I'll give you a **full CRUD example** using:-

- Express.js + Mongoose
- HTML forms for Create and Update
- EJS templates for rendering pages (list records, form pages)

Step 1: Setup your project

Make sure you have:

```
npm install express mongoose body-parser ejs method-override
```

Note: `method-override` allows HTML forms to send PUT and DELETE requests.

Step 2: Project structure (basic)

```
/express-js
  /views
    index.ejs
    new.ejs
    edit.ejs
  app.js
```

Step 3: Full `app.js` code

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

```
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const methodOverride = require('method-override');
const app = express();

mongoose.connect('mongodb://localhost:27017/mydb', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

const recordSchema = new mongoose.Schema({
  name: String,
  age: Number,
  email: String,
  isActive: Boolean,
});

const Record = mongoose.model('Record', recordSchema);

// Middleware
app.set('view engine', 'ejs');
app.use(bodyParser.urlencoded({ extended: true }));
app.use(methodOverride('_method')); // support ?_method=PUT and DELETE from
forms

// --- ROUTES ---

// INDEX - List all records
app.get('/records', async (req, res) => {
  const records = await Record.find();
  res.render('index', { records });
});

// NEW - Show form to create new record
app.get('/records/new', (req, res) => {
  res.render('new');
});

// CREATE - Add new record to DB
app.post('/records', async (req, res) => {
  const { name, age, email, isActive } = req.body;
  await Record.create({
    name,
    age: Number(age),
    email,
    isActive: isActive === 'on',
  });
  res.redirect('/records');
});

// EDIT - Show form to edit existing record
app.get('/records/:id/edit', async (req, res) => {
  const record = await Record.findById(req.params.id);
  if (!record) return res.redirect('/records');
  res.render('edit', { record });
});
```

```
// UPDATE - Update record in DB
app.put('/records/:id', async (req, res) => {
  const { name, age, email, isActive } = req.body;
  await Record.findByIdAndUpdate(req.params.id, {
    name,
    age: Number(age),
    email,
    isActive: isActive === 'on',
  });
  res.redirect('/records');
});

// DELETE - Remove record from DB
app.delete('/records/:id', async (req, res) => {
  await Record.findByIdAndDelete(req.params.id);
  res.redirect('/records');
});

// Redirect root to /records
app.get('/', (req, res) => {
  res.redirect('/records');
});

app.listen(3000, () => {
  console.log('Server running at http://localhost:3000');
});
```

Step 4: Create EJS templates

---

views/index.ejs

```
<!DOCTYPE html>
<html>
<head>
  <title>Records List</title>
</head>
<body>
  <h1>All Records</h1>
  <a href="/records/new">Add New Record</a>
  <table border="1" cellpadding="5" cellspacing="0">
    <thead>
      <tr>

<th>Name</th><th>Age</th><th>Email</th><th>Active</th><th>Actions</th>
      </tr>
    </thead>
    <tbody>
      <% records.forEach(record => { %>
        <tr>
          <td><%= record.name %></td>
          <td><%= record.age %></td>
          <td><%= record.email %></td>
```

```
          <td><%= record.isActive ? 'Yes' : 'No' %></td>
          <td>
            <a href="/records/<%= record._id %>/edit">Edit</a>
            <form action="/records/<%= record._id %>?_method=DELETE"
method="POST" style="display:inline;">
              <button type="submit" onclick="return confirm('Delete this
record?')">Delete</button>
            </form>
          </td>
        </tr>
      <% }) %>
    </tbody>
  </table>
</body>
</html>
```

---

views/new.ejs

```
<!DOCTYPE html>
<html>
<head>
  <title>New Record</title>
</head>
<body>
  <h1>Create New Record</h1>
  <form action="/records" method="POST">
    <p>
      <label>Name:</label><br />
      <input type="text" name="name" required />
    </p>
    <p>
      <label>Age:</label><br />
      <input type="number" name="age" required />
    </p>
    <p>
      <label>Email:</label><br />
      <input type="email" name="email" required />
    </p>
    <p>
      <label>Active:</label>
      <input type="checkbox" name="isActive" />
    </p>
    <button type="submit">Create</button>
  </form>
  <a href="/records">Back to list</a>
</body>
</html>
```

---

views/edit.ejs

```
<!DOCTYPE html>
<html>
<head>
```

```
  <title>Edit Record</title>
</head>
<body>
  <h1>Edit Record</h1>
  <form action="/records/<%= record._id %>?_method=PUT" method="POST">
    <p>
      <label>Name:</label><br />
      <input type="text" name="name" value="<%= record.name %>" required />
    </p>
    <p>
      <label>Age:</label><br />
      <input type="number" name="age" value="<%= record.age %>" required />
    </p>
    <p>
      <label>Email:</label><br />
      <input type="email" name="email" value="<%= record.email %>" required
/>
    </p>
    <p>
      <label>Active:</label>
      <input type="checkbox" name="isActive" <%= record.isActive ? 'checked'
: '' %> />
    </p>
    <button type="submit">Update</button>
  </form>
  <a href="/records">Back to list</a>
</body>
</html>
```

Step 5: Run the app

```
npm install method-override
node app.js
```