



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

WEB PROGRAMMING

LAB MANUAL

SUBJECT CODE: BCS23216P

Prepared by:-

RUBI KALITA

Laboratory Instructor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GUWAHATI-781017



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Course Syllabus:

Course Code: BCS23216P

Course Title: Web Technology

Credits: 2-0-2-3

Pre-requisites: None

Course Type: PCC

Course Outcomes

- Design and develop static and responsive web pages using HTML, CSS, and JavaScript
- Build backend services using Node.js and Express
- Create modern frontend applications using React.js
- Integrate backend, frontend, and database into a working full-stack application
- Develop mini-projects such as Student and Health Information Systems

Module I: Web Page Designing

- Introduction to HTML: Elements, Lists, Tables, Images, Forms, Frames
- Cascading Style Sheets (CSS)
- Introduction to DHTML
- XML Basics: DTD (Document Type Definition), XML Schemas
- DOM (Document Object Model)

Module II: Client-Side Scripting (JavaScript)

- JavaScript Basics: Variables, Data types, Control Structures
- Functions and Arrays
- Objects and Events
- Dynamic HTML using JavaScript



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Module III: Server-Side Programming using Node.js

- Web Servers: Apache (XAMPP/LAMPP), Node.js Environment
- Node.js Basics:
 - Modules, File System, HTTP, Express.js
 - Handling Forms, Routing, Middleware
- Session Management using Express
- Cookies and Authentication
- Introduction to REST API with Node.js

Module IV: Frontend & Database Integration

- React.js:
 - Components, Props, State, Events
 - Forms and Hooks
 - Axios for API Calls
- Database Connectivity:
 - MySQL Integration with Node.js
 - CRUD operations
- Case Study Implementations (mandatory):
 - Student Information System
 - Health Management System

Textbooks

- Paul Deitel, Harvey Deitel, Abbey Deitel – Internet & World Wide Web: How to Program, 5th Edition, Pearson, 2012.
- Ethan Brown – Web Development with Node and Express, O'Reilly, Latest Edition
- Robin Wieruch – The Road to React, Leanpub



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Module I – Web Page Designing

Introduction to HTML

(a) Basic HTML Structure

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to Web Technology</h1>
  <p>This page demonstrates basic HTML elements.</p>
</body>
</html>
```

□ **Output:**

Heading and paragraph displayed on a white background.

(b) Lists

```
<!DOCTYPE html>
<html>
<head><title>HTML Lists</title></head>
<body>
<h2>Ordered List</h2>
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>

<h2>Unordered List</h2>
<ul>
  <li>Frontend</li>
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
<li>Backend</li>
<li>Database</li>
</ul>
</body>
</html>
```

□ **Output:**

Numbered and bulleted lists.

(c) **Tables**

```
<!DOCTYPE html>
<html>
<head><title>Student Table</title></head>
<body>
<h2>Student Details</h2>
<table border="1" cellpadding="5">
  <tr>
    <th>Roll No</th>
    <th>Name</th>
    <th>Marks</th>
  </tr>
  <tr>
    <td>101</td>
    <td>Riya</td>
    <td>88</td>
  </tr>
  <tr>
    <td>102</td>
    <td>Aman</td>
    <td>92</td>
  </tr>
</table>
</body>
</html>
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

☐ **Output:**

Simple table with two student records.

(d) Images

```
<!DOCTYPE html>
<html>
<head><title>Image Example</title></head>
<body>
<h2>College Logo</h2>

</body>
</html>
```

☐ **Output:**

Displays the image “college_logo.jpg”.

(e) Forms

```
<!DOCTYPE html>
<html>
<head><title>Student Form</title></head>
<body>
<h2>Student Registration Form</h2>
<form>
  Name: <input type="text" name="name"><br><br>
  Email: <input type="email" name="email"><br><br>
  Gender:
  <input type="radio" name="gender" value="Male"> Male
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
<input type="radio" name="gender" value="Female"> Female <br><br>
Course:
<select>
  <option>BCA</option>
  <option>B.Tech</option>
</select><br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

☐ **Output:**

Form with input fields, radio buttons, dropdown, and submit button.

(f) Frames (using iframe)

```
<!DOCTYPE html>
<html>
<head><title>Frames Example</title></head>
<body>
<h2>Using iframe</h2>
<iframe src="https://www.wikipedia.org" width="600" height="300"></iframe>
</body>
</html>
```

☐ **Output:**

Displays Wikipedia page inside your webpage.

Cascading Style Sheets (CSS)

(a) Inline CSS

```
<p style="color:blue; font-size:20px;">This is a blue paragraph.</p>
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

(b) Internal CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body { background-color: lightyellow; }
h1 { color: green; text-align: center; }
p { font-family: Arial; color: darkblue; }
</style>
</head>
<body>
<h1>Internal CSS Example</h1>
<p>This text uses internal CSS for styling.</p>
</body>
</html>
```

□ **Output:**

Yellow background, centered green heading, blue paragraph.

(c) External CSS

□ *style.css*

```
body {
  background-color: lightcyan;
  font-family: Verdana;
}
h2 {
  color: purple;
}
```

□ *index.html*

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
</head>
<body>
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
<h2>External CSS Example</h2>
<p>This page uses an external stylesheet.</p>
</body>
</html>
```

□ **Output:**

Text styled using style.css.

Introduction to DHTML (Dynamic HTML)

Dynamic HTML = HTML + CSS + JavaScript
Used to modify page content dynamically.

```
<!DOCTYPE html>
<html>
<head>
<title>DHTML Example</title>
<style>
#text { color: darkgreen; font-size: 22px; }
</style>
</head>
<body>
<h2>Dynamic HTML Example</h2>
<p id="text">This text will change dynamically!</p>
<button onclick="changeText()">Click Me</button>
<script>
function changeText() {
    document.getElementById("text").innerHTML = "Text changed using JavaScript!";
    document.getElementById("text").style.color = "red";
}
</script>
</body>
</html>
```

□ **Output:**

Click button → text color and content change instantly.



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

XML Basics

(a) Simple XML Document

□ *student.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
  <student>
    <name>Riya</name>
    <course>BCA</course>
    <marks>88</marks>
  </student>
  <student>
    <name>Aman</name>
    <course>B.Tech</course>
    <marks>92</marks>
  </student>
</students>
```

□ **Output (when opened in browser):**

Displays structured XML data tree.

(b) DTD (Document Type Definition)

□ *student.dtd*

```
<!ELEMENT students (student+)>
<!ELEMENT student (name, course, marks)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT course (#PCDATA)>
<!ELEMENT marks (#PCDATA)>
```

□ *student.xml*

```
<?xml version="1.0"?>
<!DOCTYPE students SYSTEM "student.dtd">
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
<students>
  <student>
    <name>Riya</name>
    <course>BCA</course>
    <marks>88</marks>
  </student>
</students>
```

(c) XML Schema

□ *student.xsd*

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="students">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="student" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="course" type="xs:string"/>
              <xs:element name="marks" type="xs:integer"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

DOM (Document Object Model)

DOM allows scripts to read and modify web page elements.

```
<!DOCTYPE html>
<html>
<head><title>DOM Example</title></head>
<body>
<h2 id="heading">Original Heading</h2>
<p id="info">This is a paragraph.</p>
<button onclick="changeDOM()">Change Content</button>

<script>
function changeDOM() {
  document.getElementById("heading").innerHTML = "Updated Heading via DOM!";
  document.getElementById("info").style.color = "blue";
}
</script>
</body>
</html>
```

□ **Output:**

Click button → heading and text color update dynamically.



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

JavaScript Basics: Variables, Data Types, Control Structures

```
<!DOCTYPE html>
<html>
<head><title>JavaScript Basics</title></head>
<body>
<h2>JavaScript Basics Example</h2>
<p id="output"></p>
<script>
// Variables and Data Types
let name = "Rubi";
let age = 20;
let marks = 85.5;
let passed = true;
// Control Structure (if-else)
let grade = "";
if (marks >= 80) grade = "A";
else if (marks >= 60) grade = "B";
else if (marks >= 40) grade = "C";
else grade = "D";
// Output
document.getElementById("output").innerHTML =
`Name: ${name}<br>Age: ${age}<br>Marks: ${marks}<br>Grade: ${grade}<br>Passed:
${passed}`;
</script>
</body>
</html>
```

□ Output (in browser):

Name: Rubi
Age: 20
Marks: 85.5
Grade: A
Passed: true



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Functions and Arrays

```
<!DOCTYPE html>
<html>
<head><title>Functions and Arrays</title></head>
<body>
<h2>Functions and Arrays Example</h2>
<p id="arrOutput"></p>

<script>
function getAverage(marks) {
  let sum = 0;
  for (let m of marks) sum += m;
  return sum / marks.length;
}

let marks = [85, 90, 78, 88];
let avg = getAverage(marks);

document.getElementById("arrOutput").innerHTML =
  `Marks: ${marks}<br>Average: ${avg}`;
</script>
</body>
</html>
```

□ Output:

Marks: 85,90,78,88
Average: 85.25

Objects and Events



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
<!DOCTYPE html>
<html>
<head><title>Objects and Events</title></head>
<body>
<h2>Objects and Events Example</h2>
<button onclick="showStudent()">Show Student Info</button>
<p id="studentInfo"></p>

<script>
const student = {
  name: "Rohit",
  course: "B.Tech",
  roll: 101
};

function showStudent() {
  document.getElementById("studentInfo").innerHTML =
    `Name: ${student.name}, Course: ${student.course}, Roll: ${student.roll}`;
}
</script>
</body>
</html>
```

☐ **Click Output:**

Name: Rohit, Course: B.Tech, Roll: 101



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Dynamic HTML (DHTML) using JavaScript

```
<!DOCTYPE html>
<html>
<head><title>Dynamic HTML</title></head>
<body>
<h2 id="heading">Welcome!</h2>
<button onclick="changeContent()">Change Content</button>

<script>
function changeContent() {
  document.getElementById("heading").innerHTML = "Hello, JavaScript Learner!";
  document.body.style.backgroundColor = "lightyellow";
}
</script>
</body>
</html>
```

□ Click Output:

Changes heading text and background color dynamically.



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Module III – Server-Side Programming using Node.js

Simple Node.js Server

```
// server.js
const http = require('http');
const server = http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.end('<h1>Hello from Node.js Server</h1>');
});
server.listen(3000, () => console.log('Server running on http://localhost:3000'));
```

☐ Output in browser:

Hello from Node.js Server

Express.js Example with Routing

```
// app.js
const express = require('express');
const app = express();

app.get('/', (req, res) => res.send('Welcome to Express Home Page'));
app.get('/about', (req, res) => res.send('About Us Page'));
app.get('/contact', (req, res) => res.send('Contact Page'));

app.listen(3000, () => console.log('Server started at http://localhost:3000'));
```

☐ Output:

- / → “Welcome to Express Home Page”



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

- /about → “About Us Page”
 - /contact → “Contact Page”
-

Handling Forms

```
// formServer.js
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

app.use(bodyParser.urlencoded({ extended: true }));

app.get('/', (req, res) => {
  res.send(`
    <form method="POST" action="/submit">
      Name: <input type="text" name="name"><br>
      Age: <input type="number" name="age"><br>
      <button type="submit">Submit</button>
    </form>
  `);
});

app.post('/submit', (req, res) => {
  res.send(`Hello ${req.body.name}, you are ${req.body.age} years old!`);
});

app.listen(3000, () => console.log('Form server running on port 3000'));
```

□ Output:

Displays a form → user enters data → displays message on submission.



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

REST API Example

```
// api.js
const express = require('express');
const app = express();
app.use(express.json());

let students = [
  { id: 1, name: "Riya" },
  { id: 2, name: "Aman" }
];

app.get('/students', (req, res) => res.json(students));
app.post('/students', (req, res) => {
  students.push(req.body);
  res.json({ message: "Student added", data: students });
});

app.listen(3000, () => console.log('API running on port 3000'));
```

□ Endpoints:

- GET /students
 - POST /students
-



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Module IV – Frontend & Database Integration

React.js Basic Component

```
// App.js
import React, { useState } from 'react';

function App() {
  const [name, setName] = useState("");
  return (
    <div className="App">
      <h2>React Form Example</h2>
      <input type="text" value={name} onChange={(e) => setName(e.target.value)} />
      <p>Hello {name}</p>
    </div>
  );
}
export default App;
```

❑ **Output:** Typing updates the message in real time.

Axios API Call

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';

function StudentList() {
  const [students, setStudents] = useState([]);

  useEffect(() => {
    axios.get('http://localhost:3000/students')
      .then(res => setStudents(res.data))
      .catch(err => console.log(err));
  });
}
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
}, []);

return (
  <div>
    <h3>Students List</h3>
    <ul>{students.map(s => <li key={s.id}>{s.name}</li>)}</ul>
  </div>
);
}
```

```
export default StudentList;
```

MySQL Database Integration

```
// dbConnect.js
const mysql = require('mysql2');
const express = require('express');
const app = express();
app.use(express.json());

const conn = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'school'
});

conn.connect(err => {
  if (err) throw err;
  console.log('Connected to MySQL');
});

// CRUD - Read
app.get('/students', (req, res) => {
  conn.query('SELECT * FROM students', (err, result) => {
    if (err) throw err;
    res.json(result);
  });
});
```



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

```
app.listen(3000, () => console.log('Server with MySQL running on port 3000'));
```

□ **Output:** Returns all students as JSON.



GIRIJANANDACHOWDHURYUNIVERSITY

Hathkhowapara, Azara, Guwahati 781017, Assam

Case Study – Student Information System (Mini Project)

Tech Stack:

React (frontend) + Node/Express (backend) + MySQL (database)

Features:

- Add Student
- View Students
- Delete Student

You'll have:

```
frontend/  
├── src/  
│   ├── App.js  
│   ├── components/StudentForm.js  
│   └── components/StudentList.js  
backend/  
├── server.js  
├── dbConnect.js  
└── routes/students.js
```
