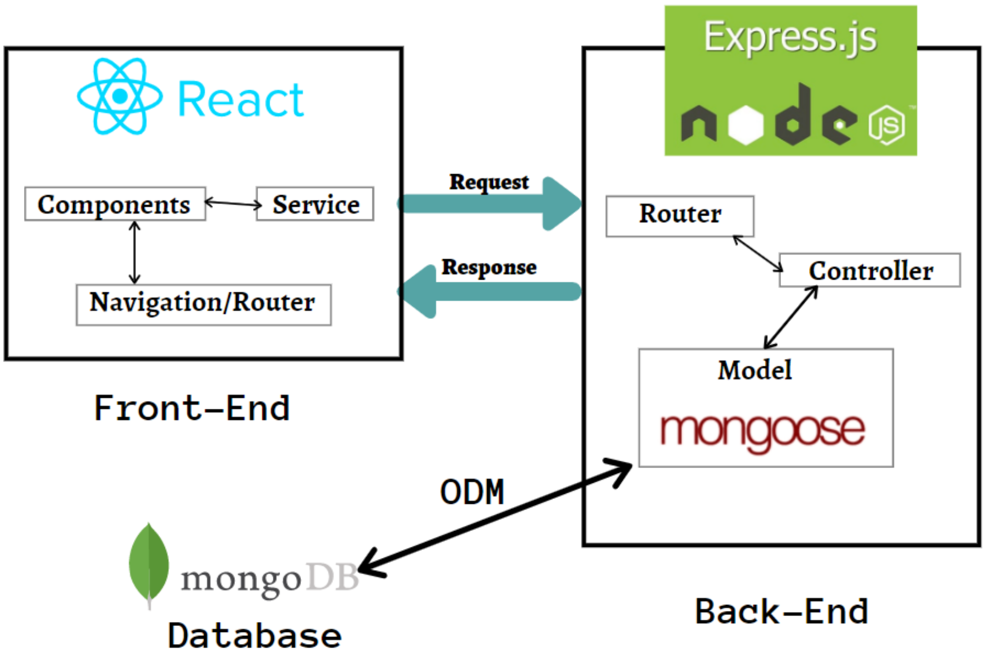
Contemporary Issues in Software Engineering
Semester 2, 2023

Worksheet 3 Ass1A: The MERN Stack

To be submitted by End of Week 4 Tutorial

Theory Overview of MERN Revision: A Simple web application using the MERN tech stack.

The **high-level view of the MERN stack** is the use of four technologies used in tandem and all based on one programming language – JavaScript.



React.js Front End (link to documentation)
The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML. React's strong suit is handling stateful, data-driven interfaces with minimal code, and it has all the bells and whistles you'd expect from a modern web framework: great support for forms, error handling, events, lists, and more. Go to https://reactjs.org for a great interactive tutorial. Also an overview at https://www.youtube.com/watch?v=jeNXbJq5o5g

Express.js and Node.js Server Tier (links to documentation)
Node.js is a cross-platform JavaScript runtime environment that was originally created for Google Chrome and open-sourced in 2008. Node.js is built on Chrome's V8 JavaScript engine and is intended to allow developers to build scalable network applications and execute JavaScript code outside of browsers.
Express.js server-side framework, running inside a Node.js server. Express.js is described as a "fast, unopinionated, minimalist web framework for Node.js,". Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.
MongoDB Database Tier (link to documentation)
If your application stores any data (user profiles, content, comments, uploads, events, etc.), then MongoDB is simple to interface with the React/ Express /Node stack. JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval. For a SaaS application hosted in the cloud, we can use Atlas - a MongoDB instance hosted in the cloud.

By making XML HTTP Requests (XHRs) or GET or POST, PUT, DELETE from your React.js front-end, you can connect to Express.js functions that power your application. Those functions in turn use MongoDB's Node.js drivers, either via callbacks or using Promises, to access and update data in your MongoDB database.

READ THIS: Theory and resources – creating a web app with the MERN stack

Typically **creating a MERN** stack has the following steps (not necessarily in this order):

1. Initialise the project with a Git/GitHub workflow and project folders
2. Create the template code and folders for a front-end Single Page Application (SPA) using React. (e.g "create-react-app" script)
3. Clean up the template – delete unnecessary stuff
4. Create react component that are part of the UI
5. Create menus, navigation bars, forms, inputs, navigation etc for the React front-end
   a. There are react libraries and components
      i. **react-select** for drop-downs
      https://www.positronx.io/react-dropdown-select-tutorial-with-react-select/
      ii. **react-table** for creating table sorting, filtering etc
      https://blog.logrocket.com/complete-guide-building-smart-data-table-react/
   b. css files can be used for formatting UI components
   c. **material-ui** is a popular UI library for React with tables, forms, buttons, dropdowns and so on.
   d. https://material-ui.com/components/tables/
   https://material-ui.com/getting-started/learn/
   e. **formik** is a popular forms library for React, including input validation.
   https://www.youtube.com/watch?v=bMSHmf_ckM8
   f. https://formik.org/docs/tutorial
   g. **bootstrap** is a popular UI library similar to material-ui
   h. often dummy data in the frontend (e.g. in a data.js file in the frontend) is used to test out the frontend with no backend yet.

6. Create "pages" as components using the react-router-dom library
   https://medium.com/@ipenywis/intro-to-react-router-for-beginners-multiple-page-apps-461f4729bd3f
   https://www.freecodecamp.org/news/react-router-tutorial/
7. Create the backend Node/ Express / Nest.js server including some APIs (often Postman or Slumber are used to test the APIs). Often dummy data in the backend are used for testing APIs
8. Connect the frontend UI to the backend server using Express / Nest routes and the **axios** library (or similar) as well as the **cors** library (feel free to look these up and see what their purpose is).
9. Setup the MongoDB database (local or in the cloud). We will set up a MongoDB database instance (a cluster) in the cloud using **MongoDB Atlas**
10. Connect the backend server to the MongoDB database – so we can get and receive data from the database (CRUD requests -Create/Read/Update/Delete). The **mongoose** library is used.
11. Deploy the backend and frontend (in that order!) to production (so users have access) in the cloud with Vercel (or Netlify another cloud service). (i.e. users just type a URL address into the browser to run the app)

This worksheet will bring together all these ideas to create a simple MERN web app.

The worksheet relies on you watching some videos on YouTube (and taking notes) and reading some online tutorials to achieve the desired skills so we can make progress on the Software Empirical Evidence Database (SPEED) app for Peter the PO.

I have chosen videos and tutorials that model the thinking needed to use MERN and reinforce principles. Mostly they are over-simplified but should give you a way forward with the SPEED product we are starting to make.

Remember to **look for PRINCIPLES**! Think about what you are doing.
Be curious, experiment, read about different ways of doing things (there is usually more than one way).

<u>What we need to be able to do for CISE and the SPEED product development</u>

There is a lot that *could* be learned about using the MERN stack to create dynamic web apps. In this course you will need to learn just enough to deliver *some* functionality for the Software Empirical Evidence Database (SPEED)app. This will involve some simple forms and data display as a minimum so that the Product Owner (PO) can test the idea out.

The SPEED application should allow practitioners to be able to find the level of evidence that there is in academic articles that supports or refutes the claimed benefits of different SE practices (eg. what is the level of evidence that using TDD will improve code quality -compared to not using TDD). The SPEED app needs the following functionality to test the idea out:

1. A way to select an SE practice (e.g., TDD) that we want to get the benefits that are claimed about, as well as the level of evidence (eg. strongly supports, weakly refutes) for those claimed benefits.
    a. The user should only be able to select from SE practices that we have evidence of claimed benefits for in our database.
    b. This could be a dropdown list of SE practices that are in our database and that we can select from to return the list of claims and evidence from our database for the selected SE Practice (from the dropdown).
        i. One way to do this is to use the “react-select” library in the React component library.
2. A way to view the level of evidence for each claimed benefit for the SE Practice selected.
    a. It will be useful for the user to see other information about the source of the evidence (the article analysed by the analyst and submitted by a submitter) -such as article title, article authors, journal name, year of publication.
    b. This could be a table of data that can be sorted and filtered.
        i. Each row would be evidence of different claimed benefit from a specific article, each column would be a different piece of info about that claimed benefit and evidence (title, etc)
        ii. The table should be sortable on any column
        iii. The table should be filterable by claimed benefit (and maybe year range of publication, and maybe level of evidence)
        iv. One way to create a table in React is using the react-table library
3. An input form for a submitter (anyone) to submit an article they think is relevant for our evidence database.
    a. The article should be about a SE practice and have some evidence about a claimed benefit(s) for this SE practice.
    b. This article may or may not be accepted (moderator will decide)
    c. If it is accepted then the analyst will read the article and decide on the level of evidence for each claimed benefit in the article and enter that into our SPEED database.

How to learn to do the things we need to do.

This course is about the practices around the coding (CI/CD, TDD, mob/pair programming, planning, retrospectives, code reviews, code quality testing etc), so we do **not** want to spend too much effort becoming expert in JavaScript/Typescript or React or Express/Nest/Node. Rather we just want to learn **enough** so we understand the **principles and can create basic functionality.**

In industry it is common to have to learn a new language or framework or tool – they are changing almost weekly, so this is a useful skill to develop – learning how to learn from diverse sources.

To learn quickly it is useful to watch others model the thinking and coding – **BUT only if you take notes.** Otherwise it is too passive, and you will not retain or learn much.

Then the ONLY way to make this useful so YOU can do it, is to **actually do it yourself !**

This is why we are using worksheets to encourage you to read, watch **and do**.

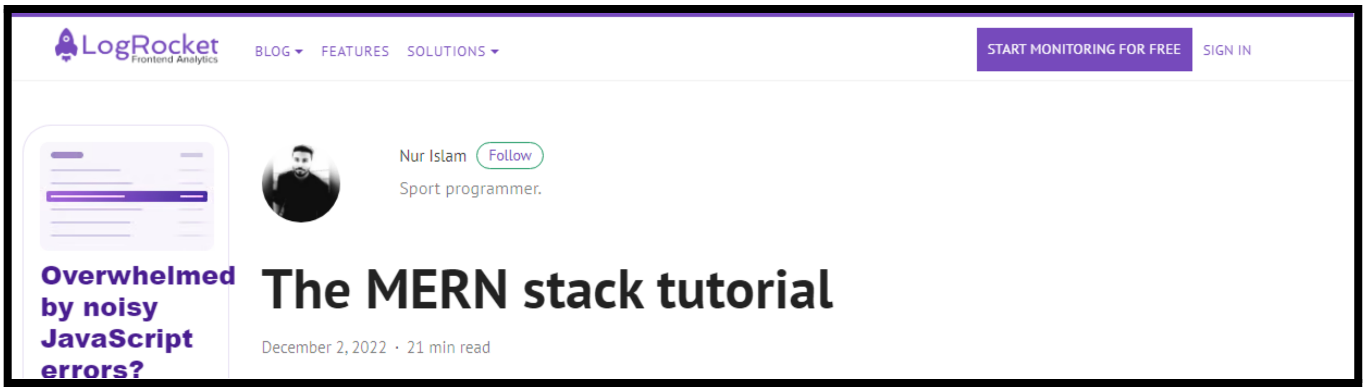
Many people in industry learn from tutorials online or take online courses, and I am sure many of you do this already. It is part of the new way of learning – not just classroom learning, but the sources of information and practice can also be from podcasts, blogs, videos, websites and so on.

The problem is there are worthless, out-of-date, or even plain wrong videos and tutorials on the web. And there are LOTS of videos and tutorials to sift through. The problem becomes finding the good ones. I [Jim Buchan] spend some time finding good ones each year and hopefully you find the ones I direct you to useful.

Worksheet Instructions
Follow the instructions for the following tutorial that creates a simple application to manage information about books. **TAKE SCREENSHOTS AS EVIDENCE** of doing the tutorial. Experiment. Note down anything you learn from doing this tutorial. Try converting the usage of **javascript** into **typescript instead** – this will help you in the next part. A reminder you can always read documentation if you forget how to use typescript here: https://www.typescriptlang.org/

NOTES:

1. You will need to create a local project folder with a Git repo and also synchronise it with a GitHub repo.
2. Remember to take lots of screenshots! Also remember to note down learning and describe challenges and trick you found.
3. Look at the comments in the tutorial – there are some interesting problems people have!
4. Look at the links in the above to get documentation about each technology from the creators
5. Be curious and experiment!
6. This app is not deployed to production in the tutorial– I have included tutorials for deploying a MERN app to Vercel in the next section
7. See if you can deploy this app to Vercel


LogRocket Frontend Analytics
BLOG
FEATURES
SOLUTIONS
START MONITORING FOR FREE
SIGN IN
Overwhelmed by noisy JavaScript errors?
Nur Islam
Follow
Sport programmer.

The MERN stack tutorial

December 2, 2022 · 21 min read

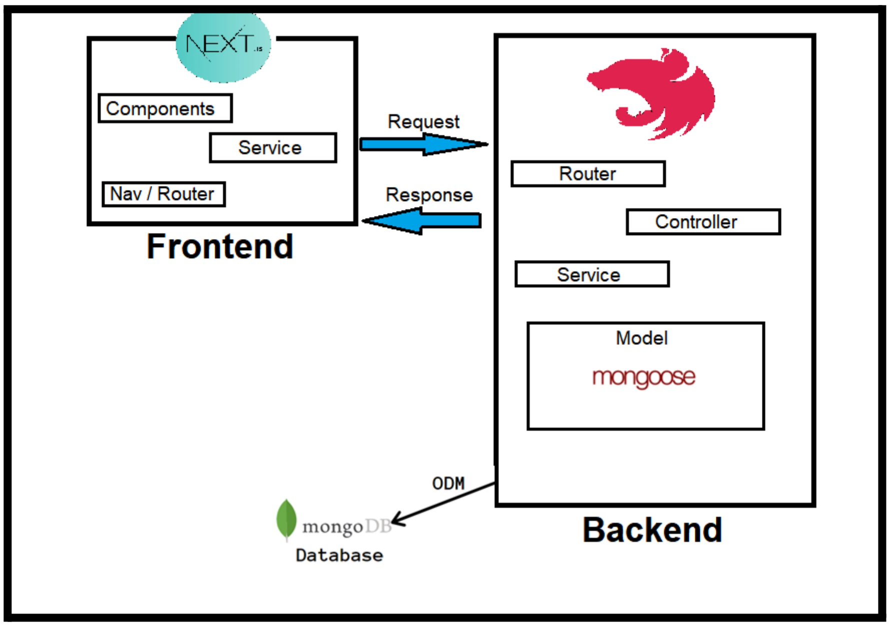
Tutorial:
https://blog.logrocket.com/mern-stack-tutorial/

Additional challenge - Adapt the MERN Tutorial to MNNN (MongoDB, Next.js, Nest.js, Node.js) & Deploy to Vercel

This step is optional but will greatly assist you for the week 4 exercise and the assignment.

Modern technologies such as nest.js and now Next.js are frameworks that keep innovating the development field. This additional challenge will assist you for the assignment and help you understand how to now use Next.js with typescript.

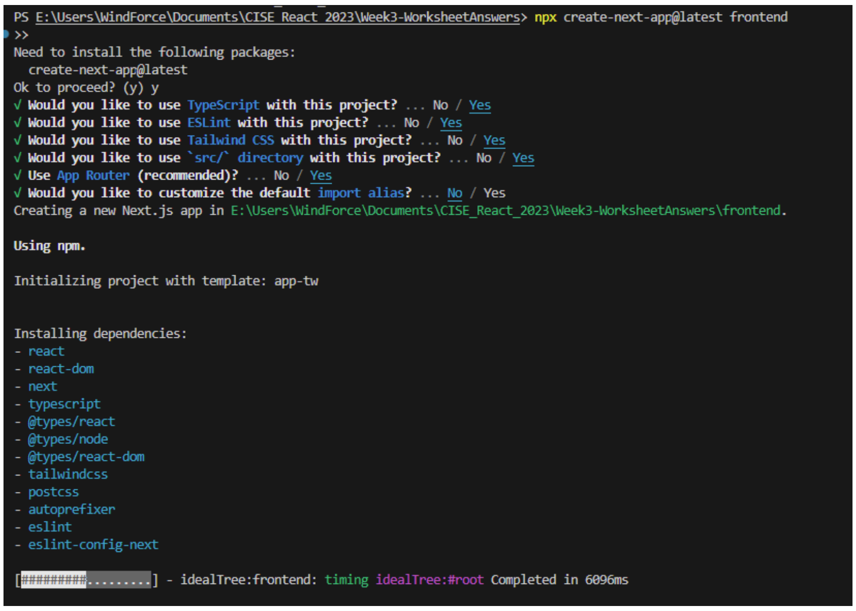
You can read about Next.js here: https://nextjs.org/docs



Before starting:

I always suggest you get yourself familiar with the technologies you are using. I suggest you first run this command:
> npx create-next-app@latest frontend-week3

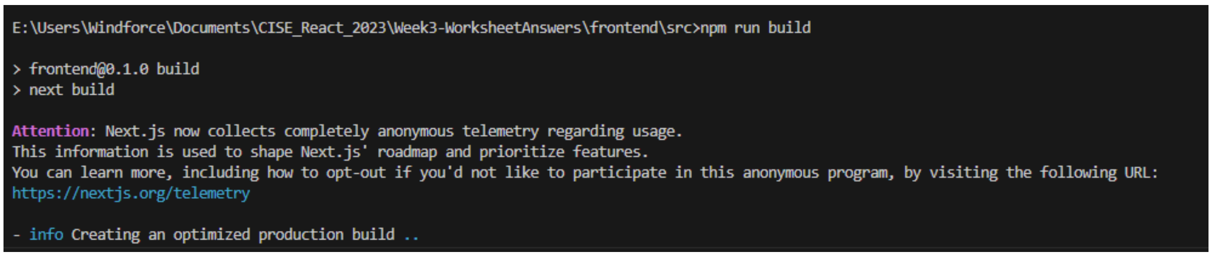
Which will prompt you to create a Next application follow these selections:



```
PS E:\Users\WindForce\Documents\CISE React 2023\Week3-WorksheetAnswers> npx create-next-app@latest frontend
>>
Need to install the following packages:
  create-next-app@latest
Ok to proceed? (y) y
√ Would you like to use TypeScript with this project? ... No / Yes
√ Would you like to use ESLint with this project? ... No / Yes
√ Would you like to use Tailwind CSS with this project? ... No / Yes
√ Would you like to use `src/` directory with this project? ... No / Yes
√ Use App Router (recommended)? ... No / Yes
√ Would you like to customize the default import alias? ... No / Yes
Creating a new Next.js app in E:\Users\WindForce\Documents\CISE_React_2023\Week3-WorksheetAnswers\frontend.

Using npm.

Initializing project with template: app-tw


Installing dependencies:
- react
- react-dom
- next
- typescript
- @types/react
- @types/node
- @types/react-dom
- tailwindcss
- postcss
- autoprefixer
- eslint
- eslint-config-next

[#########.........] - idealTree:frontend: timing idealTree:#root Completed in 6096ms
```

Before you can use the command npm start you must first run this command:
> npm run build

Do you know why?

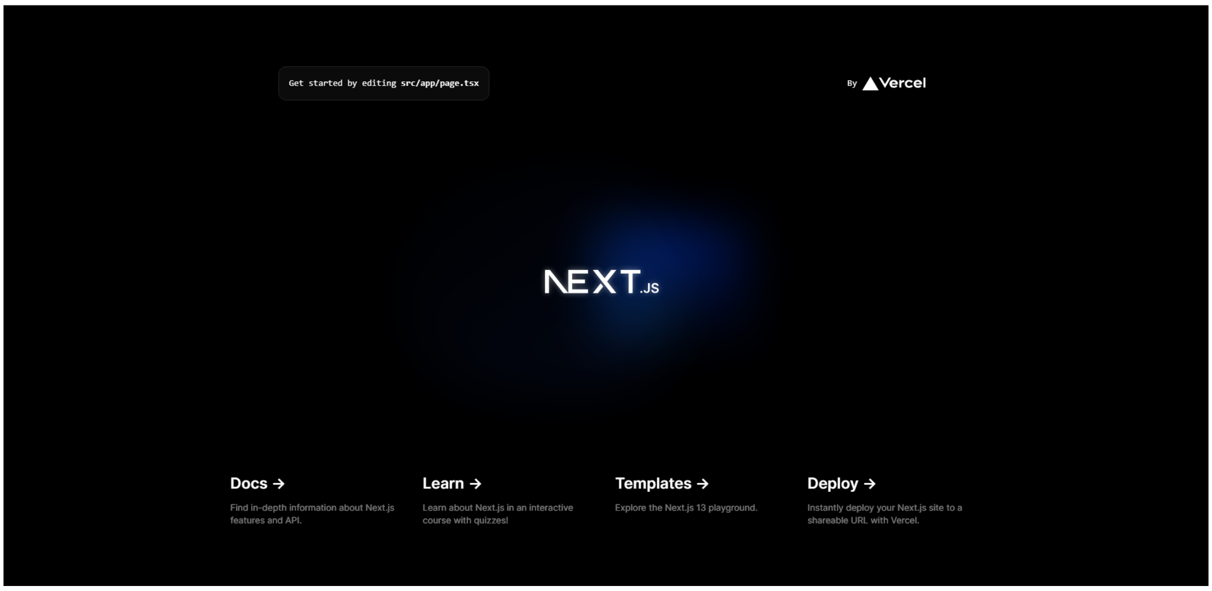


```
E:\Users\Windforce\Documents\CISE_React_2023\Week3-WorksheetAnswers\frontend\src>npm run build

> frontend@0.1.0 build
> next build

Attention: Next.js now collects completely anonymous telemetry regarding usage.
This information is used to shape Next.js' roadmap and prioritize features.
You can learn more, including how to opt-out if you'd not like to participate in this anonymous program, by visiting the following URL:
https://nextjs.org/telemetry

- info Creating an optimized production build ..
```

Now you will be able to use:
> npm start

Note you can also run a development build with command without needing to *npm run build* first
> npm dev

You should see this as the default display page – Click the links **EXPLORE** understand what this technology is (***you will be using Vercel to deploy, and this page provides helpful resources to learn from***)



Get started by editing src/app/page.tsx

By Vercel

NEXT.JS

Docs ->
Find in-depth information about Next.js features and API.

Learn ->
Learn about Next.js in an interactive course with quizzes!

Templates ->
Explore the Next.js 13 playground.

Deploy ->
Instantly deploy your Next.js site to a shareable URL with Vercel.

Helpful Links:

1. https://vercel.com/docs/getting-started-with-vercel

2.
https://www.mongodb.com/basics/technology-stack#:~:text=A%20technology%20stack%20is%20a,together%20to%20build%20any%20application.

3. https://dev.to/yakovlev_alexey/creating-a-project-with-nestjs-nextjs-3i1i

4. https://www.mongodb.com/developer/languages/javascript/nextjs-with-mongodb/

Create a shared GitHub account (extension):

I suggest that teams make a shared GitHub account which will be used to also sign up to Vercel.

You can easily sign up to GitHub here:

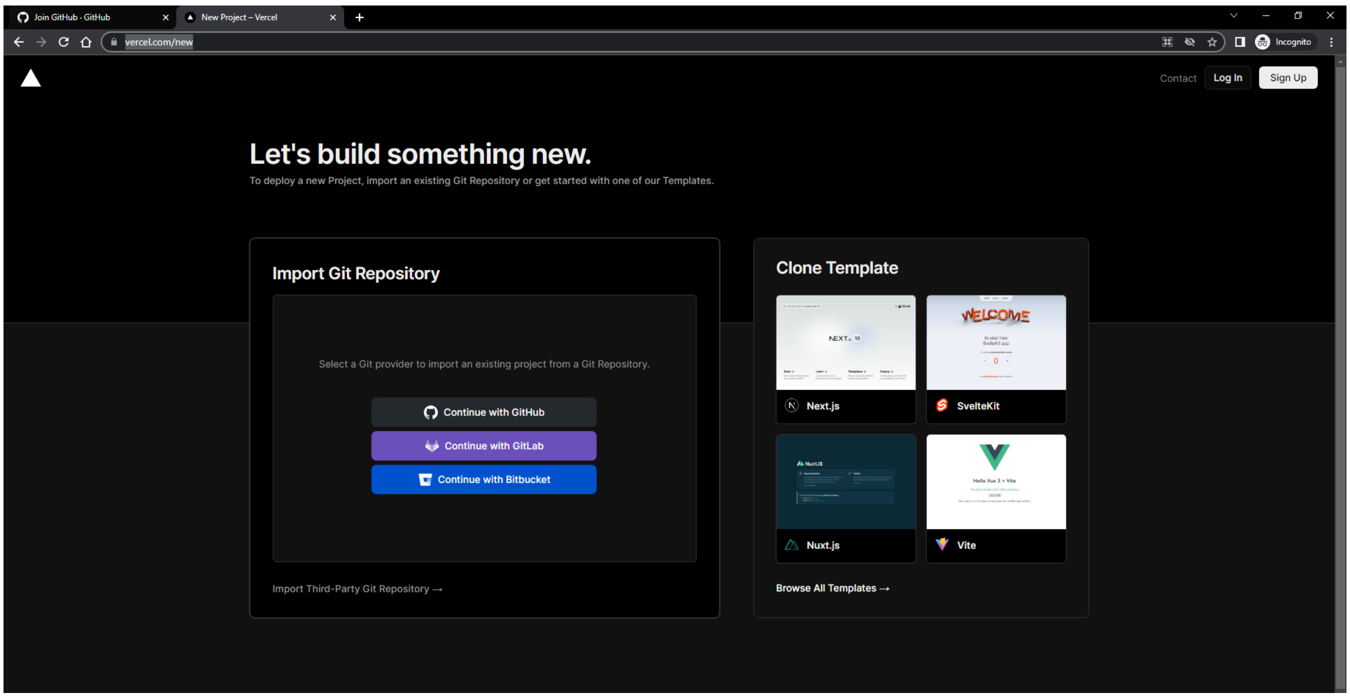
https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home

This is so during the assignment if a team member leaves (or falls apart) the present members still have access to the repo and can continue development.

Make sure you create the repo ***using the shared GitHub account*** and invite all other members.

You can then easily import the repository in Vercel and deploy the project using one hobby account and sign up using the shared GitHub account.

https://vercel.com/new



Let's build something new.

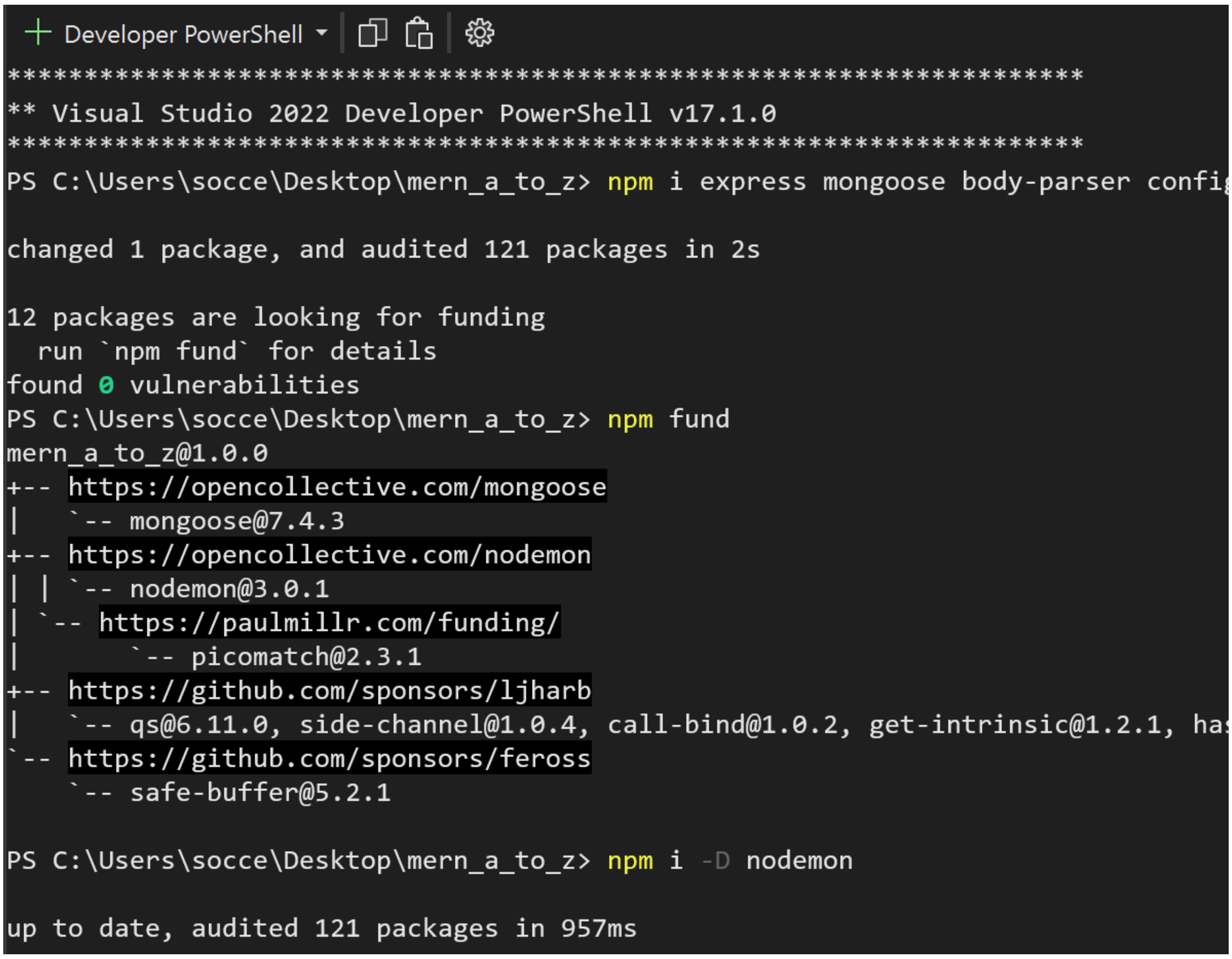
To deploy a new Project, import an existing Git Repository or get started with one of our Templates.

Select a Git provider to import an existing project from a Git Repository.

Evidence to Be uploaded to Canvas as well as Checked off by the Tutor by Week 5

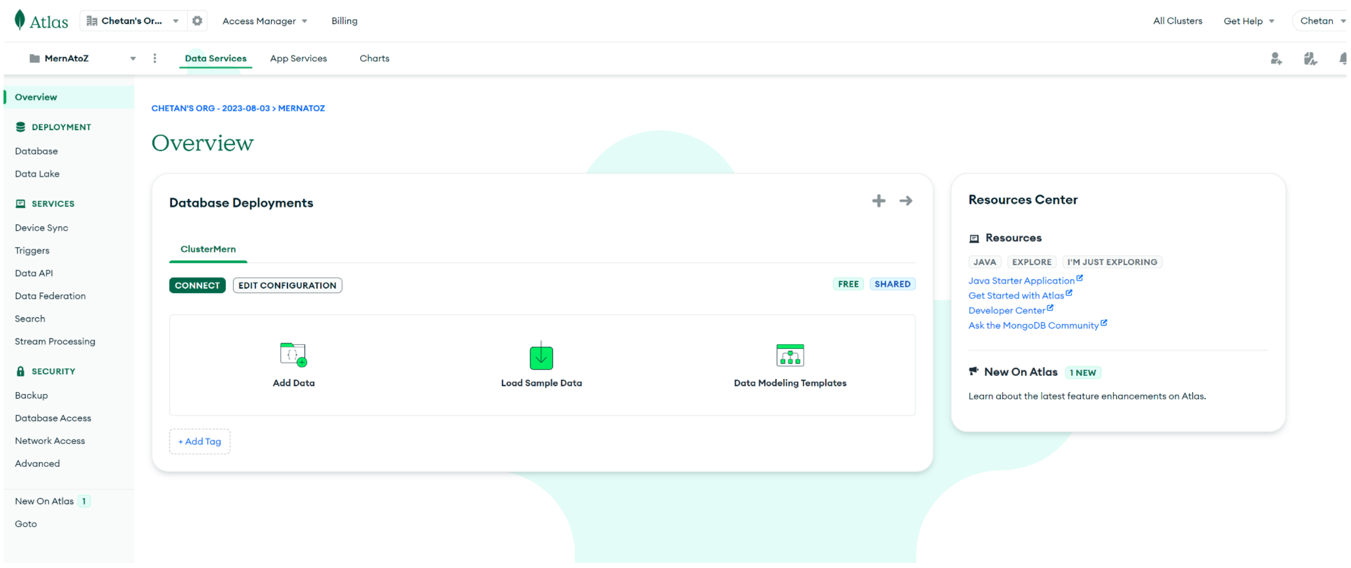
1. Screen shots of going through each step of the MERN stack tutorial that creates the Book app. Include some written notes beside the screenshots that explain what is being done and anything new you learned or any tricks or mistakes you made.

Add screen shots and notes here (include screen shot of GitHub – code for frontend and backend; screen shot of the package.json file; screen shot of the MongoDB database created; screen shot of the app running locally.

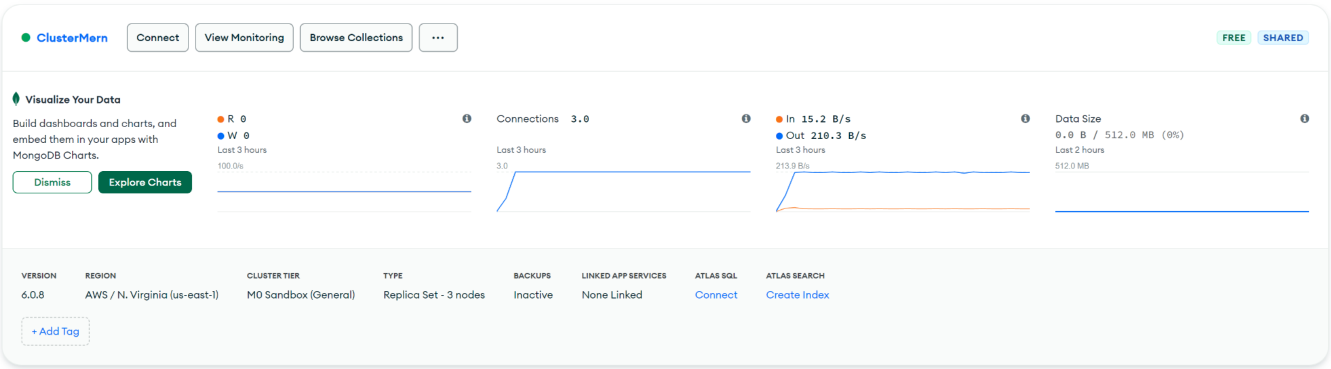


```
**********************************************************************
** Visual Studio 2022 Developer PowerShell v17.1.0
**********************************************************************
PS C:\Users\socce\Desktop\mern_a_to_z> npm i express mongoose body-parser confi

changed 1 package, and audited 121 packages in 2s

12 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\socce\Desktop\mern_a_to_z> npm fund
mern_a_to_z@1.0.0
+-- https://opencollective.com/mongoose
|   `-- mongoose@7.4.3
+-- https://opencollective.com/nodemon
| | `-- nodemon@3.0.1
| `-- https://paulmillr.com/funding/
|       `-- picomatch@2.3.1
+-- https://github.com/sponsors/ljharb
|   `-- qs@6.11.0, side-channel@1.0.4, call-bind@1.0.2, get-intrinsic@1.2.1, ha
`-- https://github.com/sponsors/feross
    `-- safe-buffer@5.2.1

PS C:\Users\socce\Desktop\mern_a_to_z> npm i -D nodemon

up to date, audited 121 packages in 957ms
```

installing the required packages.



creating mongoDB database



Visualize Your Data

Build dashboards and charts, and embed them in your apps with MongoDB Charts.

R 0
W 0
Last 3 hours
100.0/s

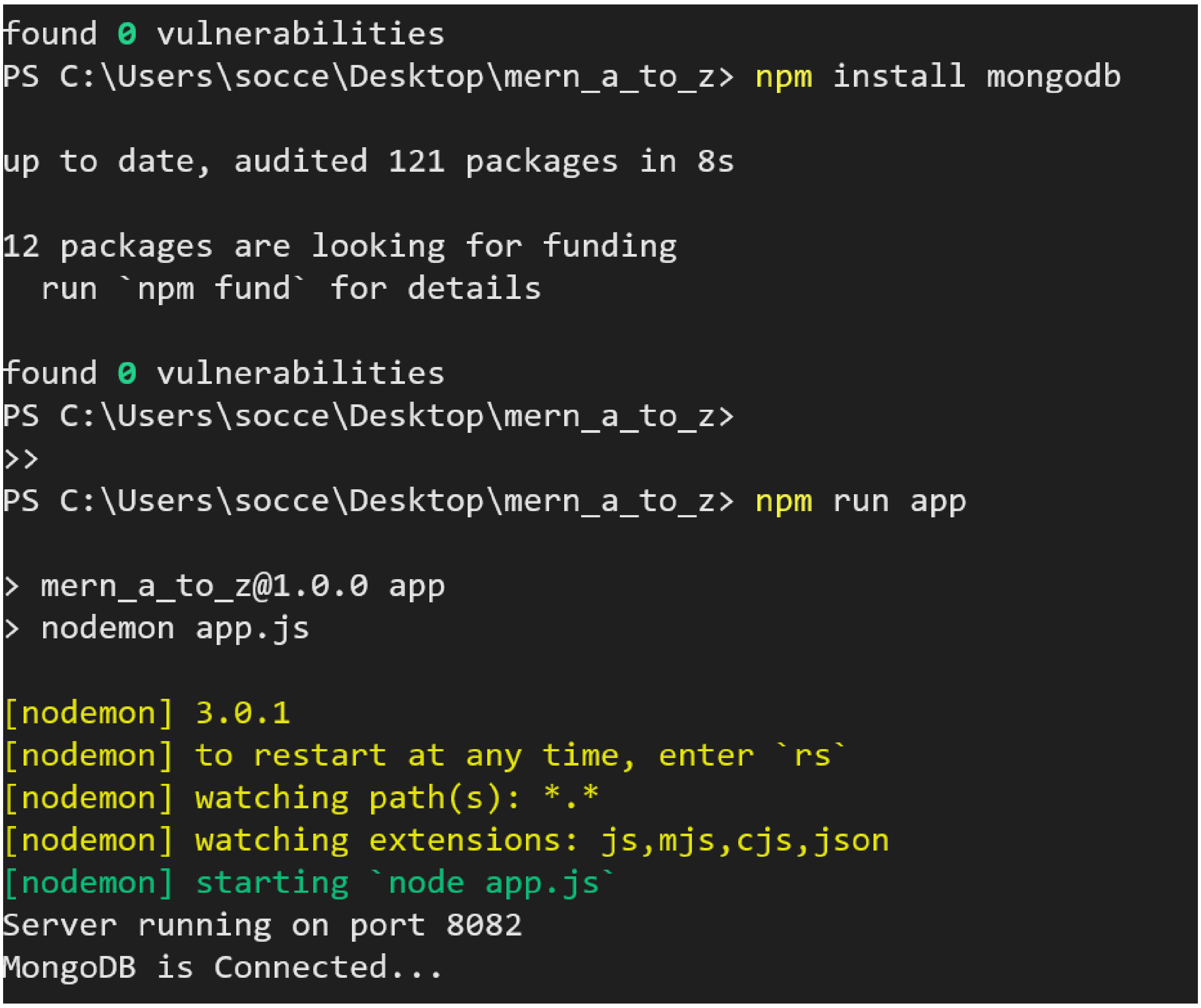
Connections 3.0
Last 3 hours
3.0

In 15.2 B/s
Out 210.3 B/s
Last 3 hours
213.9 B/s

Data Size
0.0 B / 512.0 MB (0%)
Last 2 hours
512.0 MB

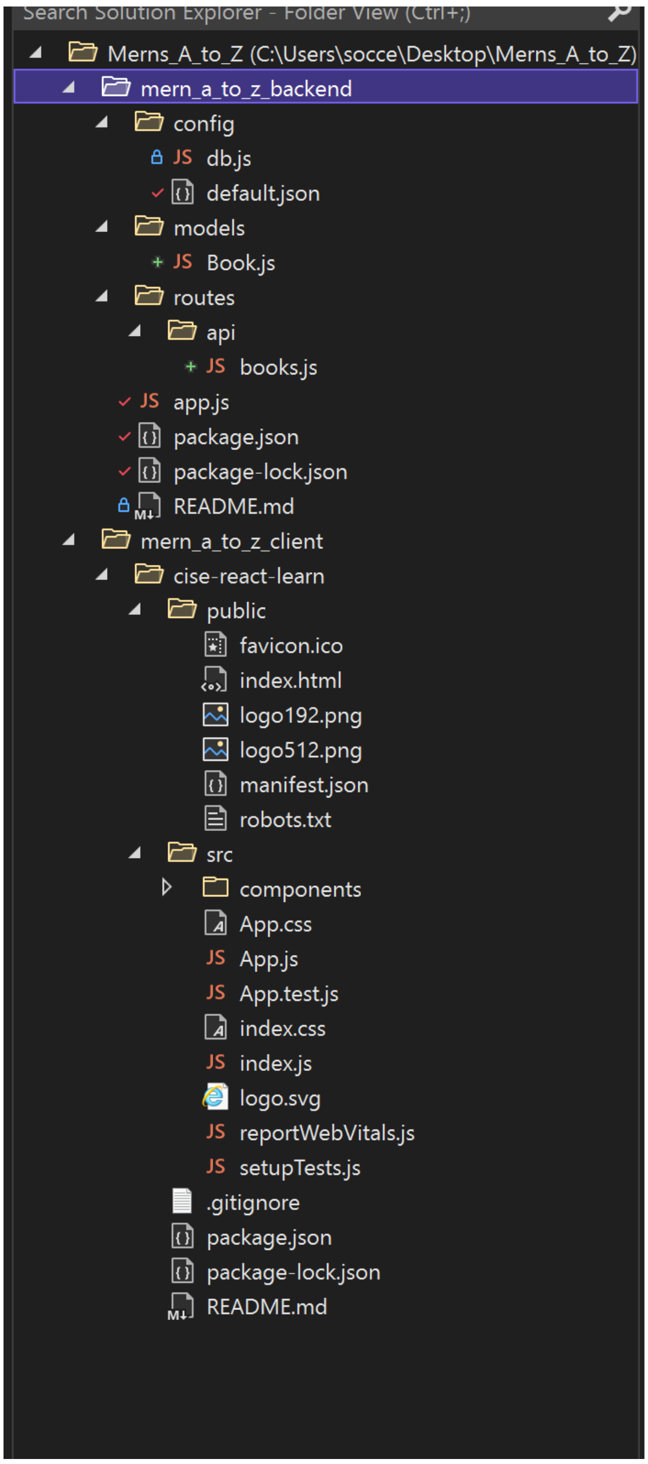
VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SQL	ATLAS SEARCH
6.0.8	AWS / N. Virginia (us-east-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Connect	Create Index

here is the server from the database up and running with proof that there has been connections to it



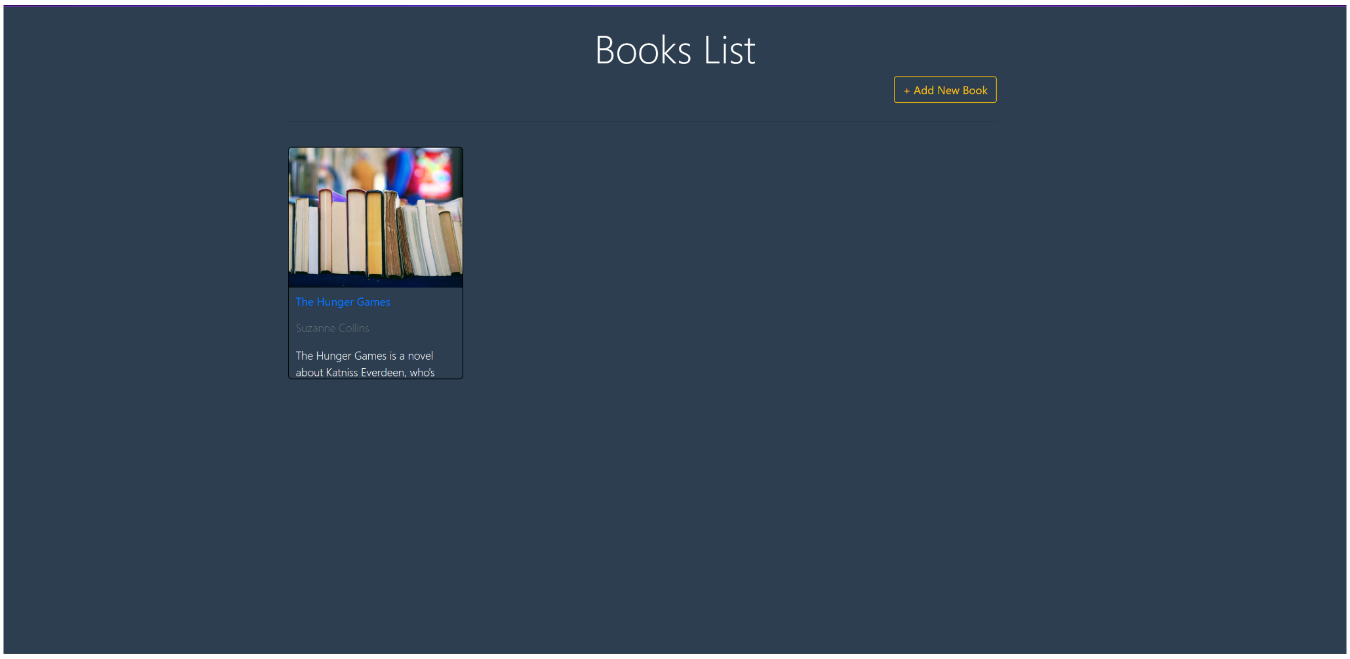
```
found 0 vulnerabilities
PS C:\Users\socce\Desktop\mern_a_to_z> npm install mongodb

up to date, audited 121 packages in 8s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\socce\Desktop\mern_a_to_z>
>>
PS C:\Users\socce\Desktop\mern_a_to_z> npm run app

> mern_a_to_z@1.0.0 app
> nodemon app.js

[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
Server running on port 8082
MongoDB is Connected...
```

when creating the front end I got many errors on the terminal related to npm and the local host wasn't working either. There was also an error in the back end app.js code where there was a 's' at the end of a word which caused an error for the "npm run app". After resolving that issue the backend and front end started working perfectly.

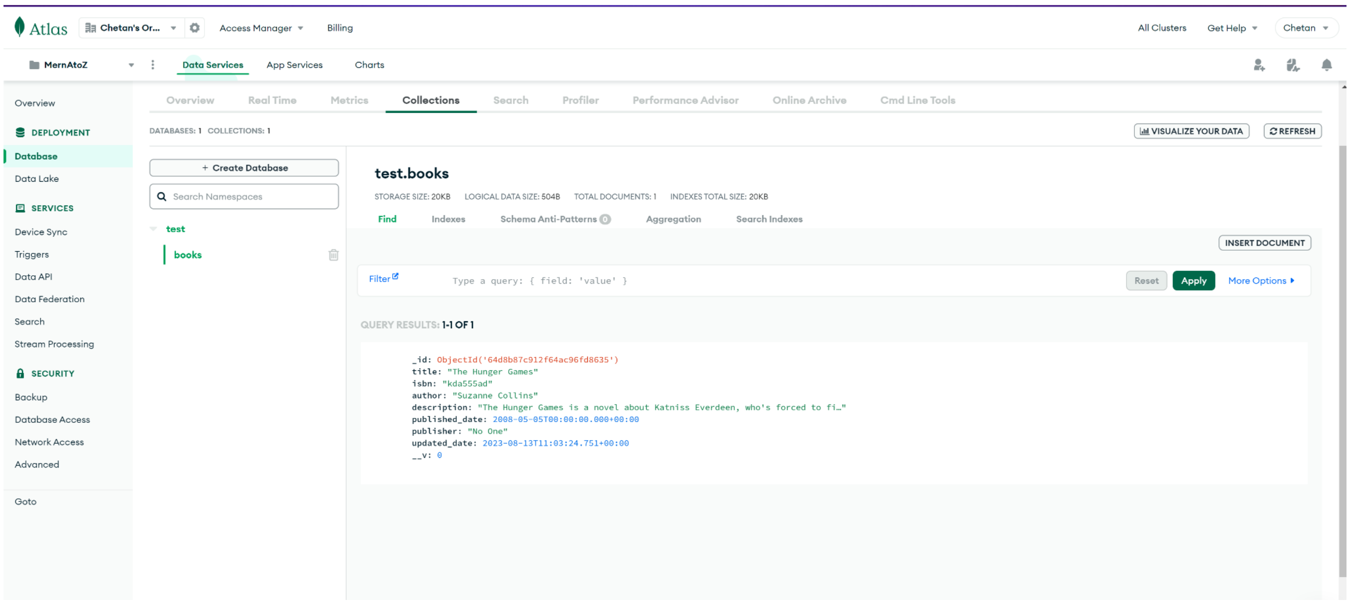


```
Search Solution Explorer - Folder View (Ctrl+;)
◢ Merns_A_to_Z (C:\Users\socce\Desktop\Merns_A_to_Z)
  ◢ mern_a_to_z_backend
    ◢ config
        db.js
        default.json
    ◢ models
        Book.js
    ◢ routes
      ◢ api
          books.js
      app.js
      package.json
      package-lock.json
      README.md
  ◢ mern_a_to_z_client
    ◢ cise-react-learn
      ◢ public
          favicon.ico
          index.html
          logo192.png
          logo512.png
          manifest.json
          robots.txt
      ◢ src
        ▷ components
          App.css
          App.js
          App.test.js
          index.css
          index.js
          logo.svg
          reportWebVitals.js
          setupTests.js
        .gitignore
        package.json
        package-lock.json
        README.md
```

here is a list of all the files in this project



Here is the final successful connection where the localhost working and I have created a hunger games book.



DATABASES: 1 COLLECTIONS: 1

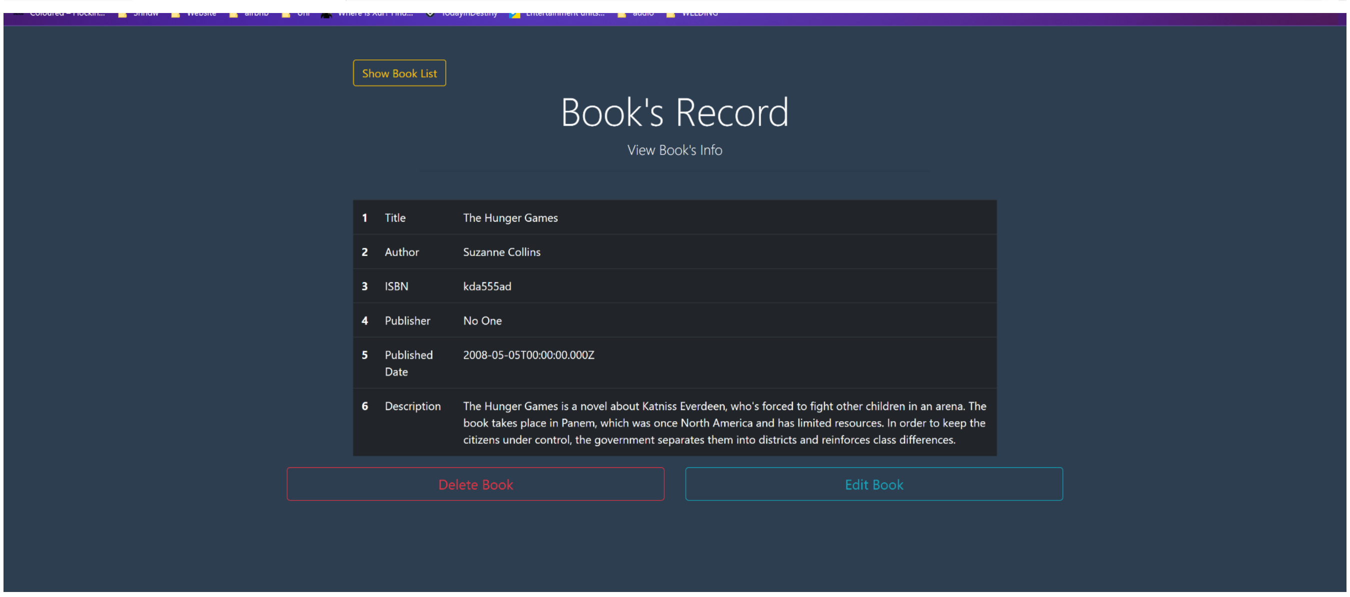
test.books

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 504B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

```
Type a query: { field: 'value' }
```

QUERY RESULTS: 1-1 OF 1

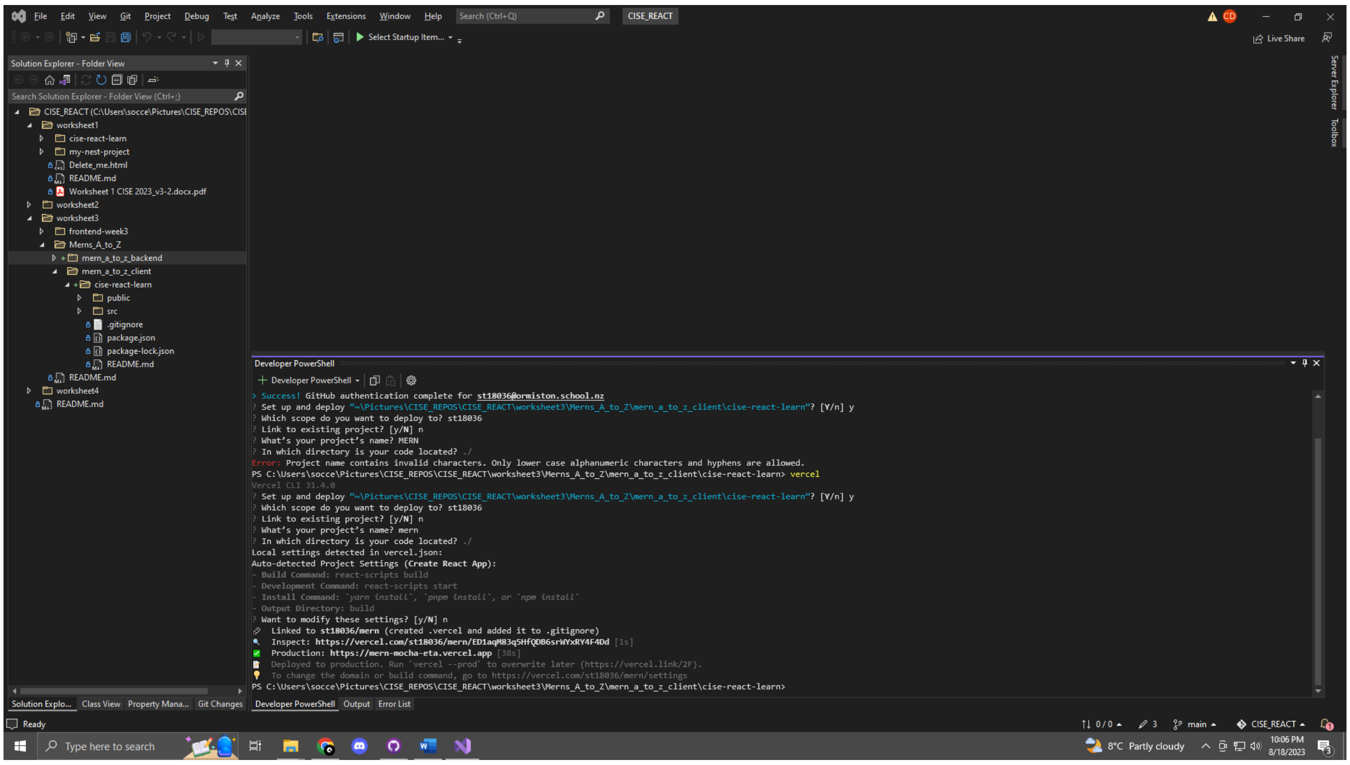
```
_id: ObjectId('64d8b87c912f64ac96fd8635')
title: "The Hunger Games"
isbn: "kda555ad"
author: "Suzanne Collins"
description: "The Hunger Games is a novel about Katniss Everdeen, who's forced to fi…"
published_date: 2008-05-05T00:00:00.000+00:00
publisher: "No One"
updated_date: 2023-08-13T11:03:24.751+00:00
__v: 0
```



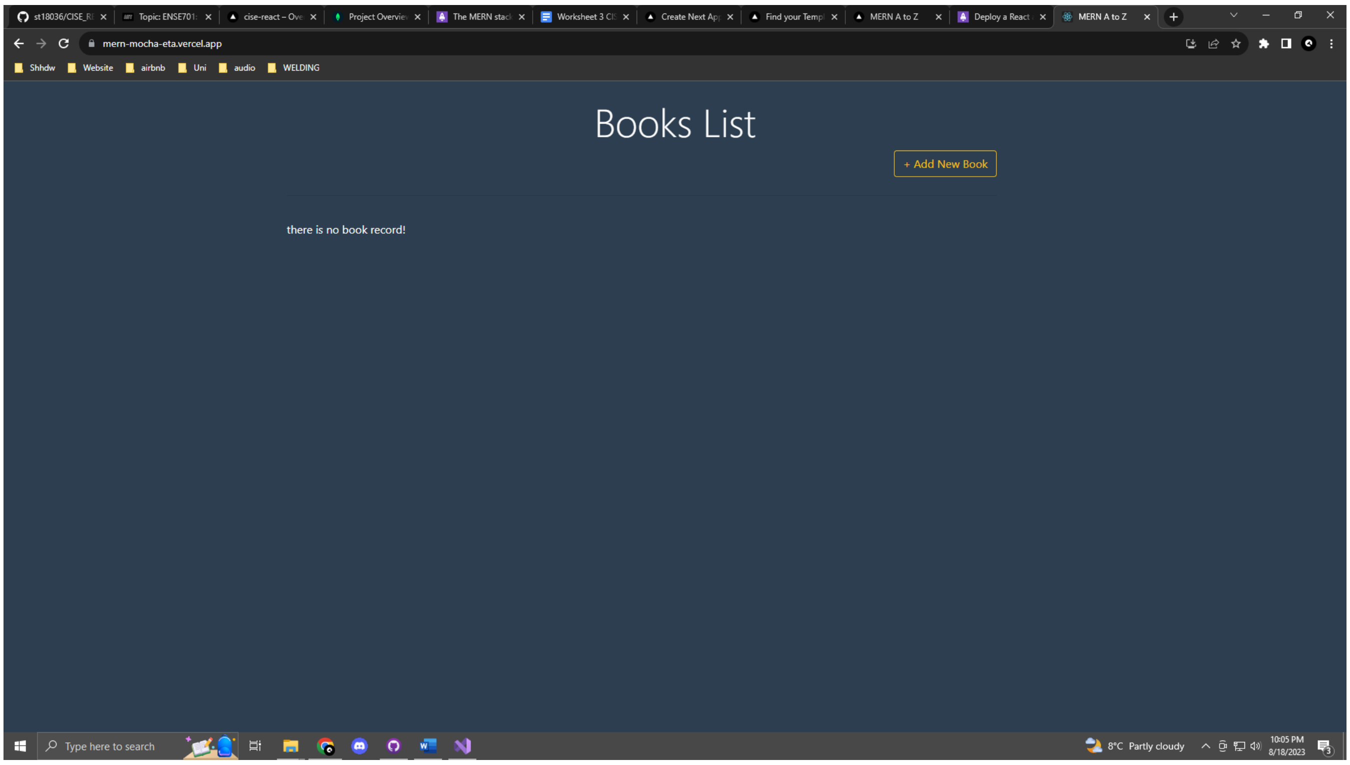
1	Title	The Hunger Games
2	Author	Suzanne Collins
3	ISBN	kda555ad
4	Publisher	No One
5	Published Date	2008-05-05T00:00:00.000Z
6	Description	The Hunger Games is a novel about Katniss Everdeen, who's forced to fight other children in an arena. The book takes place in Panem, which was once North America and has limited resources. In order to keep the citizens under control, the government separates them into districts and reinforces class differences.

2. The Vercel URL of your MERN app which has successfully been deployed to Vercel
   https://mern-mocha-eta.vercel.app

3. A screenshot of the app running on a browser being served by Vercel. Describe any new things you learned or any issues you had.



```
> Success! GitHub authentication complete for st18036@ormiston.school.nz
? Set up and deploy "~\Pictures\CISE_REPOS\CISE_REACT\worksheet3\Merns_A_to_Z\mern_a_to_z_client\cise-react-learn"? [Y/n] y
? Which scope do you want to deploy to? st18036
? Link to existing project? [y/N] n
? What's your project's name? MERN
? In which directory is your code located? ./
Error: Project name contains invalid characters. Only lower case alphanumeric characters and hyphens are allowed.
PS C:\Users\socce\Pictures\CISE_REPOS\CISE_REACT\worksheet3\Merns_A_to_Z\mern_a_to_z_client\cise-react-learn> vercel
Vercel CLI 31.4.0
? Set up and deploy "~\Pictures\CISE_REPOS\CISE_REACT\worksheet3\Merns_A_to_Z\mern_a_to_z_client\cise-react-learn"? [Y/n] y
? Which scope do you want to deploy to? st18036
? Link to existing project? [y/N] n
? What's your project's name? mern
? In which directory is your code located? ./
Local settings detected in vercel.json:
Auto-detected Project Settings (Create React App):
- Build Command: react-scripts build
- Development Command: react-scripts start
- Install Command: `yarn install`, `pnpm install`, or `npm install`
- Output Directory: build
? Want to modify these settings? [y/N] n
🔗  Linked to st18036/mern (created .vercel and added it to .gitignore)
🔍  Inspect: https://vercel.com/st18036/mern/ED1aqM83q5HfQDB6srWYxRY4F4Dd [1s]
✅  Production: https://mern-mocha-eta.vercel.app [38s]
📝  Deployed to production. Run `vercel --prod` to overwrite later (https://vercel.link/2F).
💡  To change the domain or build command, go to https://vercel.com/st18036/mern/settings
PS C:\Users\socce\Pictures\CISE_REPOS\CISE_REACT\worksheet3\Merns_A_to_Z\mern_a_to_z_client\cise-react-learn>
```



4. What is the difference in purpose between a Route in React and a Route in Nest?
   - A route in React helps display different parts of your website or web application based on the URL, allowing you to create dynamic and interactive user interfaces without full page reloads.
   - A route in Nest defines how your server responds to different URLs or paths with specific actions or data. It's about handling incoming requests from clients, processing data, and returning appropriate responses.

5. What react library options are there to create each of the following:
   a. **Drop down widget** (such as needed to select the SE practice in the SPEED app)
      - react-select to create a dropdown list
   b. An **input form** (such as the article submission form needed for the SPEED app.
      - the best and most popular choice is formik

   c. A data **display table** (such as needed to display the evidence level and claims for the SPEED app).
      - eact-native-data-table is the library use to creating data display tables

OPTIONAL EXTRAS – EVIDENCE (to make sure you get an A to A+)!

1. Show screenshots and write explanations showing you were able to (or tried to) use GitHub actions to automatically deploy your Book app using Vercel.

2. Show screenshots and written explanations of setting up the linter and prettier with VS Code and create-react-app, using AirBnB rules, and including using “fix” with the linter

3. Show screenshots and written explanations of setting up “husky” and “lint-staged” to automatically lint locally any files you stage with “git add”.