: Web Technologies** **Topic:** Embedding objects, Paragraphs, Formatting, Links, Head, Images	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of embedding objects in a document. • Learn how to format text, paragraphs, and headers effectively. • Create and insert hyperlinks within a document. • Insert and manipulate images in a document.
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Begin by asking students if they have ever seen a document with pictures or links. - Explain that today's lesson will focus on how to create such documents. - Briefly introduce the concepts of embedding, formatting, and linking. 2. **Development** (30 minutes) a. **Embedding Objects:** - Explain what embedding is and its purpose. - Demonstrate how to embed different types of - Objects (e.g., spreadsheets, presentations) into a document. b. **Paragraphs and Formatting:** - Discuss the importance of proper paragraph formatting. - Demonstrate how to adjust font styles, sizes, colors, and alignment. - Explain the use of line spacing, indentation, and tabs. c. **Links and Headers:** - Introduce the concept of hyperlinks. - Show how to create links to websites, email addresses, and other documents. - Explain the purpose of headers and how to create them 3. **Exercise** (5 minutes) – - What is the purpose of embedding an object in a document? - How do you create a hyperlink to a website? - What are the different ways to format a paragraph?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Introduction to HTML https://developer.mozilla.org/en-US/docs/Web/HTML Video of Embedding objects, Paragraphs, Formatting, Links, Head, Images https://youtu.be/Hnhh-aM0g2s

	3. Homework - Create a document with at least two embedded objects, formatted text, a hyperlink, a header, and an image. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 3	**Course Name: Web Technologies** **Topic:** Tables In HTML	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: a. Understand the basic structure of HTML tables. b. Learn how to create different types of tables. c. Apply table formatting to enhance table appearance..
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Begin by discussing the concept of tables in real life - Explain how HTML tables are used to structure data in a Web page. - Show examples of tables on different websites. 2. **Development** (30 minutes) **a. Basic table structure** - Explain the use of <table>, <tr>, and <td> tags. - Demonstrate how to create a simple table with rows and columns. **b. Table headers:** -Introduce the <th> tag for table headers. -Explain the importance of table headers for accessibility. **c. Table formatting:** -Discuss table attributes like border, cell spacing, cell padding, width, and align. -Demonstrate how to apply these attributes to style the table. **3. Exercise (5 minutes) –** - What are the three main tags used to create an HTML table? - How do you create a table header? - What is the purpose of the cell spacing attribute?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Introduction to Tables https://developer.mozilla.org/en-S/docs/Web/HTML/Element/table Video of Tables In HTML https://youtu.be/e62D-aayveY **3. Homework** -Create a simple HTML table to display personal information (name, age, city). -Experiment with different table formatting options. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	**Course Name: Web Technologies** **Topic : Lists in HTML**	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of lists in HTML. • Learn to create ordered and unordered lists. • Learn to create nested lists. • Apply list creation in basic HTML documents..
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Start by asking students about the importance of organizing information. Relate it to real-life examples like grocery lists, to-do lists, or outlines. - Introduce the concept of lists in HTML as a way to structure data effectively on web pages. 2. **Development** (30 minutes) **a. Ordered Lists:** - Explain the use of <ol> and <li> tags. - Demonstrate how to create a numbered list with different types of numbering (1, a, A, i, I). - Provide examples of ordered lists in real-world web pages. **b. Unordered Lists:** - Explain the use of <ul> and <li> tags. - Demonstrate how to create a bulleted list with different types of markers (disc, circle, square). - Provide examples of unordered lists in real-world web pages. **c. Nested Lists:** - Explain how to create lists within lists. - Demonstrate how to create hierarchical structures using nested lists. - Provide examples of nested lists in real-world web pages. 3. **Exercise** (5 minutes) – - What are the two main types of lists in HTML? - How do you create a numbered list in HTML? - What tag is used to define a list item?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Lists in HTML https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul 3. Homework - Create an HTML document containing an ordered list of your favorite movies, an unordered list of your hobbies, and a nested list of your favourite subjects and topics within them.

	Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 5	**Course Name: Web Technologies** **Topic :** Blocks, Layout, Colors, Color Values	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the basic principles of design related to blocks, layout, colors, and color values. • Learn how to use blocks to structure content. • Explore different layout options for effective visual hierarchy. • Understand the impact of colors and color values on design.
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Begin by discussing the importance of visual design in communication. - Show examples of well-designed and poorly designed websites or apps. - Explain how blocks, layout, colors, and color values contribute to overall design effectiveness. 2. **Development** (30 minutes) **a. Blocks:** - Introduce the concept of blocks as building blocks of design. - Explain how blocks can be used to group related content. - Demonstrate how to create different types of blocks (text, image, button, etc.). **b. Layout:** - Discuss the principles of balance, alignment, and proximity. - Explain different layout types (grid, fluid, responsive). - Demonstrate how to create effective layouts using blocks. **c. Colors and Color Values:** - Introduce the color wheel and basic color theory. - Explain color schemes (analogous, complementary, monochromatic, etc.). - Demonstrate how to use color values (hue, saturation, lightness) to create different effects. a. Exercise (5 minutes) – - What are the three primary colors? - What is the difference between a grid and a fluid layout? - How can you create a sense of balance in a design?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Lists in HTML https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul

	3. Homework - Create a simple digital design using basic shapes and colors. - Experiment with different layouts for a given set of content. - Explore different color schemes for a specific design project.. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 6	**Course Name: Web Technologies** **Topic :** Frames in HTML	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of frames in HTML. • Learn how to create basic HTML frames. • Explore the structure and attributes of frames.
Teaching Aids (if any)	a. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) - Begin by asking students if they have ever seen a webpage divided into multiple sections. - Explain that frames are used to divide a webpage into different sections, each displaying independent content. - Provide a real-world example, such as online newspapers or email clients, to illustrate the use of frames. 2. **Development** (30 minutes) **a. Frame Sets:** - Explain that a frame set defines the overall structure of a webpage using <frameset> and </frameset> tags. - Demonstrate how to create a simple frame set with two frames, using the rows or cols attribute. **b. Frame:** - Introduce the <frame> tag for defining individual frames within a frame set. - Explain the name, src, no resize, scrolling, and margin width attributes. **c. Inline Frames:** - Briefly mention the <iframe> tag for embedding external content within a webpage. - Explain the basic syntax and common use cases for iframes. **3. Exercise (5 minutes) –** - What is the purpose of the <frameset> tag? - Name two attributes of the <frame> tag. - What is the difference between a frame set and an inline frame?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Frames in HTML https://developer.mozilla.org/en-US/docs/Web/HTML/Element/framese Videos of Frames in HTML https://youtu.be/FIBocnLh2Yc

	3. **Homework** - Create a simple HTML page with two frames, one displaying a menu and the other displaying content. - Experiment with different frame attributes to customize the appearance and behavior of the frames. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 7	Course Name: Web Technologies Topic : Forms & Forms Controls	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the purpose and structure of HTML forms. • Learn about different types of form controls and their attributes. • Create basic HTML forms with various input elements.
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Begin by asking students about online forms they have encountered (e.g., registration, contact forms). - Explain that forms are essential for collecting user data and submitting it to a server. - Introduce the concept of form elements and their role in creating interactive forms. 2. **Development** (30 minutes) **a. Form Structure**: - Explain the <form> tag and its basic attributes (action, method). - Demonstrate how to create a simple form structure. **b. Form Controls:** - Introduce common input types: text, password, email, number, date, checkbox, radio, submit, reset. - Explain the label element for associating labels with form controls. **c. Creating a Basic Form:** - Guide students through creating a simple contact form with name, email, and message fields. - Discuss form validation (basic level) and the importance of user experience. 3. **Exercise (5 minutes)** – - What is the purpose of the <form> tag? - Name three different types of input elements. - What is the importance of the label element in a form?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Forms in HTML https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form Videos of Forms & Forms Controls in HTML https://youtu.be/oGO5NB5Cobs

	3. Homework - Create a simple registration form with required fields for name, email, and password. - Add basic validation to ensure that required fields are filled. - Experiment with different form control types and their attributes. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 8	**Course Name: Web Technologies** **Topic :** CSS, Defining Styles, Elements of Style	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the purpose of CSS and its relationship to HTML. • Learn the basic structure of a CSS rule set. • Identify common CSS properties and their effects.
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Begin by asking students to describe the appearance of their favorite website. - Explain that while HTML structures content, CSS controls the presentation and style. - Introduce CSS as a language for styling web pages. 2. **Development** (30 minutes) **a. CSS Basics**: - Explain the concept of a CSS rule set, consisting of a selector and a declaration block. - Demonstrate how to link a CSS file to an HTML document. **b. Selectors:** - Introduce different types of selectors: element, class, and ID selectors. - Provide examples of how to target specific elements using these selectors. **c. Properties and Values:** - Explain the concept of CSS properties and values. - Discuss common properties like color, font-size, font-family, text-align, and background-color. - Demonstrate how to apply these properties to elements using CSS rules. 3. **Exercise (5 minutes)** – - What is the purpose of CSS? - What are the two main components of a CSS rule set? - Give an example of a CSS property and its corresponding value.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading CSS Properties https://developer.mozilla.org/en-US/docs/Web/CSS Videos of CSS, Defining Styles, Elements of Style https://youtu.be/t8OH5eWlL9g

	3. **Homework** - Create a simple HTML document and a corresponding CSS file. - Style the HTML elements using CSS properties to change the appearance of the page. - Experiment with different selectors and properties to explore CSS capabilities. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 9	**Course Name: Web Technologies** **Topic :** Linking a Style Sheet to an HTML Document In-line Styles, External Style Sheets, Internal Style Sheets	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand different ways to apply CSS styles to HTML documents. • Learn the advantages and disadvantages of each style method. • Implement in-line, internal, and external styles effectively.
Teaching Aids (if any)	a. Power Point Presentation
Teaching Development	1. **Introduction** (5 minutes) - Briefly recap the concept of CSS and its role in styling web pages. - Introduce the different methods for linking CSS to HTML documents: in-line, internal, and external styles. - Explain that the choice of method depends on the specific project requirements. 2. **Development** (30 minutes) **a. In-line Styles**: - Demonstrate how to apply styles directly to HTML elements using the style attribute. - Discuss the limitations of in-line styles for maintainability and reusability. **b. Internal Style Sheets:** - Explain how to create a style section within the <head> section of an HTML document using the <style> tag. - Discuss the advantages of internal styles for styling a single page. **c. External Style Sheets:** - Introduce the concept of linking an external CSS file to an HTML document using the <link> tag. - Emphasize the benefits of external styles for maintainability, reusability, and efficiency. 3. **Exercise (5 minutes)** – - What is the difference between in-line and internal styles? - How do you link an external CSS file to an HTML document? - Which style method is generally preferred for large-scale projects?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading

	Linking a Style Sheet to an HTML Document https://www.w3schools.com/css/css_howto.asp Videos of : Linking a Style Sheet to an HTML Document https://youtu.be/3Fd11f00zOs 3. **Homework** - Create an HTML document and apply styles using all three methods: in-line, internal, and external. - Compare the code structure and efficiency of each method. Experiment with different CSS selectors and properties to style the content. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 10	**Course Name: Web Technologies** **Topic :** Introduction to XML	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the basic concept of XML and its purpose. • Learn the fundamental structure of an XML document. • Identify the key components of an XML document.
Teaching Aids (if any)	a. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) - Begin by asking students about the different formats of data they encounter daily (e.g., text, images, spread sheets). - Introduce XML as a language for storing and transporting data. - Explain that XML is platform-independent and human-readable. 2. **Development** (30 minutes) **a. What is XML:** - Define XML as a markup language for encoding data. - Explain the difference between XML and HTML. - Discuss the advantages of using XML for data exchange. **b. XML Structure:** - Introduce the basic structure of an XML document, including the declaration, root element, elements, attributes, and text content. - Provide simple examples to illustrate the structure. **c. Well-Formed XML:** - Explain the concept of a well-formed XML document. - Discuss the essential rules for creating valid XML documents. 3. **Exercise (5 minutes)** – - What does XML stand for? - What is the root element in an XML document? - What is the difference between an element and an attribute in XML?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Introduction to XML https://developer.mozilla.org/en-US/docs/Web/XML Videos of : Introduction to XML https://youtu.be/QmASaij0cu8 **3. Homework** - Create a simple XML document to represent a list of books

	with title, author, and publication year. - Validate the XML document using an online XML validator. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 10	**Course Name: Web Technologies** **Topic :** Introduction to JavaScript	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the role of JavaScript in web development. • Learn basic JavaScript syntax and structure. • Write simple JavaScript code to manipulate the Document Object Model (DOM).
Teaching Aids (if any)	a. Videos of : Introduction to JavaScript https://youtu.be/VlPiVmYuoqw b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Ask students about their experiences with interactive web pages (e.g., games, animations, dynamic content). • Explain that JavaScript is the programming language that brings these elements to life. • Briefly highlight the importance of JavaScript in modern web development. 2. **Development** (30 minutes) **a. What is JavaScript:** • Define JavaScript as a scripting language for creating interactive web pages. • Explain its relationship to HTML and CSS. **b. Basic Syntax:** • Introduce JavaScript's syntax, including variables, data types (numbers, strings, booleans), operators. • Write simple JavaScript code examples to demonstrate these concepts. **c. DOM Manipulation:** • Explain the Document Object Model (DOM) as a tree-like representation of an HTML document. • Demonstrate how to access and modify HTML elements using JavaScript. • Provide a basic example of changing the content of an element. 3. **Exercise (5 minutes)** – • What is JavaScript used for in web development? • What is a variable in JavaScript? • How can you display a message in the browser's console using JavaScript?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Introduction to JavaScript

	https://developer.mozilla.org/en-US/docs/Web/JavaScript 3. **Homework** - Create a simple HTML file with a few elements. - Write a JavaScript file to manipulate those elements using DOM manipulation - Try changing the text content, background color, or other properties of the elements. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 12	**Course Name: Web Technologies** **Topic :** Variables in JavaScript	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of variables in JavaScript. • Learn how to declare and initialize variables. • Differentiate between different data types in JavaScript.
Teaching Aids (if any)	a. Videos of :Variables in JavaScript https://youtu.be/VlPiVmYuoqw b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by asking students about the concept of storage containers in real life (e.g., boxes, jars). • Explain that variables in programming serve a similar purpose: storing data for later use. • Introduce JavaScript as a programming language where variables are essential. 2. **Development** (30 minutes) **a. Variable Declaration:** • Explain the use of the var, let, and const keywords to declare variables. • Discuss the differences between these keywords in terms of scope and reassignment. **b. Data Types:** • Introduce basic JavaScript data types: numbers, strings, booleans, null, and undefined. • Provide examples of each data type and how to assign them to variables. **c. Variable Assignment:** • Demonstrate how to assign values to variables using the assignment operator (=). • Explain the concept of dynamic typing in JavaScript. 3. **Exercise (5 minutes)** – • What is a variable in JavaScript? • Name three different ways to declare a variable in JavaScript? • What is the data type of the value "Hello, World!"?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Variables in JavaScript https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/vart

	3. **Homework** • Create a JavaScript program that declares variables of different data types and assigns values to them. • Display the values of the variables in the console using the console.log() method. • Experiment with different variable names and data types. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 13	Course Name: Web Technologies **Topic :** Operators in JavaScript	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of operators in JavaScript. • Learn different types of operators: arithmetic, assignment, comparison, logical. • Apply operators to perform calculations, assignments, and comparisons in JavaScript code.
Teaching Aids (if any)	a. Videos of :Operstors in JavaScript https://youtu.be/VlPiVmYuoqw b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Start with a simple math problem and ask students how they would solve it using a calculator. • Introduce the concept of operators as symbols that perform operations on values. • Explain that JavaScript uses operators to manipulate data. 2. **Development** (30 minutes) **a. Arithmetic Operators:** • Introduce the basic arithmetic operators: +, -, *, /, %. • Provide examples of how to use them to perform calculations. **b. Assignment Operators:** • Explain the assignment operator (=) and its use in assigning values to variables. • Introduce shorthand assignment operators like +=, -=, *=, /=, %=. **c. Comparison Operators:** • Explain the comparison operators: ==, !=, >, <, >=, <=. • Demonstrate how to use them to compare values and make decisions. 3. **Exercise (5 minutes)** – • What is the result of 5 % 2? • How do you assign the value 10 to a variable named x? • What is the difference between == and ===?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Operators in JavaScript https://developer.mozilla.org/en-

	US/docs/Web/JavaScript/Reference/Operators 3. **Homework** - Write a JavaScript program to calculate the area and perimeter of a rectangle using arithmetic operators. - Create a program that checks if a number is even or odd using comparison operators. - Experiment with different operators and practice writing expressions. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 14	**Course Name: Web Technologies** **Topic :** String manipulation in JavaScript	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of strings in JavaScript. • Learn common string manipulation methods. • Apply string manipulation techniques to solve basic problems.
Teaching Aids (if any)	a. Videos of : String manipulation in JavaScript https://youtu.be/VlPiVmYuoqw b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by asking students about their favorite text-based games or puzzles. • Explain that strings are sequences of characters and are fundamental to many programming tasks. • Introduce the concept of string manipulation as the process of modifying and extracting information from strings. 2. **Development** (30 minutes) **a. Basic String Operations:** • Demonstrate how to create strings using single, double, and backticks. • Explain string concatenation using the + operator. • Introduce string length property and accessing individual characters. **b. String Methods:** • Cover essential methods like toUpperCase(), toLowerCase(), substring(), indexOf(), lastIndexOf(). • Provide examples of how to use these methods for common tasks. **c. String Immutability:** • Emphasize that strings are immutable in JavaScript. • Explain that string methods create new strings instead of modifying the original. 3. **Exercise (5 minutes)** – • How do you find the length of a string in JavaScript? • What does the toUpperCase() method do? • What is the difference between indexOf() and lastIndexOf()?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading **3.** Strings Manipulation JavaScript https://developer.mozilla.org/en-

	US/docs/Web/JavaScript/Reference/Global_Objects/String **Homework** • Write a JavaScript program to reverse a given string. • Create a program to count the number of vowels in a string. • Experiment with different string methods to practice manipulation. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 15	**Course Name: Web Technologies** **Topic :** Mathematical Functions	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of inbuilt mathematical functions in JavaScript.. • Learn how to use various mathematical functions to perform calculations. • Apply mathematical functions to solve simple problems.
Teaching Aids (if any)	a. Videos of : String manipulation in JavaScript https://youtu.be/H1u09emZQts b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by asking students if they have encountered any situations where they needed to perform calculations in their daily lives. • Introduce the concept of JavaScript as a programming language that can handle mathematical operations. • Explain that JavaScript provides built-in functions to simplify these calculations. 2. **Development** (30 minutes) **a. Math Object:** • Explain that JavaScript has a built-in Math object that contains various mathematical properties and methods. • Demonstrate how to access the Math object using Math.property or Math.method() **b. Common Mathematical Functions:** • Introduce essential mathematical functions like Math.abs(), Math.ceil(), Math.floor(), Math.round(), Math.random(), Math.pow(), Math.sqrt(). • Provide examples of how to use each function with clear explanations. **c. Practical Application:** • Show how to use these functions to solve simple problems, such as calculating the area of a circle, generating random numbers, or finding the square root of a number. 3. **Exercise (5 minutes)** – • What is the output of Math.ceil(4.2)? • How do you generate a random number between 0 and 1 in JavaScript? • What function is used to find the absolute value of a

	number?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Mathematical Functions in JavaScript https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math **Homework** • Create a JavaScript program to calculate the area and circumference of a circle. • Write a program to generate a random number between a specified range. • Explore other mathematical functions provided by the Math object and experiment with their usage. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 16	Course Name: Web Technologies Topic : Statements in JavaScript	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of statements in JavaScript. • Learn different types of statements and their syntax. • Write basic JavaScript programs using statements.
Teaching Aids (if any)	a. Videos of : Statements in JavaScript https://youtu.be/uGlCPo42msE b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by explaining that a computer program is a sequence of instructions. • Introduce the concept of statements as the building blocks of JavaScript programs. • Use simple analogies to illustrate the idea of statements, such as comparing them to sentences in a language. 2. **Development** (30 minutes) **a. Declaration Statements**: • Explain how to declare variables using var, let, and const keywords. • Differentiate between the three keywords based on scope and reassignment. **b. Expression Statements:** • Define expressions as combinations of values, variables, operators, and function calls that produce a result. • Explain how expressions are used in assignment statements and as part of other statements. **3. Control Flow Statements:** • Introduce the concept of control flow, which determines the order in which statements are executed. • Briefly explain conditional statements (if, else, else if) and looping statements (for, while) without going into depth. These will be covered in more detail in subsequent lessons. **Exercise (5 minutes)** – • What are the three ways to declare a variable in JavaScript? • What is the difference between `let` and `const`? • Give an example of an expression statement.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation

	from students on these. 2. Suggested Reading Statements in JavaScript https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements **Homework** - Write a JavaScript program to declare variables for your name, age, and city and print them to the console. - Experiment with different types of expressions and print their results to the console. - Research the difference between block-scoped and function-scoped variables.. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 17	**Course Name: Web Technologies** **Topic :** Arrays in JavaScript	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of arrays in JavaScript. • Learn how to create, access, and modify array elements. • Explore basic array methods and their usage.
Teaching Aids (if any)	a. Videos of : Arrays in JavaScript https://youtu.be/TScHu-37Cqw b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by asking students to think about how they would store a list of items (e.g., shopping list, to-do list). • Introduce arrays as a data structure for storing multiple values in a single variable. • Explain the benefits of using arrays to organize and manipulate data. 2. **Development** (30 minutes) **a. Array Creation and Access**: • Demonstrate how to create an array using square brackets `[]`. • Explain how to access array elements using their index (starting from 0). • Show how to modify array elements by assigning new values to specific indices. **b. Array Length:** • Introduce the length property to determine the number of elements in an array. • Demonstrate how to use the length property to access the last element of an array. **C. Basic Array Methods:** • Introduce common array methods like push(), pop(), shift(), unshift(). • Explain the purpose of each method and provide examples of their usage. **Exercise (5 minutes)** – • How do you access the third element of an array named myArray? • What is the output of myArray.length if myArray has 5 elements? • What does the push() method do to an array?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation

	from students on these. 2. Suggested Reading Arrays in JavaScript https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array **Homework** • Create an array of your favorite movies and print each movie to the console. • Create an array of numbers and calculate the sum of all elements. • Experiment with different array methods to manipulate data. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 18	Course Name: Web Technologies Topic : Functions in JavaScript	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of arrays in JavaScript. • Learn how to create, access, and modify array elements. • Explore basic array methods and their usage.
Teaching Aids (if any)	a. Videos of : Functions in JavaScript https://youtu.be/jwOW-qlJtxg b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by asking students about repetitive tasks they perform in their daily lives. • Introduce the concept of functions as a way to automate repetitive tasks in programming. • Explain the benefits of using functions for code organization and reusability. 2. **Development** (30 minutes) **a. Function Definition:** • Explain the syntax for defining a function using the `function` keyword. • Demonstrate how to create functions with and without parameters. • Discuss the concept of function body and return statement. **b. Function Calling:** • Explain how to call a function to execute its code. • Demonstrate how to pass arguments to functions with parameters. • Show how to use the returned value from a function. **C. Function Scope:** • Briefly introduce the concept of local and global variables within functions. • Explain how variables declared inside a function are accessible only within that function.. **Exercise (5 minutes)** – • What is the correct syntax for defining a function in JavaScript? • How do you call a function in JavaScript? • What is the purpose of the `return` statement in a function?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.

	2. Suggested Reading Functions in JavaScript https://developer.mozilla.org/en- https://www.w3schools.com/js/js_functions.asp **Homework** • Create a function to calculate the factorial of a number. • Write a function to check if a number is even or odd. • Create a function to convert temperature from Celsius to Fahrenheit.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 19	**Course Name: Web Technologies** **Topic :** Data and Objects, Regular Expressions	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of data types and objects in JavaScript. • Learn how to create and manipulate objects. • Grasp the basics of regular expressions for pattern matching..
Teaching Aids (if any)	a. Videos of : Data and Objects, Regular Expressions https://youtu.be/rhzKDrUiJVk b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by discussing the importance of organizing data in programming. • Introduce the concept of data types and objects as a way to structure information. • Briefly explain the need for pattern matching in text data and how regular expressions can be used. 1. **Development** (30 minutes) **a. Data Types and Objects:** • Explain the difference between primitive data types (numbers, strings, booleans) and objects. • Introduce object literals and their syntax. • Demonstrate how to access and modify object properties. **a. Regular Expressions:** • Define regular expressions as patterns used to match text. • Explain basic regular expression syntax (e.g., special characters, quantifiers). • Provide simple examples of matching patterns in strings. **Exercise (5 minutes)** – • What is the difference between a primitive data type and an object? • How do you access the "name" property of an object named "person"? • What does the regular expression `/\d+/` match?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Data and Objects, Regular Expressions https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions

	Homework - Create an object representing a person with properties for name, age, and city. - Write a JavaScript program to validate an email address using a regular expression. - Explore different regular expression patterns and their uses.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 20	**Course Name: Web Technologies** **Topic :** Built-in Objects, Events, Rollover, Buttons, Moving Images	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of built-in objects in JavaScript. • Learn about different types of events and how to handle them. • Implement rollover effects, create interactive buttons, and animate images.
Teaching Aids (if any)	a. Videos of : Built-in Objects, Events, Rollover, Buttons, Moving Images https://youtu.be/VlPiVmYuoqw b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by discussing the interactive elements on websites and how they enhance user experience. • Introduce the concept of JavaScript as a language that brings these elements to life. • Briefly explain the role of built-in objects in creating dynamic web pages. 2. **Development** (30 minutes) **a. Built-in Objects:** • Introduce core built-in objects like document, window, and event. • Explain their roles in interacting with web page elements. **b. Events and Event Handling:** • Discuss various event types (click, mouseover, mouseout, etc.) and how to attach event listeners to HTML elements. • Demonstrate how to create simple event handlers using JavaScript. **c. Rollover Effects, Buttons, and Image Animation** • Show how to create rollover effects on images using JavaScript and CSS. • Explain the concept of buttons and how to style and add interactivity using JavaScript. • Introduce basic image animation techniques using JavaScript (e.g., changing image source, CSS transitions). • **Exercise (5 minutes)** – • What is the difference between a mouseover and a mouseout event?

	- How do you attach a click event listener to a button? - Name two built-in JavaScript objects. Named "person"?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Built-in Objects, Events, Rollover, Buttons, Moving Images https://developer.mozilla.org/en-US/docs/Web/API/Event 3. Homework - Create a simple HTML page with an image that changes its source on hover. - Build a button that displays an alert when clicked. - Experiment with different event types and explore more complex image animation techniques.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 21	**Course Name: Web Technologies** **Topic :** Servlet Life Cycle, Servlet Request and Response Disadvantages of Servlets	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the lifecycle of a servlet. • Learn how to handle HTTP requests and responses using Servlets. • Identify the limitations of servlets in web application development.
Teaching Aids (if any)	a. Videos of : Servlet Life Cycle, Servlet Request and Response Disadvantages of Servlets https://youtu.be/7TOmdDJc14s b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly introduce servlets as Java programs that run on a web server to handle HTTP requests. • Explain that servlets are essential for building dynamic web applications. • Create interest by discussing how servlets power many websites and applications we use daily. 2. **Development** (30 minutes) a. **Servlet Lifecycle:** • Explain the three phases of a servlet's lifecycle: init, service, and destroy. • Describe the purpose of each phase and when they are called. • Provide a simple example to illustrate the lifecycle. b. **Servlet Request and Response:** • Introduce the ServletRequest and ServletResponse interfaces. • Explain how to obtain information from the request (parameters, headers, etc.) and how to send a response (content, status codes, headers). • Demonstrate how to handle different HTTP methods (GET, POST, etc.) using the service method. c. **Disadvantages of Servlets**: • Discuss the challenges and limitations of using servlets: • Complexity in managing state (session management) • Potential performance issues with large-scale applications • Tight coupling with the servlet API

	Exercise (5 minutes) – • What are the three phases of a servlet's lifecycle? • How do you get a request parameter from a ServletRequest object? • What are two main disadvantages of using servlets?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Servlet Life Cycle, Servlet Request and Response Disadvantages of Servlets https://docs.oracle.com/javaee/7/tutorial/servlets.htm Homework • Create a simple servlet that displays a welcome message based on the user's name (passed as a request parameter). • Research different session management techniques used with servlets. • Explore alternatives to servlets for building web applications (e.g., frameworks like Spring MVC)..
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 22	**Course Name: Web Technologies** **Topic :** Reading Headers JSP: Scripting Elements, JSP Expression	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of reading headers in JSP. • Learn about scripting elements and JSP expressions. • Apply the knowledge to read and process header information in a JSP page.
Teaching Aids (if any)	a. Videos of : Reading Headers JSP: Scripting Elements, JSP Expression https://youtu.be/xSNndzSjmQw b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly explain the importance of headers in web applications. • Introduce the concept of JSP as a server-side technology for creating dynamic web pages. • Highlight the role of scripting elements and JSP expressions in accessing and manipulating data. 1. **Development** (30 minutes) **a. Scripting elements:** • Explain the syntax and purpose of scriptlets, declarations, and expressions. • Demonstrate how to use scriptlets to write Java code within a JSP page. • Illustrate the use of declarations to define variables and methods. **b. JSP expressions**: • Describe the syntax and usage of JSP expressions. • Explain how to evaluate and print expressions within a JSP page. c. **Reading headers**: • Introduce the HttpServletRequest object and its methods for accessing headers. • Demonstrate how to obtain header values using JSP expressions and scripting elements. • Provide examples of common header names (e.g., User-Agent, Referer, Cookie). **Exercise (5 minutes)** – • What are the three types of scripting elements in JSP? • How do you access the value of the "User-Agent" header in a JSP page? • What is the difference between a scriptlet and a JSP

	expression?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Reading Headers JSP: Scripting Elements, JSP Expression https://docs.oracle.com/javaee/7/tutorial/servlets.htm Homework • Create a JSP page that displays the values of the following headers: User-Agent, Referrer, and Host. • Experiment with different header names and their corresponding values. • Research other methods of accessing and processing headers in JSP
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 23	**Course Name: Web Technologies** **Topic :** JSP Declaration, Predefined variables/objects, using user defined functions	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of JSP declarations and their use. • Learn about predefined variables and objects available in JSP. • Understand the creation and usage of user-defined functions in JSP.
Teaching Aids (if any)	a. Videos of : JSP Declaration, Predefined variables/objects, using user defined functions https://youtu.be/OuBUUkQfBYM b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly recap the concept of JSP as a server-side technology. • Introduce the idea of structuring code within JSP pages for better organization and reusability. • Highlight the importance of predefined variables and objects for accessing implicit information. 1. **Development** (30 minutes) **a. JSP Declarations:** • Explain the syntax and purpose of declarations. • Demonstrate how to declare variables and methods within a JSP page. • Discuss the scope of declared elements. **b. Predefined Variables/Objects:** • Introduce the concept of implicit objects. • Explain the use of common implicit objects like request, response, session, application, out, config, page, and exception. • Provide examples of how to use these objects in JSP code. c. **User-Defined Functions**: • Explain the concept of creating custom functions in JSP. • Demonstrate how to define and use methods within a JSP declaration. • Discuss the benefits of using user-defined functions for code modularity and reusability. **Exercise (5 minutes)** – • What is the difference between a JSP declaration and a scriptlet?

	• Name two predefined objects available in JSP and their common uses. • How do you create a user-defined function in JSP?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading JSP Declaration, Predefined variables/objects, using user defined functions https://docs.oracle.com/cd/E19879-01/819-3669/bnall/index.html Homework • Create a JSP page that declares a variable, a method, and uses predefined objects like request and response. • Write a JSP page with a user-defined function to calculate factorial of a number. • Explore other predefined objects and their functionalities.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 24	Course Name: Web Technologies Topic working with Databases Using JSP, Inserting, Updating, and Deleting Database Records	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the integration of JSP with databases. • Learn to perform CRUD operations (Create, Read, Update, Delete) using JSP and JDBC. • Implement data persistence in web applications.
Teaching Aids (if any)	a. Videos of : working with Databases Using JSP, Inserting, Updating, and Deleting Database Records https://youtu.be/7v2OnUti2eM b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly explain the importance of databases in web applications. • Introduce JDBC as the Java API for database connectivity. • Highlight the role of JSP in creating dynamic web pages that interact with databases. 1. **Development** (30 minutes) **a. Database Connectivity:** • Explain the steps involved in establishing a database connection using JDBC. • Demonstrate how to load the JDBC driver and create a connection object. **b. CRUD Operations:** • Discuss the SQL statements for inserting, updating, and deleting records. • Show how to execute SQL statements using PreparedStatement. • Demonstrate how to handle exceptions and error handling **c. JSP and Database Integration:** • Explain how to use JSP to retrieve data from a database and display it on a web page. • Demonstrate how to process user input and update the database accordingly.. **Exercise (5 minutes)** – • What are the four basic operations performed on a database? • What is the role of JDBC in JSP database connectivity?

	• How do you handle exceptions when working with databases in JSP?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading working with Databases Using JSP, Inserting, Updating, and Deleting Database Records https://www.youtube.com/playlist?list=PLJ-lrQx0LAbzPd8TtiR8ap2OWMp6doDO9 Homework • Create a JSP page to insert, update, and delete records in a database table of your choice. • Implement error handling and validation for user input. • Explore using JSTL for database operations.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 25	**Course Name: Web Technologies** **Topic** ASP. NET Standard Controls	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of standard controls in ASP.NET. • Learn about the common types of standard controls and their properties. • Implement basic user interfaces using standard controls.
Teaching Aids (if any)	a. Videos of: ASP. NET Standard Controls https://youtu.be/ildv4ohAUeM b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly introduce ASP.NET as a web development framework. • Explain the role of user interface elements in web applications. • Introduce the concept of standard controls as pre-built UI components. 1. **Development** (30 minutes) **a. Types of standard controls:** • Discuss the main categories of standard controls (e.g., Label, TextBox, Button, Image, HyperLink, Panel, etc.). • Provide examples of each type and their common properties. **b. Control properties and events:** • Explain how to set and access control properties. • Demonstrate how to handle control events (e.g., Button click, TextBox text changed). **c. Creating basic user interfaces** • Build simple web forms using a combination of standard controls. • Demonstrate how to arrange controls on a web page using layout controls (e.g., Table, Panel). **Exercise** (**5 minutes**) – • What is the difference between a Label and a TextBox control? • How do you handle a button click event in ASP.NET? • Name three common layout controls in ASP.NET.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading ASP. NET Standard Controls

	https://learn.microsoft.com/en-us/teamblog/asp-net-on-docs Homework • Create a simple registration form using standard controls (TextBox, Label, Button, etc.). • Implement basic validation for the form fields. • Explore additional standard controls and their properties..
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 26	**Course Name: Web Technologies** **Topic** Regular expressions	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of standard controls in ASP.NET. • Learn about the common types of standard controls and their properties. • Implement basic user interfaces using standard controls.
Teaching Aids (if any)	a. Videos of: Regular expressions https://youtu.be/8HfjMr-VQzg b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Understand the concept of regular expressions and their use in pattern matching. • Learn how to use regular expressions in ASP.NET to validate and manipulate text. • Apply regular expressions to real-world scenarios in web development. 1. **Development** (30 minutes) **a. Regular Expression Syntax:** • Introduce basic regular expression meta characters and their meanings (e.g., ., *, +, ?, ^, $). • Explain quantifiers and character classes for pattern matching. • Demonstrate how to create simple regular expressions for common patterns (e.g., email, phone number). **b. Regular Expressions in ASP.NET:** • Introduce the Regex class and its methods for working with regular expressions. • Demonstrate how to use Regex.IsMatch to check if a string matches a pattern. • Explain how to extract matched groups using Regex.Match and Regex.Matches. **c. Practical Examples** • Provide examples of using regular expressions for input validation (e.g., email, password, date). • Demonstrate how to use regular expressions for data extraction and manipulation (e.g., parsing text, replacing patterns). **Exercise (5 minutes)** – • What is a metacharacter in regular expressions? • How do you check if a string matches a regular

	expression in ASP.NET? • Give an example of a regular expression to validate an email address.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Regular expressions https://vlasovstudio.com/regent/documentation/Microsoft-Word-Wildcards-as-Regular-Expressions.html Homework • Create a simple ASP.NET web page with input fields for email and phone number. • Validate the input using regular expressions and display appropriate error messages. • Experiment with different regular expression patterns to match various text formats.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 27	Course Name: Web Technologies Topic Creating Server pages, page Life Cycle	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of server-side scripting and ASP.NET web pages. • Learn the stages involved in the ASP.NET page life cycle. • Create a basic ASP.NET web page using Visual Studio or a similar IDE.
Teaching Aids (if any)	a. Videos of: Creating Server pages, page Life Cycle https://youtu.be/y60P5yPlAcM b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Explain the difference between static and dynamic websites. • Introduce ASP.NET as a framework for creating dynamic web pages. • Highlight the benefits of using server-side scripting to process user interactions. 2. **Development** (30 minutes) **a. Creating Server Pages:** • Demonstrate how to create a new ASP.NET web page project in Visual Studio (or alternative IDE). • Explain the basic structure of an ASP.NET page (HTML markup with code-behind files). • Introduce common controls like buttons, textboxes, and labels. **b. ASP.NET Page Life Cycle:** • Describe the different stages of the ASP.NET page life cycle (Page Request, Initialization, Load, PreRender, Render, Unload). • Explain the purpose of each stage and its significance in page processing. • Illustrate how events are triggered at different points in the life cycle. **c. Simple Code Examples** • Develop a basic ASP.NET page with a button click event handler. • Use server-side code to process button click and display a message on the page. • Explain how the page life cycle facilitates data handling and interaction.

	Exercise (5 minutes) – • What is the main difference between a static HTML page and an ASP.NET page? • What are the two main components of an ASP.NET web page? • Name three stages of the ASP.NET page life cycle involved in processing user input..
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Creating Server pages, page Life Cycle https://learn.microsoft.com/en-s/aspnet/core/?view=aspnetcore-8.0 Homework • Create a simple ASP.NET page that displays a welcome message and changes the message based on the user's name entered in a textbox. • Implement the logic within the appropriate event handler based on the page life cycle. • Explore additional controls and functionalities available in ASP.NET.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 28	**Course Name: Web Technologies** **Topic** HTTP Request Object, HTTP Response Object	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of HTTP requests and responses. • Learn about the structure and components of HTTP request and response objects. • Explore how to access and manipulate request and response data in JSP.
Teaching Aids (if any)	a. Videos of: HTTP Request Object, HTTP Response Object https://youtu.be/kMsHak9gKKY b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly explain the client-server model of web applications. • Introduce the concept of HTTP as the underlying protocol for web communication. • Highlight the role of request and response objects in the HTTP process. 2. **Development** (30 minutes) **a. HTTP Request Object:** • Describe the structure of an HTTP request, including headers, methods, parameters, and body. • Explain how to access request parameters, headers, and other information using JSP. • Demonstrate how to retrieve data from request objects in different scenarios (e.g., form submission, query string parameters). **b. HTTP Response Object:** • Explain the structure of an HTTP response, including status codes, headers, and body. • Demonstrate how to set response headers, status codes, and content type in JSP. • Show how to send different types of responses (e.g., HTML, JSON, text). c. Relationship between Request and Response • Explain the interaction between request and response objects in a web application. • Demonstrate how to process request data and generate an appropriate response. **Exercise (5 minutes)** – • What is the difference between an HTTP GET and

	POST request? - How do you set the content type of a response to JSON in JSP? - What is the purpose of response headers?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading HTTP Request Object, HTTP Response Object https://docs.oracle.com/cd/B14504_01/dl/web/B10321_01/overview.htm Homework - Create a JSP page that displays the request method, request URI, and all request headers. - Create a JSP page that accepts user input through a form and displays the submitted data in the response. - Experiment with different response status codes and content types.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 29	Course Name: Web Technologies Topic Post back, Tracing & Debugging ASP. NET page	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of postback in ASP.NET. • Learn how to trace execution flow in an ASP.NET page. • Utilize debugging tools to identify and resolve issues in ASP.NET applications.
Teaching Aids (if any)	a. Videos of: Post back, Tracing & Debugging ASP. NET page https://youtu.be/Xorwq3OzPrY b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly explain the concept of web forms and their lifecycle. • Introduce postback as a fundamental mechanism for user interaction. • Highlight the importance of tracing and debugging for effective development. 2. **Development** (30 minutes) **a. Postback:** • Explain how postback occurs when a form is submitted. • Discuss the role of view state in preserving page data during postback. • Demonstrate how to handle postback events in code-behind **b. Tracing:** • Introduce the concept of tracing for monitoring application behavior. • Explain how to enable tracing in an ASP.NET page. • Demonstrate how to use trace information to analyze page execution. c. **Debugging** • Overview of debugging tools available in Visual Studio. • Explain the use of breakpoints, step-into, step-over, and step-out. • Demonstrate how to debug ASP.NET pages to find and fix errors. **Exercise (5 minutes)** –

	• What is a postback in ASP.NET? • How can you enable tracing in an ASP.NET page? • What is the purpose of breakpoints in debugging?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Post back, Tracing & Debugging ASP. NET page https://youtu.be/KVlXccl-XBA Homework • Create a simple ASP.NET web form with a button and a label. • Implement postback logic to change the label text when the button is clicked. • Enable tracing and examine the trace output. • Introduce a deliberate error in the code and use debugging to find and fix it.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 30	Course Name: Web Technologies **Topic** State Management and Types of State Management	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of state management in web applications. • Learn about different types of state management techniques. • Identify appropriate state management strategies for different use cases.
Teaching Aids (if any)	a. Videos of: State Management and Types of State Management https://youtu.be/Y27gU5izvdU b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Explain the concept of state in web applications and its importance. • Discuss the challenges associated with managing state. • Introduce the need for different state management techniques 2. **Development** (30 minutes) **a. Client-side state management:** • Explain how data is stored on the client-side (browser). • Discuss techniques like local storage, session storage, and cookies. • Highlight advantages and disadvantages of each method. **b. Server-side state management** • Explain how data is stored on the server-side. • Discuss techniques like session management, hidden fields, and database storage. • Highlight advantages and disadvantages of each method. **c. URL-based state management** • Explain how data is passed through the URL. • Discuss query parameters and route parameters. • Highlight advantages and disadvantages of this approach. **Exercise (5 minutes)** – • What is the difference between client-side and server-side state management? • Name two methods of client-side state management.

	- What is the main advantage of URL-based state management?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading State Management and Types of State Management https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API Homework - Research a specific state management technique in detail. - Create a simple example demonstrating how to use the chosen technique. - Analyze a web application and identify the state management strategies used.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 31	**Course Name: Web Technologies** **Topic** HTTP Session, HTTP Application, Query String Method	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concepts of HTTP Session, HTTP Application, and Query String in ASP.NET. • Learn how to manage state using sessions and application objects. • Apply query strings to pass data between web pages.
Teaching Aids (if any)	a. Videos of: HTTP Session, HTTP Application, Query String Method https://youtu.be/r0Wz1zR7hyc b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly explain the stateless nature of HTTP and the challenges it poses for web applications. • Introduce the concept of ASP.NET as a framework for building dynamic web pages. • Highlight the importance of managing state to maintain user information and application data. 2. **Development** (30 minutes) **a. HTTP Session:** • Explain the concept of a session and its purpose in ASP.NET. • Demonstrate how to create, access, and manage session variables using ASP.NET objects. • Discuss session expiration and methods for handling session timeouts. **b. HTTP Application** • Introduce the concept of application state and its use for sharing data across all users. • Explain how to access and modify application variables using the Application object. • Discuss scenarios where application state is beneficial (e.g., storing website configuration). **c. Query String** • Describe the concept of a query string and its role in passing data between web pages. • Demonstrate how to construct and access query strings in ASP.NET using URL parameters. • Explain techniques for handling and validating data received through query strings.

	Exercise (5 minutes) – • What is the difference between HTTP Session and HTTP Application state in ASP.NET? • How do you access the value of a session variable named "UserName" in an ASP.NET page? • Explain how query strings are used to pass information from one ASP.NET page to another.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading HTTP Session, HTTP Application, Query String Method https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.http.querystring?view=aspnetcore-8.0 Homework • Create an ASP.NET web application that demonstrates the use of session state. • Store user information (e.g., name, preferences) in session variables. • Develop another page that retrieves the stored information and displays it to the user. • Create a second ASP.NET application that utilizes the application state object. • Store a common value (e.g., website title, current user count) accessible by all users.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 32	Course Name: Web Technologies Topic State Management using Post back URL	Course No.: MCA-302

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of state management in web applications. • Learn how to use postback URLs for maintaining state information. • Implement basic state management using postback URLs in a web application.
Teaching Aids (if any)	a. Videos of: State Management using Post back URL https://youtu.be/e44pOfL_Ly8 b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Explain the concept of state and its importance in web applications. • Discuss the challenges of maintaining state in a stateless HTTP protocol. • Introduce postback URLs as a simple method for state management. 2. **Development** (30 minutes) **a. Postback URL Basics:** • Explain the concept of a postback URL. • Describe how to construct a postback URL with query parameters. • Demonstrate how to retrieve values from the postback URL in a subsequent request. **b. State Management with Postback URLs:** • Discuss the limitations of postback URLs for complex state management. • Provide examples of simple state management scenarios suitable for postback URLs (e.g., maintaining user preferences, tracking page visits). • Explain how to encode and decode data for inclusion in postback URLs. **c. Security Considerations** • Emphasize the importance of security when using postback URLs. • Discuss potential vulnerabilities and mitigation techniques (e.g., input validation, encryption). **Exercise (5 minutes)** – • What is a postback URL? • How can you pass data to a subsequent request using a postback URL? • What are the limitations of using postback URLs for state management?

Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading State Management using Post back URL https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.http.querystring?view=aspnetcore-8.0 Homework • Create a simple web application that uses a postback URL to store a user's name and display it on subsequent page loads. • Experiment with passing different types of data (e.g., numbers, dates) in the postback URL. • Research other state management techniques (e.g., session, cookies) and compare them to postback URLs.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 33	**Course Name: Web Technologies** **Topic** View State. Validation Controls	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of View State and its role in web applications. • Learn about different types of validation controls available in ASP.NET. • Implement basic validation on web forms using validation controls.
Teaching Aids (if any)	a. Videos of: View State. Validation Controls https://youtu.be/Z-KJpzTrkms b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Briefly explain the concept of state management in web applications. • Highlight the importance of data validation for web forms. • Introduce the concept of View State as a state management technique. • **Development** (30 minutes) **a. View State:** • Explain what View State is and how it works. • Demonstrate how to enable and disable View State. • Discuss the advantages and disadvantages of using View State. **b. Validation Controls:** • Introduce different types of validation controls (RequiredFieldValidator, CompareValidator, RangeValidator, CustomValidator). • Demonstrate how to use validation controls on web forms. • Explain how to customize error messages for validation controls. **Exercise (5 minutes)** – • What is View State and how is it used in web applications? • What are the different types of validation controls available in ASP.NET? • How do you customize the error message displayed by a validation control?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading

	View State. Validation Controls https://www.c-sharpcorner.com/UploadFile/puranindia/validation-controls-in-Asp-Net/ Homework • Create a simple web form with textboxes for name, email, and age. • Implement validation using appropriate validation controls for each field. • Display custom error messages for invalid input
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 34	**Course Name: Web Technologies** **Topic** Data Bind Control and Types of Data Bind Controls Repeater, Data List	**Course No.: MCA-302**

Objectives	At the end of the lesson the student shall be able to: • Understand the concept of data binding in web development. • Learn about the Repeater and Data List controls. • Differentiate between Repeater and Data List controls. • Be able to use Repeater and Data List controls effectively in web applications.
Teaching Aids (if any)	a. Videos of: Data Bind Control and Types of Data Bind Controls Repeater, Data List https://youtu.be/WB7GjRlqGL8 b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) • Begin by explaining the importance of displaying dynamic data on web pages. • Introduce the concept of data binding as a mechanism to connect data sources with UI elements. • Briefly mention the benefits of using data binding controls. 2. **Development** (30 minutes) **a. Data Binding** • Explain the basic concept of data binding, binding data sources to UI elements. • Discuss the two main types of data binding: one-way and two-way. **b. Repeater Control:** • Define the Repeater control and its purpose. • Demonstrate how to create a Repeater control and bind it to a data source. • Explain the use of templates for defining the layout of repeated items. **c. Data List Control** • Define the Data List control and its purpose. • Explain the difference between Repeater and Data List controls. • Demonstrate how to create a Data List control and bind it to a data source.

	Exercise (5 minutes) – • What is data binding? • What is the difference between a Repeater and a Data List control? • When would you use a Repeater control over a Data List control?
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Data Bind Control and Types of Data Bind Controls Repeater, Data List https://www.c-sharpcorner.com/UploadFile/puranindia/data-bind-controls-in-Asp-Net/ Homework • Create a simple web page using either a Repeater or Data List control to display a list of items (e.g., products, news articles). • Experiment with different data sources and templates to customize the output.
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on Text formatting tags Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 4	**Course Name: Web Technologies**	**Course No.: MCA-313**

Objectives	At the end of the lesson the student shall be able to: a. Understand embedding objects in HTML b. Use of paragraphs in HTML
Teaching Aids (if any)	a. Power point presentation b. Use of Nearpod tool for online quiz
Teaching Development	1. **Introduction** (5 minutes) - Ask questions How objects can be embedded in HTML? What are the different objects that can be embedded? How we can use paragraph tag in HTML? 2. **Development** (30 minutes) a. Embedding Objects - video -audio -flash animations b. Paragraph tag -HTML Paragraph -Space inside HTML Paragraph -Use of tag with paragraph? -Use of <hr> tag with paragraph? **Exercise** (5 minutes) – Use google forms and Nearpod tool for quiz based on the topic Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Homework -Design a web page by making use of embedding tag and paragraph tags in HTML Spend 5 minutes to wrap up and consolidate the learnings

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod quiz Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 16	**Course Name: Web Technologies**	**Course No.: MCA-313**

Objectives	At the end of the lesson the student shall be able to: a. Implement event driven programming using JavaScript b. Classification of different events
Teaching Aids (if any)	a. Power point presentation b. Use of Nearpod tool for online quiz c. Whiteboard and Marker
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What is an event What is an event driven programming What are the benefits of using event driven programming? Introduce the concept of event driven programming using JavaScript Show figures on Slide Events in JavaScript, classification of events, Mouse events, keyboard events, form events 2. **Development** (30 minutes) a. Evets in Java script - Introduction to events in JavaScript -Introduction to event driven programming -How different events are used b. Classification of events -mouse events -keyboard events -form events c. Different mouse events - click -mouseover -mouseout -mousedown -mouseup -mousemove -By using an Array constructor d. Different keyboard events -keyboardup -keyboarddown

	e. Different form events -focus -submit -blur -change 3. **Exercise** (5 minutes) – Use google forms and Nearpod tool for quiz based on the topic Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested video lecture 3. Homework -Write a program to calculate factorial of a no using events in JavaScript -Write a program to design a bio data form and perform data validation using events in JavaScript Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on events Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 15	**Course Name: Web Technologies**	**Course No.: MCA-313**

Objectives	At the end of the lesson the student shall be able to: a. articulate the concept of functions in JavaScript b. Importance of functions c. Understanding Objects
Teaching Aids (if any)	a. Power point presentation b. Use of Nearpod tool for online quiz c. White board and Marker
Teaching Development	1. **Introduction** (5 minutes) - Ask questions What is a function Why we use functions What is code reusability Introduce the concept of objects in JavaScript Show figures on Slide Introduction to functions, syntax to declare a function, calling a function 2. **Development** (30 minutes) a. Functions in Java script - Introduction to functions -Advantages of using functions -JavaScript Function Syntax -JavaScript Function Arguments -Function with Return Value b. Objects -Creating Objects in JavaScript -JavaScript Object by object literal -By creating instance of Object -By using an Object constructor 3. **Exercise** (5 minutes) – Use google forms and Nearpod tool for quiz based on the topic Use Nearpod to collect responses and discuss the answers.
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested video lecture https://www.youtube.com/watch?v=o1IaduQICO0Homework - Write a program to calculate factorial of a no using functions - Explain functions in detail and give suitable example in support of your answer Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Open ended questions using Nearpod tool

	Spend 5 minutes to evaluate student assimilation of the lesson contents