HTML**(index.html):**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<title>File Upload</title>

</head>

<body>

<h1>File Upload</h1>

<input type="file" id="fileInput" />

<button onclick="uploadFile()">Upload</button>

<script src="app.js"></script>

</body>

</html>

**JavaScript (app.js):**

function uploadFile() {

const fileInput = document.getElementById('fileInput');

const file = fileInput.files[0];

if (file) {

// Use FormData to send the file to the server

const formData = new FormData();

formData.append('file', file);

// Use fetch to send the file to the server

fetch('/upload', {

method: 'POST',

body: formData

})

.then(response => response.json())

.then(data => {

// Handle the response from the server

console.log('Server response:', data);

})

.catch(error => console.error('Error:', error));

} else {

alert('Please select a file before uploading.');

}

}

**Node.js (server.js):**

const express = require('express');

const multer = require('multer');

const mongoose = require('mongoose');

const app = express();

const port = 3000;

// Connect to MongoDB (make sure to install and run MongoDB locally or use a cloud-based service)

mongoose.connect('mongodb://localhost/file-upload', { useNewUrlParser: true, useUnifiedTopology: true });

// Define a schema for storing file information

const fileSchema = new mongoose.Schema({

filename: String,

originalname: String,

mimetype: String,

size: Number,

timestamp: { type: Date, default: Date.now }

});

const File = mongoose.model('File', fileSchema);

// Set up Multer for handling file uploads

const storage = multer.memoryStorage();

const upload = multer({ storage: storage });

// Handle file uploads

app.post('/upload', upload.single('file'), (req, res) => {

const file = req.file;

// Store file information in the database

const newFile = new File({

filename: file.fieldname,

originalname: file.originalname,

mimetype: file.mimetype,

size: file.size

});

newFile.save((err, savedFile) => {

if (err) {

console.error('Error saving file information to the database:', err);

res.status(500).json({ error: 'Internal Server Error' });

} else {

// Log the file information to the console

console.log('File information:', savedFile);

// Send a response to the front-end

res.json({ message: 'File uploaded successfully', file: savedFile });

}

});

});

app.listen(port, () => {

console.log(`Server is running at http://localhost:${port}`);

});

Bash:

npm install express multer mongoose

Explain a simple javascript front end programm or algorithm that allows users to uploads files from the front end, stores the information of the file uploaded, and other decisions can be made with the data or files uploaded. It also stores these data in a database, and log it to a database