**Module 3 Assignment: Data Access**

Taj Taylor

Arizona State University

IFT 458

Professor Sthapit

February 19, 2023

Link to repository: <https://github.com/ttaylo56/IFT-458-M3-Assignment-2>

Code:

app.js

const express = require('express');

const morgan = require('morgan');

const cors = require('cors');

const bodyParser = require('body-parser');

const app = express();

*// enable CORS for all origins*

app.use(cors());

*//This is CORS Ref:https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS*

const corsOptions = {

origin: 'http://localhost:8080'

};

app.use(cors(corsOptions));

*// 1) MIDDLEWARES Morgan is used for debugging*

*if* (process.env.NODE\_ENV === 'development') {

app.use(morgan('dev'));

}

*// 2) MIDDLEWARE json is used for injecting the body attribute in the pipeline*

app.use(bodyParser.json());

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

app.use((req, res, next) => {

console.log('Hello from the middleware 👋');

next();

});

*// 3) MIDDLE ROUTES loading*

const salesRouter = require('./routes/salesRoutes');

*// const userRouter = require('./routes/userRoutes');*

app.use('/api/v1/sales', salesRouter);

*// app.use('/api/v1/users', userRouter);*

*// define a simple GET route*

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

res.send('Hello, world!');

});

*// listen on port 6060 for incoming requests*

const port = 6060;

app.listen(port, () => {

console.log(`Server is listening on port ${port}`);

});

module.exports = app;

config.env

NODE\_ENV=development

PORT=8000

DBUSERNAME=sqlfamily

PASSWORD=sqlf@m1ly

DATABASE=AdventureWorks

Queries.sql

*-- SELECT*

*-- SalesLT.Customer.CustomerID, o.OrderDate, p.ProductName, od.Quantity*

*-- FROM*

*-- SalesLT.Customer c*

*-- JOIN Orders o ON c.CustomerID = o.CustomerID*

*-- JOIN OrderDetails od ON o.OrderID = od.OrderID*

*-- JOIN SalesLT.Product p ON od.ProductID = p.ProductID;*

*-- SELECT TABLE\_NAME*

*-- FROM INFORMATION\_SCHEMA.TABLES*

*-- WHERE TABLE\_TYPE = 'BASE TABLE'*

*-- ORDER BY TABLE\_NAME;*

SELECT c.CustomerID, c.FirstName, c.MiddleName, c.LastName, a.AddressLine1, a.City, a.StateProvince, p.ProductID, p.Name, p.ListPrice

FROM SalesLT.Customer c

JOIN SalesLT.CustomerAddress ca ON c.CustomerID = ca.CustomerID

JOIN SalesLT.Address a ON ca.AddressID = a.AddressID

JOIN SalesLT.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID

JOIN SalesLT.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID

JOIN SalesLT.Product p ON sod.ProductID = p.ProductID;N

server.js

const dotenv = require('dotenv');

dotenv.config({ path: './config.env' }); *// this line has to come first since*

*// we need to tell where is the config file.*

const app = require('./app');

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

app.listen(port, () => {

console.log(`App running on port ${port}...`);

});

db.config.js

const sqlConfig = {

server: "sqlservercentralpublic.database.windows.net",

user: process.env.DBUSERNAME,

password: process.env.PASSWORD,

database: process.env.DATABASE,

pool: {

idleTimeoutMillis: 60000

},

options: {

encrypted: true, *// for azure*

trustServerCertificate: false,

useUTC: true

}

}

module.exports = sqlConfig;

db.manager.js

const sql = require('mssql');

const dbConnection = require('./db.config');

console.log(dbConnection);

async function getSalesProducts() {

console.log(' Connecting to SQL....... Cloud Server');

let dbContext = *await* sql.connect(dbConnection);

console.log('The Databse connection was Successful');

console.log('Getting data');

let results = *await* dbContext.request()

.query(`

SELECT TOP(20)

productId,

name,

productNumber,

color

listPrice

FROM

salesLT.Product

`);

console.log(`Returned SQL results`);

*return* results;

}

*//Export*

module.exports = { getSalesProducts: getSalesProducts };

customersController.js

const db = require('../config/db.manager');

exports.getAllCustomers = function (req, res) {

const customers = db.getCustomers().then(results => {

console.log(results);

res.status(200).json({

status: 'successful',

data: results.recordsets[0]

}); *// send all the data*

});

};

exports.getCustomerById = function (req, res) {

const { id } = req.params; *// get id*

res.status(200).json({

status: 'not implemented'

});

};

exports.createCustomer = function (req, res) { *// must select the body to be raw->JSON in Postman*

const { body } = req;*// req.body (attribute)*

const { id } = req.params;*// get (attribute)*

res.status(200).json({

status: 'not implemented'

});

};

exports.updateCustomerById = function (req, res) { *// must select the body to be raw->JSON in Postman*

const { body } = req;*// req.body (attribute)*

const { id } = req.params;*// get id (attribute)*

res.status(200).json({

status: 'not implemented'

});

};

exports.deleteCustomerById = function (req, res) { *// must select the body to be raw->JSON in Postman*

const { body } = req;*// req.body (attribute)*

const { id } = req.params;*// get id*

res.status(200).json({

status: 'not implemented'

});

};

salesController.js

const db = require('../config/db.manager');

exports.getAllSales = function (req, res) {

const salesProduct = db.getSalesProducts().then(results => {

console.log(results);

res.status(200).json({

status: 'successful',

data: results.recordsets[0]

}); *// send all the data*

});

}

exports.getSalesByID = function (req, res) {

const { id } = req.params; *// get id*

res.status(200).json({

status: 'no implemented'

});

}

exports.createNewSales = function (req, res) { *// must select the body to be raw->JSON in Postman*

const { body } = req;*// req.body (attribute)*

const { id } = req.params;*// get (attribute)*

res.status(200).json({

status: 'no implemented'

});

}

exports.patchSalesById = function (req, res) { *// must select the body to be raw->JSON in Postman*

const { body } = req;*// req.body (attribute)*

const { id } = req.params;*// get id (attribute)*

res.status(200).json({

status: 'no implemented'

});

}

exports.deleteSalesByID = function (req, res) { *// must select the body to be raw->JSON in Postman*

const { body } = req;*// req.body (attribute)*

const { id } = req.params;*// get id*

res.status(200).json({

status: 'no implemented'

});

}

customersRoutes.js

const express = require('express');

const router = express.Router();

const customersController = require('../controllers/customersController');

router

.route('/')

.get(customersController.getAllCustomers)

.post(customersController.createCustomer);

router

.route('/:id')

.get(customersController.getCustomerById)

.patch(customersController.updateCustomerById)

.delete(customersController.deleteCustomerById);

module.exports = router;

salesRoutes.js

const express = require('express');

const router = express.Router();

const salesController = require('../controllers/salesController');

router

.route('/')

.get(salesController.getAllSales)

.post(salesController.createNewSales);

router

.route('/:id')

.get(salesController.getSalesByID)

.patch(salesController.patchSalesById)

.delete(salesController.deleteSalesByID);

module.exports = router;

package.json

{

"name": "lab2mssql",

"version": "1.0.0",

"description": "Node.js CRUD example with SQL Server (MSSQL)",

"main": "app.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

"start:dev": "nodemon server.js",

"start:prod": "NODE\_ENV=production nodemon server.js"

},

"keywords": [

"node",

"js",

"crud",

"sql",

"server",

"mssql",

"express",

"sequelize",

"rest",

"api"

],

"author": "Taj Taylor",

"license": "ISC",

"dependencies": {

"body-parser": "^1.20.1",

"cors": "^2.8.5",

"dotenv": "^16.0.3",

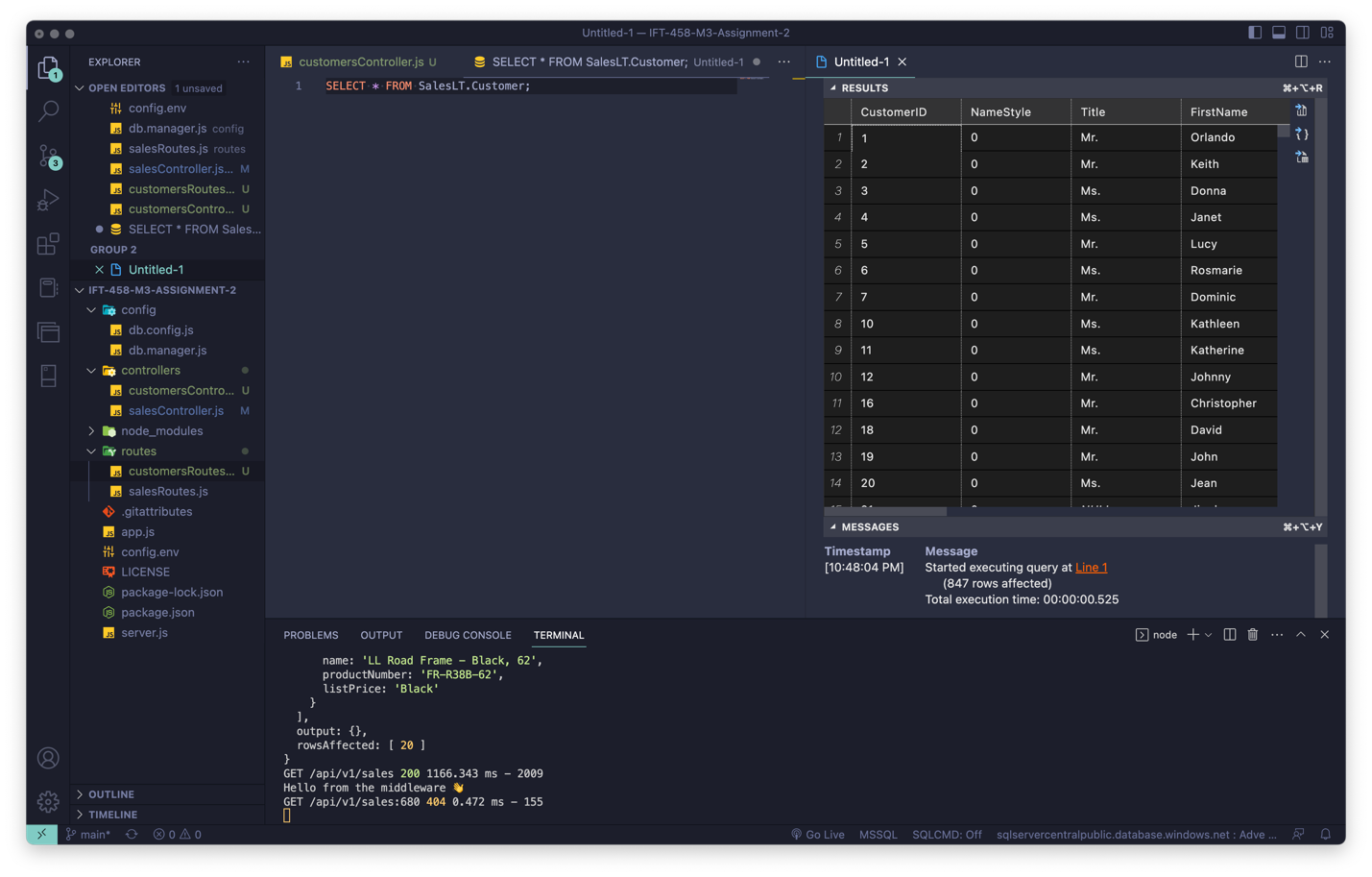
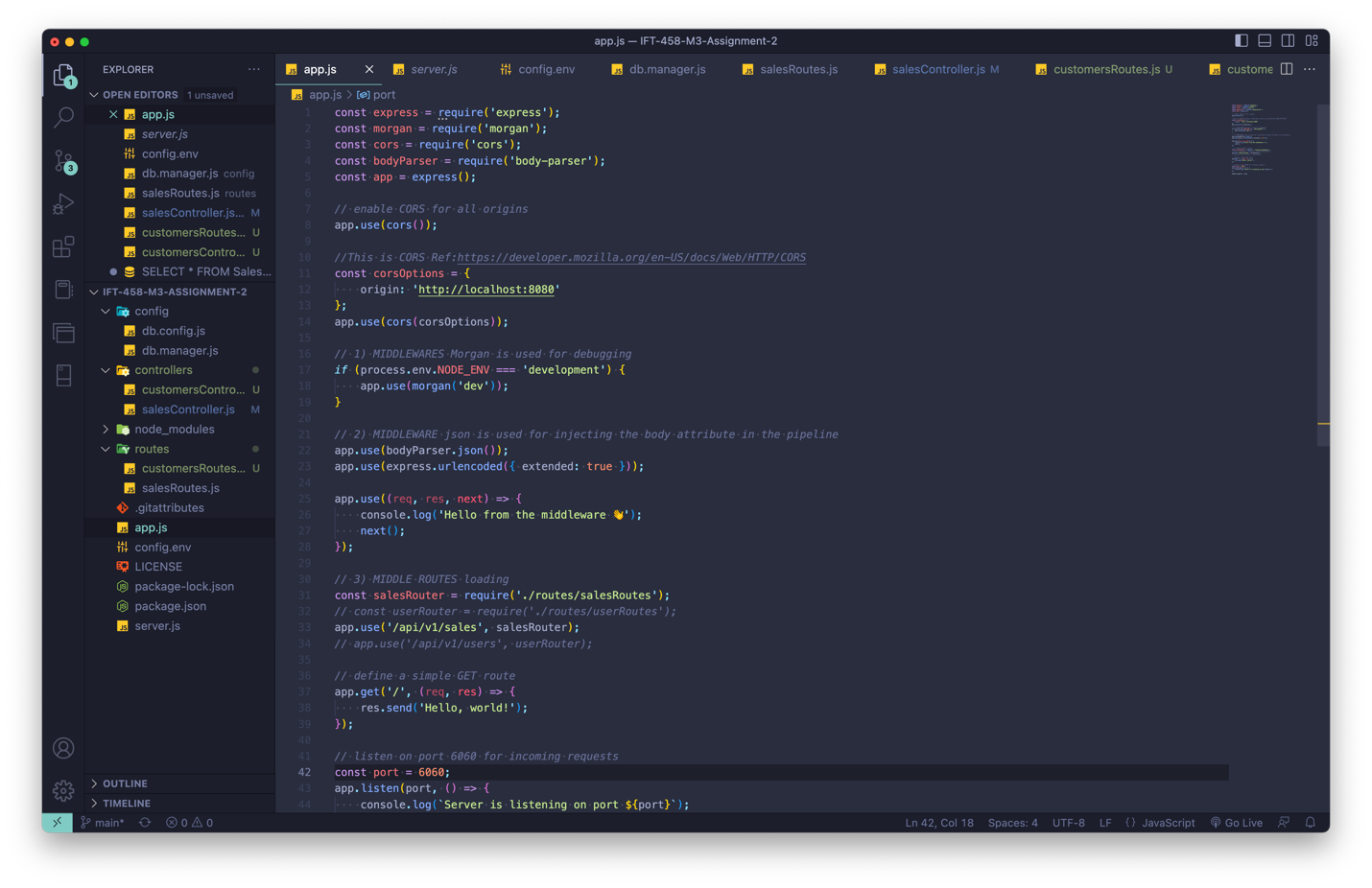
"express": "^4.18.2",

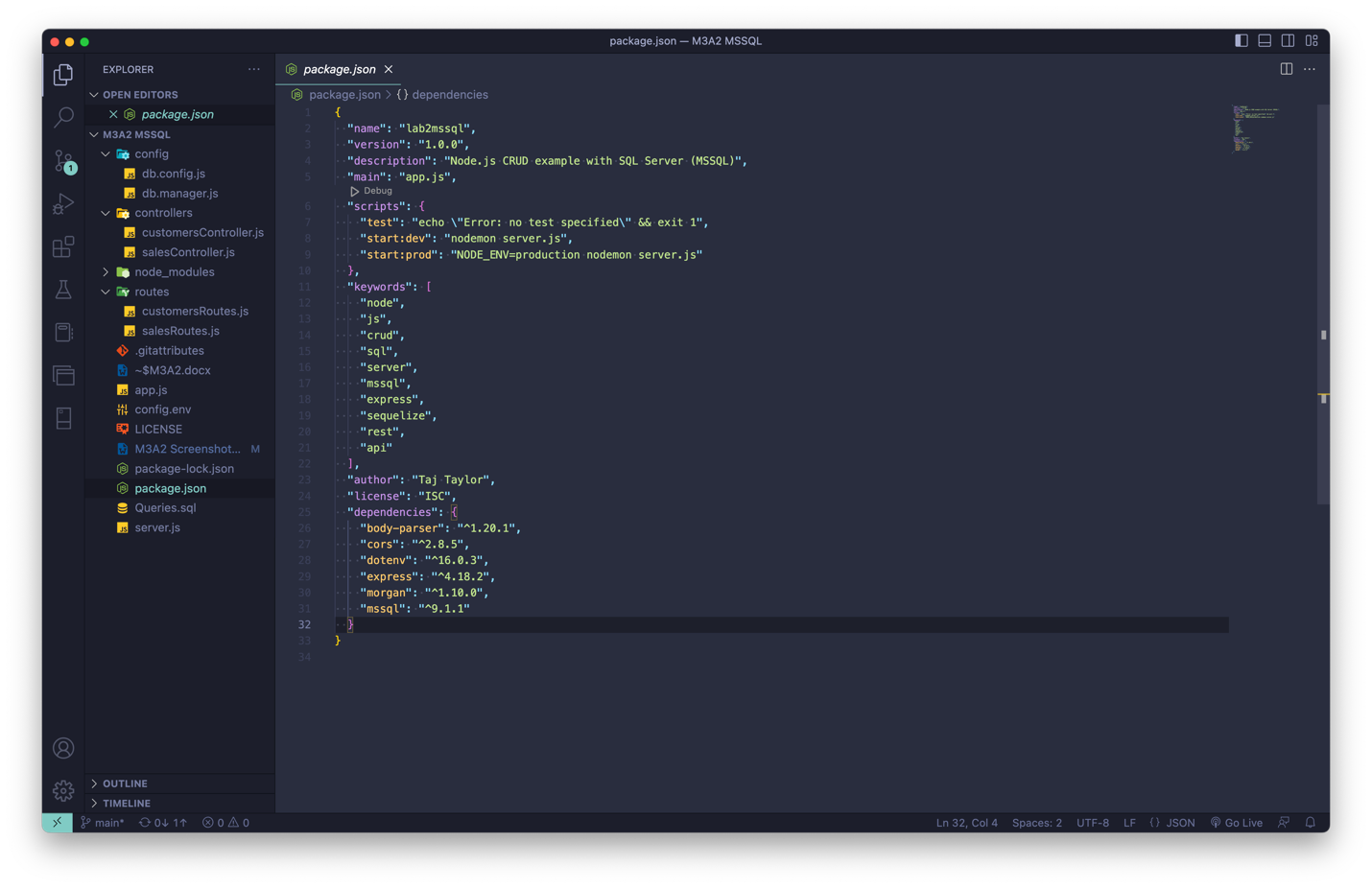
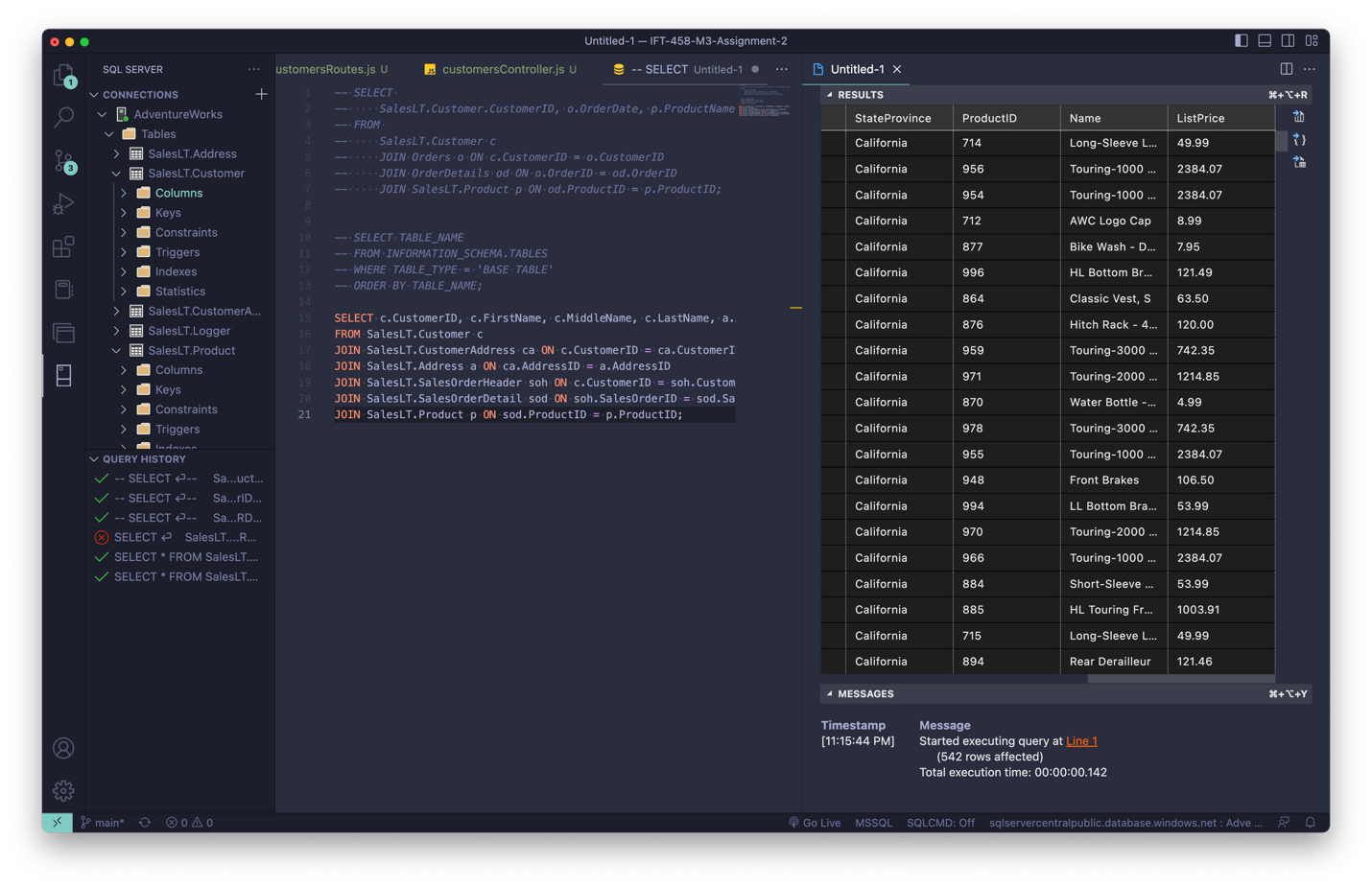
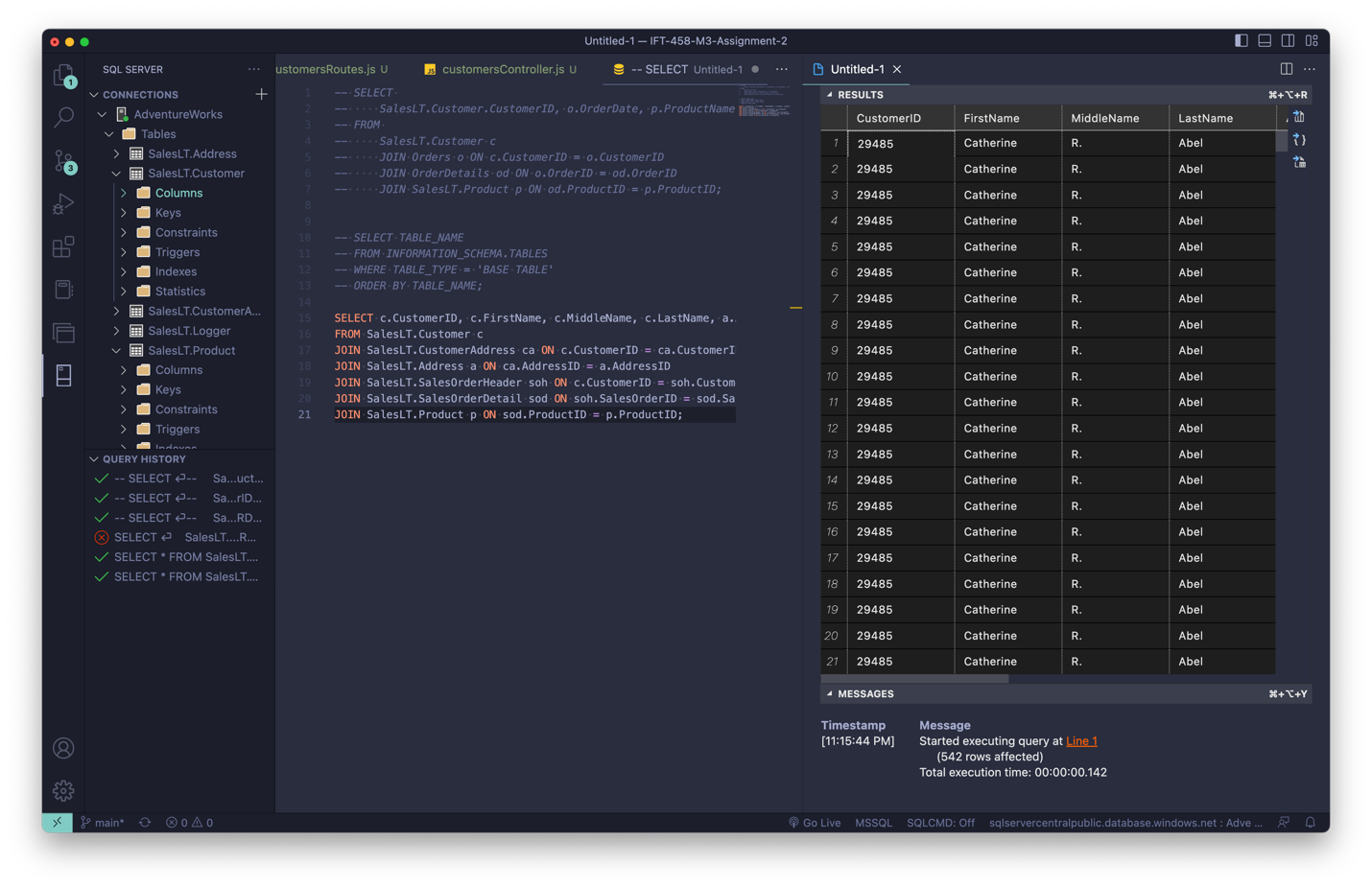
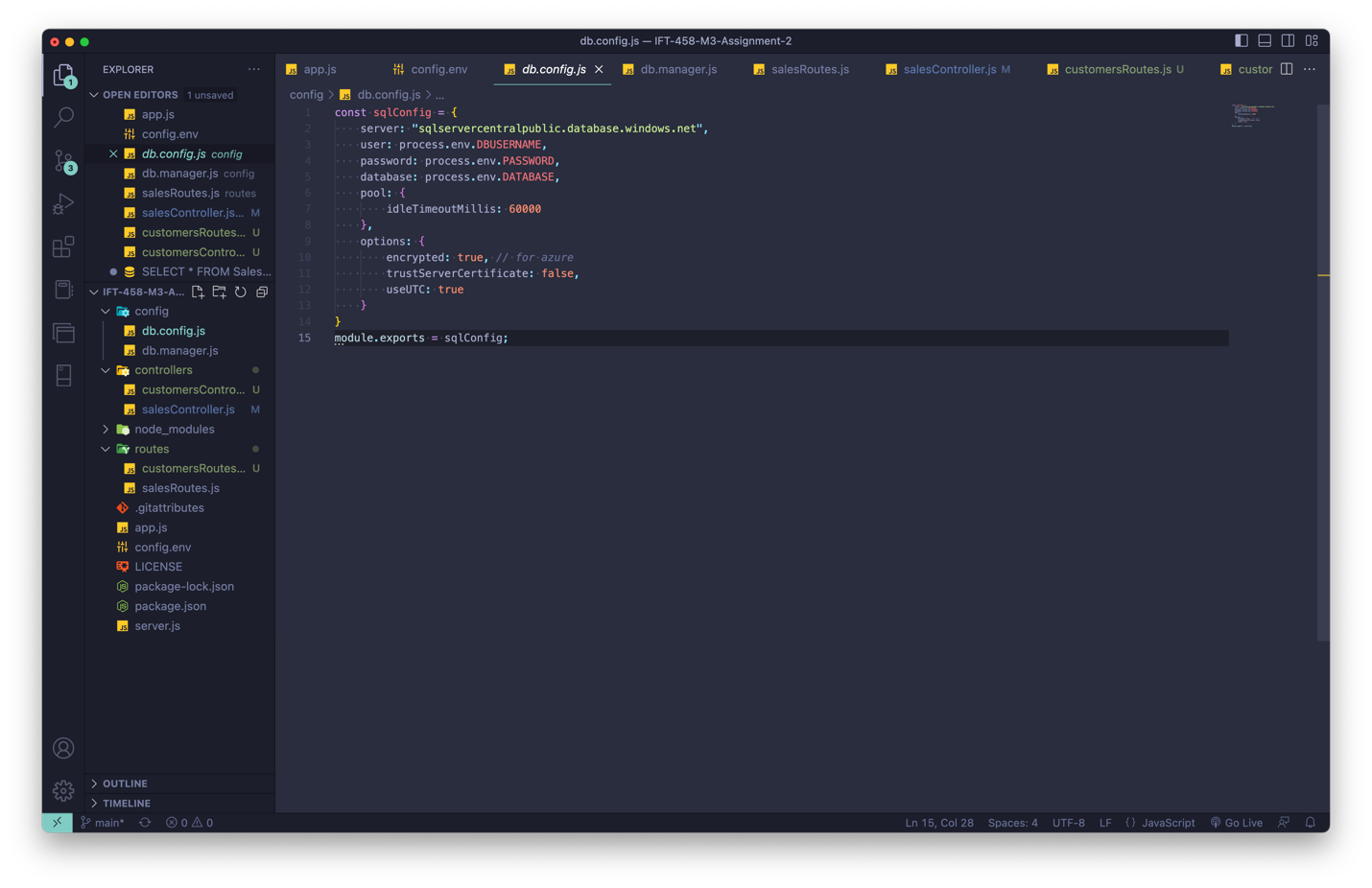
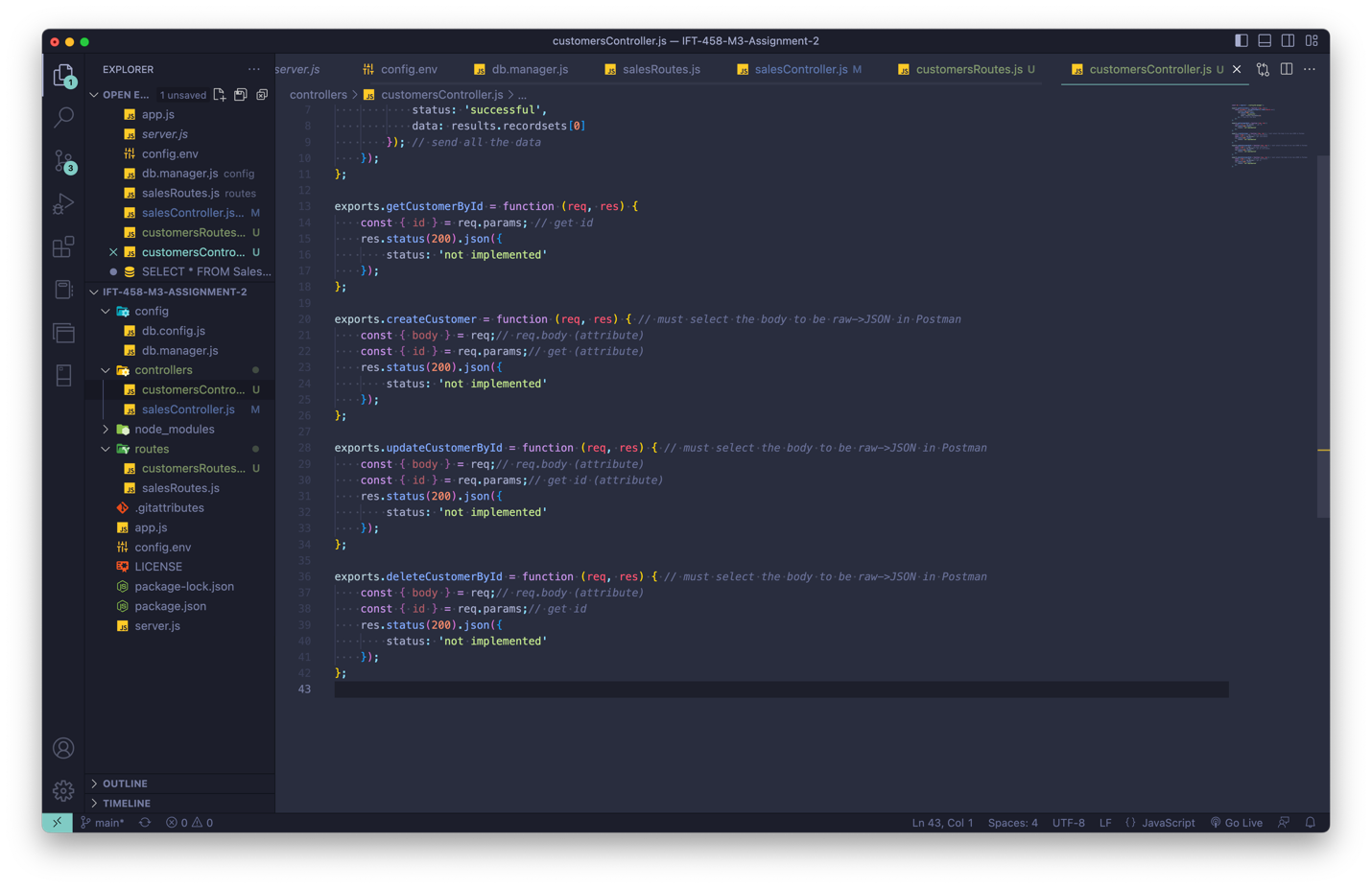
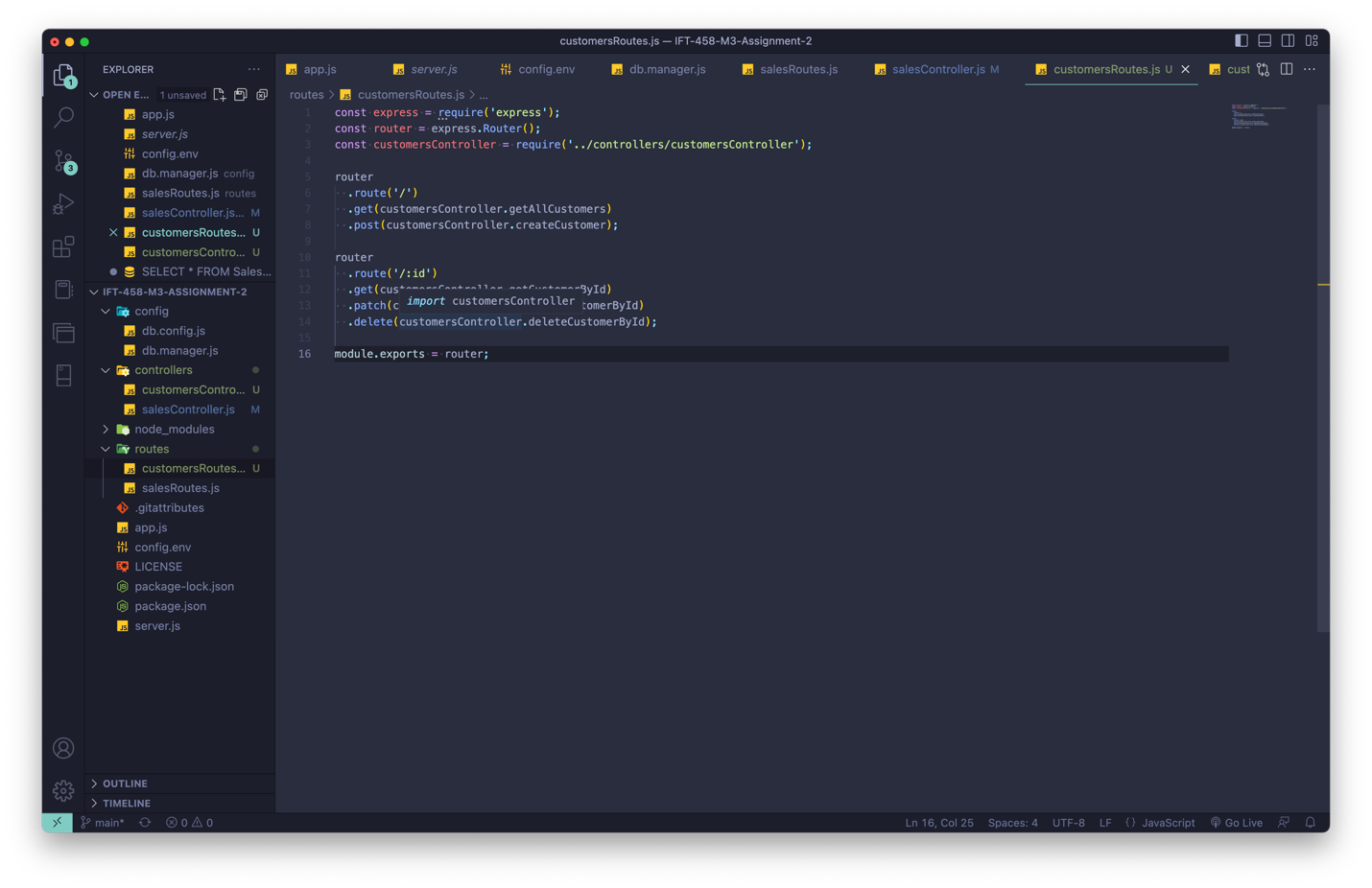
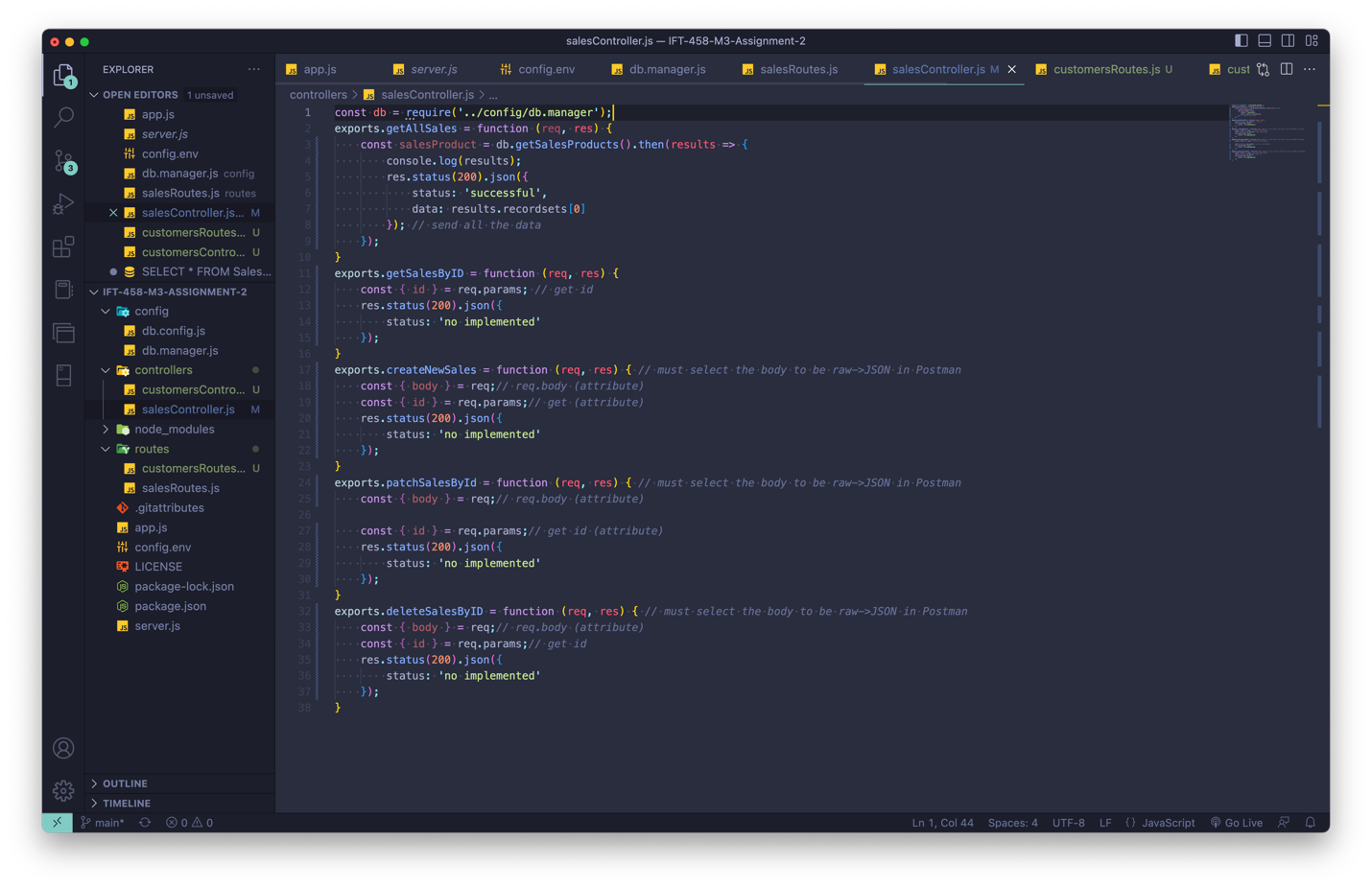
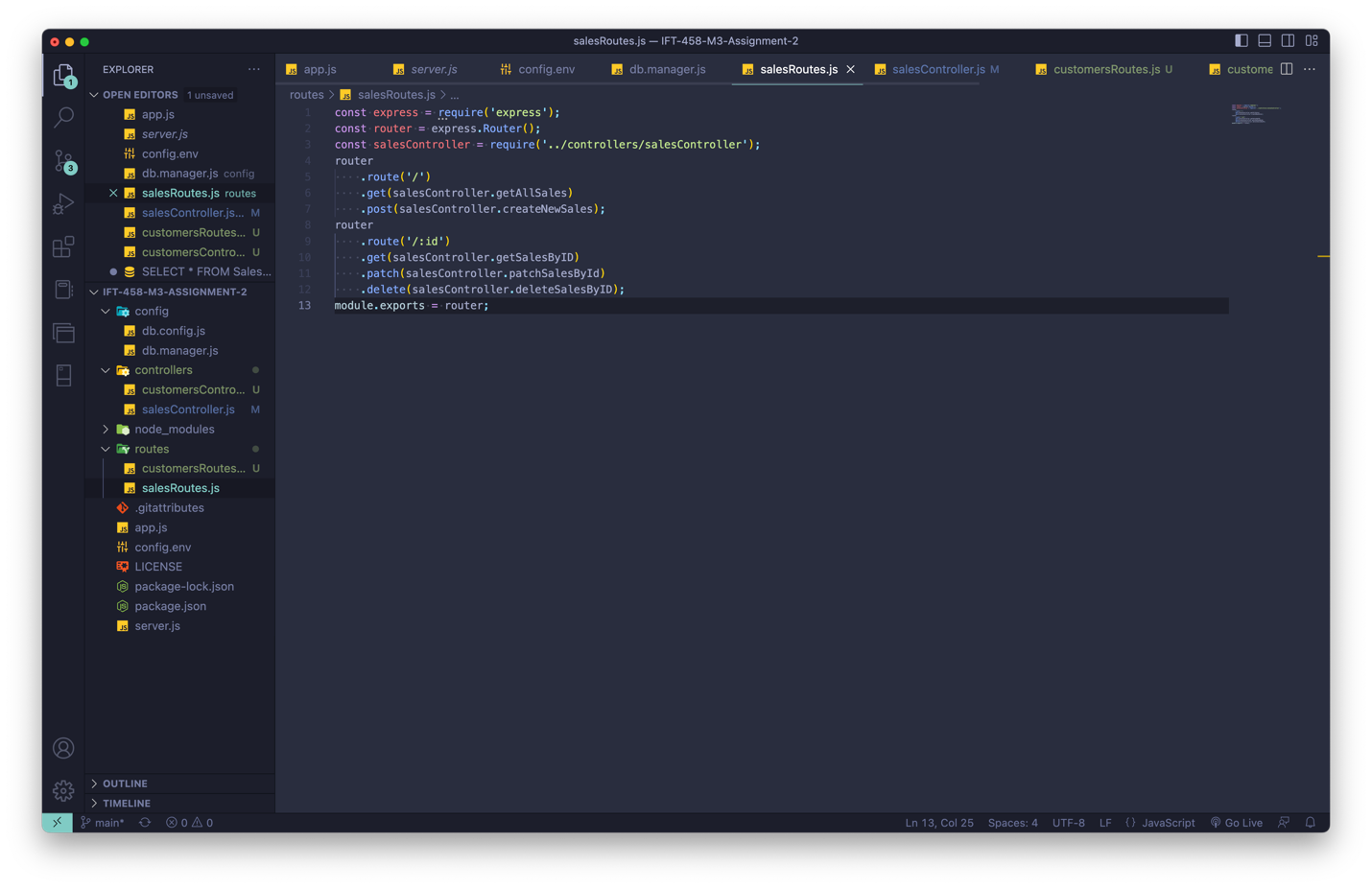
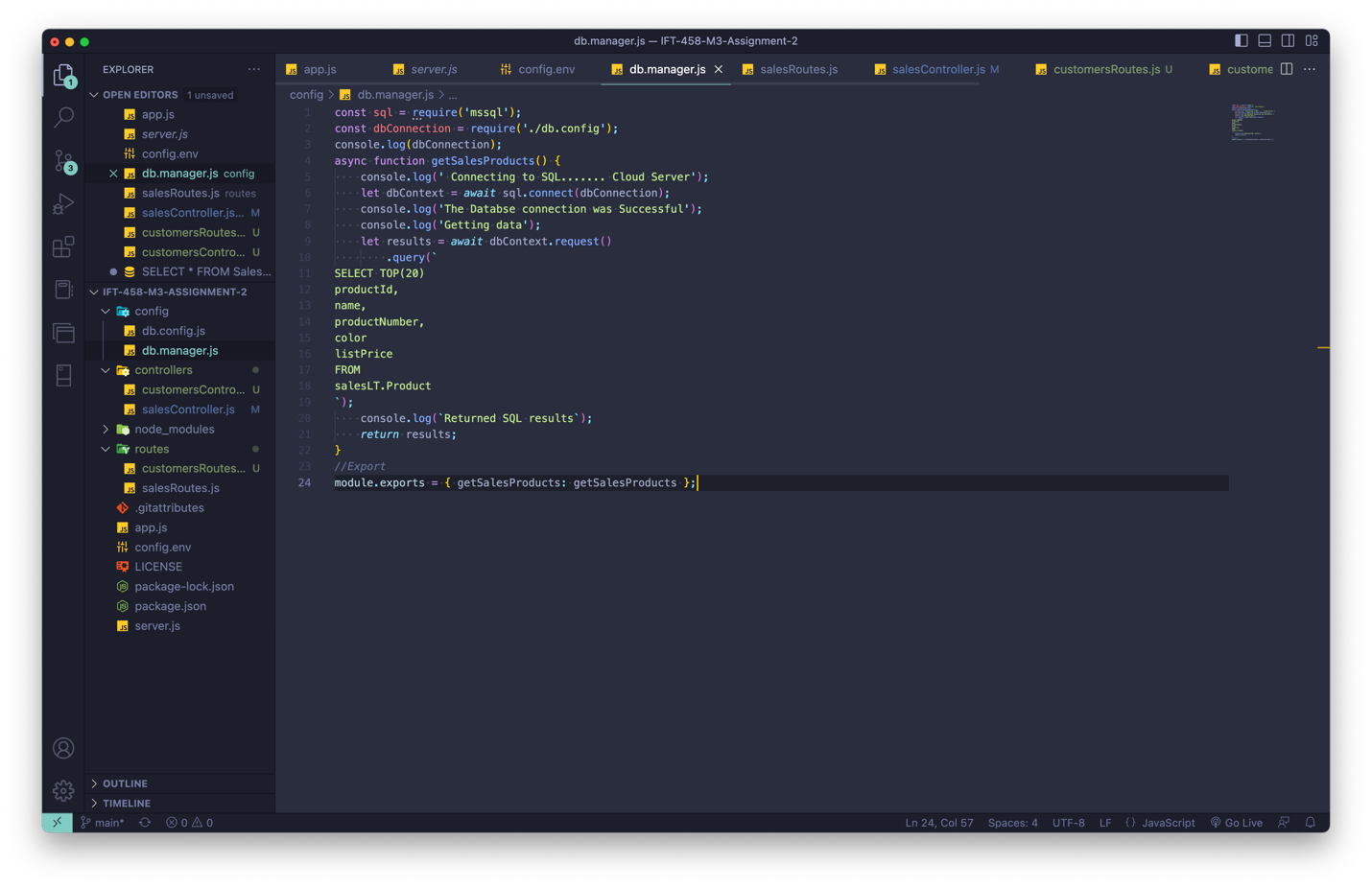
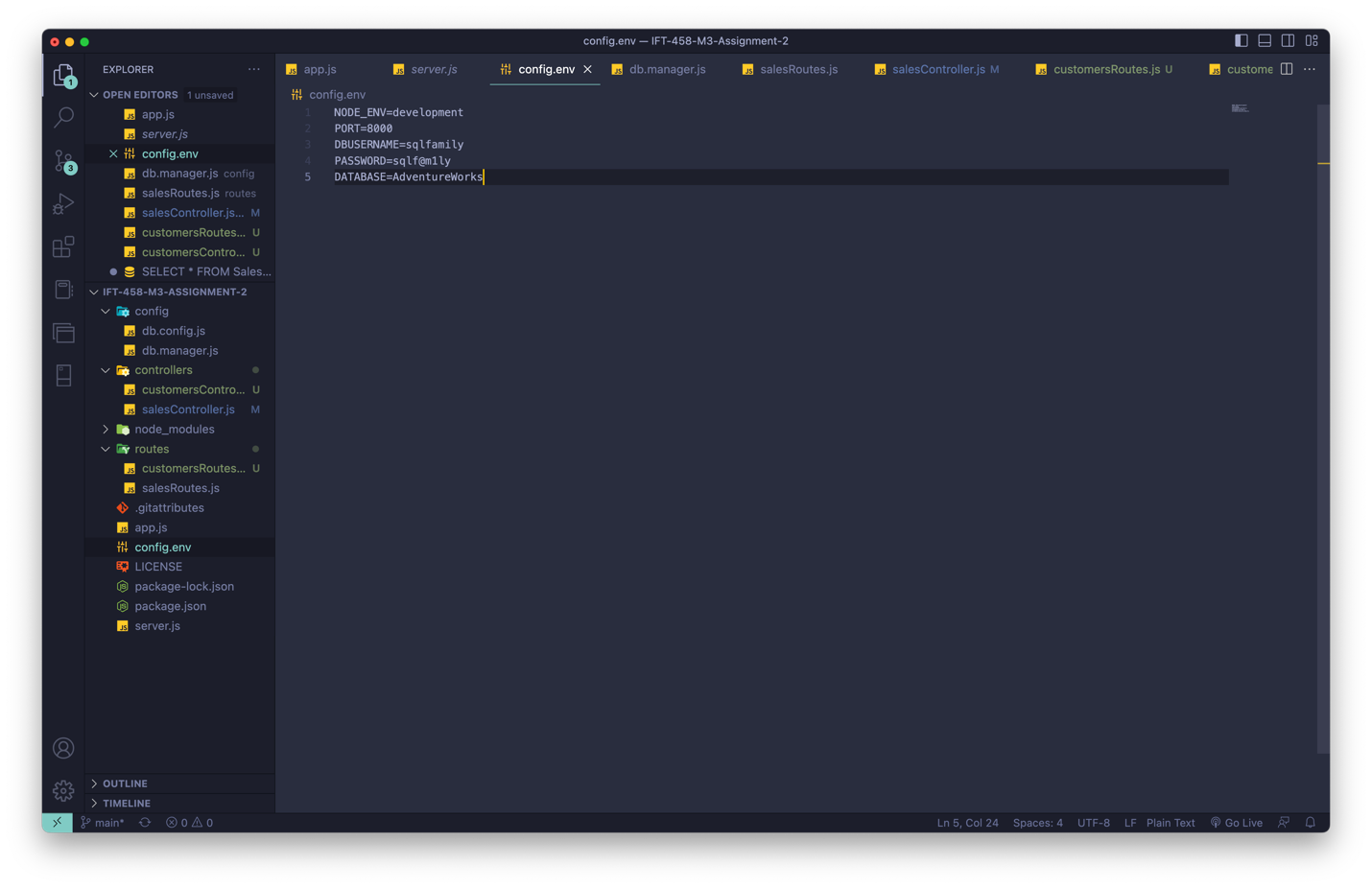
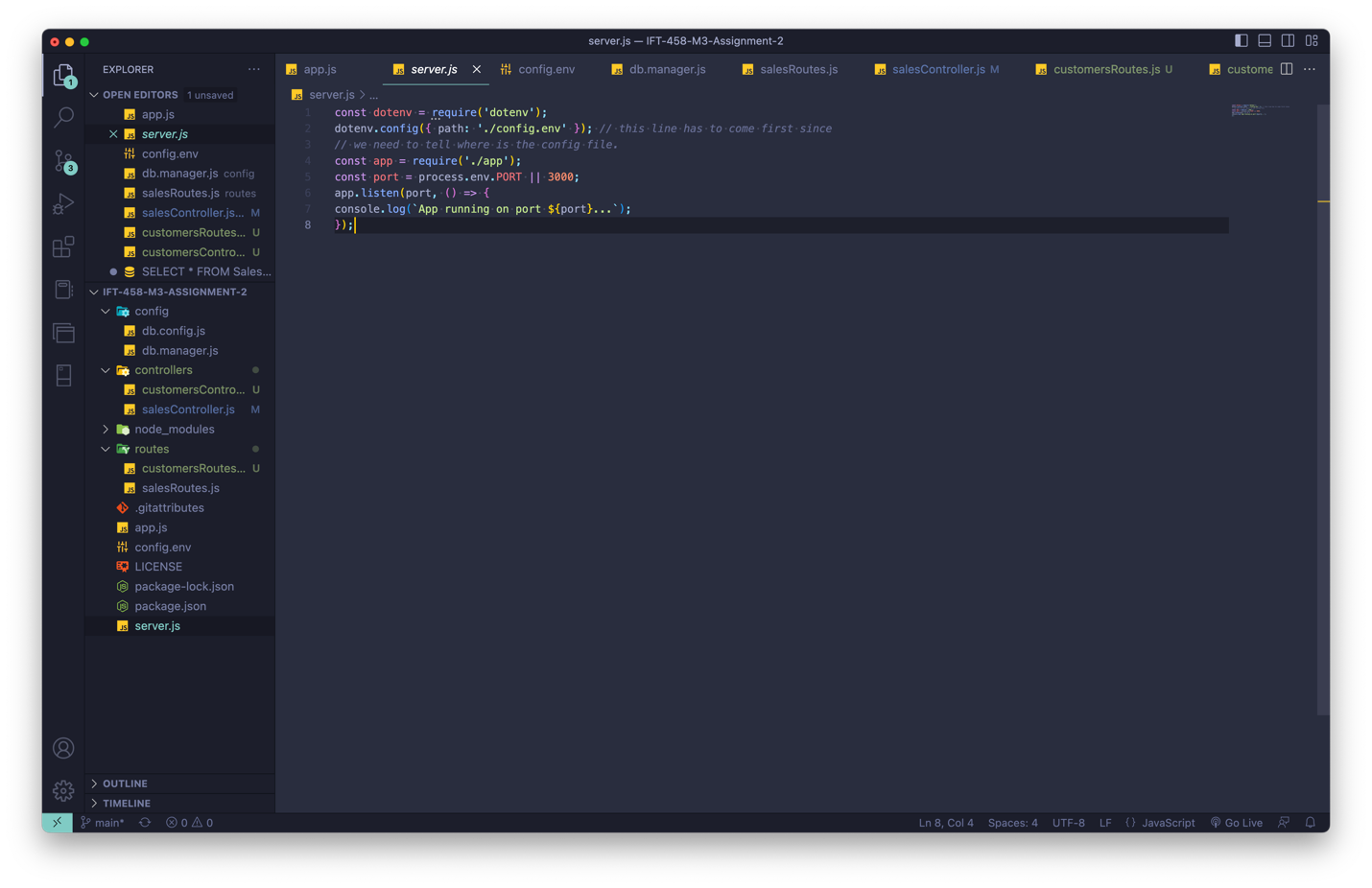
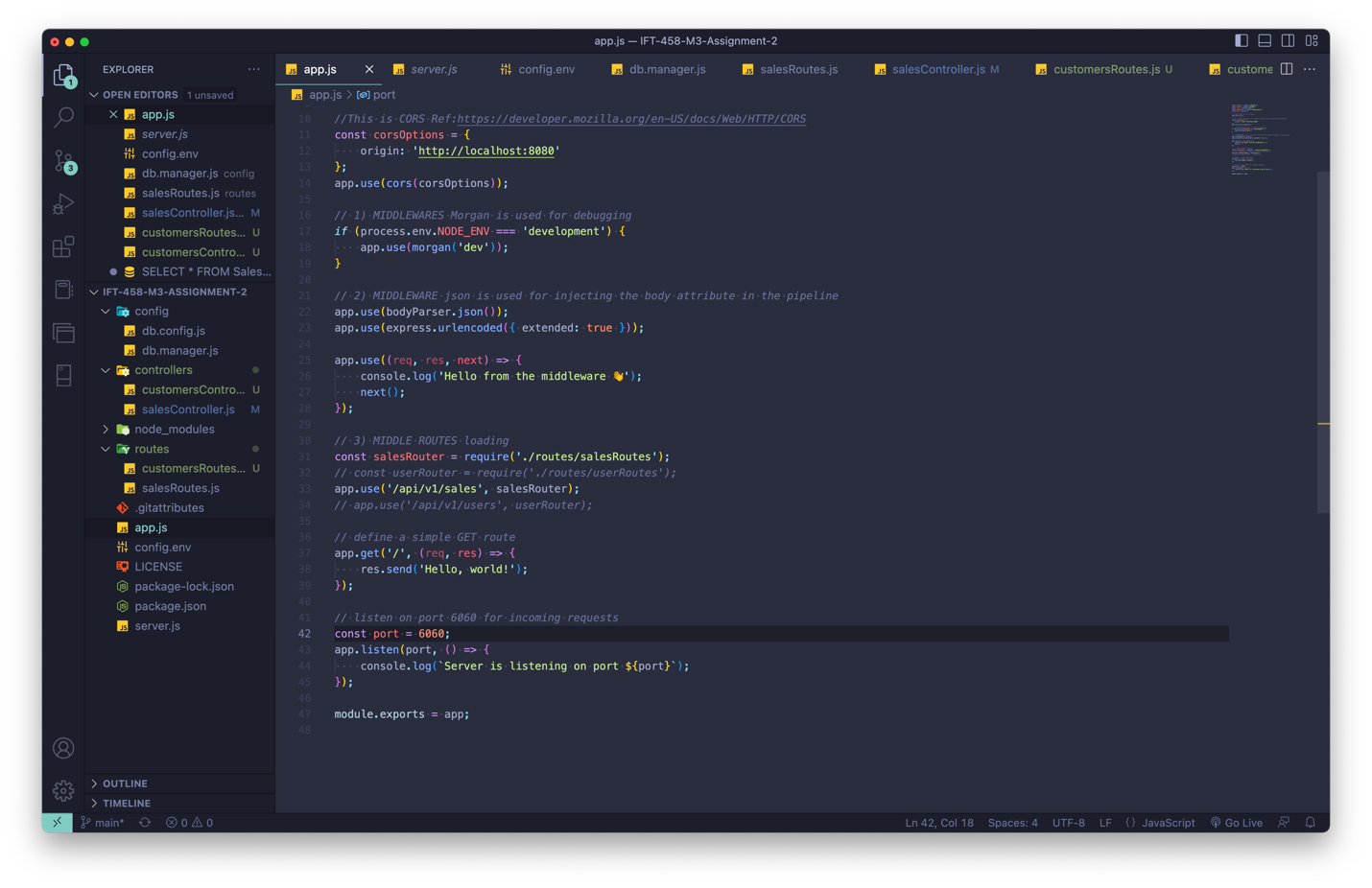
"morgan": "^1.10.0",

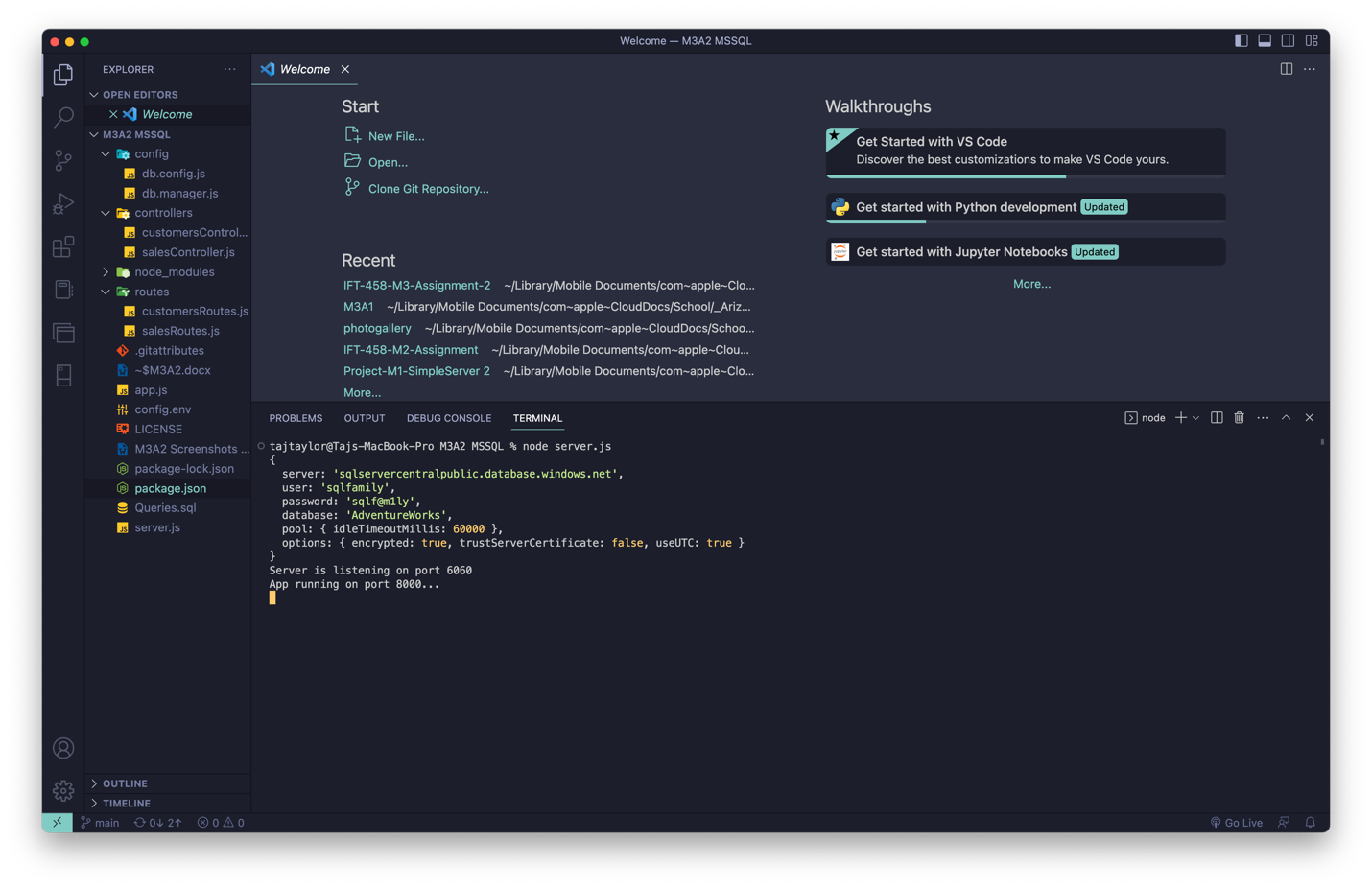
"mssql": "^9.1.1"

}

}

Screenshots: 

Graphical user interface, text

Description automatically generatedGraphical user interface, application

Description automatically generated