Case Study:

Simple Bookstore API (MongoDB + Node.js)

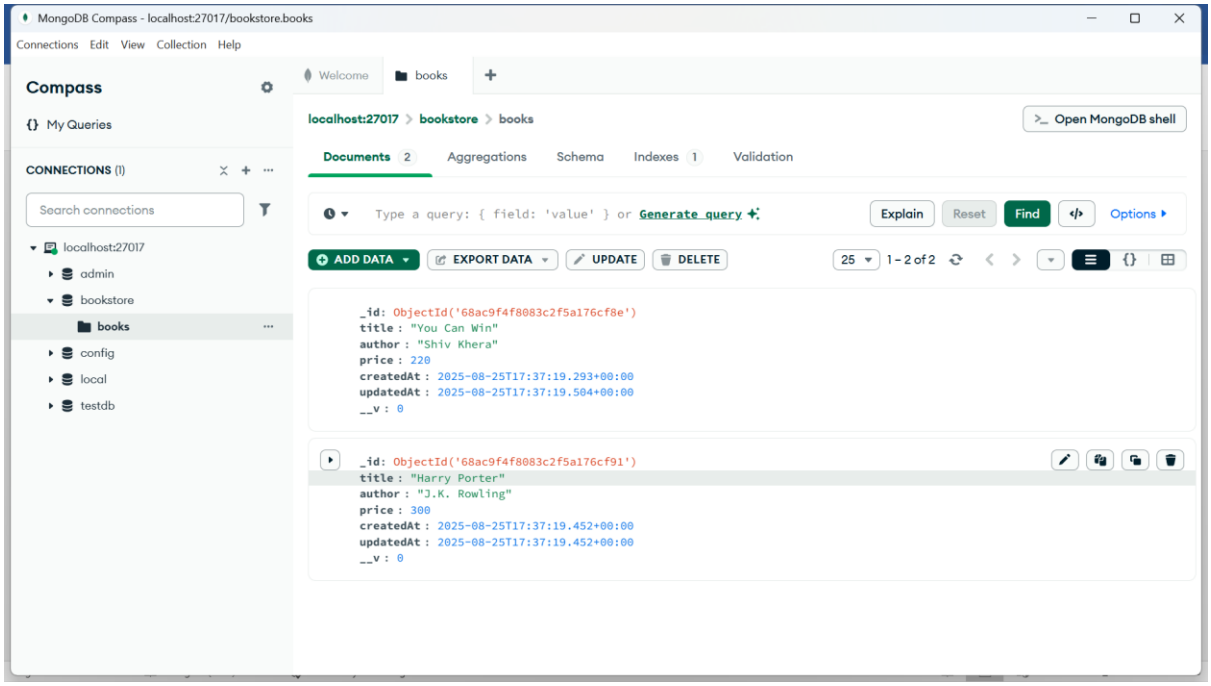
Create a Bookstore where you can:

1. **Add books (title, author, price) 2. List all books 3. Find a book by title 4. Update book price**

```
// bookstore.js
const mongoose = require("mongoose");

const uri = "mongodb://127.0.0.1:27017/bookstore";
mongoose.connect(uri, { useNewUrlParser: true, useUnifiedTopology: true })
  .then(() => console.log("Connected to MongoDB"))
  .catch(err => console.error("MongoDB connection error:", err));

const bookSchema = new mongoose.Schema({
  title: { type: String, required: true },
  author: { type: String, required: true },
  price: { type: Number, required: true }
}, { timestamps: true });

const Book = mongoose.model("Book", bookSchema);

async function addBook(title, author, price) {
  const book = new Book({ title, author, price });
  await book.save();
  console.log("Book added:", book);
}

async function getAllBooks() {
  const books = await Book.find();
```

```
  console.log("All books:", books);
}

async function findBookByTitle(title) {
  const book = await Book.findOne({ title });
  if (book) {
    console.log("Book found:", book);
  } else {
    console.log("Book not found");
  }
}

async function updateBookPrice(title, newPrice) {
  const book = await Book.findOneAndUpdate(
    { title },
    { price: newPrice },
    { new: true }
  );
  if (book) {
    console.log("Updated book:", book);
  } else {
    console.log("Book not found");
  }
}

async function main() {

  await addBook("You Can Win", "Shiv Khera", 200);
  await addBook("Harry Porter", "J.K. Rowling", 300);

  await getAllBooks();
```

```
  await findBookByTitle("1984");

  await updateBookPrice("1984", 220);

  await getAllBooks();
}

main()
  .then(() => mongoose.disconnect())
  .catch(err => {
    console.error("Error:", err);
    mongoose.disconnect();
  });
```



Case Study:

Simple Employee Management System

Build a Node.js app with MySQL to manage employees.

Features: 1. Add new employees (name, email, department). 2. List all employees. 3. Update employee information. 4. Delete an employee.

```
// employeeManagement.js
const mysql = require('mysql2/promise');

const dbConfig = {
  host: 'localhost',
  user: 'root',
  password: 'Lalithakumar4180',
  database: 'employeeDB'
};

let connection;

async function connectDB() {
  connection = await mysql.createConnection(dbConfig);
  console.log("Connected to MySQL database");
}

async function addEmployee(name, email, department) {
  const query = 'INSERT INTO employees (name, email, department) VALUES (?, ?, ?)';
  try {
    const [result] = await connection.execute(query, [name, email, department]);
    console.log("Employee added with ID:", result.insertId);
  } catch (err) {
    console.error("Error adding employee:", err.message);
  }
```

```
}

async function getAllEmployees() {
  const query = 'SELECT * FROM employees';
  const [rows] = await connection.execute(query);
  console.log("All employees:", rows);
}

async function updateEmployee(id, name, email, department) {
  const query = 'UPDATE employees SET name = ?, email = ?, department = ? WHERE id = ?';
  const [result] = await connection.execute(query, [name, email, department, id]);
  if (result.affectedRows > 0) {
    console.log(`Employee ID ${id} updated successfully`);
  } else {
    console.log(`Employee ID ${id} not found`);
  }
}

async function deleteEmployee(id) {
  const query = 'DELETE FROM employees WHERE id = ?';
  const [result] = await connection.execute(query, [id]);
  if (result.affectedRows > 0) {
    console.log(`Employee ID ${id} deleted successfully`);
  } else {
    console.log(`Employee ID ${id} not found`);
  }
}

async function main() {
  await connectDB();
```

```
  await addEmployee("Lalitha", "L@example.com", "CSE");
  await addEmployee("Pavan Birlangi", "PB@example.com", "IT");

  await getAllEmployees();

  await updateEmployee(1, "Lalitha Birlangi", "LB@example.com", "IT");

  await deleteEmployee(2);

  await getAllEmployees();

  await connection.end();
  console.log("MySQL connection closed");
}

main().catch(err => console.error(err));
```