Module 3 Lab 1

Brandon Trinkle

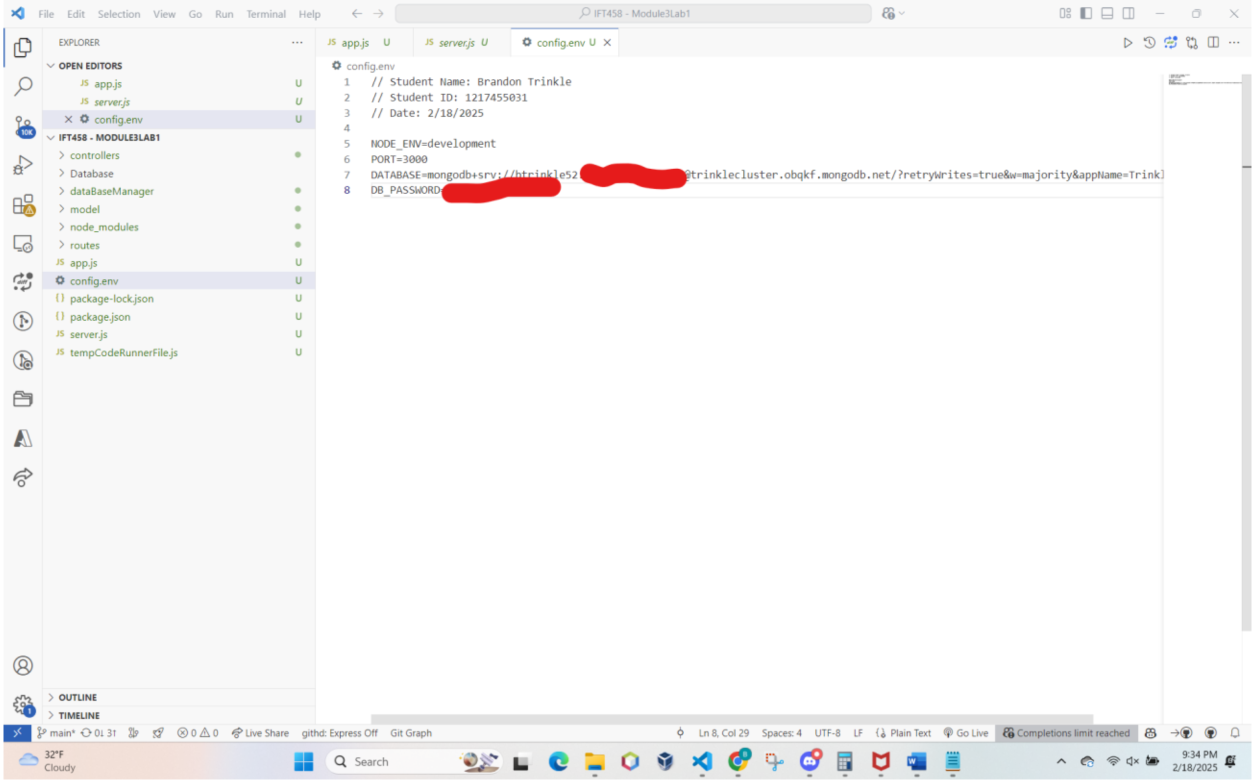
IFT 458: Middleware Prog & Database Sec

Professor: Dinesh Sthapit

Feb 18, 2025

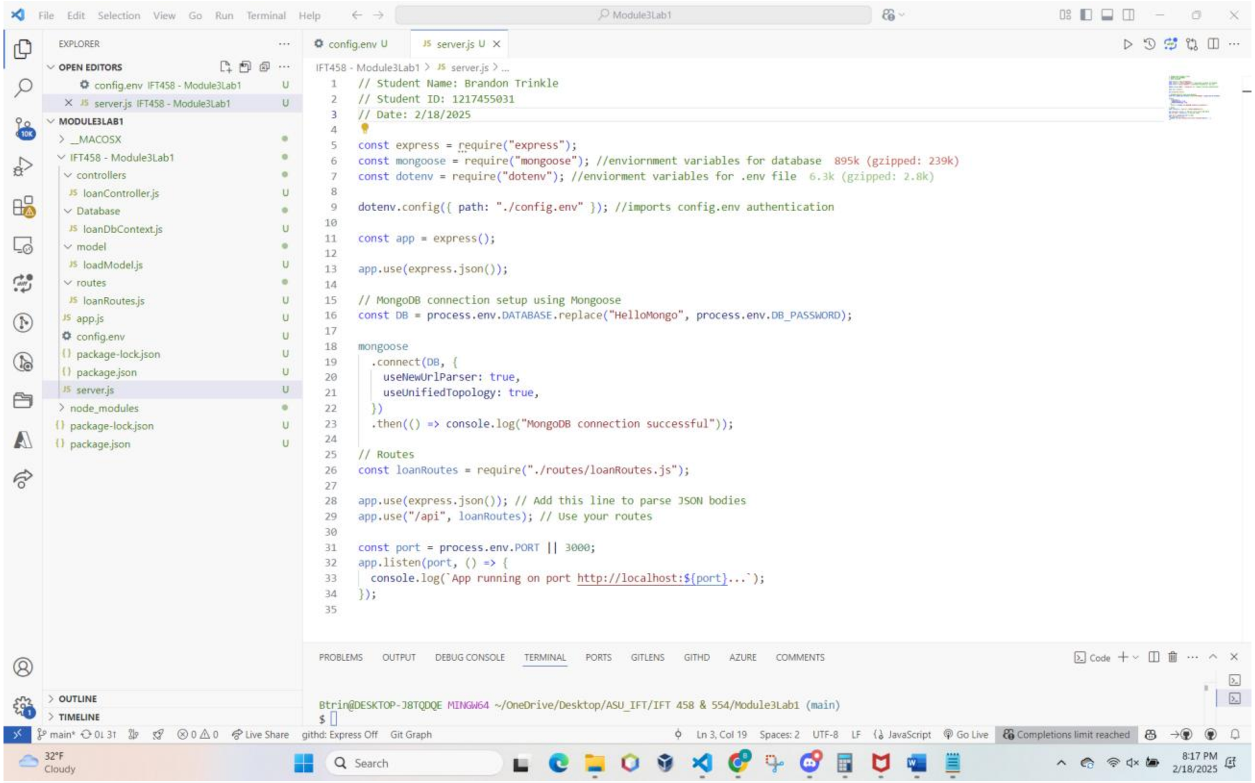
Module 3 Lab 1

1. *Screenshot of the config.env file showing the added environment variables.*



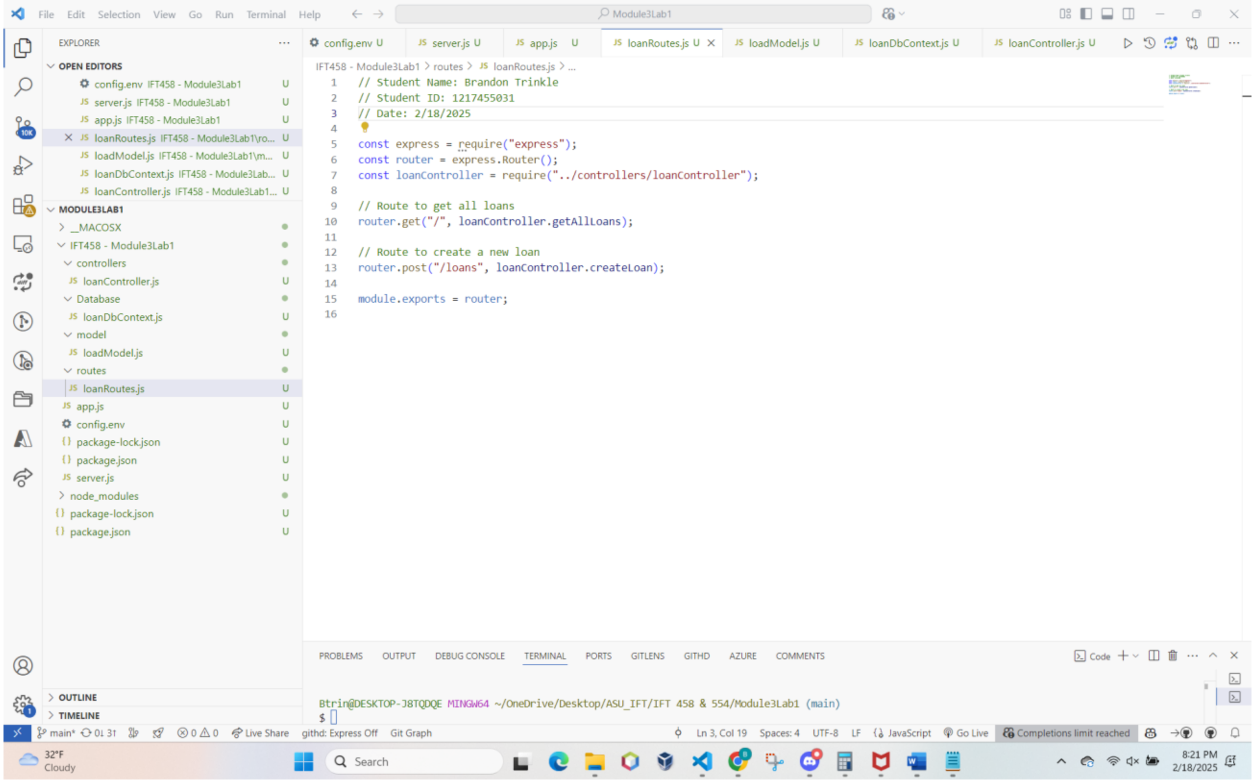
```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

NODE_ENV=development
PORT=3000
DATABASE=mongodb+srv://btrinkle52[redacted]@trinklecluster.obqkf.mongodb.net/?retryWrites=true&w=majority&appName=Trinkl
DB_PASSWORD=[redacted]
```

2. *Screenshots of the loanModel.js, loanController.js, loanRoutes.js, and server.js files showing the code and file structure. **Multiple screenshots, see deliverables.***



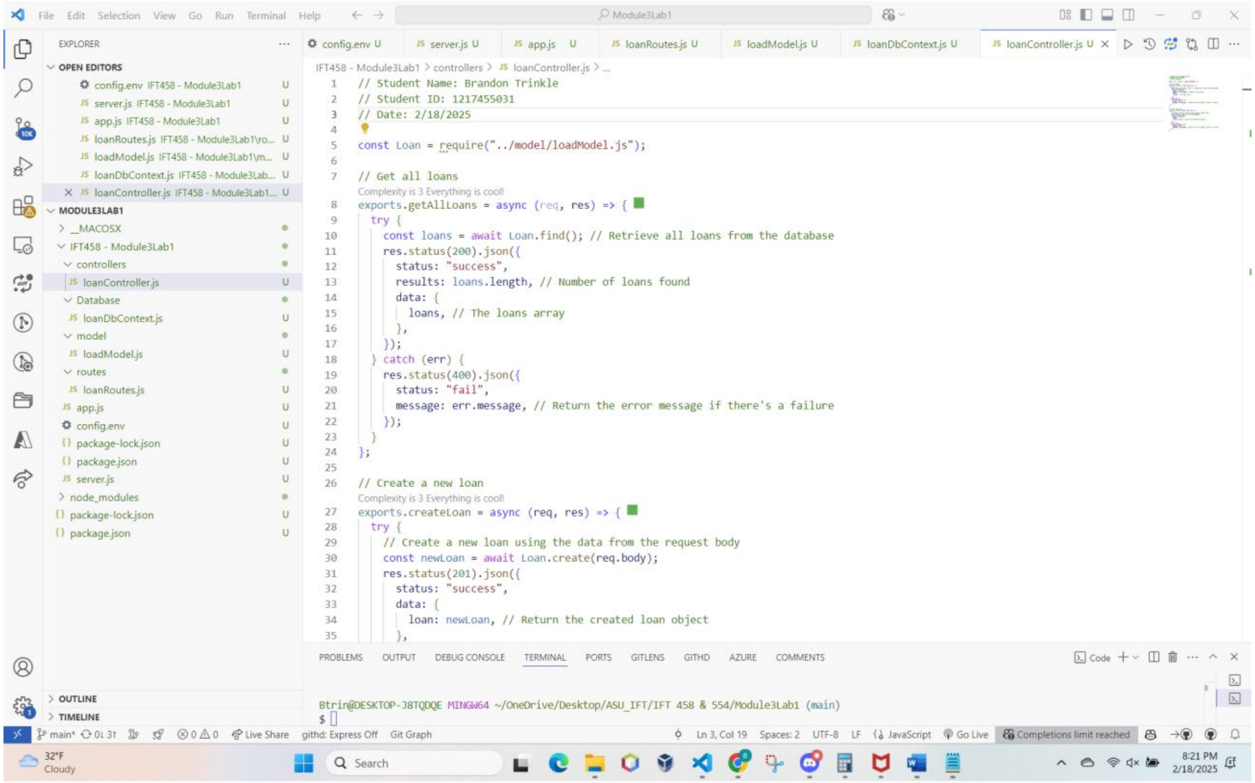
```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const express = require("express");
const mongoose = require("mongoose"); //enviornment variables for database  895k (gzipped: 239k)
const dotenv = require("dotenv"); //enviorment variables for .env file  6.3k (gzipped: 2.8k)

dotenv.config({ path: "./config.env" }); //imports config.env authentication

const app = express();

app.use(express.json());

// MongoDB connection setup using Mongoose
const DB = process.env.DATABASE.replace("HelloMongo", process.env.DB_PASSWORD);

mongoose
  .connect(DB, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then(() => console.log("MongoDB connection successful"));

// Routes
const loanRoutes = require("./routes/loanRoutes.js");

app.use(express.json()); // Add this line to parse JSON bodies
app.use("/api", loanRoutes); // Use your routes

const port = process.env.PORT || 3000;
app.listen(port, () => {
  console.log(`App running on port http://localhost:${port}...`);
});
```

```
Btrin@DESKTOP-J8TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1 (main)
$ 
```



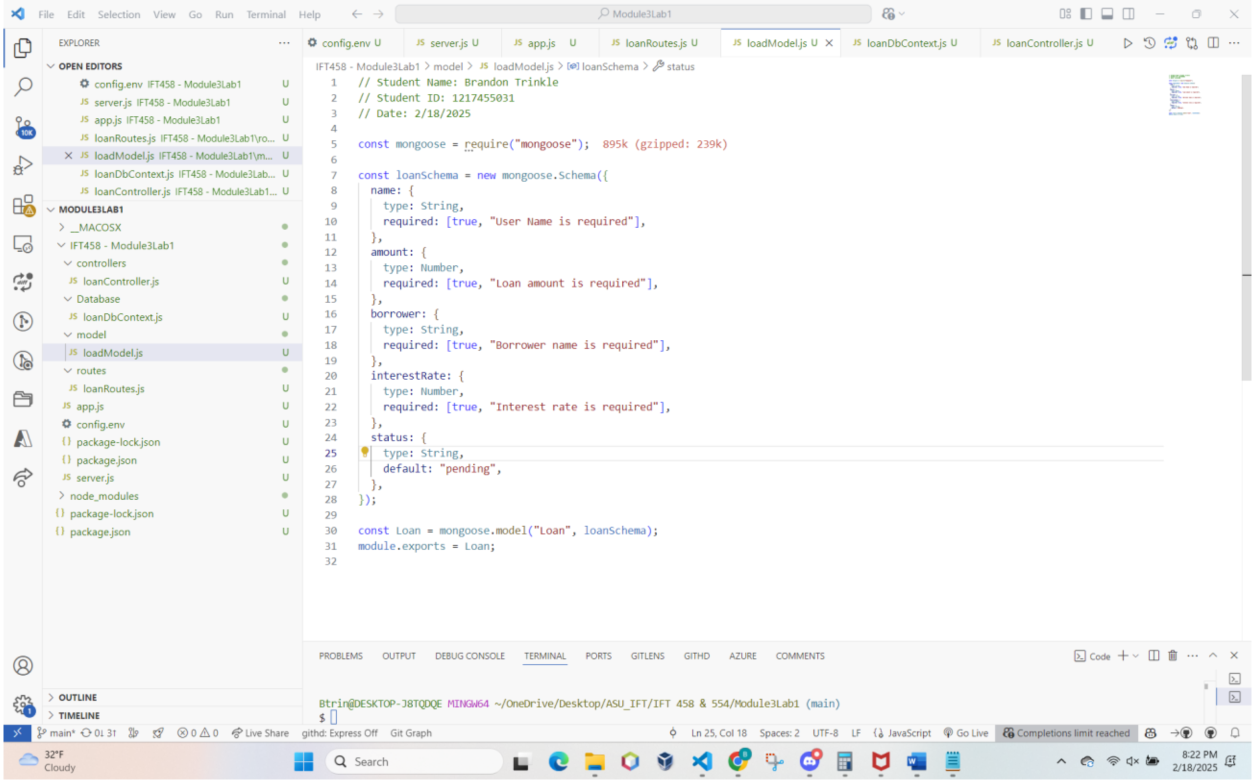
```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const express = require("express");
const router = express.Router();
const loanController = require("../controllers/loanController");

// Route to get all loans
router.get("/", loanController.getAllLoans);

// Route to create a new loan
router.post("/loans", loanController.createLoan);

module.exports = router;
```

```
Btrin@DESKTOP-J8TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1 (main)
$ 
```


IFT458 - Module3Lab1 > controllers > loanController.js > ...

```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const Loan = require("../model/loadModel.js");

// Get all loans
Complexity is 3 Everything is cool!
exports.getAllLoans = async (req, res) => {
  try {
    const loans = await Loan.find(); // Retrieve all loans from the database
    res.status(200).json({
      status: "success",
      results: loans.length, // Number of loans found
      data: {
        loans, // The loans array
      },
    });
  } catch (err) {
    res.status(400).json({
      status: "fail",
      message: err.message, // Return the error message if there's a failure
    });
  }
};

// Create a new loan
Complexity is 3 Everything is cool!
exports.createLoan = async (req, res) => {
  try {
    // Create a new loan using the data from the request body
    const newLoan = await Loan.create(req.body);
    res.status(201).json({
      status: "success",
      data: {
        loan: newLoan, // Return the created loan object
      },
```

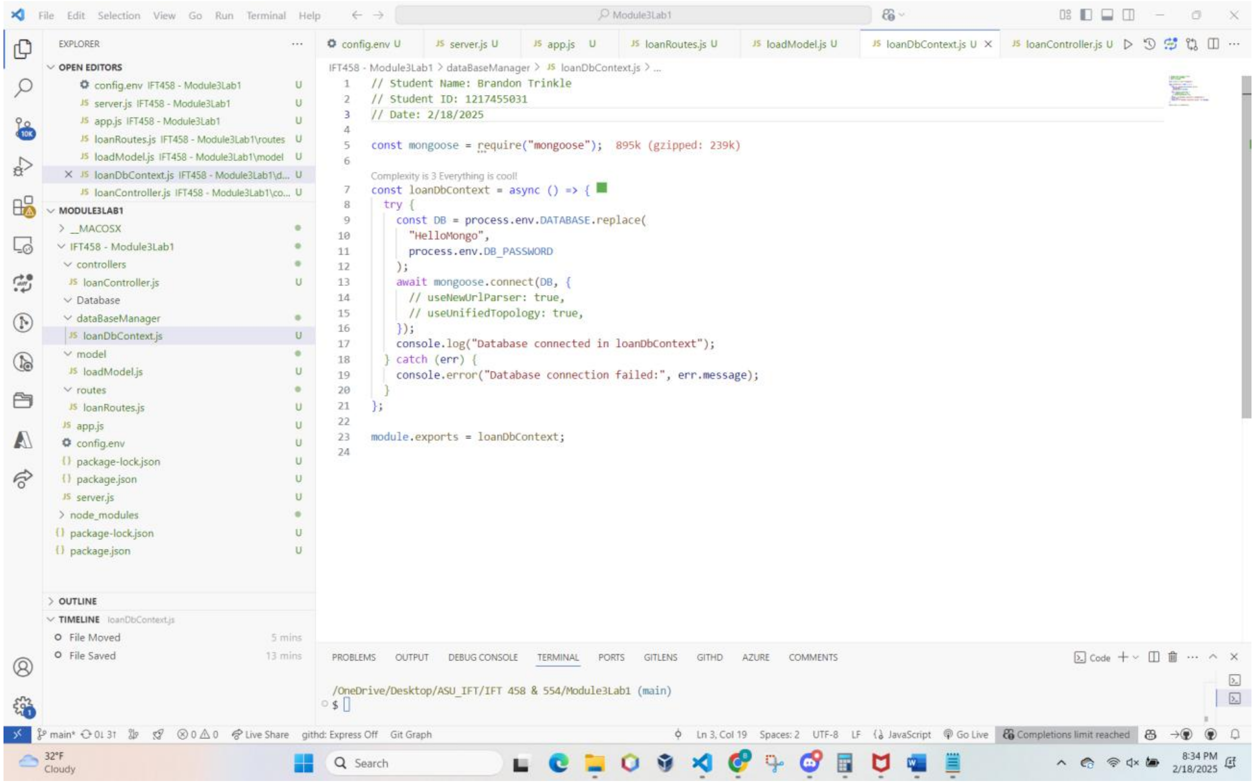
```
Btrin@DESKTOP-J8TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1 (main)
$
```


IFT458 - Module3Lab1 > model > loadModel.js > loanSchema > status

```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const mongoose = require("mongoose");  895k (gzipped: 239k)

const loanSchema = new mongoose.Schema({
  name: {
    type: String,
    required: [true, "User Name is required"],
  },
  amount: {
    type: Number,
    required: [true, "Loan amount is required"],
  },
  borrower: {
    type: String,
    required: [true, "Borrower name is required"],
  },
  interestRate: {
    type: Number,
    required: [true, "Interest rate is required"],
  },
  status: {
    type: String,
    default: "pending",
  },
});

const Loan = mongoose.model("Loan", loanSchema);
module.exports = Loan;
```

```
Btrin@DESKTOP-J8TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1 (main)
$
```

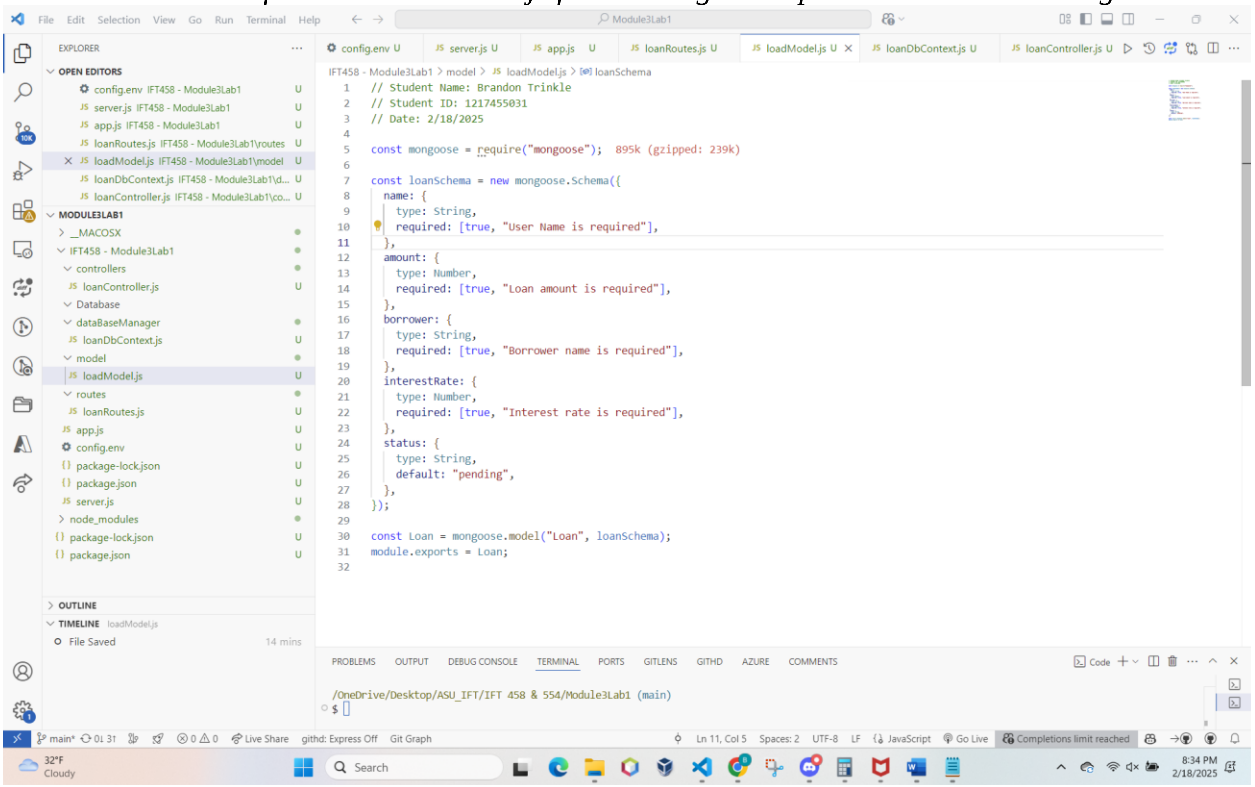
3. *Deliverable: Screenshot of the loanDbContext.js file within the dataBaseManager folder showing the modified code.*



```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const mongoose = require("mongoose");  895k (gzipped: 239k)

Complexity is 3 Everything is cool!
const loanDbContext = async () => {
  try {
    const DB = process.env.DATABASE.replace(
      "HelloMongo",
      process.env.DB_PASSWORD
    );
    await mongoose.connect(DB, {
      // useNewUrlParser: true,
      // useUnifiedTopology: true,
    });
    console.log("Database connected in loanDbContext");
  } catch (err) {
    console.error("Database connection failed:", err.message);
  }
};

module.exports = loanDbContext;
```

```
/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1 (main)
$
```

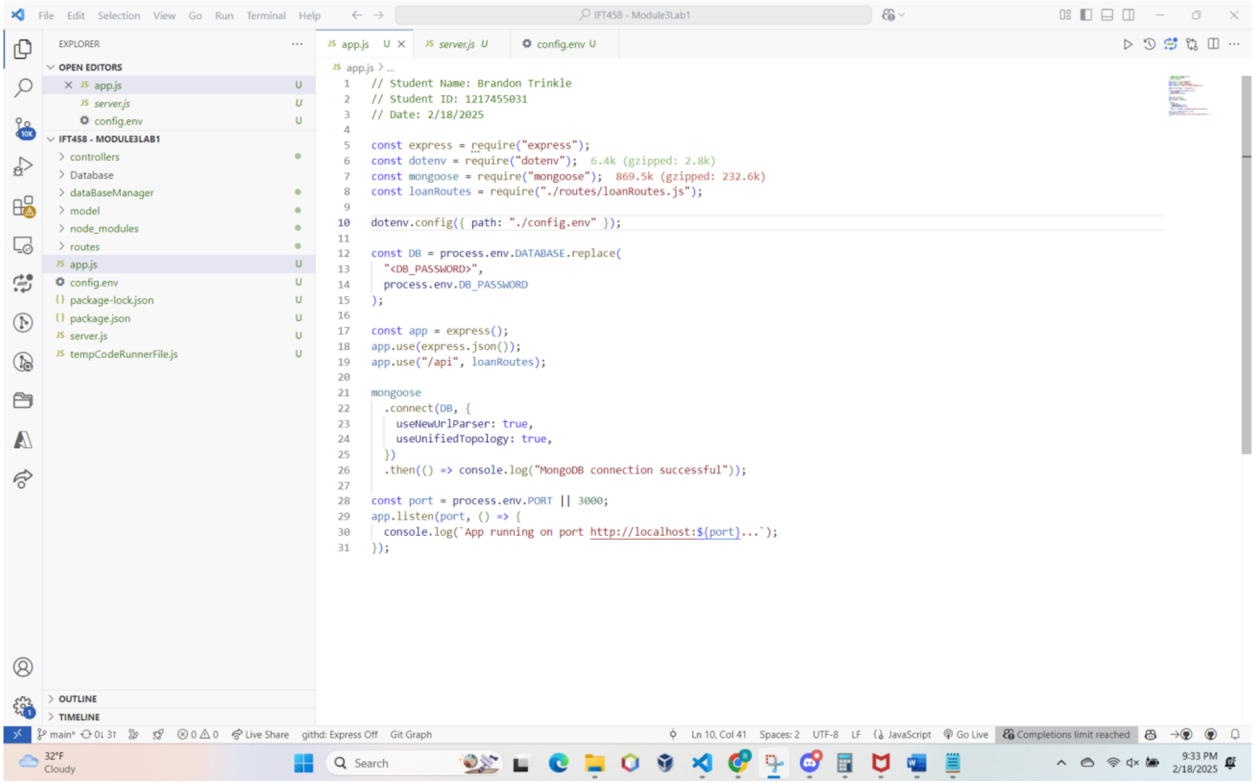
4. *Screenshot of the loanController.js file showing the implemented controller logic.*



```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const mongoose = require("mongoose");  895k (gzipped: 239k)

const loanSchema = new mongoose.Schema({
  name: {
    type: String,
    required: [true, "User Name is required"],
  },
  amount: {
    type: Number,
    required: [true, "Loan amount is required"],
  },
  borrower: {
    type: String,
    required: [true, "Borrower name is required"],
  },
  interestRate: {
    type: Number,
    required: [true, "Interest rate is required"],
  },
  status: {
    type: String,
    default: "pending",
  },
});

const Loan = mongoose.model("Loan", loanSchema);
module.exports = Loan;
```

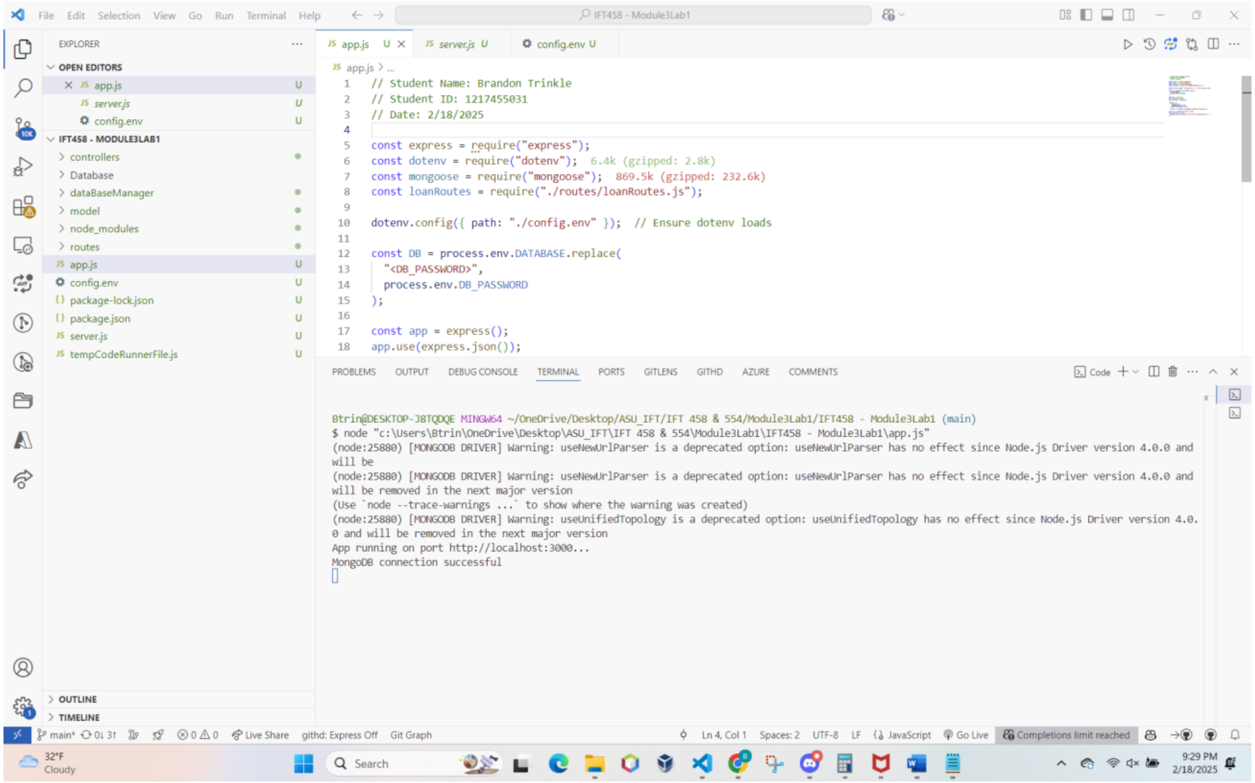
```
/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1 (main)
$
```

5. *Screenshot of the app.js file showing the added routes and their configuration.*



```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const express = require("express");
const dotenv = require("dotenv");  6.4k (gzipped: 2.8k)
const mongoose = require("mongoose");  869.5k (gzipped: 232.6k)
const loanRoutes = require("./routes/loanRoutes.js");

dotenv.config({ path: "./config.env" });

const DB = process.env.DATABASE.replace(
  "<DB_PASSWORD>",
  process.env.DB_PASSWORD
);

const app = express();
app.use(express.json());
app.use("/api", loanRoutes);

mongoose
  .connect(DB, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then(() => console.log("MongoDB connection successful"));

const port = process.env.PORT || 3000;
app.listen(port, () => {
  console.log(`App running on port http://localhost:${port}...`);
});
```

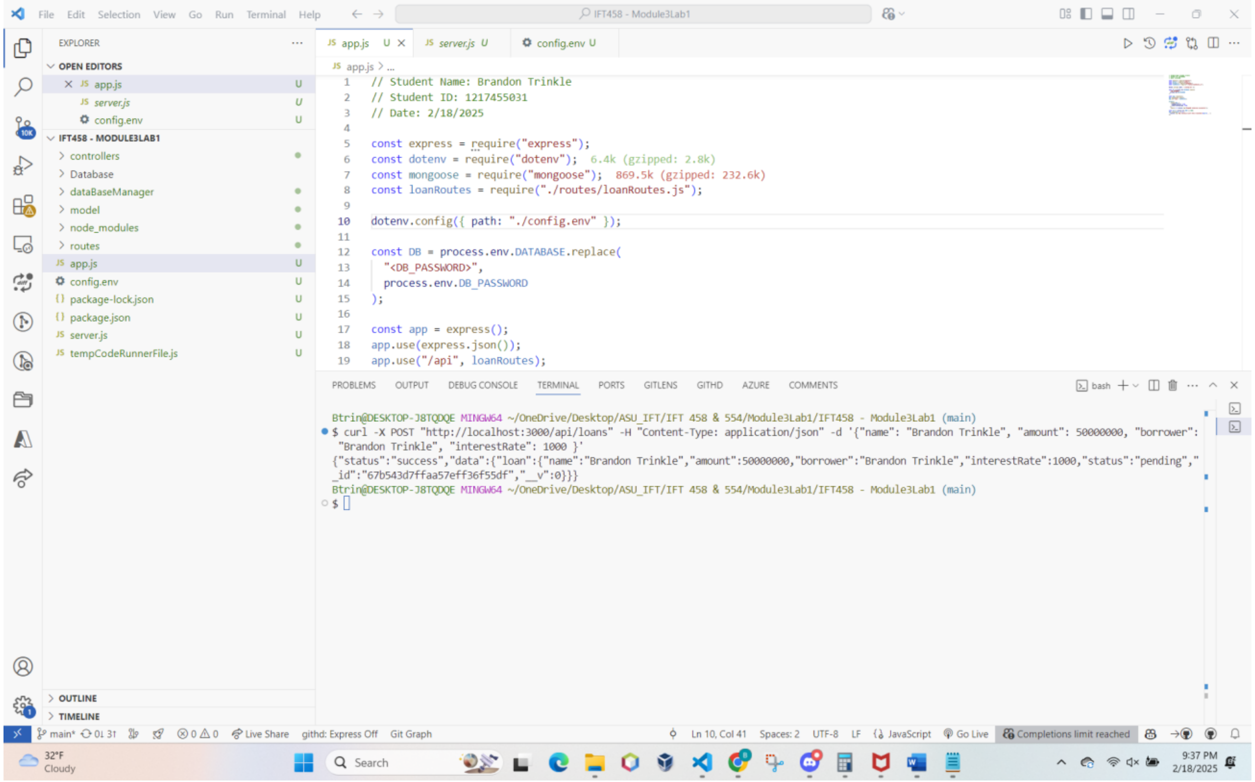
- *6*
  - *A. Take a screenshot of the Postman request showing the successful addition of data.*
  - *Open MongoDB terminal and verify that the data has been successfully inserted.*
  - *Take a screenshot of the MongoDB terminal showing the added data.*



```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const express = require("express");
const dotenv = require("dotenv");  6.4k (gzipped: 2.8k)
const mongoose = require("mongoose");  869.5k (gzipped: 232.6k)
const loanRoutes = require("./routes/loanRoutes.js");

dotenv.config({ path: "./config.env" });  // Ensure dotenv loads

const DB = process.env.DATABASE.replace(
  "<DB_PASSWORD>",
  process.env.DB_PASSWORD
);

const app = express();
app.use(express.json());
```

```
Btrin@DESKTOP-J8TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1/IFT458 - Module3Lab1 (main)
$ node "c:\Users\Btrin\OneDrive\Desktop\ASU_IFT\IFT 458 & 554\Module3Lab1\IFT458 - Module3Lab1\app.js"
(node:25880) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and
will be
(node:25880) [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:25880) [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
App running on port http://localhost:3000...
MongoDB connection successful
```

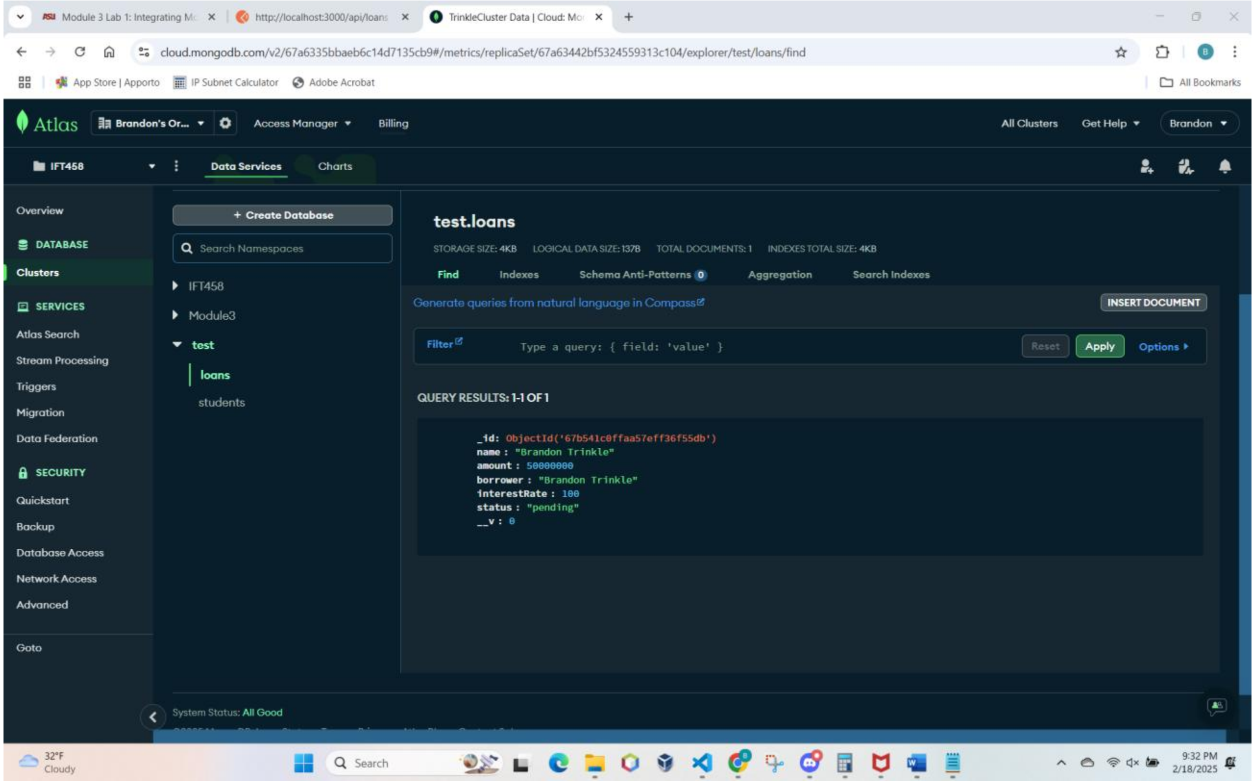
Note: connected to MongoDB



```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/18/2025

const express = require("express");
const dotenv = require("dotenv");  6.4k (gzipped: 2.8k)
const mongoose = require("mongoose");  869.5k (gzipped: 232.6k)
const loanRoutes = require("./routes/loanRoutes.js");

dotenv.config({ path: "./config.env" });

const DB = process.env.DATABASE.replace(
  "<DB_PASSWORD>",
  process.env.DB_PASSWORD
);

const app = express();
app.use(express.json());
app.use("/api", loanRoutes);
```

```
Btrin@DESKTOP-J8TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1/IFT458 - Module3Lab1 (main)
$ curl -X POST "http://localhost:3000/api/loans" -H "Content-Type: application/json" -d '{"name": "Brandon Trinkle", "amount": 50000000, "borrower":
 "Brandon Trinkle", "interestRate": 1000 }'
{"status":"success","data":{"loan":{"name":"Brandon Trinkle","amount":50000000,"borrower":"Brandon Trinkle","interestRate":1000,"status":"pending","
_id":"67b543d7ffaa57eff36f55df","__v":0}}}
Btrin@DESKTOP-J8TQDQE MINGW64 ~/OneDrive/Desktop/ASU_IFT/IFT 458 & 554/Module3Lab1/IFT458 - Module3Lab1 (main)
$ 
```

Note: I had issues with postman, so I used curl to send a post request to the server. I gave myself an absurd interest rate as a result!



test.loans

STORAGE SIZE: 4KB LOGICAL DATA SIZE: 137B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 4KB

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('67b541c0ffaa57eff36f55db')
name : "Brandon Trinkle"
amount : 50000000
borrower : "Brandon Trinkle"
interestRate : 100
status : "pending"
__v : 0
```

Note: MongoDB was updated.