Installing Pacakages   
npm install express // node framework

npm install serve-favicon // api router favicon sets

npm install mongoose // mongoDB database schema-based solution to model your application data

npm install dotenv // environment variables

npm install corss // corss-origin-allow-all

npm install -D eslint prettier

npx install-peerdeps --dev eslint-config-airbnb-base

npm install -D eslint-config-prettier eslint-plugin-prettier  
  
database   
// external modules import

const mongoose = require("mongoose");

const connectionString = process.env.MONGO\_URI;

const connectDatabase = async () => {

try {

await mongoose

.connect(connectionString, {

useNewUrlParser: true,

useUnifiedTopology: true,

// useCreateIndex: true, // for mongoose 6.x

// useFindAndModify: false, // for mongoose 6.x

})

.then(() => {

console.log("Connected to MongoDB database successfully.");

})

.catch((error) => {

console.log("Error connecting to MongoDB: ", error.message);

});

} catch (error) {

console.log("Database connection error: ", error.message);

}

};

module.exports = connectDatabase;

App.js

// imports modules & dependencies

const express = require("express");

const env = require("dotenv");

const favicon = require("serve-favicon");

var path = require("path");

var cors = require("cors");

// imports routes, middleware, and configs

const todos = require("./src/routes/todos.route");

const { notFoundRoute, errorHandler } = require("./src/configs/errorHandler");

// loads environment variables from .env file

env.config();

// initializes express app

const app = express();

// application database connection establishment

const connectDatabase = require("./src/db/connect");

connectDatabase();

// corss-origin-allow-all

app.use(cors());

// sets favicon in routes

app.use(favicon(path.join(\_\_dirname, "public", "favicon.ico")));

// sets static folder

app.use(express.static(path.join(\_\_dirname, "public")));

// parse requests of content-type - application/json

app.use(express.json());

// parse requests of content-type - application/x-www-form-urlencoded

app.use(express.urlencoded({ extended: true }));

// sets default route

app.get("/", (req, res) => {

res.status(200).json({ message: "Welcome to TODO Node.js application backend." });

});

// todos api routes

app.use(process.env.APP\_API\_PREFIX, todos);

// 404 - not found error handler

app.use(notFoundRoute);

// error handler

app.use(errorHandler);

// app listens to defined port

app.listen(process.env.APP\_PORT, () => {

console.log("TODO-App backend server running on: " + process.env.APP\_BASE\_URL);

});

]

