MERN STACK CRUD OPERATION

Step 1: npm init -y
Step2: npm i express mongodb mongoose
Step 3: npm i nodemon
Step 4: Create a file index.js (Outside)
Step 5: Write the code in package.json
"serve": "node index.js",
"dev": "nodemon index.js"

```
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "serve": "node index.js",
    "dev": "nodemon index.js"
  },
```

Step 6: Create 3 files (Outside) Model, controller, routes
Step 7: Create js file inside it [internship.model.js | internship.controller.js | internship.routes.js]

Step 8: Write the code in index.js file

```
// Router is used for sending data from one to another usin API

const express = require("express"); // for connecting HTTP requests &
response
const mongoose = require("mongoose"); // for connecting Mongo DB
const routeapi = require("./routes/internship.routes.js");

const app = express(); // for storing in object
app.use(express.json()); //Data in JSON format

app.use('/api/users',routeapi) // routes are defined here API calling here



// Connecting to MongoDB server using Mongoose
mongoose
  .connect("mongodb://localhost:27017/internship") // Mongo db la poi edu
da
  .then(() => {
    console.log("Connected to the database");
    app.listen(3002, () => {
      // port no settuping
      console.log("Server is running at port 3002");
    });
```

```
  })
  .catch(() => {
    // (err)
    console.log("Not connected to MongoDB");
  });
app.get("/", (req, res) => { // For getting data using API
  res.send("the Node server is running,API");
});
```

Step 9: Write the code in Model/internship.model.js file

```
const { default: mongoose } = require("mongoose");
const mangoose=require("mongoose"); // MONGO DB connection
const internSchema = mongoose.Schema({ // Mongo Table creation
    name: {
        type: String,
        required: true,
    },
    email: {
        type: String,
        required: true,
    },
    stack: {
        type: String,
        required: true,
    },
    contact: {
        type: Number,
        required: false,
    }
},
{
    timestamps : true,    // It provides createdAt and updatedAt as
}
);
const lavaagni = mongoose.model('intern',internSchema) // Table name &
stored in a object "lavaagni"
module.exports = lavaagni; // Exporting the table
```

Step 10: Write the code in Controller/internship.controller.js file

```
const lavaagni = require('../model/internship.model.js'); // Imported form
internship.model.js file
const getUser = async(req,res)=>{ // Get user function
    try {
        const user = await lavaagni.find({}); // Get user
        res.status(200).json(user) // 200 for success
    } catch (error) {
        res.status(500).json({message:error.message}); // 500 for error
    }
}

const  postUser = async (req,res)=> { // Post User Function
    try {
        const user = await  lavaagni.create(req.body); // Create a new
user from the request body
        res.status(200).json(user); // 200
    } catch (error) {
        res.status(500).json({message:error.message}); // 500 for error
    }
}

const updateuser = async (req,res) =>{
    try {
        const {id}= req.params; // Id of the user which we want to update
| id ahh parameter la send pandrom
        const user = await  lavaagni.findByIdAndUpdate(id,req.body); //
Create a new user from the request body // (id,req.body)
        const users = await lavaagni.findById(id); // Get user
        res.status(200).json(users) // 200 for success
    } catch (error) {
        res.status(500).json({message:error.message}); // 500 for error
    }
}

const deleteuser = async (req,res) =>{
    try {
        const {id}= req.params; // Id of the user which we want to update
| id ahh parameter la send pandrom
```

```
        const user = await  lavaagni.findByIdAndDelete(id); // Create a
new user from the request body
        //const users = await lavaagni.findById(id); // Get user
        res.status(200).json("User Deleted") // 200 for success
    } catch (error) {
        res.status(500).json({message:error.message}); // 500 for error
    }
}

module.exports = { //For exporting functions
    getUser,
    postUser,
    updateuser,
    deleteuser
}
```

Step 11: Write the code in Routes/internship.routes.js file

```
const express=require("express") // Importing Express
const router=express.Router(); // Router sending API
const{getUser, postUser, updateuser, deleteuser} =
require("../controller/internship.controller.js");// Call the controller
function to get user data from database

router.get('/',getUser) // GET uSER
router.post('/',postUser) // Create USER
router.put('/:id',updateuser) // Update USER
router.delete('/:id',deleteuser) // Delete USER


module.exports=router;
```

Step 12: Also have some work in Postman and MongoDb Compass software