CNT Assignment 5

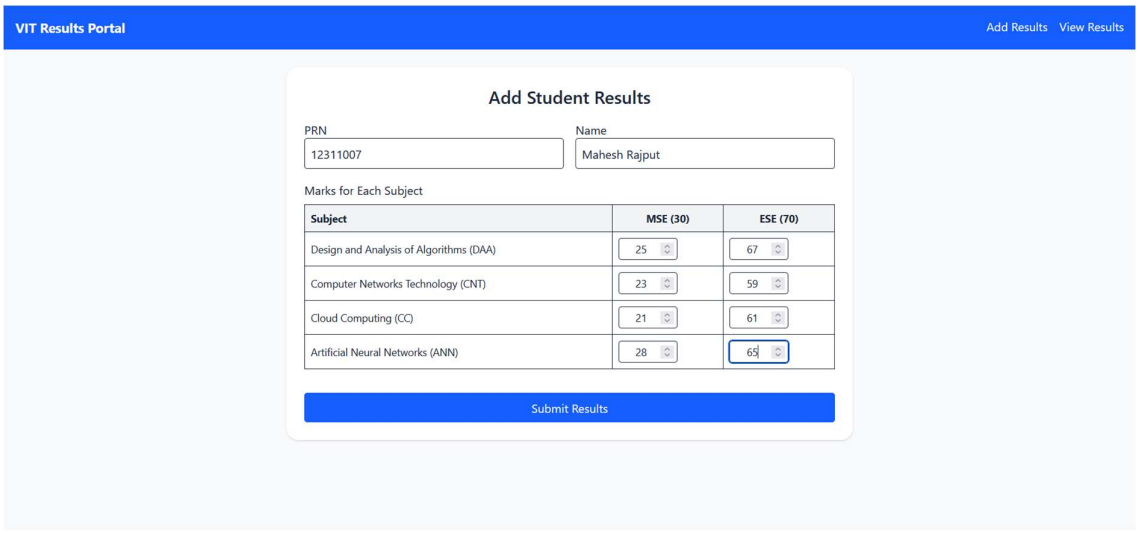
Name: Kuber Kishore **Class:** TY CSAI - A

Roll Number: 57 **Batch:** 3

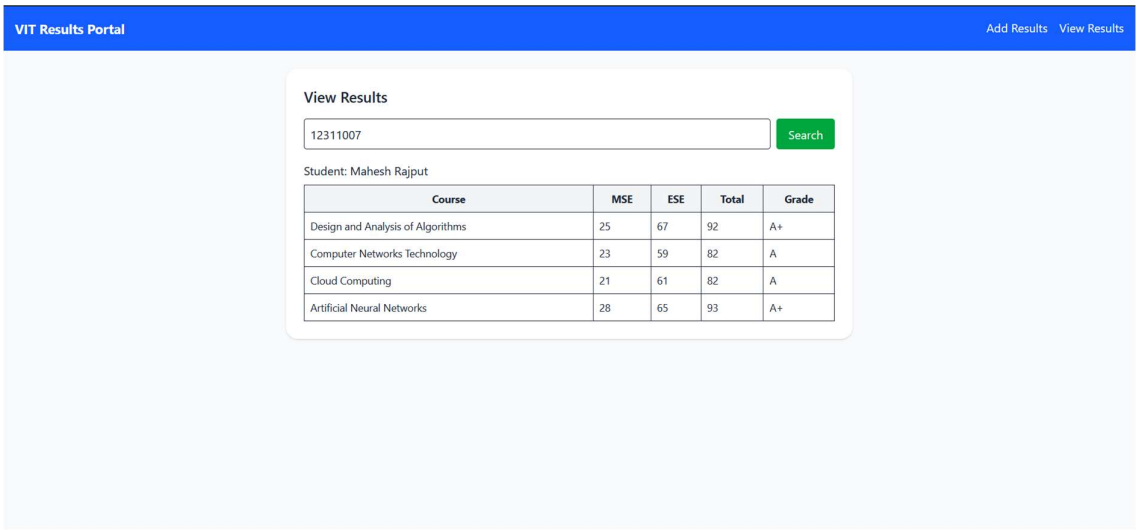
Assignment: Design and develop a responsive website to prepare one semester result of VIT students using REACT, Springboot and MySQL/ MongoDB/Oracle. Take any four subjects with MSE Marks (30%) ESE Marks (70%).

OUTPUT

Add Results Page



View Results Page



CODE

Backend

server.js

```
import express from "express";
import cors from "cors";
import dotenv from "dotenv";
import resultsRoutes from "./routes/results.js";

dotenv.config();

const app = express();
app.use(cors());
app.use(express.json());

app.use("/api", resultsRoutes);

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {
  console.log(`🚀 Server running on port ${PORT}`);
});
```

db.js

```
import mysql from "mysql2";
import dotenv from "dotenv";

dotenv.config();

const db = mysql.createConnection({
  host: process.env.DB_HOST,
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME,
});

db.connect((err) => {
  if (err) {
    console.error("❌ Database connection failed:", err.message);
  } else {
    console.log("✅ Connected to MySQL database");
  }
});

export default db;
```

routes/results.js

```
import express from "express";
import db from "../db.js";

const router = express.Router();

// 📘 Route 1: Insert marks for a student
router.post("/insert", (req, res) => {
  const { prn, name, marks } = req.body;

  if (!prn || !name || !marks) {
    return res.status(400).json({ error: "Missing required fields" });
  }

  // Insert student if not already present
  db.query(
    "INSERT IGNORE INTO student (prn, name) VALUES (?, ?)",
    [prn, name],
    (err) => {
      if (err) return res.status(500).json({ error: err.message });

      // Insert marks for each subject
      const queries = marks.map((m) => [
        prn,
        m.course_id,
        m.mse_marks,
        m.ese_marks,
      ]);

      db.query(
        "INSERT INTO marks (prn, course_id, mse_marks, ese_marks) VALUES ?",
        [queries],
        (err2) => {
          if (err2) return res.status(500).json({ error: err2.message });
          res.json({ message: "Marks inserted successfully" });
        }
      );
    }
  );
});

// 📊 Route 2: Get result by PRN
router.get("/results/:prn", (req, res) => {
  const prn = req.params.prn;

  const sql = `
    SELECT s.name, c.name AS course_name, m.mse_marks, m.ese_marks,
           (m.mse_marks + m.ese_marks) AS total,
```

```
          CASE
            WHEN (m.mse_marks + m.ese_marks) >= 91 THEN 'A+'
            WHEN (m.mse_marks + m.ese_marks) >= 81 THEN 'A'
            WHEN (m.mse_marks + m.ese_marks) >= 71 THEN 'B+'
            WHEN (m.mse_marks + m.ese_marks) >= 61 THEN 'B'
            WHEN (m.mse_marks + m.ese_marks) >= 51 THEN 'C'
            WHEN (m.mse_marks + m.ese_marks) >= 41 THEN 'D'
            ELSE 'F'
          END AS grade
    FROM marks m
    JOIN student s ON m.prn = s.prn
    JOIN course c ON m.course_id = c.id
    WHERE m.prn = ?
  `;

  db.query(sql, [prn], (err, results) => {
    if (err) return res.status(500).json({ error: err.message });
    if (results.length === 0)
      return res.status(404).json({ error: "No records found" });
    res.json(results);
  });
});

export default router;
```

FRONTEND

index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>frontend</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

src/main.jsx

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
```

```
import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

src/App.jsx

```
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-
dom";
import AddResults from "./pages/AddResults";
import ViewResults from "./pages/ViewResults";

function App() {
  return (
    <Router>
      <div className="min-h-screen bg-gray-50 text-gray-800">
        {/* Navbar */}
        <nav className="bg-blue-600 text-white p-4 flex justify-between">
          <h1 className="text-lg font-bold">VIT Results Portal</h1>
          <div className="space-x-4">
            <Link to="/add" className="hover:underline">
              Add Results
            </Link>
            <Link to="/view" className="hover:underline">
              View Results
            </Link>
          </div>
        </nav>

        {/* Routes */}
        <div className="p-6">
          <Routes>
            <Route path="/" element={<AddResults />} />
            <Route path="/add" element={<AddResults />} />
            <Route path="/view" element={<ViewResults />} />
          </Routes>
        </div>
      </div>
    </Router>
  );
}

export default App;
```

src/pages/AddResults.jsx

```jsx
import React, { useState } from "react";

const subjects = [
  { id: 1, name: "Design and Analysis of Algorithms (DAA)" },
  { id: 2, name: "Computer Networks Technology (CNT)" },
  { id: 3, name: "Cloud Computing (CC)" },
  { id: 4, name: "Artificial Neural Networks (ANN)" },
];

const AddResults = () => {
  const [prn, setPrn] = useState("");
  const [name, setName] = useState("");
  const [marks, setMarks] = useState(
    subjects.map((s) => ({ course_id: s.id, mse_marks: "", ese_marks: "" }))
  );
  const [message, setMessage] = useState("");

  const handleMarkChange = (index, field, value) => {
    const updated = [...marks];
    updated[index][field] = value;
    setMarks(updated);
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    setMessage("");

    try {
      const res = await fetch("http://localhost:5000/api/insert", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ prn, name, marks }),
      });

      const data = await res.json();
      if (res.ok) {
        setMessage("✅ Results added successfully!");
        setPrn("");
        setName("");
        setMarks(
          subjects.map((s) => ({
            course_id: s.id,
            mse_marks: "",
            ese_marks: "",
          }))
        );
      } else {
```

```
      setMessage(`❌ ${data.error}`);
    }
  } catch (err) {
    setMessage("❌ Server error");
  }
};

return (
  <div className="max-w-3xl mx-auto bg-white shadow p-6 rounded-2xl">
    <h2 className="text-2xl font-semibold mb-4 text-center">
      Add Student Results
    </h2>

    <form onSubmit={handleSubmit} className="space-y-4">
      {/* PRN + Name */}
      <div className="grid grid-cols-2 gap-4">
        <div>
          <label className="block font-medium">PRN</label>
          <input
            type="text"
            value={prn}
            onChange={(e) => setPrn(e.target.value)}
            className="border p-2 w-full rounded"
            required
          />
        </div>
        <div>
          <label className="block font-medium">Name</label>
          <input
            type="text"
            value={name}
            onChange={(e) => setName(e.target.value)}
            className="border p-2 w-full rounded"
            required
          />
        </div>
      </div>

      {/* Marks Table */}
      <div>
        <h3 className="font-medium mb-2">Marks for Each Subject</h3>
        <table className="w-full border text-sm">
          <thead className="bg-gray-100">
            <tr>
              <th className="border p-2 text-left">Subject</th>
              <th className="border p-2">MSE (30)</th>
              <th className="border p-2">ESE (70)</th>
            </tr>
```

```
          </thead>
          <tbody>
            {subjects.map((s, i) => (
              <tr key={s.id}>
                <td className="border p-2">{s.name}</td>
                <td className="border p-2">
                  <input
                    type="number"
                    placeholder="MSE"
                    value={marks[i].mse_marks}
                    onChange={(e) =>
                      handleMarkChange(i, "mse_marks", e.target.value)
                    }
                    className="border p-1 w-20 text-center rounded"
                    required
                    min="0"
                    max="30"
                  />
                </td>
                <td className="border p-2">
                  <input
                    type="number"
                    placeholder="ESE"
                    value={marks[i].ese_marks}
                    onChange={(e) =>
                      handleMarkChange(i, "ese_marks", e.target.value)
                    }
                    className="border p-1 w-20 text-center rounded"
                    required
                    min="0"
                    max="70"
                  />
                </td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>

      <button
        type="submit"
        className="bg-blue-600 text-white px-6 py-2 rounded hover:bg-blue-
700 w-full mt-4"
      >
        Submit Results
      </button>
    </form>
```

```
      {message && (
        <p className="mt-4 text-center font-medium">
          {message}
        </p>
      )}
    </div>
  );
};

export default AddResults;
```

src/pages/ViewResults.jsx

```
import React, { useState } from "react";

const ViewResults = () => {
  const [prn, setPrn] = useState("");
  const [results, setResults] = useState([]);
  const [error, setError] = useState("");

  const handleSearch = async () => {
    setError("");
    setResults([]);

    try {
      const res = await fetch(`http://localhost:5000/api/results/${prn}`);
      const data = await res.json();

      if (res.ok) {
        setResults(data);
      } else {
        setError(data.error || "No records found");
      }
    } catch {
      setError("Server error");
    }
  };

  return (
    <div className="max-w-3xl mx-auto bg-white shadow p-6 rounded-2xl">
      <h2 className="text-xl font-semibold mb-4">View Results</h2>

      <div className="flex gap-2 mb-4">
        <input
          type="text"
          placeholder="Enter PRN"
          value={prn}
          onChange={(e) => setPrn(e.target.value)}
```

```
          className="border p-2 rounded w-full"
        />
        <button
          onClick={handleSearch}
          className="bg-green-600 text-white px-4 py-2 rounded hover:bg-green-
700"
        >
          Search
        </button>
      </div>

      {error && <p className="text-red-600">{error}</p>}

      {results.length > 0 && (
        <div>
          <h3 className="font-medium mb-2">Student: {results[0].name}</h3>
          <table className="w-full border text-sm">
            <thead className="bg-gray-100">
              <tr>
                <th className="border p-2">Course</th>
                <th className="border p-2">MSE</th>
                <th className="border p-2">ESE</th>
                <th className="border p-2">Total</th>
                <th className="border p-2">Grade</th>
              </tr>
            </thead>
            <tbody>
              {results.map((r, i) => (
                <tr key={i}>
                  <td className="border p-2">{r.course_name}</td>
                  <td className="border p-2">{r.mse_marks}</td>
                  <td className="border p-2">{r.ese_marks}</td>
                  <td className="border p-2">{r.total}</td>
                  <td className="border p-2">{r.grade}</td>
                </tr>
              ))}
            </tbody>
          </table>
        </div>
      )}
    </div>
  );
};

export default ViewResults;
```