Module 3 Lab 2

Brandon Trinkle

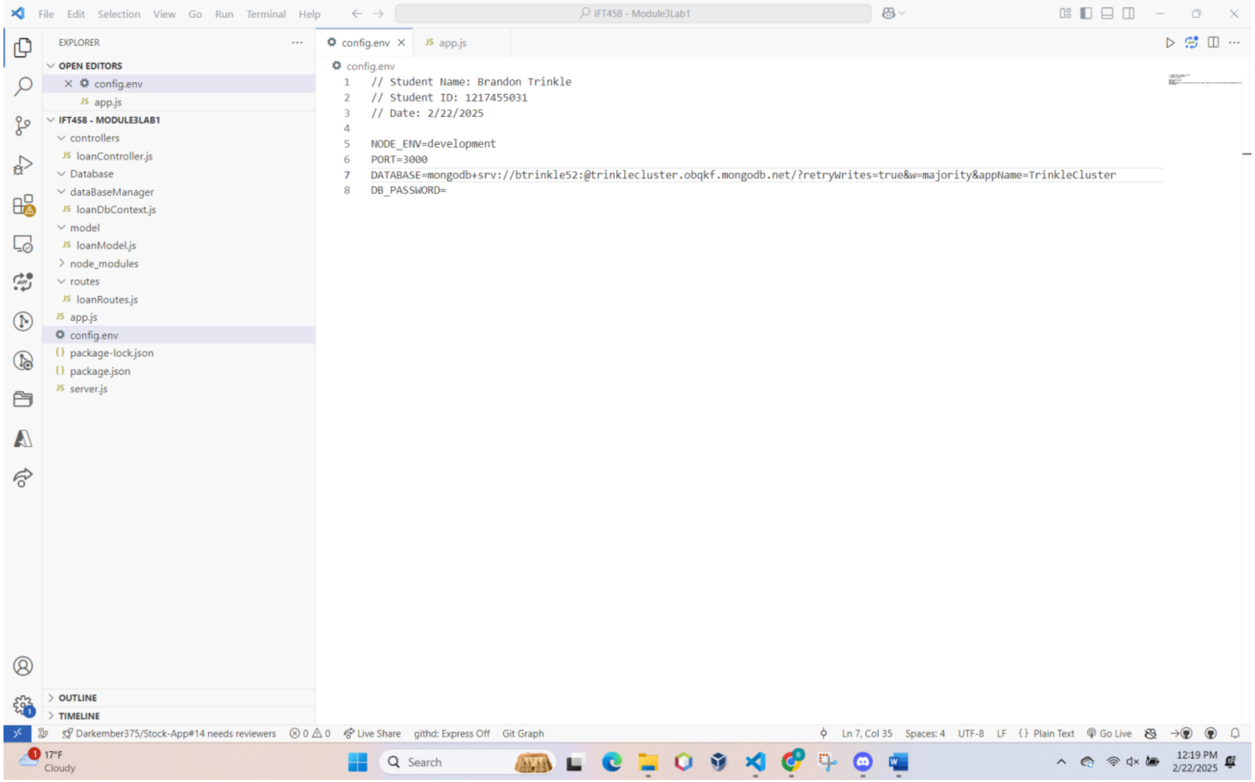
IFT 458: Middleware Prog & Database Sec

Professor: Dinesh Sthapit

Feb 22, 2025

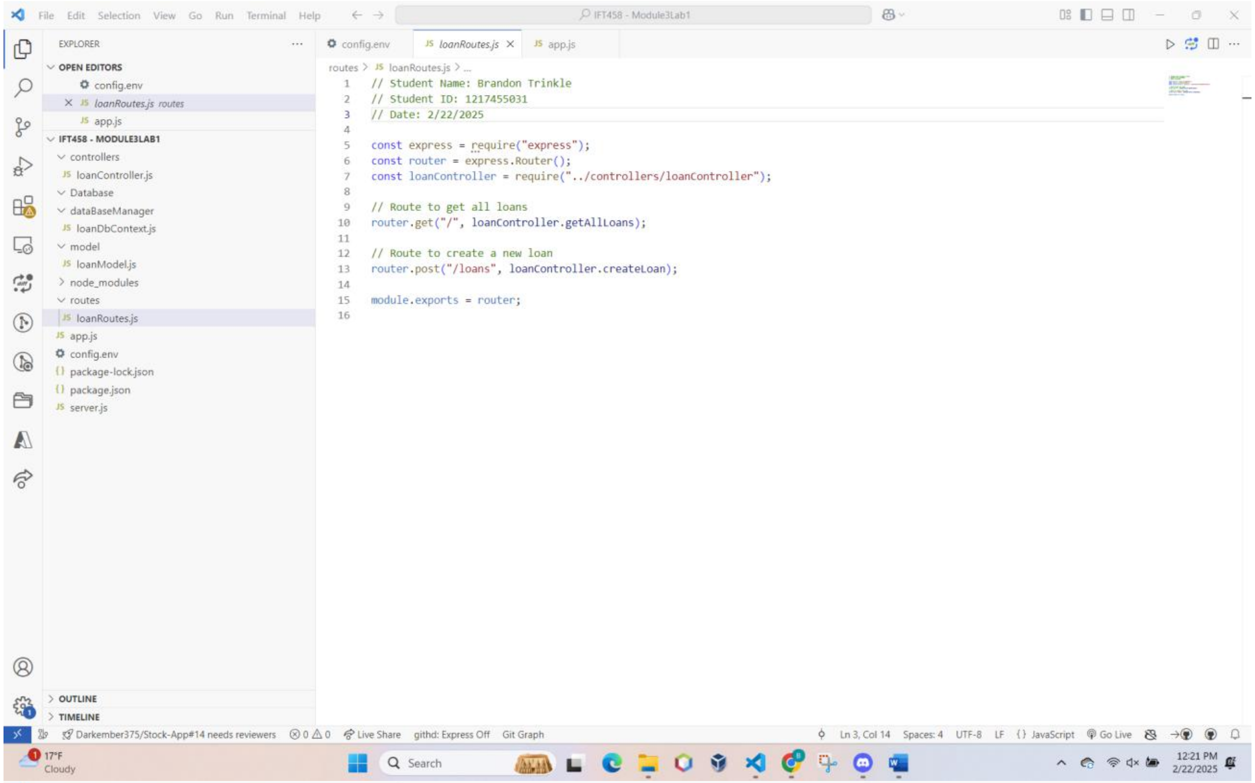
Module 3 Lab 2

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

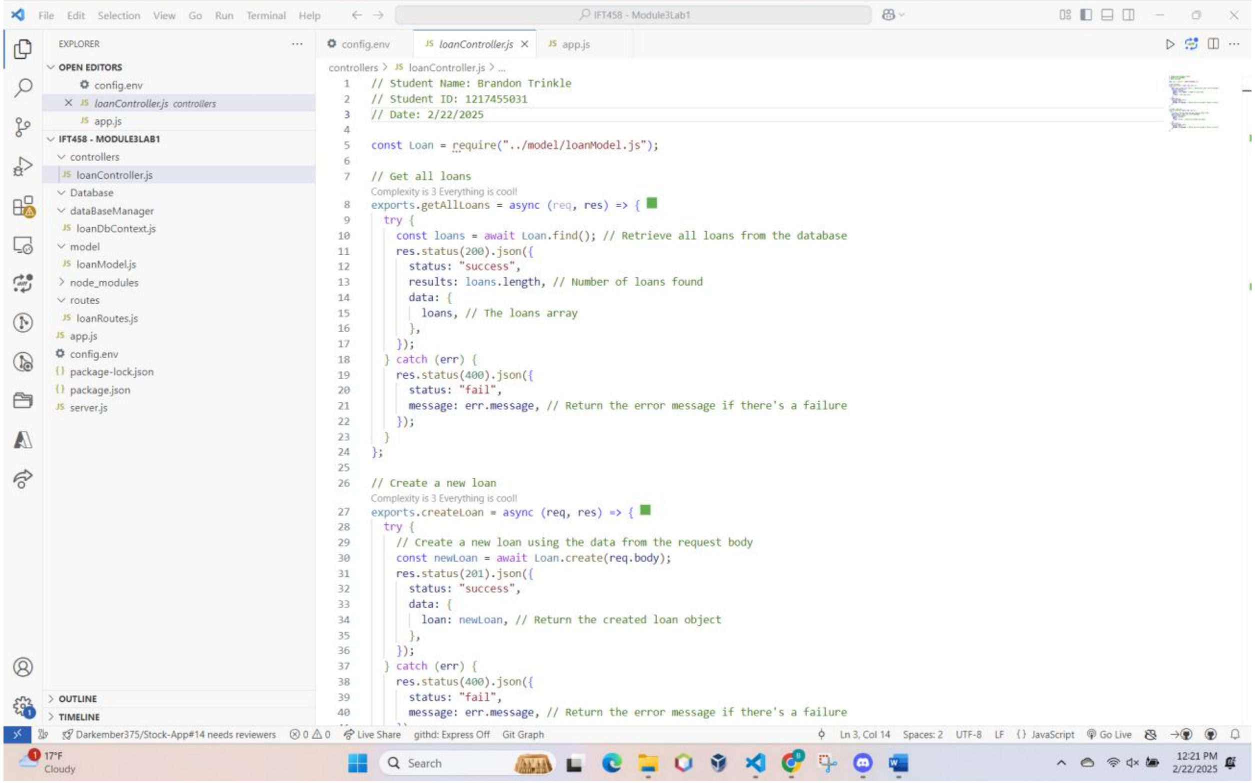


```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

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

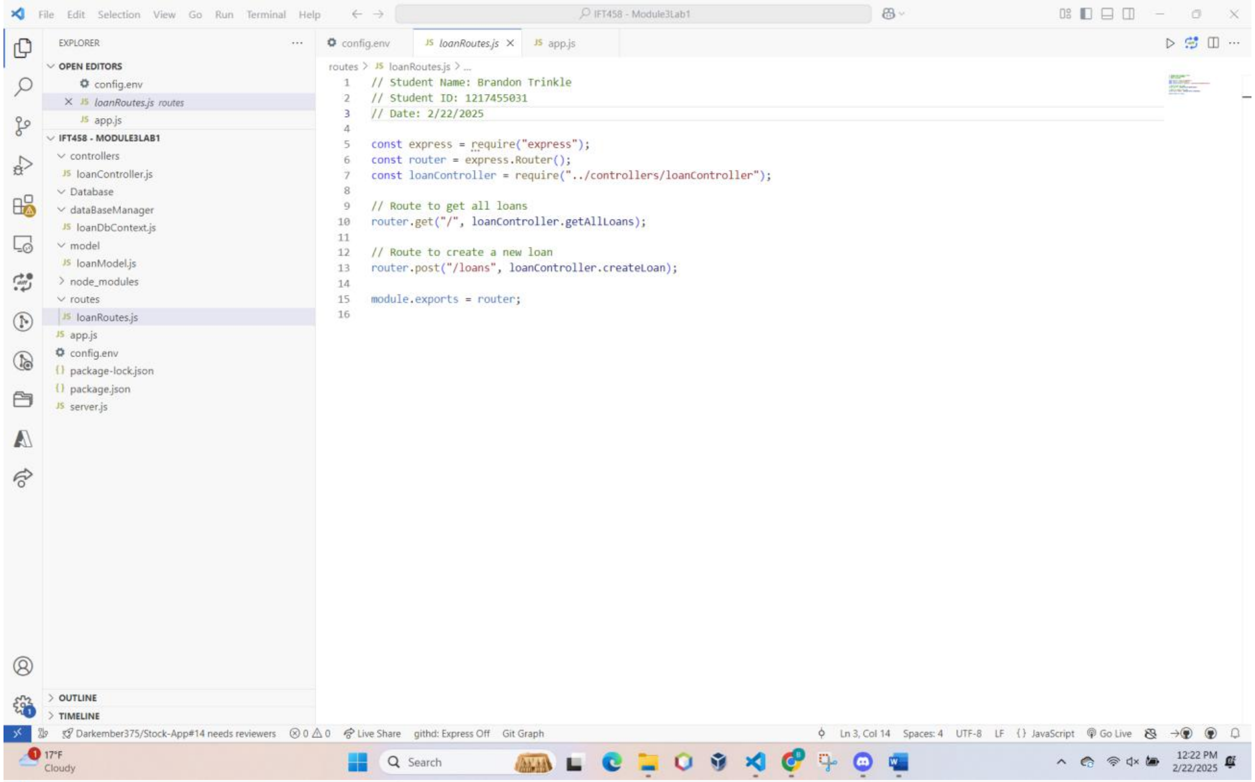
2. *Deliverable: Screenshots of the loanModel.js, loanController.js, loanRoutes.js, and server.js files showing the code and file structure. **Multiple screenshots, see deliverables.***



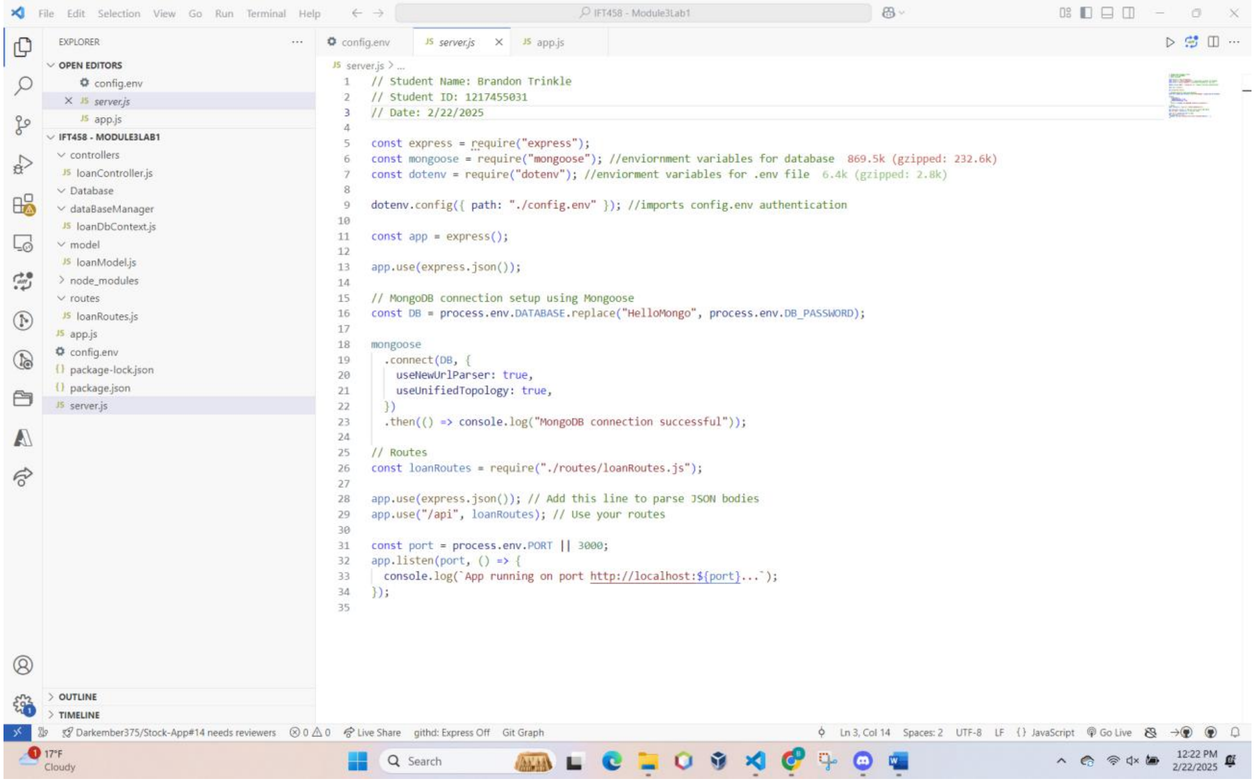
```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

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;
```



```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

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

// Get all loans
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
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
      },
    });
  } catch (err) {
    res.status(400).json({
      status: "fail",
      message: err.message, // Return the error message if there's a failure
```

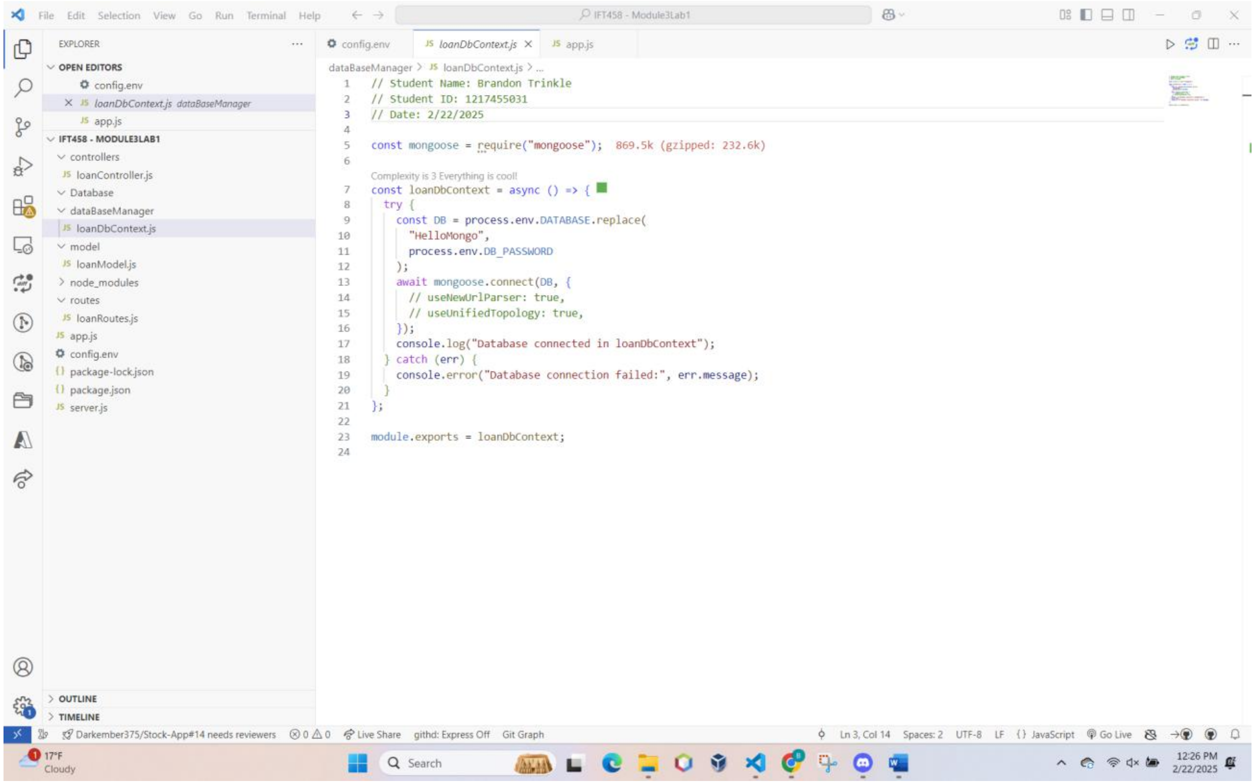


```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

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;
```



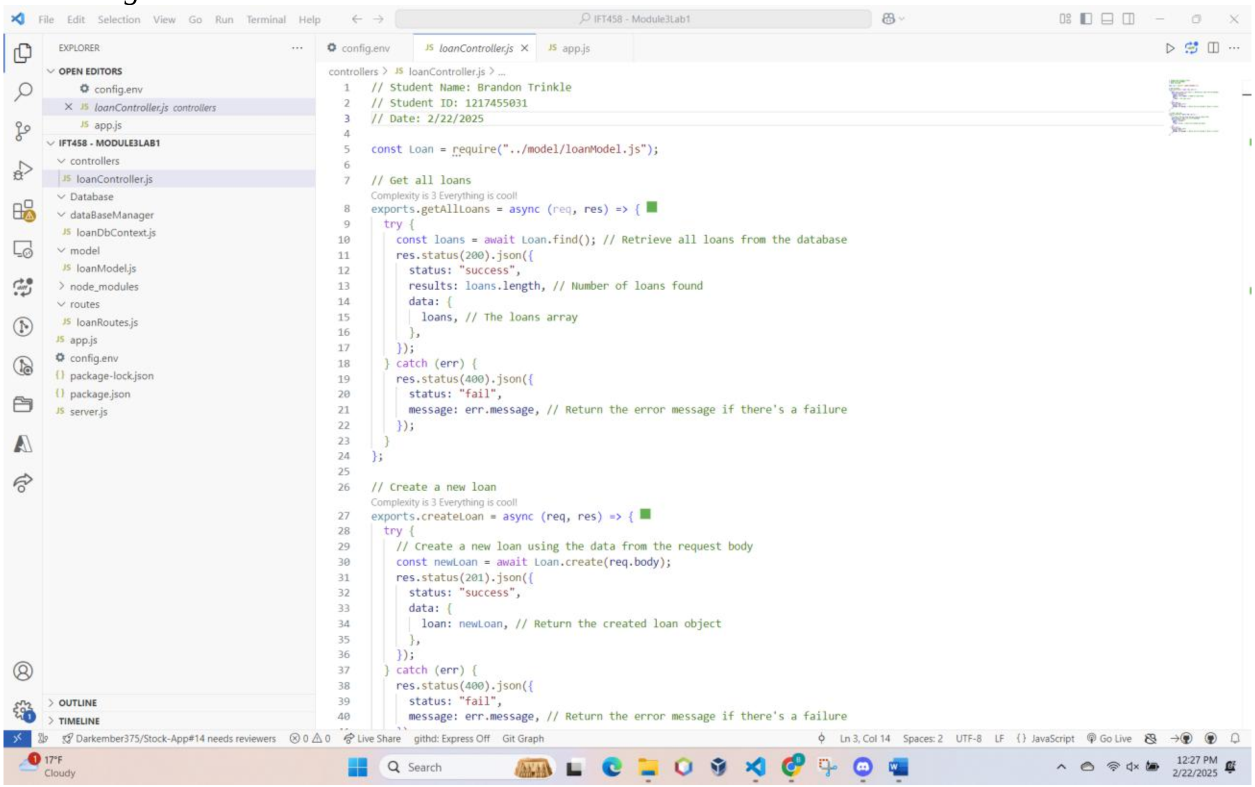
```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

const express = require("express");
const mongoose = require("mongoose"); //enviornment variables for database  869.5k (gzipped: 232.6k)
const dotenv = require("dotenv"); //enviorment variables for .env file  6.4k (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}...`);
});
```

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



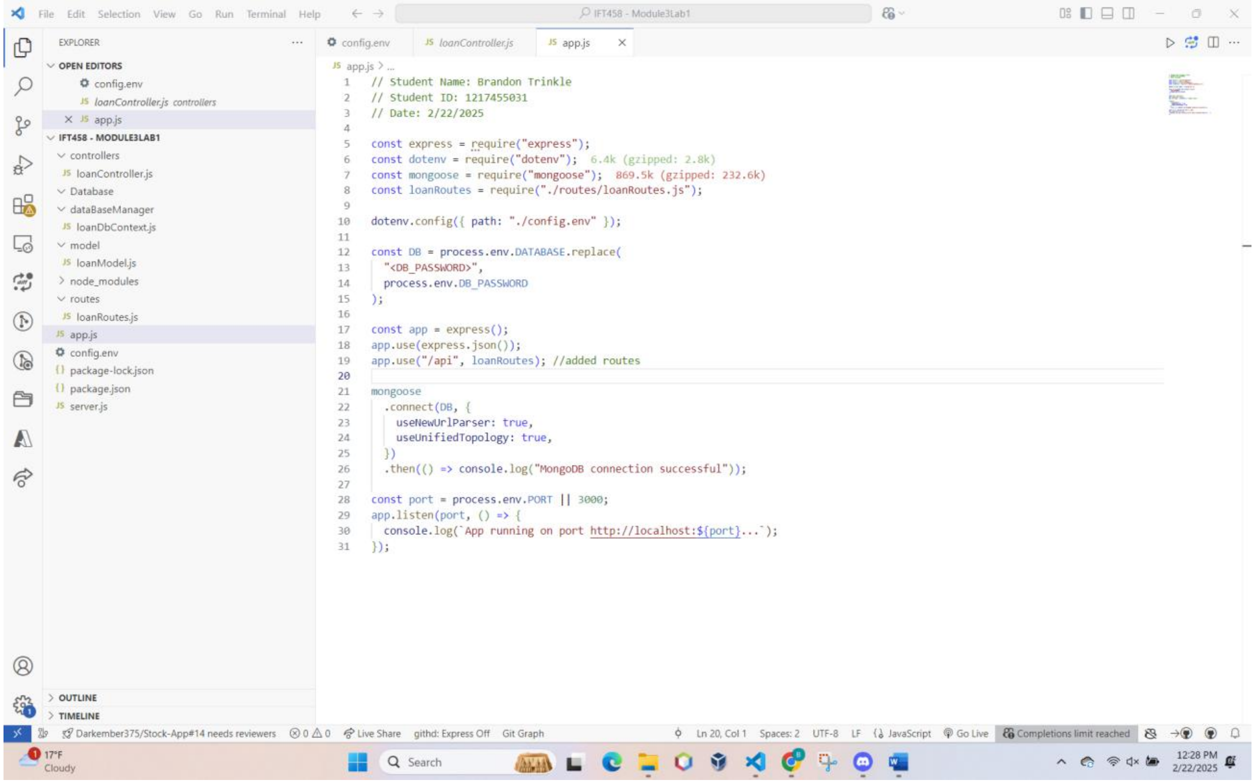
```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

const mongoose = require("mongoose");  869.5k (gzipped: 232.6k)

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;
```

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



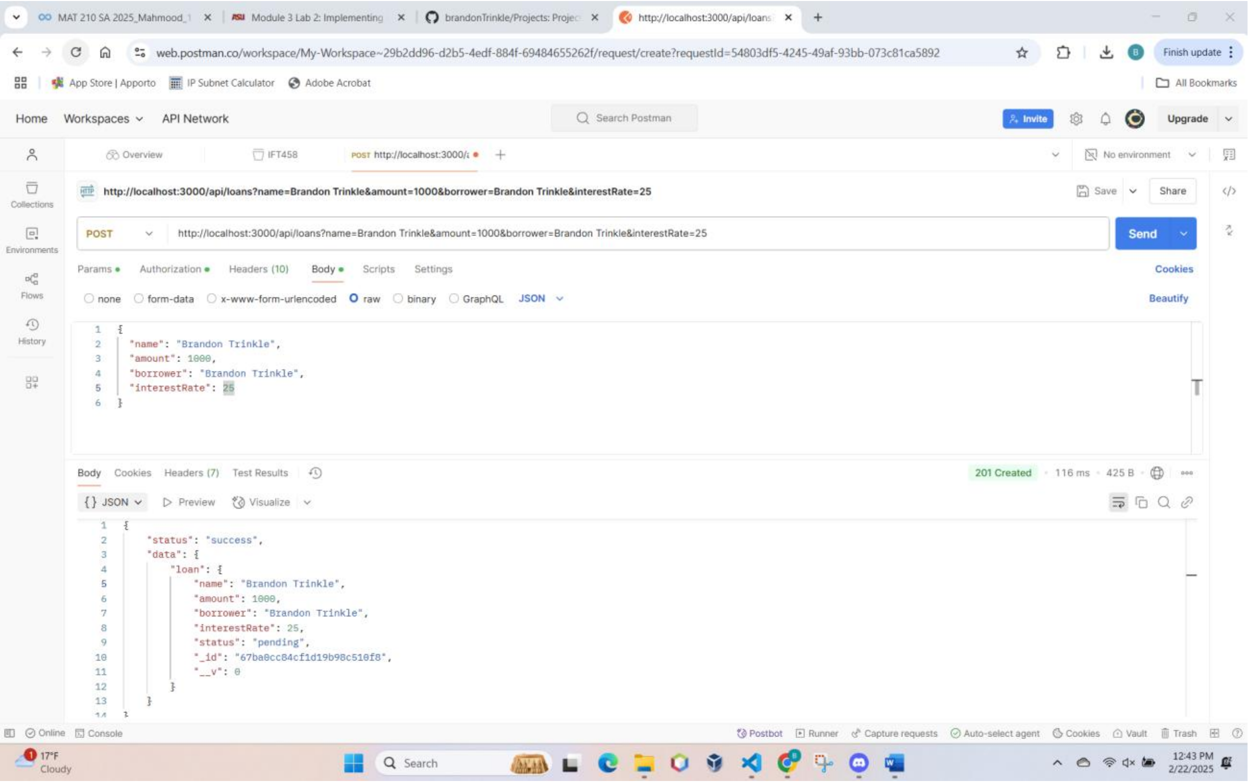
```javascript
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

const Loan = require("../model/loanModel.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
      },
    });
  } catch (err) {
    res.status(400).json({
      status: "fail",
      message: err.message, // Return the error message if there's a failure
```

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



```
// Student Name: Brandon Trinkle
// Student ID: 1217455031
// Date: 2/22/2025

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); //added routes

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. *Deliverable: Take a screenshot of the Postman request showing the successful addition of data.*



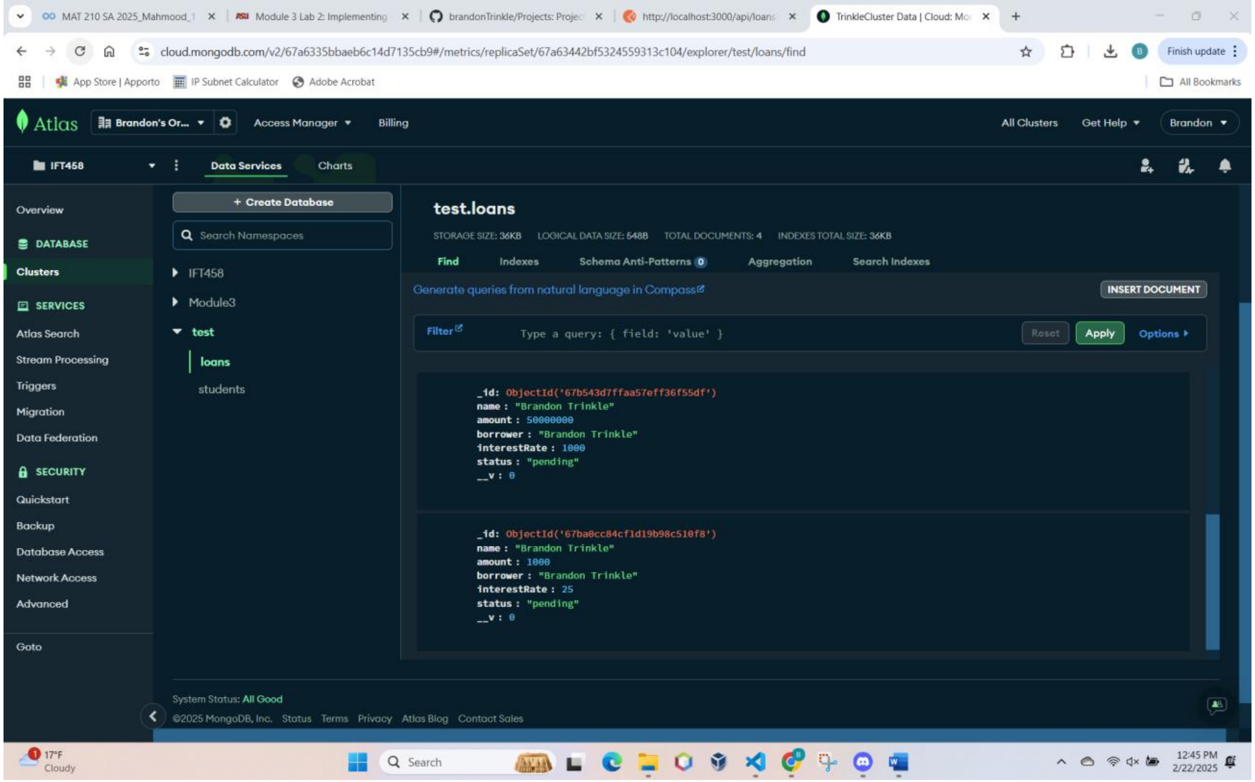
POST http://localhost:3000/api/loans?name=Brandon Trinkle&amount=1000&borrower=Brandon Trinkle&interestRate=25

```
{
  "name": "Brandon Trinkle",
  "amount": 1000,
  "borrower": "Brandon Trinkle",
  "interestRate": 25
}
```

201 Created · 116 ms · 425 B

```
{
    "status": "success",
    "data": {
        "loan": {
            "name": "Brandon Trinkle",
            "amount": 1000,
            "borrower": "Brandon Trinkle",
            "interestRate": 25,
            "status": "pending",
            "_id": "67ba0cc84cf1d19b98c510f8",
            "__v": 0
        }
    }
```

7. *Open MongoDB terminal and verify that the data has been successfully inserted.*



test.loans

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 548B TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 36KB

```
_id: ObjectId('67b543d7ffaa57eff36f55df')
name : "Brandon Trinkle"
amount : 50000000
borrower : "Brandon Trinkle"
interestRate : 1000
status : "pending"
__v : 0
```

```
_id: ObjectId('67ba0cc84cf1d19b98c510f8')
name : "Brandon Trinkle"
amount : 1000
borrower : "Brandon Trinkle"
interestRate : 25
status : "pending"
__v : 0
```