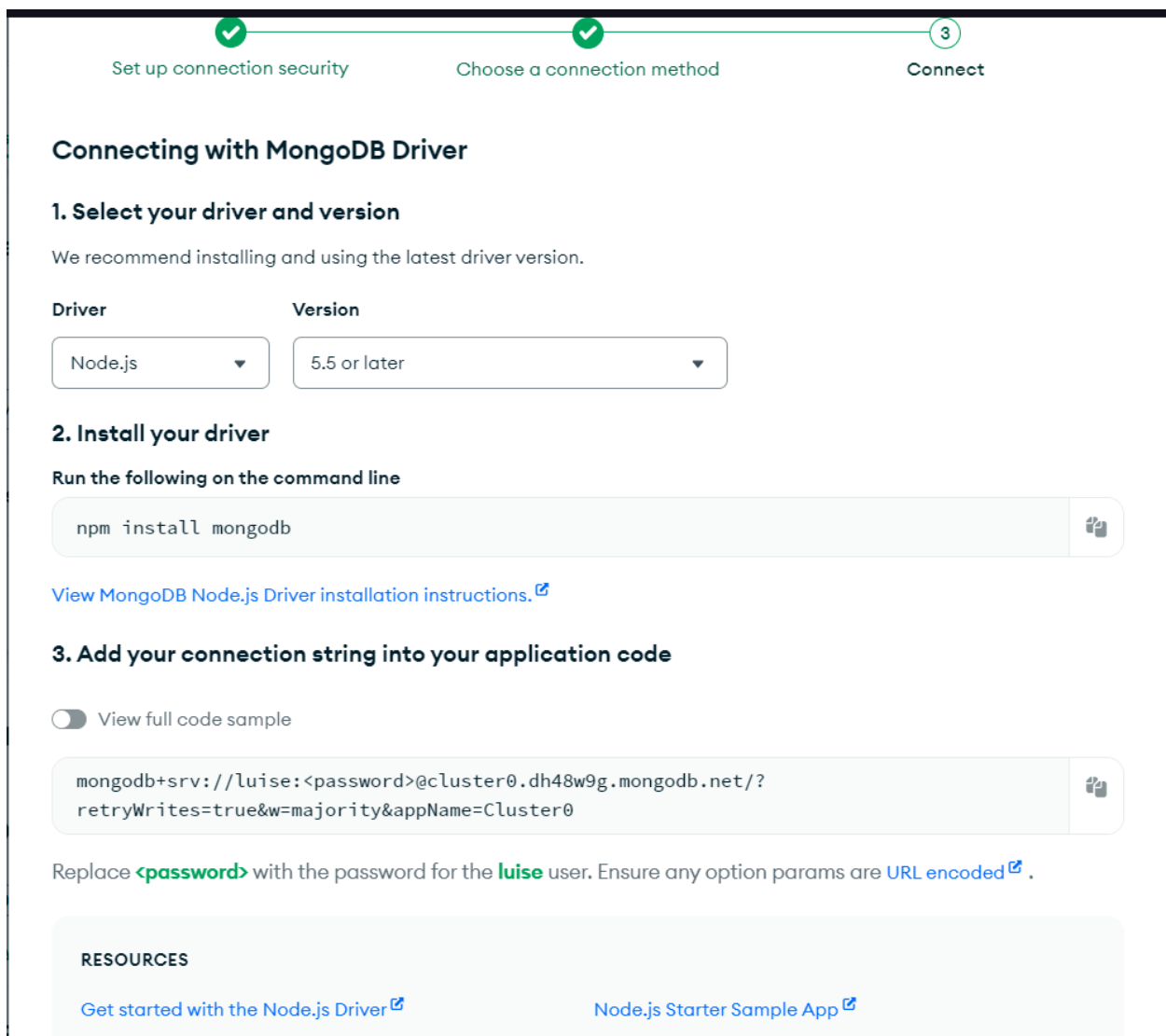
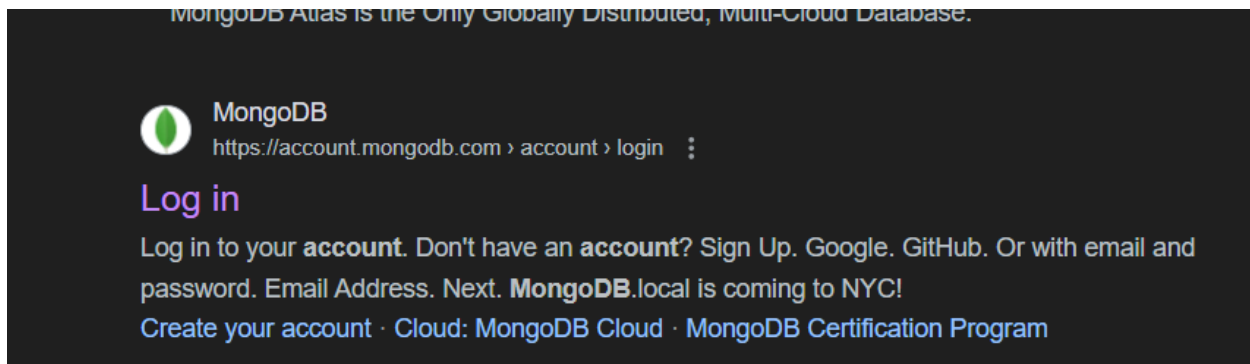


1<sup>st</sup> step is I made MongoDB Account and made cluster 0 :



2<sup>nd</sup> step is get the String connection and use express to route it and fetch:

```
//Indicate the connection string from mongodb
var CONNECTION_STRING = "mongodb+srv://luise:luisedb29@cluster0.dh48w9g.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0";
```

3<sup>rd</sup> step is install all packages using npm

```
"license": "ISC",
"dependencies": {
  "cors": "^2.8.5",
  "express": "^4.19.2",
  "mongodb": "^4.1.0",
  "multer": "^1.4.5-lts.1"
```

4<sup>th</sup> step is import httpClient to fetch the APIs

```
import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http';
```

```
38
39 app.post('/api/books/AddBook', multer().none(), async (req, res) => {
40   try {
41     const numOfDocs = await database.collection("Books").countDocuments();
42     await database.collection("Books").insertOne({
43       id: (numOfDocs + 1),
44       title: req.body.title,
45       description: req.body.description,
46       price: parseFloat(req.body.price).toFixed(2)
47     });
48     res.json("Added Successfully");
49   } catch (error) {
50     console.error("Error adding book:", error);
51     res.status(500).json({ error: "Failed to add book" });
52   }
53 });
54
55 HTML CODE:
56 app.delete('/api/books/DeleteBook', (req, res) => {
57   database.collection("Books").deleteOne({
58     id: parseInt(req.query.id)
59   });
60   res.json("Deleted successfully!");
```

5<sup>th</sup> is format and create css and html with fetching it to the view

```

export class AppComponent {
  title = 'booksapp';
  //set the link of the based route
  readonly APIUrl="http://localhost:5038/api/books/";

  constructor(private http:HttpClient){
    //initialize the books array
    books:any=[];
  }

  refreshBooks(){
    this.http.get(this.APIUrl+'GetBooks').subscribe(data=>{
      this.books=data;
    });
  }

  ngOnInit(){
    this.refreshBooks();
  }
}

```

```

</tr>
</thead>
<tbody *ngFor="let book of books">
  <tr>
    <td>{{book.title}}</td>
    <td>{{book.description}}</td>
    <td>{{book.price}}</td>
    <td><button (click)="deleteBook(book.id)">Delete book</button></td>
  </tr>
</tbody>
</table>

```

HTML CODE:

The screenshot shows a web browser at localhost:4200 displaying the 'Books Application'. The application has a form to add a new book with fields for title, description, and price, and an 'Add book' button. Below the form is a table listing books with columns for Title, Description, Price, and a 'Delete book' button.

Title	Description	Price	
How to Cook Adobo	A Filipino Cookbook	150	Delete book
How to train your dragon	a book for training your dragon	210	Delete book
Moonlight	a knowledgeable book about moons	250	Delete book
how to unlove someone	a book about moving on	150	Delete book
One Piece	volume 700	550	Delete book
IT matters	book for Information Technology	200	Delete book

The VS Code editor shows the HTML code for the application component:

```

1 <div id="container">
2   <h1>Books Application</h1>
3   <br>
4   <div class="bookapply">
5     <input id="newBook" placeholder="title"> <br>
6     <input id="newDesc" placeholder="description"> <br>
7     <input id="newPrice" placeholder="book price"> <br>
8     <button (click)="addBook()">Add book</button>
9   </div>
10
11   <table>
12     <thead>
13       <tr>
14         <th>Title</th>
15         <th>Description</th>
16         <th>Price</th>
17         <th></th>
18       </tr>
19     </thead>
20     <tbody *ngFor="let book of books">
21       <tr>
22         <td>{{book.title}}</td>
23         <td>{{book.description}}</td>
24         <td>{{book.price}}</td>
25         <td>Delete book</td>
26       </tr>
27     </tbody>
28   </table>
29 </div>

```

CSS CODE:

The screenshot shows the same web browser displaying the 'Books Application' with CSS styling applied. The form and table are now styled with a light pink background and rounded corners. The VS Code editor shows the CSS code for the application component:

```

1 table, tr, td {
2   border: 3px solid #f0f0f0;
3   border-collapse: collapse;
4   text-align: center;
5   padding: 20px;
6 }
7
8
9
10 tbody:nth-child(even) {
11   background-color: #f0f0f0;
12 }
13
14 #container {
15   min-height: 100vh;
16   padding: 5%;
17   height: auto;
18   width: 100%;
19   display: flex;
20   align-items: center;
21   background: linear-gradient(to top right, #f0f0f0 1%, #f0f0f0 1%, #f0f0f0 99%, #f0f0f0 99%);
22   flex-direction: column;
23 }

```

Ts code:

The screenshot shows a web browser at localhost:4200 displaying the 'Books Application'. The application has a form to add a new book with fields for title, description, and price, and an 'Add book' button. Below the form is a table listing existing books with columns for Title, Description, Price, and a 'Delete book' button for each row.

Title	Description	Price	
How to Cook Adobo	A filipino Cookbook	150	Delete book
How to train your dragon	a book for training your dragon	210	Delete book
Moonlight	a knowledgable book about moons	250	Delete book
how to unlove someone	a book about moving on	150	Delete book
One Piece	volume 700	550	Delete book
IT matters	book for Information Technology	200	Delete book

The VS Code editor shows the `app.component.ts` file with the following code:

```

1 title = 'booksapp';
2 //set the link of the based route
3 readonly APIurl='http://localhost:5038/api/books/';
4
5 constructor(private http:HttpClient){
6 }
7 //initialize the books array
8 books:any=[];
9
10 refreshBooks(){
11   this.http.get(this.APIurl+'GetBooks').subscribe(data=>{
12     this.books=data;
13   })
14 }
15
16 ngOnInit(){
17   this.refreshBooks();
18   console.log(this.books);
19 }
20
21 addBook(){
22   var newBook=(<HTMLInputElement>document.getElementById("newBook")).val
23   var newDesc=(<HTMLInputElement>document.getElementById("newDesc")).val

```

The terminal shows the command `ls` and the output of the directory listing:

```

Directory: C:\Users\luise\OneDrive\Desktop\school\code\mean-finals\API
Mode                LastWriteTime         Length Name
----                -
da---l             4/12/2024   3:01 PM             node_modules
-a---l             4/12/2024   5:36 PM             index.js
-a---l             4/12/2024   3:01 PM          39916 package-lock.json
-a---l             4/12/2024   3:01 PM             344 package.json

```

Index.js Code:

The screenshot shows the same web browser interface as before. The VS Code editor shows the `index.js` file with the following code:

```

1 var express = require('express');
2 var MongoClient = require('mongodb').MongoClient;
3 var cors = require('cors');
4 const multer = require('multer');
5
6 //Create an instance of express app
7 var app=express();
8 //Make use of the CORS module
9 app.use(cors());
10
11 //Indicate the connection string from mongodb
12 var CONNECTION_STRING = "mongodb+srv://luise:luisedb29@cluster0.dh4dw9g.mongodb.net/";
13
14 //Indicate the name of the database
15 var DATABASENAME = "mydb";
16
17 //Instantiate the mongodb client
18 var database;
19
20 //create a listener
21 app.listen(5038, ()=>{
22   MongoClient.connect(CONNECTION_STRING,(error,client)>{
23     database=client.db(DATABASENAME);
24     console.log("Yay! Now connected to Cluster");
25   })
26 })
27
28 //ROUTES TO ALL ACTIONS
29 //get all dbase data
30 app.get('/api/books/GetBooks',(req, res) => {
31   database.collection("Books").find({}).toArray((error,result)>{
32     res.send(result);
33   })
34 })
35
36 app.post('/api/books/AddBook', multer().none(), async (req, res) => {

```

Cluster 0 of Mongoddb:

Atlas

Luise Gian's ...

Access Manager

Billing

All ClustersGet HelpLuise Gian

Project 0

Data ServicesApp ServicesCharts

Overview

DEPLOYMENTDatabaseData LakeSERVICESDevice SyncTriggersData APIData FederationAtlas SearchStream ProcessingMigrationSECURITYQuickstartBackupDatabase AccessNetwork AccessAdvanced

Search Namespaces

mydbBooks

sample\_mflix

STORAGE SIZE: 36KBLOGICAL DATA SIZE: 676BTOTAL DOCUMENTS: 6INDEXES TOTAL SIZE: 36KB

FindIndexesSchema Anti-PatternsAggregationSearch Indexes

INSERT DOCUMENT

FilterType a query: { field: 'value' }ResetApplyOptions

\_id: ObjectId('66172dfaf2bbc85280f018ae')

id: 5

title: "One Piece"

description: "volume 700"

price: 550

\_id: ObjectId('661a1577702826faec1025eb')

id: 6

title: "this is the newest book"

description: "A new book by Luise Yambao"

price: "2300.00"

System Status: All Good

©2024 MongoDB, Inc. StatusTermsPrivacyAtlas BlogContact Sales

Books Application

this is the newest book

A new book by Luise Yamba

2300

Add book

Title	Description	Price	
How to Cook Adobo	A filipino Cookbook	150	Delete book
How to train your dragon	a book for training your dragon	210	Delete book
Moonlight	a knowledgeable book about moons	250	Delete book
how to unlove someone	a book about moving on	150	Delete book
One Piece	volume 700	550	Delete book
this is the newest book	A new book by Luise Yambao	2300.00	Delete book