Index.js …

*const* port = process.env.PORT || 3000;

*const* express = *require*("express");

*const* connect = *require*("./data/connection");

*const* router = *require*("./data/routes");

*connect*();

*const* app = *express*();

app.*use*(express.*json*());

app.*use*("/api", router);

app.*listen*(port, () *=>* {

  console.*log*("listening on port " + port);

});

Package.json …

{

  "scripts": {

    "start": "nodemon index.js",

    "production": "NODE\_ENV=production nodemon index.js"

  },

  "dependencies": {

    "body-parser": "^1.20.0",

    "cookie-parser": "^1.4.6",

    "cors": "^2.8.5",

    "dotenv": "^16.0.3",

    "ejs": "^3.1.8",

    "express": "^4.18.1",

    "mongoose": "^6.7.0",

    "multer": "^1.4.5-lts.1"

  }

}

.env ….

DB\_URL = "mongodb://localhost:27017/YTStudentRestDb"

Connection.js …

*require*("dotenv").*config*();

*const* mongoose = *require*("mongoose");

*const* db\_url = process.env.DB\_URL;

*const* *connect* = *function* () {

  mongoose

    .*connect*(db\_url)

    .*then*((*val*) *=>* console.*log*("Database connection established"))

    .*catch*((*err*) *=>* console.*log*("Error on connecting database."));

};

module.exports = connect;

model.js …

*const* mongoose = *require*("mongoose");

*const* schema = new mongoose.*Schema*({

  name: { type: String, required: true },

  age: { type: Number, required: true },

});

module.exports = mongoose.*model*("Student", schema);

routes.js …

*const* express = *require*("express");

*const* router = express.*Router*();

*const* Student = *require*("./model");

router.*get*("/student/getall", async (*req*, *res*) *=>* {

  try {

*const* data = *await* Student.*find*();

    res.*status*(200).*json*(data);

  } catch (error) {

    res.*status*(500).*json*({ error: error.message });

  }

});

router.*get*("/student/getbyid/:id", async (*req*, *res*) *=>* {

  try {

*const* id = req.params.id;

*const* data = *await* Student.*findById*(id);

    res.*status*(200).*json*(data);

  } catch (error) {

    res.*status*(500).*json*({ error: error.message });

  }

});

router.*post*("/student/add", async (*req*, *res*) *=>* {

  try {

*const* student = new *Student*({ name: req.body.name, age: req.body.age });

*const* result = *await* student.*save*();

    res.*status*(200).*json*(result);

  } catch (error) {

    res.*status*(500).*json*({ error: error.message });

  }

});

router.*put*("/student/update/:id", async (*req*, *res*) *=>* {

  try {

*const* id = req.params.id;

*const* data = req.body;

*const* options = { new: true };

*const* result = *await* Student.*findByIdAndUpdate*(id, data, options);

    res.*status*(200).*json*(result);

  } catch (error) {

    res.*status*(500).*json*({ error: error.message });

  }

});

router.*delete*("/student/delete/:id", async (*req*, *res*) *=>* {

  try {

*const* id = req.params.id;

*const* result = *await* Student.*findByIdAndDelete*(id);

    res.*status*(200).*json*({ message: "Deleted successfully" });

  } catch (error) {

    res.*status*(500).*json*({ error: error.message });

  }

});

module.exports = router;