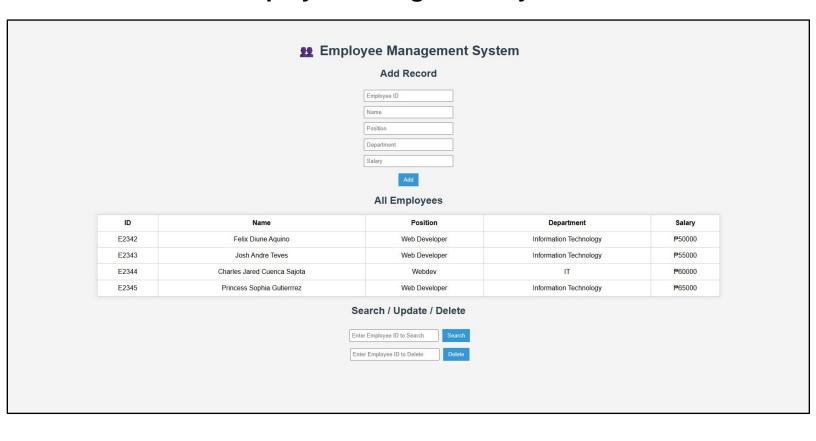
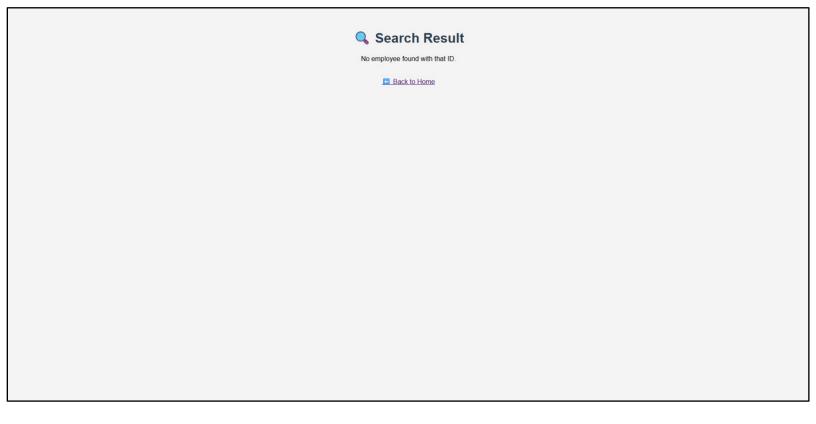
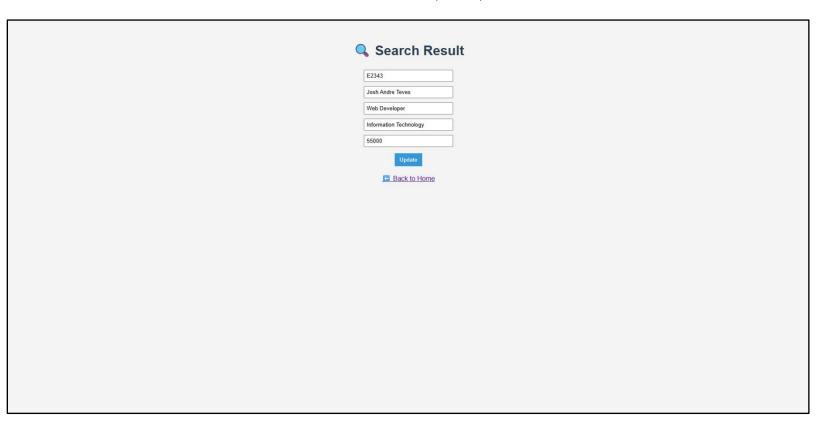
Employee Management System



Search Screen (Not Found)



Search Screen (Found)



Server.js

```
BIT35 > Aquino > EA6 > JS server.js > ..
       const express = require('express');
       const { MongoClient, ObjectId } = require('mongodb');
      const bodyParser = require('body-parser');
      const path = require('path');
      const app = express();
      const port = 3000;
      const mongoUrl = 'mongodb://127.0.0.1:27017';
const dbName = 'ea6db';
11
      // Middleware
12
      app.use(bodyParser.urlencoded({ extended: true }));
      app.use(express.static(path.join(_dirname, 'public')));
app.set('view engine', 'ejs');
app.set('views', path.join(_dirname, 'views'));
14
15
17
      // Connect to MongoDB
18
      MongoClient.connect(mongoUrl).then(client => {
20
       db = client.db(dbName);
      console.log('▼ Connected to MongoDB');
}).catch(err => console.error('★ DB Connection Error:', err));
21
23
      // 🏠 Home page - Add employee form and list
24
      app.get('/', async (req, res) => {
        const employees = await db.collection('employees').find().toArray();
res.render('index', { employees });
26
27
      });
29
      // + Add new employee
      app.post('/add', async (req, res) => {
       const { empid, empname, position, department, salary } = req.body;
await db.collection('employees').insertOne({ empid, empname, position, department, salary: parseFloat(salary) });
32
33
         res.redirect('/');
35
36
      });
      // 🔍 Search by empid
      app.post('/search', async (req, res) => {
  const { empid } = req.body;
  const employee = await db.collection('employees').findOne({ empid });
38
41
         res.render('search', { employee });
42
      });
44
       // 🖸 Update employee
45
      app.post('/update', async (req, res) => {
  const { empid, empname, position, department, salary } = req.body;
47
         await db.collection('employees').updateOne(
           { empid },
```

Server.js (cont.)

```
await db.collection('employees').updateOne(
47
          { empid },
48
49
         { $set: { empname, position, department, salary: parseFloat(salary) } }
50
       );
51
      res.redirect('/');
52
     });
53
     // W Delete employee
54
     app.post('/delete', async (req, res) => {
55
       const { empid } = req.body;
56
       await db.collection('employees').deleteOne({ empid });
57
       res.redirect('/');
58
59
     });
60
     app.listen(port, () => {
61
      console.log(` \( \varphi \) App running at http://localhost:\( \) (port\);
62
     });
63
64
```

Index.ejs

```
D: > BIT35 > Aquino > EA6 > views > 💠 index.ejs > 🔗 html > 😭 body > 😭 form > 🤡 br
     <!DOCTYPE html>
     <html>
     <head>
       <title>EA6 Employee Management</title>
       <link rel="stylesheet" href="/style.css">
      </head>
      <body>
       <form action="/add" method="POST">
12
         <input name="empid" placeholder="Employee ID" required><br>
         <input name="empname" placeholder="Name" required><br>
<input name="position" placeholder="Position" required><br>
<input name="department" placeholder="Department" required><br>
13
15
         <input name="salary" placeholder="Salary" required><br>
17
         <button type="submit">Add</button>
18
       </form>
19
20
       <h2>All Employees</h2>
       22
23
           IDNamePositionDepartmentSalary
24
25
         <% employees.forEach(emp => { %>
           27
            <</td>
28
             <%= emp.empname %>
29
            <</td>
30
             <%= emp.department %>
31
             P<%= emp.salary %>
           33
         <% }) %>
       35
       <h2>Search / Update / Delete</h2>
       <form action="/search" method="POST">
         <input name="empid" placeholder="Enter Employee ID to Search" required>
39
         <button type="submit">Search</button>
40
       </form>
41
       <form action="/delete" method="POST">
        <input name="empid" placeholder="Enter Employee ID to Delete" required>
         <button type="submit">Delete</button>
       </form>
45
     </body>
     </html>
```

Search.ejs

```
<!DOCTYPE html>
1
2
     <html>
3
     <head>
      <title>Search Result - EA6</title>
4
5
      <link rel="stylesheet" href="/style.css">
6
7
     <body>
8
      9
      <% if (employee) { %>
10
        <form action="/update" method="POST">
11
          <input name="empid" value="<%= employee.empid %>" readonly><br>
12
          <input name="empname" value="<%= employee.empname %>" required><br>
13
14
          <input name="position" value="<%= employee.position %>" required><br>
15
          <input name="department" value="<%= employee.department %>" required><br>
          <input name="salary" value="<%= employee.salary %>" required><br>
16
17
          <button type="submit">Update</button>
18
        </form>
19
      <% } else { %>
20
       No employee found with that ID.
21
      <% } %>
22
23
      24
    </body>
25
    </html>
26
```

CSS

```
1
     body {
 2
       font-family: Arial, sans-serif;
3
       text-align: center;
4
       background-color: □#f4f4f4;
5
      padding: 20px;
6
8
     input {
       margin: 5px;
10
       padding: 5px;
11
      width: 200px;
12
13
14
     button {
       padding: 8px 12px;
15
       margin-top: 10px;
16
       background-color: ■#3498db;
17
       color: □white;
18
       border: none;
19
20
       cursor: pointer;
21
22
23
     table {
      margin: 20px auto;
24
25
       border-collapse: collapse;
      width: 80%;
27
       background-color: □white;
28
     box-shadow: 0 0 10px □rgba(0,0,0,0.1);
29
30
31
     th, td {
32
     padding: 10px;
33
      border: 1px solid □#ccc;
34
35
36
     h1, h2 {
     color: ■#2c3e50;
37
38
```