Code: BCSL504 **Course:Fullstackdevelopment lab**

Credits: 1 **L:T:P - 0:0:2**

SEE: 50 **CIE: 50**

PracticalHours:2 hrs/week **Max.Marks: 100**

Course Outcomes:

On the successful completion of the course, the student will be able to

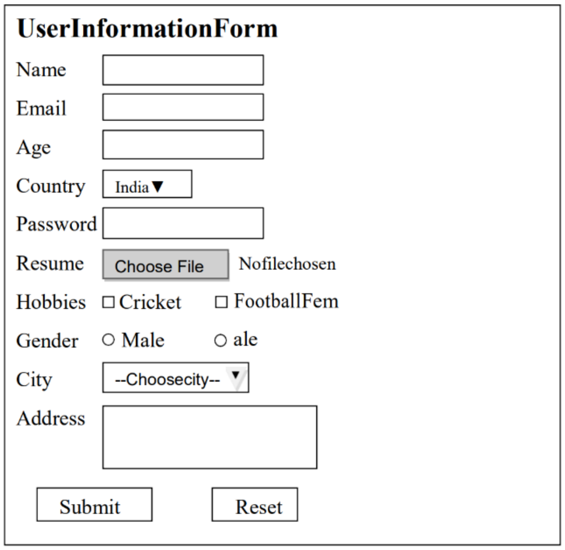
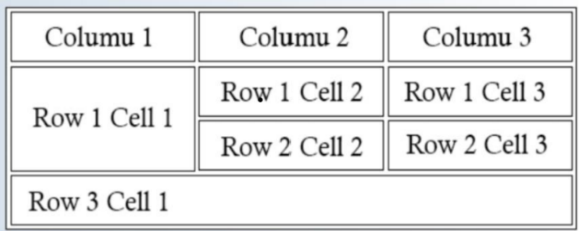
COs	**Course Outcomes**	**Bloom's level**
CO1	Implement web pages using HTML and CSS.	Apply
CO2	Demonstrate core constructs and event handling mechanisms of JavaScript to develop webPages.	Apply
CO3	Implement interactive web applications using ReactJS.	Analyze
CO4	Build dynamic and user-responsive web applications with Node.js and Express.js.	Apply
CO5	Demonstrate the creation of dynamic web pages and connecting to MYSQL/MongoDB	Apply

Mapping with POs and PSOs

COs	PO 1	PO 2	PO3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO 12		PS O1	PS O2
CO1	3	2	3	2	2	1	-	1	-	-	1	2		3	3
CO2	3	2	2	2	2	1	-	1	-	-	1	2		3	3
CO3	3	2	3	2	3	1	-	1	-	-	1	2		3	3
CO4	3	2	3	2	3	1	-	1	-	-	2	2		3	3
CO5	3	2	3	2	3	1	-	1	-	-	2	2		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Sl.No	**Lab Experiments**	**Course outcome**

<table>
<tr>
<td>1</td>
<td>Create the below registration form using forms in HTML.

UserInformationForm
Name
Email
Age
Country India▼
Password
Resume Choose File Nofilechosen
Hobbies □Cricket □FootballFem
Gender ○ Male ○ ale
City --Choosecity-- ▼
Address
Submit Reset</td>
<td>CO1</td>
</tr>
<tr>
<td>2</td>
<td>Design a webpage using HTML that meets the following requirements:
1. Create a table as shown below.
2. The first row of the table should include the headers: <b>"Column 1"</b>, <b>"Column 2"</b>and <b>"Column 3"</b>.
3. Add two additional rows with sample data under these headers.
4. Use <b>inline CSS</b> to:
a. Add a <b>border</b> to the table and its cells.
b. <b>Center-align</b> all the text within the table.

<table>
<tr><td>Columu 1</td><td>Columu 2</td><td>Columu 3</td></tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table></td>
<td>CO1</td>
</tr>
<tr>
<td>3</td>
<td>Create a visually appealing webpage that showcases the CSS box model by applying styling to a<div> element, incorporating padding, margins and borders. Additionally, include an image and a hyperlink within the page to enhance the layout.</td>
<td>CO1</td>
</tr>
</table>

4	Write a JavaScript program that demonstrates how to handle events from different HTML elements. The program should: 1. Trigger an alert when the user clicks anywhere on the body of the page. 2. Change the text of a button when it is clicked and display a success message below the button. 3. Dynamically update a paragraph with the text entered in a text box as the user types. 4. Display a password strength message (weak, medium, or strong) based on the length of the entered password, updating the message as the user types. 5. Ensure that the program uses appropriate event handlers for each element and explains the purpose of each event handler in the program.	CO2
5	Create a JavaScript program where a random number between 1 and 100 is generated and the user is prompted to guess the number. Provide feedback on whether the guess is too high, too low, or correct, and allow the user to keep guessing until they get it right. After a correct guess, ask if they want to play again using confirm(); if yes, restart the game, otherwise, display "Thanks for playing!" and end the program.	CO2
6	Write a JavaScript program to create an array of student objects, convert their marks to CGPA using the map method and display a list of students with a CGPA of 9 or higher. Also, show the total count of students with CGPA 9 and above using the filter and reduce methods.	CO2
7	1. Create an HTML page with a list of items, a few buttons and some paragraphs. 2. Write JavaScript to do the following: a) Change the background color of an element by its id. b) Change the text color of all list items (<li>) to blue. c) Change the text of all buttons to "Clicked!" when clicked. d) Increase the font size of all paragraphs. e) Select the first element with a specific class and change its text.	CO2
8	Demonstrate all the methods of mounting and updating phase of react class Component by changing the city name from “Mysore” to “Bangalore” after 2 seconds and display the same.	CO3
9	Illustrate react program to implement error boundaries to find 1. Division by zero error. 2. Array index of out of bound.	CO3
10	Implement a React functional component that: 1. Accepts an age prop and displays it inside a heading (<h1>). 2. Includes an input field and a button that allows the user to update the age in the heading using a state variable (utilizing useState&useEffect).	CO3
11	Write a react program to create a list of departments and iterate through all the elements of the list and return an unordered list with each department as a list item using a function component.	CO3
12	Write a react function component to create a counter that is incremented by one on button click and display a message 'End of the count ' when count reaches 10.	CO3

13	Write a Node.js program to read input from the console and store it in a file using file server module.	CO4
14	Apply the concept of custom modules in Node.js. 1. Create a custom module fileOperations.js with functions to read a file, write to a file and append data to a file. 2. Use this module in another file app.js to demonstrate its functionality by performing file operations.	CO4
15	Illustrate how to add an event listener, emit an event to trigger the listener and then remove the event listener in node.js.	CO4
16	Explain various debugging and logging methods in Node.js with example to demonstrates the following: 1. Measuring execution time of code blocks. 2. Grouping related log messages and displaying them hierarchically. 3. Using assertions to conditionally display error messages. 4. Displaying data in a tabular format. 5. Logging debugging details for analysis.	CO4
17	Write a Node.js program to: 1. Create an HTTP server that listens on port 8000. 2. Respond with “Welcome to Nodejs” when the root URL (/) is accessed. 3. Return a 404 status with a custom message for all other routes.	CO4
18	Demonstrate the following MongoDB operations 1. Write a script to create a new database named CollegeDB and a collection named students. 2. Insert three documents into the students collection with fields name, age, sem, usn and grade. 3. Insert multiple documents into a collection named courses with fields courseName, courseCode, and credits. 4. Write queries to retrieve all documents from the students collection. 5. Retrieve all documents where the age is greater than 18. 6. Write a query to find all students with a grade of "A". 7. Write a query to find all courses with credits greater than or equal to 3.	CO5
19	Demonstrate the following operations using Mongo shell script 1. Switch to the WorkDB database. 2. Create a teachers collection. 3. Insert documents with fields name, subject and experience. 4. Use the Mongo shell to export the students collection to a JSON file. 5. Import the JSON file back into a new collection named studentsBackup.	CO5
20	Write a MongoDB aggregation query for the following to analyze an employees collection with an example for each:Group the employees by their department and calculate the total number of employees in each department. The average salary of employees in each department. 1. Sort the aggregated results by the total number of employees in descending order. 2. Example Data in the employees Collection.	CO5

Code:BCS505 **Course: Full Stack Development**

Credits: 3 **L:T:P - 3:0:0**

SEE: 50 **CIE: 50**

SEE Hours: 3 Max. Marks: 100

Prerequisites if any	Computer networks
Learning objectives	1. Create web page using HTML & CSS 2. Develop familiarity with the JavaScript language,realize concepts commonly used in dynamic language programming, such as introspection, higher-order functions, closures, familiar with common libraries and tools that are used in web application development. 3. To Create React Components, lifecycle of components, rendering list and portal and perform some simple tests, and error handling. 4. To create Node.js modules, events and database access and interact with databases using MongoDB .

Course Outcomes:*On the successful completion of the course, the student will be able to*

COs	**Course Outcomes**	**Bloom's level**
CO1	Prepare web pages using HTML and CSS.	Apply
CO2	Apply core constructs and event handling mechanisms of JavaScript to develop webPages.	Apply
CO3	Build interactive web applications using ReactJS	Analyze
CO4	Develop interactive web applications using Node.js and Express.js.	Apply
CO5	Demonstrate the creation of dynamic web pages and connecting to MYSQL/MongoDB	Apply

Mapping with POs and PSOs:

COs	**PO1**	**PO 2**	**PO3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1	3	2	3	2	2	1	-	1	-	-	1	2		3	3
CO2	3	2	3	2	2	1	-	1	-	-	1	2		3	3
CO3	3	2	3	2	3	1	-	1	-	-	1	2		3	3
CO4	3	2	3	2	3	1	-	1	-	-	1	2		3	3
CO5	3	2	3	2	3	1	-	1	-	-	1	2		3	3

Mapping Strength: Strong– 3 Medium – 2 Low – 1

Course Structures

Nos.	Modules	No. of Lecture Hours	No. of Tutorial Hours	No. of Practical Hours
Modules -1:HTML &CSS				
1.1	Introduction to HTML, Basic syntax, Standard HTML Document Structure, Images,	1	-	-
1.2	Paragraphs, Line Breaks, Heading Tags, Font Styles, Sizes, Color, Character Entity	1	-	-
1.3	links, Lists, tables, form tags, Font properties, text decoration, List property,	2	-	-
1.4	CSS Colors and background, CSS Box Model, CSS Margins, Padding, Borders.Background images, span and div tags, Grid and flex.	2	-	-
Module 2: JavaScript				
2.1	Introduction to JavaScript, General syntactic characteristics	1	-	-
2.2	Primitives, operations, expressions	1	-	-
2.3	Screen output and keyboard input, Control statements,	1	-	-
2.4	Object creation &modification, arrays, array methods, Functions,	2	-	-
2.5	JavaScript and XHTML Documents: The JavaScript execution environment, TheDocument Object Model, Element access in JavaScript, Events and event handling,	2	-	-
2.6	Handling events from the Body elements, Button elements, Text box and Password elements.	2	-	-
Module – 3: ReactJS				
3.1	Introduction, Templating using JSX	1	-	-
3.2	Components, State, Props,Hooks	2	-	-
3.3	Lifecycle of Components	2	-	-
3.4	Rendering List and Portals	2	-	-
3.5	Error Handling, Routers	2	-	-
Module4: NodeJS				
4.1	Node.js overview, Node.js – basics and setup.	1	-	-
4.2	Node.js console, Node.js command utilities	1	-	-

4.3	Node.js modules, Node.js concepts	2	-	-
4.4	Node.js events, Node.js database access.	2		-
4.5	Introduction to Express.js • Creating routes (GET, POST, PUT, DELETE) • Handling request and response objects • Middleware in Express (built-in, custom, third-party)	2		
Module 5: MongoDB				
5.1	MongoDB basics: Documents, collections, database query language, mongo shell.	2	-	-
5.2	MongoDB CRUD operations (create, read, projection, update, delete aggregate)	2	-	-
5.3	reading from MongoDB, writing from MongoDB	2	-	-
5.4	MongoDB with PHP, MongoDB with NodeJS	2	-	-
	Total No. of Lecture Hours	**40**		-
	Total No. of Tutorial Hour		**00**	-
	Total No. of Practical Hours			**00**

Textbook:

1. Programming the World Wide Web, Robert W. Sebesta, 4th Edition, Pearson Education, 2008.
2. Full-Stack React Projects: Learn MERN Stack Development by Building Modern Web Apps Using MongoDB, Express, React, and Node.js, 2nd Edition.

Reference Book:

1. Open-Source Web Development with Lamp, James Lee and Brent Ware, Pearson Education, 2009.

2. Internet and World Wide Web: How to Program -Harvey M. Deitel, Paul J. Deitel, 4thedition, Pearson, 2009.
3. The Web Programming Building Internet Applications- Chris Bates, 3rd Edition, Wiley India, 2006

Online Resources:

1. MongoDB Notes for Professionals book Tutorial for Beginners in PDF by GoalKicker.com
2. https://docs.google.com/viewer?url=https://www.computer-pdf.com/pdf/0840-mongodb-notes-for-professionals-book.pdf
3. https://www.javatpoint.com/reactjs-tutorial