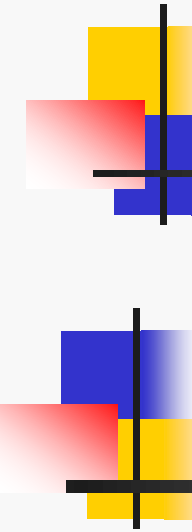
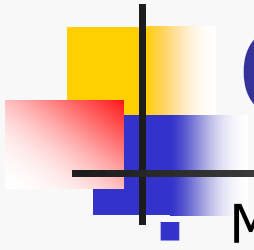


DMET 601
Web Technologies and Usability

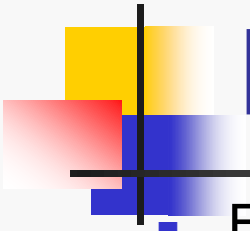
MERN Stack

Prof. Dr. Rimon Elias



Contents

- MERN stack



Full Stack

- Full stack web development includes both client-side and server-side scripting.

- Full Stack comprises
  - Programming a **browser** using (e.g., JavaScript, jQuery, Angular, or Vue)
  - Programming a **server** using (e.g., PHP, ASP, Python, or Node)
  - Programming a **database** using (e.g., SQL, SQLite, or MongoDB)
  - This is in addition to HTML and CSS.

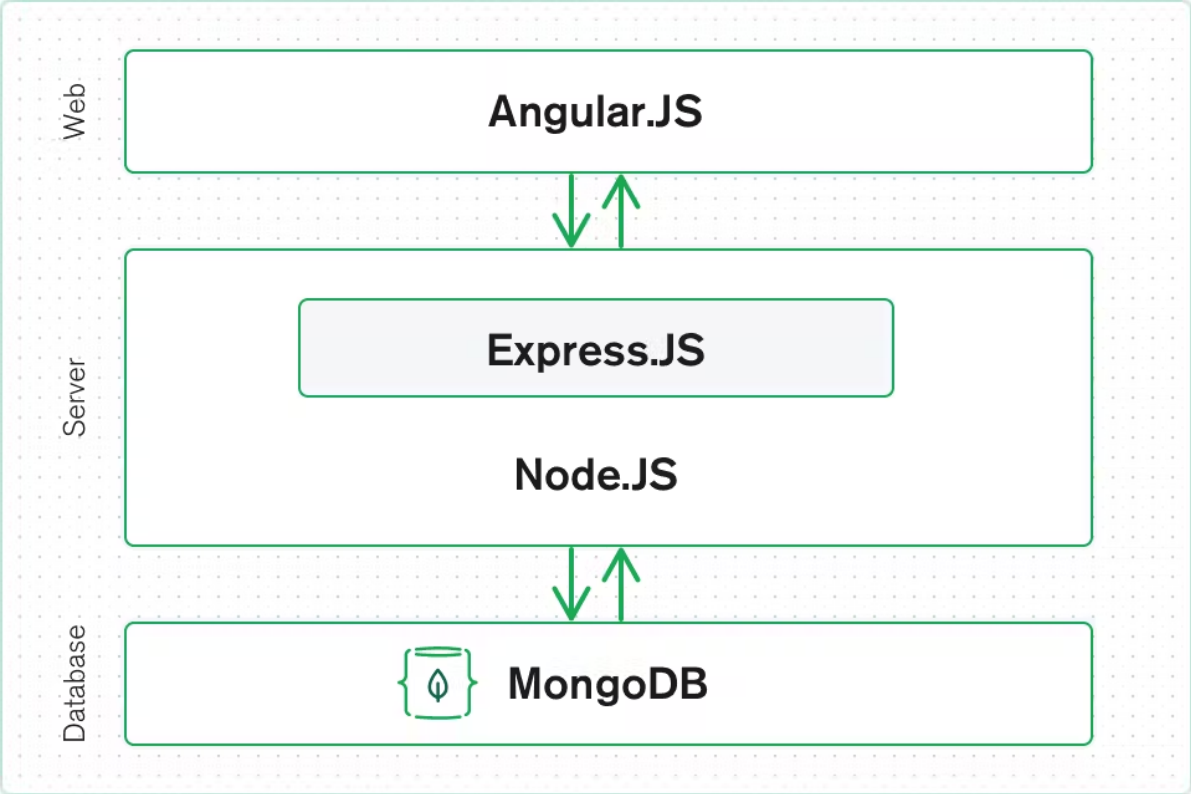
MEAN Stack

- The MEAN stack is a JavaScript-based framework for developing web applications.
- MEAN is named after **M**ongoDB, **E**xpress, **A**ngular, and **N**ode
  - MongoDB — document database
  - Express(.js) — Node.js web framework
  - Angular(.js) — a client-side JavaScript framework
  - Node(.js) — the premier JavaScript web server

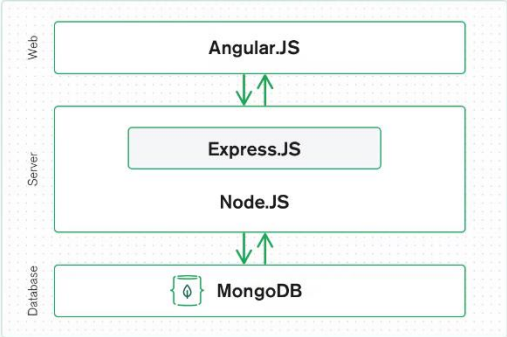
- Variations:
  - MERN (replacing Angular.js with React.js)
  - MEVN (replacing Angular.js with Vue.js)
- https://www.mongodb.com/mean-stack
- https://www.javatpoint.com/mean-stack-tutorial

MEAN Stack

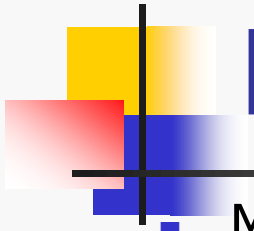
- Using MEAN, one can construct a 3-tier architecture (front end, back end, database) entirely using JavaScript and JSON.



MEAN Installation



- **How to Install the MEAN Stack on Windows**
- Install NodeJS on Windows.
- Install MongoDB on Windows. Create the MongoDB data directory. Start MongoDB Server on Windows.
- Install GIT on Windows.
- Install Angular-CLI on Windows.
  - https://www.simplilearn.com/how-to-install-angular-and-nodejs-on-windows-article
- Install the MEAN Boilerplate on Windows.
- Running Your Sample MEAN Application.
- https://techievor.com/blog/post/how-to-install-the-mean-stack-on-windows



MERN Stack

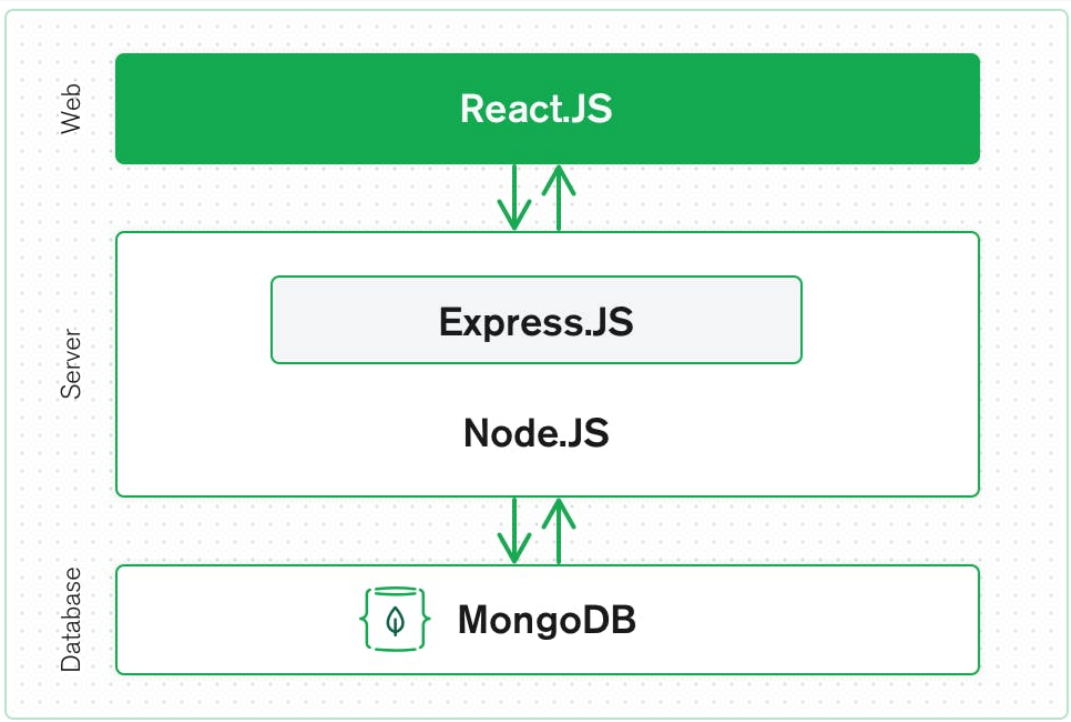
- MERN is one of several variations of the MEAN stack, where the traditional Angular.js front-end framework is replaced with React.js.

- MERN stands for **M**ongoDB, **E**xpress, **R**eact, **N**ode, after the four key technologies that make up the stack.
  - MongoDB — document database
  - Express(.js) — Node.js web framework
  - React(.js) — a client-side JavaScript framework
  - Node(.js) — the premier JavaScript web server

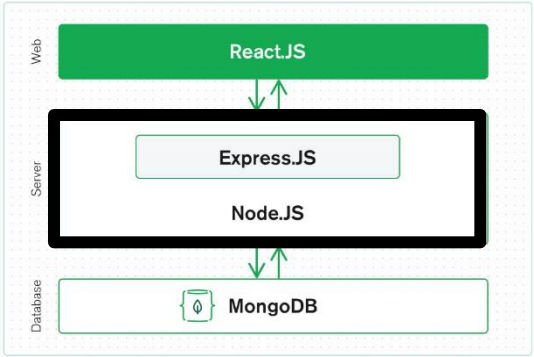
- https://www.mongodb.com/mern-stack

MERN Stack

- Using MERN, one can construct a 3-tier architecture (front end, back end, database) entirely using JavaScript and JSON.

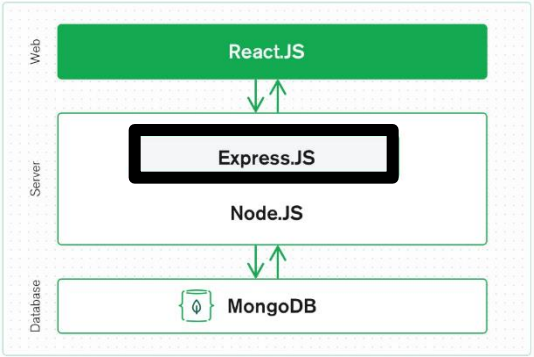


Node.JS



- Node.js is an open source server environment.
- Using Node.js, you can run JavaScript on the server.

Express.JS



- The next level down in the MERN stack is Express.js.
- Express.js runs on a Node.js server.
- Express is a minimal and flexible Node.js web application framework that provides a robust set of features

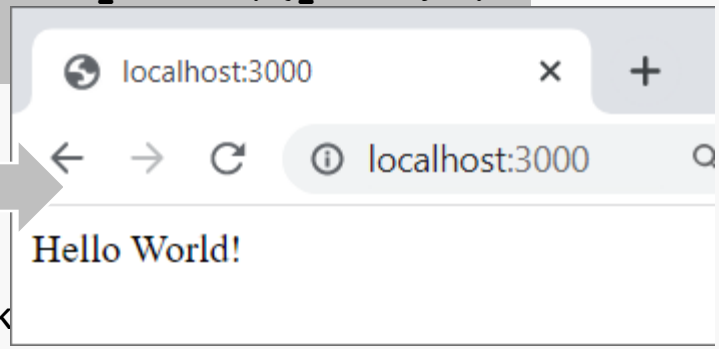
```
npm install express --save
```

Express.JS: Hello World! Example

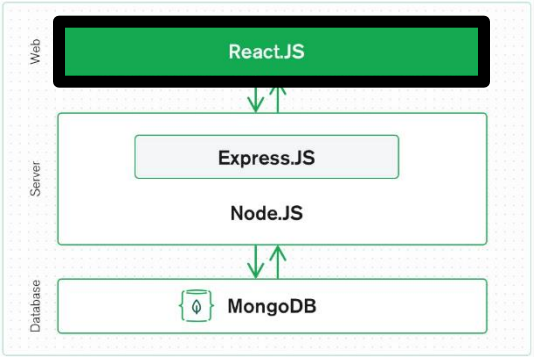
- The next

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

```
F:\>node exp.js
Example app listening on port 3000
```

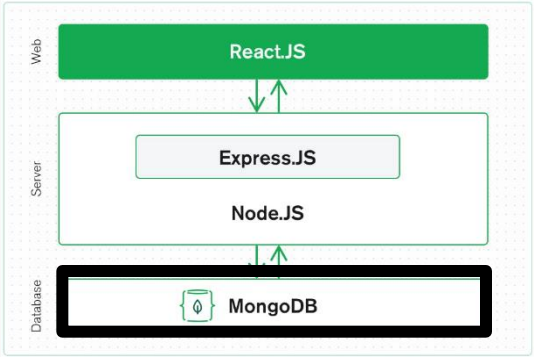


React.js



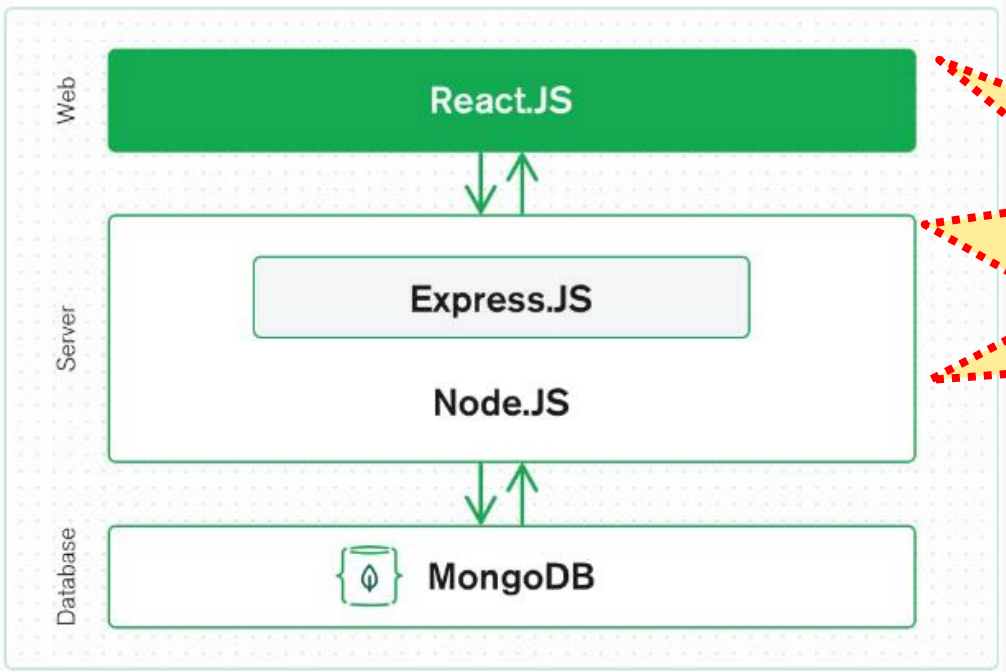
- React.js is the front end top tier of the MERN stack.
- It is a JavaScript framework for creating dynamic client-side applications
- Using React, one can build up interfaces and connect them to data on back-end server.

MongoDB



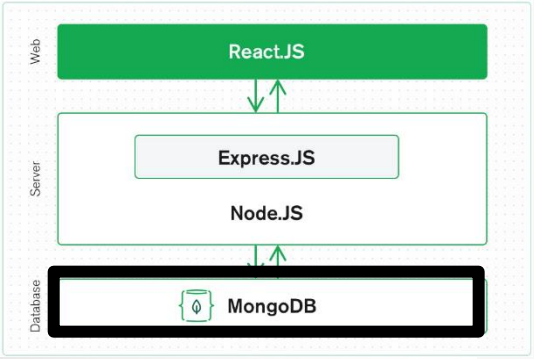
- MongoDB is a **NoSQL** database with support for server-side **JavaScript** execution

MERN

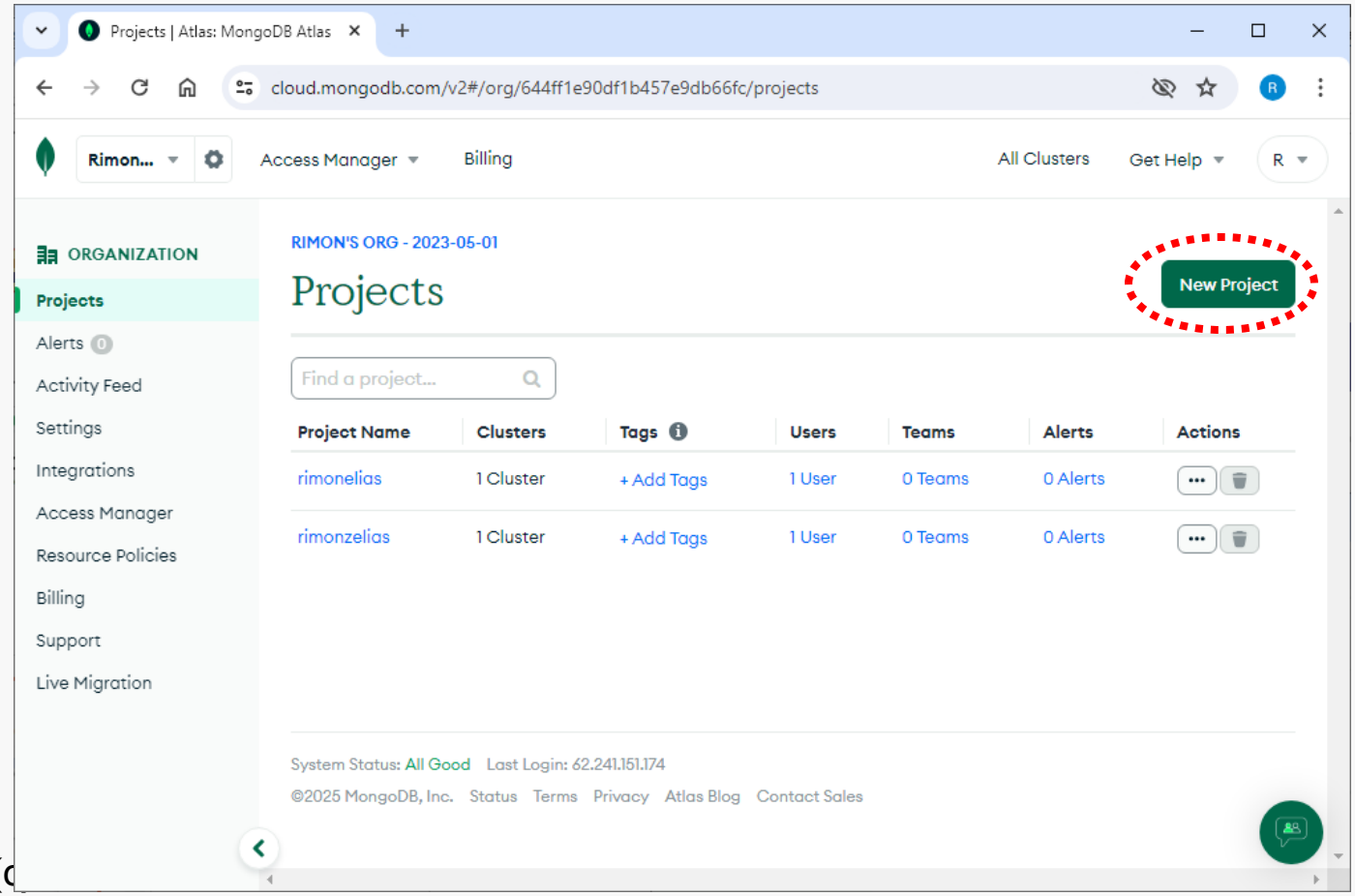


MERN Installation

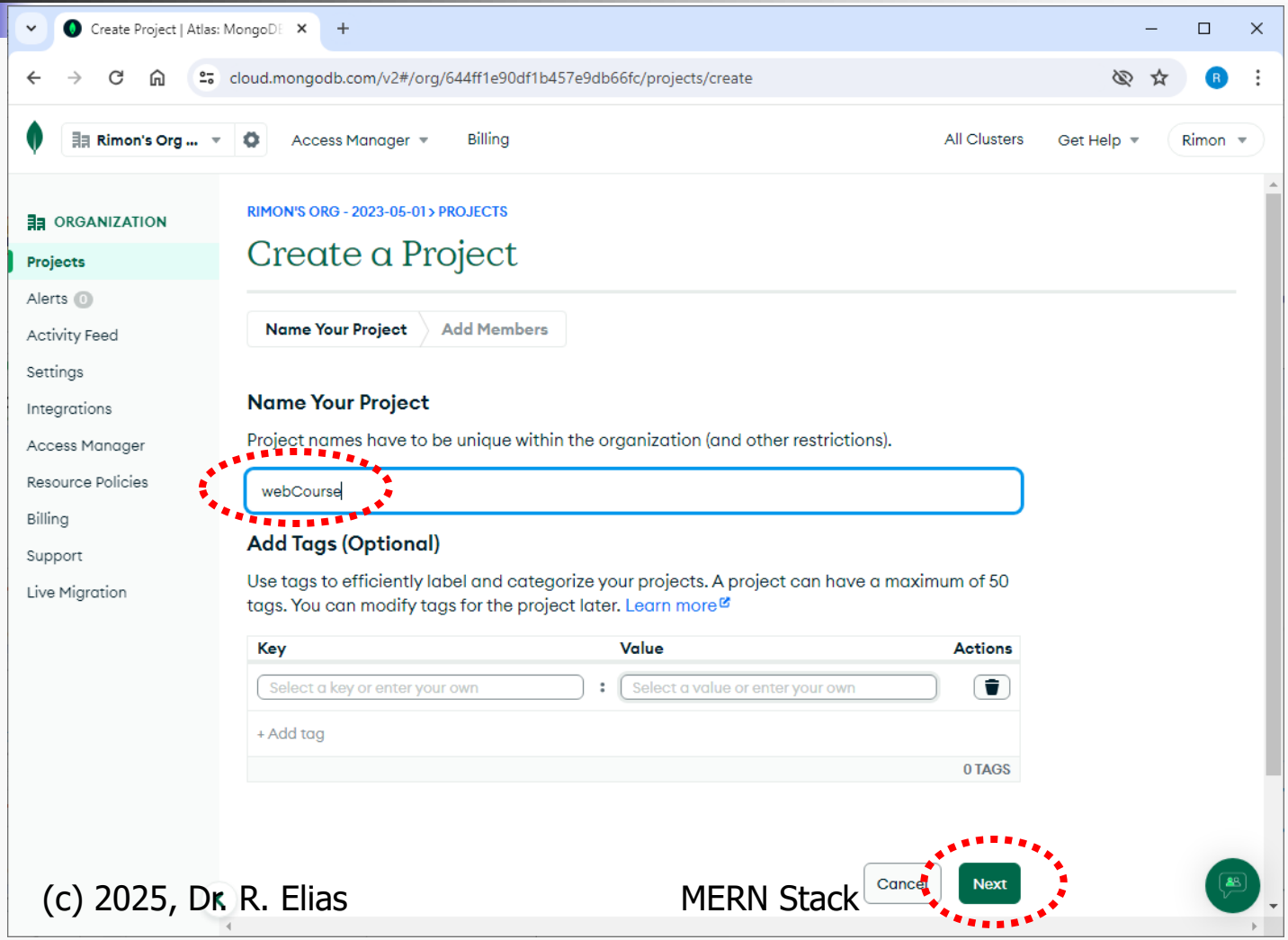
Atlas: Start with MongoDB



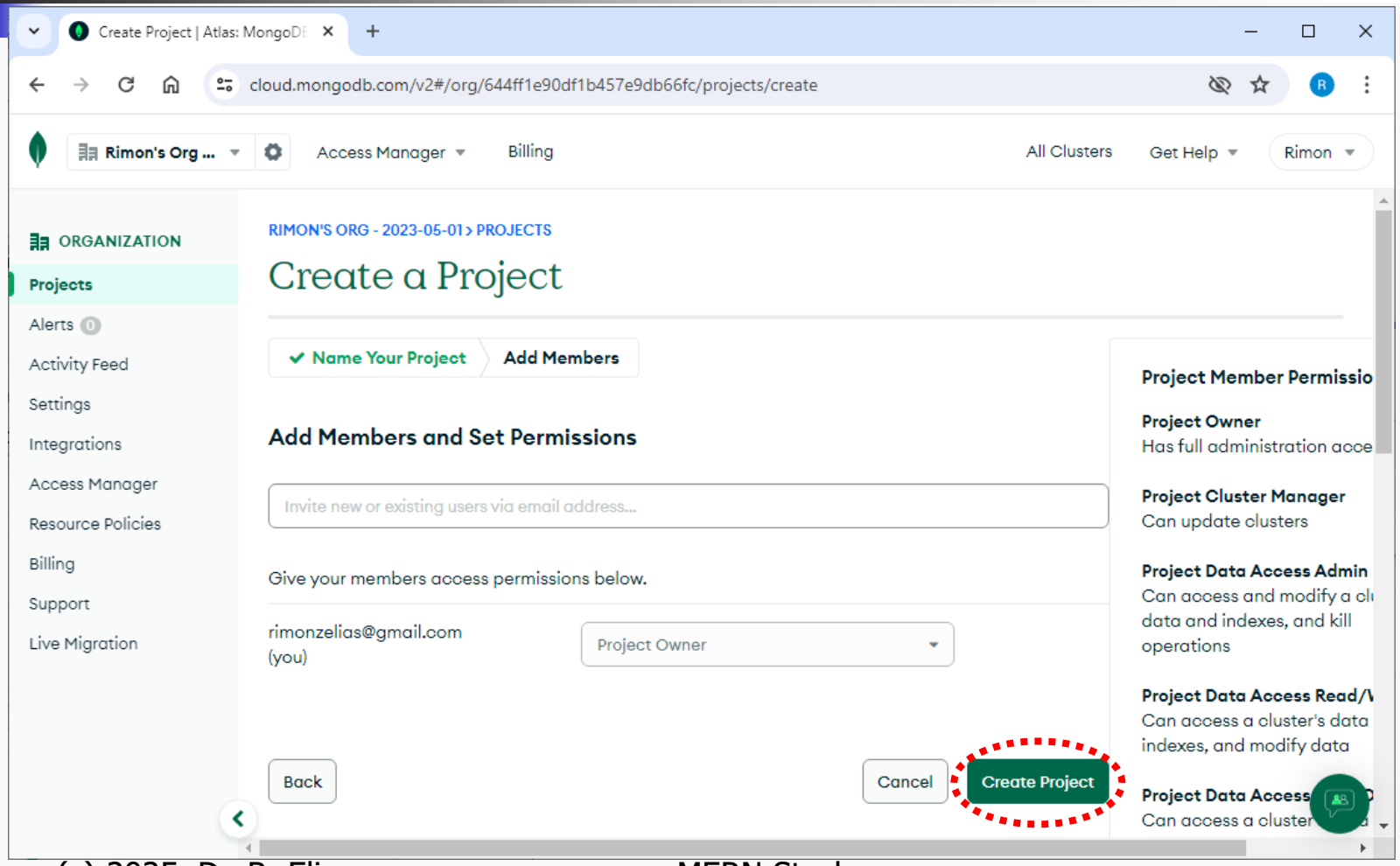
- Go to https://www.mongodb.com/ and sign in and go to “Projects”



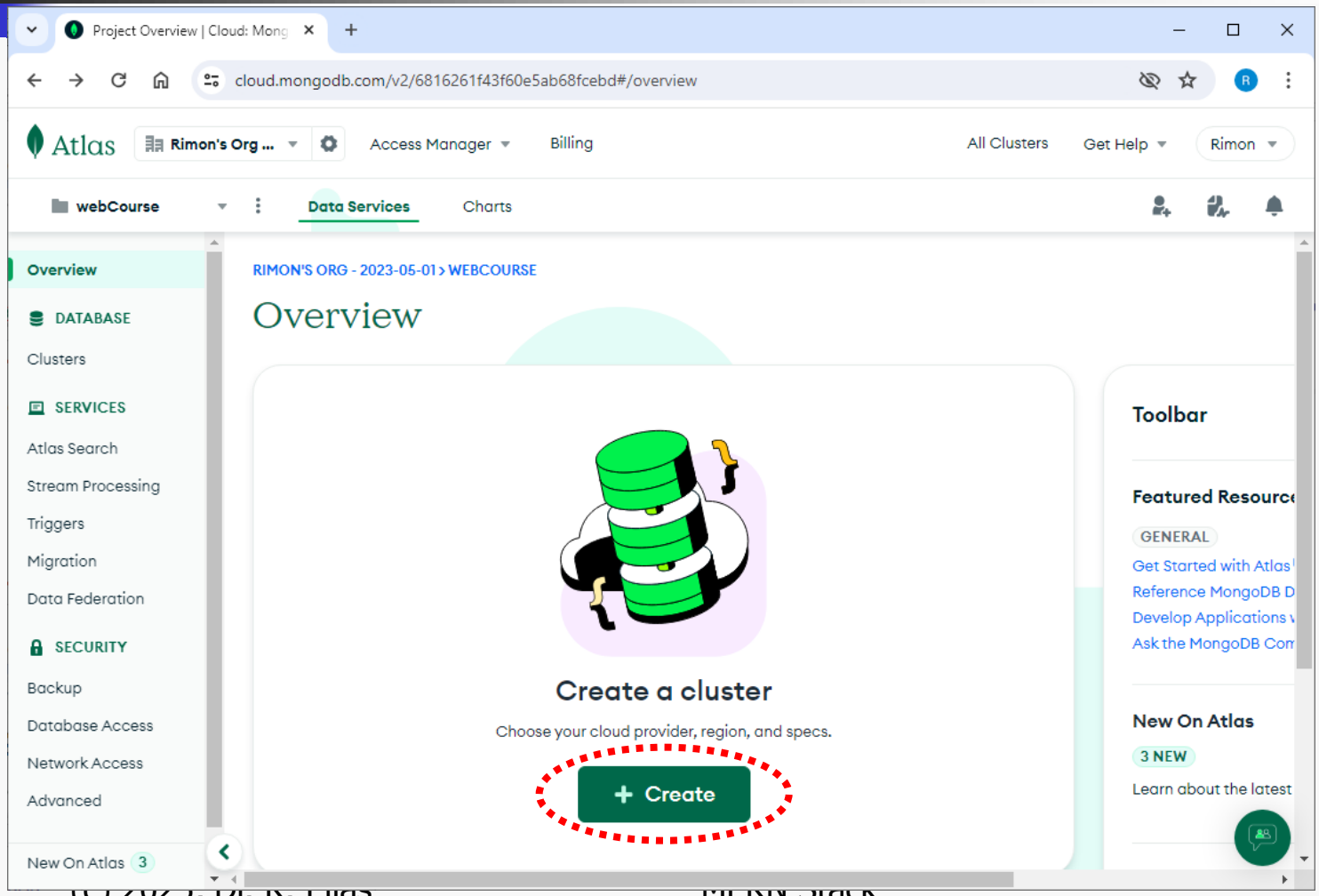
Atlas: Start with MongoDB



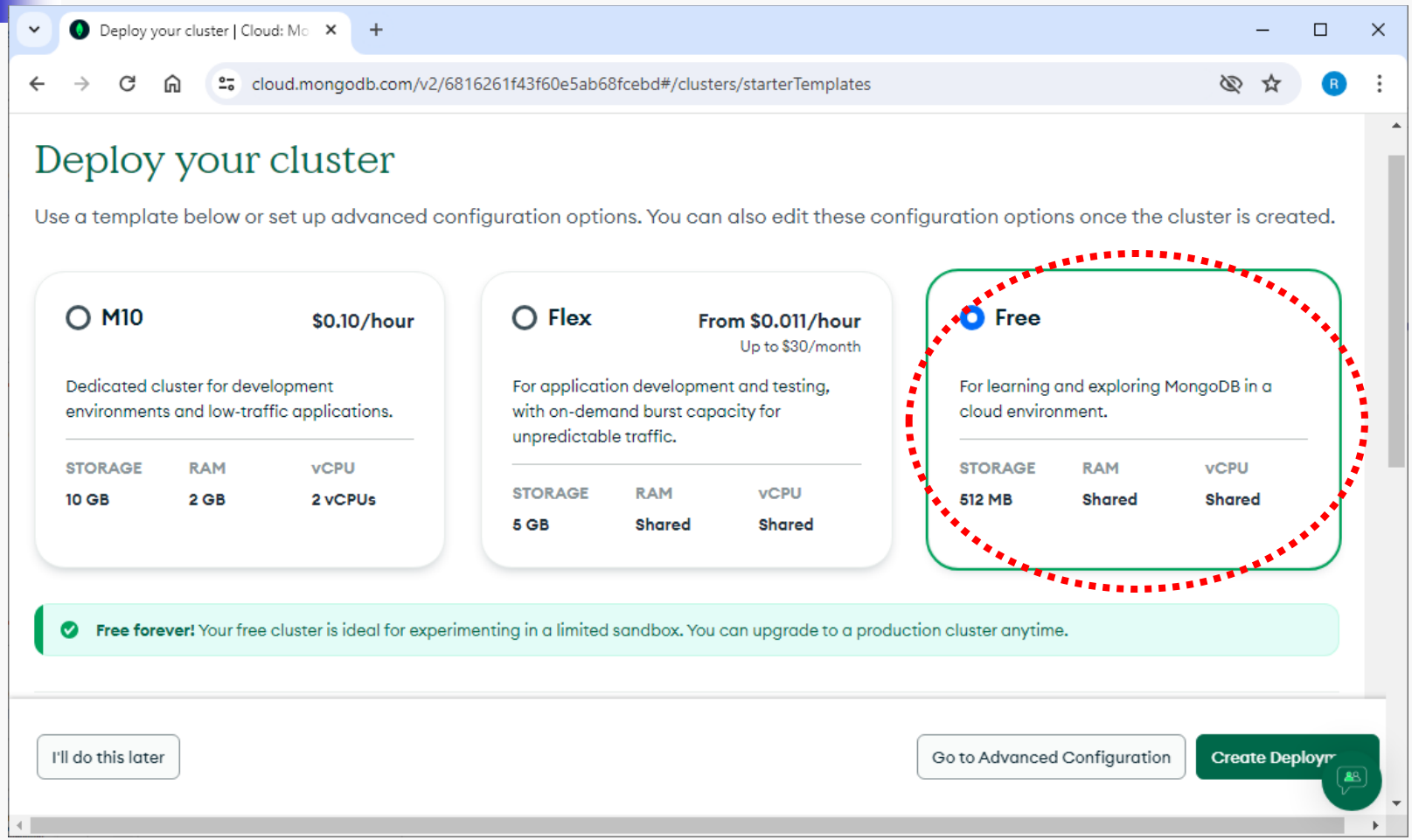
Atlas: Start with MongoDB



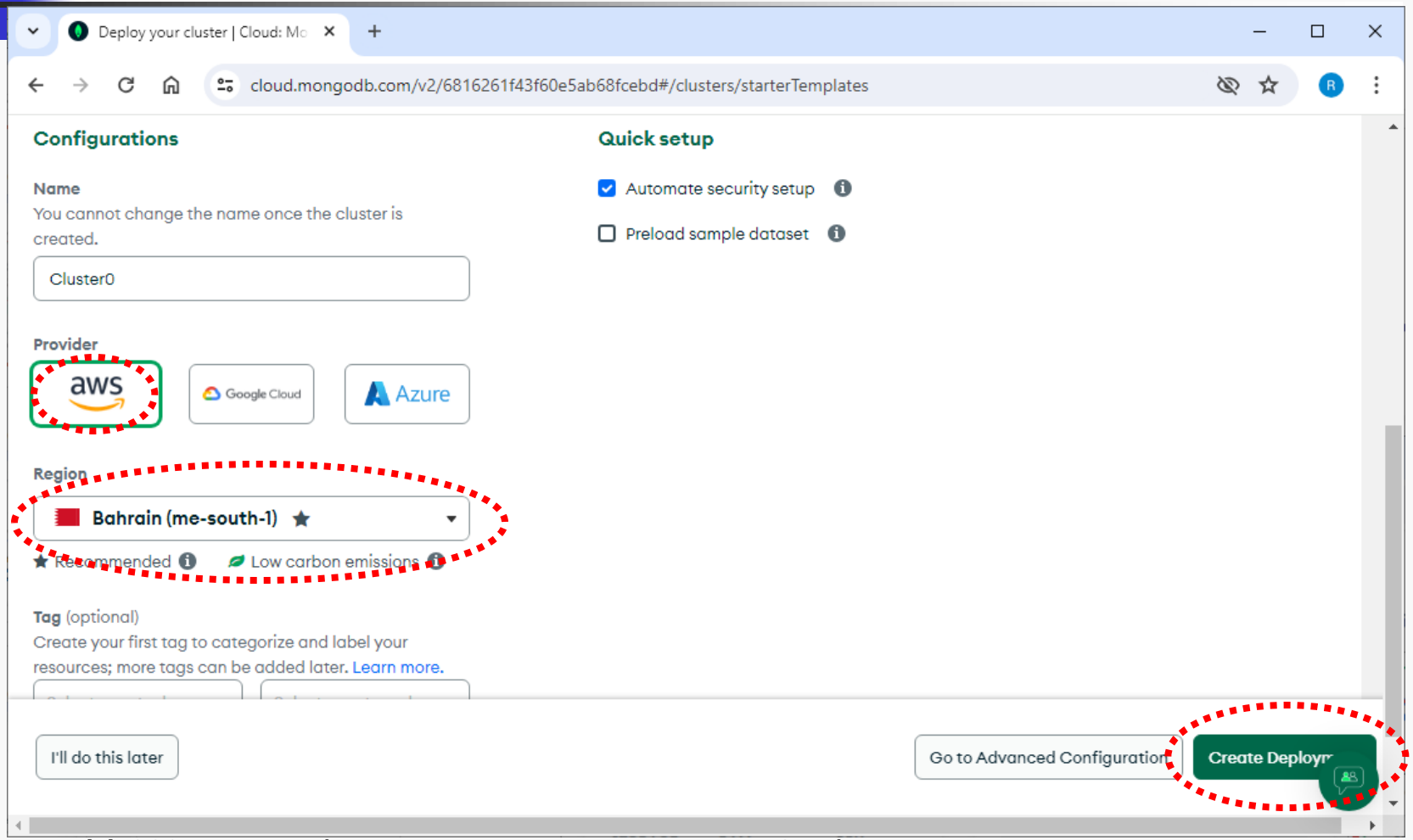
Atlas: Start with MongoDB



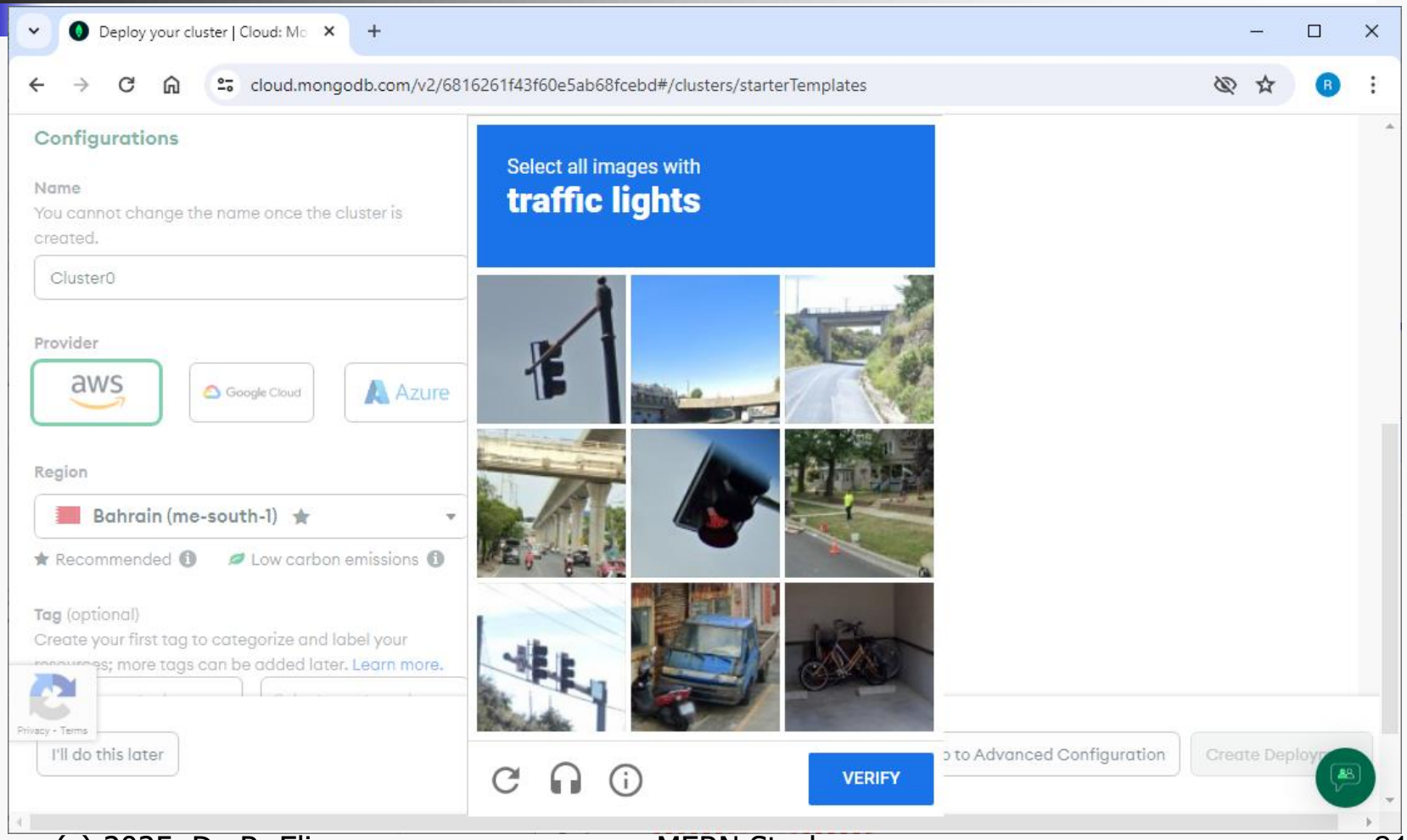
Atlas: Start with MongoDB



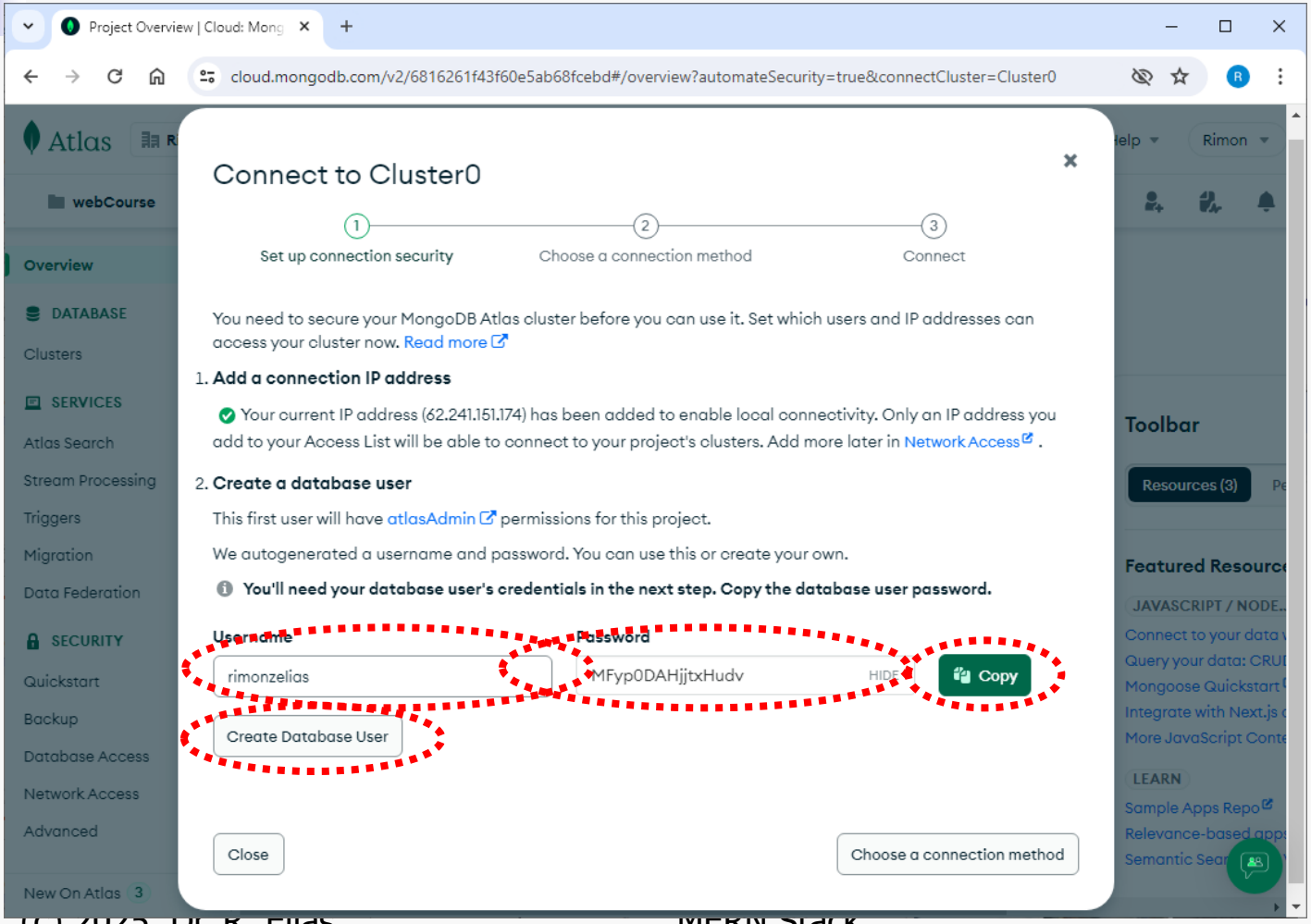
Atlas: Start with MongoDB



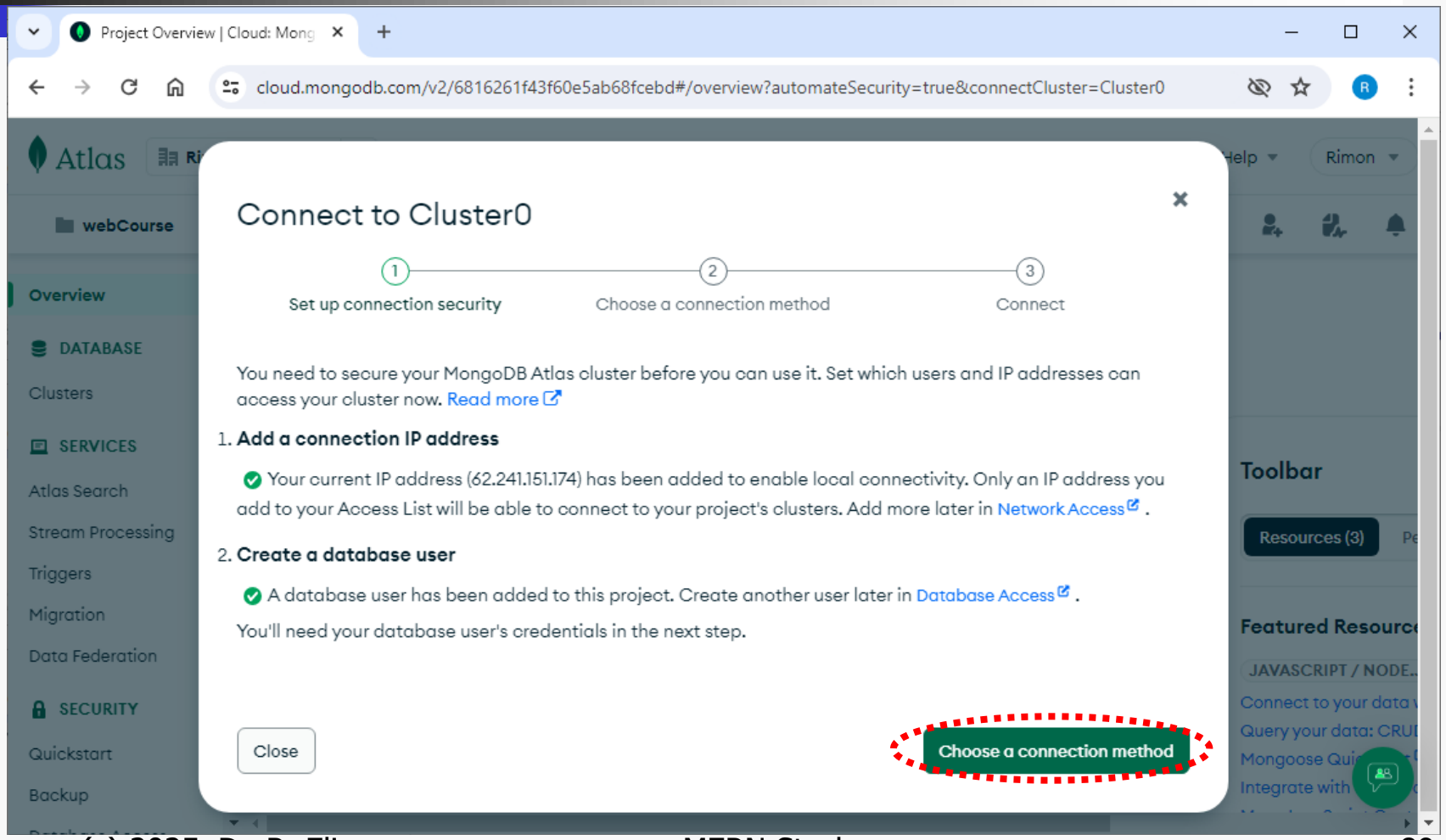
Atlas: Start with MongoDB



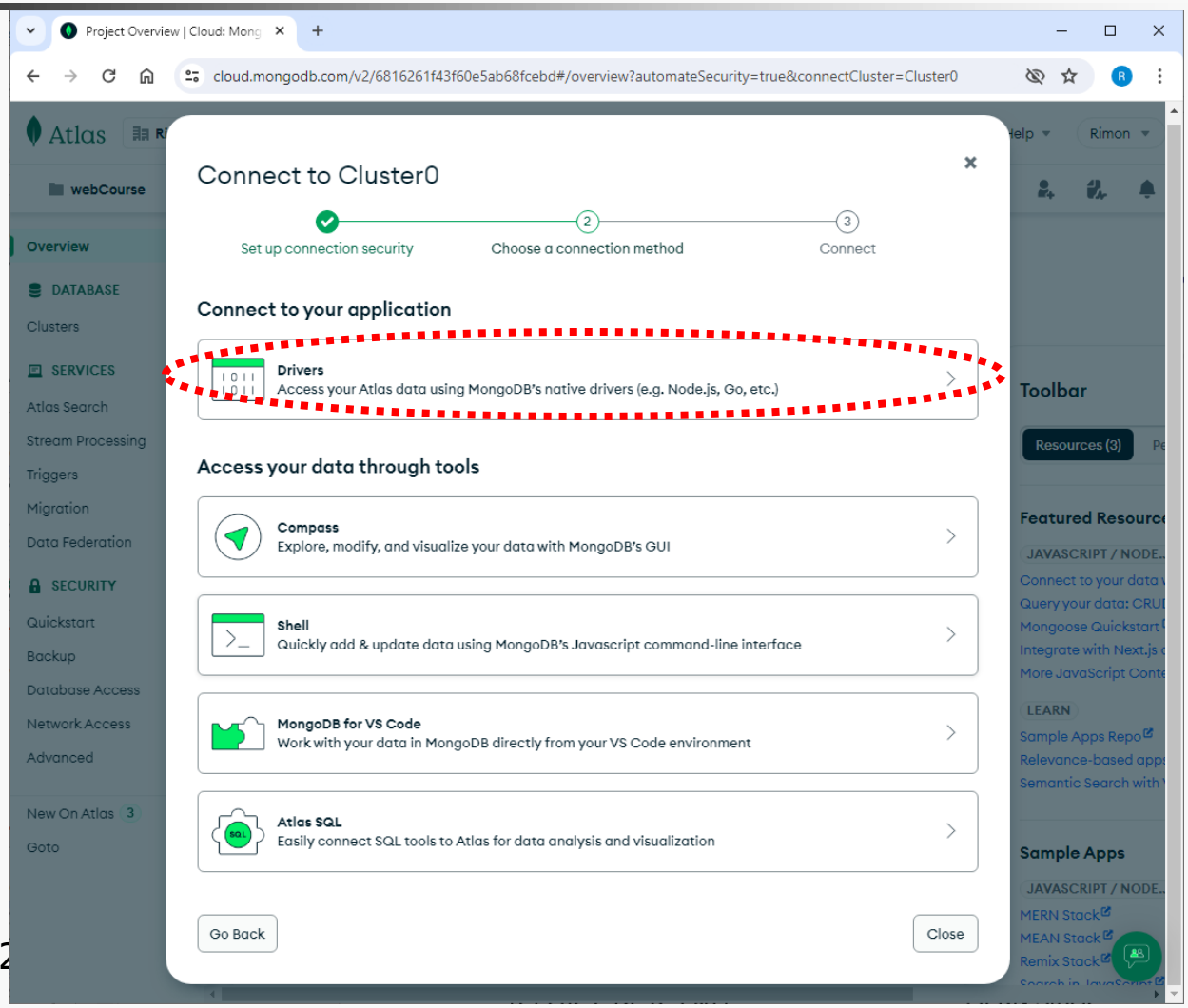
Atlas: Start with MongoDB



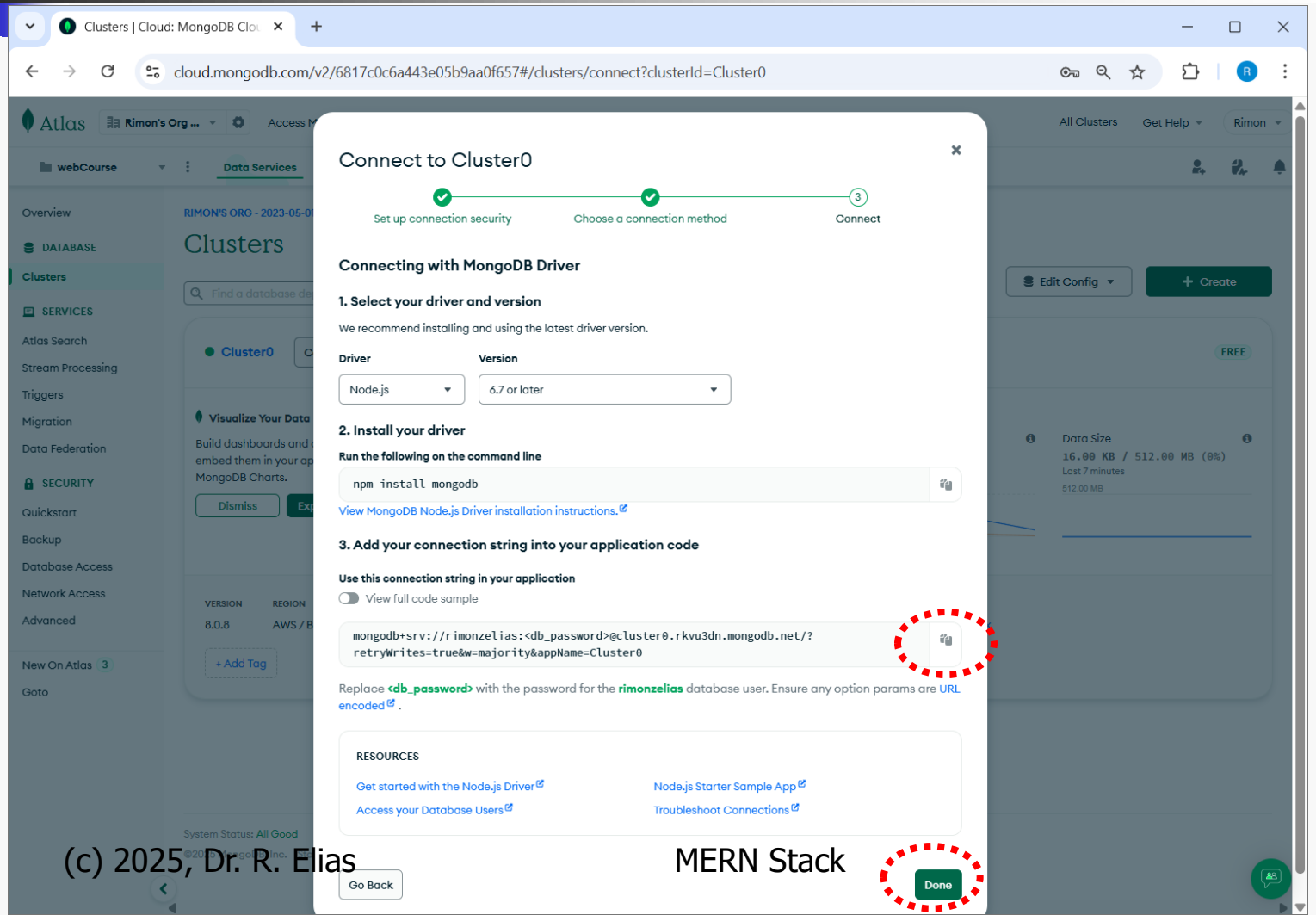
Atlas: Start with MongoDB



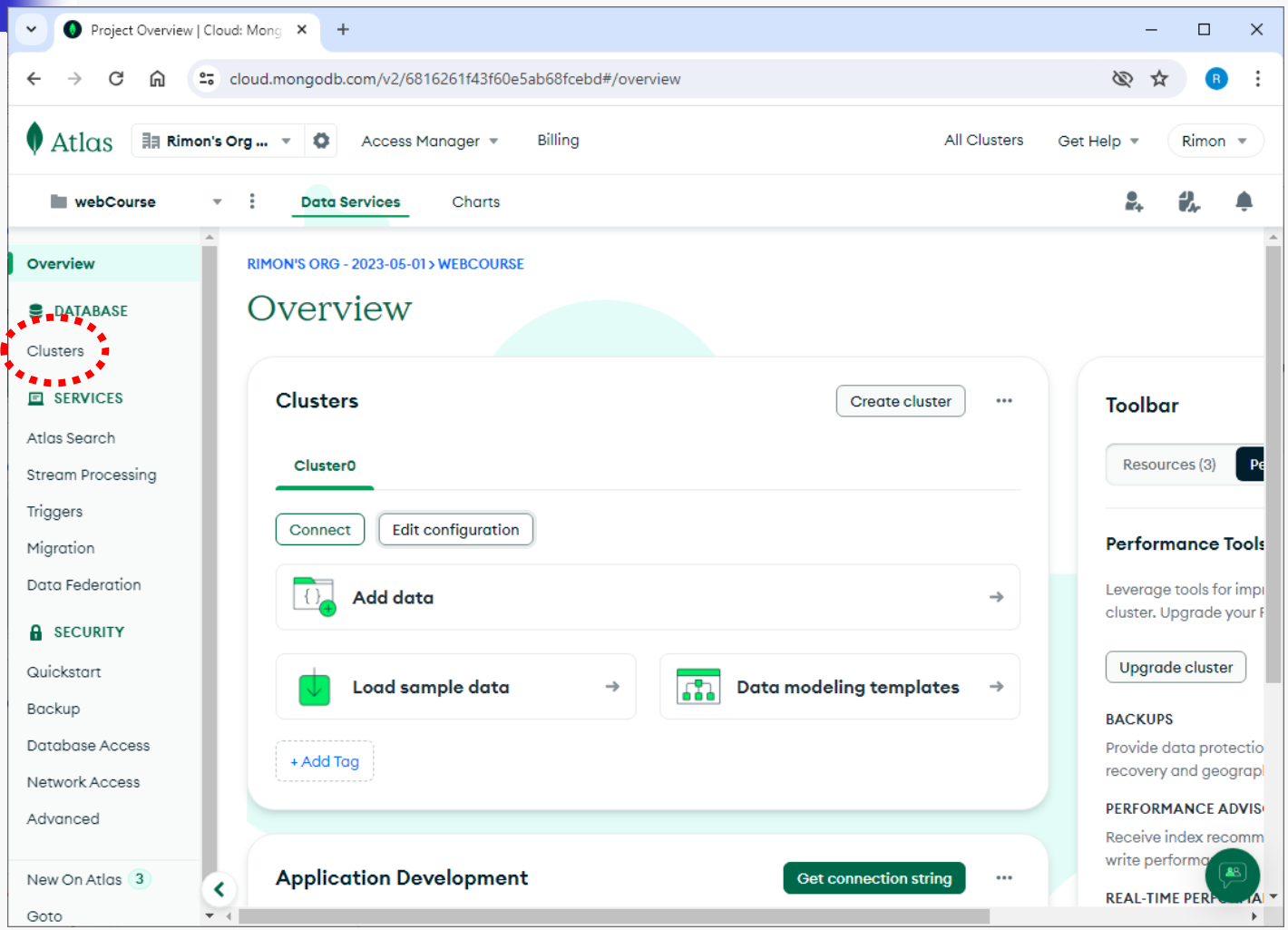
Atlas: Start with MongoDB



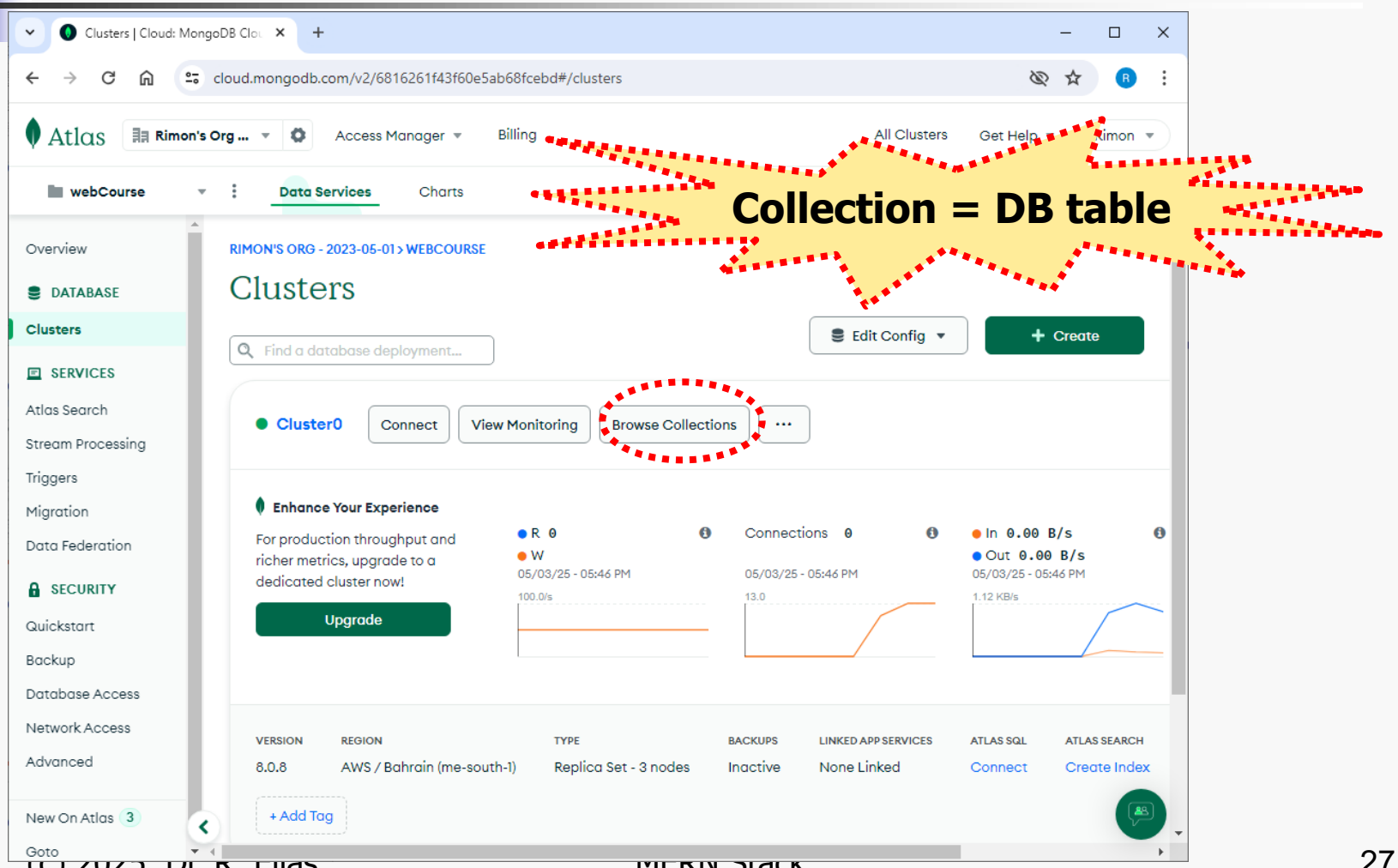
Atlas: Start with MongoDB



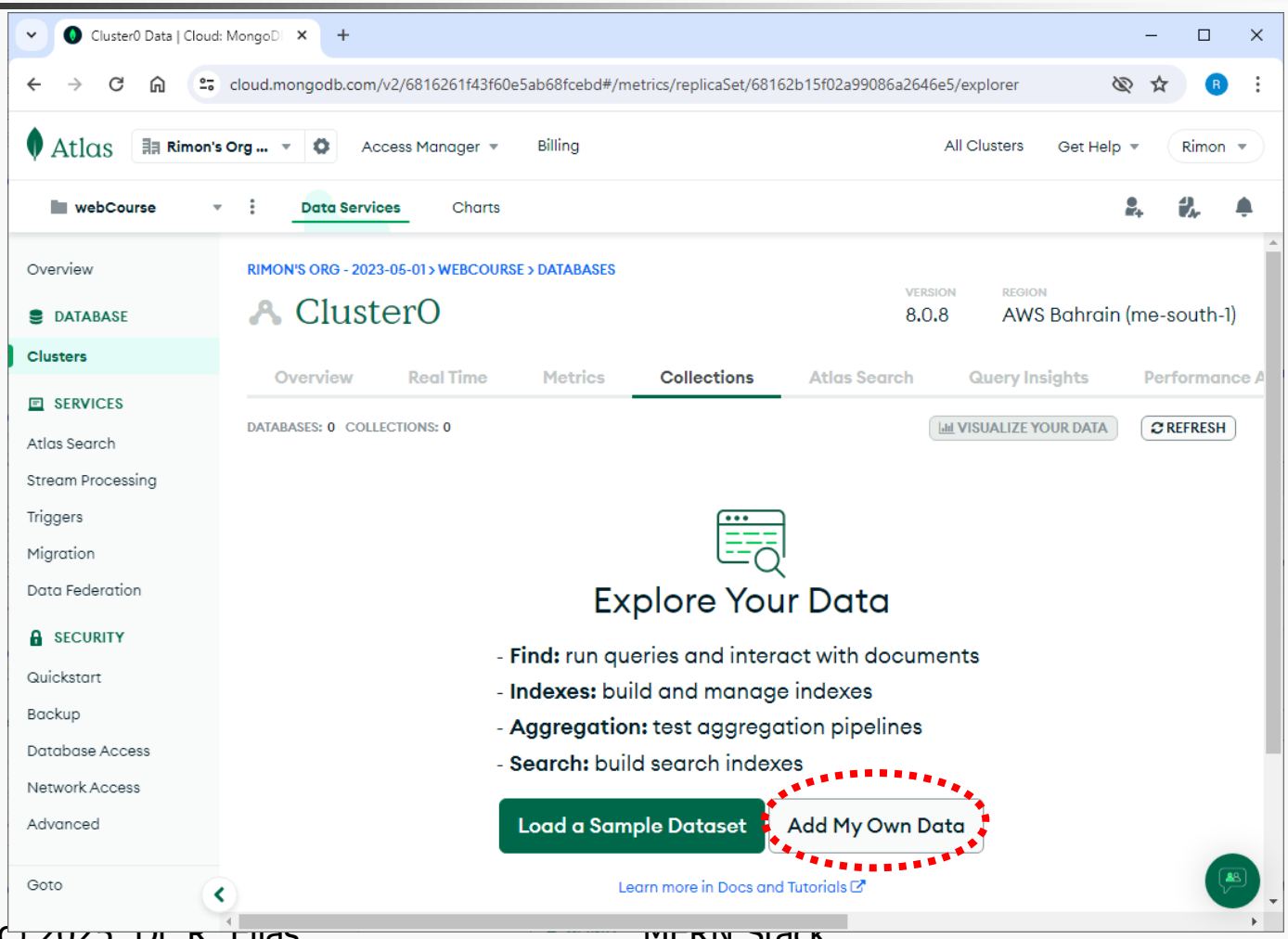
Atlas: Start with MongoDB



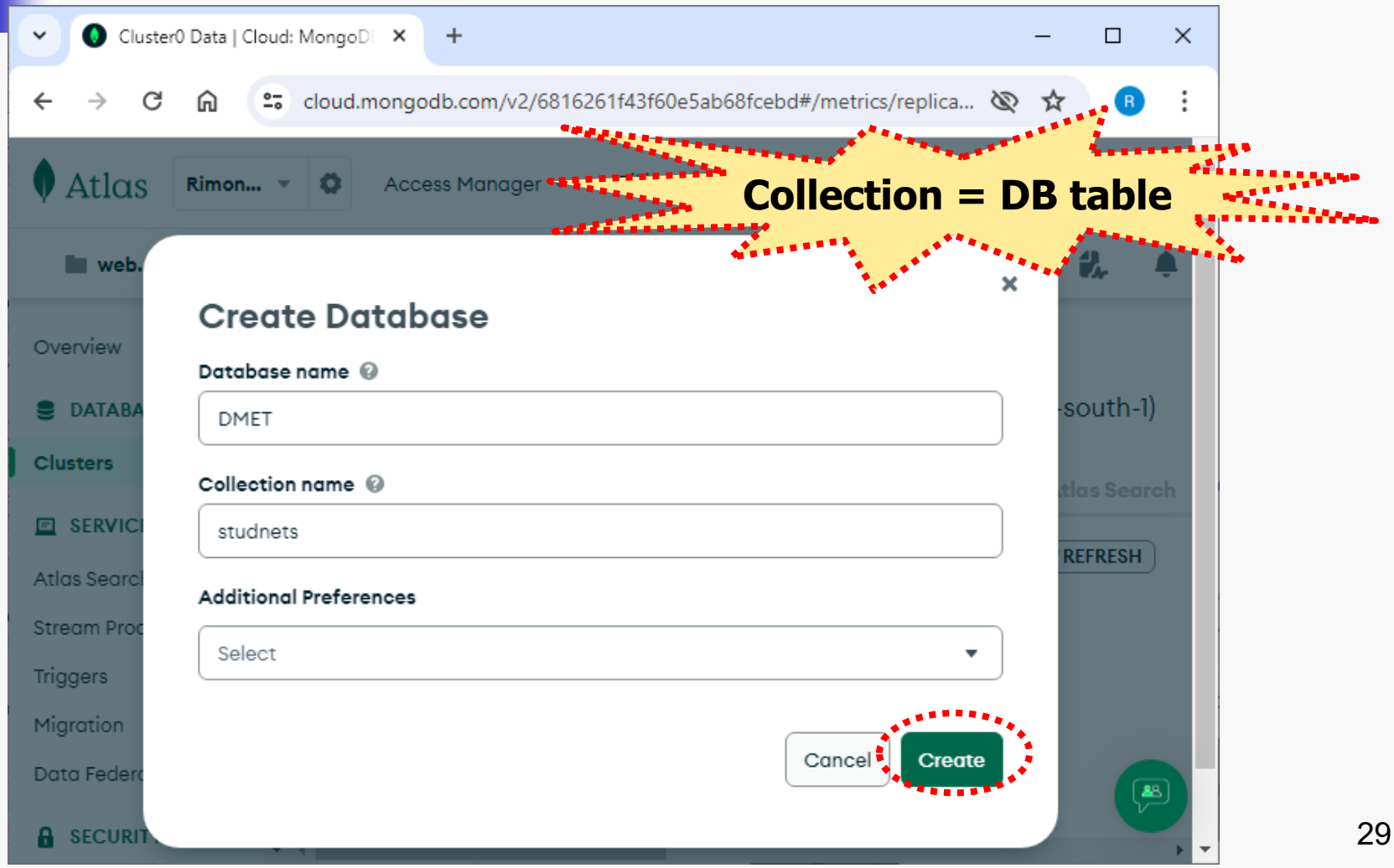
Atlas: Start with MongoDB



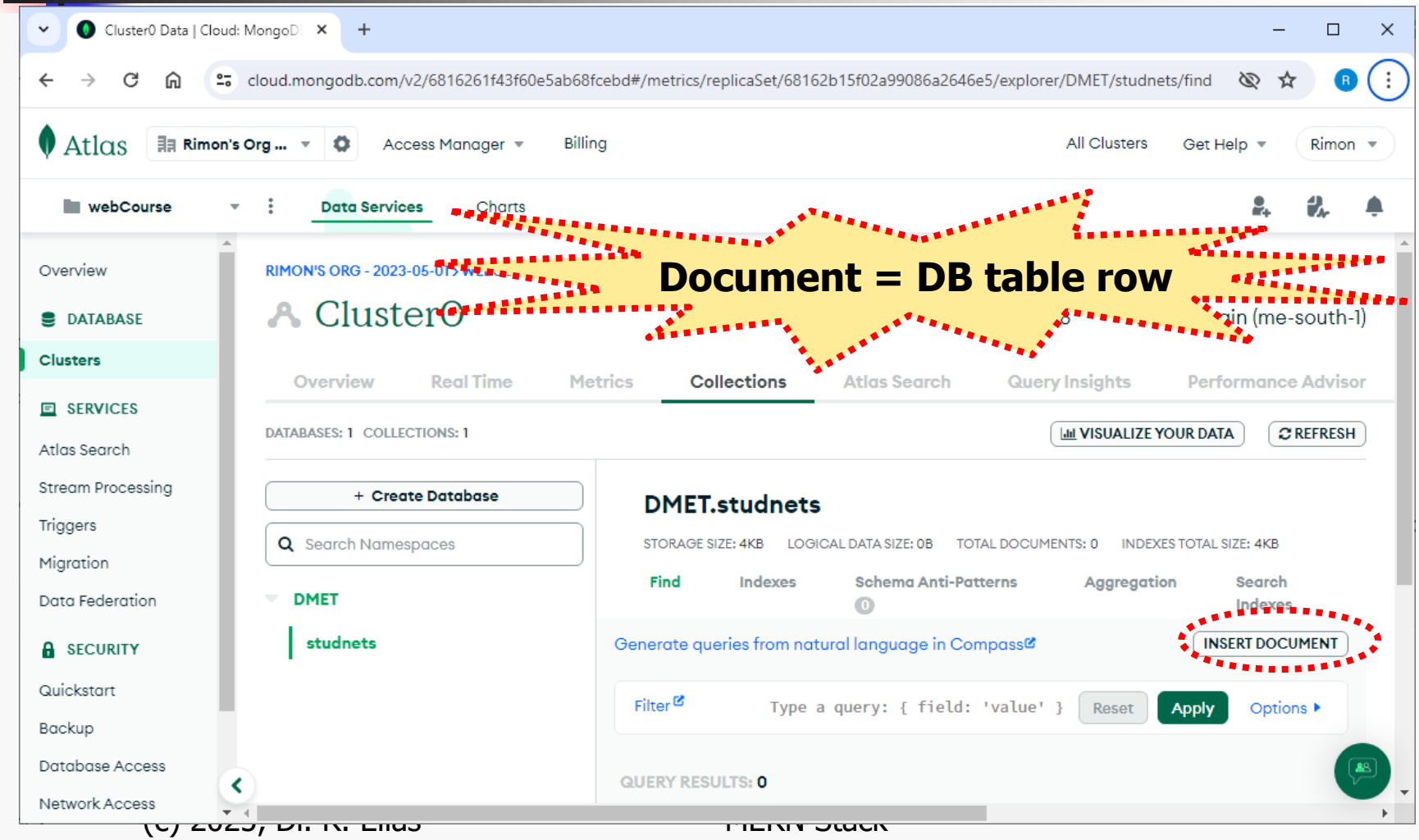
Atlas: Start with MongoDB



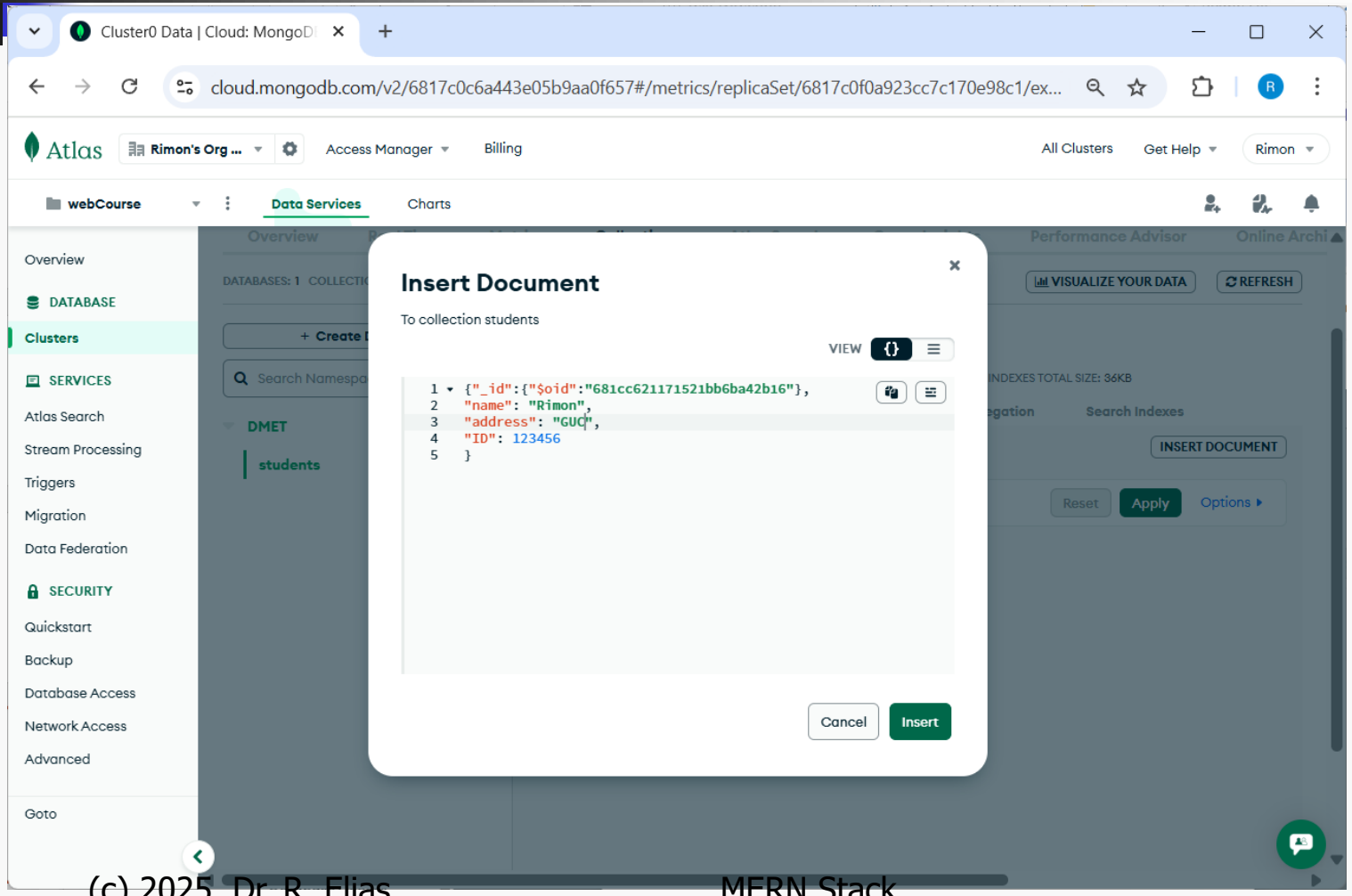
Atlas: Start with MongoDB



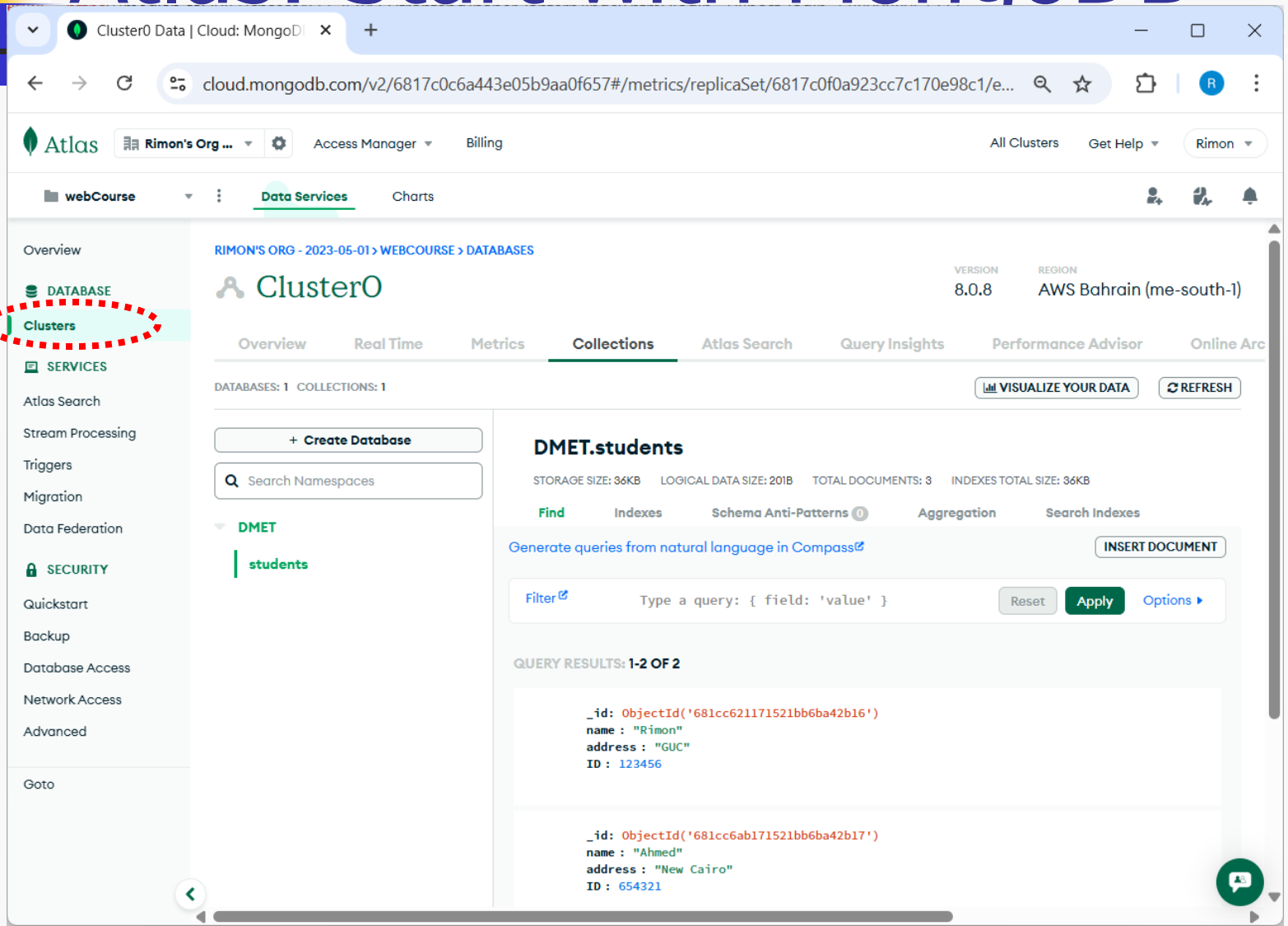
Atlas: Start with MongoDB



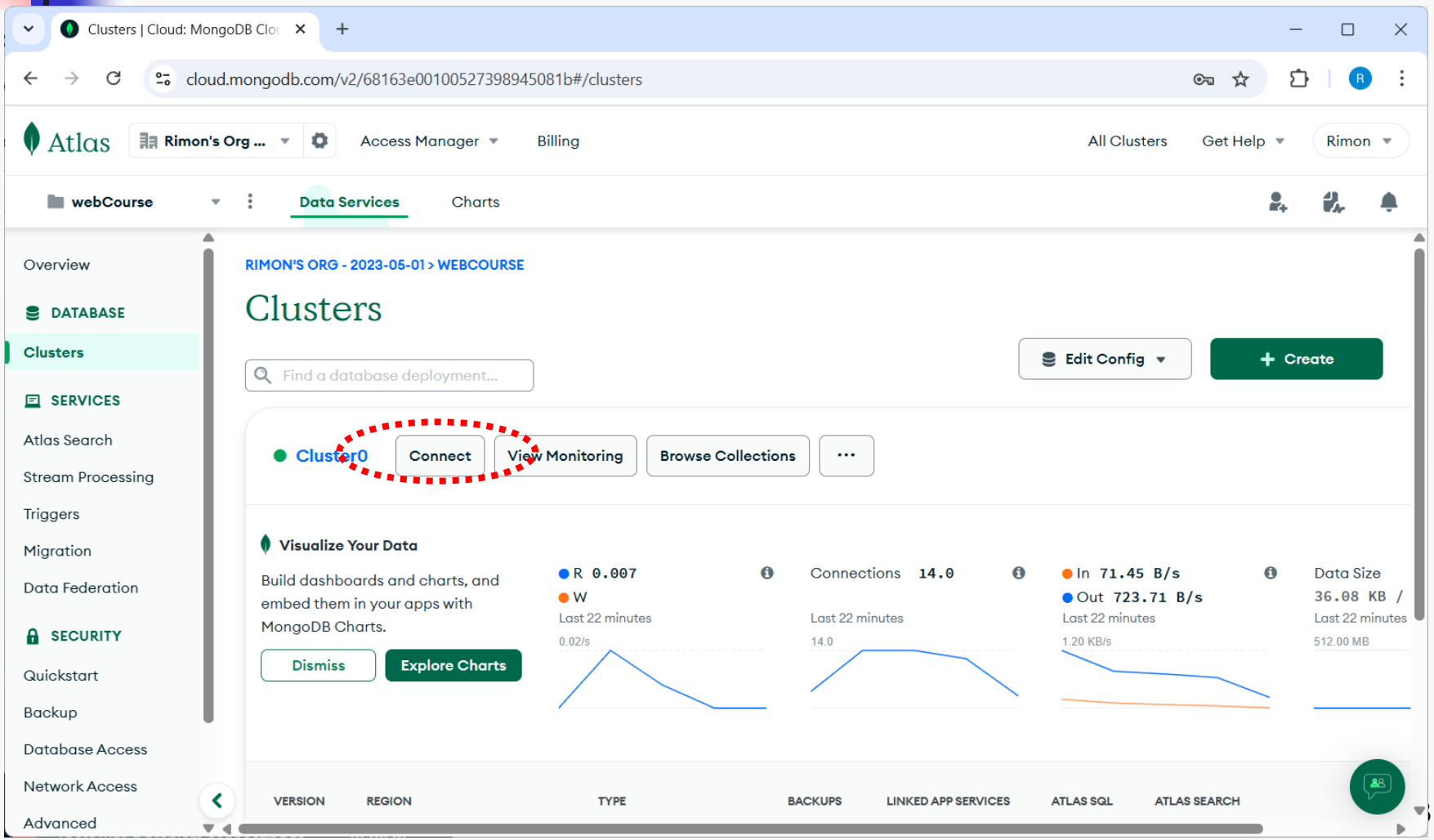
Atlas: Start with MongoDB



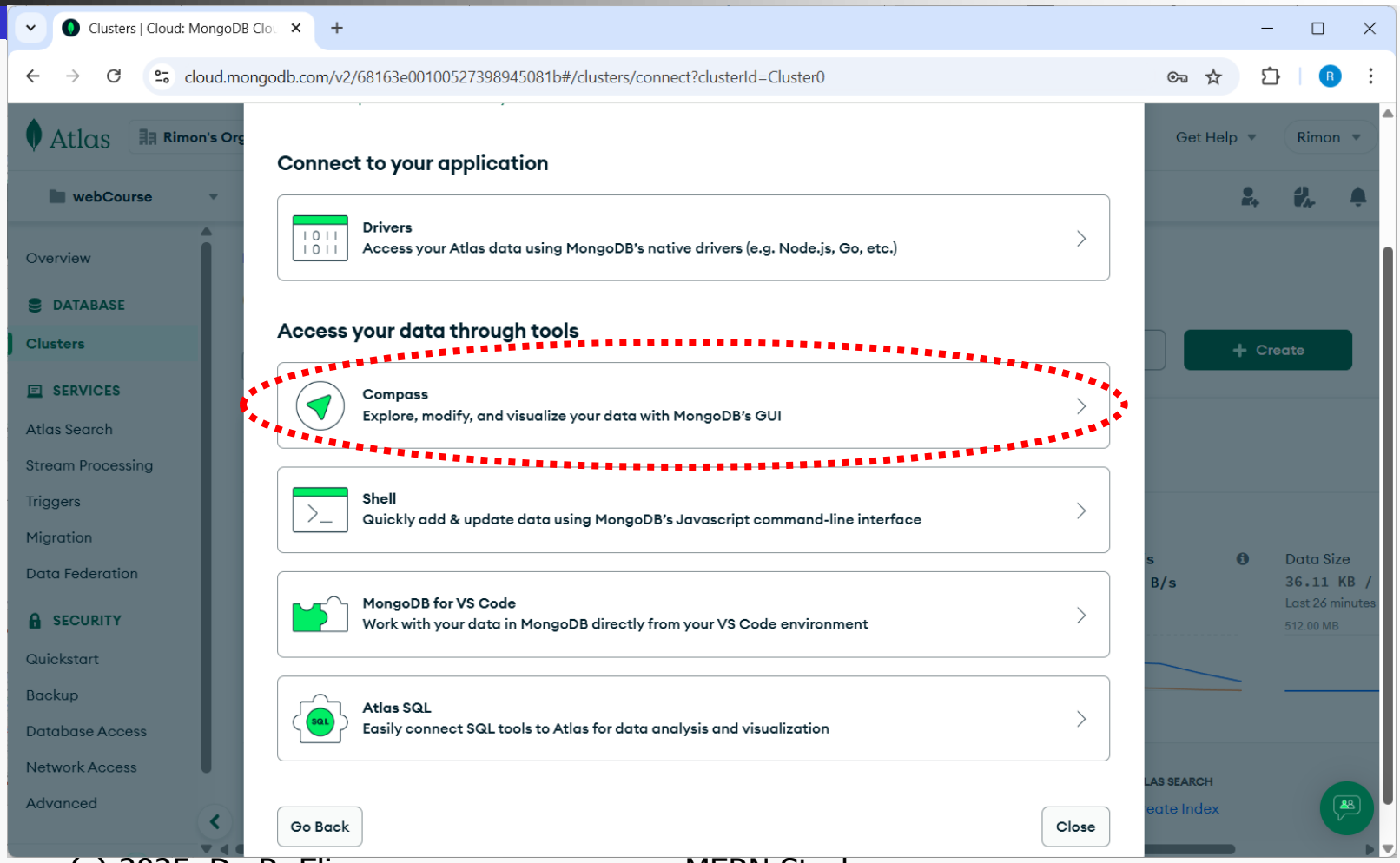
Atlas: Start with MongoDB



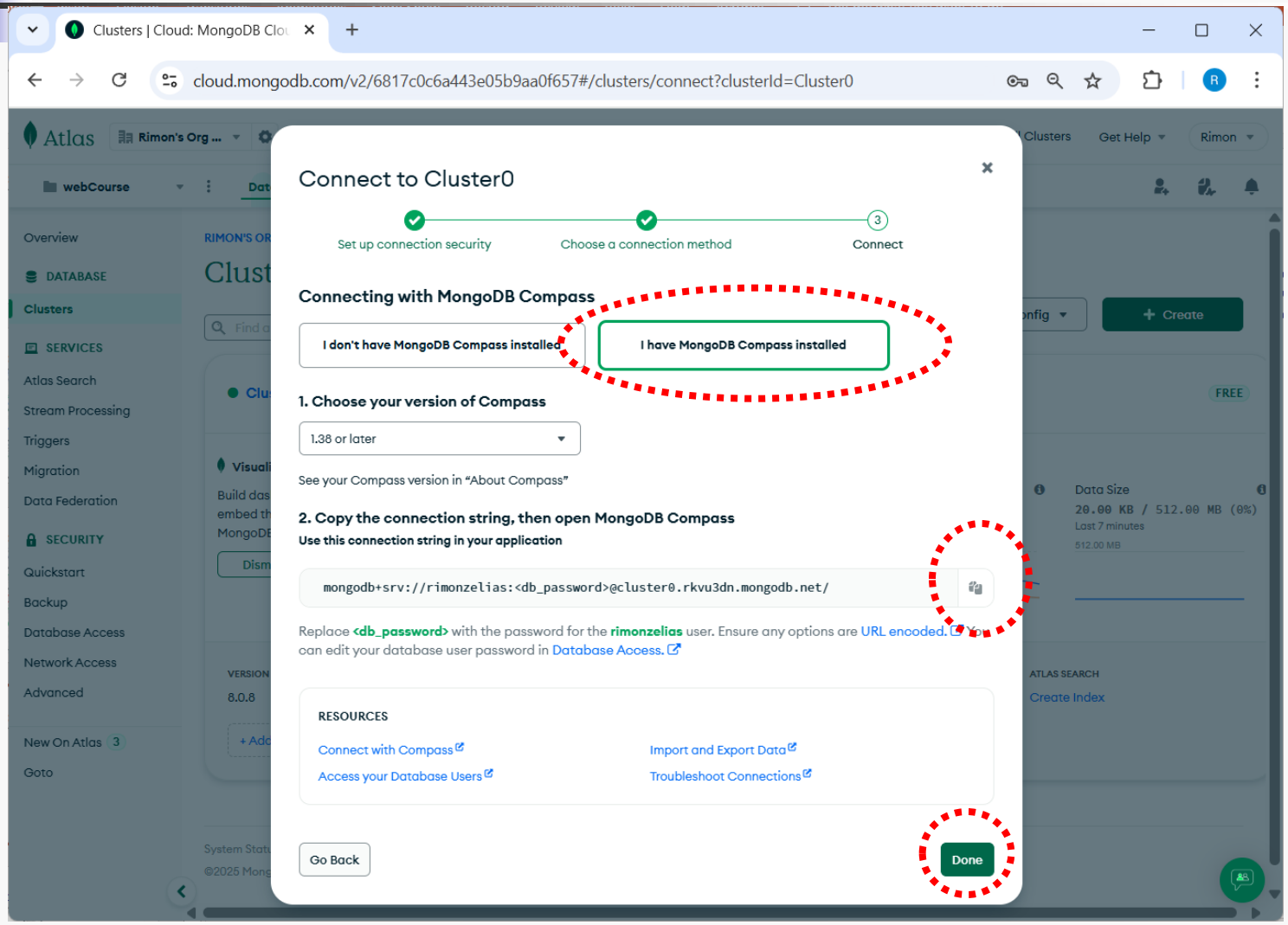
Atlas: Start with MongoDB



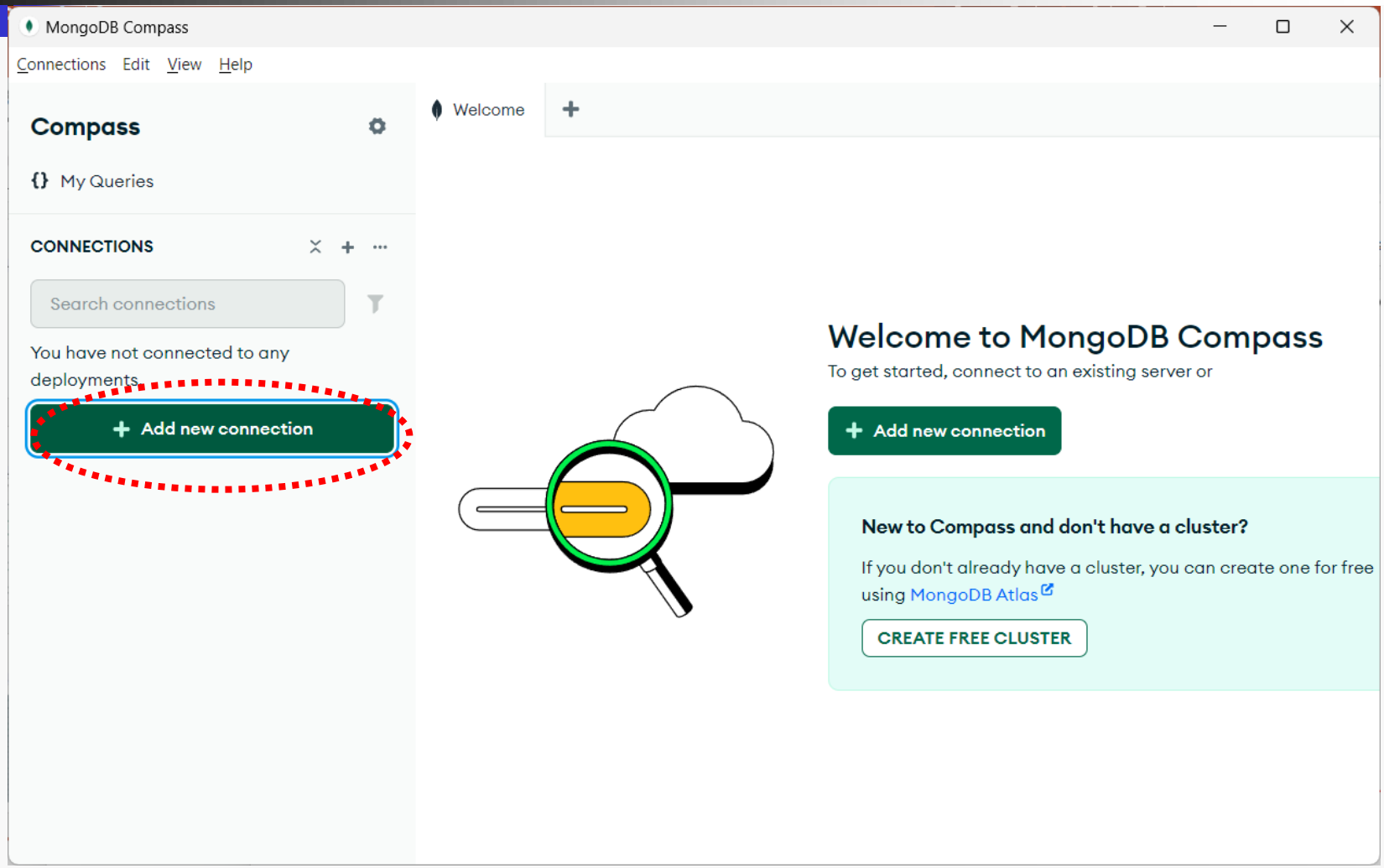
Atlas: Start with MongoDB



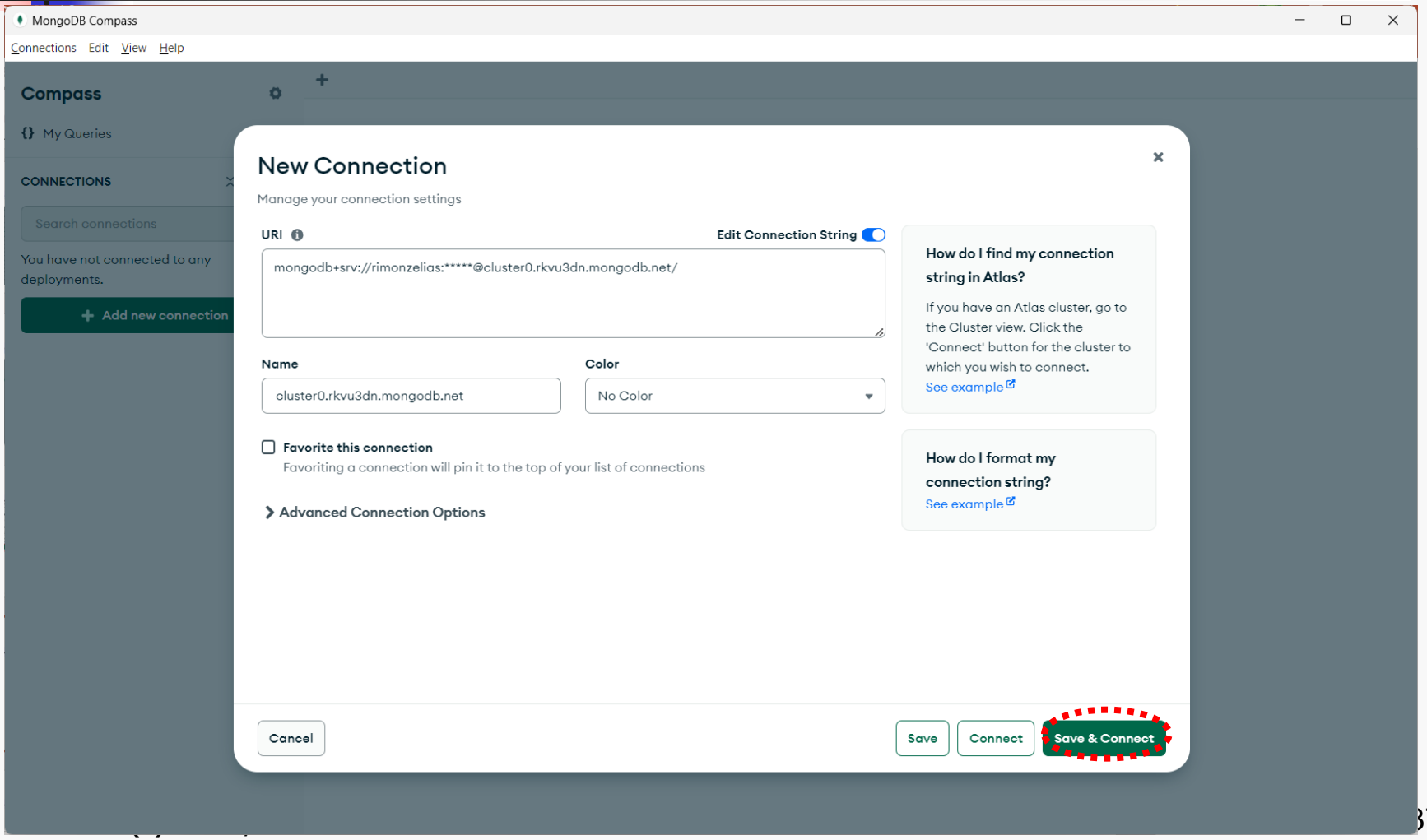
Atlas: Start with MongoDB



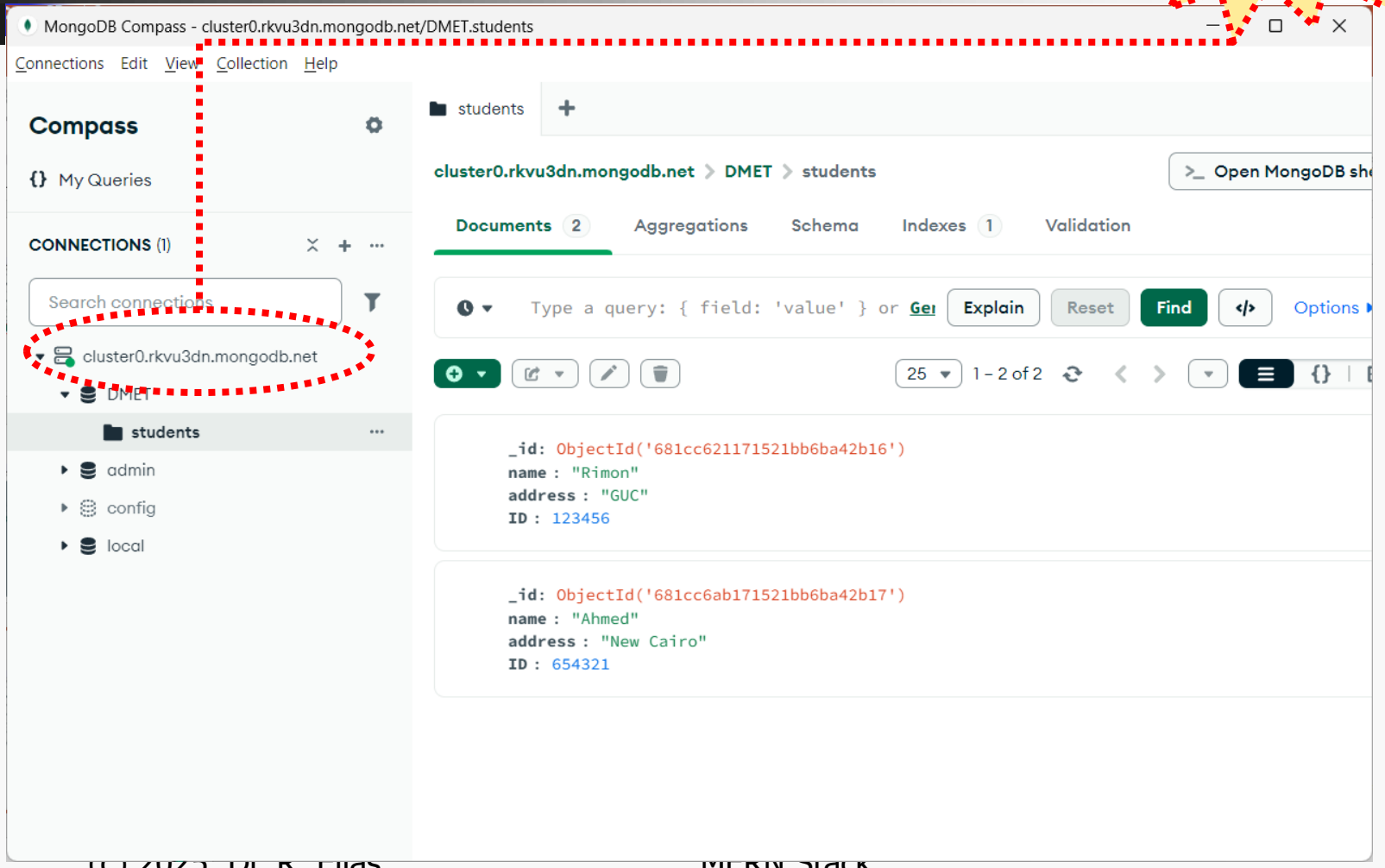
MongoDB Compass



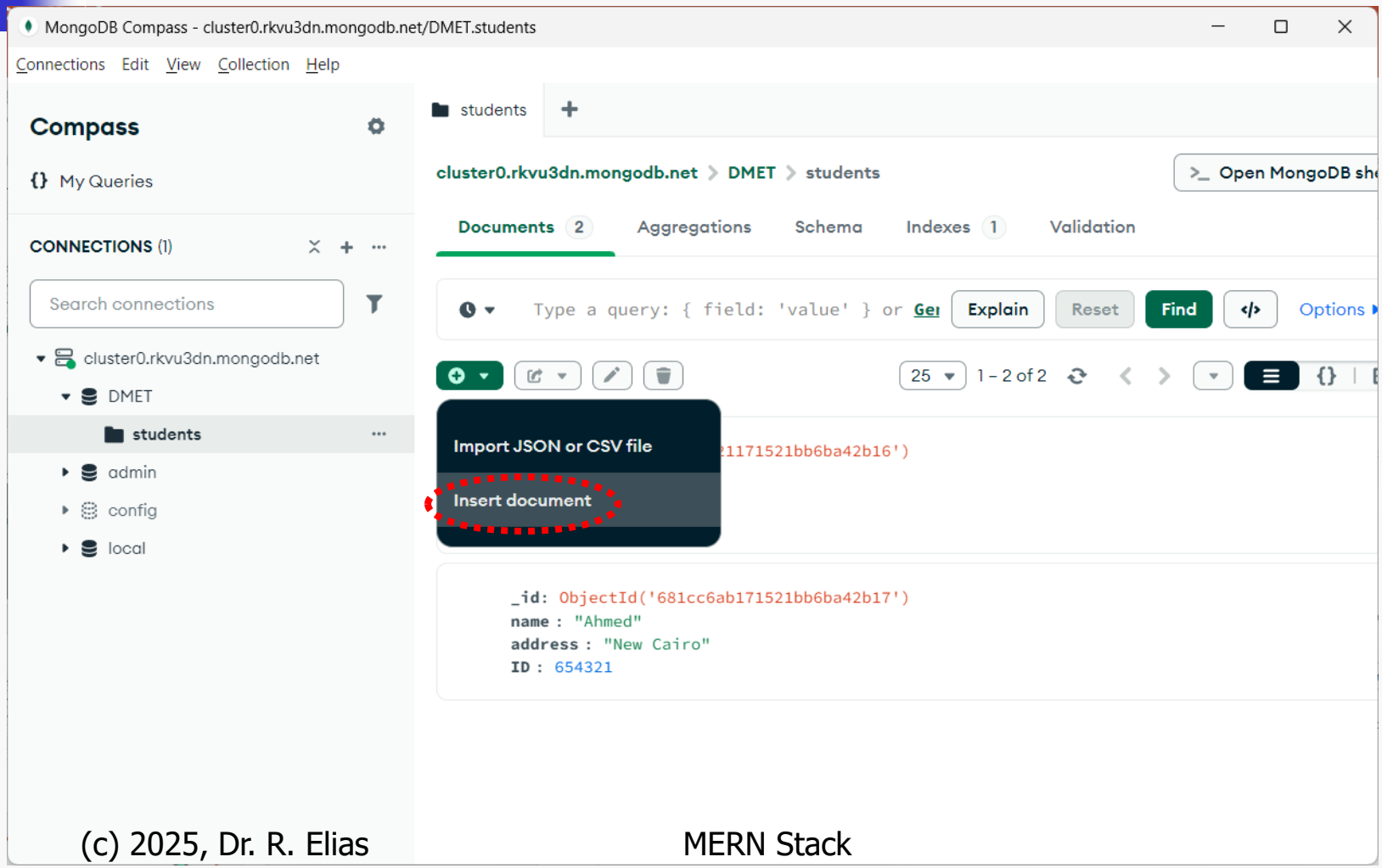
MongoDB Compass



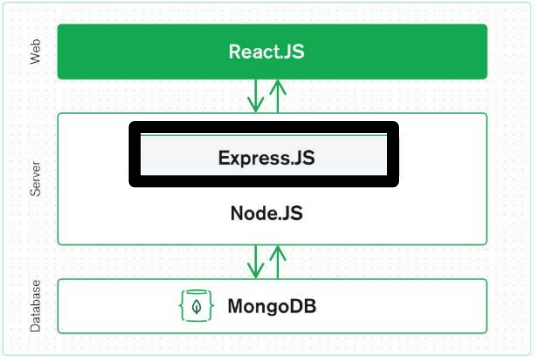
MongoDB Compass



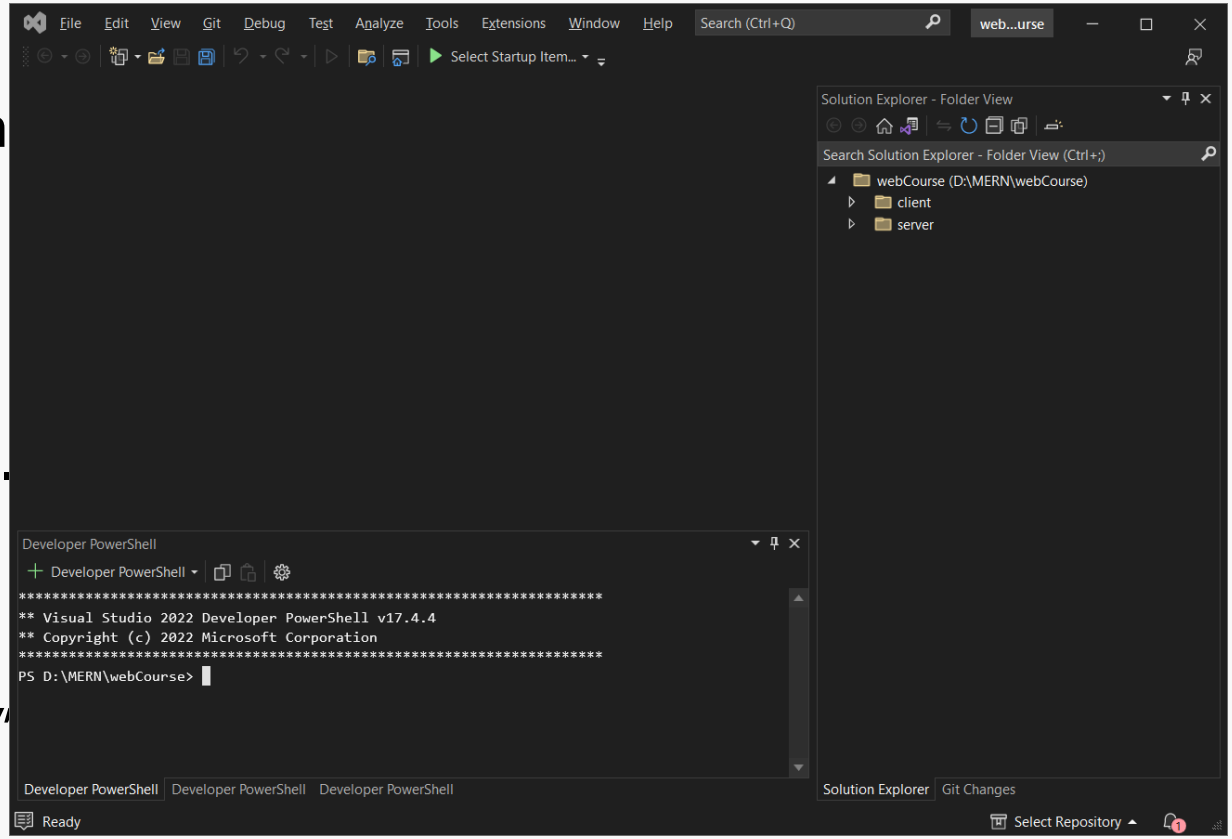
MongoDB Compass



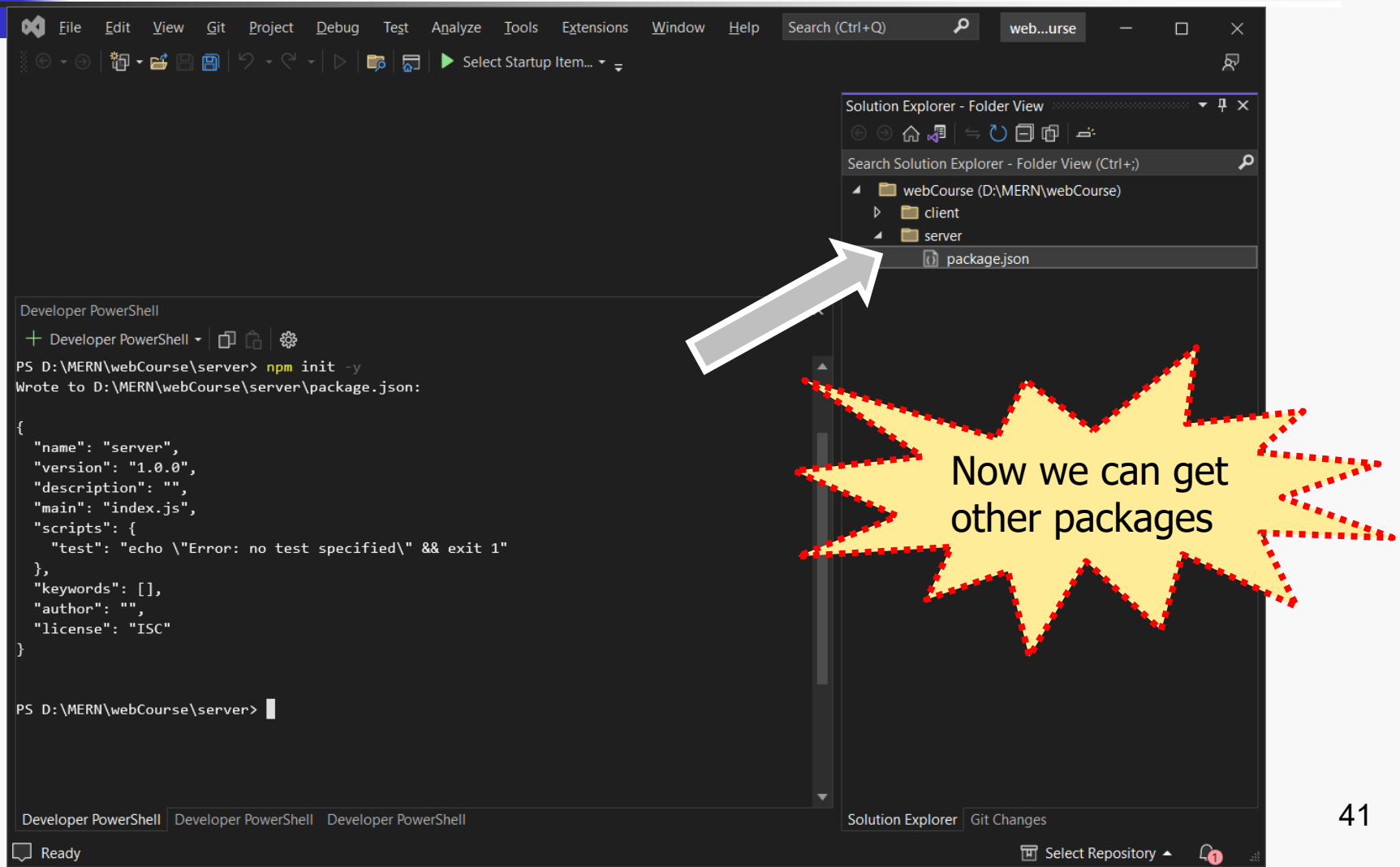
Express



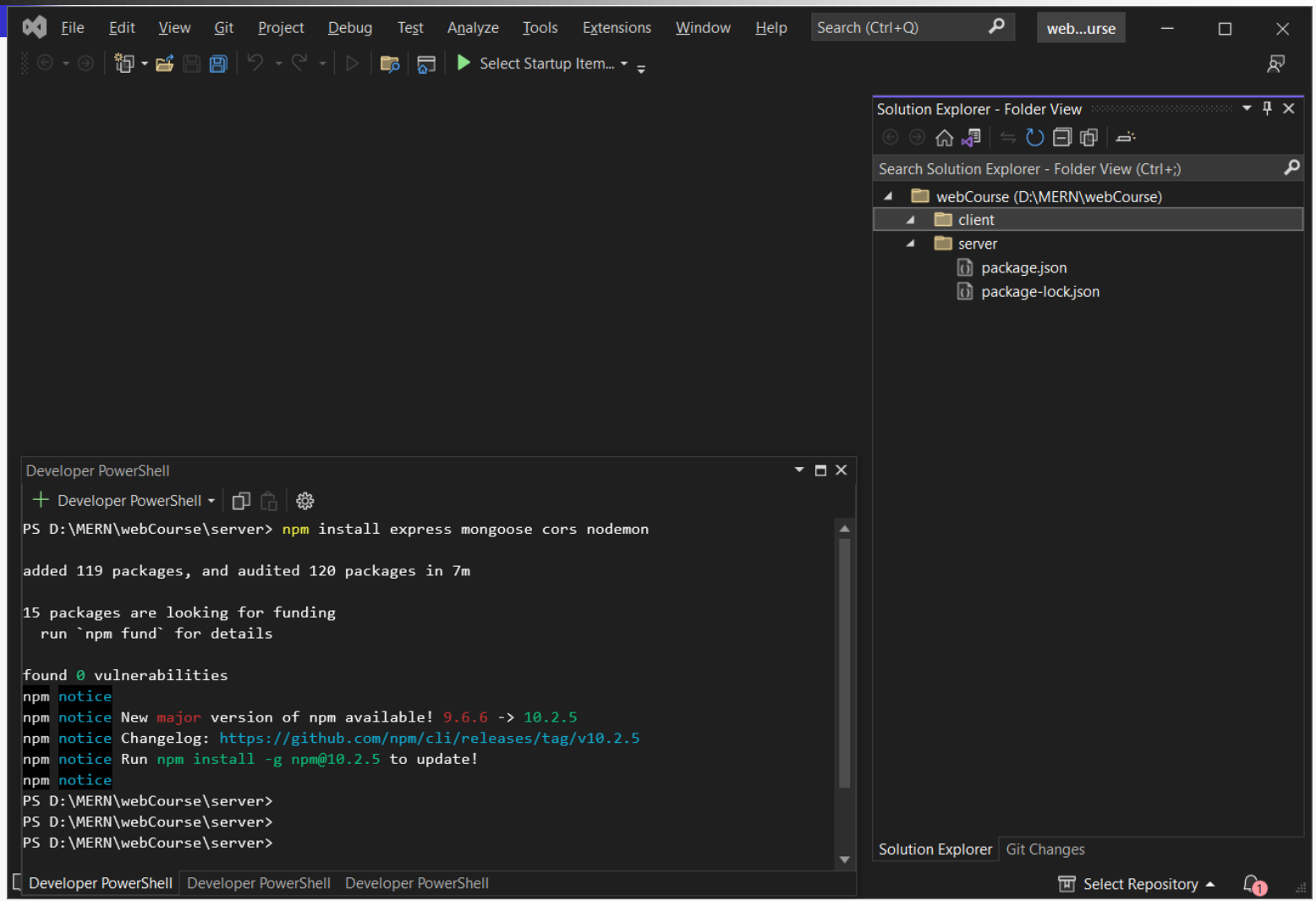
- Under your project folder (i.e., webCourse), create two folders; server and client.
- We will use the folder **server** with **Express** (back end) and
- We will use the folder **client** with **React** (front end).
- Right-click on the webCourse folder and choose "Open with Visual Studio"



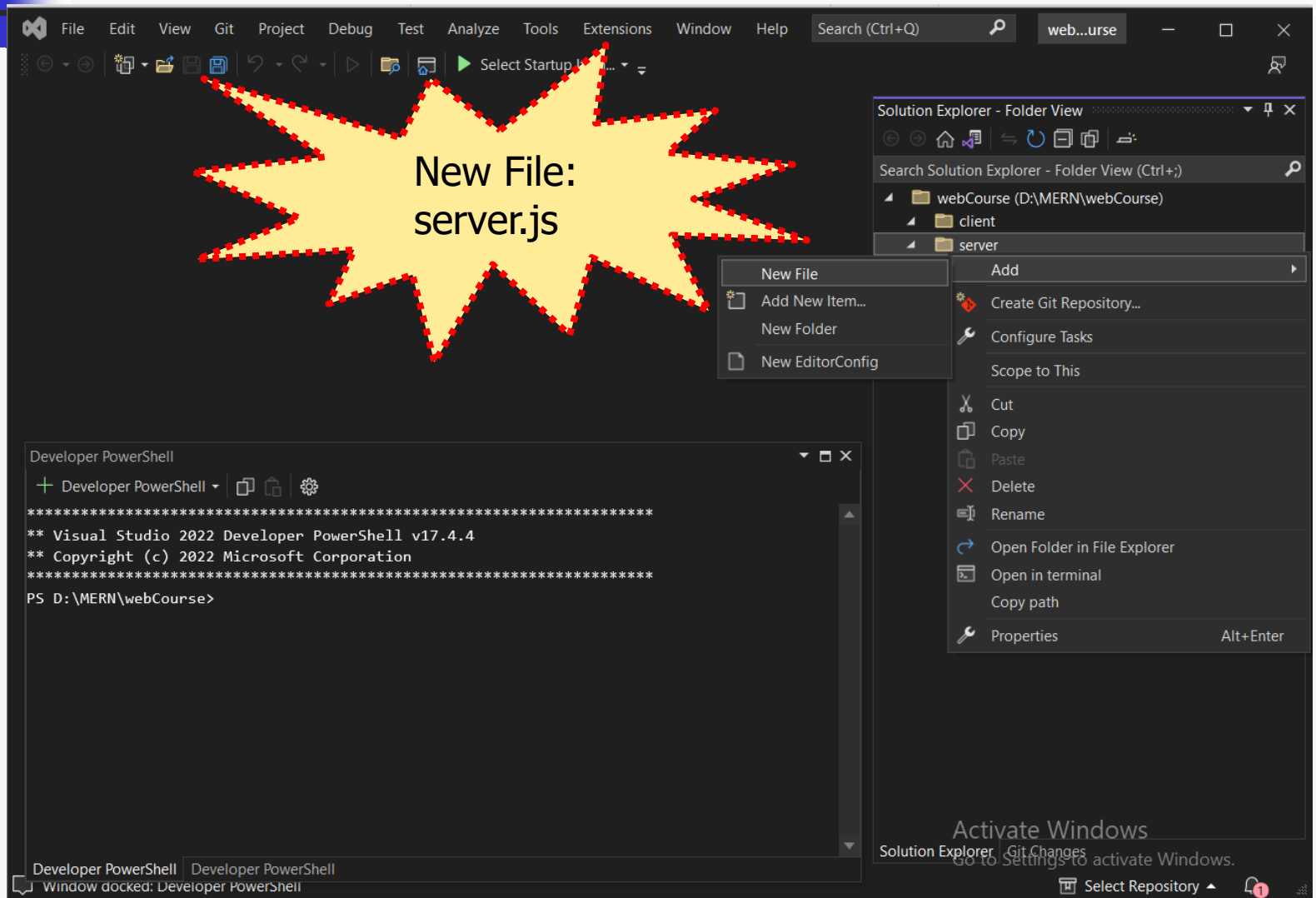
Express


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
web...urse
Select Startup Item...
Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)
webCourse (D:\MERN\webCourse)
client
server
package.json
Developer PowerShell
Developer PowerShell
PS D:\MERN\webCourse\server> npm init -y
Wrote to D:\MERN\webCourse\server\package.json:
{
"name": "server",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}
PS D:\MERN\webCourse\server>
Now we can get other packages
Developer PowerShell
Developer PowerShell
Developer PowerShell
Solution Explorer
Git Changes
Ready
Select Repository

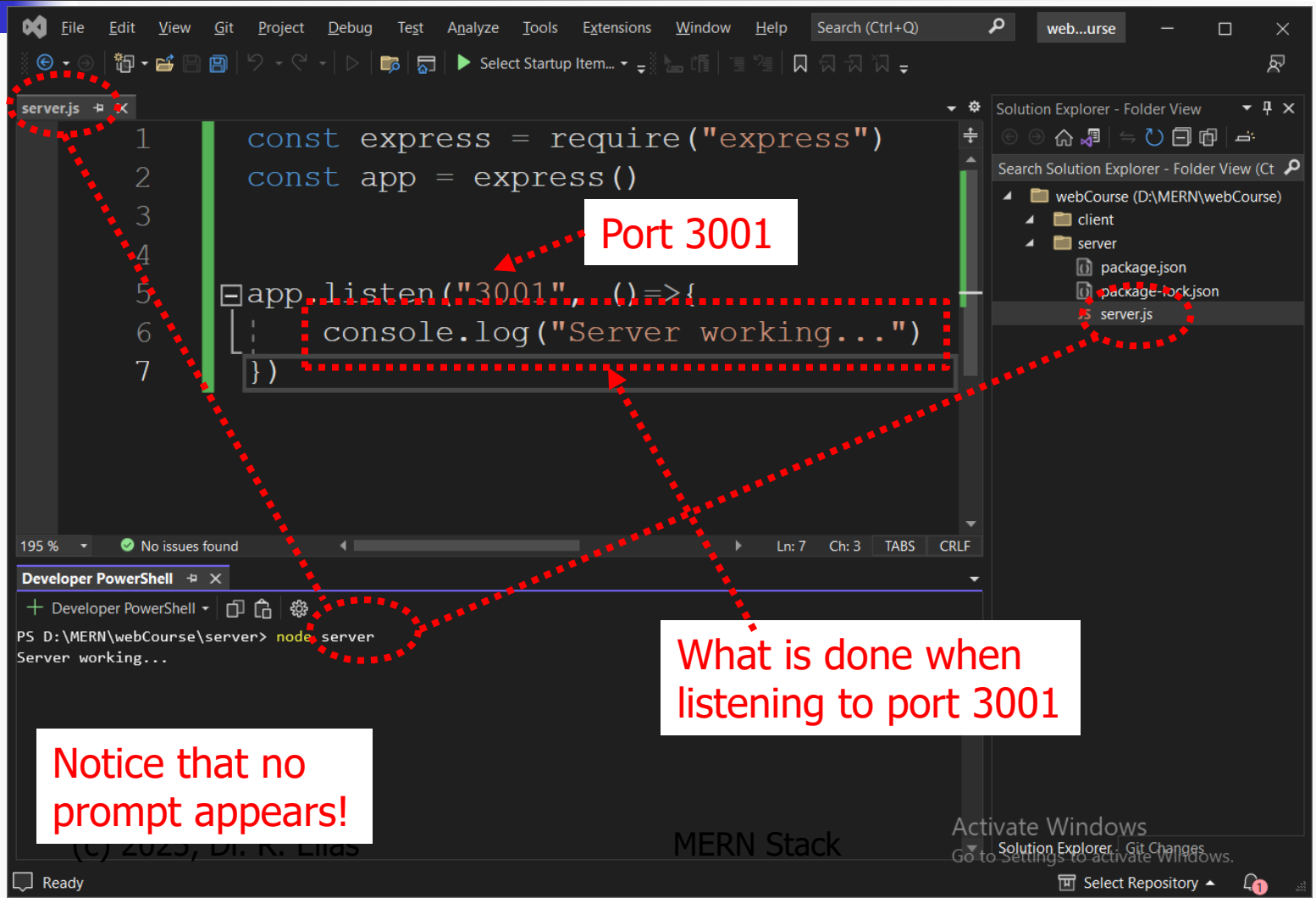
Express and Other Packages



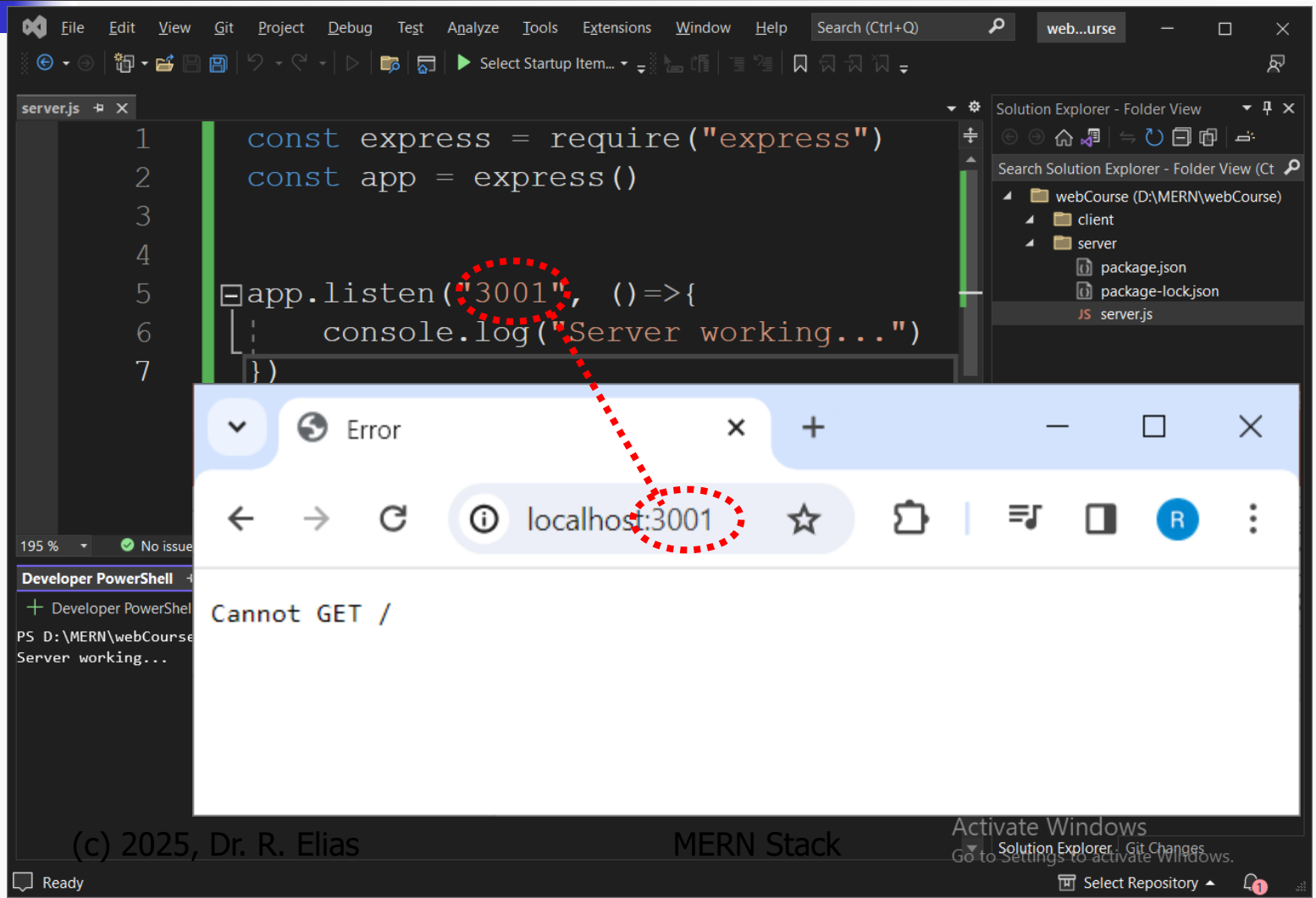
server.js


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
web...urse
Select Startup
New File:
server.js
Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)
webCourse (D:\MERN\webCourse)
client
server
New File
Add New Item...
New Folder
New EditorConfig
Add
Create Git Repository...
Configure Tasks
Scope to This
Cut
Copy
Paste
Delete
Rename
Open Folder in File Explorer
Open in terminal
Copy path
Properties
Alt+Enter
Developer PowerShell
Developer PowerShell
** Visual Studio 2022 Developer PowerShell v17.4.4
** Copyright (c) 2022 Microsoft Corporation
PS D:\MERN\webCourse>
Developer PowerShell
Developer PowerShell
Solution Explorer
Git Changes
Select Repository
Activate Windows
Go to Settings to activate Windows.

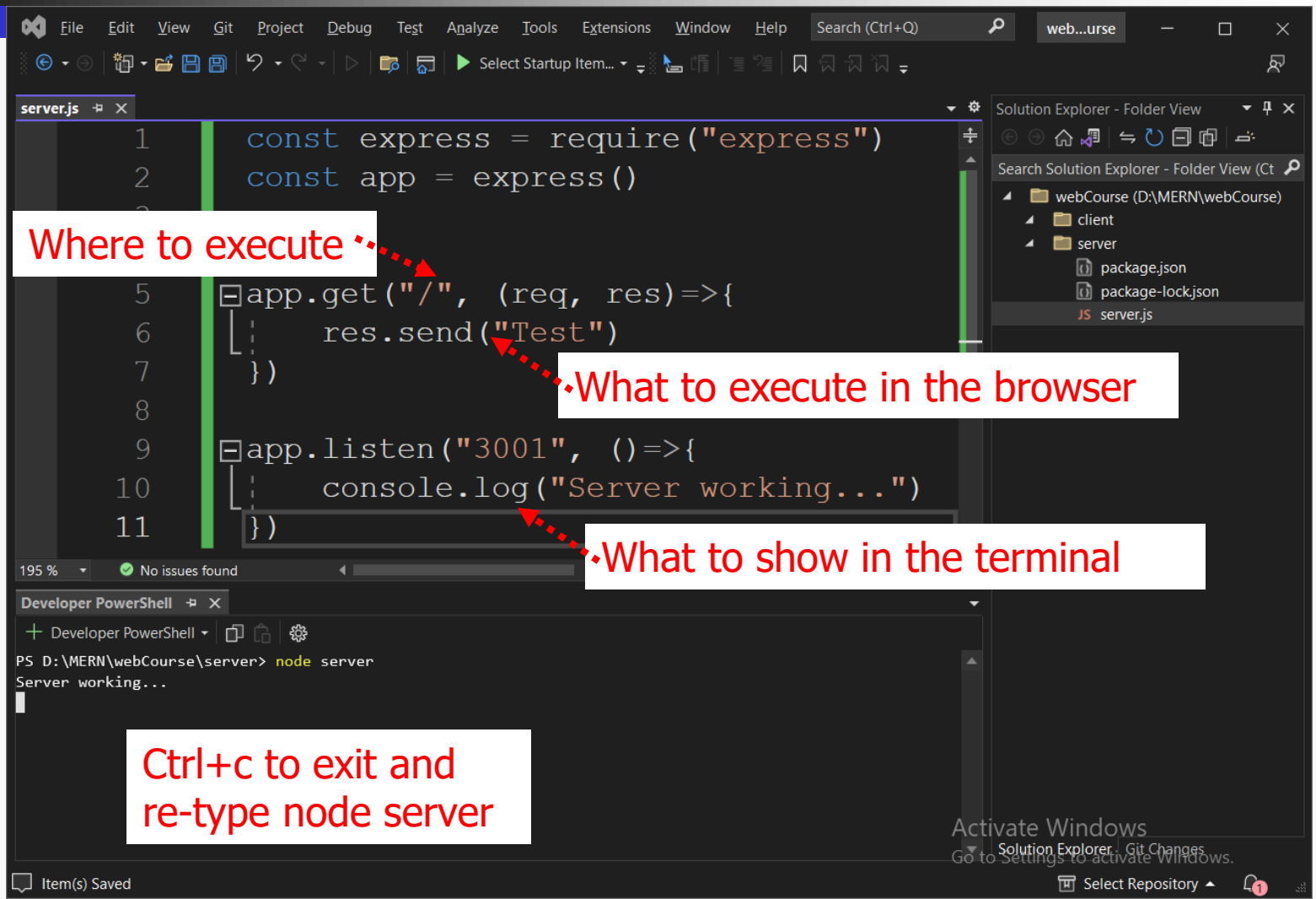
server.js


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
web...urse
Select Startup Item...
server.js
1 const express = require("express")
2 const app = express()
3
4
5 app.listen("3001", ()=>{
6 console.log("Server working...")
7 })
Port 3001
Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ct
webCourse (D:\MERN\webCourse)
client
server
package.json
package-lock.json
server.js
195 %
No issues found
Ln: 7 Ch: 3 TABS CRLF
Developer PowerShell
Developer PowerShell
PS D:\MERN\webCourse\server> node server
Server working...
What is done when listening to port 3001
Notice that no prompt appears!
Solution Explorer Git Changes
Ready
Select Repository

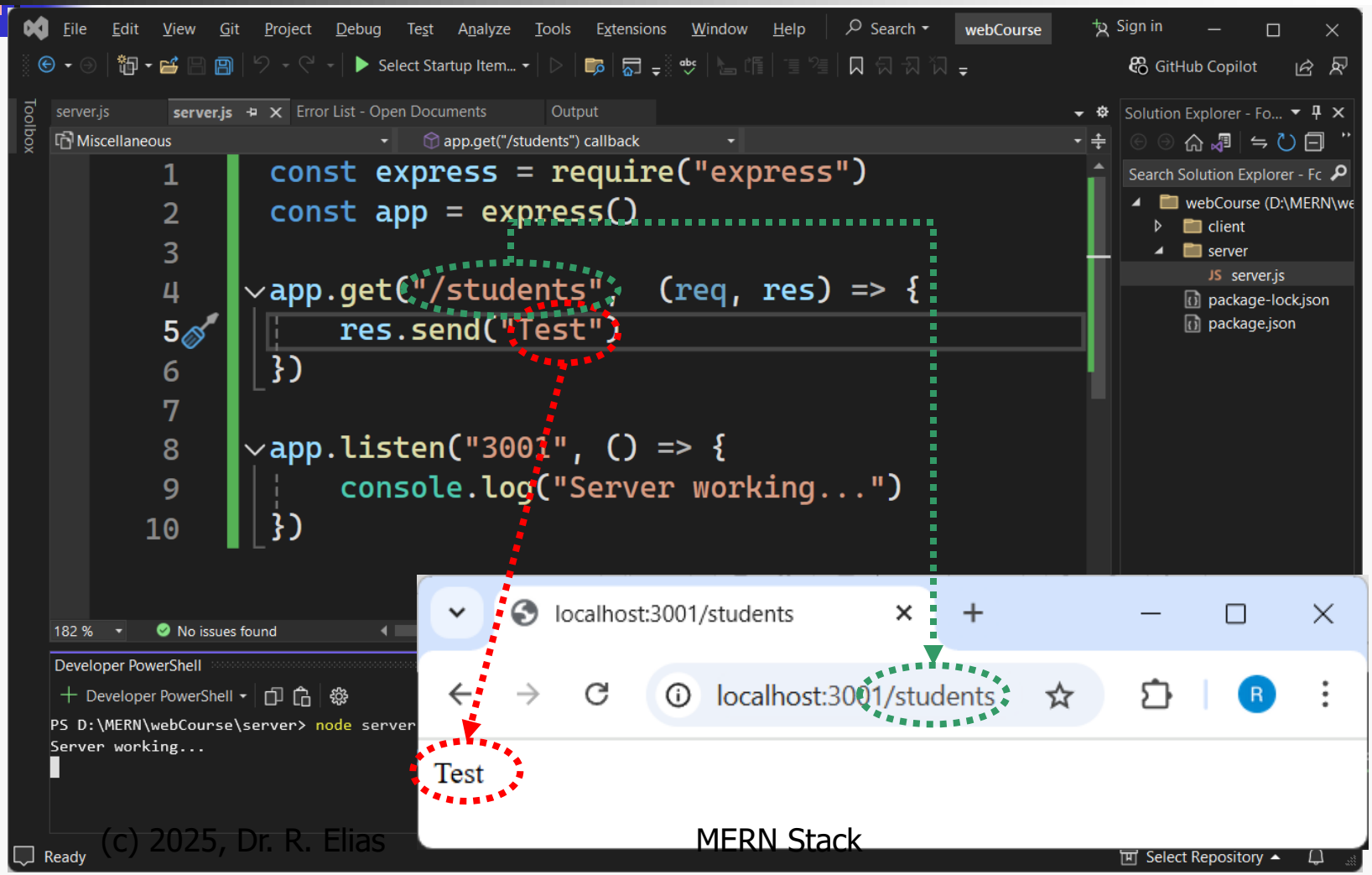
server.js


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
web...urse
Select Startup Item...
server.js
1 const express = require("express")
2 const app = express()
3
4
5 app.listen("3001", ()=>{
6 console.log("Server working...")
7 })
Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ct
webCourse (D:\MERN\webCourse)
client
server
package.json
package-lock.json
server.js
195 %
No issue
Developer PowerShell
Developer PowerShel
PS D:\MERN\webCourse
Server working...
Error
localhost:3001
Cannot GET /
Solution Explorer
Git Changes
Ready
Select Repository

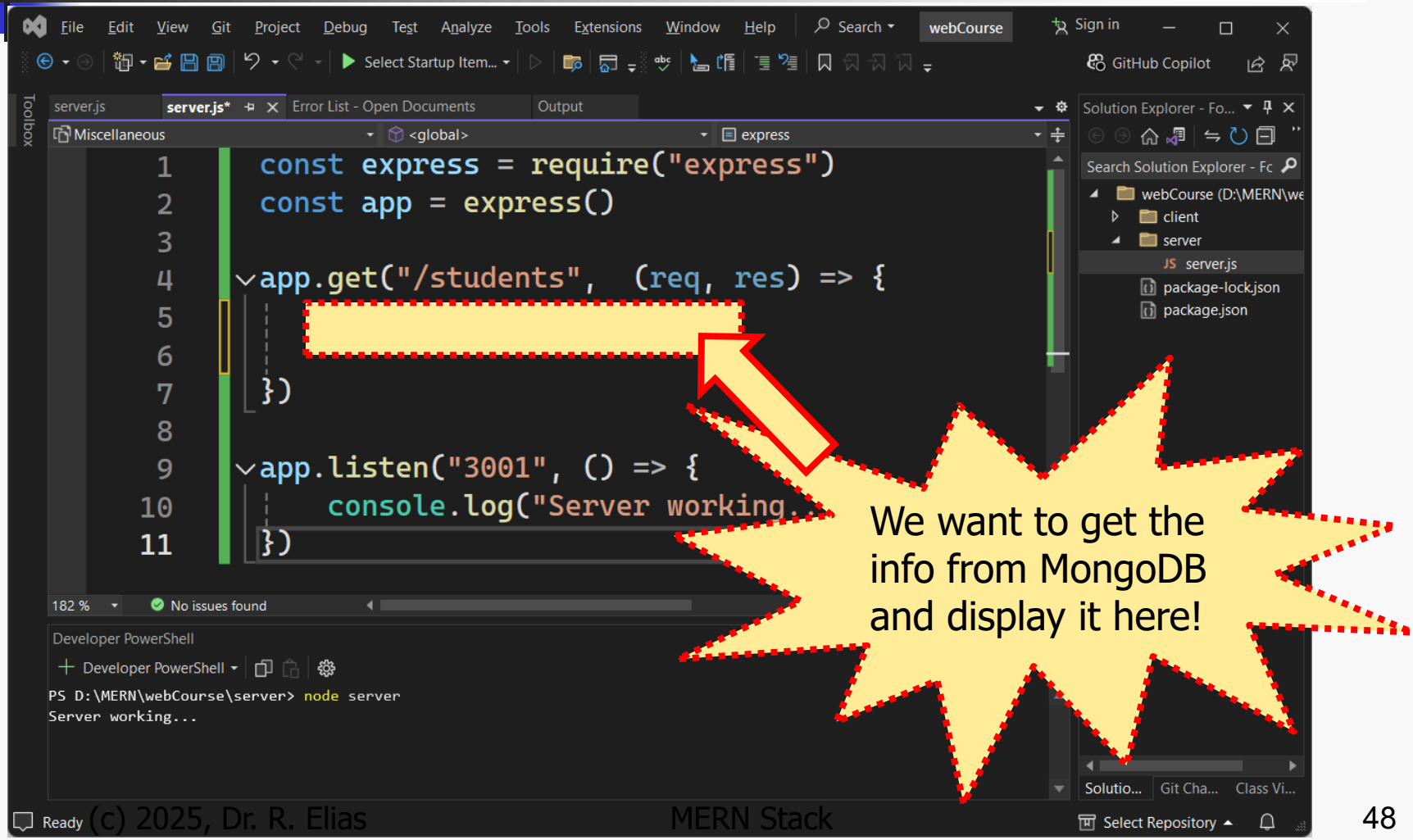
server.js


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
web...urse
Select Startup Item...
server.js
1 const express = require("express")
2 const app = express()
5 app.get("/", (req, res)=>{
6 res.send("Test")
7 })
8
9 app.listen("3001", ()=>{
10 console.log("Server working...")
11 })
Where to execute
What to execute in the browser
What to show in the terminal
195 %
No issues found
Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ct
webCourse (D:\MERN\webCourse)
client
server
package.json
package-lock.json
server.js
Developer PowerShell
Developer PowerShell
PS D:\MERN\webCourse\server> node server
Server working...
Ctrl+c to exit and
re-type node server
Activate Windows
Go to Settings to activate Windows.
Solution Explorer
Git Changes
Item(s) Saved
Select Repository

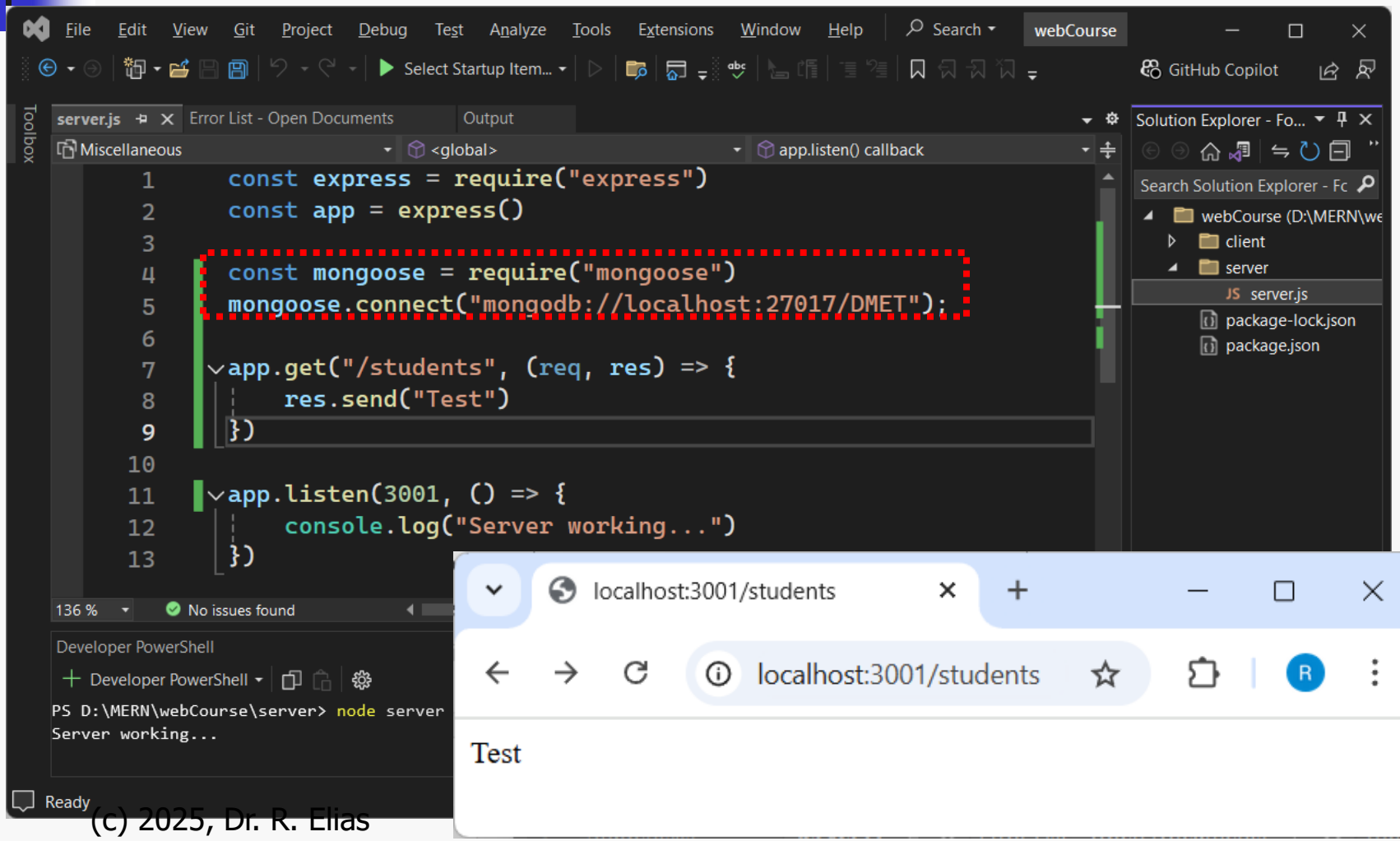
server.js


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search
webCourse
Sign in
Select Startup Item...
GitHub Copilot
server.js
server.js
Error List - Open Documents
Output
Miscellaneous
app.get("/students") callback
Solution Explorer - Fo...
Search Solution Explorer - Fc
webCourse (D:\MERN\we
client
server
server.js
package-lock.json
package.json
1 const express = require("express")
2 const app = express()
3
4 app.get("/students", (req, res) => {
5 res.send("Test")
6 })
7
8 app.listen("3001", () => {
9 console.log("Server working...")
10 })
182 %
No issues found
Developer PowerShell
Developer PowerShell
PS D:\MERN\webCourse\server> node server
Server working...
localhost:3001/students
localhost:3001/students
Test
Ready
Select Repository

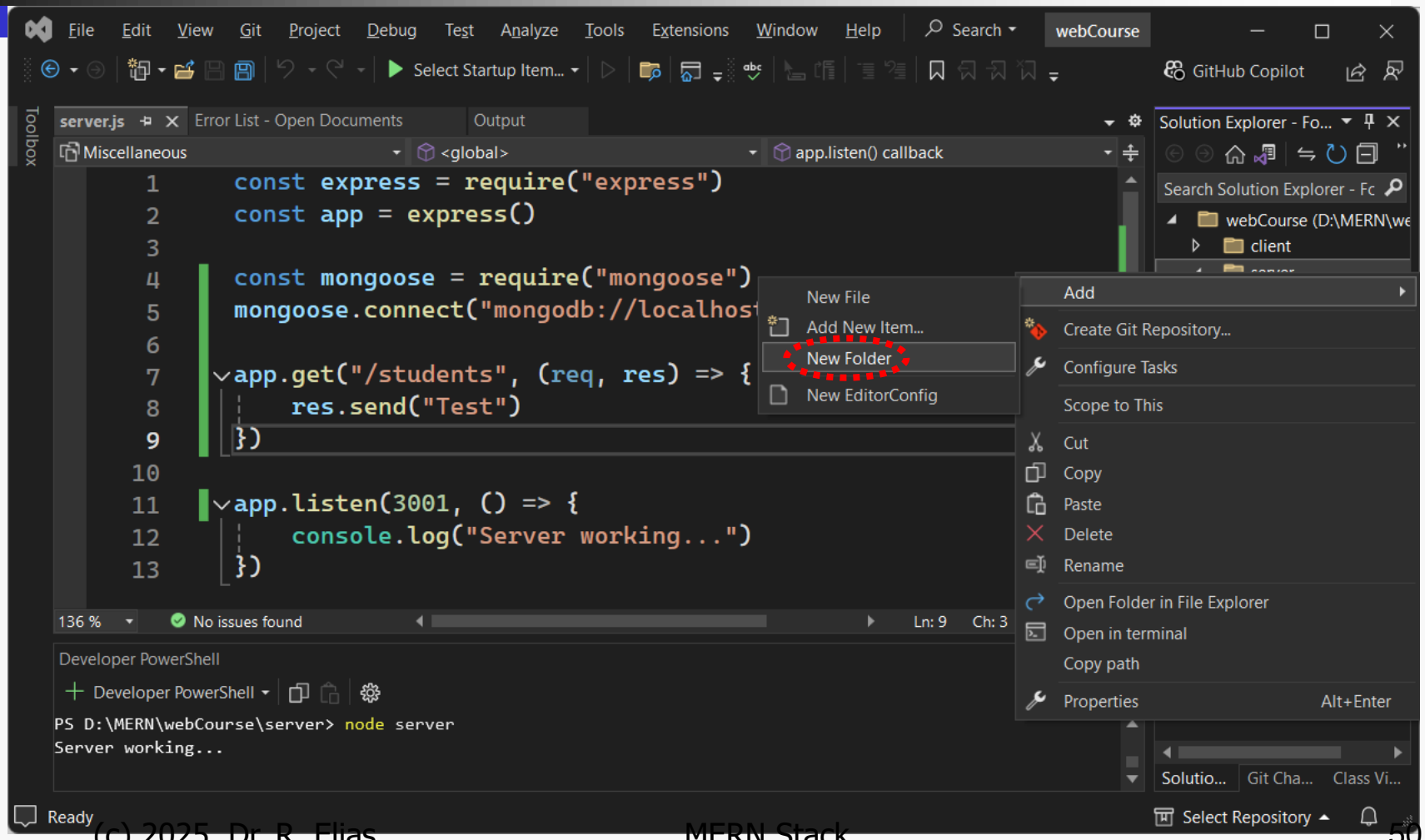
Mongo DB + Express: server.js



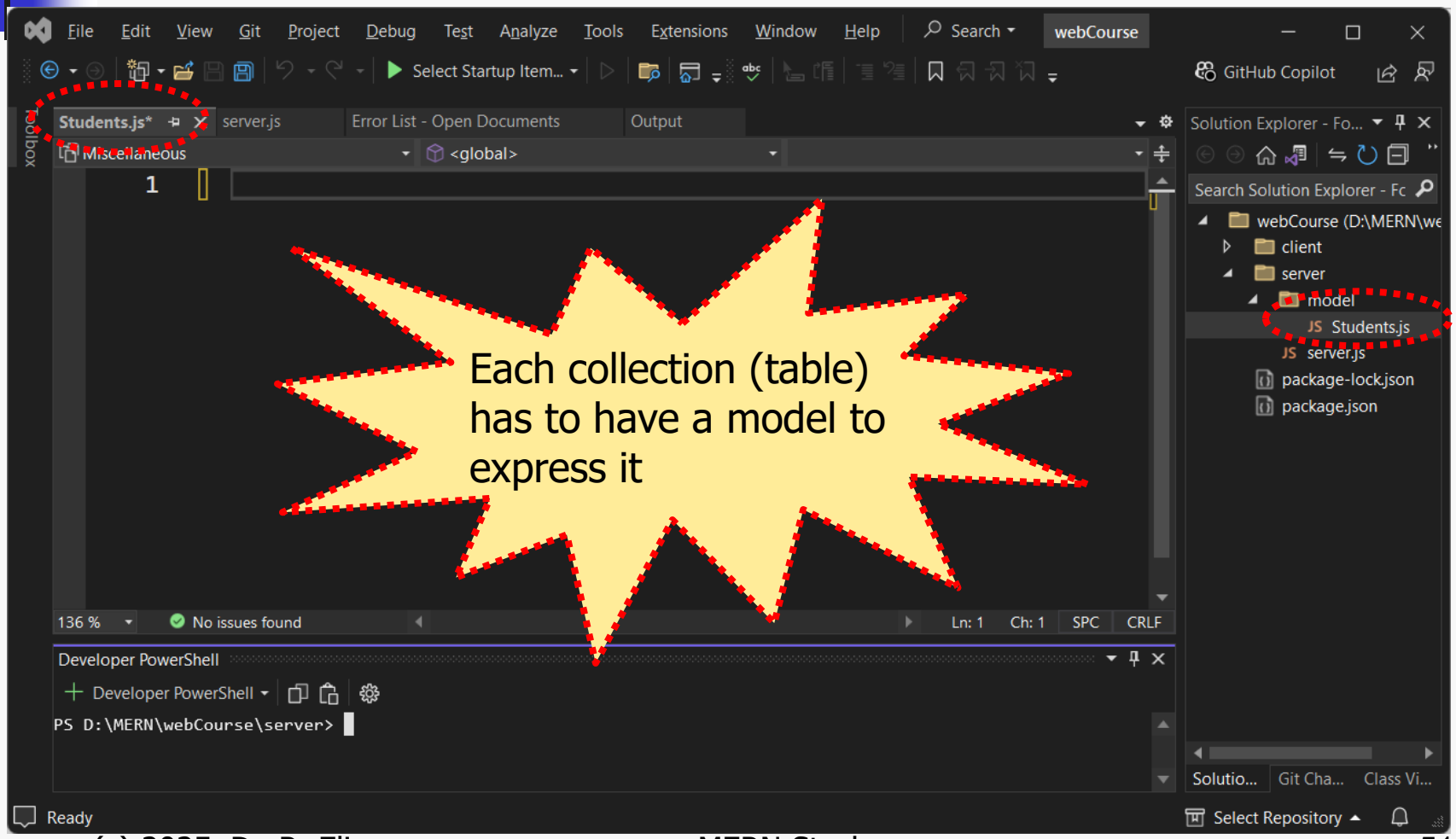
server.js



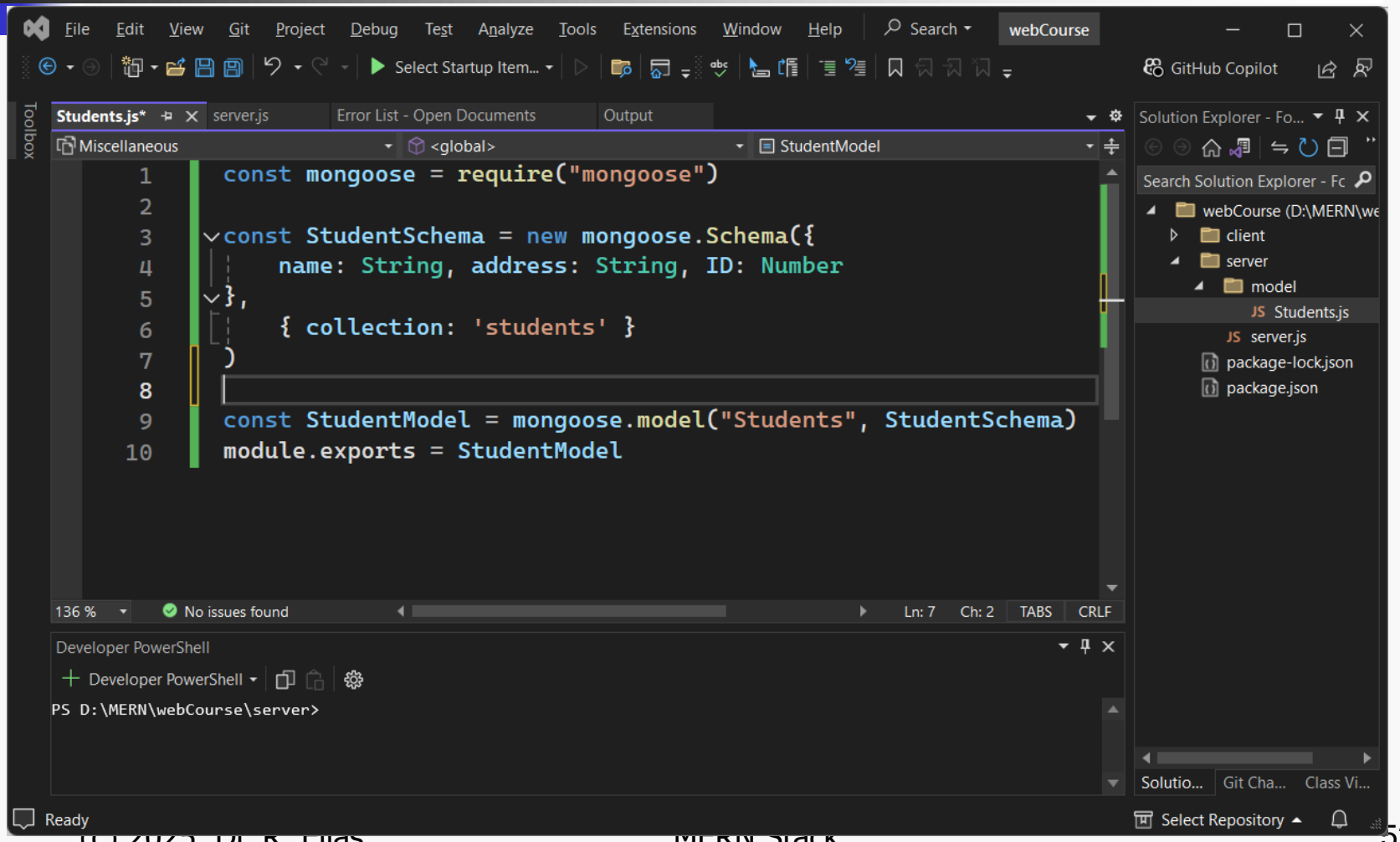
server.js: Interaction using Model



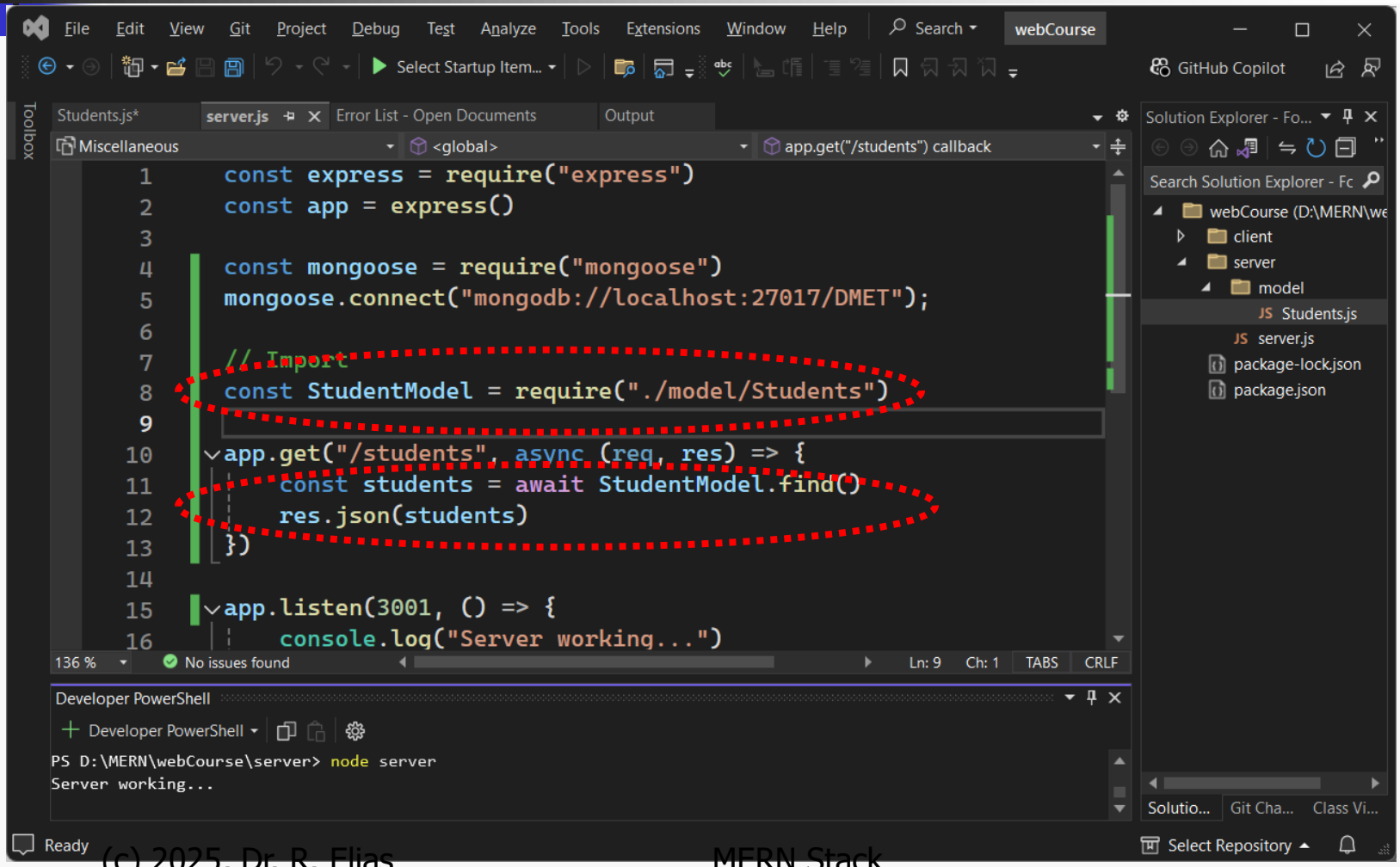
Persons.js: Interaction using Model



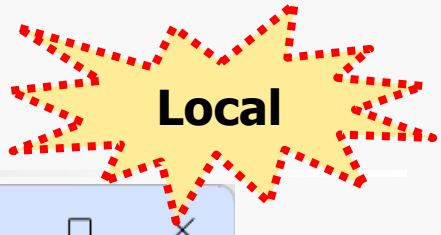
Persons.js: Interaction using Model



Mongo DB + Express: Interaction using Model



Back-end



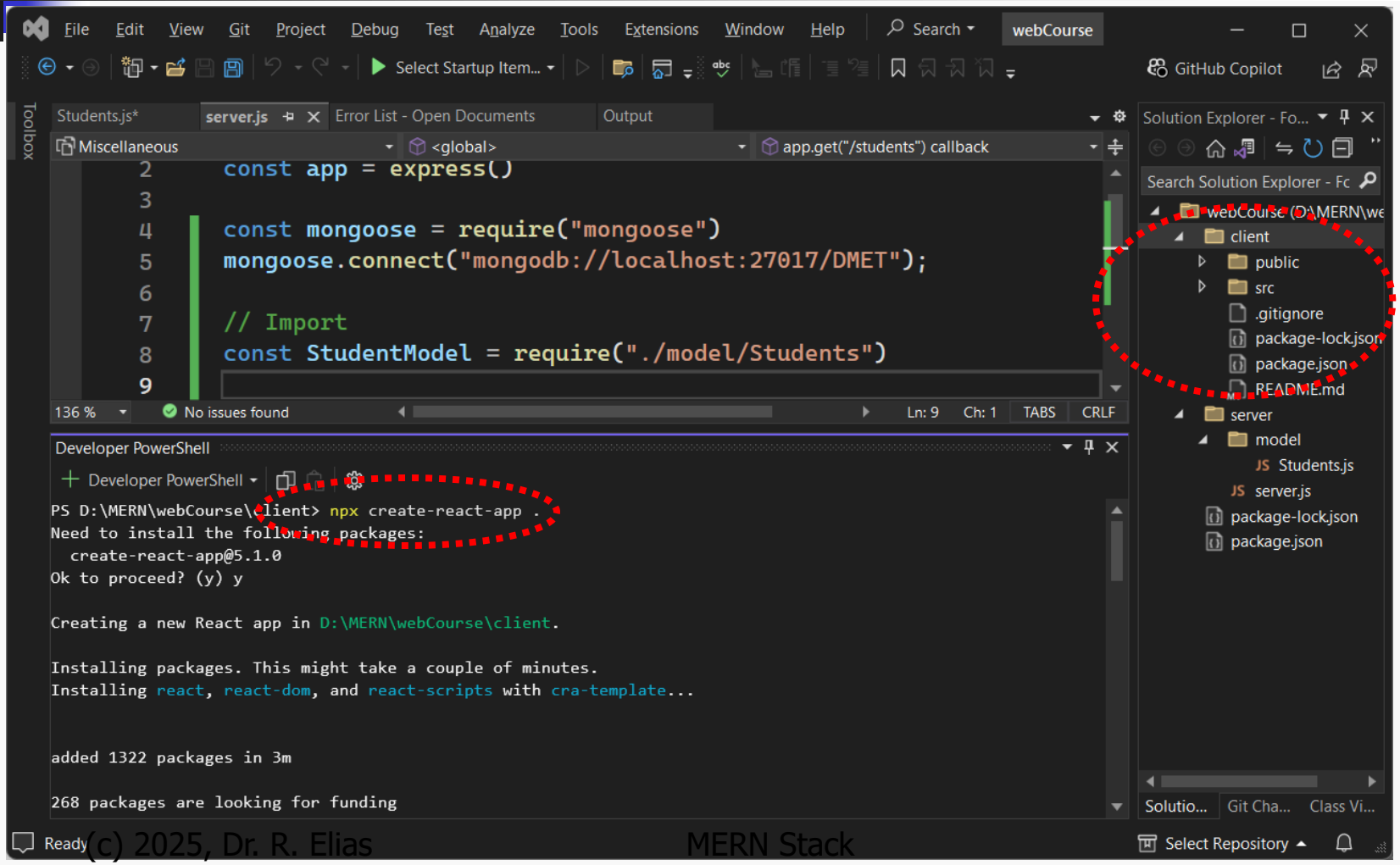
localhost:3001/students

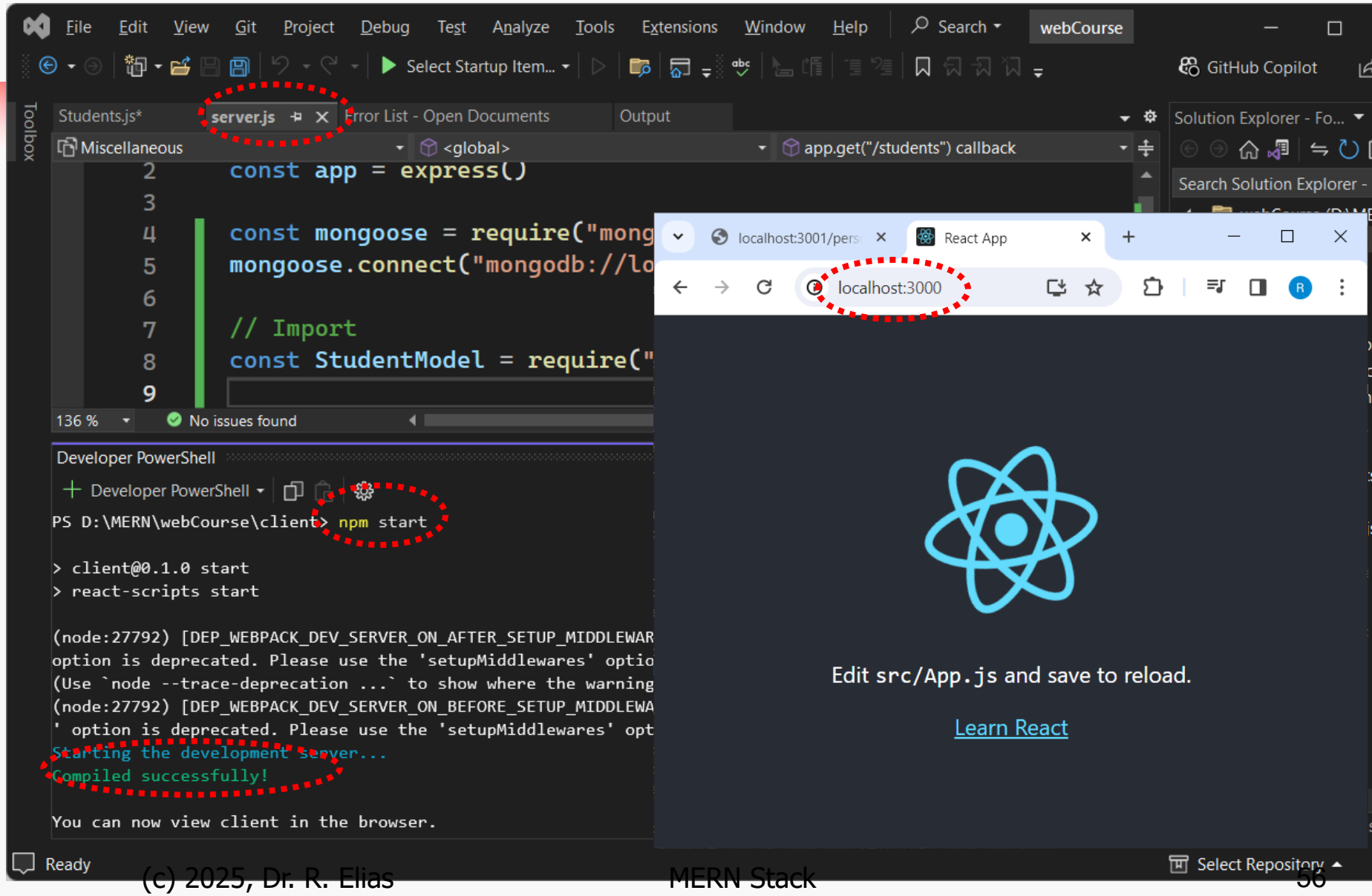
localhost:3001/students

Pretty-print

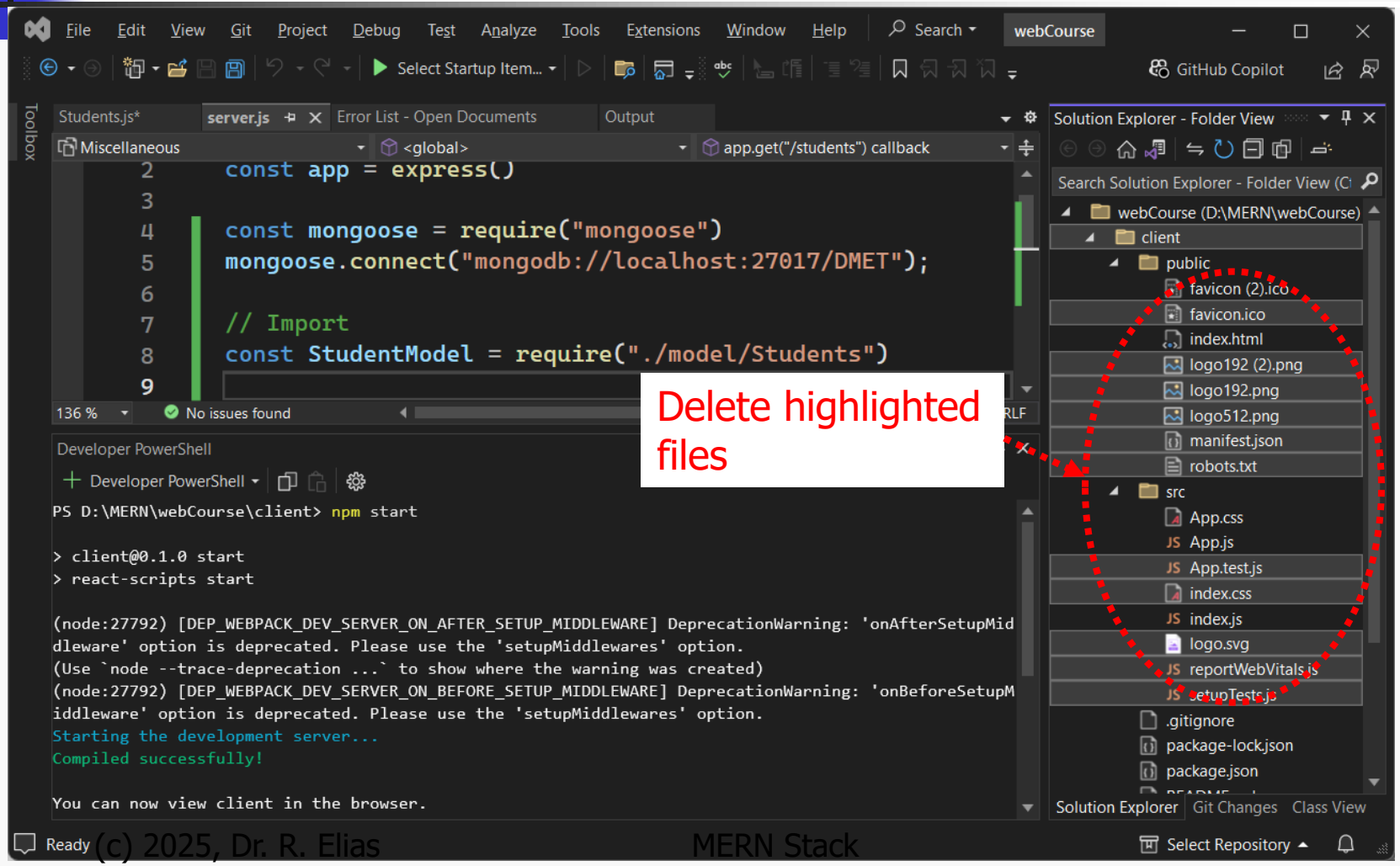

```
[
  {
    "_id": "681cbe0fb3e83f0dc5ce1c6c",
    "name": "Rimon",
    "address": "GUC",
    "ID": 123456
  },
  {
    "_id": "681cc18763b4f82c2dac33ec",
    "name": "Ahmed",
    "address": "New Cairo",
    "ID": 654321
  },
  {
    "_id": "681cc20563b4f82c2dac33ee",
    "name": "Ahmed",
    "address": "New Cairo",
    "ID": 333322
  }
]
```

React

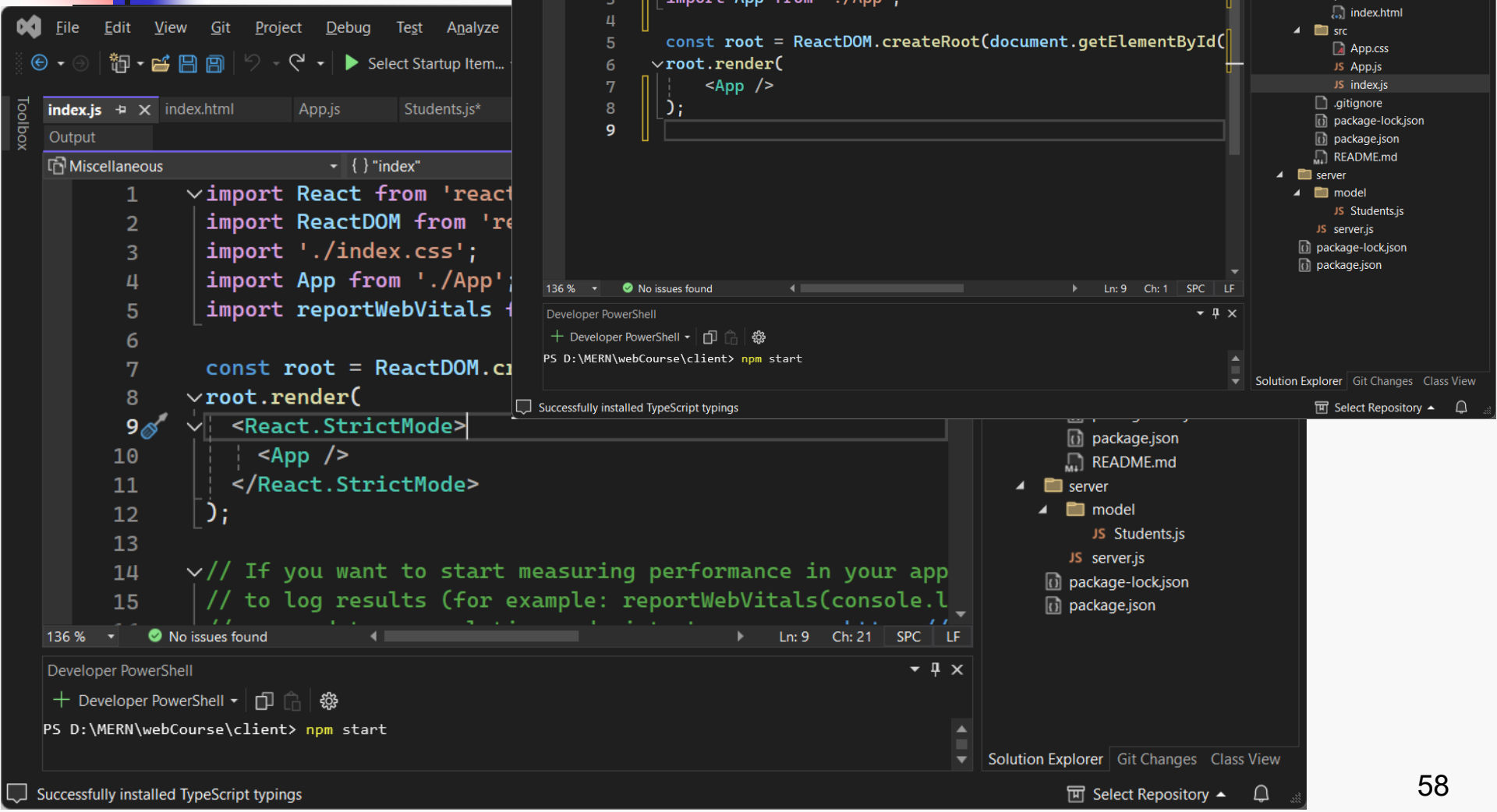

File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search
webCourse
Select Startup Item...
GitHub Copilot
Students.js*
server.js
Error List - Open Documents
Output
Miscellaneous
<global>
app.get("/students") callback
const app = express()
const mongoose = require("mongoose")
mongoose.connect("mongodb://localhost:27017/DMET");
// Import
const StudentModel = require("./model/Students")
136 %
No issues found
Ln: 9
Ch: 1
TABS
CRLF
Developer PowerShell
PS D:\MERN\webCourse\client> npx create-react-app .
Need to install the following packages:
create-react-app@5.1.0
Ok to proceed? (y) y
Creating a new React app in D:\MERN\webCourse\client.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
added 1322 packages in 3m
268 packages are looking for funding
Solution Explorer - Fo...
Search Solution Explorer - Fc
webCourse (D:\MERN\we
client
public
src
.gitignore
package-lock.json
package.json
README.md
server
model
Students.js
server.js
package-lock.json
package.json
Solutio...
Git Cha...
Class Vi...
Ready
Select Repository


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search
webCourse
Select Startup Item...
GitHub Copilot
Students.js*
server.js
Error List - Open Documents
Output
Solution Explorer - Fo...
Search Solution Explorer -
Miscellaneous
<global>
app.get("/students") callback
const app = express()
const mongoose = require("mong
mongoose.connect("mongodb://lo
// Import
const StudentModel = require("
136 %
No issues found
Developer PowerShell
Developer PowerShell
PS D:\MERN\webCourse\client> npm start
> client@0.1.0 start
> react-scripts start
(node:27792) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWAR
option is deprecated. Please use the 'setupMiddlewares' optio
(Use `node --trace-deprecation ...` to show where the warning
(node:27792) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWA
' option is deprecated. Please use the 'setupMiddlewares' opt
Starting the development server...
Compiled successfully!
You can now view client in the browser.
localhost:3001/pers
React App
localhost:3000
Edit src/App.js and save to reload.
Learn React
Ready
Select Repository

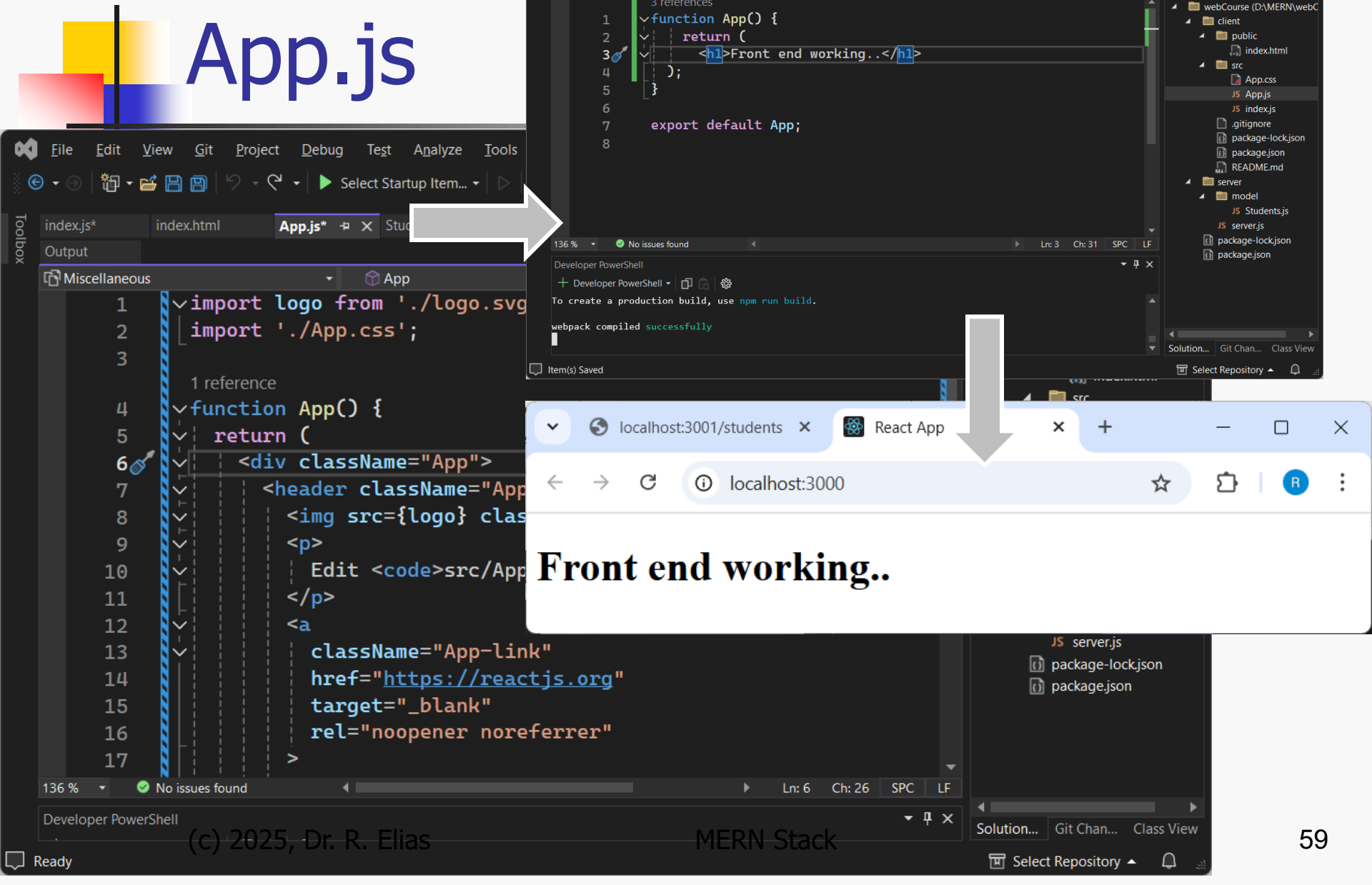
React


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search
webCourse
Select Startup Item...
GitHub Copilot
Students.js*
server.js
Error List - Open Documents
Output
Miscellaneous
<global>
app.get("/students") callback
const app = express()
const mongoose = require("mongoose")
mongoose.connect("mongodb://localhost:27017/DMET");
// Import
const StudentModel = require("./model/Students")
136 %
No issues found
Delete highlighted files
Developer PowerShell
PS D:\MERN\webCourse\client> npm start
> client@0.1.0 start
> react-scripts start
(node:27792) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:27792) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!
You can now view client in the browser.
Solution Explorer - Folder View
Search Solution Explorer - Folder View
webCourse (D:\MERN\webCourse)
client
public
favicon (2).ico
favicon.ico
index.html
logo192 (2).png
logo192.png
logo512.png
manifest.json
robots.txt
src
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
Solution Explorer
Git Changes
Class View
Ready
Select Repository

index.js


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
webCourse
Select Startup Item...
GitHub Copilot
index.js* index.html App.js Students.js* server.js Error List - Open Documents
Output
Miscellaneous
{ } "index"
root
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
const root = ReactDOM.createRoot(document.getElementById(
root.render(
<App />
);
136 %
No issues found
Ln: 9 Ch: 1 SPC LF
Developer PowerShell
PS D:\MERN\webCourse\client> npm start
Successfully installed TypeScript typings
Solution Explorer - Folder View
Search Solution Explorer - Folder View
webCourse (D:\MERN\webCourse)
client
public
index.html
src
App.css
App.js
index.js
.gitignore
package-lock.json
package.json
README.md
server
model
Students.js
server.js
package-lock.json
package.json
Solution Explorer Git Changes Class View
Select Repository
Toolbox
File Edit View Git Project Debug Test Analyze
Select Startup Item...
index.js index.html App.js Students.js*
Output
Miscellaneous
{ } "index"
import React from 'react
import ReactDOM from 're
import './index.css';
import App from './App'
import reportWebVitals
const root = ReactDOM.cr
root.render(
<React.StrictMode>
<App />
</React.StrictMode>
);
// If you want to start measuring performance in your app
// to log results (for example: reportWebVitals(console.l
136 %
No issues found
Ln: 9 Ch: 21 SPC LF
Developer PowerShell
PS D:\MERN\webCourse\client> npm start
Successfully installed TypeScript typings
package.json
README.md
server
model
Students.js
server.js
package-lock.json
package.json
Solution Explorer Git Changes Class View
Select Repository

App.js

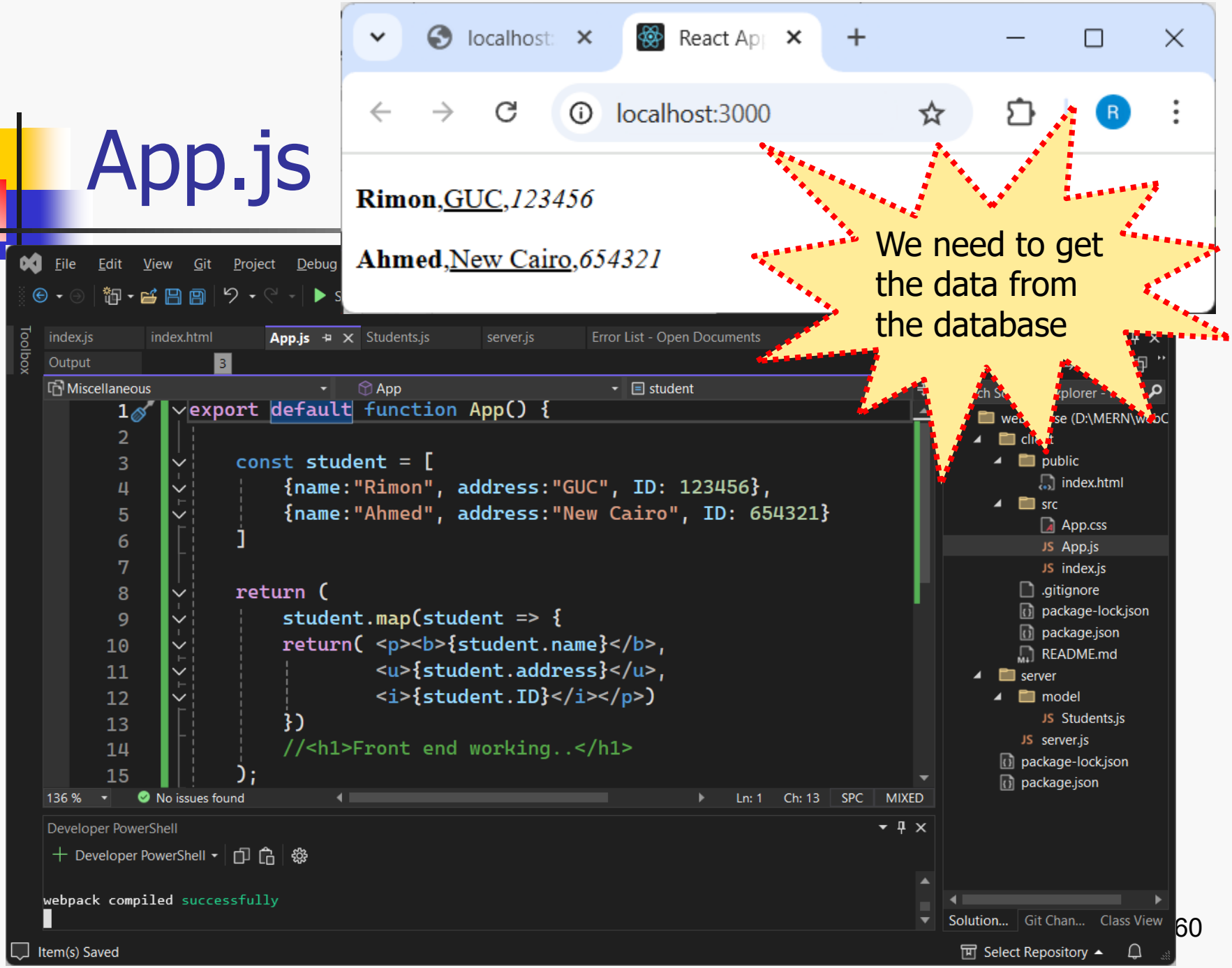

File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
webCourse
index.js index.html App.js Students.js server.js Error List - Open Documents
Output
Miscellaneous App
3 references
function App() {
return (
<h1>Front end working..</h1>
);
}
export default App;
Solution Explorer - Folde...
webCourse (D:\MERN\webC
client
public
index.html
src
App.css
App.js
index.js
.gitignore
package-lock.json
package.json
README.md
server
model
Students.js
server.js
package-lock.json
package.json
Developer PowerShell
To create a production build, use npm run build.
webpack compiled successfully
Item(s) Saved
Select Repository
import logo from './logo.svg
import './App.css';
1 reference
function App() {
return (

<header className="App
<img src={logo} clas

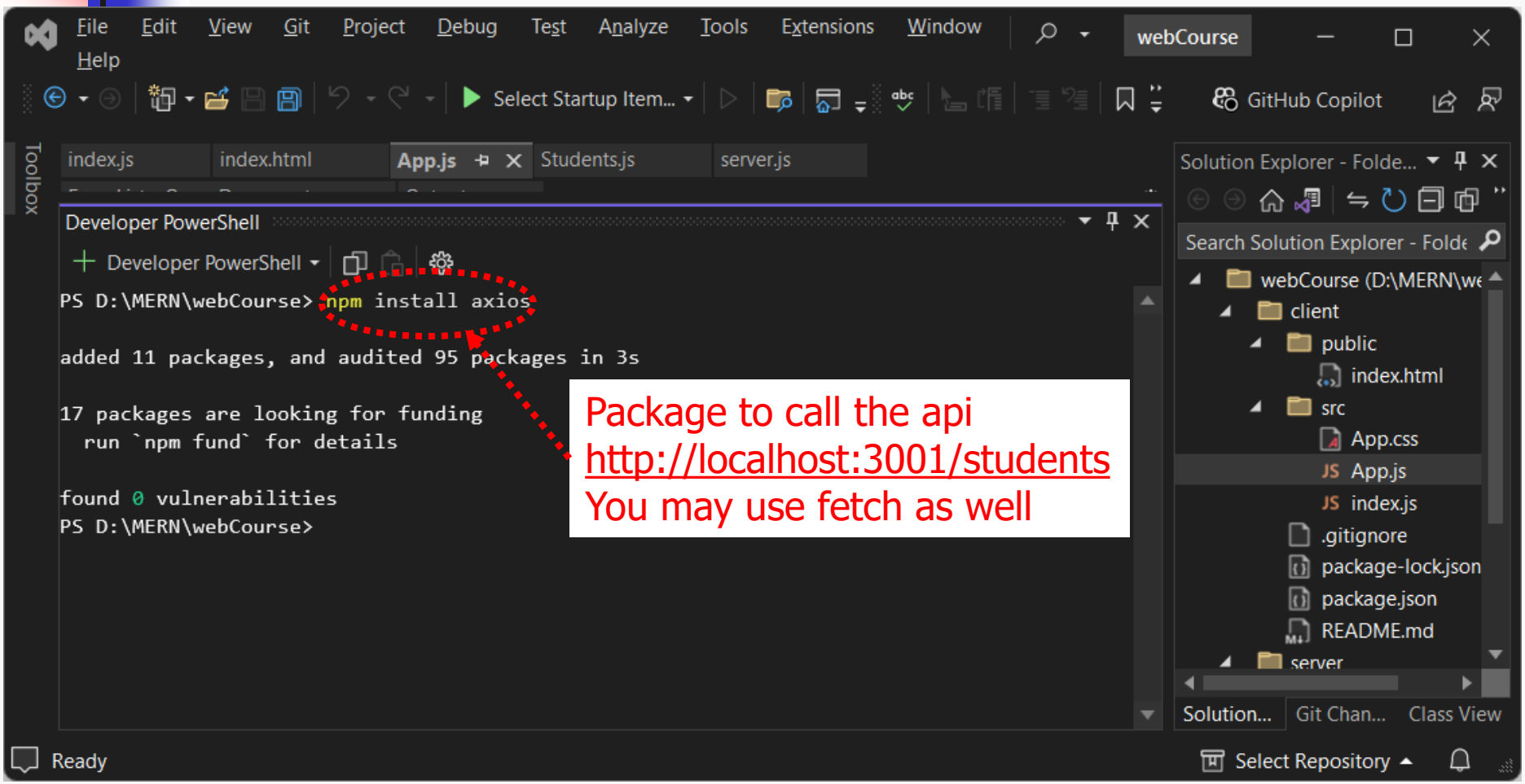
Edit <code>src/App

<a
className="App-link"
href="https://reactjs.org"
target="_blank"
rel="noopener noreferrer"
>
136 % No issues found Ln: 6 Ch: 26 SPC LF
Developer PowerShell
Solution... Git Chan... Class View
Ready
localhost:3001/students
React App
localhost:3000
Front end working..

App.js


localhost:
React Ap
localhost:3000
Rimon,GUC,123456
Ahmed,New Cairo,654321
We need to get the data from the database
File Edit View Git Project Debug
index.js
index.html
App.js
Students.js
server.js
Error List - Open Documents
Output
Miscellaneous
App
student
export default function App() {
const student = [
{name:"Rimon", address:"GUC", ID: 123456},
{name:"Ahmed", address:"New Cairo", ID: 654321}
]
return (
student.map(student => {
return({student.name},
<u>{student.address}</u>,
{student.ID})
})
//<h1>Front end working..</h1>
);
136 %
No issues found
Ln: 1
Ch: 13
SPC
MIXED
Developer PowerShell
Developer PowerShell
webpack compiled successfully
public
index.html
src
App.css
App.js
index.js
.gitignore
package-lock.json
package.json
README.md
server
model
Students.js
server.js
package-lock.json
package.json
Solution...
Git Chan...
Class View
Item(s) Saved
Select Repository

Axios


File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
webCourse
Select Startup Item...
GitHub Copilot
index.js
index.html
App.js
Students.js
server.js
Developer PowerShell
Developer PowerShell
PS D:\MERN\webCourse> npm install axios
added 11 packages, and audited 95 packages in 3s
17 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
PS D:\MERN\webCourse>
Package to call the api
http://localhost:3001/students
You may use fetch as well
Solution Explorer - Folde...
Search Solution Explorer - Folde
webCourse (D:\MERN\we
client
public
index.html
src
App.css
App.js
index.js
.gitignore
package-lock.json
package.json
README.md
server
Solution...
Git Chan...
Class View
Ready
Select Repository

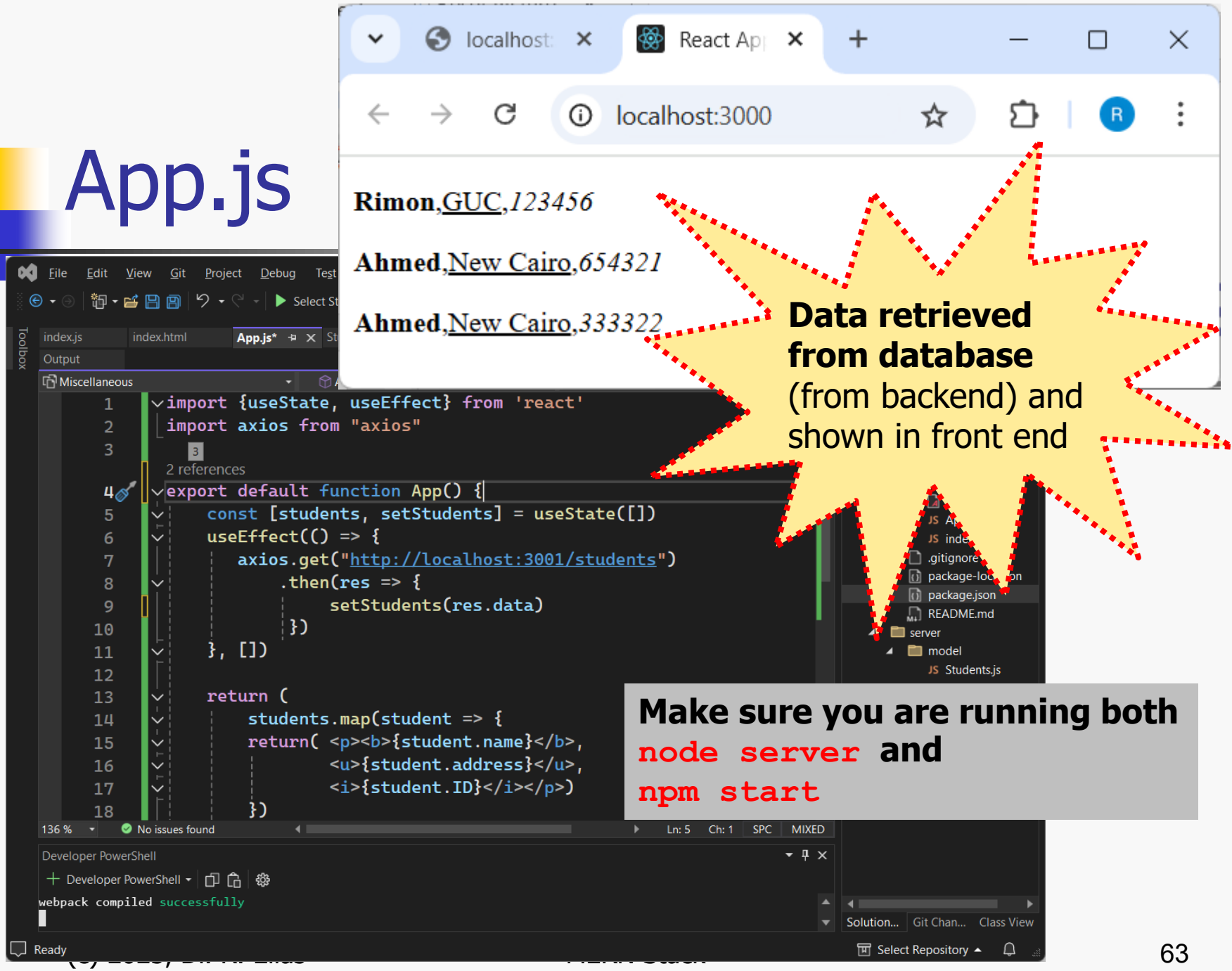
server.js

npm install cors

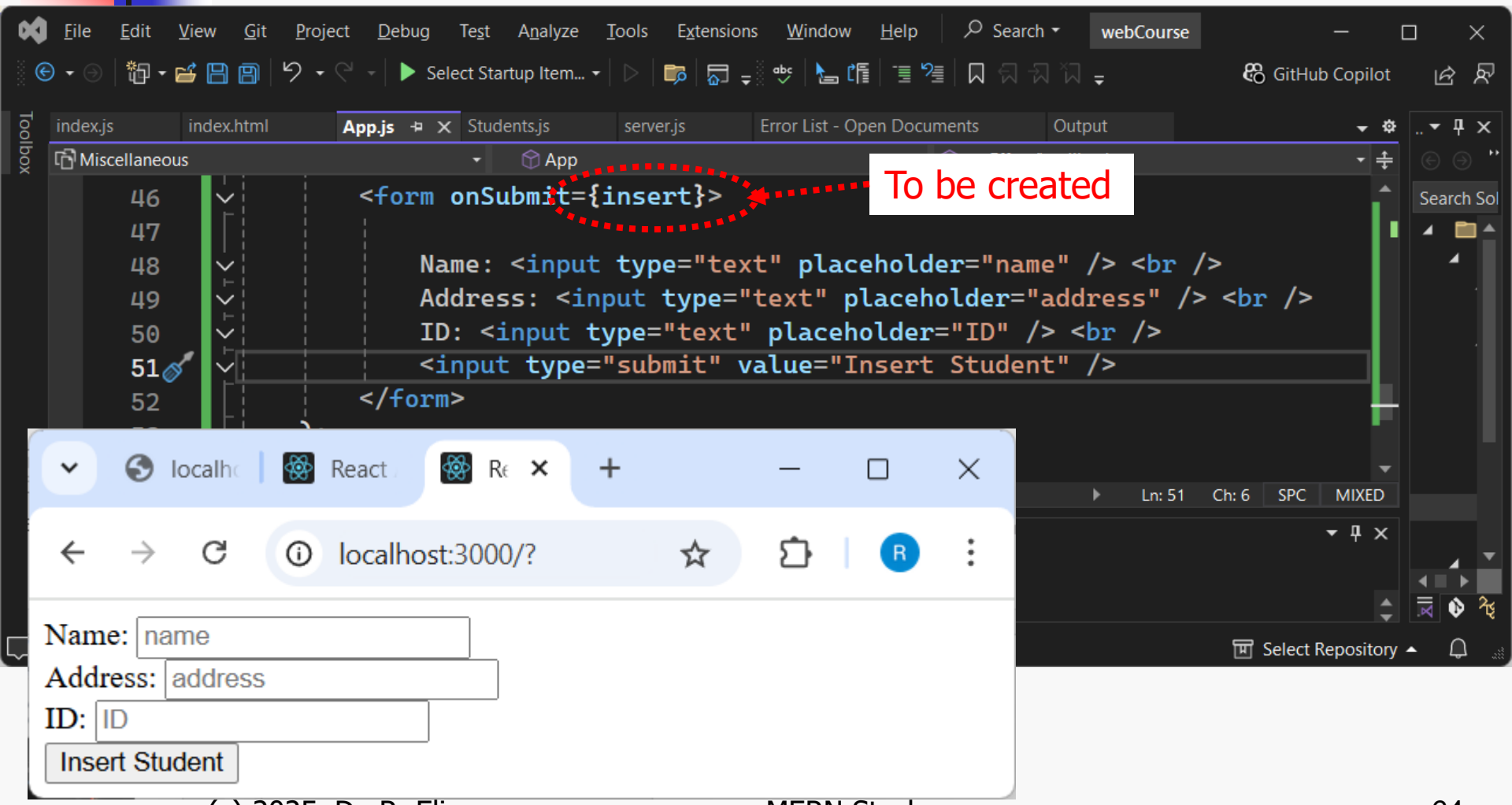
```
const express = require("express")
const app = express()

var cors = require('cors');
app.use(cors());

const mongoose = require("mongoose")
```

Add these lines

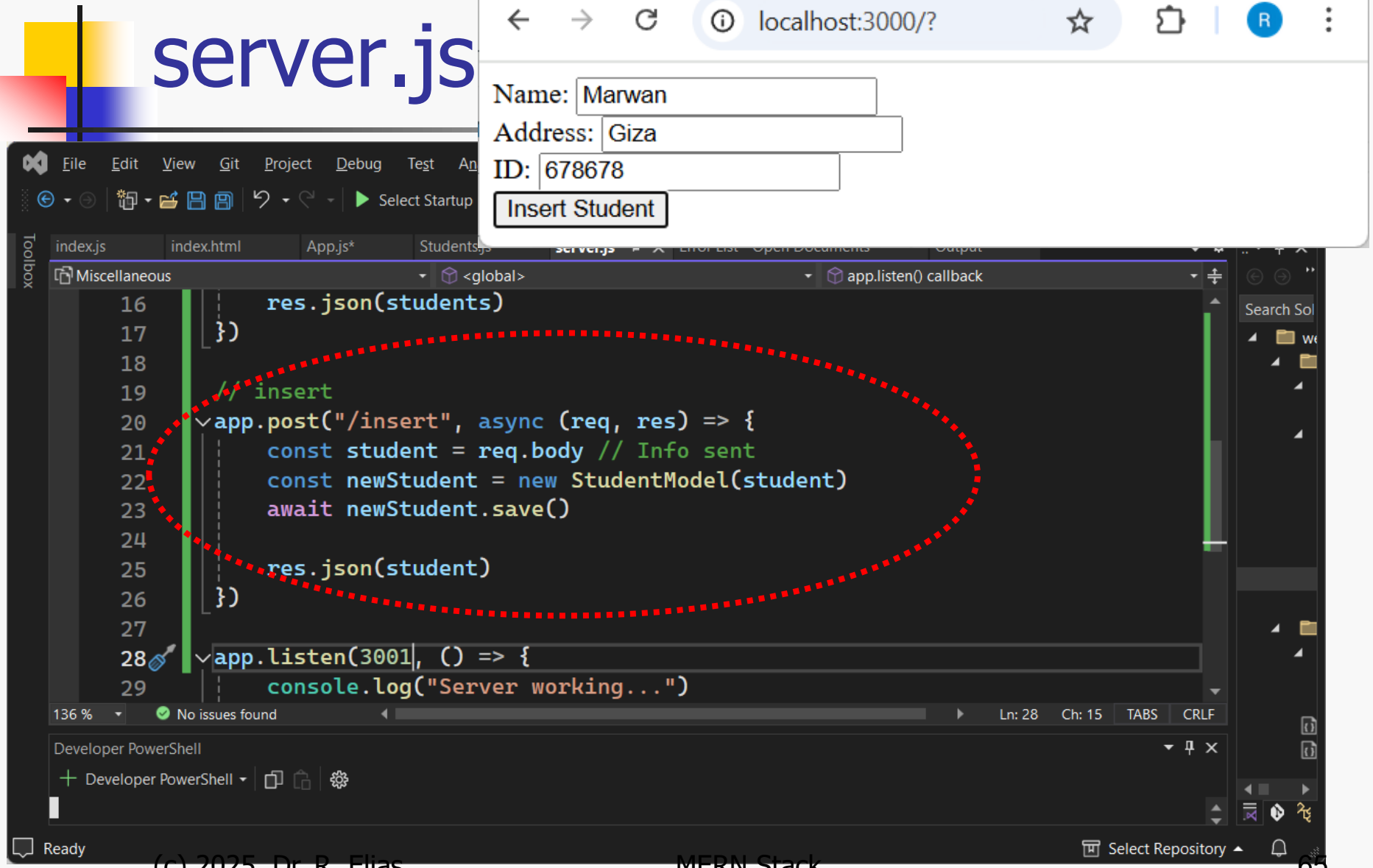
App.js


localhost:
React Ap
localhost:3000
Rimon,GUC,123456
Ahmed,New Cairo,654321
Ahmed,New Cairo,333322
Data retrieved from database (from backend) and shown in front end
File Edit View Git Project Debug Test
index.js
index.html
App.js*
Output
Miscellaneous
import {useState, useEffect} from 'react'
import axios from "axios"
2 references
export default function App() {
const [students, setStudents] = useState([])
useEffect(() => {
axios.get("http://localhost:3001/students")
.then(res => {
setStudents(res.data)
})
}, [])
return (
students.map(student => {
return({student.name},
<u>{student.address}</u>,
{student.ID})
})
136 %
No issues found
Ln: 5 Ch: 1 SPC MIXED
Developer PowerShell
webpack compiled successfully
package.json
README.md
server
model
Students.js
Make sure you are running both node server and npm start
Solution...
Git Chan...
Class View
Ready
Select Repository

User Interface in App.js



server.js

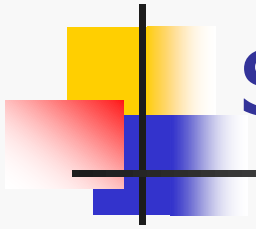

localh
React
Re
localhost:3000/?
Name: Marwan
Address: Giza
ID: 678678
Insert Student
File Edit View Git Project Debug Test An
Select Startup
index.js
index.html
App.js*
Miscellaneous
<global>
app.listen() callback
16 res.json(students)
17 })
18
19 // insert
20 app.post("/insert", async (req, res) => {
21 const student = req.body // Info sent
22 const newStudent = new StudentModel(student)
23 await newStudent.save()
24
25 res.json(student)
26 })
27
28 app.listen(3001, () => {
29 console.log("Server working...")
136 %
No issues found
Ln: 28
Ch: 15
TABS
CRLF
Developer PowerShell
Developer PowerShell
Ready
Select Repository

App.js

```
5     const [students, setStudents] = useState([])
6     const [name, setName] = useState("")
7     const [address, setAddress] = useState("")
8     const [ID, setID] = useState("")
```

```
18    const insert = async (e) => {
19        await Axios.post("http://localhost:3001/insert", {
20            name: name, //"Emad",
21            address: address, // "GUC",
22            ID: ID // 222244
23        })
24        alert('Inserted!');
```

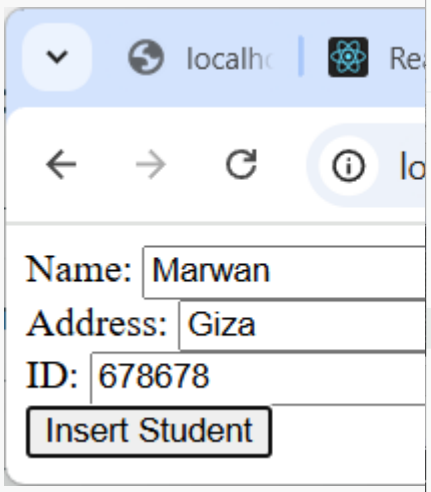
```
34    <form onSubmit={insert}>
35
36        Name: <input type="text" value={name} placeholder="name" onChange={e=>setName(e.target.value)} /> <br />
37        Address: <input type="text" value={address} placeholder="address" onChange={e => setAddress(e.target.value)} />
38        ID: <input type="text" value={ID} placeholder="ID" onChange={e => setID(e.target.value)} /> <br />
39        <input type="submit" value="Insert Student" />
40    </form>
```



server.js

```
1   const express = require("express")
2   const app = express()
3   app.use(express.json())
```

Interface & Compass



Local

MongoDB Compass - localhost:27017/DMET.students

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (1)

Search connections

localhost:27017

DMET

students

admin

config

local

webCourse

students | webCourse | DMET

localhost:27017 > DMET > students

Open MongoDB sh

Documents 4 | Aggregations | Schema | Indexes 1 | Validation

Type a query: { field: 'value' } or Ge

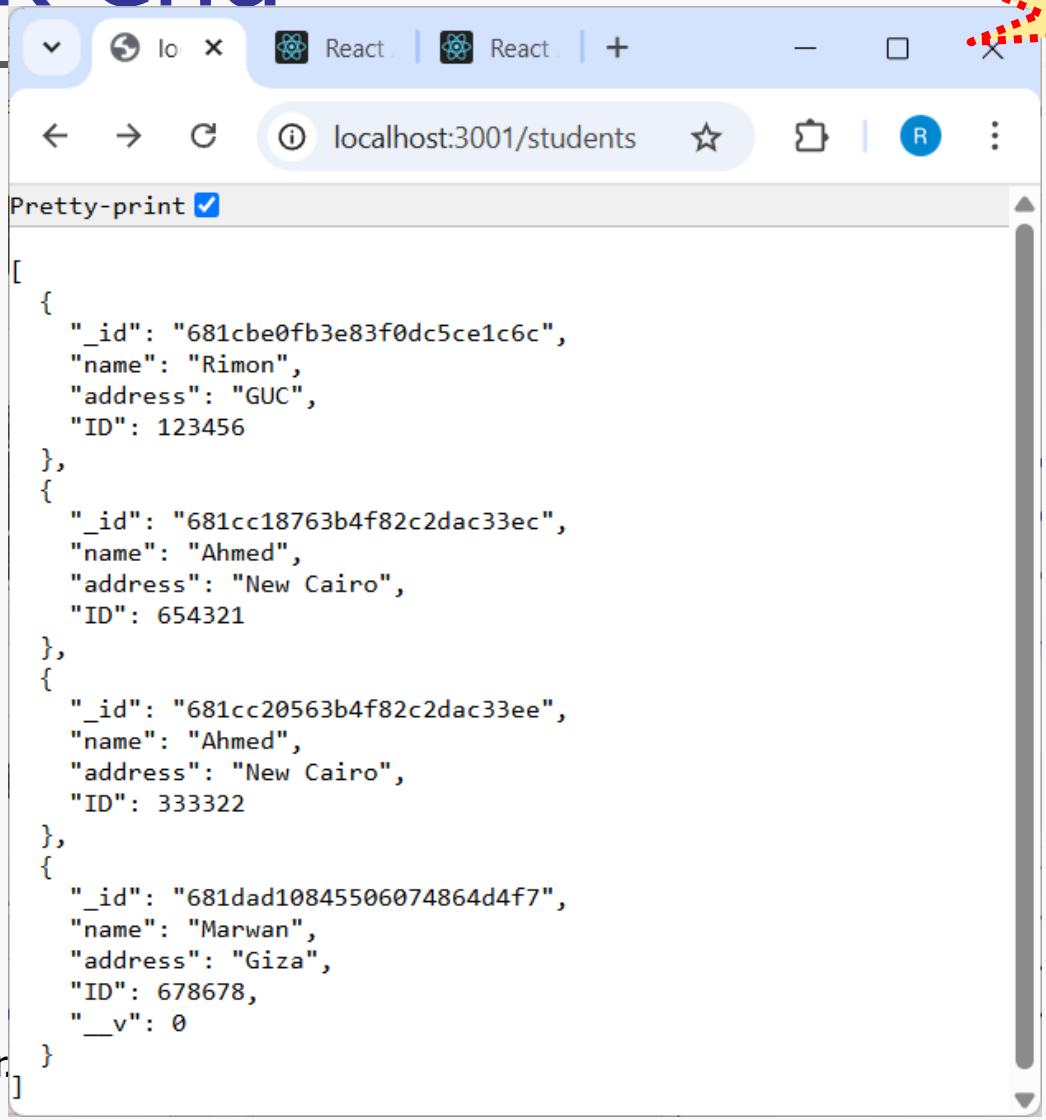
Explain | Reset | Find | Options

25 | 1 – 4 of 4

```
_id: ObjectId('681cbe0fb3e83f0dc5ce1c6c')
name : "Rimon"
address : "GUC"
ID : 123456

_id: ObjectId('681cc18763b4f82c2dac33ec')
name : "Ahmed"
address : "New Cairo"
ID : 654321

_id: ObjectId('681cc20563b4f82c2dac33ee')
name : "Ahmed"
address : "New Cairo"
ID : 333322

_id: ObjectId('681dad10845506074864d4f7')
name : "Marwan"
address : "Giza"
ID : 678678
__v : 0
```

Back-end


Local


lo
React
React
localhost:3001/students
Pretty-print
[
{
"_id": "681cbe0fb3e83f0dc5ce1c6c",
"name": "Rimon",
"address": "GUC",
"ID": 123456
},
{
"_id": "681cc18763b4f82c2dac33ec",
"name": "Ahmed",
"address": "New Cairo",
"ID": 654321
},
{
"_id": "681cc20563b4f82c2dac33ee",
"name": "Ahmed",
"address": "New Cairo",
"ID": 333322
},
{
"_id": "681dad10845506074864d4f7",
"name": "Marwan",
"address": "Giza",
"ID": 678678,
"__v": 0
}
]