# EXPERIMENT 10

**AIM:** Database connectivity (MongoDB) – CRUD operations using Express and Node js.

**Here are the steps to install MongoDB on different operating systems:**

**For Windows:**

### Step 1: Download MongoDB Installer

- Go to the MongoDB Download Center and select Windows as your operating system.
- Download the latest MongoDB Community Server (choose the MSI package for easy installation).

### Step 2: Install MongoDB

1. Run the downloaded MSI file.
2. During installation:
   - Select Complete setup.
   - Check the box for Install MongoDB as a Service.
   - Leave other options as default (install MongoDB Compass, choose default paths).
1. Click Install to begin the installation.

### Step 3: Add MongoDB to System Path (Optional)

MongoDB is typically added to your system path automatically, but if it's not, you can do it manually:

1. Go to Control Panel > System and Security > System.
2. Click on Advanced system settings > Environment Variables.
3. Under System variables, find Path, click Edit, and add the path where MongoDB is installed.

4. By default, it is:
   C:\Program Files\MongoDB\Server\<version>\bin

**Step 4: Run MongoDB**

Open Command Prompt and start MongoDB by running: Run the command in root also run under express-mongodb-crud directory.

```
mongod
```

If the service was installed, it should start automatically when Windows boots.

**Here are the steps to install Postman on different operating systems:**

**For Windows:**

**Step 1: Download Postman**

- Go to the official Postman download page.
- Select Windows as your operating system and download the latest version of Postman.

**Step 2: Install Postman**

1. Once the download is complete, run the .exe installer.
2. Follow the prompts to install Postman on your system (installation is automatic and simple).
3. After installation, Postman will launch automatically.

**Step 3: Sign In or Create an Account**

- After installation, Postman will ask you to sign in. You can either:
- Sign in using an existing Postman account (using email, Google, or GitHub).
- Skip signing in (Postman will work offline without an account but with limited features).

**To create a Node.js and Express application with MongoDB connectivity and CRUD (Create, Read, Update, Delete) operations, follow these steps.**

**Steps to Create the Project:**

**Initialize Node.js Project:** Run these commands in your terminal:

```
mkdir express-mongodb-crud
cd express-mongodb-crud
npm init -y
```

**Install Dependencies:** You’ll need to install **Express** and **Mongoose** (for MongoDB connection):

```
npm install express mongoose
```

**Install Nodemon** (optional, for easier development): Install Nodemon to automatically restart the server when code changes:

```
npm install -g nodemon
```

**Set up MongoDB:**

- Make sure you have MongoDB installed locally or you can use **MongoDB Atlas** (cloud-based MongoDB).
- If you're using MongoDB locally, ensure the MongoDB service is running.

**Project Structure:**

```
express-mongodb-crud/
├── app.js
├── models/
│   └── user.js
├── routes/
│   └── userRoutes.js
├── package.json
└── package-lock.json
```

### 1. app.js (Main Application)

This file will set up the Express app and connect to the MongoDB database using Mongoose:

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
// Import routes
const userRoutes = require('./routes/userRoutes');
// Initialize express app
const app = express();
// Middleware to parse request body
app.use(bodyParser.json());
// Connect to MongoDB
mongoose.connect('mongodb://127.0.0.1:27017/usersDB', { useNewUrlParser: true,
useUnifiedTopology: true })
  .then(() => console.log("Connected to MongoDB"))
  .catch((err) => console.log("Error connecting to MongoDB:", err));
// Routes
app.use('/api/users', userRoutes);
// Start the server
const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

### 2. models/user.js (User Schema)

This file defines the Mongoose schema for a user:

```
const mongoose = require('mongoose');
const userSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
  },
  email: {
    type: String,
    required: true,
    unique: true,
  },
  age: {
    type: Number,
    required: true,
  },
}, { timestamps: true });
const User = mongoose.model('User', userSchema);
module.exports = User;
```

**3. routes/userRoutes.js (CRUD Operations)**

This file defines the Express routes for performing CRUD operations:

```
const express = require('express');
const User = require('../models/user');
const router = express.Router();
// Create a new user (POST)
router.post('/', async (req, res) => {
  try {
    const { name, email, age } = req.body;
    const newUser = new User({ name, email, age });
    await newUser.save();
    res.status(201).json(newUser);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});
```

```
// Get all users (GET)
router.get('/', async (req, res) => {
  try {
    const users = await User.find();
    res.status(200).json(users);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});
// Get a single user by ID (GET)
router.get('/:id', async (req, res) => {
  try {
    const user = await User.findById(req.params.id);
    if (!user) return res.status(404).json({ message: 'User not found' });
    res.status(200).json(user);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});
// Update a user by ID (PUT)
router.put('/:id', async (req, res) => {
  try {
    const { name, email, age } = req.body;
    const updatedUser = await User.findByIdAndUpdate(
      req.params.id,
      { name, email, age },
      { new: true, runValidators: true }
    );
    if (!updatedUser) return res.status(404).json({ message: 'User not found' });
    res.status(200).json(updatedUser);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});
// Delete a user by ID (DELETE)
router.delete('/:id', async (req, res) => {
  try {
    const user = await User.findByIdAndDelete(req.params.id);
    if (!user) return res.status(404).json({ message: 'User not found' });
    res.status(200).json({ message: 'User deleted' });
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});
module.exports = router;
```

**4. Run the Application:**

1. Start MongoDB locally (or make sure your MongoDB Atlas cluster is set up).

Run the following command to start your server:

```
nodemon app.js
```

Now, your server should be running on http://localhost:3000.

**To test CRUD (Create, Read, Update, Delete) operations on an API, you can use Postman or other API testing tools. Here's a detailed guide on how to execute and test CRUD operations using Postman.**

**CRUD Operations Overview:**

1. **Create** (POST): Add new data to the database.
2. **Read** (GET): Retrieve data from the database.
3. **Update** (PUT or PATCH): Modify existing data.

4. **Delete** (DELETE): Remove data from the database.

**Setup for Testing:**

Before testing, ensure that:

- Your backend API is running.
- You have installed **Postman** on your machine.
- You have an API that handles CRUD operations (e.g., the Node.js and MongoDB application we discussed earlier).

We will assume you have the following API endpoints for **users**:

- POST /api/users – Create a new user.
- GET /api/users – Get all users.
- GET /api/users/:id – Get a specific user by ID.
- PUT /api/users/:id – Update a specific user.
- DELETE /api/users/:id – Delete a specific user.

**Detailed Steps to Execute CRUD Operations Using Postman:**

**1. Create a New User (POST Request)**

**Endpoint:** POST /api/users

**Goal:** To create a new user.

**Steps:**

1. **Open Postman** and click on **New** to create a new request.
2. Set the request type to **POST**.

In the **URL bar**, enter your API endpoint:

```
http://localhost:3000/api/users
```

1. Go to the **Body** tab, select **raw**, and choose **JSON** as the format.

In the body, provide the JSON data for the new user. For example:

```
{
  "name": "Alice",
  "email": "alice@example.com",
  "age": 25
}
```

1. **Add other other examples with different data. No copy paste from given document.**
2. Click **Send**.
3. **Check the Response:**
    - If successful, you should receive a response with the created user data, including an auto-generated ID, timestamps, etc.
    - Status code: `201 Created`

**Expected Response:**

```
{
  "_id": "615fbcf4bf1a1c001c6015a6",
  "name": "Alice",
  "email": "alice@example.com",
  "age": 25,
  "createdAt": "2023-10-10T12:00:00.000Z",
  "updatedAt": "2023-10-10T12:00:00.000Z",
  "__v": 0
}
```

### 2. Read All Users (GET Request)

**Endpoint:** GET /api/users

**Goal:** To retrieve a list of all users from the database.

**Steps:**

1. In **Postman**, create a new request and set it to **GET**.

In the **URL bar**, enter the endpoint:

http://localhost:3000/api/users

1. Click **Send**.
2. **Check the Response:**
   - You should receive an array of user objects.
   - Status code: 200 OK

**Expected Response:**

```
[
  {
    "_id": "615fbcf4bf1a1c001c6015a6",
    "name": "Alice",
    "email": "alice@example.com",
    "age": 25,
    "createdAt": "2023-10-10T12:00:00.000Z",
    "updatedAt": "2023-10-10T12:00:00.000Z",
    "__v": 0
  },
  {
    "_id": "615fbcd9bf1a1c001c6015a5",
    "name": "Bob",
    "email": "bob@example.com",
    "age": 30,
    "createdAt": "2023-10-09T12:00:00.000Z",
    "updatedAt": "2023-10-09T12:00:00.000Z",
    "__v": 0
  }
]
```

### 3. Read a Single User by ID (GET Request)

**Endpoint:** GET /api/users/:id

**Goal:** To retrieve a specific user by their unique ID.

**Steps:**

1. In **Postman**, create a new request and set it to **GET**.

In the **URL bar**, enter the endpoint with the user’s ID:

```
http://localhost:3000/api/users/615fbcf4bf1a1c001c6015a6
```

1. Click **Send**.
2. **Check the Response:**
   - You should receive the user object with the given ID.
   - Status code: 200 OK
   - If the user doesn't exist, the API may return a 404 Not Found status.

**Expected Response:**

```
{
  "_id": "615fbcf4bf1a1c001c6015a6",
  "name": "Alice",
  "email": "alice@example.com",
  "age": 25,
  "createdAt": "2023-10-10T12:00:00.000Z",
  "updatedAt": "2023-10-10T12:00:00.000Z",
  "__v": 0
}
```

**4. Update a User (PUT Request)**

**Endpoint:** `PUT /api/users/:id`

**Goal:** To update a user's information.

**Steps:**

1. In **Postman**, create a new request and set it to **PUT**.

In the **URL bar**, enter the endpoint with the user's ID:

```
http://localhost:3000/api/users/615fbcf4bf1a1c001c6015a6
```

1. Go to the **Body** tab, select **raw**, and choose **JSON** as the format.

Provide the updated data. For example, changing Alice's age:

```
{
 "name": "Alice",
 "email": "alice@example.com",
 "age": 26
}
```

1. Click **Send**.
2. **Check the Response:**
   - You should receive the updated user object.

- Status code: 200 OK

**Expected Response:**

```
{
  "_id": "615fbcf4bf1a1c001c6015a6",
  "name": "Alice",
  "email": "alice@example.com",
  "age": 26,
  "createdAt": "2023-10-10T12:00:00.000Z",
  "updatedAt": "2023-10-11T12:00:00.000Z",
  "__v": 0
}
```

### 5. Delete a User (DELETE Request)

**Endpoint:** DELETE /api/users/:id

**Goal:** To delete a user by their unique ID.

**Steps:**

1. In **Postman**, create a new request and set it to **DELETE**.

In the **URL bar**, enter the endpoint with the user's ID:

http://localhost:3000/api/users/615fbcf4bf1a1c001c6015a6

1. Click **Send**.
2. **Check the Response:**
    - If successful, you should receive a confirmation message.
    - Status code: 200 OK

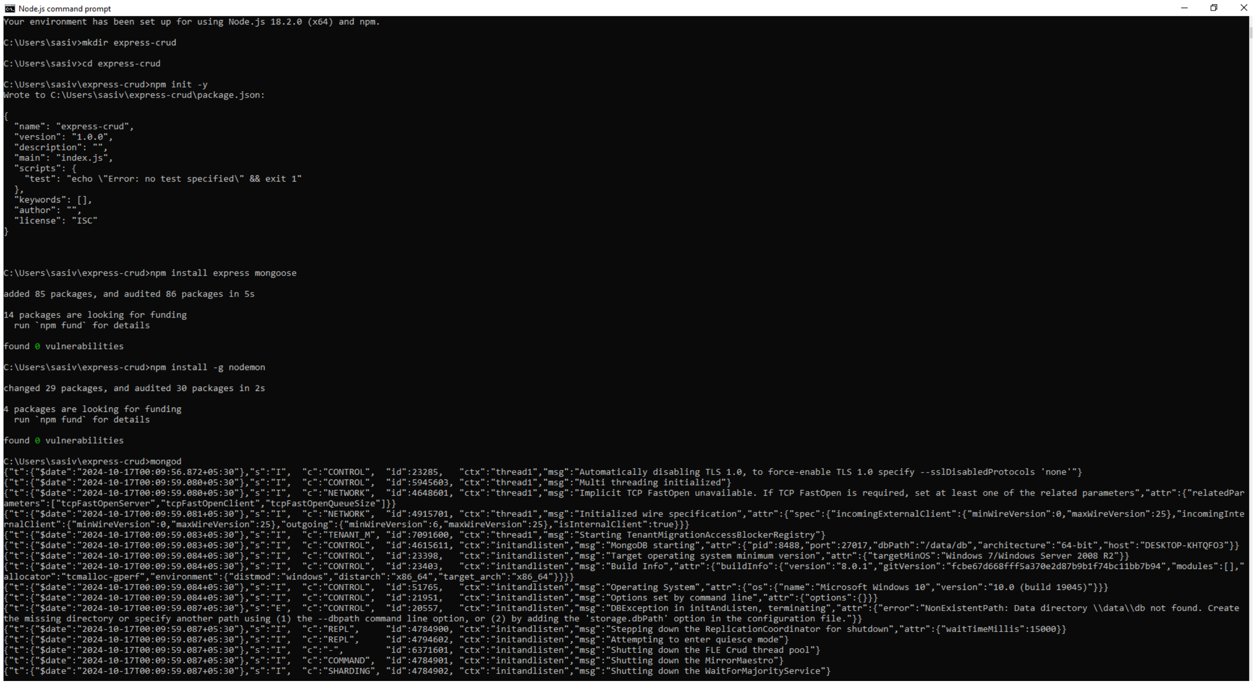
**Expected Response:**

```
{
  "message": "User deleted"
}
```

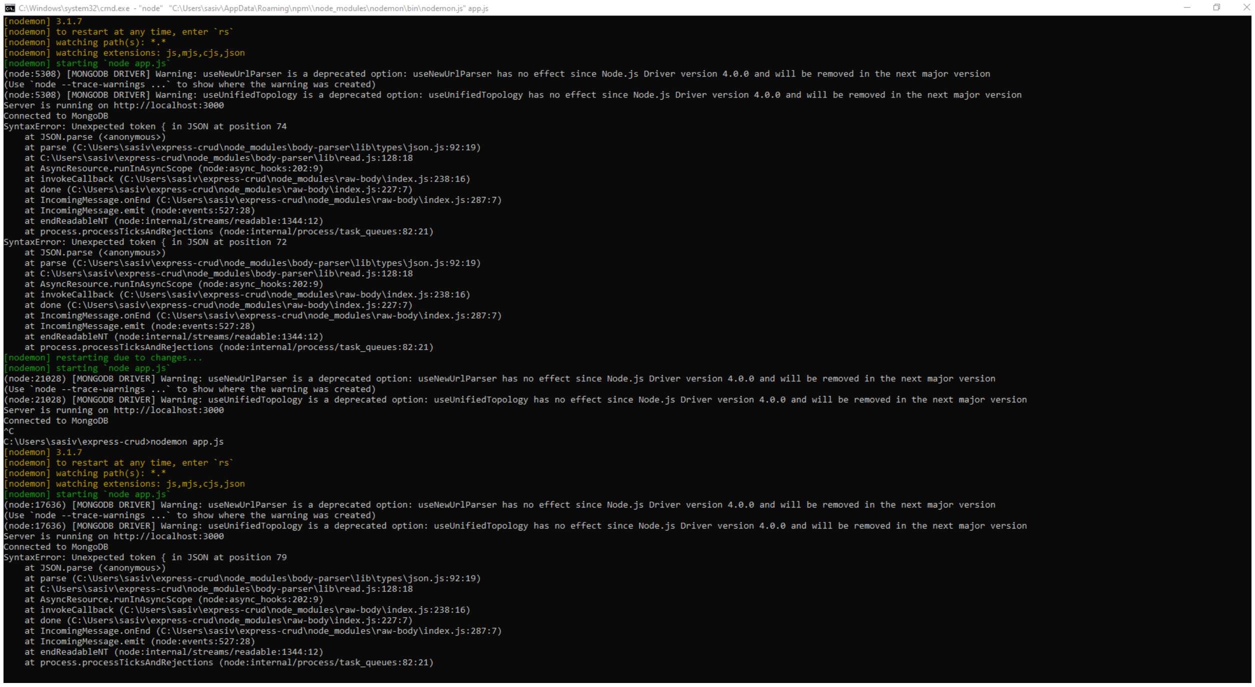
**Key Points:**

- **HTTP Methods:** Each CRUD operation uses a specific HTTP method:
  - POST for creating.
  - GET for reading.
  - PUT for updating.
  - DELETE for deleting.
- **API Endpoints:** You need to provide the correct endpoint for each operation. Dynamic IDs should be replaced with actual values (e.g., :id should be replaced with a real user ID).
- **Request Body:** For POST and PUT requests, the body must be in JSON format and should match the expected structure of the API (e.g., name, email, age).

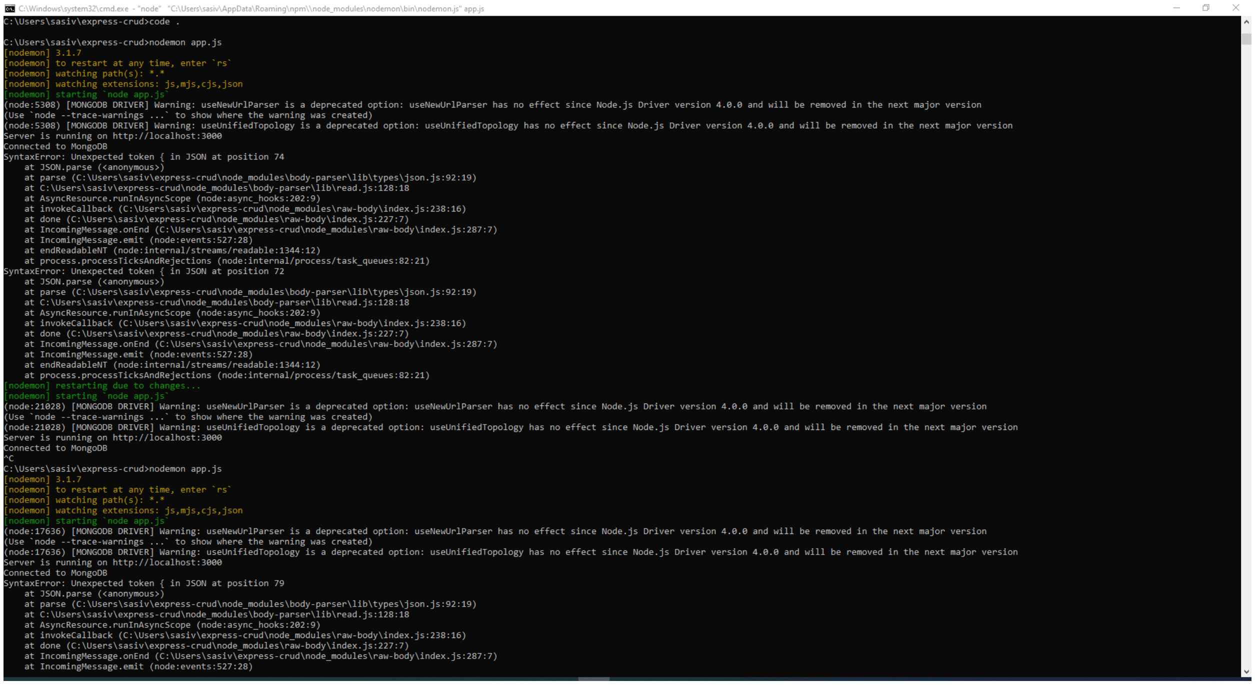
**Output Screenshots:**



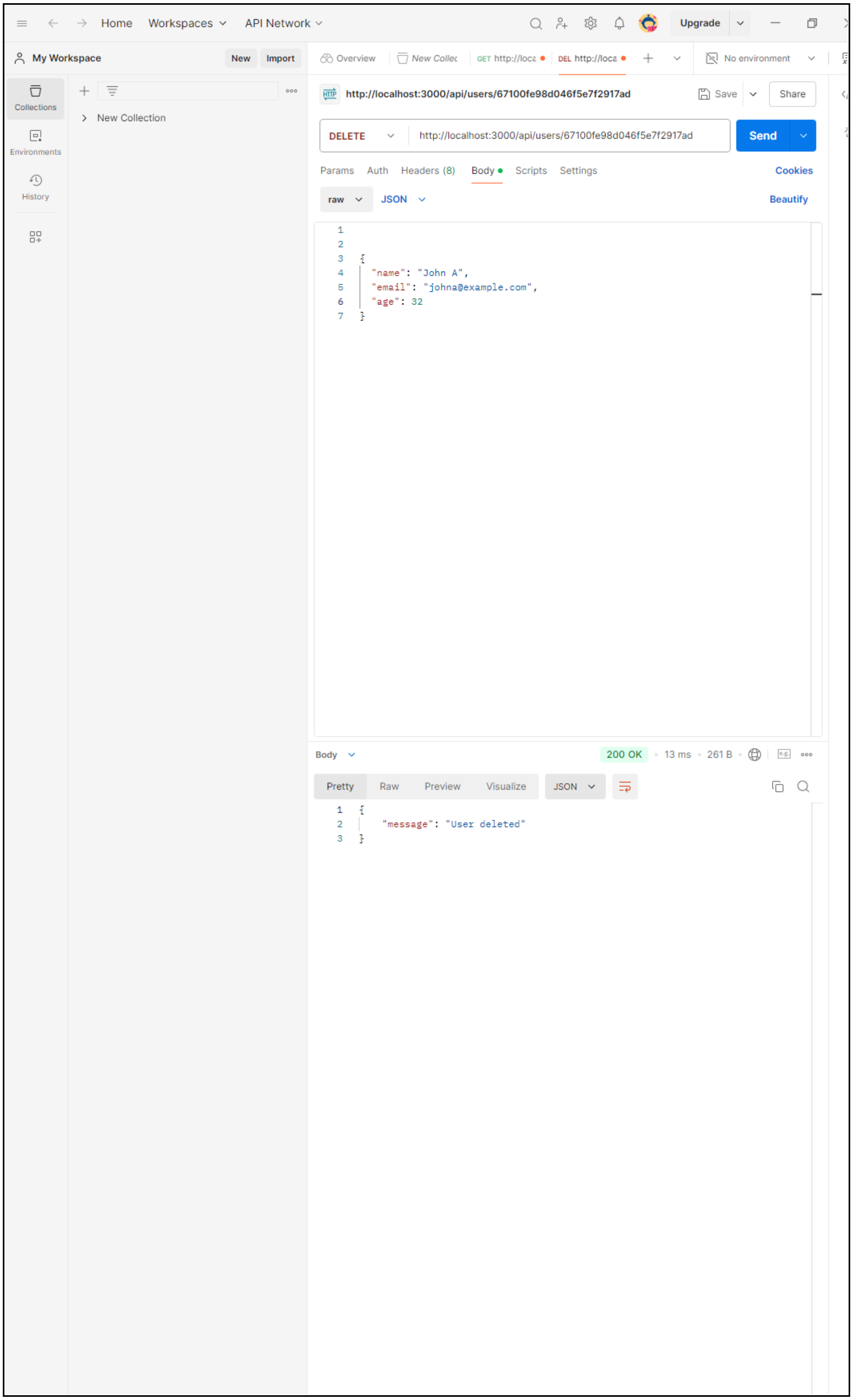
```
Your environment has been set up for using Node.js 18.2.0 (x64) and npm.

C:\Users\sasiv>mkdir express-crud

C:\Users\sasiv>cd express-crud

C:\Users\sasiv\express-crud>npm init -y
Wrote to C:\Users\sasiv\express-crud\package.json:

{
  "name": "express-crud",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}



C:\Users\sasiv\express-crud>npm install express mongoose

added 85 packages, and audited 86 packages in 5s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\sasiv\express-crud>npm install -g nodemon

changed 29 packages, and audited 30 packages in 2s

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\sasiv\express-crud>mongod
{"t":{"$date":"2024-10-17T00:09:56.872+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-10-17T00:09:59.080+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2024-10-17T00:09:59.080+05:30"},"s":"I",  "c":"NETWORK",  "id":4648601, "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related parameters","attr":{"relatedPar
ameters":["tcpFastOpenServer","tcpFastOpenClient","tcpFastOpenQueueSize"]}}
{"t":{"$date":"2024-10-17T00:09:59.081+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25},"incomingInte
rnalClient":{"minWireVersion":0,"maxWireVersion":25},"outgoing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient":true}}}
{"t":{"$date":"2024-10-17T00:09:59.083+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
{"t":{"$date":"2024-10-17T00:09:59.083+05:30"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":8488,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"DESKTOP-KHTQFO3"}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"8.0.1","gitVersion":"fcbe67d668fff5a370e2d87b9b1f74bc11bb7b94","modules":[],"
allocator":"tcmalloc-gperf","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19045)"}}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{}}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"E",  "c":"CONTROL",  "id":20557,   "ctx":"initandlisten","msg":"DBException in initAndListen, terminating","attr":{"error":"NonExistentPath: Data directory \\data\\db not found. Create
the missing directory or specify another path using (1) the --dbpath command line option, or (2) by adding the 'storage.dbPath' option in the configuration file."}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"REPL",     "id":4784900, "ctx":"initandlisten","msg":"Stepping down the ReplicationCoordinator for shutdown","attr":{"waitTimeMillis":15000}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"REPL",     "id":4794602, "ctx":"initandlisten","msg":"Attempting to enter quiesce mode"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"-",        "id":6371601, "ctx":"initandlisten","msg":"Shutting down the FLE Crud thread pool"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"COMMAND",  "id":4784901, "ctx":"initandlisten","msg":"Shutting down the MirrorMaestro"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"SHARDING", "id":4784902, "ctx":"initandlisten","msg":"Shutting down the WaitForMajorityService"}
```


C:\Windows\system32\cmd.exe - "node"  "C:\Users\sasiv\AppData\Roaming\npm\\node_modules\nodemon\bin\nodemon.js" app.js

```
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
(node:5308) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:5308) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
Connected to MongoDB
SyntaxError: Unexpected token { in JSON at position 74
    at JSON.parse (<anonymous>)
    at parse (C:\Users\sasiv\express-crud\node_modules\body-parser\lib\types\json.js:92:19)
    at C:\Users\sasiv\express-crud\node_modules\body-parser\lib\read.js:128:18
    at AsyncResource.runInAsyncScope (node:async_hooks:202:9)
    at invokeCallback (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:238:16)
    at done (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:227:7)
    at IncomingMessage.onEnd (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:287:7)
    at IncomingMessage.emit (node:events:527:28)
    at endReadableNT (node:internal/streams/readable:1344:12)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21)
SyntaxError: Unexpected token { in JSON at position 72
    at JSON.parse (<anonymous>)
    at parse (C:\Users\sasiv\express-crud\node_modules\body-parser\lib\types\json.js:92:19)
    at C:\Users\sasiv\express-crud\node_modules\body-parser\lib\read.js:128:18
    at AsyncResource.runInAsyncScope (node:async_hooks:202:9)
    at invokeCallback (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:238:16)
    at done (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:227:7)
    at IncomingMessage.onEnd (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:287:7)
    at IncomingMessage.emit (node:events:527:28)
    at endReadableNT (node:internal/streams/readable:1344:12)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21)
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
(node:21028) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:21028) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
Connected to MongoDB
^C
C:\Users\sasiv\express-crud>nodemon app.js
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
(node:17636) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:17636) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
Connected to MongoDB
SyntaxError: Unexpected token { in JSON at position 79
    at JSON.parse (<anonymous>)
    at parse (C:\Users\sasiv\express-crud\node_modules\body-parser\lib\types\json.js:92:19)
    at C:\Users\sasiv\express-crud\node_modules\body-parser\lib\read.js:128:18
    at AsyncResource.runInAsyncScope (node:async_hooks:202:9)
    at invokeCallback (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:238:16)
    at done (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:227:7)
    at IncomingMessage.onEnd (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:287:7)
    at IncomingMessage.emit (node:events:527:28)
    at endReadableNT (node:internal/streams/readable:1344:12)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21)
```


C:\Windows\system32\cmd.exe - "node"  "C:\Users\sasiv\AppData\Roaming\npm\\node_modules\nodemon\bin\nodemon.js" app.js

```
C:\Users\sasiv\express-crud>code .

C:\Users\sasiv\express-crud>nodemon app.js
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
(node:5308) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:5308) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
Connected to MongoDB
SyntaxError: Unexpected token { in JSON at position 74
    at JSON.parse (<anonymous>)
    at parse (C:\Users\sasiv\express-crud\node_modules\body-parser\lib\types\json.js:92:19)
    at C:\Users\sasiv\express-crud\node_modules\body-parser\lib\read.js:128:18
    at AsyncResource.runInAsyncScope (node:async_hooks:202:9)
    at invokeCallback (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:238:16)
    at done (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:227:7)
    at IncomingMessage.onEnd (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:287:7)
    at IncomingMessage.emit (node:events:527:28)
    at endReadableNT (node:internal/streams/readable:1344:12)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21)
SyntaxError: Unexpected token { in JSON at position 72
    at JSON.parse (<anonymous>)
    at parse (C:\Users\sasiv\express-crud\node_modules\body-parser\lib\types\json.js:92:19)
    at C:\Users\sasiv\express-crud\node_modules\body-parser\lib\read.js:128:18
    at AsyncResource.runInAsyncScope (node:async_hooks:202:9)
    at invokeCallback (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:238:16)
    at done (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:227:7)
    at IncomingMessage.onEnd (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:287:7)
    at IncomingMessage.emit (node:events:527:28)
    at endReadableNT (node:internal/streams/readable:1344:12)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21)
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
(node:21028) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:21028) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
Connected to MongoDB
^C
C:\Users\sasiv\express-crud>nodemon app.js
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
(node:17636) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:17636) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
Connected to MongoDB
SyntaxError: Unexpected token { in JSON at position 79
    at JSON.parse (<anonymous>)
    at parse (C:\Users\sasiv\express-crud\node_modules\body-parser\lib\types\json.js:92:19)
    at C:\Users\sasiv\express-crud\node_modules\body-parser\lib\read.js:128:18
    at AsyncResource.runInAsyncScope (node:async_hooks:202:9)
    at invokeCallback (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:238:16)
    at done (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:227:7)
    at IncomingMessage.onEnd (C:\Users\sasiv\express-crud\node_modules\raw-body\index.js:287:7)
    at IncomingMessage.emit (node:events:527:28)
```



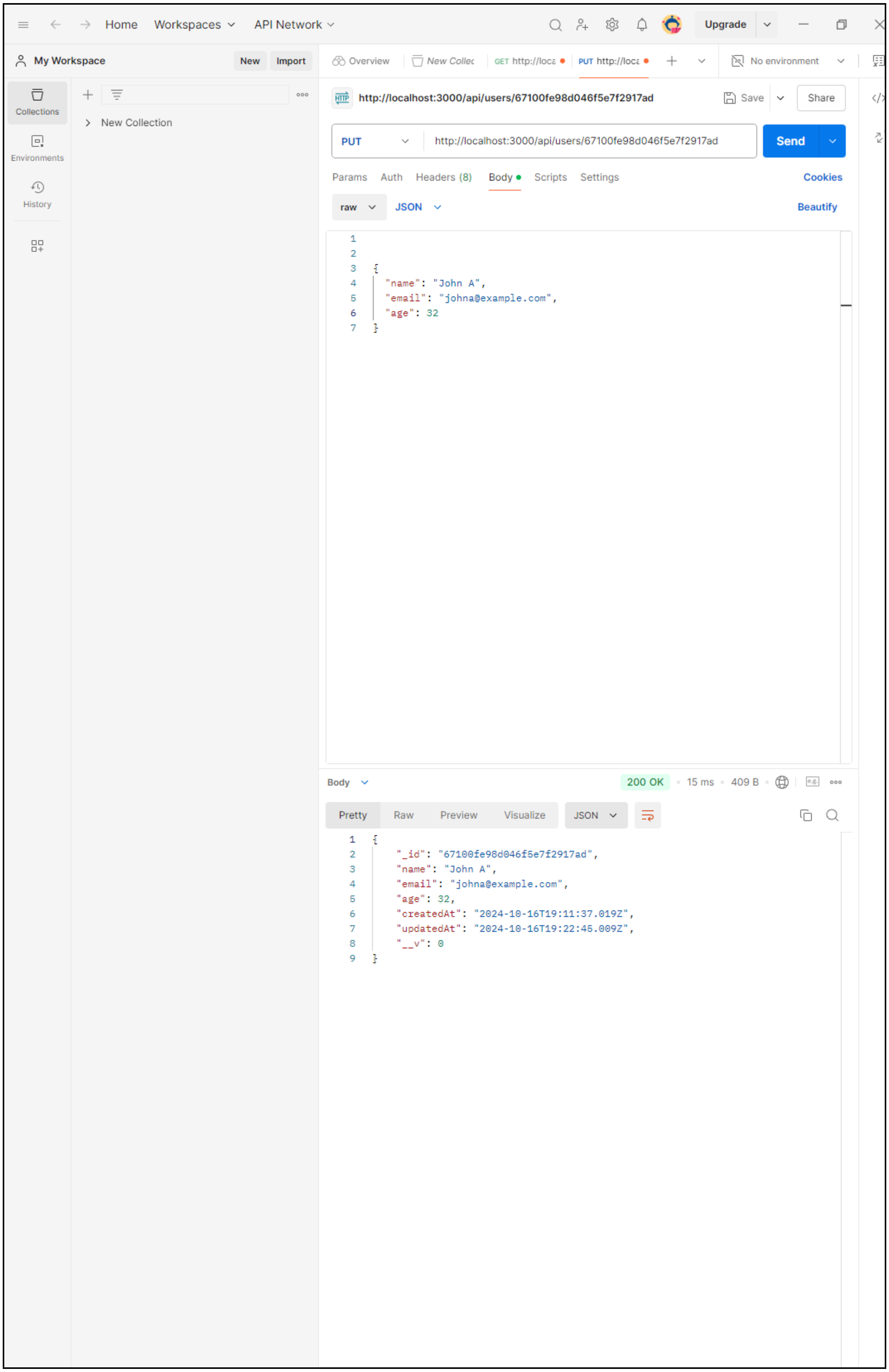
http://localhost:3000/api/users/67100fe98d046f5e7f2917ad

DELETE http://localhost:3000/api/users/67100fe98d046f5e7f2917ad

```


{
  "name": "John A",
  "email": "johna@example.com",
  "age": 32
}
```

Body 200 OK 13 ms 261 B

```
{
    "message": "User deleted"
}
```



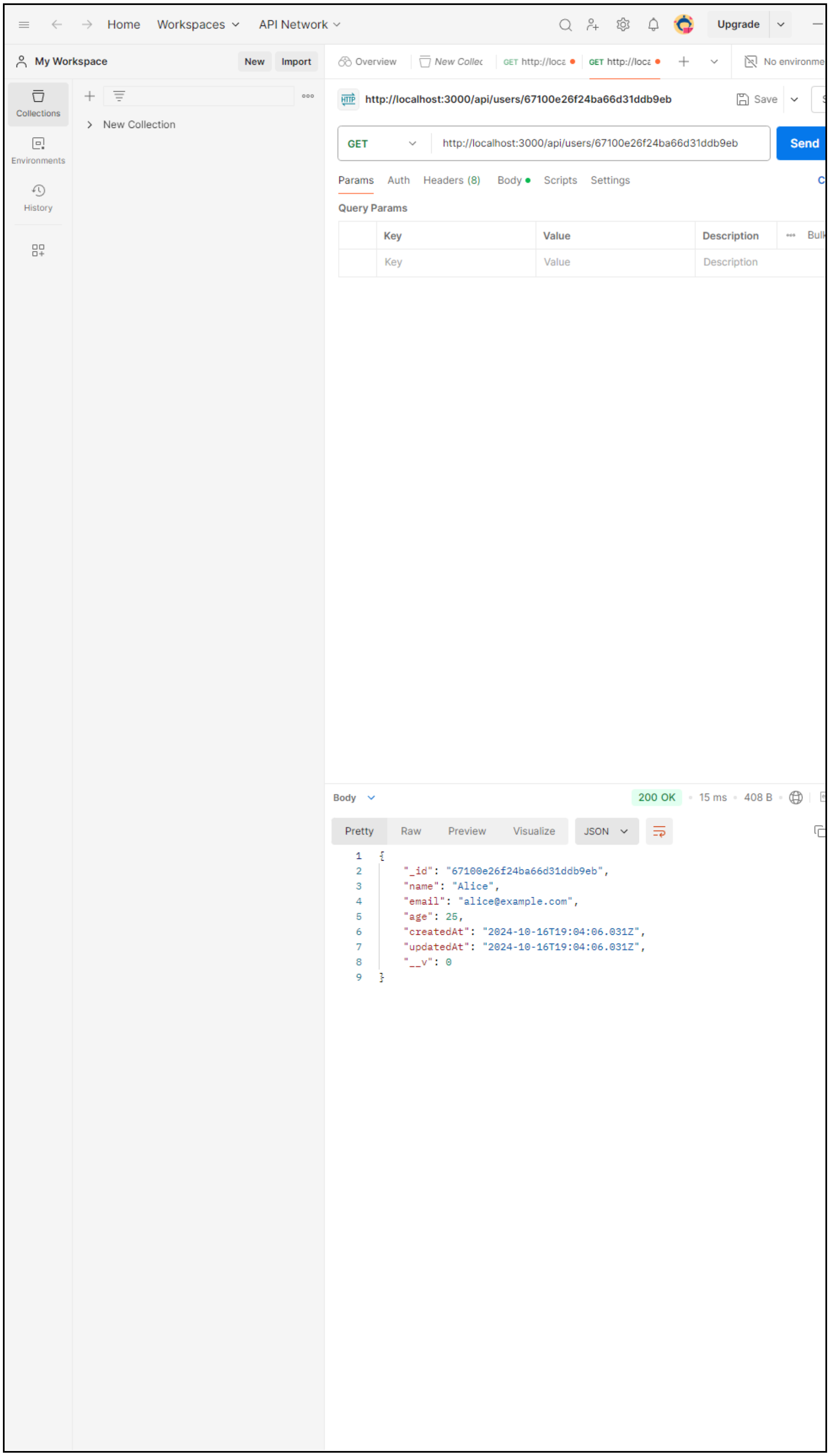
http://localhost:3000/api/users/67100fe98d046f5e7f2917ad

PUT http://localhost:3000/api/users/67100fe98d046f5e7f2917ad

```


{
  "name": "John A",
  "email": "johna@example.com",
  "age": 32
}
```

Body 200 OK • 15 ms • 409 B

```
{
    "_id": "67100fe98d046f5e7f2917ad",
    "name": "John A",
    "email": "johna@example.com",
    "age": 32,
    "createdAt": "2024-10-16T19:11:37.019Z",
    "updatedAt": "2024-10-16T19:22:45.009Z",
    "__v": 0
}
```



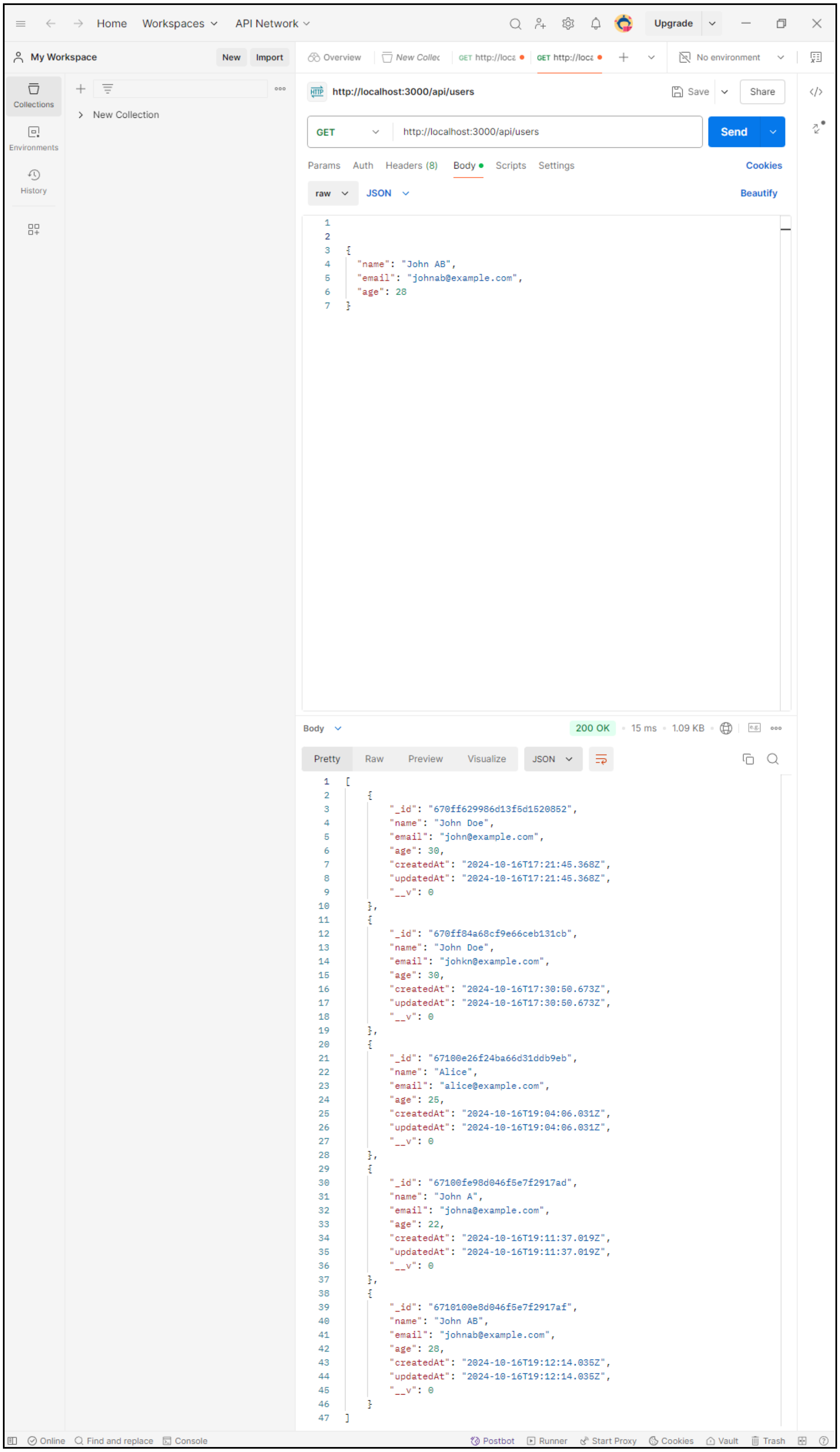
http://localhost:3000/api/users/67100e26f24ba66d31ddb9eb

GET http://localhost:3000/api/users/67100e26f24ba66d31ddb9eb

Send

200 OK · 15 ms · 408 B

```
{
    "_id": "67100e26f24ba66d31ddb9eb",
    "name": "Alice",
    "email": "alice@example.com",
    "age": 25,
    "createdAt": "2024-10-16T19:04:06.031Z",
    "updatedAt": "2024-10-16T19:04:06.031Z",
    "__v": 0
}
```



http://localhost:3000/api/users

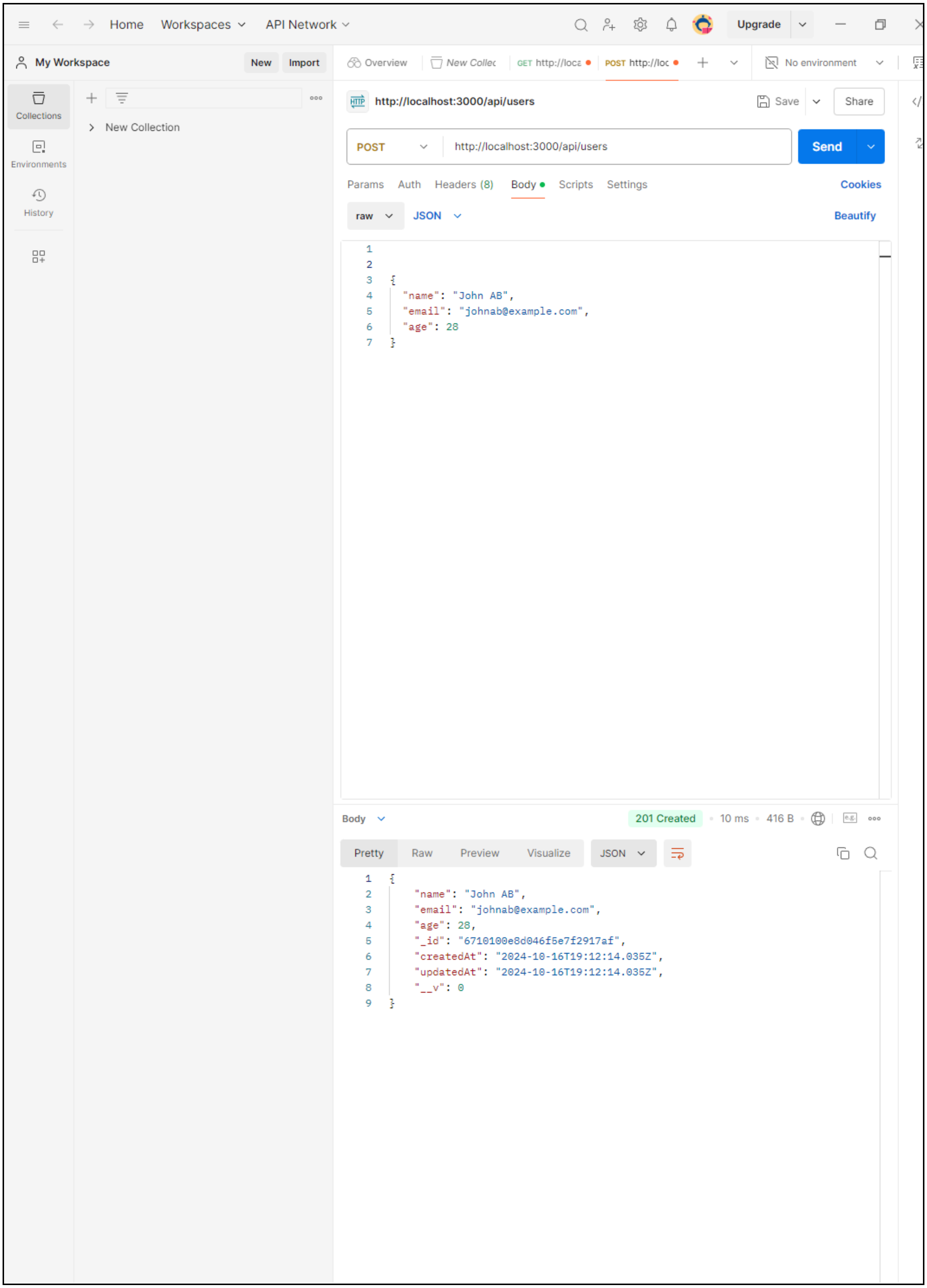
GET http://localhost:3000/api/users

```


{
  "name": "John AB",
  "email": "johnab@example.com",
  "age": 28
}
```

Body 200 OK · 15 ms · 1.09 KB

JSON

```
[
    {
        "_id": "670ff629986d13f5d1520852",
        "name": "John Doe",
        "email": "john@example.com",
        "age": 30,
        "createdAt": "2024-10-16T17:21:45.368Z",
        "updatedAt": "2024-10-16T17:21:45.368Z",
        "__v": 0
    },
    {
        "_id": "670ff84a68cf9e66ceb131cb",
        "name": "John Doe",
        "email": "johkn@example.com",
        "age": 30,
        "createdAt": "2024-10-16T17:30:50.673Z",
        "updatedAt": "2024-10-16T17:30:50.673Z",
        "__v": 0
    },
    {
        "_id": "67100e26f24ba66d31ddb9eb",
        "name": "Alice",
        "email": "alice@example.com",
        "age": 25,
        "createdAt": "2024-10-16T19:04:06.031Z",
        "updatedAt": "2024-10-16T19:04:06.031Z",
        "__v": 0
    },
    {
        "_id": "67100fe98d046f5e7f2917ad",
        "name": "John A",
        "email": "johna@example.com",
        "age": 22,
        "createdAt": "2024-10-16T19:11:37.019Z",
        "updatedAt": "2024-10-16T19:11:37.019Z",
        "__v": 0
    },
    {
        "_id": "6710100e8d046f5e7f2917af",
        "name": "John AB",
        "email": "johnab@example.com",
        "age": 28,
        "createdAt": "2024-10-16T19:12:14.035Z",
        "updatedAt": "2024-10-16T19:12:14.035Z",
        "__v": 0
    }
]
```

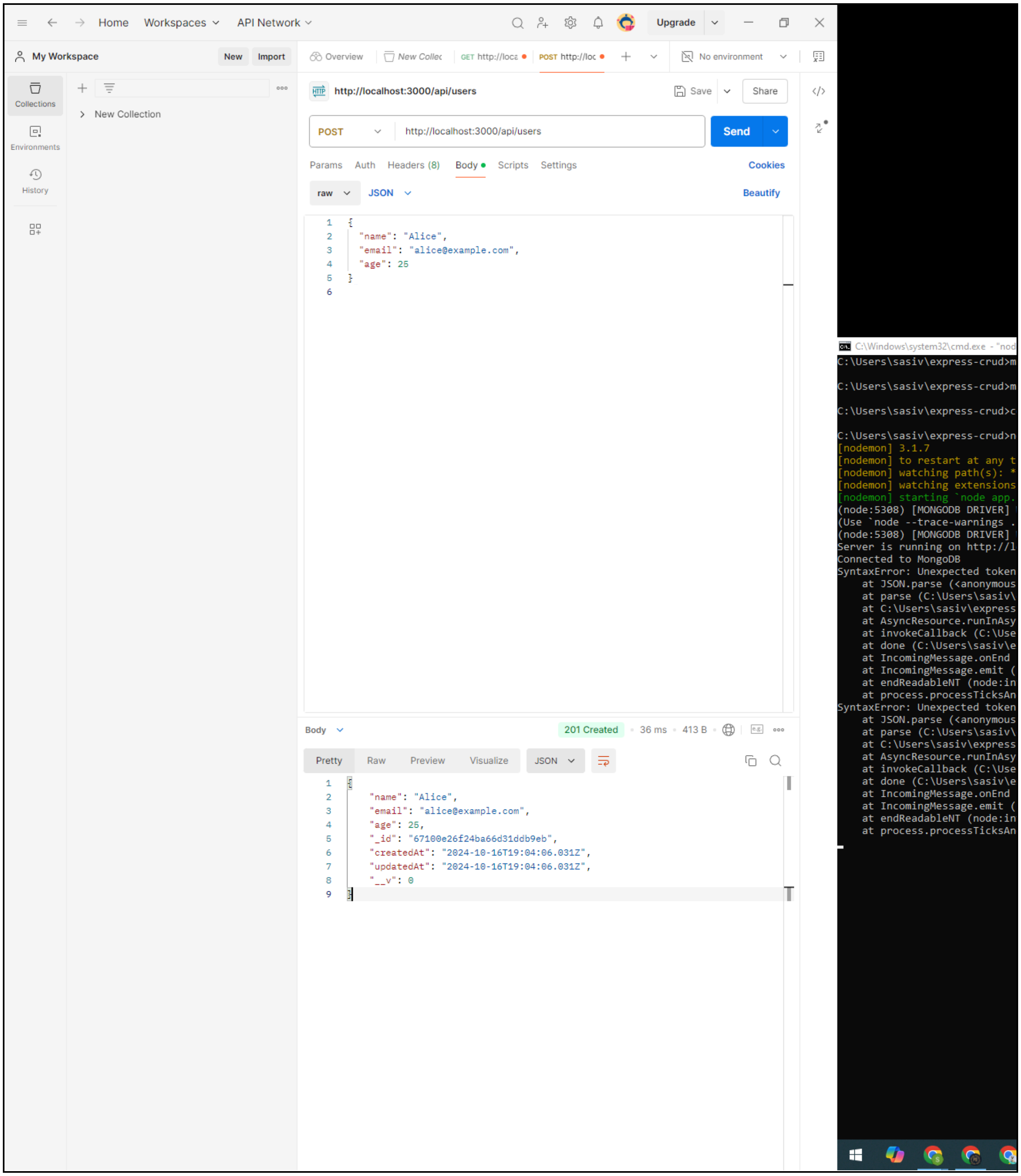


POST http://localhost:3000/api/users

```


{
  "name": "John AB",
  "email": "johnab@example.com",
  "age": 28
}
```

Body — 201 Created · 10 ms · 416 B

```
{
    "name": "John AB",
    "email": "johnab@example.com",
    "age": 28,
    "_id": "6710100e8d046f5e7f2917af",
    "createdAt": "2024-10-16T19:12:14.035Z",
    "updatedAt": "2024-10-16T19:12:14.035Z",
    "__v": 0
}
```



POST http://localhost:3000/api/users

```
{
  "name": "Alice",
  "email": "alice@example.com",
  "age": 25
}
```

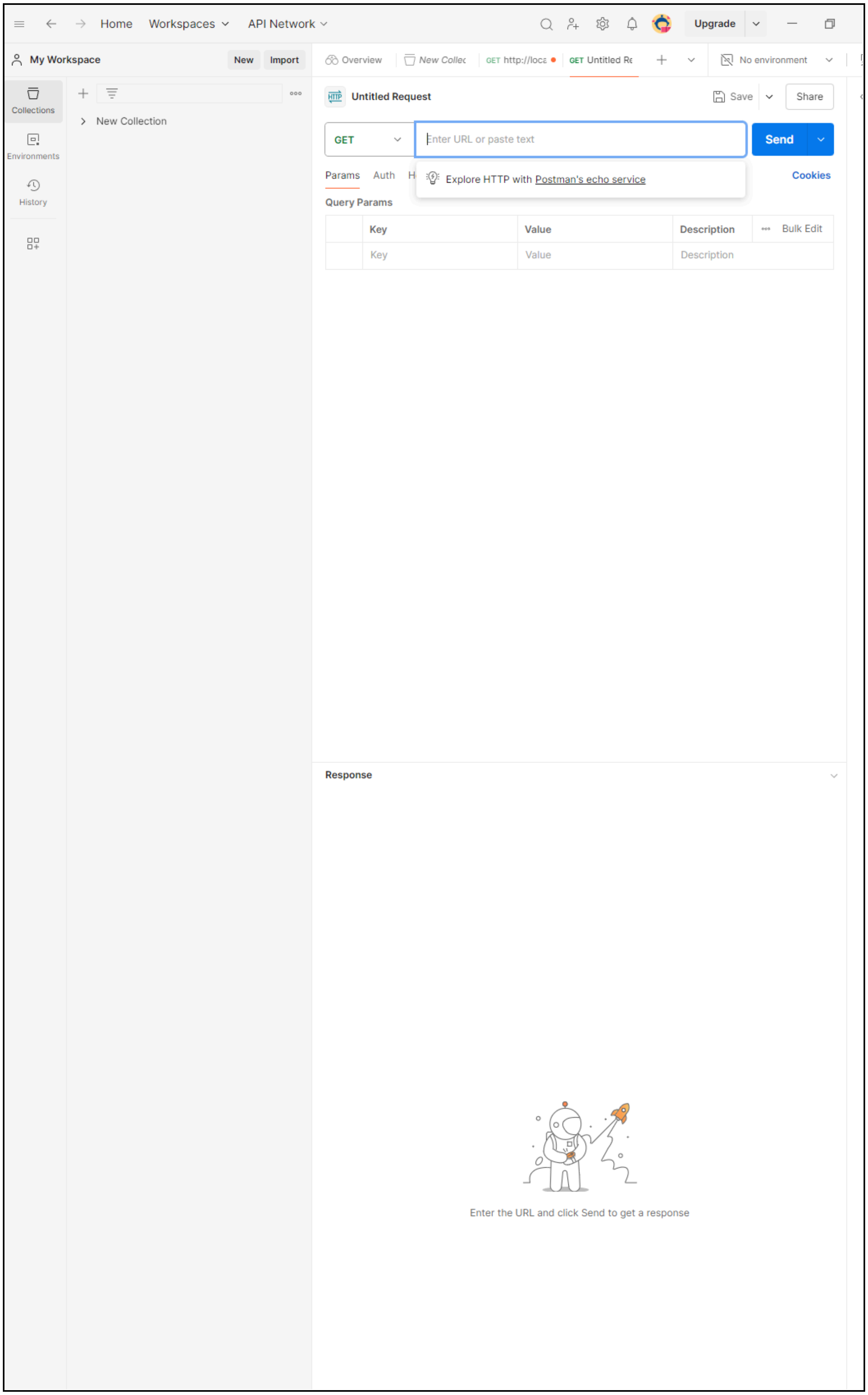
Body · 201 Created · 36 ms · 413 B

```
{
    "name": "Alice",
    "email": "alice@example.com",
    "age": 25,
    "_id": "67100e26f24ba66d31ddb9eb",
    "createdAt": "2024-10-16T19:04:06.031Z",
    "updatedAt": "2024-10-16T19:04:06.031Z",
    "__v": 0
}
```

C:\Windows\system32\cmd.exe - "nod

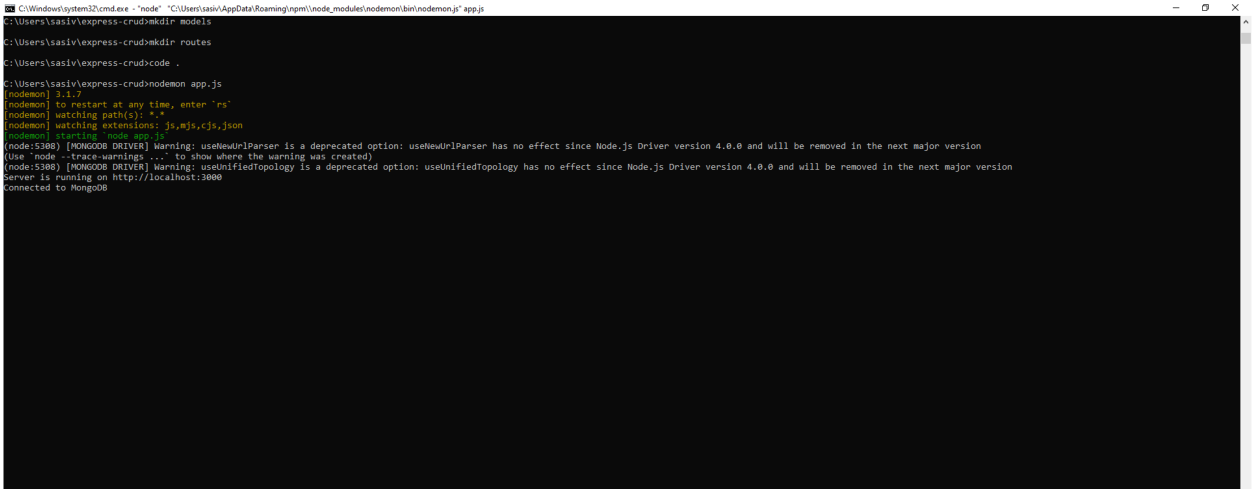
```
C:\Users\sasiv\express-crud>m

C:\Users\sasiv\express-crud>m

C:\Users\sasiv\express-crud>c

C:\Users\sasiv\express-crud>n
[nodemon] 3.1.7
[nodemon] to restart at any t
[nodemon] watching path(s): *
[nodemon] watching extensions
[nodemon] starting `node app.
(node:5308) [MONGODB DRIVER]
(Use `node --trace-warnings .
(node:5308) [MONGODB DRIVER]
Server is running on http://l
Connected to MongoDB
SyntaxError: Unexpected token
    at JSON.parse (<anonymous
    at parse (C:\Users\sasiv\
    at C:\Users\sasiv\express
    at AsyncResource.runInAsy
    at invokeCallback (C:\Use
    at done (C:\Users\sasiv\e
    at IncomingMessage.onEnd
    at IncomingMessage.emit (
    at endReadableNT (node:in
    at process.processTicksAn
SyntaxError: Unexpected token
    at JSON.parse (<anonymous
    at parse (C:\Users\sasiv\
    at C:\Users\sasiv\express
    at AsyncResource.runInAsy
    at invokeCallback (C:\Use
    at done (C:\Users\sasiv\e
    at IncomingMessage.onEnd
    at IncomingMessage.emit (
    at endReadableNT (node:in
    at process.processTicksAn
```

]



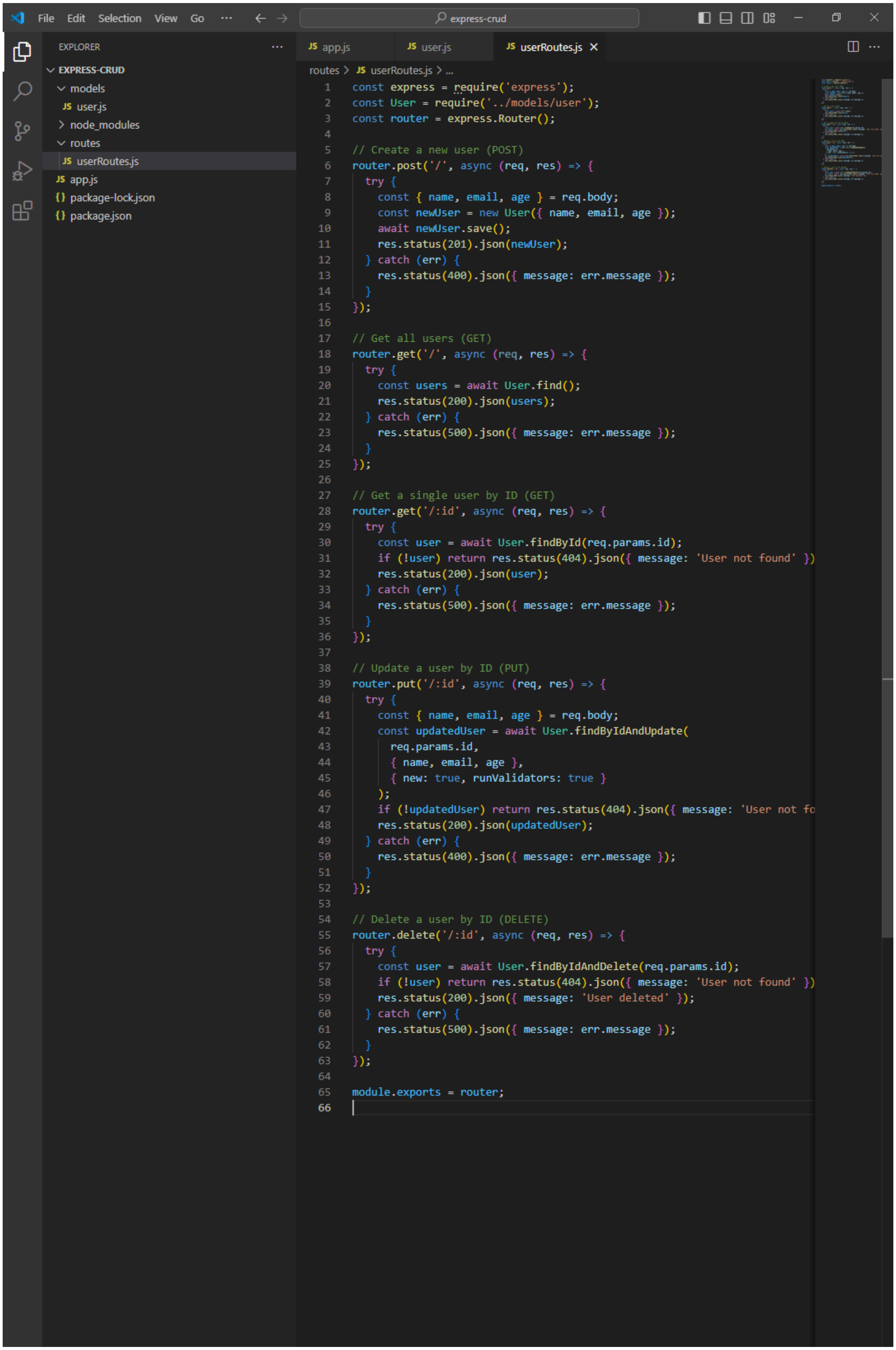
| | Key | Value | Description |
|---|---|---|---|
| | Key | Value | Description |

Enter the URL and click Send to get a response

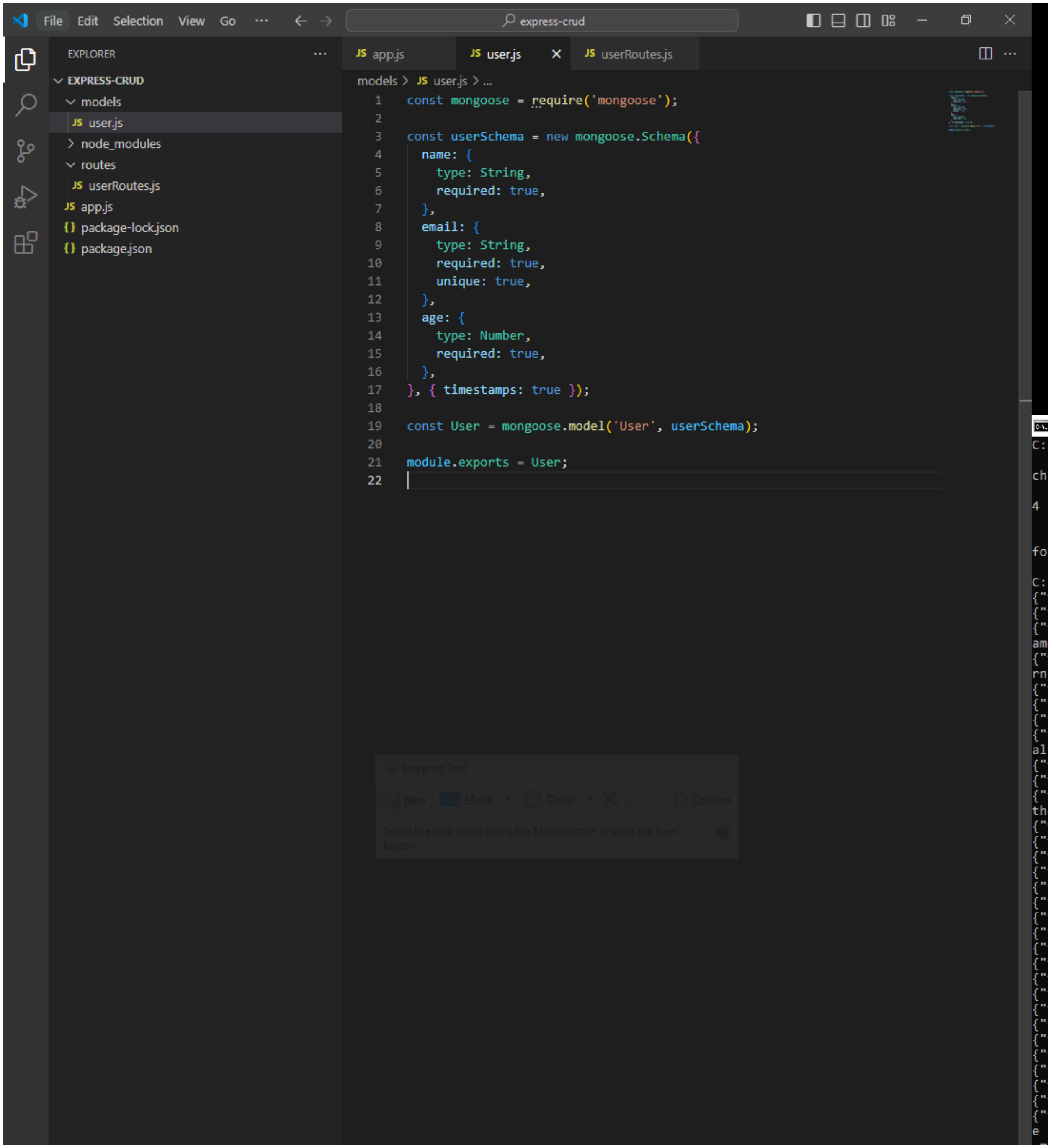


C:\Windows\system32\cmd.exe - "node"  "C:\Users\sasiv\AppData\Roaming\npm\\node_modules\nodemon\bin\nodemon.js" app.js

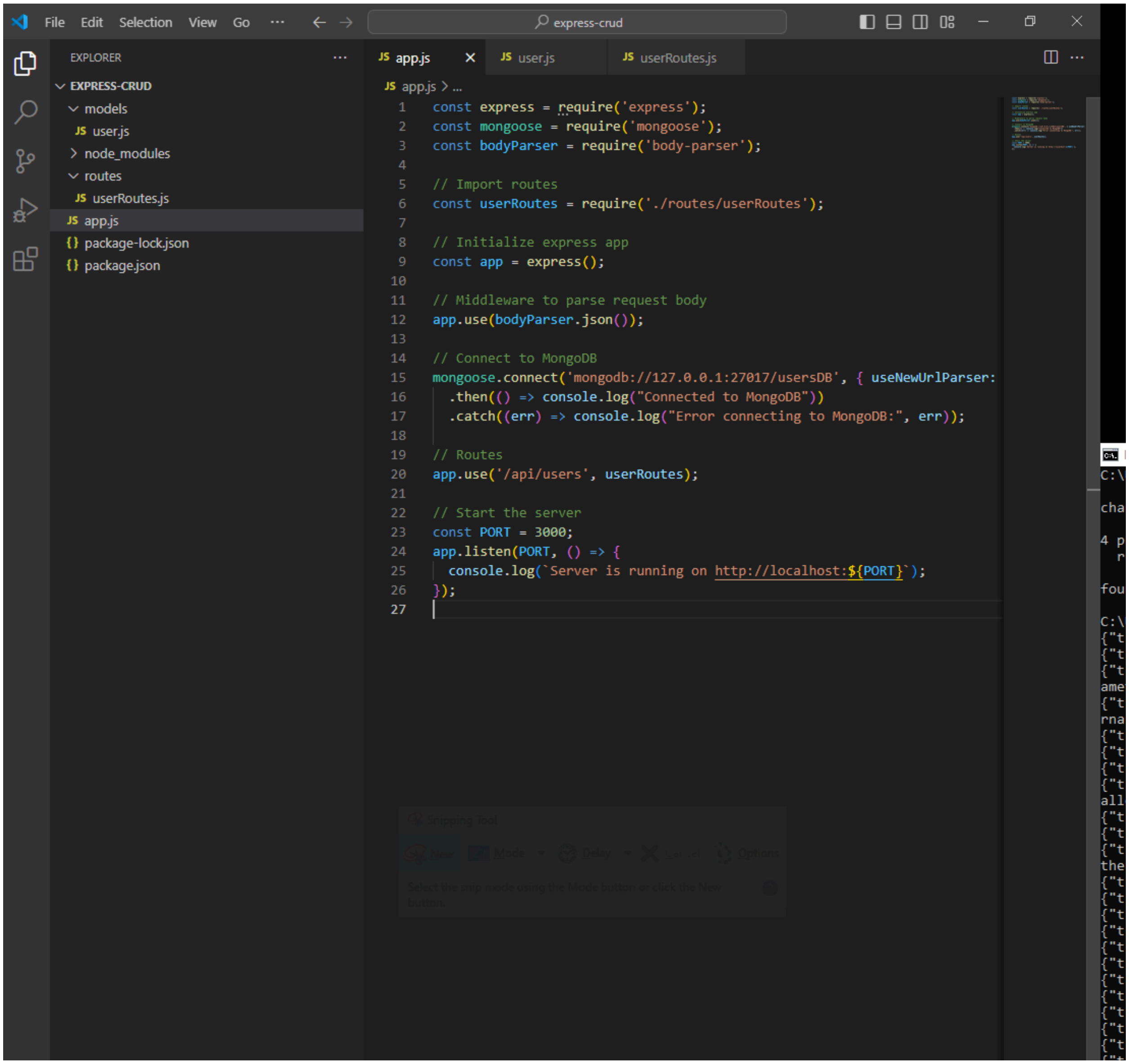
```
C:\Users\sasiv\express-crud>mkdir models

C:\Users\sasiv\express-crud>mkdir routes

C:\Users\sasiv\express-crud>code .

C:\Users\sasiv\express-crud>nodemon app.js
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
(node:5308) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:5308) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on http://localhost:3000
Connected to MongoDB
```



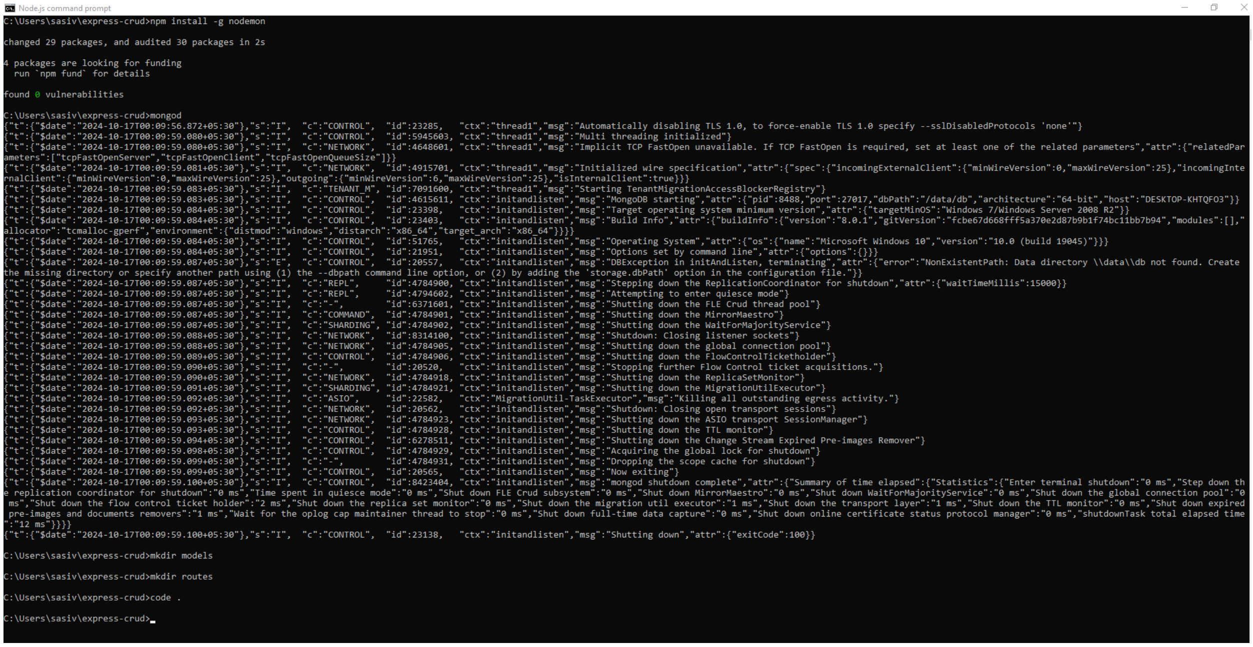
```javascript
const express = require('express');
const User = require('../models/user');
const router = express.Router();

// Create a new user (POST)
router.post('/', async (req, res) => {
  try {
    const { name, email, age } = req.body;
    const newUser = new User({ name, email, age });
    await newUser.save();
    res.status(201).json(newUser);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});

// Get all users (GET)
router.get('/', async (req, res) => {
  try {
    const users = await User.find();
    res.status(200).json(users);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

// Get a single user by ID (GET)
router.get('/:id', async (req, res) => {
  try {
    const user = await User.findById(req.params.id);
    if (!user) return res.status(404).json({ message: 'User not found' })
    res.status(200).json(user);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

// Update a user by ID (PUT)
router.put('/:id', async (req, res) => {
  try {
    const { name, email, age } = req.body;
    const updatedUser = await User.findByIdAndUpdate(
      req.params.id,
      { name, email, age },
      { new: true, runValidators: true }
    );
    if (!updatedUser) return res.status(404).json({ message: 'User not fo
    res.status(200).json(updatedUser);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
});

// Delete a user by ID (DELETE)
router.delete('/:id', async (req, res) => {
  try {
    const user = await User.findByIdAndDelete(req.params.id);
    if (!user) return res.status(404).json({ message: 'User not found' })
    res.status(200).json({ message: 'User deleted' });
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
});

module.exports = router;
```



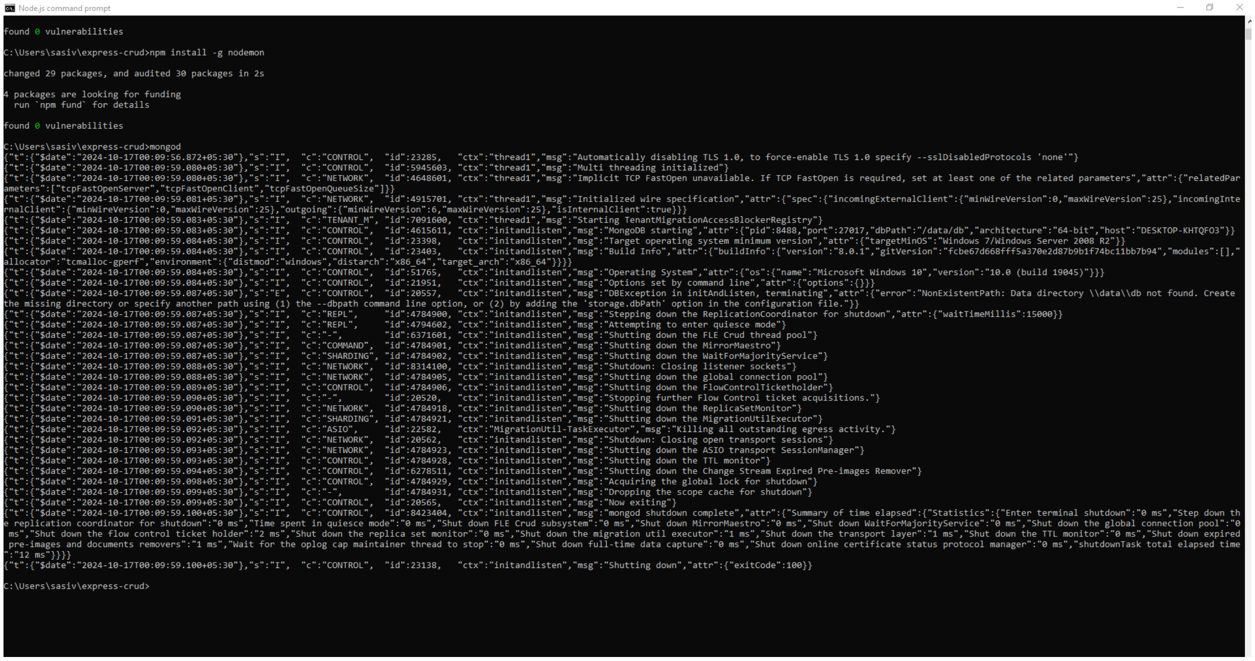
```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
  },
  email: {
    type: String,
    required: true,
    unique: true,
  },
  age: {
    type: Number,
    required: true,
  },
}, { timestamps: true });

const User = mongoose.model('User', userSchema);

module.exports = User;

```



```javascript
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

// Import routes
const userRoutes = require('./routes/userRoutes');

// Initialize express app
const app = express();

// Middleware to parse request body
app.use(bodyParser.json());

// Connect to MongoDB
mongoose.connect('mongodb://127.0.0.1:27017/usersDB', { useNewUrlParser:
  .then(() => console.log("Connected to MongoDB"))
  .catch((err) => console.log("Error connecting to MongoDB:", err));

// Routes
app.use('/api/users', userRoutes);

// Start the server
const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```



```
C:\Users\sasiv\express-crud>npm install -g nodemon

changed 29 packages, and audited 30 packages in 2s

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\sasiv\express-crud>mongod
{"t":{"$date":"2024-10-17T00:09:56.872+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-10-17T00:09:59.080+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2024-10-17T00:09:59.080+05:30"},"s":"I",  "c":"NETWORK",  "id":4648601, "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related parameters","attr":{"relatedPar
ameters":["tcpFastOpenServer","tcpFastOpenClient","tcpFastOpenQueueSize"]}}
{"t":{"$date":"2024-10-17T00:09:59.081+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25},"incomingInte
rnalClient":{"minWireVersion":0,"maxWireVersion":25},"outgoing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient":true}}}
{"t":{"$date":"2024-10-17T00:09:59.083+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
{"t":{"$date":"2024-10-17T00:09:59.083+05:30"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":8488,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"DESKTOP-KHTQFO3"}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"8.0.1","gitVersion":"fcbe67d668fff5a370e2d87b9b1f74bc11bb7b94","modules":[],"
allocator":"tcmalloc-gperf","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19045)"}}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{}}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"E",  "c":"CONTROL",  "id":20557,   "ctx":"initandlisten","msg":"DBException in initAndListen, terminating","attr":{"error":"NonExistentPath: Data directory \\data\\db not found. Create
the missing directory or specify another path using (1) the --dbpath command line option, or (2) by adding the 'storage.dbPath' option in the configuration file."}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"REPL",     "id":4784900, "ctx":"initandlisten","msg":"Stepping down the ReplicationCoordinator for shutdown","attr":{"waitTimeMillis":15000}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"REPL",     "id":4794602, "ctx":"initandlisten","msg":"Attempting to enter quiesce mode"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"-",        "id":6371601, "ctx":"initandlisten","msg":"Shutting down the FLE Crud thread pool"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"COMMAND",  "id":4784901, "ctx":"initandlisten","msg":"Shutting down the MirrorMaestro"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"SHARDING", "id":4784902, "ctx":"initandlisten","msg":"Shutting down the WaitForMajorityService"}
{"t":{"$date":"2024-10-17T00:09:59.088+05:30"},"s":"I",  "c":"NETWORK",  "id":8314100, "ctx":"initandlisten","msg":"Shutdown: Closing listener sockets"}
{"t":{"$date":"2024-10-17T00:09:59.088+05:30"},"s":"I",  "c":"NETWORK",  "id":4784905, "ctx":"initandlisten","msg":"Shutting down the global connection pool"}
{"t":{"$date":"2024-10-17T00:09:59.089+05:30"},"s":"I",  "c":"CONTROL",  "id":4784906, "ctx":"initandlisten","msg":"Shutting down the FlowControlTicketholder"}
{"t":{"$date":"2024-10-17T00:09:59.090+05:30"},"s":"I",  "c":"-",        "id":20520,   "ctx":"initandlisten","msg":"Stopping further Flow Control ticket acquisitions."}
{"t":{"$date":"2024-10-17T00:09:59.090+05:30"},"s":"I",  "c":"NETWORK",  "id":4784918, "ctx":"initandlisten","msg":"Shutting down the ReplicaSetMonitor"}
{"t":{"$date":"2024-10-17T00:09:59.091+05:30"},"s":"I",  "c":"SHARDING", "id":4784921, "ctx":"initandlisten","msg":"Shutting down the MigrationUtilExecutor"}
{"t":{"$date":"2024-10-17T00:09:59.092+05:30"},"s":"I",  "c":"ASIO",     "id":22582,   "ctx":"MigrationUtil-TaskExecutor","msg":"Killing all outstanding egress activity."}
{"t":{"$date":"2024-10-17T00:09:59.092+05:30"},"s":"I",  "c":"NETWORK",  "id":20562,   "ctx":"initandlisten","msg":"Shutdown: Closing open transport sessions"}
{"t":{"$date":"2024-10-17T00:09:59.093+05:30"},"s":"I",  "c":"NETWORK",  "id":4784923, "ctx":"initandlisten","msg":"Shutting down the ASIO transport SessionManager"}
{"t":{"$date":"2024-10-17T00:09:59.093+05:30"},"s":"I",  "c":"CONTROL",  "id":4784928, "ctx":"initandlisten","msg":"Shutting down the TTL monitor"}
{"t":{"$date":"2024-10-17T00:09:59.094+05:30"},"s":"I",  "c":"CONTROL",  "id":6278511, "ctx":"initandlisten","msg":"Shutting down the Change Stream Expired Pre-images Remover"}
{"t":{"$date":"2024-10-17T00:09:59.098+05:30"},"s":"I",  "c":"CONTROL",  "id":4784929, "ctx":"initandlisten","msg":"Acquiring the global lock for shutdown"}
{"t":{"$date":"2024-10-17T00:09:59.099+05:30"},"s":"I",  "c":"-",        "id":4784931, "ctx":"initandlisten","msg":"Dropping the scope cache for shutdown"}
{"t":{"$date":"2024-10-17T00:09:59.099+05:30"},"s":"I",  "c":"CONTROL",  "id":20565,   "ctx":"initandlisten","msg":"Now exiting"}
{"t":{"$date":"2024-10-17T00:09:59.100+05:30"},"s":"I",  "c":"CONTROL",  "id":8423404, "ctx":"initandlisten","msg":"mongod shutdown complete","attr":{"Summary of time elapsed":{"Statistics":{"Enter terminal shutdown":"0 ms","Step down th
e replication coordinator for shutdown":"0 ms","Time spent in quiesce mode":"0 ms","Shut down FLE Crud subsystem":"0 ms","Shut down MirrorMaestro":"0 ms","Shut down WaitForMajorityService":"0 ms","Shut down the global connection pool":"0
 ms","Shut down the flow control ticket holder":"2 ms","Shut down the replica set monitor":"0 ms","Shut down the migration util executor":"1 ms","Shut down the transport layer":"1 ms","Shut down the TTL monitor":"0 ms","Shut down expired
 pre-images and documents removers":"1 ms","Wait for the oplog cap maintainer thread to stop":"0 ms","Shut down full-time data capture":"0 ms","Shut down online certificate status protocol manager":"0 ms","shutdownTask total elapsed time
":"12 ms"}}}}
{"t":{"$date":"2024-10-17T00:09:59.100+05:30"},"s":"I",  "c":"CONTROL",  "id":23138,   "ctx":"initandlisten","msg":"Shutting down","attr":{"exitCode":100}}

C:\Users\sasiv\express-crud>mkdir models

C:\Users\sasiv\express-crud>mkdir routes

C:\Users\sasiv\express-crud>code .

C:\Users\sasiv\express-crud>_
```



```
found 0 vulnerabilities

C:\Users\sasiv\express-crud>npm install -g nodemon

changed 29 packages, and audited 30 packages in 2s

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\sasiv\express-crud>mongod
{"t":{"$date":"2024-10-17T00:09:56.872+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-10-17T00:09:59.080+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}
{"t":{"$date":"2024-10-17T00:09:59.080+05:30"},"s":"I",  "c":"NETWORK",  "id":4648601, "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related parameters","attr":{"relatedPar
ameters":["tcpFastOpenServer","tcpFastOpenClient","tcpFastOpenQueueSize"]}}
{"t":{"$date":"2024-10-17T00:09:59.081+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25},"incomingInte
rnalClient":{"minWireVersion":0,"maxWireVersion":25},"outgoing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient":true}}}
{"t":{"$date":"2024-10-17T00:09:59.083+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
{"t":{"$date":"2024-10-17T00:09:59.083+05:30"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":8488,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"DESKTOP-KHTQFO3"}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"8.0.1","gitVersion":"fcbe67d668fff5a370e2d87b9b1f74bc11bb7b94","modules":[],"
allocator":"tcmalloc-gperf","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19045)"}}}
{"t":{"$date":"2024-10-17T00:09:59.084+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{}}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"E",  "c":"CONTROL",  "id":20557,   "ctx":"initandlisten","msg":"DBException in initAndListen, terminating","attr":{"error":"NonExistentPath: Data directory \\data\\db not found. Create
the missing directory or specify another path using (1) the --dbpath command line option, or (2) by adding the 'storage.dbPath' option in the configuration file."}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"REPL",     "id":4784900, "ctx":"initandlisten","msg":"Stepping down the ReplicationCoordinator for shutdown","attr":{"waitTimeMillis":15000}}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"REPL",     "id":4794602, "ctx":"initandlisten","msg":"Attempting to enter quiesce mode"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"-",        "id":6371601, "ctx":"initandlisten","msg":"Shutting down the FLE Crud thread pool"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"COMMAND",  "id":4784901, "ctx":"initandlisten","msg":"Shutting down the MirrorMaestro"}
{"t":{"$date":"2024-10-17T00:09:59.087+05:30"},"s":"I",  "c":"SHARDING", "id":4784902, "ctx":"initandlisten","msg":"Shutting down the WaitForMajorityService"}
{"t":{"$date":"2024-10-17T00:09:59.088+05:30"},"s":"I",  "c":"NETWORK",  "id":8314100, "ctx":"initandlisten","msg":"Shutdown: Closing listener sockets"}
{"t":{"$date":"2024-10-17T00:09:59.088+05:30"},"s":"I",  "c":"NETWORK",  "id":4784905, "ctx":"initandlisten","msg":"Shutting down the global connection pool"}
{"t":{"$date":"2024-10-17T00:09:59.089+05:30"},"s":"I",  "c":"CONTROL",  "id":4784906, "ctx":"initandlisten","msg":"Shutting down the FlowControlTicketholder"}
{"t":{"$date":"2024-10-17T00:09:59.090+05:30"},"s":"I",  "c":"-",        "id":20520,   "ctx":"initandlisten","msg":"Stopping further Flow Control ticket acquisitions."}
{"t":{"$date":"2024-10-17T00:09:59.090+05:30"},"s":"I",  "c":"NETWORK",  "id":4784918, "ctx":"initandlisten","msg":"Shutting down the ReplicaSetMonitor"}
{"t":{"$date":"2024-10-17T00:09:59.091+05:30"},"s":"I",  "c":"SHARDING", "id":4784921, "ctx":"initandlisten","msg":"Shutting down the MigrationUtilExecutor"}
{"t":{"$date":"2024-10-17T00:09:59.092+05:30"},"s":"I",  "c":"ASIO",     "id":22582,   "ctx":"MigrationUtil-TaskExecutor","msg":"Killing all outstanding egress activity."}
{"t":{"$date":"2024-10-17T00:09:59.092+05:30"},"s":"I",  "c":"NETWORK",  "id":20562,   "ctx":"initandlisten","msg":"Shutdown: Closing open transport sessions"}
{"t":{"$date":"2024-10-17T00:09:59.093+05:30"},"s":"I",  "c":"NETWORK",  "id":4784923, "ctx":"initandlisten","msg":"Shutting down the ASIO transport SessionManager"}
{"t":{"$date":"2024-10-17T00:09:59.093+05:30"},"s":"I",  "c":"CONTROL",  "id":4784928, "ctx":"initandlisten","msg":"Shutting down the TTL monitor"}
{"t":{"$date":"2024-10-17T00:09:59.094+05:30"},"s":"I",  "c":"CONTROL",  "id":6278511, "ctx":"initandlisten","msg":"Shutting down the Change Stream Expired Pre-images Remover"}
{"t":{"$date":"2024-10-17T00:09:59.098+05:30"},"s":"I",  "c":"CONTROL",  "id":4784929, "ctx":"initandlisten","msg":"Acquiring the global lock for shutdown"}
{"t":{"$date":"2024-10-17T00:09:59.099+05:30"},"s":"I",  "c":"-",        "id":4784931, "ctx":"initandlisten","msg":"Dropping the scope cache for shutdown"}
{"t":{"$date":"2024-10-17T00:09:59.099+05:30"},"s":"I",  "c":"CONTROL",  "id":20565,   "ctx":"initandlisten","msg":"Now exiting"}
{"t":{"$date":"2024-10-17T00:09:59.100+05:30"},"s":"I",  "c":"CONTROL",  "id":8423404, "ctx":"initandlisten","msg":"mongod shutdown complete","attr":{"Summary of time elapsed":{"Statistics":{"Enter terminal shutdown":"0 ms","Step down th
e replication coordinator for shutdown":"0 ms","Time spent in quiesce mode":"0 ms","Shut down FLE Crud subsystem":"0 ms","Shut down MirrorMaestro":"0 ms","Shut down WaitForMajorityService":"0 ms","Shut down the global connection pool":"0
 ms","Shut down the flow control ticket holder":"2 ms","Shut down the replica set monitor":"0 ms","Shut down the migration util executor":"1 ms","Shut down the transport layer":"1 ms","Shut down the TTL monitor":"0 ms","Shut down expired
 pre-images and documents removers":"1 ms","Wait for the oplog cap maintainer thread to stop":"0 ms","Shut down full-time data capture":"0 ms","Shut down online certificate status protocol manager":"0 ms","shutdownTask total elapsed time
":"12 ms"}}}}
{"t":{"$date":"2024-10-17T00:09:59.100+05:30"},"s":"I",  "c":"CONTROL",  "id":23138,   "ctx":"initandlisten","msg":"Shutting down","attr":{"exitCode":100}}

C:\Users\sasiv\express-crud>
```