**CRUD APPLICATION**

npm init -y

npm install express mongoose body-parser

const express = require("express");

const mongoose = require("mongoose");

const app = express();

app.use(express.json());

mongoose.connect("mongodb://127.0.0.1:27017/mydatabase", {

  useNewUrlParser: true,

  useUnifiedTopology: true

})

  .then(() => console.log("MongoDB Connected"))

  .catch(err => console.error("MongoDB Connection Error:", err));

const User = mongoose.model("User", new mongoose.Schema({ name: String, email: String, age: Number }));

app.post("/adduser", async (req, res) => {

  const user = new User(req.query);

  await user.save();

  res.send(user);

});

app.get("/users", async (req, res) => res.send(await User.find()));

app.get("/users/:id", async (req, res) => res.send(await User.findById(req.params.id)));

app.patch("/updateuser", async (req, res) => {

  const updatedUser = await User.findByIdAndUpdate(req.query.id, req.query, { new: true });

  res.send(updatedUser);

});

app.delete("/deleteuser", async (req, res) => {

  await User.findByIdAndDelete(req.query.id);

  res.send({ message: "User deleted" });

});

app.listen(5000, () => console.log("Server running on port 5000"));

**API Endpoints & Methods**

|  |  |  |
| --- | --- | --- |
| Action | Method | URL |
| Create User | POST | http://localhost:5000/adduser?name=John&email=john@mail.com&age=25 |
| Get All Users | GET | http://localhost:5000/users |
| Get User by ID | GET | http://localhost:5000/users/<id> |
| Update User | PATCH | http://localhost:5000/updateuser?id=<id>&email=new@mail.com |
| Delete User | GET | http://localhost:5000/deleteuser?id={id} |

Here are **5 MongoDB queries** and **5 API queries** based on your **Express + Mongoose** setup.

### ****1. Find Users Older Than 30****

db.users.find({ age: { $gt: 30 } });

### ****2. Update Email of a Specific User****

(Replace *{user\_id}* with the actual ID)

db.users.updateOne({ \_id: ObjectId("{user\_id}") }, { $set: { email: "newemail@example.com" } });

### ****3. Delete Users Younger Than 25****

db.users.deleteMany({ age: { $lt: 25 } });

### ****4. Count Number of Users in the Database****

db.users.countDocuments();

### ****5. Retrieve Users Sorted by Age (Descending Order)****

db.users.find().sort({ age: -1 });

### ****1. Get All Users****

GET /users

### ****2. Get a Specific User by ID****

(Replace *{user\_id}* with an actual ID)

GET /users/{user\_id}

### ****3. Add a New User****

POST /adduser?name=Alex&email=alex@example.com&age=29

### ****4. Update a User’s Age****

(Replace *{user\_id}* with an actual ID)

PATCH /updateuser?id={user\_id}&age=35

### ****5. Delete a User by ID****

(Replace *{user\_id}* with an actual ID)

DELETE /deleteuser?id={user\_id}