Use node.js, mongoose database, html and visual studio code. Database has device serial number, temperature, humidity and time date. Use visual studio code. post serial number, temperature, humidity and time date and get serial number, temperature, humidity and time date from server and database. result is shown as table in html. row is shown 10 by 10 rows.

**ChatGPT said:**

ChatGPT

Here’s a step-by-step guide to achieve your goal:

**1. Setting up the Project**

1. Install Node.js and Visual Studio Code if not already installed.
2. Create a project folder, e.g., DeviceDataApp.
3. Open the terminal in Visual Studio Code, and initialize a Node.js project:

bash

Copy code

npm init -y

1. Install the necessary dependencies:

bash

Copy code

npm install express mongoose body-parser ejs

**2. Setting up the Backend**

**File: server.js**

javascript

Copy code

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const app = express();

const PORT = 3000;

// Middleware

app.use(bodyParser.urlencoded({ extended: true }));

app.use(bodyParser.json());

app.set('view engine', 'ejs');

app.use(express.static('public'));

// Connect to MongoDB

mongoose.connect('mongodb://127.0.0.1:27017/deviceDB', { useNewUrlParser: true, useUnifiedTopology: true })

.then(() => console.log('Connected to MongoDB'))

.catch(err => console.error(err));

// Define Mongoose Schema and Model

const deviceSchema = new mongoose.Schema({

serialNumber: String,

temperature: Number,

humidity: Number,

dateTime: Date

});

const Device = mongoose.model('Device', deviceSchema);

// Routes

// Render HTML table

app.get('/', async (req, res) => {

const page = parseInt(req.query.page) || 1;

const limit = 10;

const skip = (page - 1) \* limit;

const devices = await Device.find().sort({ dateTime: -1 }).skip(skip).limit(limit);

const total = await Device.countDocuments();

res.render('index', {

devices,

totalPages: Math.ceil(total / limit),

currentPage: page

});

});

// POST data to the database

app.post('/devices', async (req, res) => {

const { serialNumber, temperature, humidity, dateTime } = req.body;

const device = new Device({ serialNumber, temperature, humidity, dateTime: new Date(dateTime) });

await device.save();

res.send({ message: 'Device data saved successfully' });

});

// Start Server

app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));

**3. Setting up HTML and CSS**

**File: views/index.ejs**

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Device Data</title>

<style>

table {

width: 100%;

border-collapse: collapse;

}

table, th, td {

border: 1px solid black;

}

th, td {

padding: 10px;

text-align: left;

}

.pagination {

margin: 20px 0;

}

.pagination a {

margin: 0 5px;

text-decoration: none;

}

</style>

</head>

<body>

<h1>Device Data</h1>

<table>

<thead>

<tr>

<th>Serial Number</th>

<th>Temperature</th>

<th>Humidity</th>

<th>Date & Time</th>

</tr>

</thead>

<tbody>

<% devices.forEach(device => { %>

<tr>

<td><%= device.serialNumber %></td>

<td><%= device.temperature %></td>

<td><%= device.humidity %></td>

<td><%= new Date(device.dateTime).toLocaleString() %></td>

</tr>

<% }); %>

</tbody>

</table>

<div class="pagination">

<% for (let i = 1; i <= totalPages; i++) { %>

<a href="/?page=<%= i %>" <%= i === currentPage ? 'style="font-weight:bold"' : '' %>>Page <%= i %></a>

<% } %>

</div>

</body>

</html>

**4. Testing the Project**

1. **Start MongoDB**: Ensure your MongoDB server is running.
2. **Run the server**:

bash

Copy code

node server.js

1. Open your browser and go to http://localhost:3000.
2. Use tools like Postman or a custom frontend to send POST requests to the /devices endpoint.

**5. Example POST Data**

To add device data, send a POST request to http://localhost:3000/devices with a JSON body like this:

json

Copy code

{

"serialNumber": "SN12345",

"temperature": 25.6,

"humidity": 60,

"dateTime": "2025-01-01T10:00:00Z"

}

Now you can view the data in a paginated table and add new records via POST requests!

4o

**You said:**

how to see HTML

**ChatGPT said:**

ChatGPT

To see the HTML file in action, you need to ensure the server is running and the correct setup is followed. Here's how you can see the HTML:

**Steps to View the HTML**

1. **Start the Server**
   * Open the terminal in your project folder.
   * Run the command:

bash

Copy code

node server.js

* + The server will start running on http://localhost:3000.

1. **Open a Browser**
   * Open your favorite browser (e.g., Chrome, Edge, or Firefox).
   * Type http://localhost:3000 in the address bar and press **Enter**.
2. **View the Table**
   * You should see the paginated table showing the device data fetched from the MongoDB database.
3. **Test Pagination**
   * If you have more than 10 records in the database, pagination links (e.g., "Page 1, Page 2") will appear below the table.
   * Click the links to navigate through the data.

**How to Add Data**

If there’s no data in the table initially, add some records to the database. Here's how:

**Option 1: Using Postman**

1. Install Postman.
2. Open Postman and create a new POST request to http://localhost:3000/devices.
3. Add a JSON body like this:

json

Copy code

{

"serialNumber": "SN98765",

"temperature": 22.5,

"humidity": 55,

"dateTime": "2025-01-03T15:30:00Z"

}

1. Click **Send**. You’ll get a confirmation response if successful.

**Option 2: Using MongoDB Compass**

1. Open MongoDB Compass and connect to your MongoDB instance.
2. Navigate to the deviceDB database and devices collection.
3. Add new documents manually.

**View the HTML in Development**

If you want to see the raw HTML:

1. **Open the Developer Tools**
   * Right-click anywhere on the page in the browser and select **Inspect** (or press F12).
   * Go to the **Elements** tab.
2. **View Source**
   * Right-click the page and select **View Page Source** to see the rendered HTML as sent by the server.

Let me know if you encounter any issues!

4o

**You said:**

how to put it in internet free

**ChatGPT said:**

ChatGPT

If you want to make your project accessible on the internet for free, you can use platforms and services that offer free hosting for web applications. Here’s how you can do it: