Simple Web Server

9. Write a Node.js program to create a simple web server that responds with "Hello, Node.js!". Modify the server to respond with different messages based on the URL path (e.g., /about responds with "About Us", /contact responds with "Contact Us").

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

// Create an HTTP server
const server = http.createServer((req, res) => {
    // Set the response HTTP header with HTTP status and Content type
    res.writeHead(200, { 'Content-Type': 'text/plain' });
    // Get the URL path
    const path = req.url;
    // Respond with different messages based on the URL path
    if (path === '/') {
        res.end('Hello, Node.js!');
    } else if (path === '/about') {
        res.end('About Us');
    } else if (path === '/contact') {
        res.end('Contact Us');
    } else {
        res.writeHead(404, { 'Content-Type': 'text/plain' });
        res.end('404 Not Found');
    }
```

```
});

// The server listens on port 3000

server.listen(3000, () => {

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

});
```

File System Operations

10. Write Node.js program that reads a text file and prints its contents to the console. Then, extend the program to create a new file and write some data into it.

Step 1: Reading a Text File

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

const path = require('path');

// Specify the path to the text file

const filePath = path.join(__dirname, 'input.txt');

// Read the file

fs.readFile(filePath, 'utf8', (err, data) => {

   if (err) {

      console.error('Error reading the file:', err);

      return;

   }

   console.log('File contents:');

   console.log(data);

});
```

Step 2: Creating and Writing to a New File

```
const fs = require('fs');
// Create and write to a file
fs.writeFile('example.txt', 'Hello, World!', (err) => {
  if (err) throw err;
  console.log('File created and written to!');

  // Read the file
  fs.readFile('example.txt', 'utf8', (err, data) => {
    if (err) throw err;

    console.log('File content:', data);

    // Append to the file
    fs.appendFile('example.txt', ' How are you?', (err) => {
      if (err) throw err;
      console.log('File updated!');

      // Read the updated file
      fs.readFile('example.txt', 'utf8', (err, updatedData) => {
        if (err) throw err;
        console.log('Updated file content:', updatedData);
```

```
      // Delete the file
      fs.unlink('example.txt', (err) => {
        if (err) throw err;
        console.log('File deleted!');
      });
    });
  });
});
```

Database connection

11.Write a Node.js program using the mysql package to perform create database, create table, insert value, select from, update values, delete values on a mysql collection using student data

```
const mysql = require('mysql');
// Create a connection to the MySQL server
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'root'
});
// Connect to the MySQL server
```

```
connection.connect((err) => {
  if (err) {
    return console.error('Error connecting to the MySQL server:', err);
  }
  console.log('Connected to the MySQL server.');
  // Create a new database
  connection.query('CREATE DATABASE IF NOT EXISTS school', (err, result) => {
    if (err) throw err;
    console.log('Database created or already exists.');
    // Use the new database
    connection.query('USE school', (err, result) => {
      if (err) throw err;
      // Create a new table
      const createTableQuery = `
        CREATE TABLE IF NOT EXISTS students (
          id INT AUTO_INCREMENT PRIMARY KEY,
          name VARCHAR(255) NOT NULL,
          age INT NOT NULL,
          grade VARCHAR(10) NOT NULL
        )
      `;
      connection.query(createTableQuery, (err, result) => {
```

```
if (err) throw err;
console.log('Table created or already exists.');
// Insert values into the table
const insertQuery = `
  INSERT INTO students (name, age, grade)
  VALUES ('John Doe', 18, 'A'),
      ('Jane Smith', 20, 'B'),
      ('Alice Johnson', 19, 'A')
`;
connection.query(insertQuery, (err, result) => {
  if (err) throw err;
  console.log('Values inserted.');
  // Select values from the table
  connection.query('SELECT * FROM students', (err, results) => {
    if (err) throw err;
    console.log('Selected values:');
    console.log(results);

    // Update values in the table
    const updateQuery = `
      UPDATE students
      SET grade = 'A+'
```

```
    WHERE name = 'Jane Smith'
  `;
  connection.query(updateQuery, (err, result) => {
    if (err) throw err;
    console.log('Values updated.');

     //Delete values from the table
    const deleteQuery = `
      DELETE FROM students
      WHERE name = 'Alice Johnson'
    ;
    connection.query(deleteQuery, (err, result) => {
      if (err) throw err;
      console.log('Values deleted.');

      // Close the connection
      connection.end((err) => {
        if (err) throw err;
        console.log('Connection closed.');
      });
    });
  });
```

```
                    });
                });
            });
        });
    });
});
```