Week 8 Portfolio

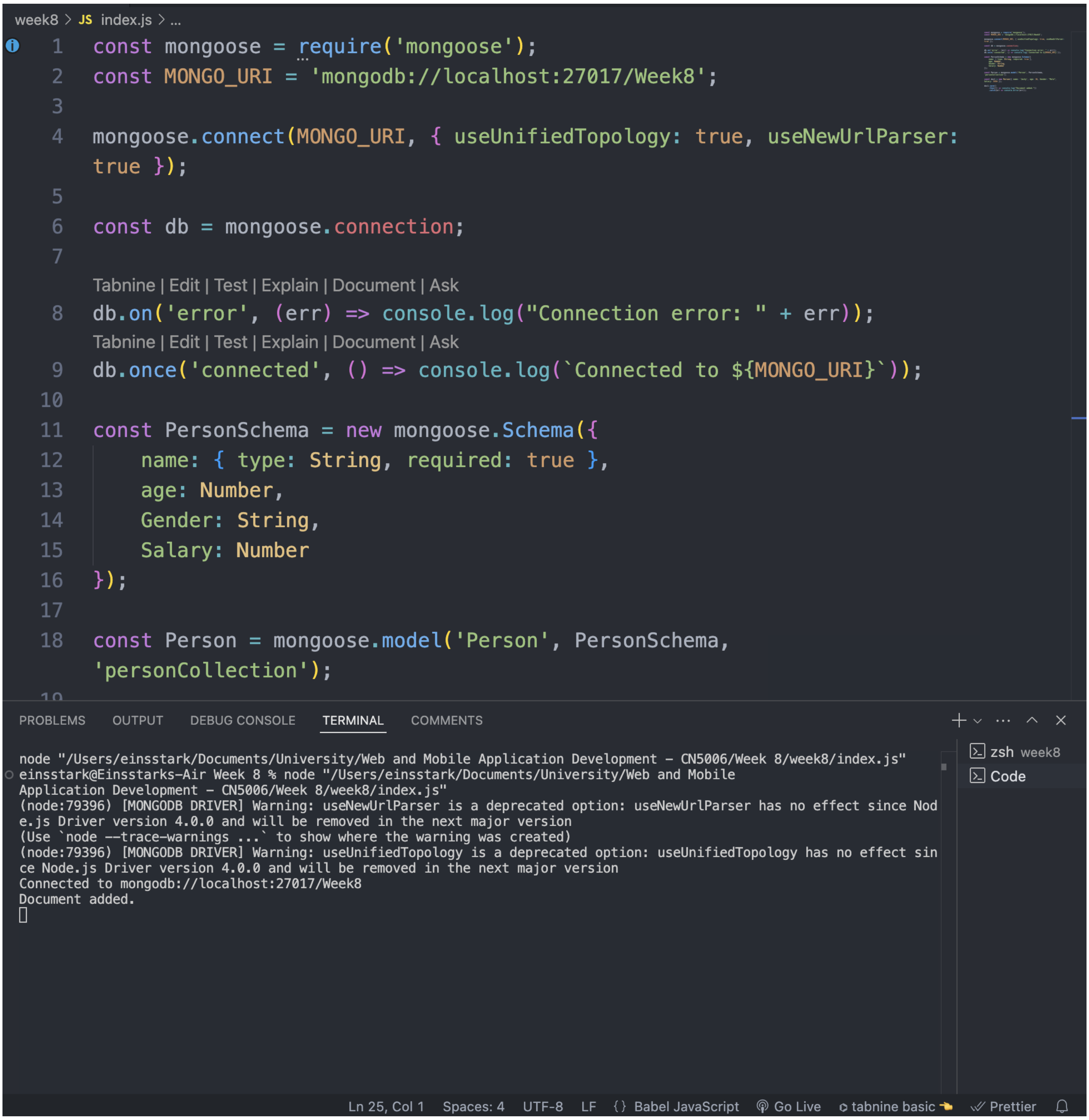
Name: Deepak Kumar
Student ID: 2284279

Using MongoDB CRUD Operations in a Programming Application Tutorial

1. Screenshots

Task 1: Connect with MongoDB and Add a Single Document

In this task, installed `mongoose` , connected to MongoDB, and made a schema for people. Then I created one person named Jacky and used `.save()` to add it to the database. It worked!

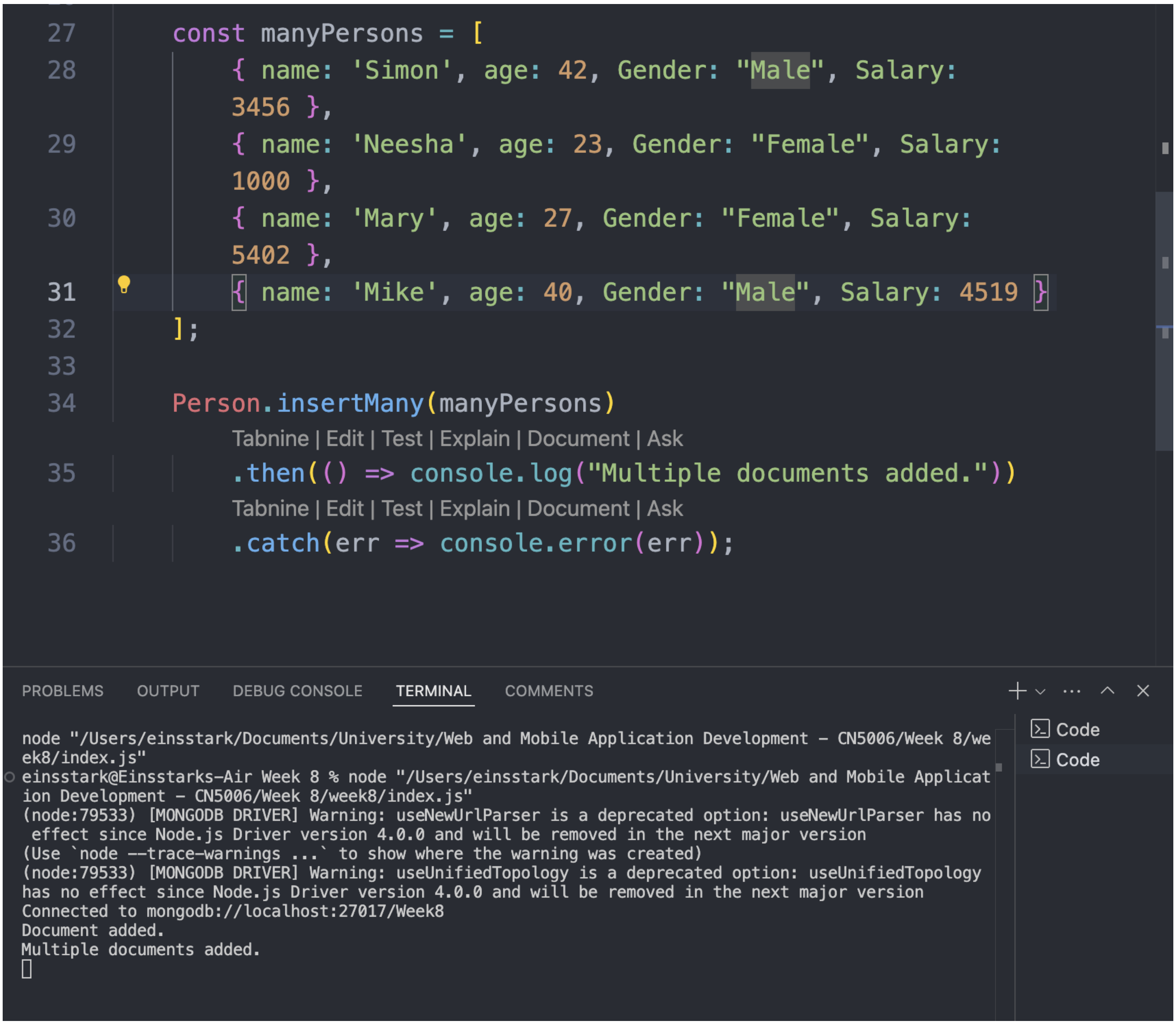


```javascript
const mongoose = require('mongoose');
const MONGO_URI = 'mongodb://localhost:27017/Week8';

mongoose.connect(MONGO_URI, { useUnifiedTopology: true, useNewUrlParser: true });

const db = mongoose.connection;

db.on('error', (err) => console.log("Connection error: " + err));
db.once('connected', () => console.log(`Connected to ${MONGO_URI}`));

const PersonSchema = new mongoose.Schema({
    name: { type: String, required: true },
    age: Number,
    Gender: String,
    Salary: Number
});

const Person = mongoose.model('Person', PersonSchema, 'personCollection');
```

```
node "/Users/einsstark/Documents/University/Web and Mobile Application Development - CN5006/Week 8/week8/index.js"
einsstark@Einsstarks-Air Week 8 % node "/Users/einsstark/Documents/University/Web and Mobile
Application Development - CN5006/Week 8/week8/index.js"
(node:79396) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Nod
e.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:79396) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect sin
ce Node.js Driver version 4.0.0 and will be removed in the next major version
Connected to mongodb://localhost:27017/Week8
Document added.
```

Task 2: Add Multiple Documents

I made a list of people with names, ages, genders, and salaries. Then I used `.insertMany()` to add all of them to the database in one go. It showed "Document added" on the terminal.



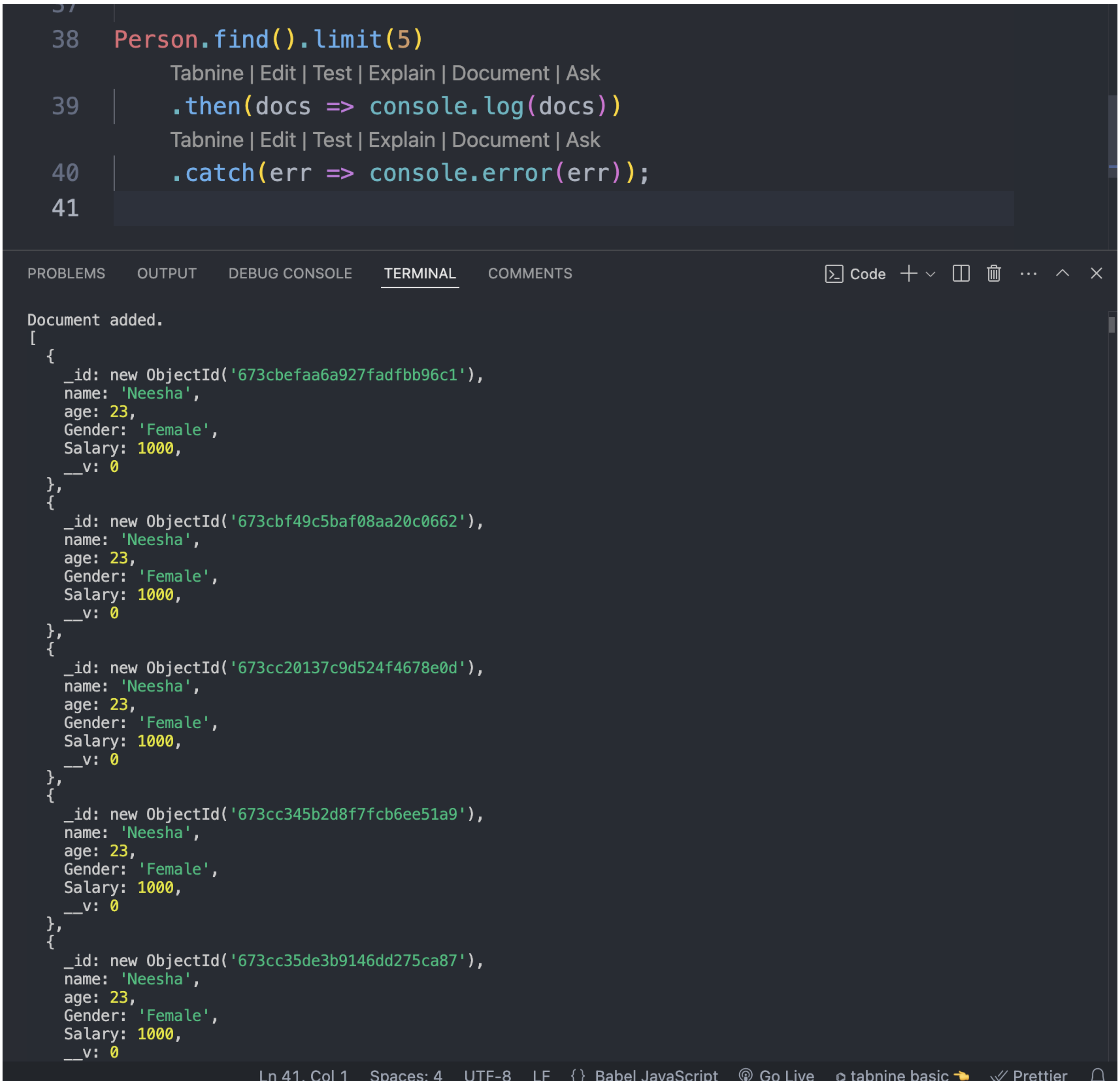
```
const manyPersons = [
    { name: 'Simon', age: 42, Gender: "Male", Salary: 3456 },
    { name: 'Neesha', age: 23, Gender: "Female", Salary: 1000 },
    { name: 'Mary', age: 27, Gender: "Female", Salary: 5402 },
    { name: 'Mike', age: 40, Gender: "Male", Salary: 4519 }
];

Person.insertMany(manyPersons)
    .then(() => console.log("Multiple documents added."))
    .catch(err => console.error(err));
```

Tabnine | Edit | Test | Explain | Document | Ask
Tabnine | Edit | Test | Explain | Document | Ask
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
COMMENTS
Code
Code

```
node "/Users/einsstark/Documents/University/Web and Mobile Application Development - CN5006/Week 8/we
ek8/index.js"
einsstark@Einsstarks-Air Week 8 % node "/Users/einsstark/Documents/University/Web and Mobile Applicat
ion Development - CN5006/Week 8/week8/index.js"
(node:79533) [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:79533) [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
Connected to mongodb://localhost:27017/Week8
Document added.
Multiple documents added.
```

Task 3: Fetch All Data

I used `.find()` to get all documents from the database. It only fetched 5 since limit was set to 5.

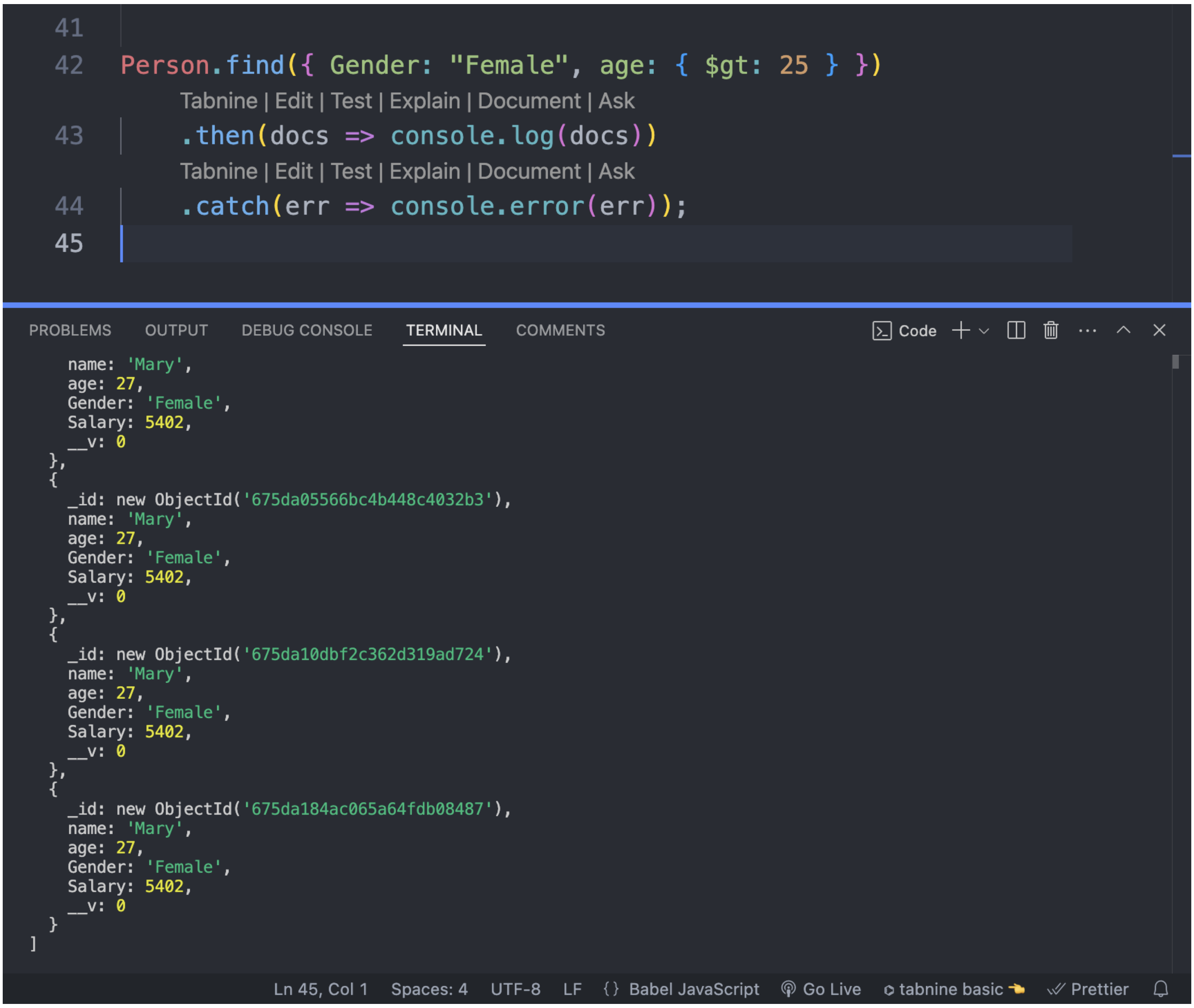


```
Person.find().limit(5)
    .then(docs => console.log(docs))
    .catch(err => console.error(err));
```

```
Document added.
[
  {
    _id: new ObjectId('673cbefaa6a927fadfbb96c1'),
    name: 'Neesha',
    age: 23,
    Gender: 'Female',
    Salary: 1000,
    __v: 0
  },
  {
    _id: new ObjectId('673cbf49c5baf08aa20c0662'),
    name: 'Neesha',
    age: 23,
    Gender: 'Female',
    Salary: 1000,
    __v: 0
  },
  {
    _id: new ObjectId('673cc20137c9d524f4678e0d'),
    name: 'Neesha',
    age: 23,
    Gender: 'Female',
    Salary: 1000,
    __v: 0
  },
  {
    _id: new ObjectId('673cc345b2d8f7fcb6ee51a9'),
    name: 'Neesha',
    age: 23,
    Gender: 'Female',
    Salary: 1000,
    __v: 0
  },
  {
    _id: new ObjectId('673cc35de3b9146dd275ca87'),
    name: 'Neesha',
    age: 23,
    Gender: 'Female',
    Salary: 1000,
    __v: 0
```

Task 4: Filtered Find Command

I used `.find()` with filters to get only females older than 25.

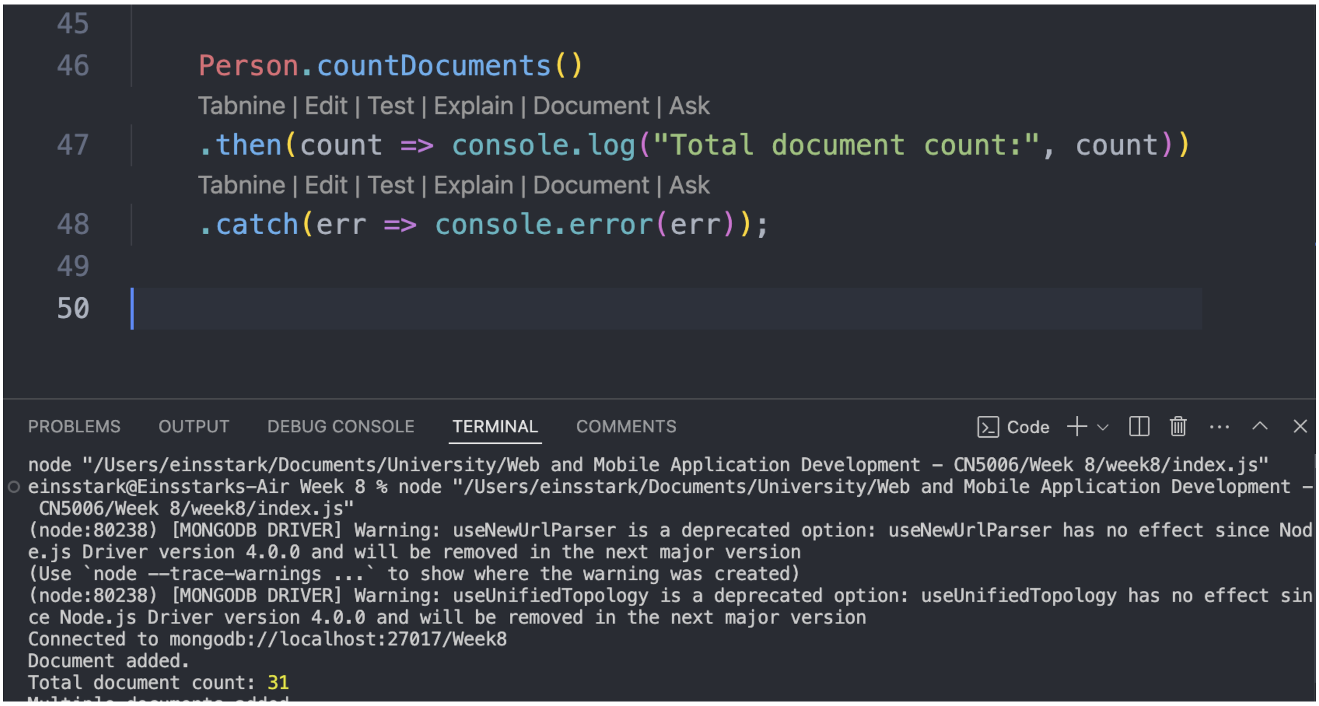


```javascript

Person.find({ Gender: "Female", age: { $gt: 25 } })
    .then(docs => console.log(docs))
    .catch(err => console.error(err));

```

```
    name: 'Mary',
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  },
  {
    _id: new ObjectId('675da05566bc4b448c4032b3'),
    name: 'Mary',
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  },
  {
    _id: new ObjectId('675da10dbf2c362d319ad724'),
    name: 'Mary',
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  },
  {
    _id: new ObjectId('675da184ac065a64fdb08487'),
    name: 'Mary',
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  }
]
```

Task 5: Count Documents

`.countDocuments()` to count how many records are in the collection.

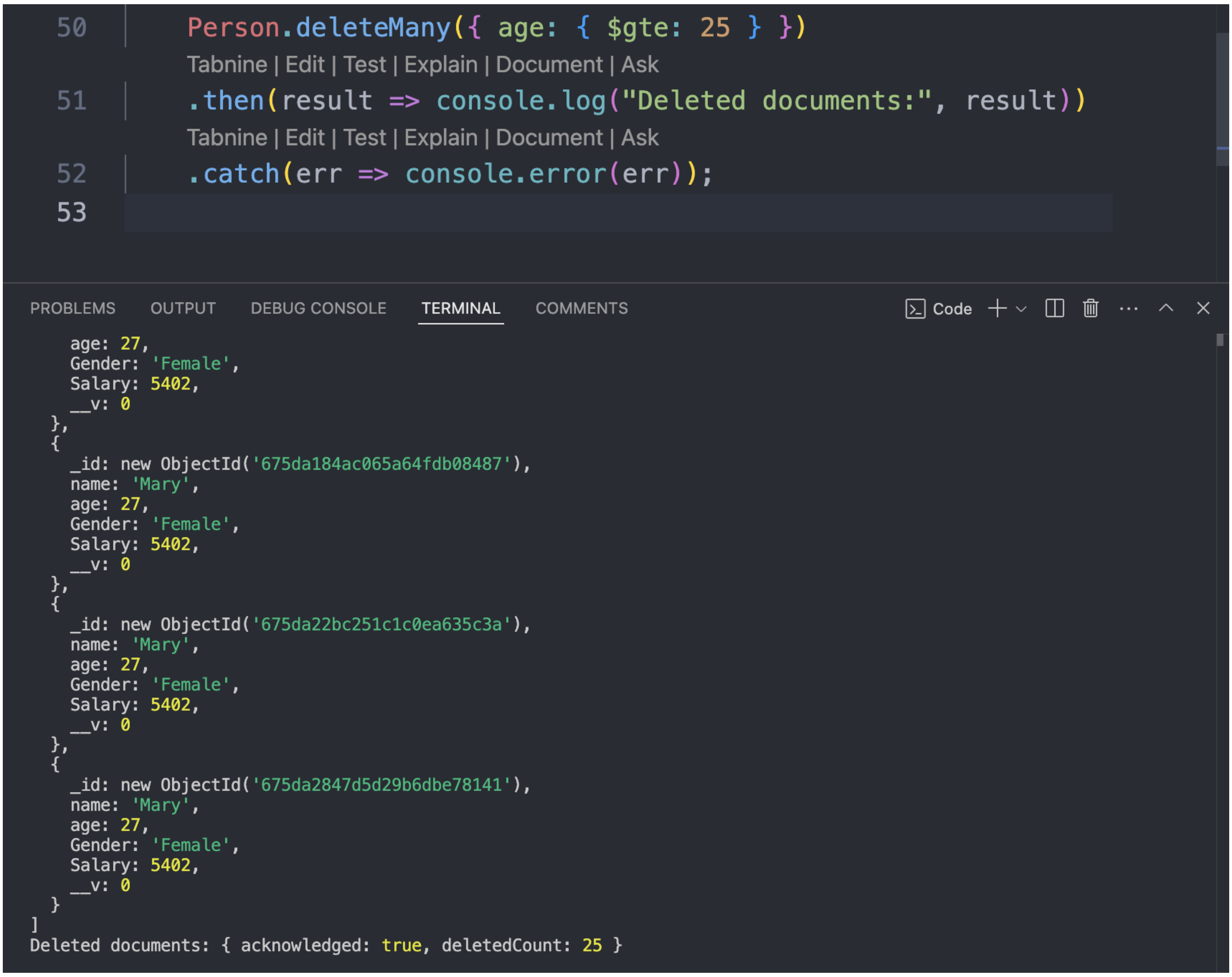


```javascript

    Person.countDocuments()
    .then(count => console.log("Total document count:", count))
    .catch(err => console.error(err));


```

```
node "/Users/einsstark/Documents/University/Web and Mobile Application Development - CN5006/Week 8/week8/index.js"
○ einsstark@Einsstarks-Air Week 8 % node "/Users/einsstark/Documents/University/Web and Mobile Application Development - CN5006/Week 8/week8/index.js"
(node:80238) [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:80238) [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
Connected to mongodb://localhost:27017/Week8
Document added.
Total document count: 31
Multiple documents added.
```

Task 6: Delete Documents

To delete documents with `age >= 25`:

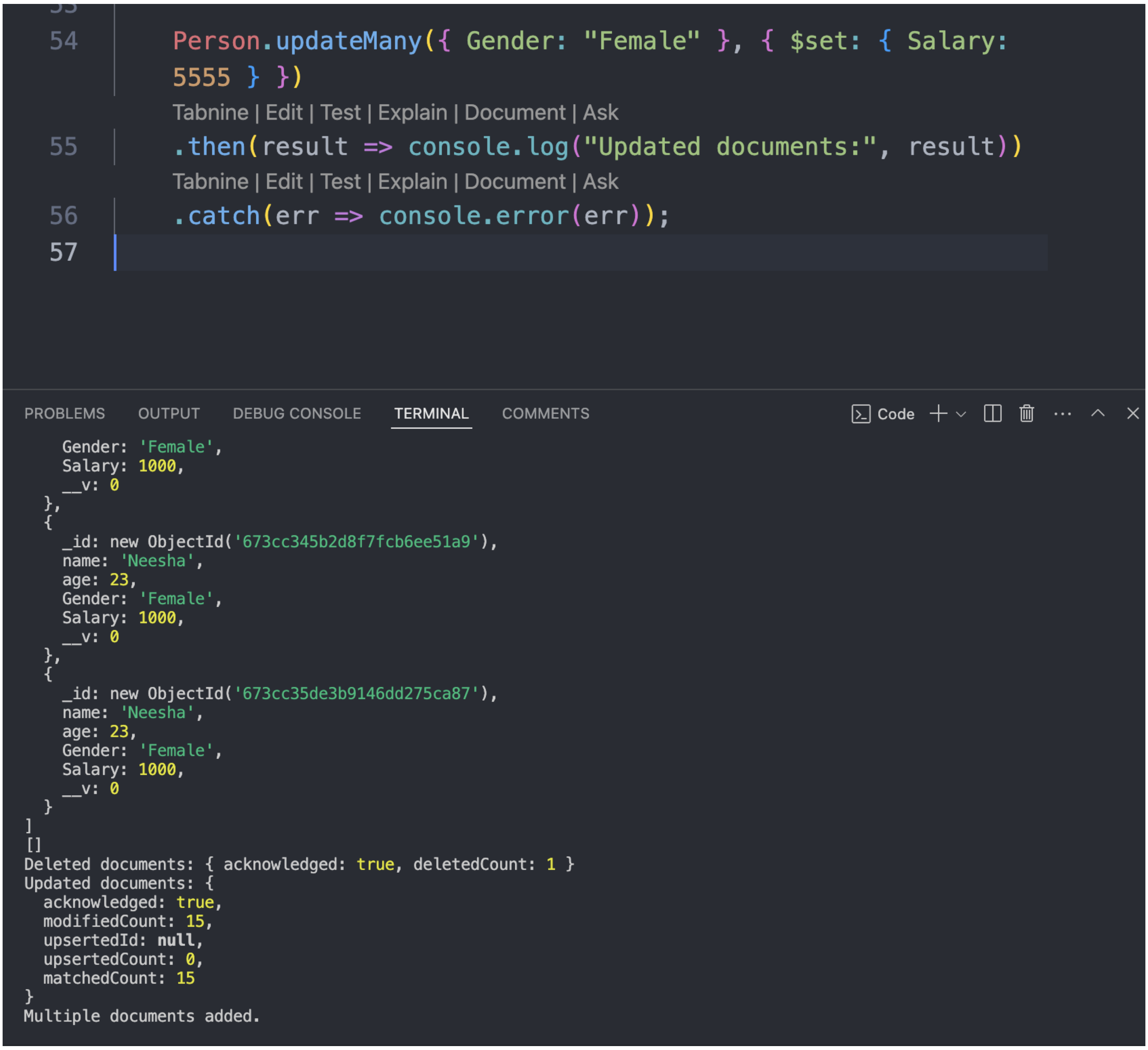


```
    Person.deleteMany({ age: { $gte: 25 } })
    .then(result => console.log("Deleted documents:", result))
    .catch(err => console.error(err));
```

```
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  },
  {
    _id: new ObjectId('675da184ac065a64fdb08487'),
    name: 'Mary',
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  },
  {
    _id: new ObjectId('675da22bc251c1c0ea635c3a'),
    name: 'Mary',
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  },
  {
    _id: new ObjectId('675da2847d5d29b6dbe78141'),
    name: 'Mary',
    age: 27,
    Gender: 'Female',
    Salary: 5402,
    __v: 0
  }
]
Deleted documents: { acknowledged: true, deletedCount: 25 }
```

Task 7: Update Documents

Here, I updated the `Salary` to 5555 for all documents with `Gender: "Female"`:



```
Person.updateMany({ Gender: "Female" }, { $set: { Salary:
5555 } })
.then(result => console.log("Updated documents:", result))
.catch(err => console.error(err));
```

Tabnine | Edit | Test | Explain | Document | Ask
Tabnine | Edit | Test | Explain | Document | Ask
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
COMMENTS
Code

```
    Gender: 'Female',
    Salary: 1000,
    __v: 0
  },
  {
    _id: new ObjectId('673cc345b2d8f7fcb6ee51a9'),
    name: 'Neesha',
    age: 23,
    Gender: 'Female',
    Salary: 1000,
    __v: 0
  },
  {
    _id: new ObjectId('673cc35de3b9146dd275ca87'),
    name: 'Neesha',
    age: 23,
    Gender: 'Female',
    Salary: 1000,
    __v: 0
  }
]
[]
Deleted documents: { acknowledged: true, deletedCount: 1 }
Updated documents: {
  acknowledged: true,
  modifiedCount: 15,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 15
}
Multiple documents added.
```

2. Reflection

This week, using using MongoDB CRUD operations extended the tutorial from week 2, where i learned how collections are just organised data and databases hold it all together. In the lab, I learned how using Mongoose made it way easier to interact with the database without writing raw queries.

I learned various operations. Using `.save()` for a single document was straightforward, then `.insertMany()` to add ton of data. I struggled with `.find()` though, especially getting filters right, but once I saw only females over 25 show up, it felt rewarding. Counting documents counted how much data I was dealing with. Updating salaries with `.updateMany()` was fun, and it made me realize how much control I had over the data- that I update anything already defined. Now, MongoDB does not seem difficult now.