

Question 1

Assume that you are maintaining the MongoDB collection named 'Service_Booking' with fields like Service ID, Service Description, Customer Name, Email ID, Mobile Number, and Vehicle Number. You are requested to develop a web application in which implement pattern search based on Vehicle Number, let's say searching for the vehicles with TN23 registration and display the complete result in HTML table format. Apply jQuery as mentioned below:

- Once the form element got its focus, its background should be in cyan color.
- Once any control loses its focus, change its background to light pink.

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Vehicle Search</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <script>
    $(document).ready(function(){
      $("input").focus(function(){
        $(this).css("background-color", "cyan")
      }).blur(function(){
        $(this).css("background-color", "lightpink")
      })

      $.get("/search", function(data) {
        let tableBody = $("#results tbody")
        tableBody.empty()
        data.forEach(service => {
          let row = `<tr>
            <td>${service.serviceID}</td>
            <td>${service.description}</td>
            <td>${service.customerName}</td>
            <td>${service.email}</td>
            <td>${service.mobile}</td>
            <td>${service.vehicleNumber}</td>
```

```

        </tr>`
        tableBody.append(row)
    })
}
})
</script>
</head>
<body>
    <div class="container mt-4">
        <h2 class="text-center">Search Results for TN23 Vehicles</h2>
        <table id="results" class="table table-bordered">
            <thead>
                <tr>
                    <th>Service ID</th>
                    <th>Description</th>
                    <th>Customer Name</th>
                    <th>Email</th>
                    <th>Mobile</th>
                    <th>Vehicle Number</th>
                </tr>
            </thead>
            <tbody></tbody>
        </table>
    </div>
</body>
</html>

```

Server.js

```

const express = require("express")
const mongoose = require("mongoose")
const path = require("path")

const app = express()
app.use(express.static(path.join(__dirname, "public")))

mongoose.connect("mongodb://localhost:27017/serviceDB", { useNewUrlParser: true,
useUnifiedTopology: true })

const serviceSchema = new mongoose.Schema({
    serviceID: String,
    description: String,

```

```

    customerName: String,
    email: String,
    mobile: String,
    vehicleNumber: String
  })

const Service = mongoose.model("Service_Booking", serviceSchema)

app.get("/search", async (req, res) => {
  const results = await Service.find({ vehicleNumber: /^TN23/ })
  res.json(results)
})

app.listen(3000, () => console.log("Server running on port 3000"))

```

Question 2

Develop a React JS Controlled Form component called HR to capture the Employee ID, Employee Name, Basic Pay (BP), Allowances (A), and Deductions (D). Ensure Gross Pay ($GP=BP+A$) and Net Pay ($NP=GP-D$) are auto calculated and displayed in non-editable text box. On successful submission display the details in Table view.

App.jsx

```

import React, { useState } from "react"

function App() {
  const [formData, setFormData] = useState({
    empID: "",
    empName: "",
    basicPay: "",
    allowances: "",
    deductions: "",
    grossPay: 0,
    netPay: 0
  })
  const [submittedData, setSubmittedData] = useState(null)

  const handleChange = (e) => {
    const { name, value } = e.target
    const updatedData = { ...formData, [name]: value }
  }

```

```

        updatedData.grossPay = parseFloat(updatedData.basicPay || 0) +
parseFloat(updatedData.allowances || 0)
        updatedData.netPay = updatedData.grossPay - parseFloat(updatedData.deductions || 0)
        setFormData(updatedData)
    }

    const handleSubmit = (e) => {
        e.preventDefault()
        setSubmittedData(formData)
    }

    return (
        <div className="container">
            <h2 className="text-center">HR Employee Form</h2>
            <form onSubmit={handleSubmit}>
                <input type="text" name="empID" value={formData.empID}
onChange={handleChange} placeholder="Employee ID" required />
                <input type="text" name="empName" value={formData.empName}
onChange={handleChange} placeholder="Employee Name" required />
                <input type="number" name="basicPay" value={formData.basicPay}
onChange={handleChange} placeholder="Basic Pay" required />
                <input type="number" name="allowances" value={formData.allowances}
onChange={handleChange} placeholder="Allowances" required />
                <input type="number" name="deductions" value={formData.deductions}
onChange={handleChange} placeholder="Deductions" required />
                <input type="number" name="grossPay" value={formData.grossPay} readOnly
placeholder="Gross Pay" />
                <input type="number" name="netPay" value={formData.netPay} readOnly
placeholder="Net Pay" />
                <button type="submit">Submit</button>
            </form>

            {submittedData && (
                <table border="1">
                    <thead>
                        <tr>
                            <th>Employee ID</th>
                            <th>Name</th>
                            <th>Basic Pay</th>
                            <th>Allowances</th>
                            <th>Deductions</th>
                            <th>Gross Pay</th>
                            <th>Net Pay</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>{submittedData.empID}</td>
                            <td>{submittedData.empName}</td>
                            <td>{submittedData.basicPay}</td>
                            <td>{submittedData.allowances}</td>
                            <td>{submittedData.deductions}</td>
                            <td>{submittedData.grossPay}</td>
                            <td>{submittedData.netPay}</td>
                        </tr>
                    </tbody>
                </table>
            )}
        </div>
    )

```

```
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>{submittedData.empID}</td>
          <td>{submittedData.empName}</td>
          <td>{submittedData.basicPay}</td>
          <td>{submittedData.allowances}</td>
          <td>{submittedData.deductions}</td>
          <td>{submittedData.grossPay}</td>
          <td>{submittedData.netPay}</td>
        </tr>
      </tbody>
    </table>
  )}
</div>
}
```

```
export default App
```