Lab-9

1. Create a module for area calculation in node.js for circle, rectangle, and square. Use this module in your current file to find area of all shape.

Area.js :-

module.exports = {

  circle: function (radius) {

    return Math.PI \* radius \* radius;

  },

  rectangle: function (length, width) {

    return length \* width;

  },

  square: function (side) {

    return side \* side;

  },

};

Node\_area.js :-

var http = require("http");

var url = require("url");

var area = require("./area");

http

  .createServer(function (req, res) {

    if (req.method === "GET") {

      var parsedUrl = url.parse(req.url, true);

      if (parsedUrl.pathname === "/") {

        res.writeHead(200, { "Content-Type": "text/html" });

        res.write(`

          <h1>Enter Dimensions to Calculate Areas</h1>

          <form action="/calculate" method="get">

            <label>Radius (for Circle):</label>

            <input type="number" step="any" name="radius" /><br><br>

            <label>Length (for Rectangle):</label>

            <input type="number" step="any" name="length" /><br><br>

            <label>Width (for Rectangle):</label>

            <input type="number" step="any" name="width" /><br><br>

            <label>Side (for Square):</label>

            <input type="number" step="any" name="side" /><br><br>

            <input type="submit" value="Calculate Areas">

          </form>

        `);

        res.end();

      } else if (parsedUrl.pathname === "/calculate") {

        var q = parsedUrl.query;

        var radius = q.radius ? parseFloat(q.radius) : 0;

        var length = q.length ? parseFloat(q.length) : 0;

        var width = q.width ? parseFloat(q.width) : 0;

        var side = q.side ? parseFloat(q.side) : 0;

        res.writeHead(200, { "Content-Type": "text/html" });

        res.write("<h1>Area Calculations</h1>");

        if (radius > 0) {

          var circleArea = area.circle(radius).toFixed(2);

          res.write(

            `<p>Area of Circle with radius ${radius}: ${circleArea}</p>`

          );

        }

        if (length > 0 && width > 0) {

          var rectangleArea = area.rectangle(length, width).toFixed(2);

          res.write(

            `<p>Area of Rectangle with length ${length} and width ${width}: ${rectangleArea}</p>`

          );

        }

        if (side > 0) {

          var squareArea = area.square(side).toFixed(2);

          res.write(`<p>Area of Square with side ${side}: ${squareArea}</p>`);

        }

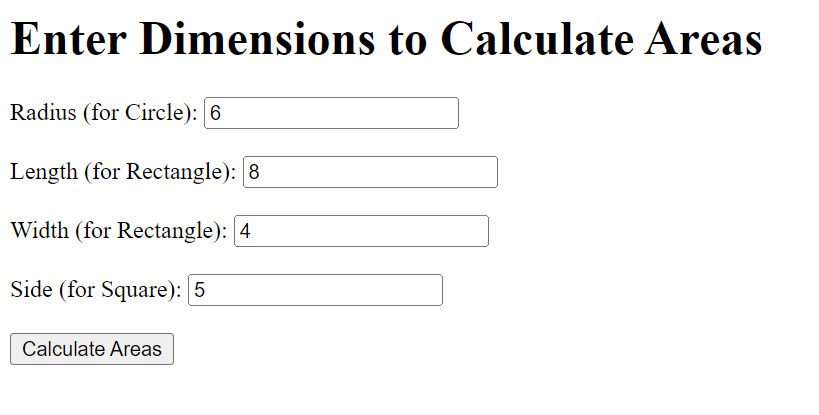
        res.end();

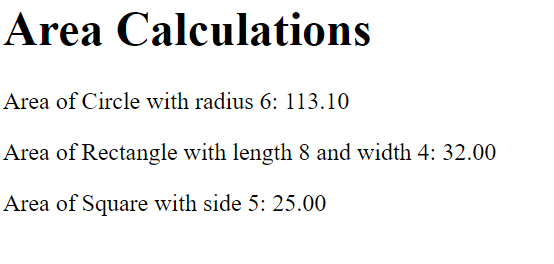
      }

    }

  })

  .listen(8080);





1. Write a program to read a file from server and check whether no read from file is armstrong or not using node.js

Armstrong.txt :-

153

370

371

9474

123

500

Node\_Armstrong.js :-

const http = require("http");

const fs = require("fs");

const path = require("path");

function isArmstrong(number) {

  const digits = number.toString().split("");

  const numDigits = digits.length;

  const sum = digits.reduce(

    (acc, digit) => acc + Math.pow(Number(digit), numDigits),

    0

  );

  return sum === number;

}

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

  if (req.method === "GET") {

    const filePath = path.join(\_\_dirname, "armstrong.txt");

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

      if (err) {

        res.writeHead(500, { "Content-Type": "text/html" });

        res.end("<h1>Error reading the file</h1>");

        return;

      }

      const numbers = data.split("\n").map((num) => Number(num.trim()));

      let results = "<h1>Armstrong Number Check</h1><ul>";

      numbers.forEach((number) => {

        if (isArmstrong(number)) {

          results += `<li>${number} is an Armstrong number.</li>`;

        } else {

          results += `<li>${number} is not an Armstrong number.</li>`;

        }

      });

      results += "</ul>";

      res.writeHead(200, { "Content-Type": "text/html" });

      res.end(results);

    });

  }

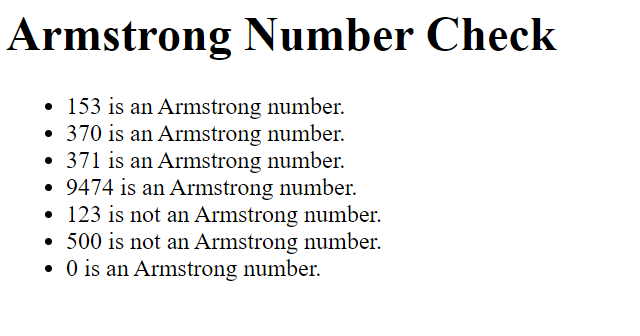
});

const PORT = 8080;

server.listen(PORT, () => {

  console.log(`Server is running at http://localhost:${PORT}`);

});



1. Write a program to read student.json file from a server and display the information of student for given PRN no in tabular format. JSON file contains information about student name, PRN, Branch, Semester, address, and subject name.

Student.json :-

[

  {

    "name": "Khushi Patel",

    "prn": "123456",

    "branch": "Computer Science",

    "semester": "5",

    "address": "OP Road, Vadodara",

    "subjects": [

      "Basic Web Programming",

      "Computer Organization",

      "Computer Graphics"

    ]

  },

  {

    "name": "Dhriti Modi",

    "prn": "789012",

    "branch": "Electrical Engineering",

    "semester": "3",

    "address": "Padra, Vadodara",

    "subjects": ["Circuit Theory", "Electromagnetics", "Signal Processing"]

  },

  {

    "name": "Manan Patel",

    "prn": "893172",

    "branch": "mechanical Engineering",

    "semester": "4",

    "address": "Anand",

    "subjects": ["Engg. Drawing", "Fitting", "Machine Design"]

  },

  {

    "name": "Hetvik Shah",

    "prn": "993013",

    "branch": "Chemical Engineering",

    "semester": "3",

    "address": "Akota, Vadodara",

    "subjects": ["Inorganic", "Plastic", "Hydrocarbons"]

  }

]

Node\_student.js :-

const http = require("http");

const fs = require("fs");

const url = require("url");

const PORT = 8080;

http

  .createServer((req, res) => {

*// Display the form for PRN input*

    if (req.method === "GET" && req.url === "/") {

      res.writeHead(200, { "Content-Type": "text/html" });

      res.write(`

        <h1>Student Information Lookup</h1>

        <form action="/student" method="get">

          <label for="prn">Enter PRN:</label>

          <input type="text" id="prn" name="prn" required />

          <input type="submit" value="Submit" />

        </form>

      `);

      res.end();

    }

*// Handle the student information request*

    else if (req.method === "GET" && req.url.startsWith("/student")) {

      const query = url.parse(req.url, true).query;

      const prn = query.prn;

      fs.readFile("student.json", "utf8", (err, data) => {

        if (err) {

          res.writeHead(500, { "Content-Type": "text/plain" });

          res.end("Error reading file.");

          return;

        }

        const students = JSON.parse(data);

        const student = students.find((s) => s.prn === prn);

        res.writeHead(200, { "Content-Type": "text/html" });

        if (student) {

          res.write(`

                    <h1>Student Information</h1>

                    <table border="1">

                        <tr>

                            <th>Name</th>

                            <th>PRN</th>

                            <th>Branch</th>

                            <th>Semester</th>

                            <th>Address</th>

                            <th>Subjects</th>

                        </tr>

                        <tr>

                            <td>${student.name}</td>

                            <td>${student.prn}</td>

                            <td>${student.branch}</td>

                            <td>${student.semester}</td>

                            <td>${student.address}</td>

                            <td>${student.subjects.join(", ")}</td>

                        </tr>

                    </table>

                `);

        } else {

          res.write(`

                    <h1>No Student Found</h1>

                    <p>No student found with PRN: ${prn}</p>

                `);

        }

        res.end();

      });

    } else {

      res.writeHead(404, { "Content-Type": "text/plain" });

      res.end("404 Not Found");

    }

  })

  .listen(PORT, () => {

    console.log(`Server is running on http://localhost:${PORT}`);

  });



