

G Varun
IoT A
21011102033

Web Tech Lab 8 REST API using MongoDB, Express and NodeJS

index.js

```
const express = require("express");
const mongoose = require("mongoose");
const app = express();

mongoose.connect("mongodb://localhost:27017/college", {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(() => {
  console.log("Connection successful");
})
.catch((error) => {
  console.error("Error connecting to MongoDB:", error);
});

const userschema = new mongoose.Schema({
  name: String,
  email: String
});

const userModel = mongoose.model("students", userschema);

// Insert a new document into the "students" collection
const newStudent = new userModel({
  name: "John Doe",
  email: "john.doe@example.com"
});

newStudent.save()
.then(() => {
  console.log("New student added to the database");
})
.catch((error) => {
  console.error("Error adding new student:", error);
});

app.get("/users", (req, res) => {
  userModel.find({})
  .then((students) => {
    res.json(students);
  });
});
```

```

    })
    .catch((err) => {
      console.log(err);
      res.status(500).json({ error: "Internal server error" });
    });
  });
});

app.listen(3000, () => {
  console.log("Server listening on port 3000");
});

```

outputs:
before running

The screenshot shows the MongoDB Compass application running on localhost:27017. The 'Databases' tab is selected, displaying a list of databases on the left: 'admin', 'config', and 'local'. The main panel shows details for the 'admin' database, including its storage size (20.48 kB) and the number of collections (0). Below this, details for the 'config' database are shown (storage size: 4.10 kB, collections: 0). The 'local' database is partially visible at the bottom.

after running:

```

PS C:\Users\Admin\Express\mongoAPI> node index.js
(node:13684) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
Server listening on port 3000
Connection successful
New student added to the database

```

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/users'. The page content shows a JSON array representing a student record: [{"_id": "652f5d2c338300f05d2247be", "name": "John Doe", "email": "john.doe@example.com", "__v": 0}].

Databases

Search

admin

college

students

config

local

college.students

Documents Aggregations Schema Indexes Validation

Filter

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA

EXPORT DATA

```
_id: ObjectId('652f5d2c338300f05d2247be')
name: "John Doe"
email: "john.doe@example.com"
__v: 0
```