

## **Module 3 Lab 2: Integrating MongoDB with Express.js**

Mohiddeen Vilak Mohammad

Department of Information Technology, Arizona State University

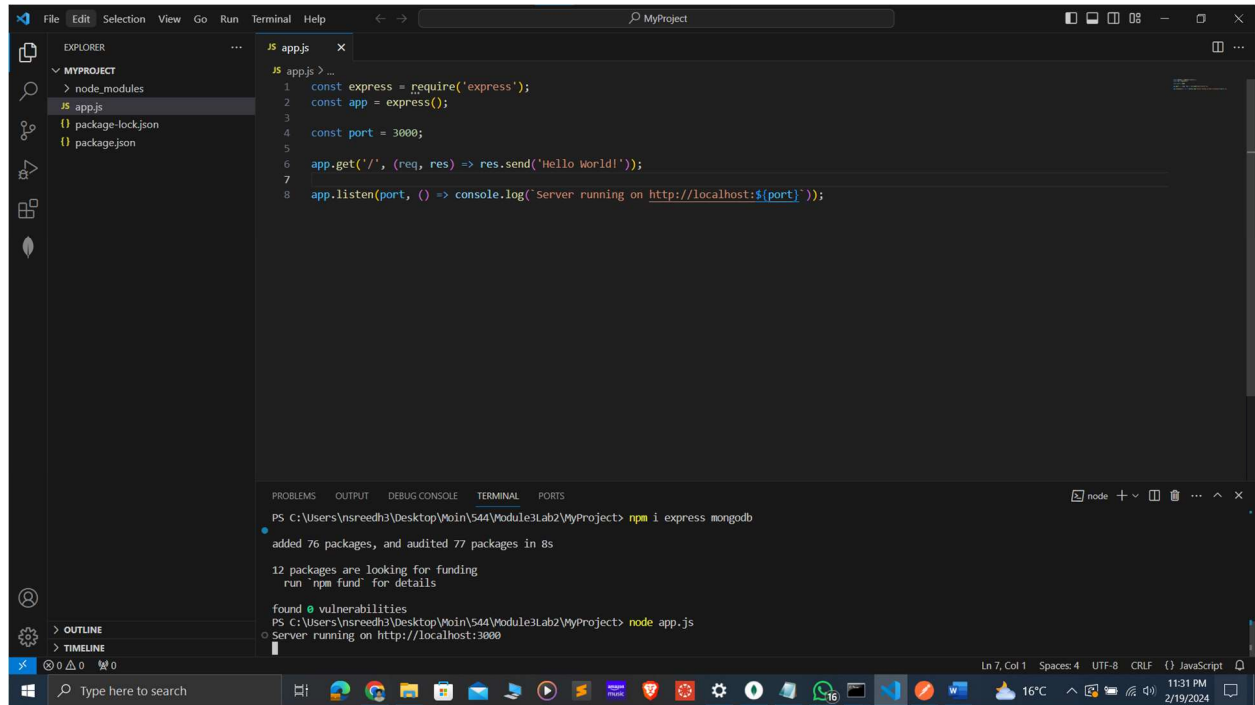
IFT 554: Middleware Programming & Database Security

Dinesh Sthapit

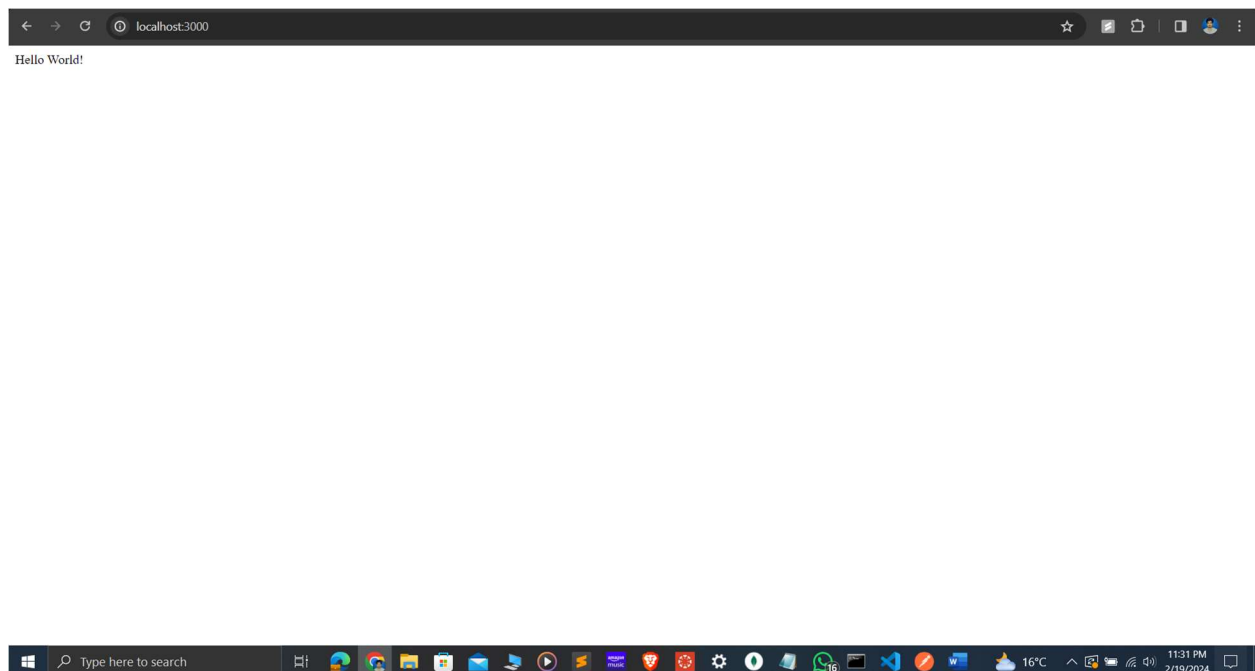
February 18, 2024

**Figure 1:**

*Manually adding config.env file*

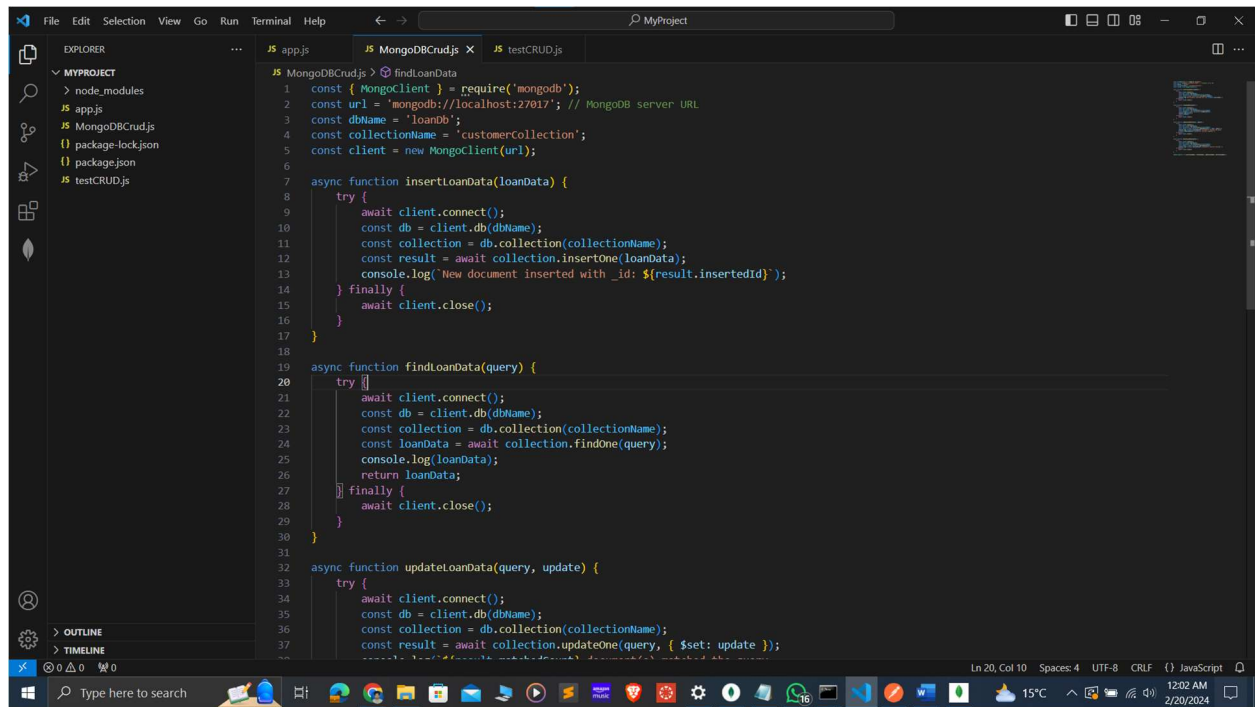


**Figure 2:**



**Figure 3:**

*MongoDBCrud.js file*



```
1 const { MongoClient } = require('mongodb');
2 const url = 'mongodb://localhost:27017'; // MongoDB server URL
3 const dbName = 'loanDb';
4 const collectionName = 'customerCollection';
5 const client = new MongoClient(url);
6
7 async function insertLoanData(loanData) {
8   try {
9     await client.connect();
10    const db = client.db(dbName);
11    const collection = db.collection(collectionName);
12    const result = await collection.insertOne(loanData);
13    console.log('New document inserted with _id: ${result.insertedId}');
14  } finally {
15    await client.close();
16  }
17 }
18
19 async function findLoanData(query) {
20   try {
21     await client.connect();
22     const db = client.db(dbName);
23     const collection = db.collection(collectionName);
24     const loanData = await collection.findOne(query);
25     console.log(loanData);
26     return loanData;
27   } finally {
28     await client.close();
29   }
30 }
31
32 async function updateLoanData(query, update) {
33   try {
34     await client.connect();
35     const db = client.db(dbName);
36     const collection = db.collection(collectionName);
37     const result = await collection.updateOne(query, { $set: update });
38     console.log('Document updated with _id: ${result.upsertedId}');
```

**Figure 4:**

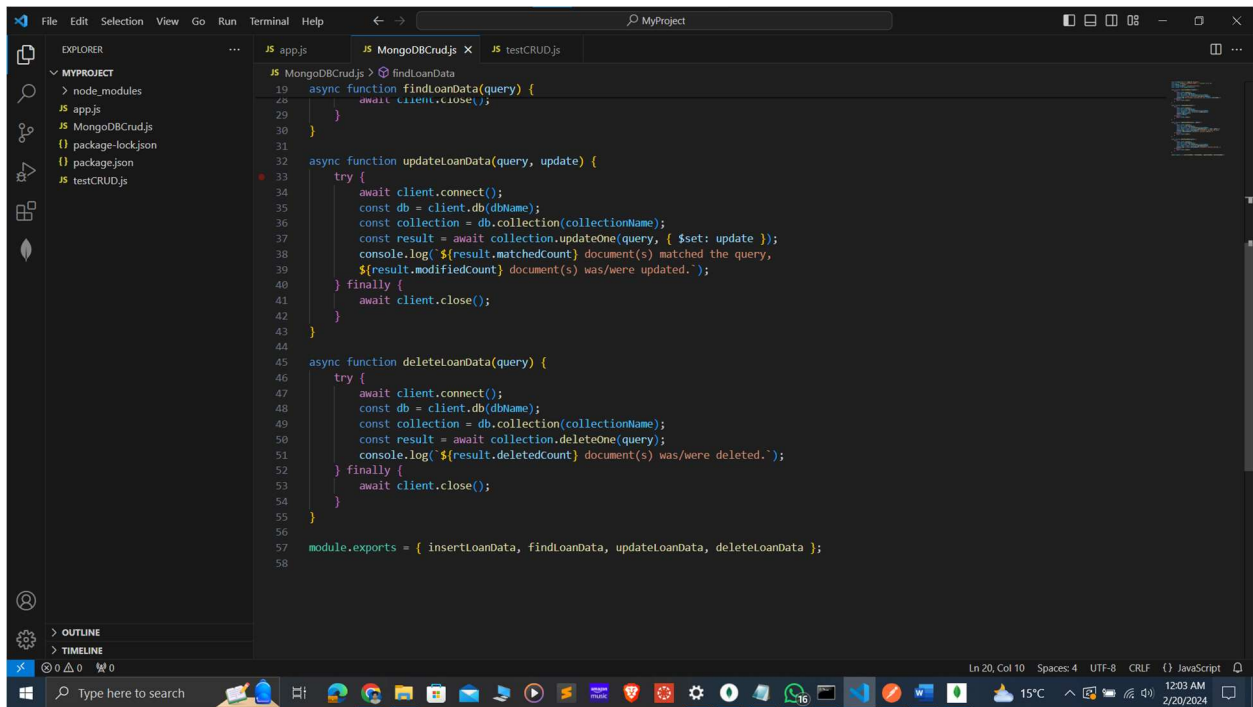


Figure 5:

*testCRUD.js file*

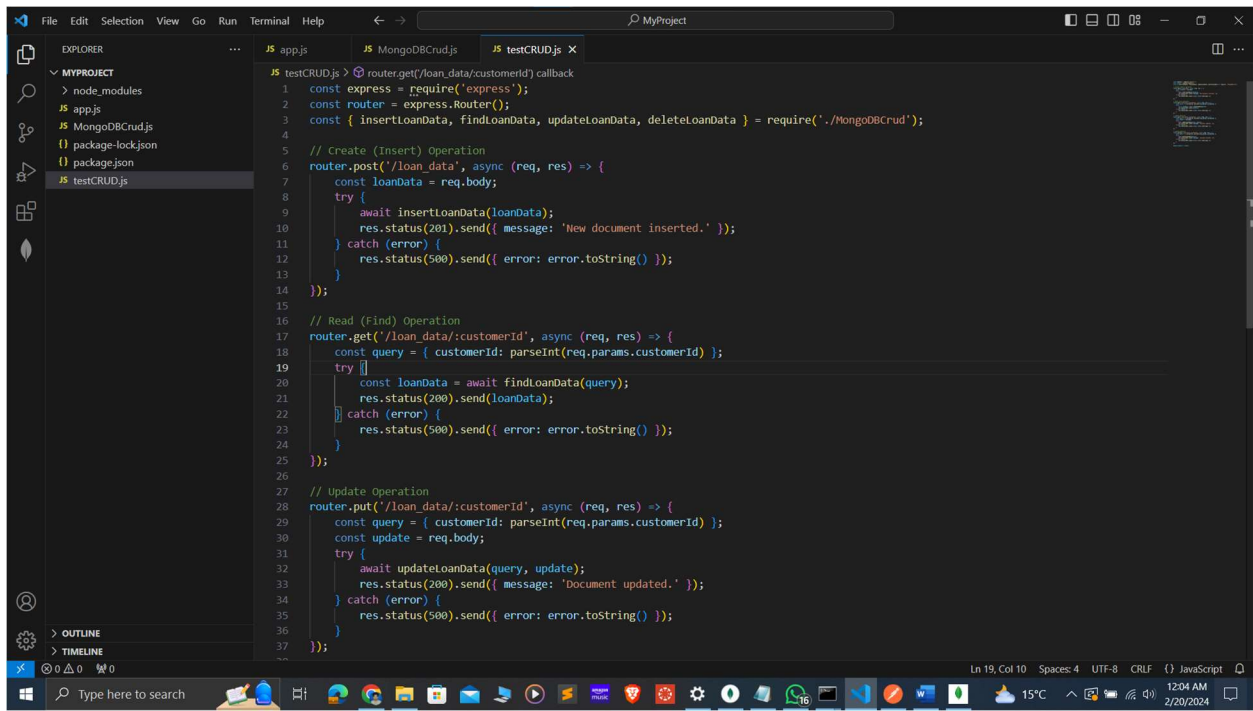
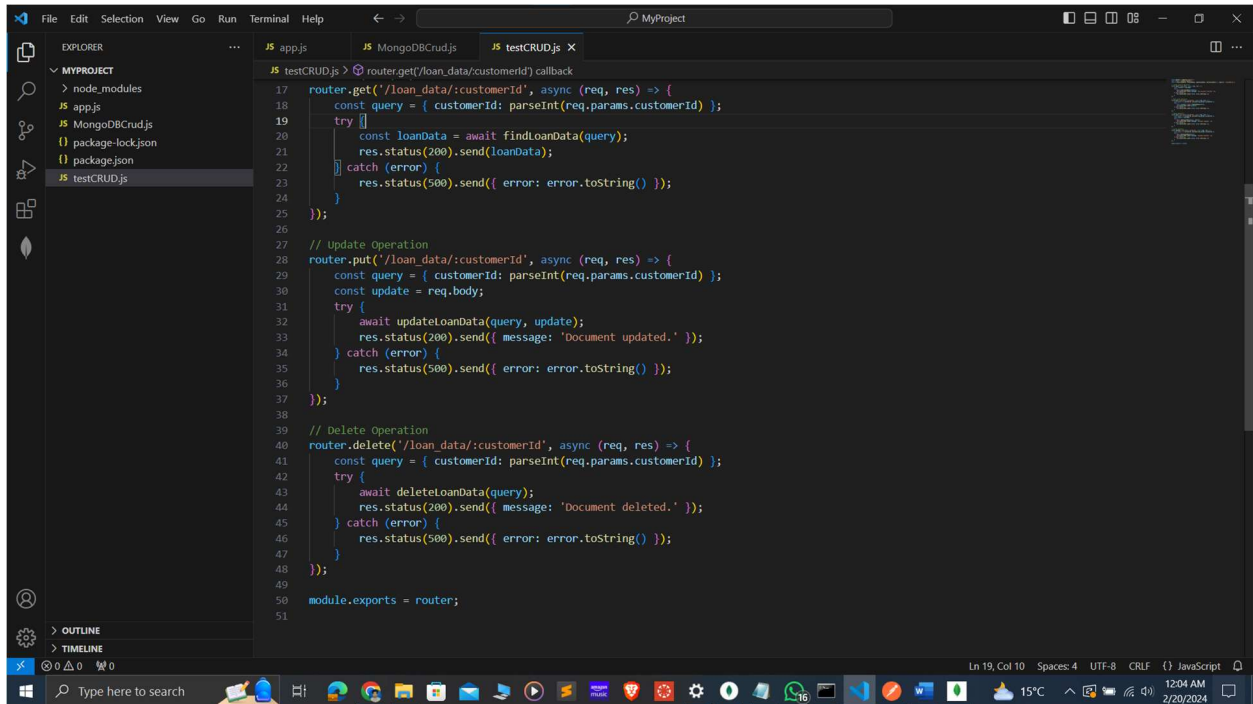


Figure 6:



```
17 router.get('/loan_data/:customerId', async (req, res) => {
18   const query = { customerId: parseInt(req.params.customerId) };
19   try {
20     const loanData = await findLoanData(query);
21     res.status(200).send(loanData);
22   } catch (error) {
23     res.status(500).send({ error: error.toString() });
24   }
25 });
26
27 // Update Operation
28 router.put('/loan_data/:customerId', async (req, res) => {
29   const query = { customerId: parseInt(req.params.customerId) };
30   const update = req.body;
31   try {
32     await updateLoanData(query, update);
33     res.status(200).send({ message: 'Document updated.' });
34   } catch (error) {
35     res.status(500).send({ error: error.toString() });
36   }
37 });
38
39 // Delete Operation
40 router.delete('/loan_data/:customerId', async (req, res) => {
41   const query = { customerId: parseInt(req.params.customerId) };
42   try {
43     await deleteLoanData(query);
44     res.status(200).send({ message: 'Document deleted.' });
45   } catch (error) {
46     res.status(500).send({ error: error.toString() });
47   }
48 });
49
50 module.exports = router;
51
```

Figure 7:

*Post request using postman*

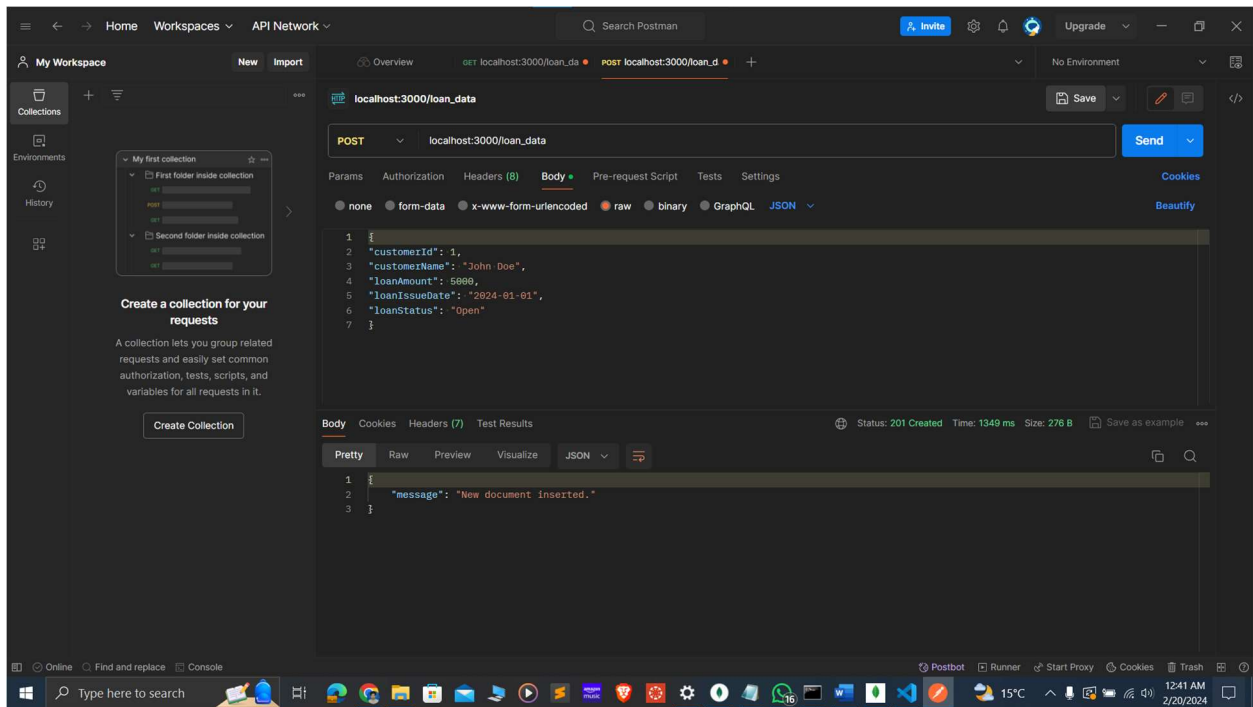
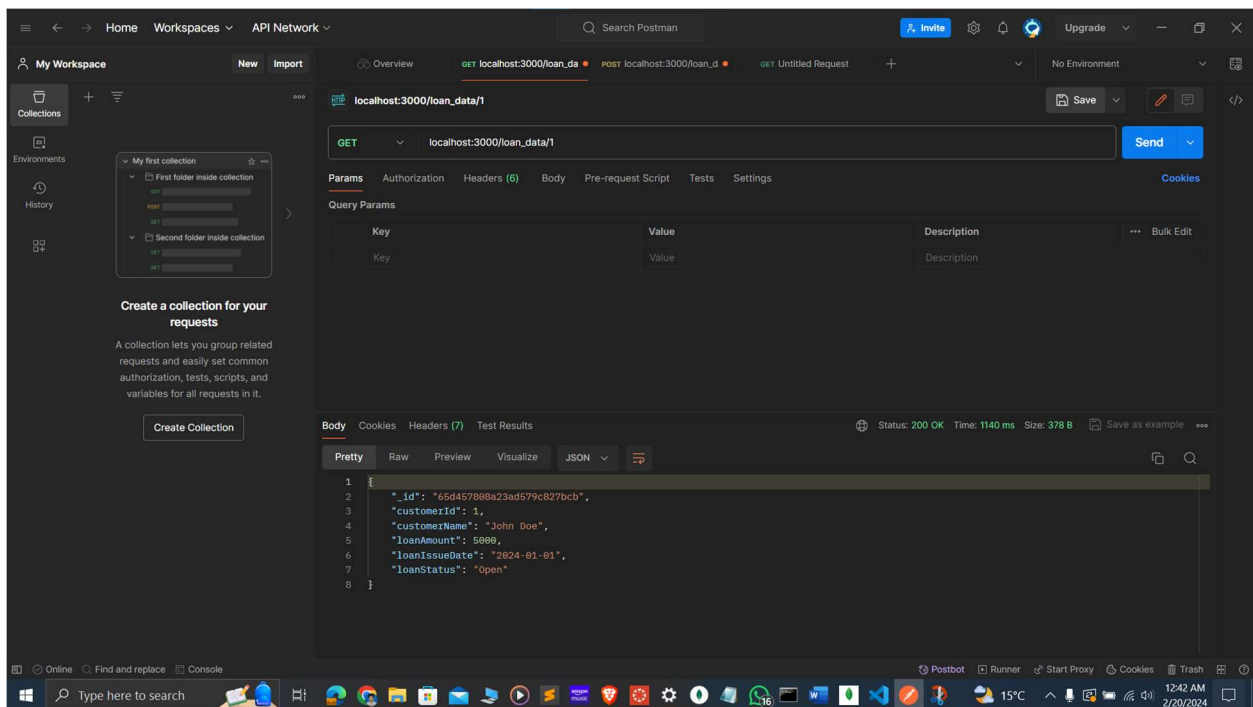


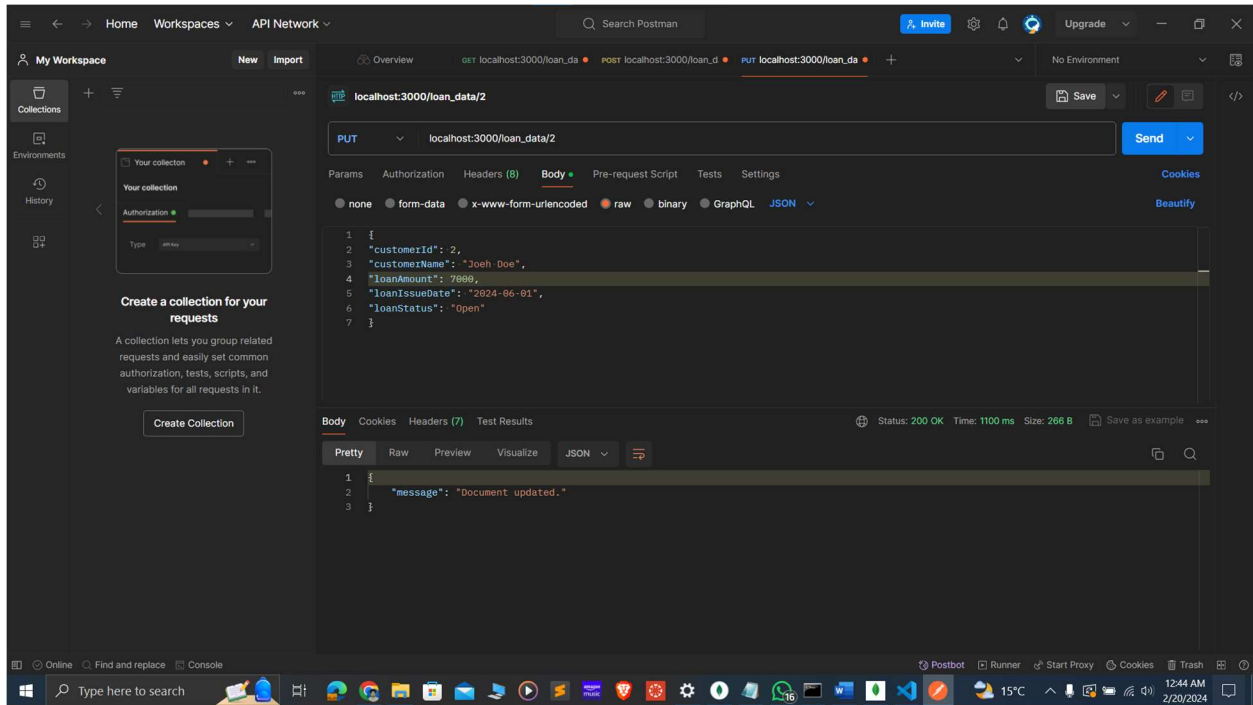
Figure 8:

*Gett request using postman*



**Figure 9:**

*Put request using postman*



**Figure 10:**

*Delete request using postman*

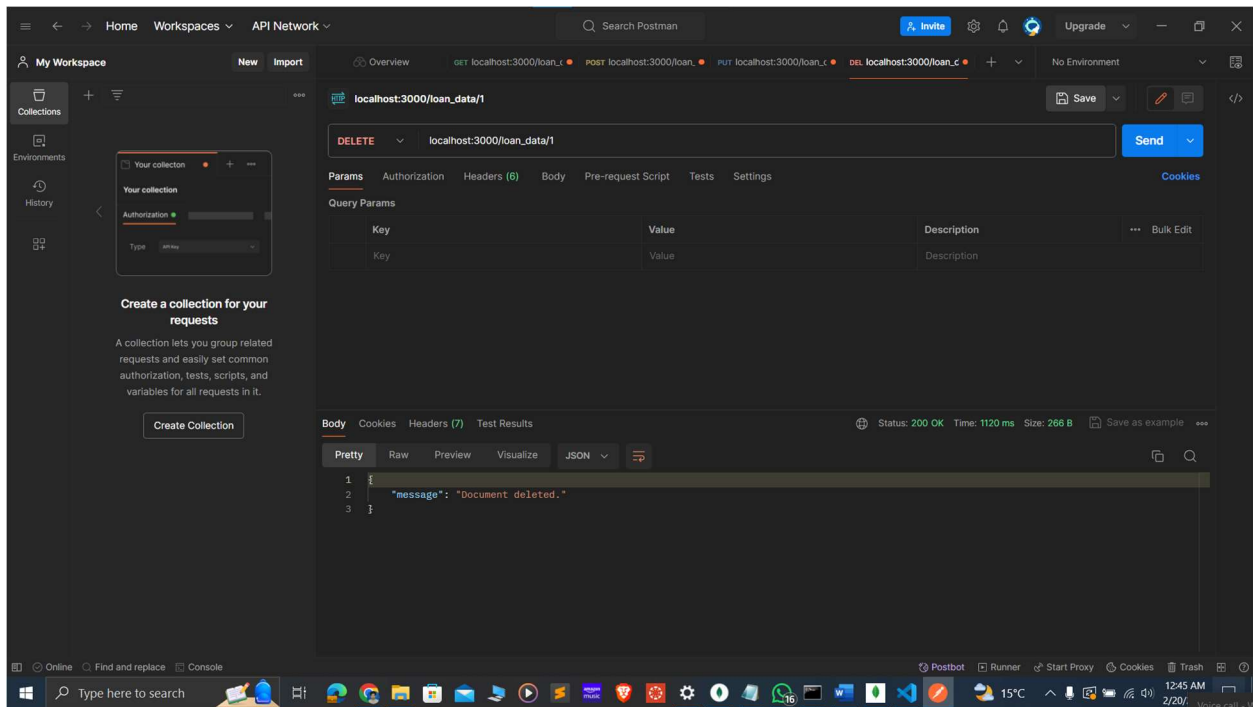


Figure 11:

*Database after all the operations*

