Rayat Shikshan Sanstha’s

R.B. Narayanrao Borawake College, Shrirampur

(Autonomous)

National Education Policy

(Affiliated to Savitribai Phule Pune University)

Two Years Degree Program in Computer Science (Faculty ofScience and Technology)

Syllabus under Autonomy and National Education Policy

M. Sc. (Computer Science) Part- II

Practical Assignment Lab Book

Choice Based Credit System [CBCS] Syllabus for National Education Policy to be implemented from Academic Year 2024-2025

Full Stack Development (CS-ME-5637P)

Assignment Completion Sheet

Name of the Student:

Roll No:

Sr. No.	Assignment Title	Marks Obtained	Signature of Instructor
1	Create a file node.js that will covert output "HelloWorld" in uppercase.		
2	Create node.js file that select all record from "customer" tables and delete specific record		
3	Create node.js file that write HTML form withupload field.		
4	Create node.js file that opens the requested file andreturn the content to the Client if anything goes wrong throw a 404 error.		
5	Create node.js file that read data from multiple file asynchrony using promises and async/await.		
6	Implement a simple server using Node.js that handles multiple client connection.		
7	Create node.js file to insert multiple record instudent tables and display result object on Console.		
8	Create node.js server using node.js		

Total Marks :

Converted into 10 Marks :
Date:

Signature of In Charge **Head**

Internal Examiner **External Examiner**

Assignment No 1

Q.1) Create a file node.js that will covert output “Hello World” in uppercase.

```
const originalText = "Hello World";
const upperText = originalText.toUpperCase();
console.log(upperText);
```

Assignment Evaluation

0: Not Done		1: Incomplete		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date: **Practical In-charge**

Assignment No .2

Q.2) Create node.js file that select all record from “customer” tables and delete specific record .

```
 const mysql = require('mysql2');
const connection = mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password:  'password',
    database:  'your_database_name'
});
connection.connect(err => {
    if (err) {
        return console.error('Error connecting: ' + err.stack);
    }
    console.log('Connected  as  id  '  +  connection.threadId);
    const selectAllCustomers = () => {
        connection.query('SELECT * FROM customer', (error, results) => {
            if (error) {
                return console.error('Error fetching records: ' + error.stack);
            }
            console.log('All Customers:', results);
            const customerIdToDelete = 1;
            deleteCustomer(customerIdToDelete);
        });
    };
```

```
  const deleteCustomer = (customerId) => {
    connection.query('DELETE FROM customer WHERE id = ?', [customerId], (error, results) => {
      if (error) {
        return console.error('Error deleting record: ' + error.stack);
      }
      console.log(`Deleted ${results.affectedRows} record(s) with ID ${customerId}`);
      connection.end();
    });
  };
  selectAllCustomers();
});
```

Assignment Evaluation

0: Not Done		1: Incomplete		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date: **Practical In-charge**

Assignment No. 3

Q.3) Create node.js file that write HTML form with upload field.

```
const http = require('http');

const server = http.createServer((req, res) => { res.writeHead(200, {

    'Content-Type': 'text/html'
});

    res.end(

        <html>

        <body>
            <h1>Upload Form</h1>

            <form action="/upload" method="post"
enctype="multipart/form-data">

                <label for="file">Select file:</label>

                <input type="file" id="file"
name="file"><br><br>

                <input type="submit" value="Upload">

            </form>

        </body>

        </html>

    `);

});

server.listen(3000, () => {

    console.log('Server is running on
http://localhost:3000');

});
```

Assignment Evaluation

0: Not Done		1: Incomplete		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date: **Practical In-charge**

Assignment No .4

Q.4) Create node.js file that opens the requested file and return the content to the Client if anything goes wrong throw a 404 error.

```
 const http = require('http');
const fs = require('fs');
const path = require('path');
const server = http.createServer((req, res) => {
const filePath = path.join(__dirname, req.url);

  fs.readFile(filePath, (err, data) => {
    if (err) {
      res.writeHead(404, { 'Content-Type': 'text/plain' });
      return res.end('404 Not Found');
    }
    res.writeHead(200,  {  'Content-Type':  'text/plain'  });
    res.end(data);
  });
});

server.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

Assignment Evaluation

0: Not Done		1: Incomplete		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date: **Practical In-charge**

Assignment No.5

Q.5) Create node.js file that read data from multiple file asynchronsly using promises and async/await .

```
const fs = require('fs').promises;
async function readFiles(fileNames) {
   const fileReadPromises = fileNames.map(fileName => fs.readFile(fileName, 'utf-8'));
   try {
      const results = await Promise.all(fileReadPromises);
      console.log('File Contents:', results);
   } catch (error) {
      console.error('Error reading files:', error);
   }
}

const filesToRead = ['file1.txt', 'file2.txt', 'file3.txt'];
readFiles(filesToRead);
```

Assignment Evaluation

0: Not Done		1: Incomplete		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date: **Practical In-charge**

Assignment No.6

Q.6) Implement a simple server using Node.js that handles multiple client connection .

```
 const http = require('http');
const server = http.createServer((req, res) => {
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    res.end('Hello, Client! You are connected to the server.\n');
});

server.listen(3000, () => {
    console.log('Server is running on http://localhost:3000');
});
```

Assignment Evaluation

0: Not Done		1: Incomplet		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date: **Practical In-charge**

Assignment No 7

Q.7) Create node.js file to insert multiple record in student tables and display result object on Console .

```
const mysql = require('mysql2/promise');

async function insertStudents() {
  const connection = await mysql.createConnection({
    host: 'localhost',
    user: 'root',
    password: 'password',
    database: 'your_database_name'
  });

  const students = [
    { name: 'Alice', age: 20 },
    { name: 'Bob', age: 22 },
    { name: 'Charlie', age: 23 }
  ];
  const insertPromises = students.map(student => {
    return connection.execute('INSERT INTO student (name, age) VALUES
(?, ?)', [student.name, student.age]);
  });


  try {
    const results = await Promise.all(insertPromises);
    results.forEach((result, index) => {
      console.log(`Inserted: ${students[index].name}, Result:`, result);
    });
  } catch (error) {
        console.error('Error inserting records:', error);
    } finally {
        await connection.end();
    }
}

insertStudents();
```

Assignment Evaluation

0: Not Done		1: Incomplete		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date: **Practical In-charge**

Assignment No.8

Q.8) Create node.js server using node.js

```
 const http = require('http');
  const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello, World! This is a simple Node.js server.\n');
});

  const PORT = 3000;
  server.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

Assignment Evaluation

0: Not Done		1: Incomplete		2: Late Complete	
3: Need Improvement		4: Completed		5: Well Done	

Date:

Practical In-charge