Core Practical 1 For the students admitted from A.Y. 2023-2024& onwards		
Offering Department: **Computer Application**	Offered to: **Master of Computer Application**	
Semester – I		
Course Code	Course Title	Course Credit and Hours
23MCACC105	**Core 1:** ResponsiveWeb Design and TypeScript	**2 Credits - 4 hrs/wk**

Course Description:

This comprehensive course covers web development with Bootstrap 5, HTML 5, CSS 3, Typescript, Animation using CSS, and Responsive Web Design. Suitable for beginners and experienced developers, the course starts with an introduction to web development and the basics of HTML 5, CSS 3, and Bootstrap 5, before delving into advanced topics such as animation using CSS and responsive web design techniques. Throughout the course, students will work on hands-on projects to develop their skills and build a professional-quality web portfolio, enabling them to design, develop, and deploy responsive and engaging web applications using the latest web technologies and techniques.

Course Purpose:

The purpose of this course is to equip students with the knowledge and skills required to become proficient in web development using Bootstrap 5, HTML 5, CSS 3, Typescript, Animation using CSS, and Responsive Web Design. By the end of the course, students will have gained a comprehensive understanding of the latest web development technologies and techniques, and be able to create professional-quality, responsive, and engaging web applications. Through hands-on projects, students will gain practical experience and build a portfolio of work that demonstrates their skills to potential employees.

Course Outcomes: Upon completion of this course, the learners will be able to

CO No.	CO Statement	Bloom's Taxonomy Level (K_1 to K_6)
CO_1	Identify the fundamental concepts and components of web development, such as HTML 5, CSS 3, and Bootstrap 5.	K1
CO_2	Explain how web development technologies and techniques work, such as responsive web design and Typescript programming.	K2
CO_3	Create web pages and web applications using HTML 5, CSS 3, and Bootstrap 5, and apply responsive web design principles to ensure they work across different devices.	K3
CO_4	Assess the effectiveness of different web development technologies and techniques, and recommend the most appropriate ones for a given project.	K5
CO_5	**Apply** PSD to HTML Responsiveweb design.	K6

Course Content	Hours
Exercise-I: Introduction to CSS and Web Forms – Part - I	**9 hrs**
Topic: Creating selectors using property and value 1. Create an HTML document with at least 3 box elements. 2. Use CSS to apply a border to each box element.	

Course Content	Hours
3. Set the padding and margin for each box element. 4. Apply different position values to each box element. 5. Use CSS to float one or more of the box elements. 6. Apply transformation and transition effects to at least one box element. 7. Create an animation using CSS for one of the box elements. **Topic: Creating a form to collect contact information** 1. Create an HTML form with fields for name, email, phone number, and message. 2. Use CSS to style the input fields, including textboxes, bordered inputs, colored inputs, and focused inputs. 3. Add icons to the input fields using CSS. 4. Style the text areas and select menus in the form. 5. Use CSS to style the input buttons in the form. 6. Create a responsive form using CSS media queries. **Topic: Creating a responsive design with CSS3 and media query** 1. Create an HTML document with a header, navigation menu, content area, and footer. 2. Use CSS to create a responsive design using media queries. 3. Adjust the layout and styling of the header, navigation menu, content area, and footer based on screen size. 4. Use CSS to create a responsive menu that collapses on smaller screens. 5. Apply a different background color to the header and footer on different screen sizes.	
Exercise-II:Introduction to CSS and Web Forms – Part II	**9 hrs**
Topic: Creating selectors using property and value 1. Create an HTML document with at least 4 div elements, each containing an image and a heading. 2. Use CSS to apply a border, padding, and margin to each div element. 3. Apply different position values to each div element to create a grid layout. 4. Use CSS to float the images to the left or right of the headings. 5. Apply transformation and transition effects to the images. 6. Create a hover effect using CSS for the div elements. **Topic: Creating a form to collect contact information** 1. Create an HTML form with fields for name, email, phone number, and address. 2. Use CSS to style the input fields, including textboxes, bordered inputs, colored inputs, and focused inputs. 3. Add icons to the input fields using CSS. 4. Style the text areas and select menus in the form. 5. Use CSS to style the input buttons in the form. 6. Use HTML5 form validation to validate the email and phone number fields. 7. Create a responsive form using CSS media queries. **Topic: Creating a responsive design with CSS3 and media query** 1. Create an HTML document with a header, navigation menu, content area, and footer. 2. Use CSS to create a responsive design using media queries. 3. Adjust the layout and styling of the header, navigation menu, content area, and footer based on screen size. 4. Use CSS to create a responsive navigation menu that collapses on smaller screens and expands on larger screens. 5. Apply a different background color to the header and footer on different screen sizes. 6. Use CSS to create a responsive image slider using CSS3 and media queries.	

Course Content	Hours
Exercise-III: Introduction to Bootstrap	**9 hrs**
Topic: Uses of powerful mobile-first flex box grid to build layouts of all shapes and sizes twelve column systems. 1. Create an HTML document with a header, navigation menu, content area, and footer. 2. Use Bootstrap's mobile-first flex box grid to create a layout for the document. 3. Use the 12 column system to create different sized columns for the content area. 4. Apply responsive design principles to the layout using media queries to adjust the layout for different screen sizes. 5. Use Bootstrap's utility classes to style the layout and adjust the padding and margin of the elements. 6. Topic: Examples for Bootstrap's media object to construct highly repetitive components like blog comments, tweets, etc. 7. Create an HTML document with a blog post containing comments. 8. Use Bootstrap's media object to create a comment section with a user avatar, username, comment text, and date. 9. Use CSS to style the media object and adjust the spacing and font size. 10. Apply responsive design principles to the comment section using media queries to adjust the layout for different screen sizes. **Topic: Managing Content Using Bootstrap** 1. Create an HTML document with different types of content, including headings, paragraphs, lists, and images. 2. Use Bootstrap's typography classes to style the content, including the global settings, headings, body text, lists, and more. 3. Use Bootstrap's utility classes to style the content and adjust the padding and margin of the elements. **Topic: Examples for displaying inline and multiline blocks of code with Bootstrap.** 1. Create an HTML document with inline and multiline blocks of code. 2. Use Bootstrap's code and preformatted text classes to display the code and adjust the font family and size. 3. Apply responsive design principles to the code blocks using media queries to adjust the layout for different screen sizes. **Topic: Examples of images into responsive behaviour (so they never become larger than their parent elements) and add lightweight styles to them—all via classes.** 1. Create an HTML document with images of different sizes. 2. Use Bootstrap's responsive image classes to make the images responsive and never become larger than their parent elements. 3. Use CSS to add lightweight styles to the images, including borders, rounded corners, and shadows. 4. Apply responsive design principles to the images using media queries to adjust the layout for different screen sizes. **Topic: Examples for opt-in styling of tables with Bootstrap.** 1. Create an HTML document with a table containing data. 2. Use Bootstrap's table classes to style the table, including striped rows, hover rows, and	

Course Content	Hours
bordered tables. 3. Use CSS to adjust the font size and color of the table text. 4. Apply responsive design principles to the table using media queries to adjust the layout for different screen sizes.	
Unit-IV: Advance concept of Bootstrap	**9 hrs**
1. Use the Bootstrap 5 alert component to show different types of messages on a webpage. Create alerts for success, warning, info, and error messages. 2. Create a responsive card layout using Bootstrap 5 cards. The cards should have images, titles, text, and buttons. 3. Use Bootstrap 5's button group component to group a series of buttons together on a single line. Add event listeners to the buttons that change the appearance of the webpage. 4. Create a responsive table using Bootstrap 5's table component. The table should have sorting and filtering functionality using Bootstrap's plugins. 5. Use Bootstrap 5's badge component to add labels to different parts of a webpage. Create badges for important notifications, status indicators, and other types of labels. 6. Implement Bootstrap 5's scrollspy feature to a one-page website. Use the scrollspy to highlight the current section of the webpage on the navigation menu. 7. Use Bootstrap 5's popover component to add additional information to elements on a webpage. Create popovers that appear when users hover over or click on specific elements. 8. Create a responsive progress bar using Bootstrap 5's progress component. The progress bar should show the completion status of a task or process, and should update dynamically as the task progresses.	
Exercise-V:Introduction of TypeScript	**9 hrs**
1. Write a TypeScript program that calculates the area of a rectangle given its length and width. Store the length and width in variables and the result in another variable. Display the result in the console. 2. Write a TypeScript program that prompts the user to enter a string and then displays the string in uppercase letters. 3. Write a TypeScript program that declares an array of numbers and finds the sum of all the elements in the array. 4. Write a TypeScript program that declares a tuple containing a person's name and age, and displays a message saying "Hello {name}, you are {age} years old." in the console. 5. Write a TypeScript program that prompts the user to enter a number and then displays whether the number is even or odd. 6. Write a TypeScript program that declares an array of strings and displays each string in the console using a for loop. 7. Write a TypeScript program that declares a function that takes two numbers as arguments and returns their sum. Call the function and display the result in the console. 8. Write a TypeScript program that declares a function that takes a string as an argument and returns the number of vowels in the string. Call the function and display the result in the console.	

Text books:

- Jonathan Fielding, “Beginning Responsive Web Design with HTML5 and CSS3”, Apress, ISBN:978-1-4302-6694-5
- Hege Refsnes, Stale Refsnes, Kai Jim Refsnes, Jan EgilRefsnes, C. Michael Woodward, “Learn HTMLand CSS with w3schools”, Wiley Publication, Inc., ISBN: 978-0-470-61195-1

Reference books:

- Jennifer Nierderst Robbins, Learning Web Design, 4th Edition, O’Reilly, ISBN: 978-1-449-31927-4
- Bootstrap : https://getbootstrap.com/docs/4.1/getting-started/introduction/

Pedagogic tools:

- Chalk and Board
- Power point presentation
- Seminar
- Videos
- Flipped classroom approach
- Peer instruction
- Collaborative learning
- Problem-based learning
- Lectures and demonstrations

Methods of Assessment & Tools:

Components of CIA: 40 marks

Sr. No.	Component	Content	Duration (if any)	Marks	Sub Total
A	Practical Skill Assessment	All Experiments	3 hours	30 (Set for 30)	30
B	Observation and Book of Records	All Experiments	-	10 (10 marks)	10
				Grand Total	**40**
Lab Methods:	• Hands-on exercises • Programming assignments • Demonstrations • Code reviews				
Assessment Tools:	• Self-assessments • Peer assessments • Code documentation				