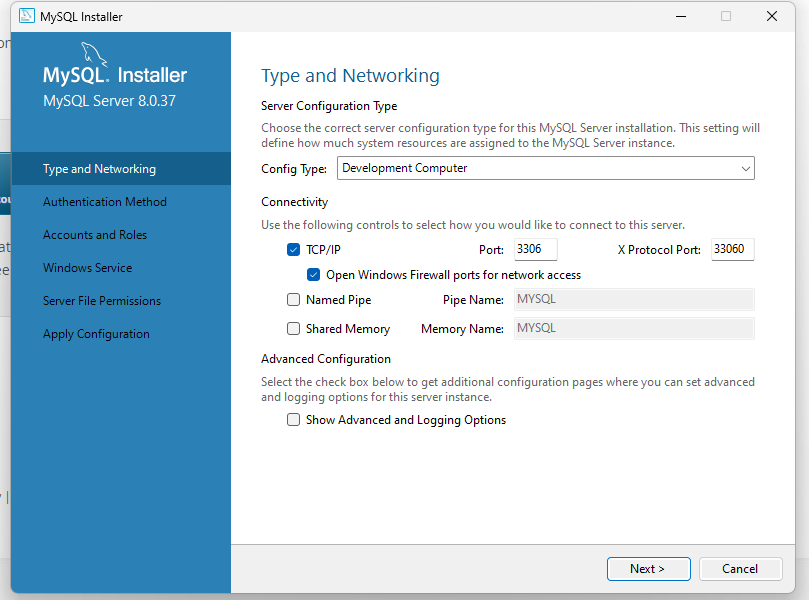
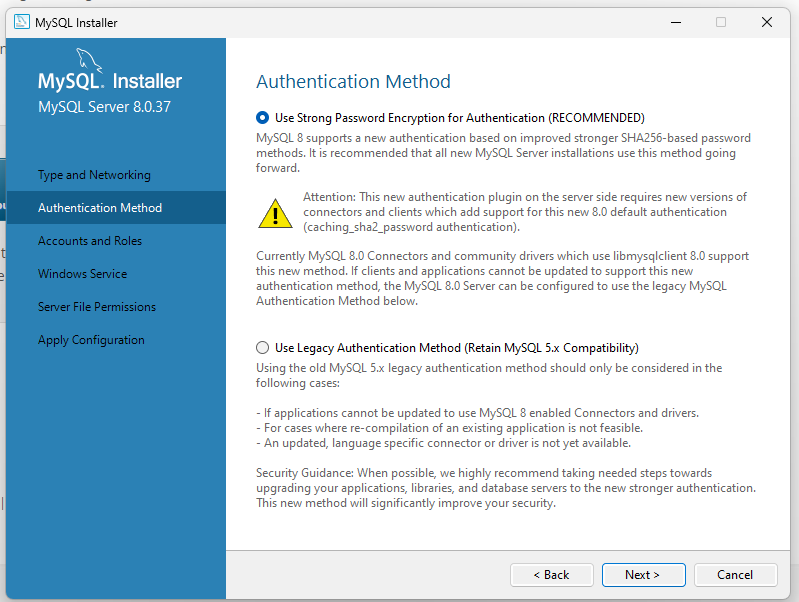
**Key information**

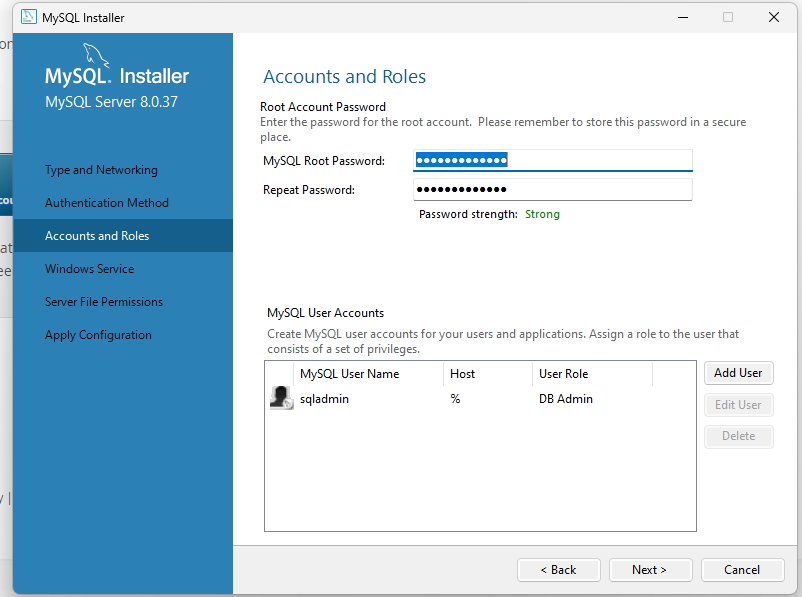
ssh -i "SG-ken-predator.pem" [ubuntu@ec2-13-213-11-59.ap-southeast-1.compute.amazonaws.com](mailto:ubuntu@ec2-13-213-11-59.ap-southeast-1.compute.amazonaws.com)

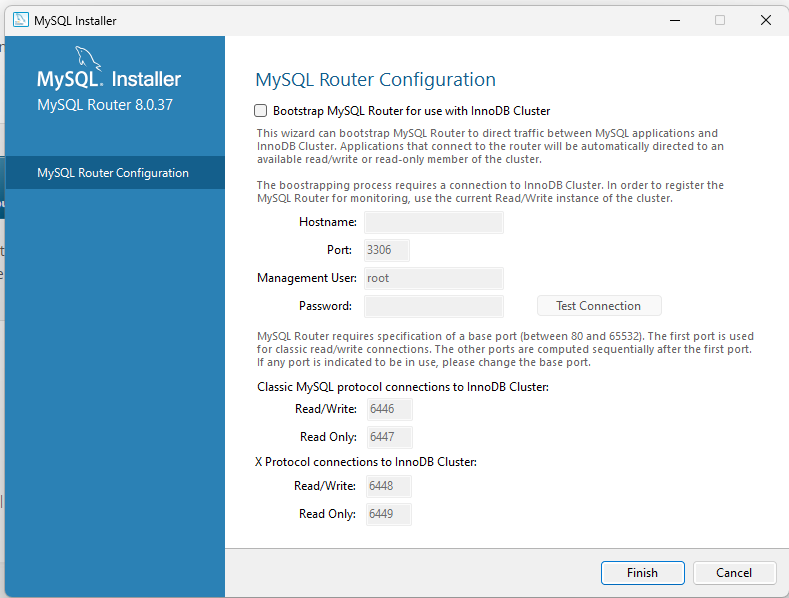
React Node.js MySQL CRUD Tutorial for Beginners

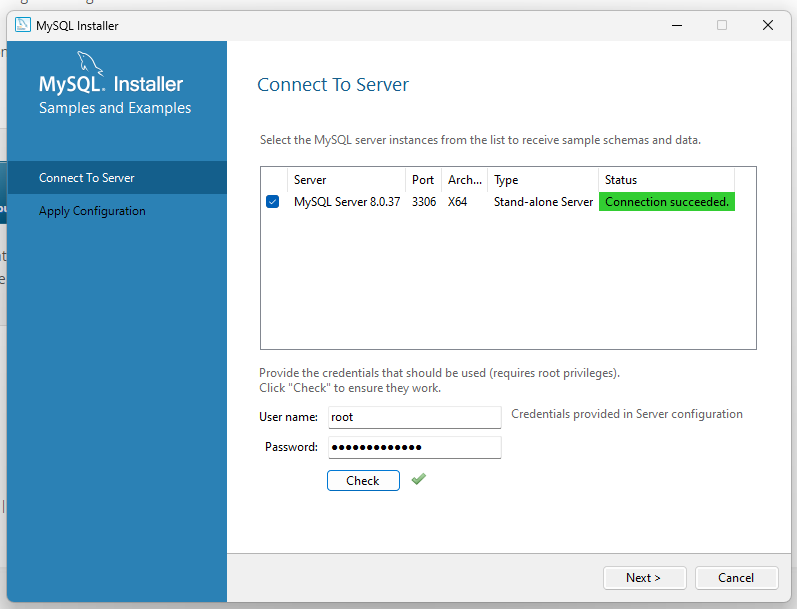
<https://www.youtube.com/watch?v=fPuLnzSjPLE&list=PLXE2Bj4edhg56h_EB2JbUKPfknkPo_Jyv>

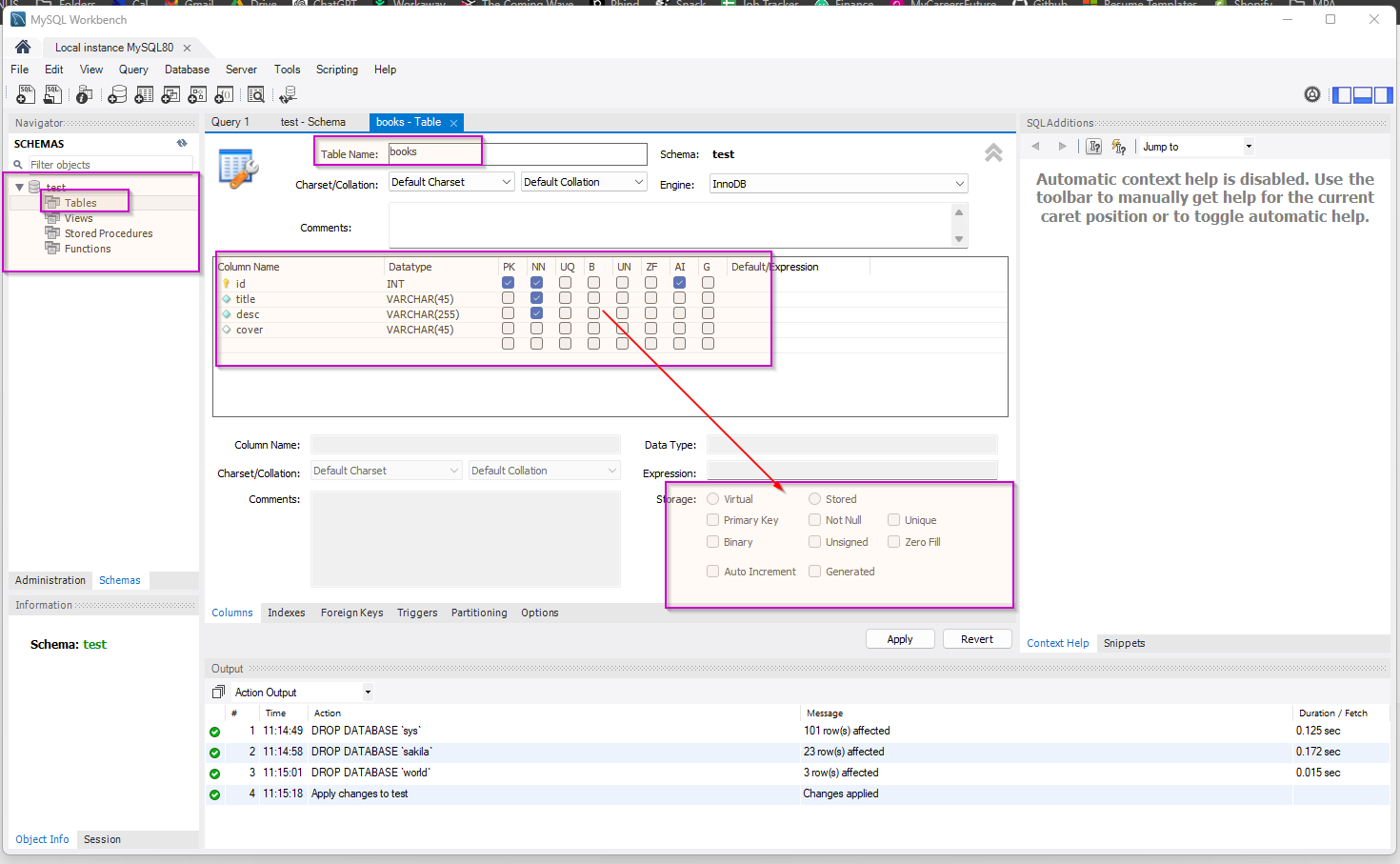


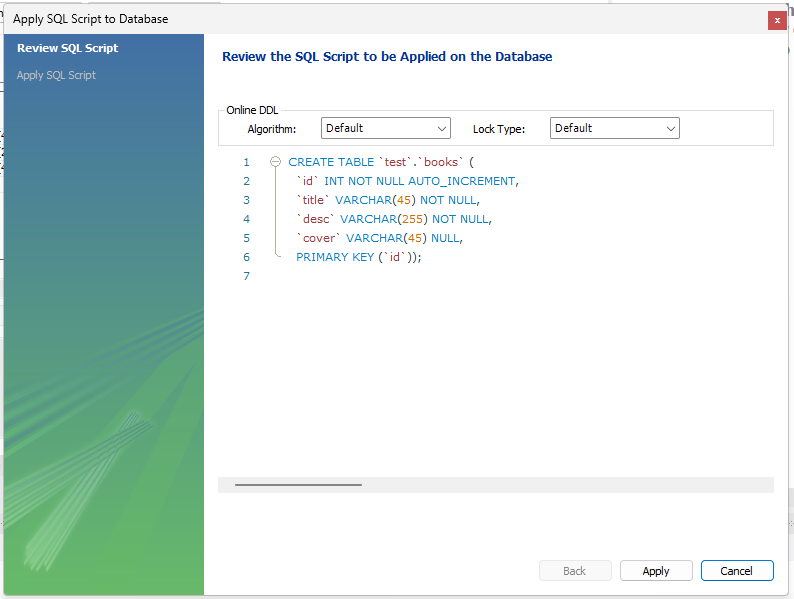


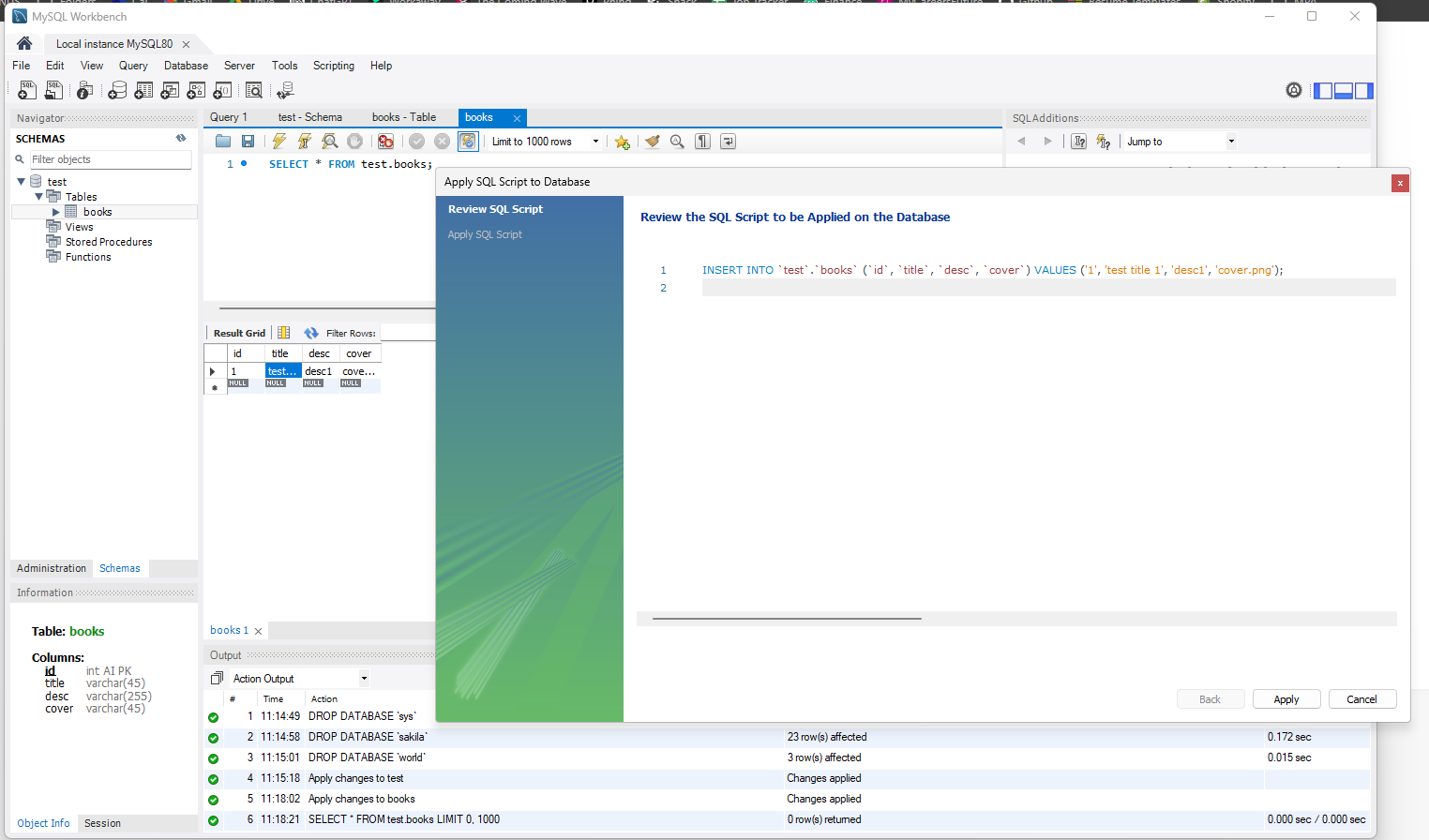


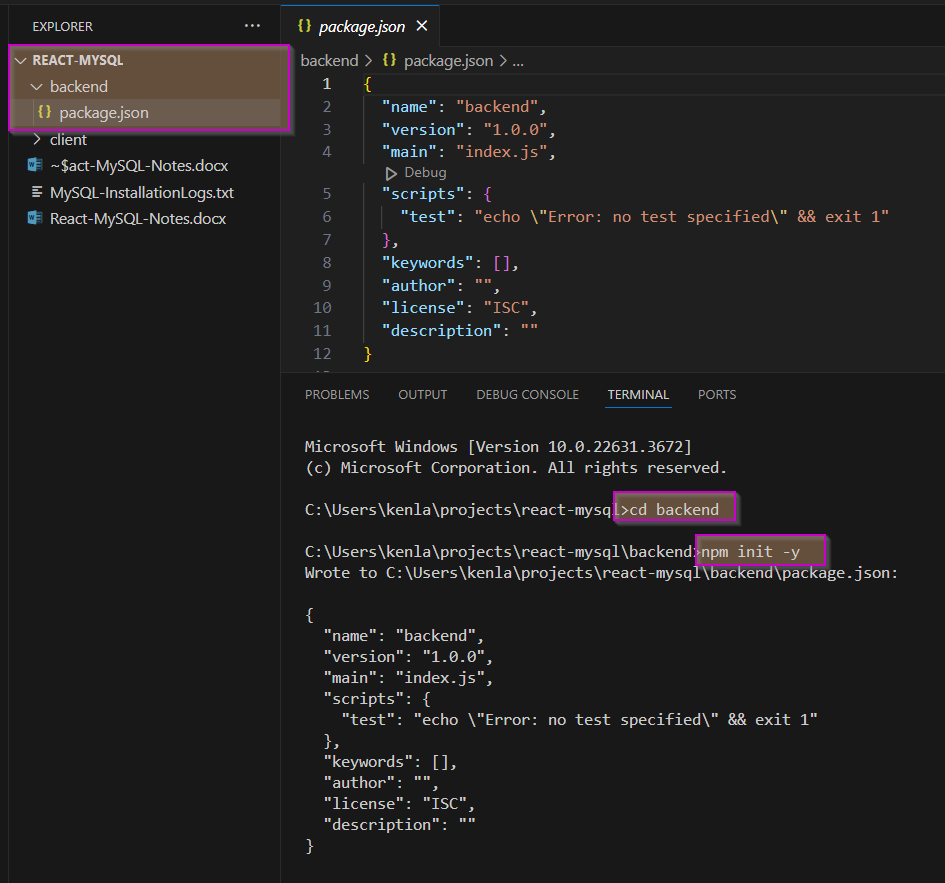


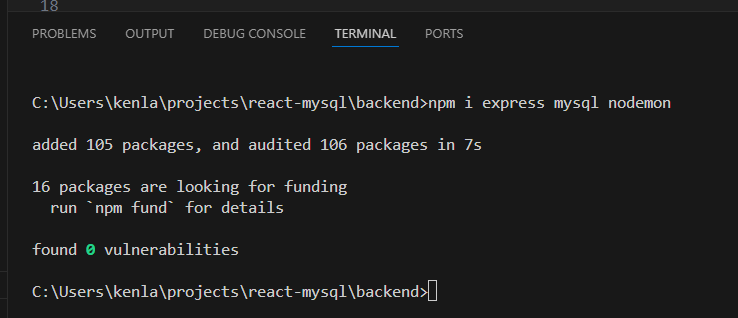


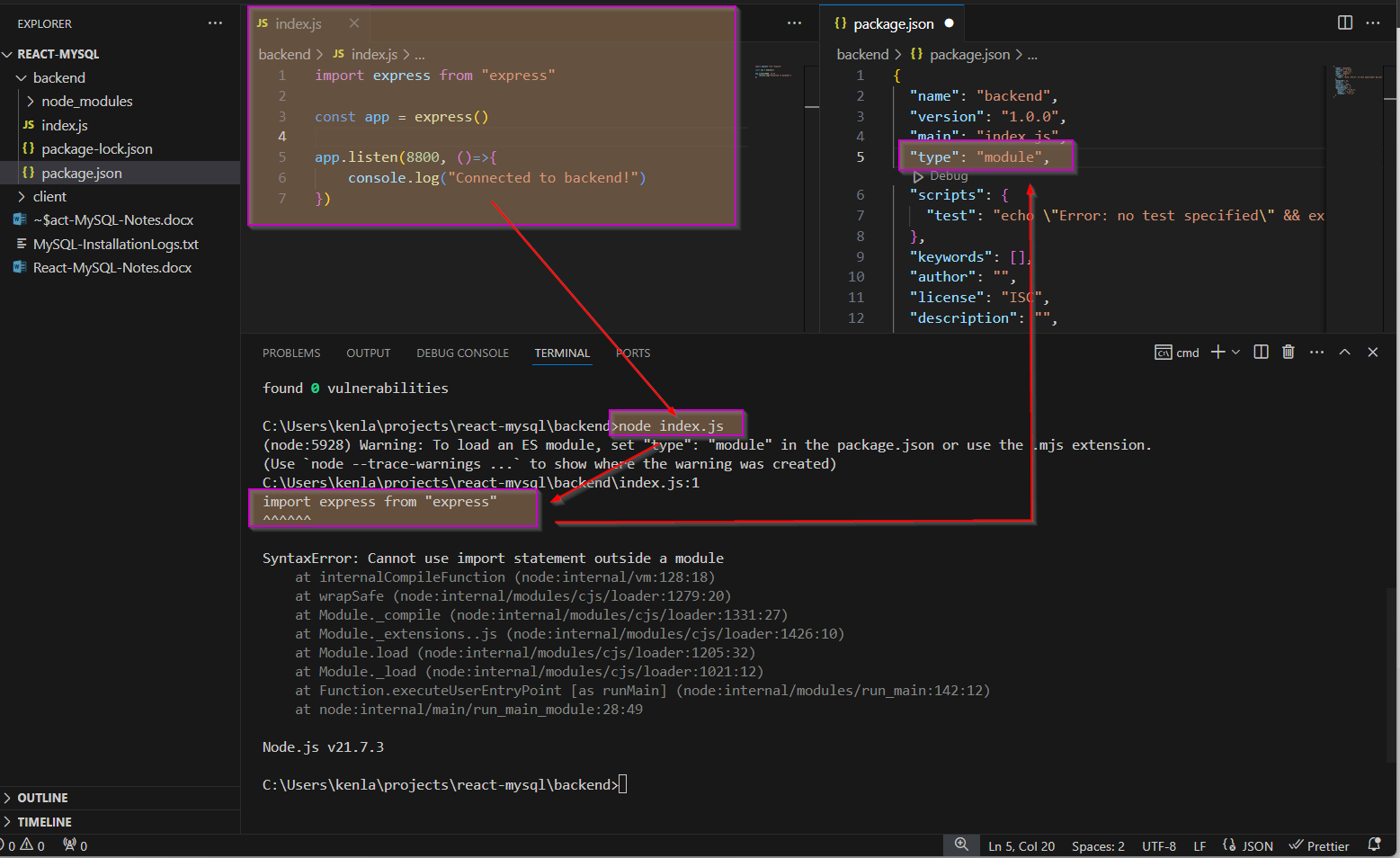


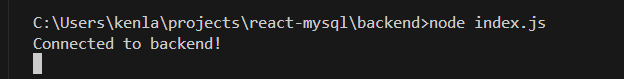


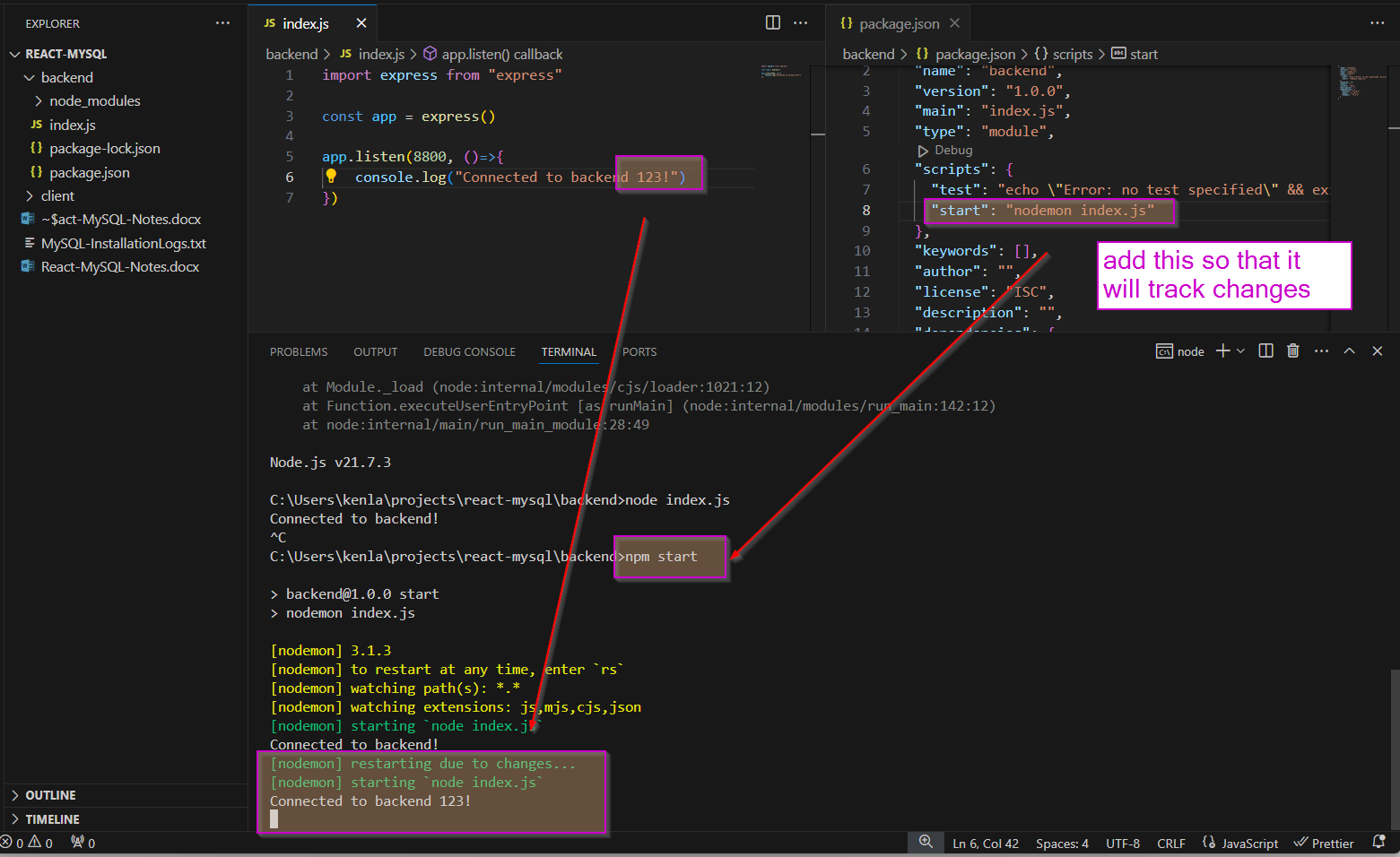


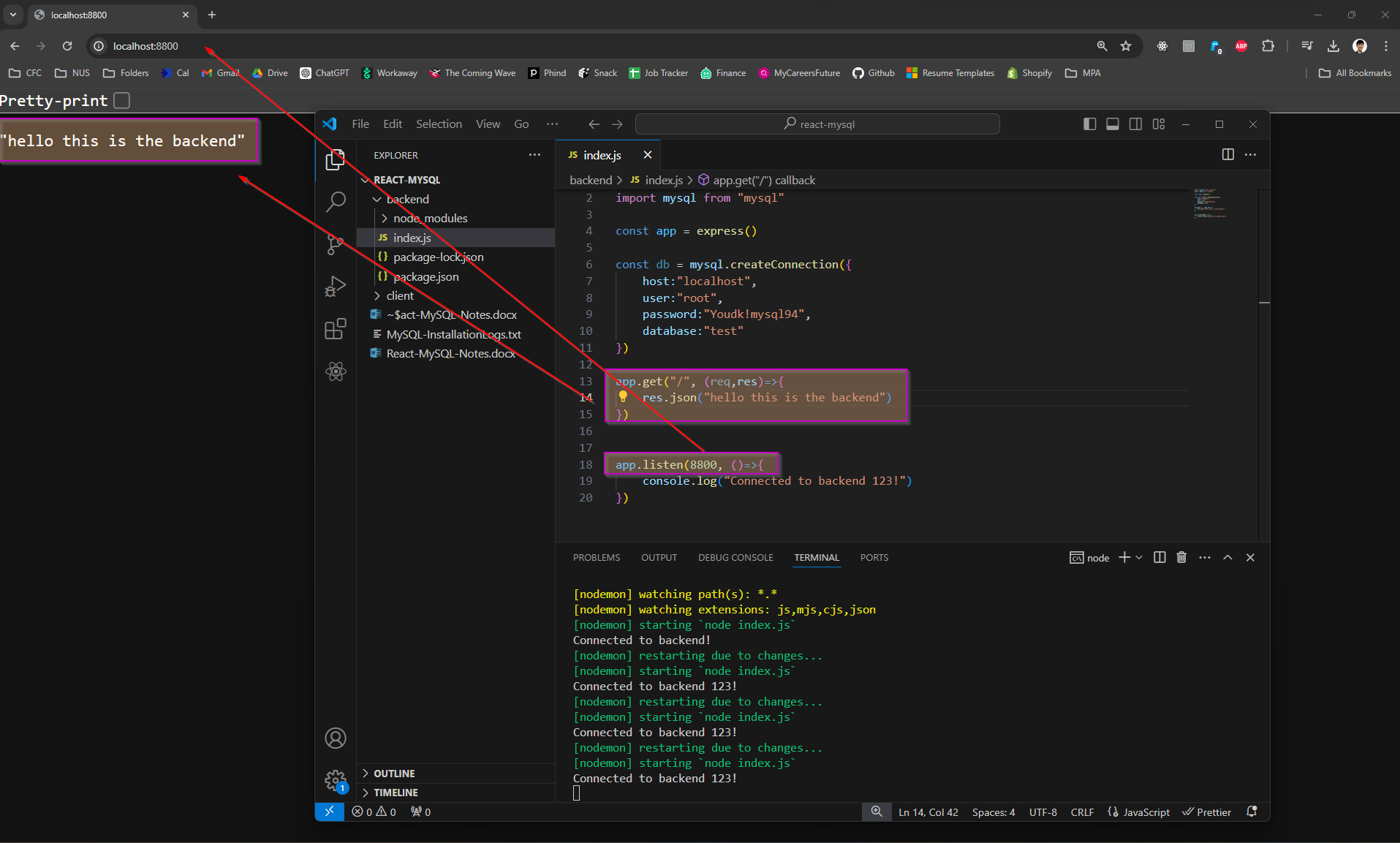


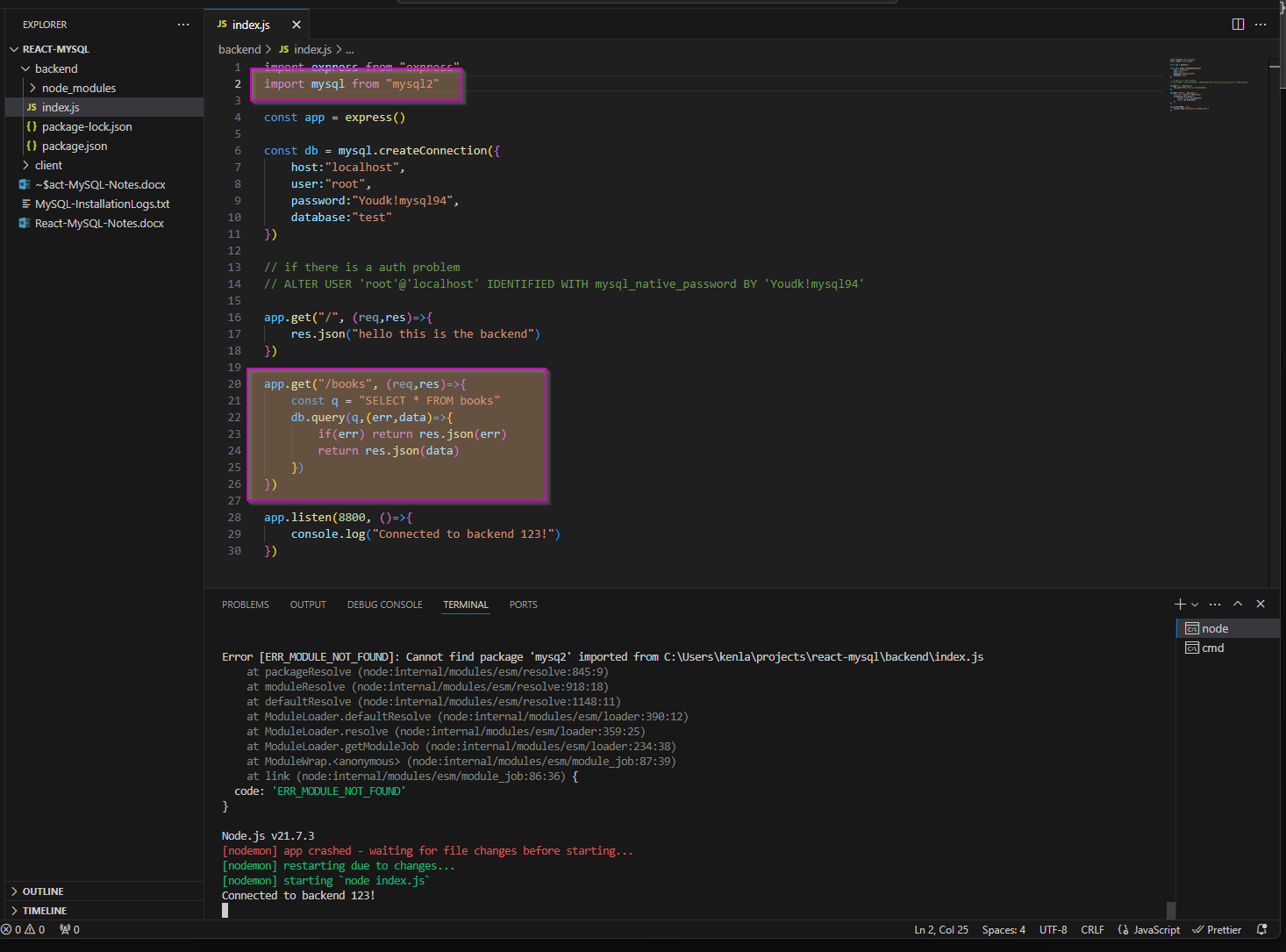


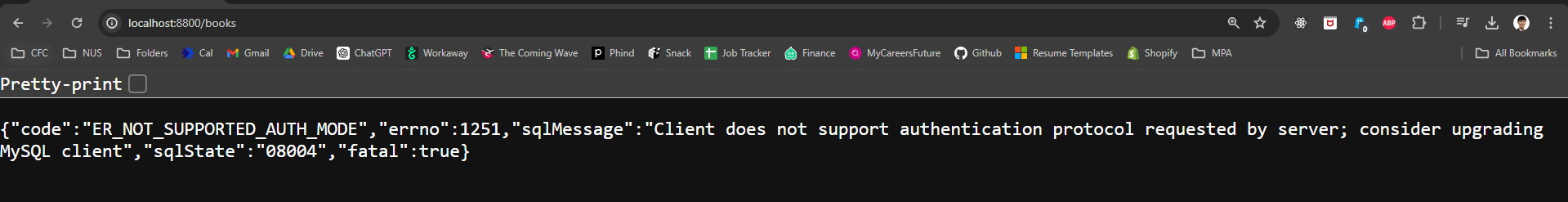


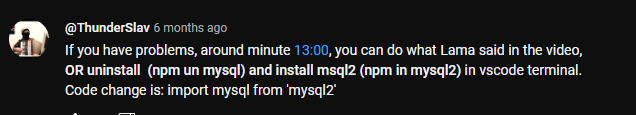


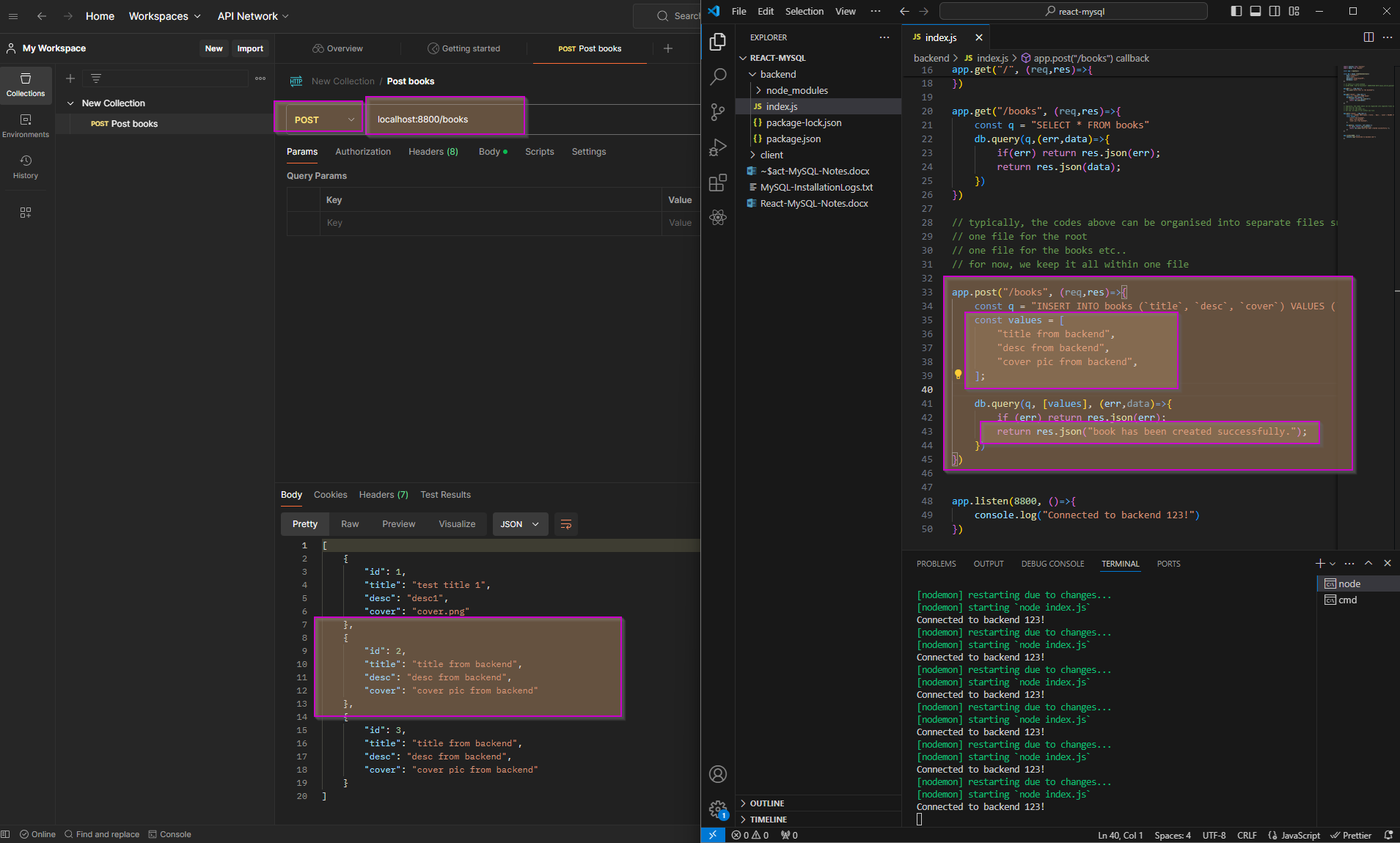


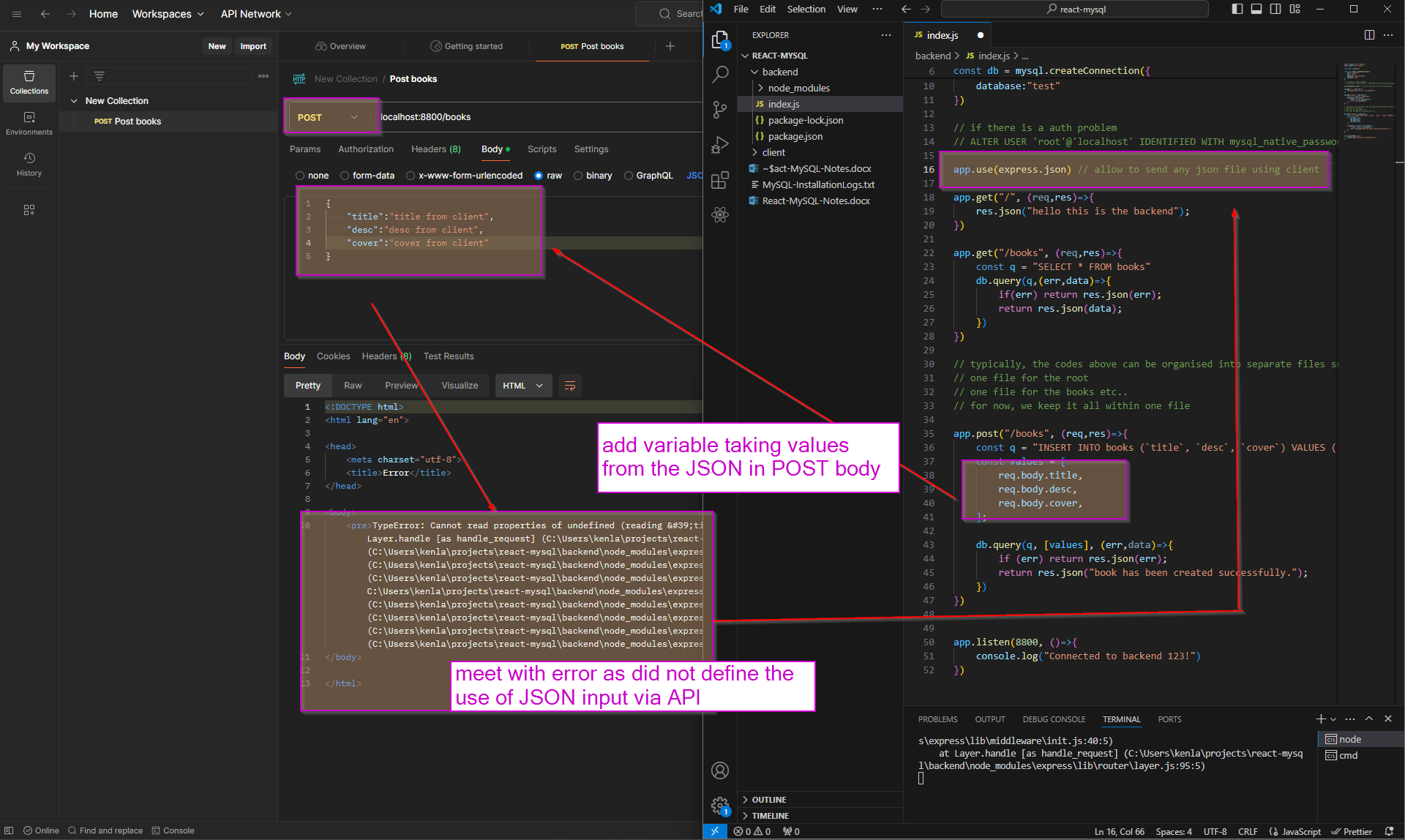




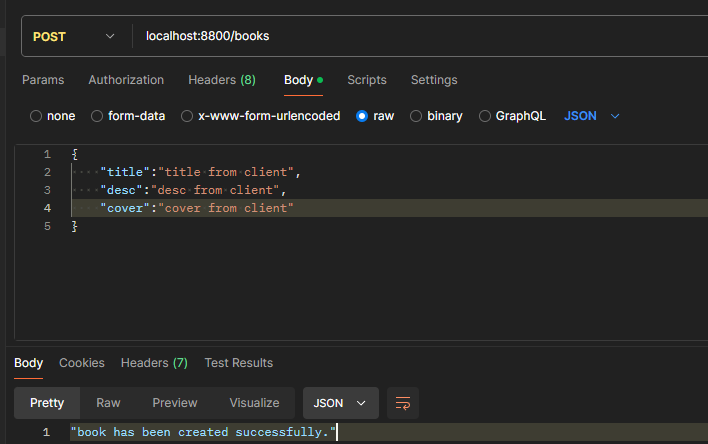


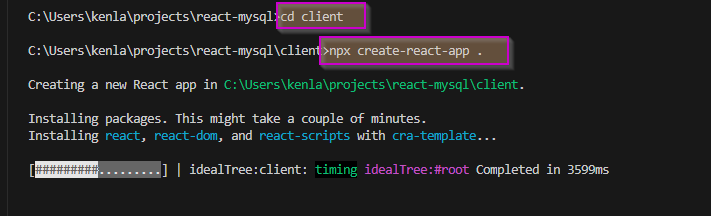


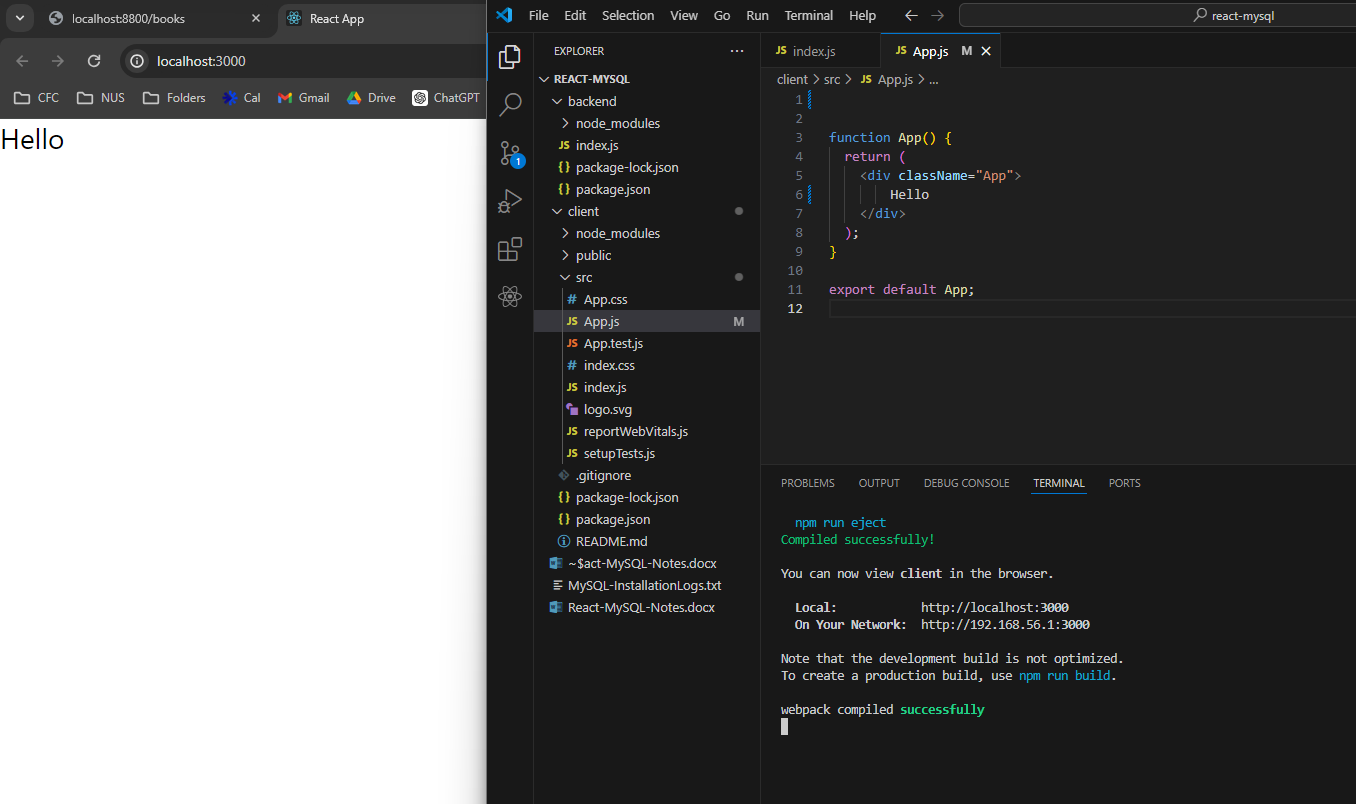


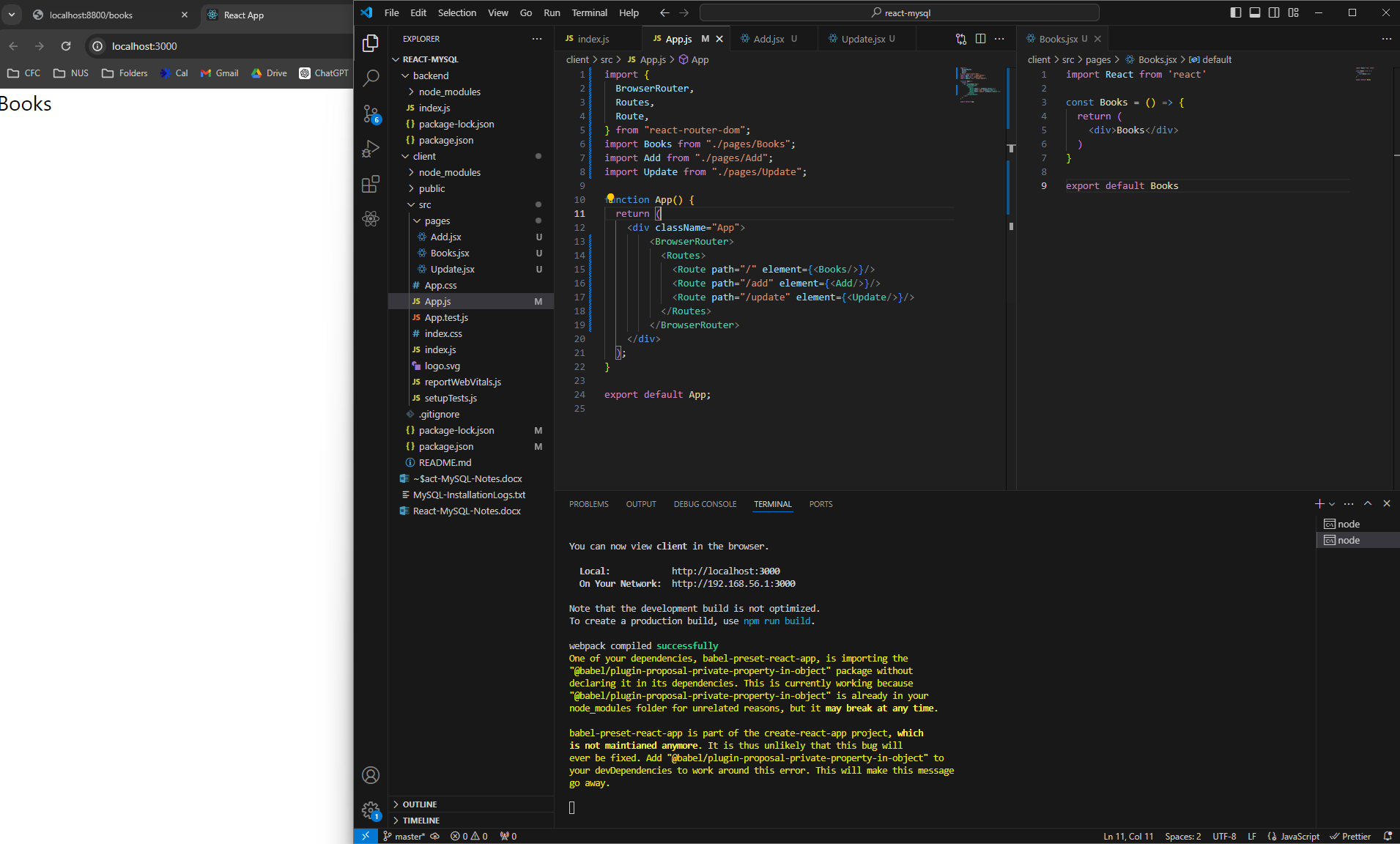


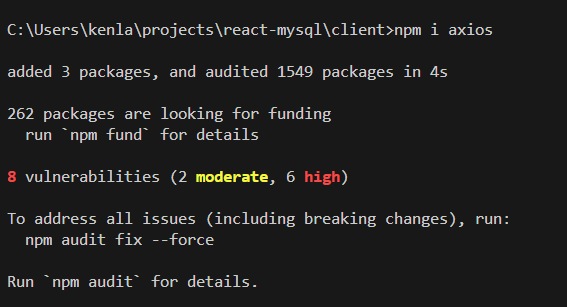


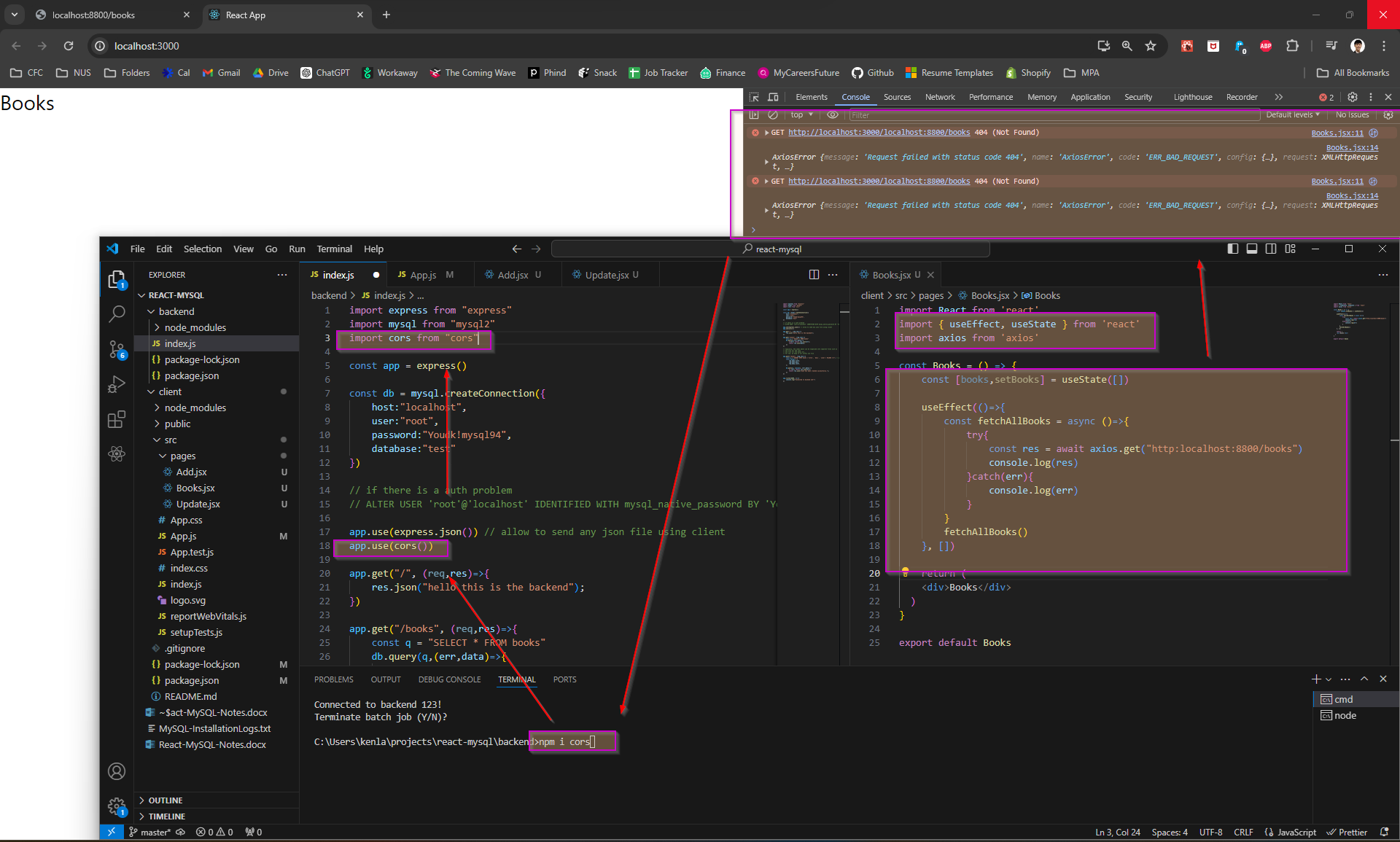




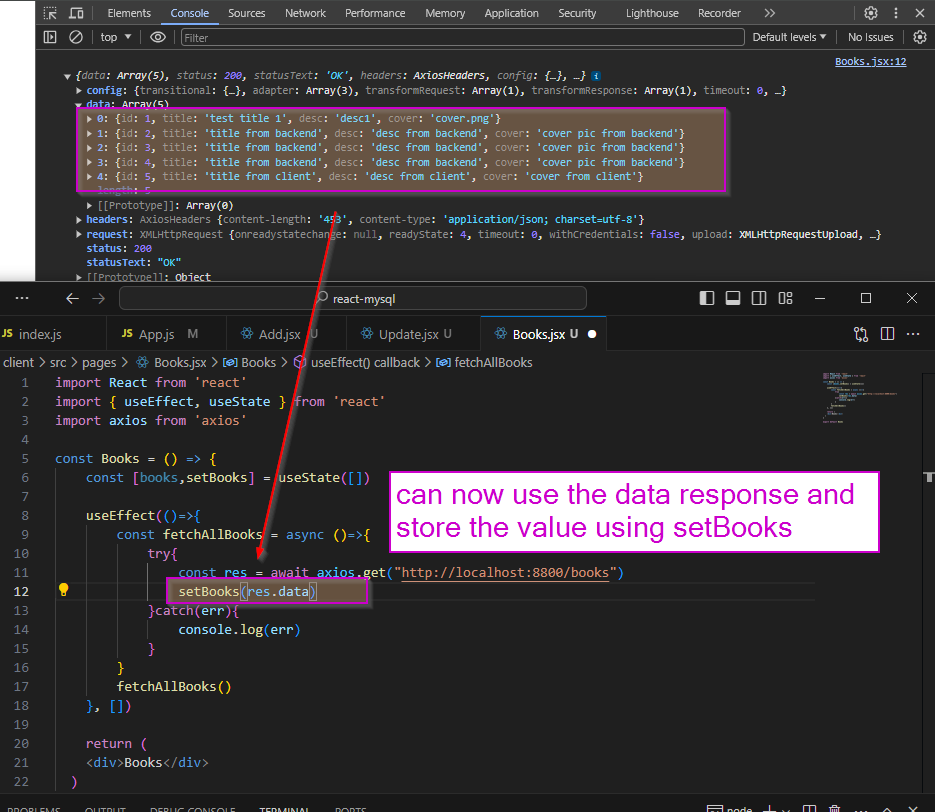


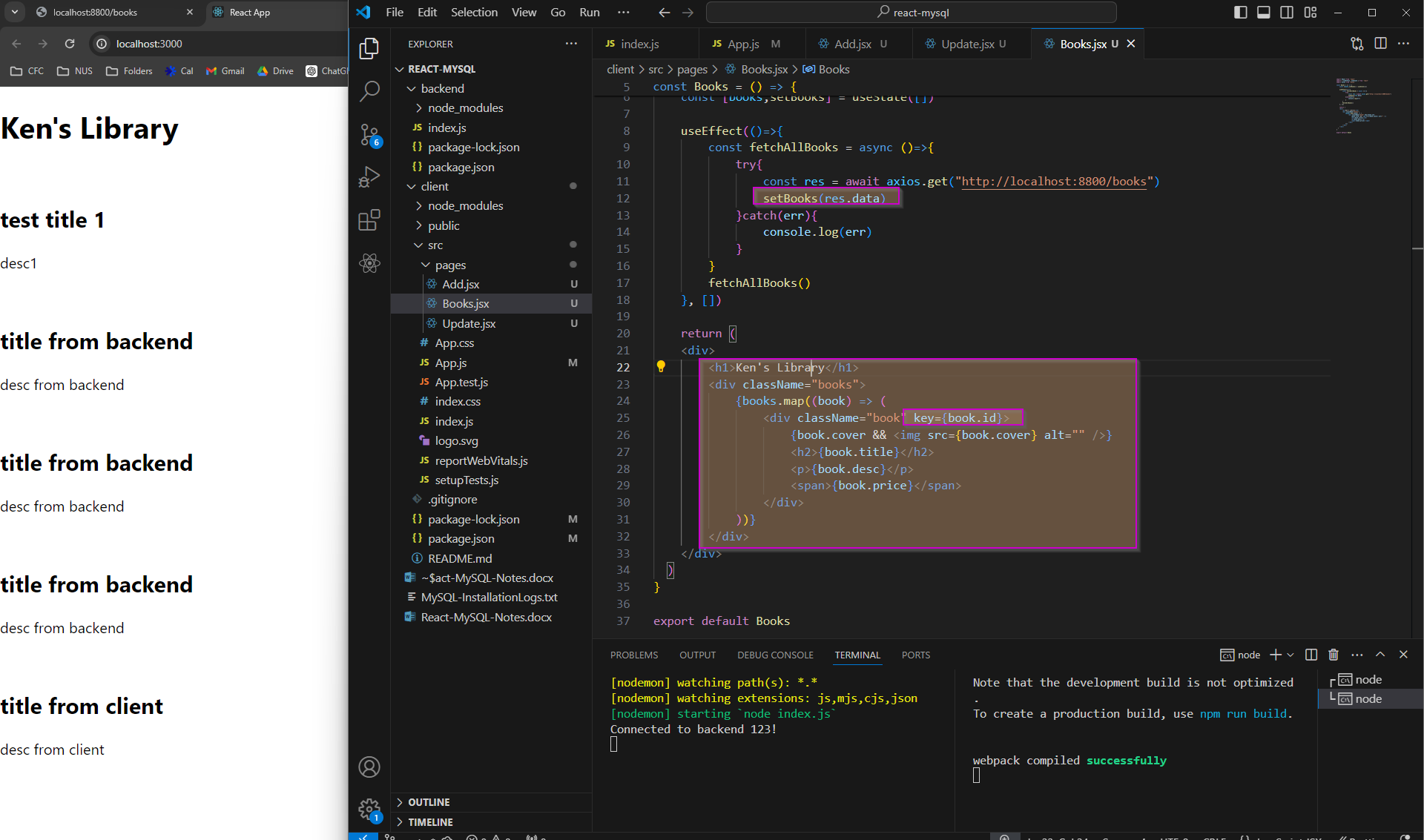




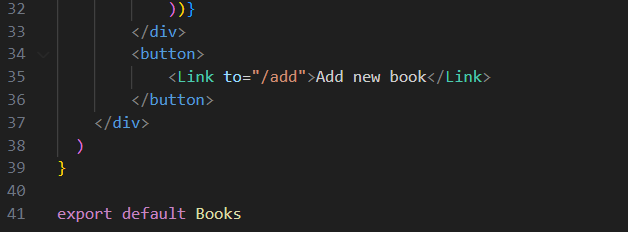


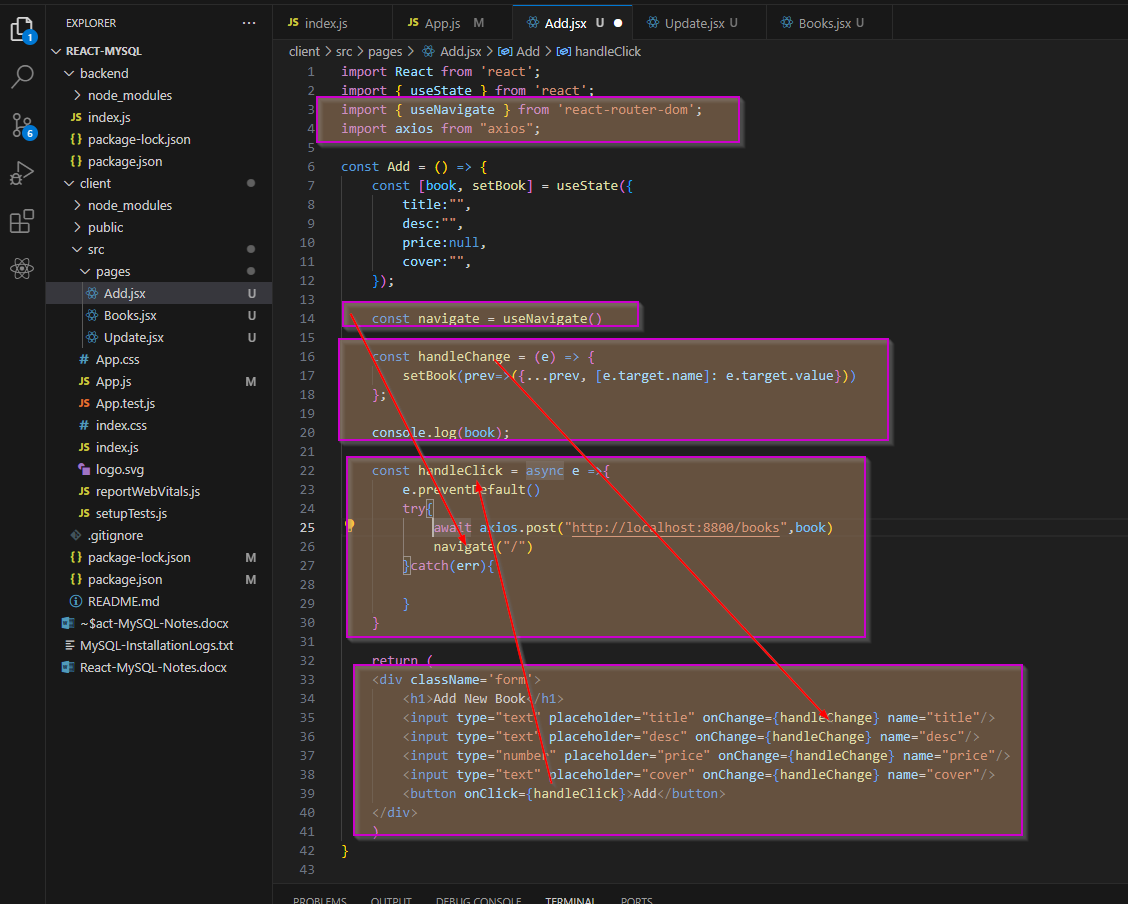


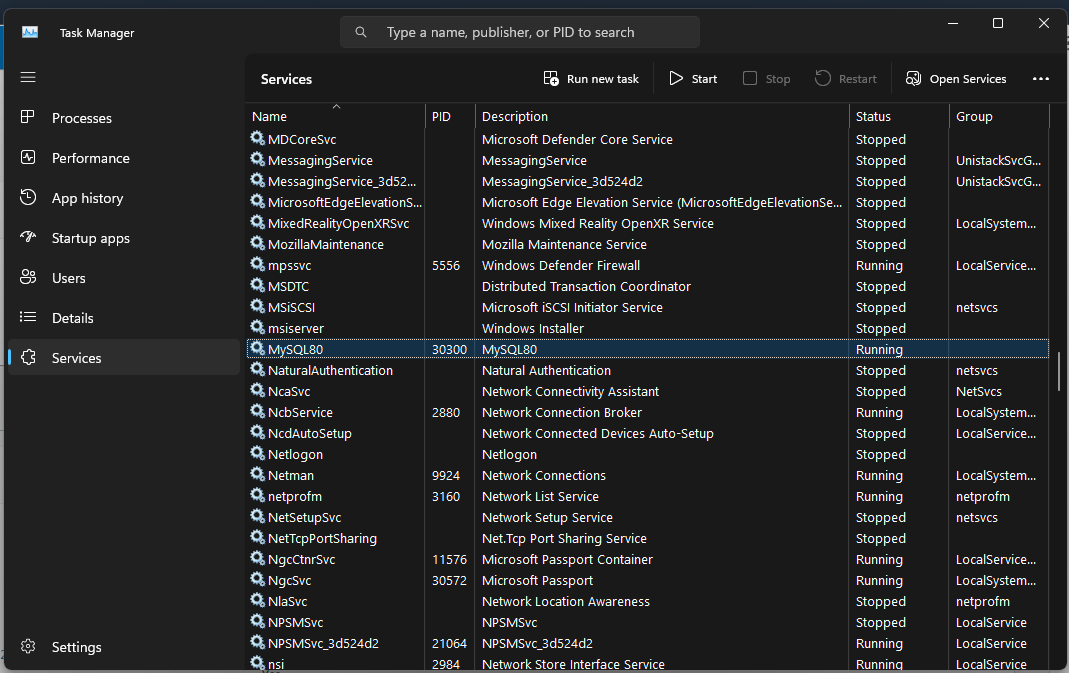










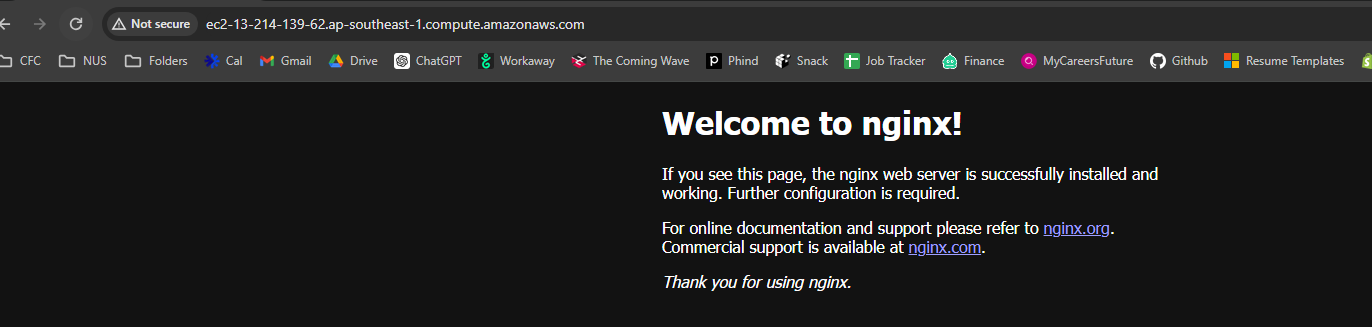


Need to make sure that MySQL80 service is running on localhost

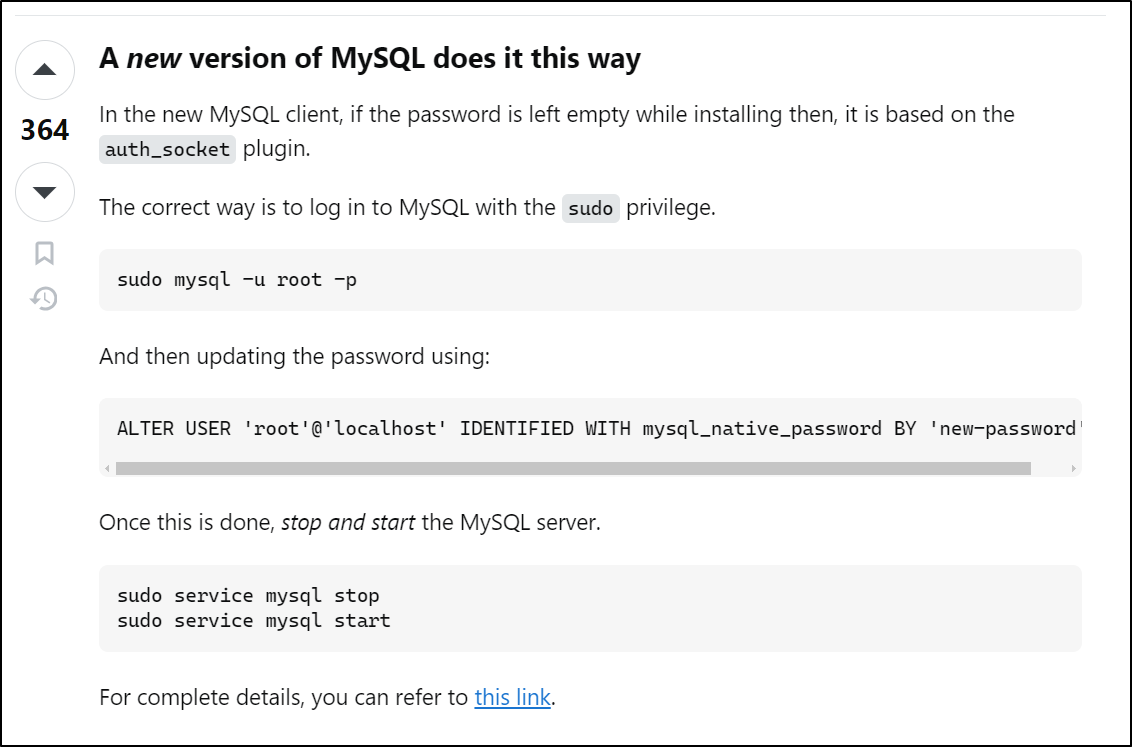
# Deploy onto EC2 instance (without RDS)

<https://towardsdev.com/deploying-a-react-node-mysql-app-to-aws-ec2-2022-1dfc98496acf>

1. Create EC2 instance with Ubuntu OS
   1. SSH into the EC2 instance
   2. Sudo apt update
   3. Sudo apt install git
2. Sudo apt install nginx
   1. Verify if nginx is installed, using http to the AWS EC2 IPv4 DNS address

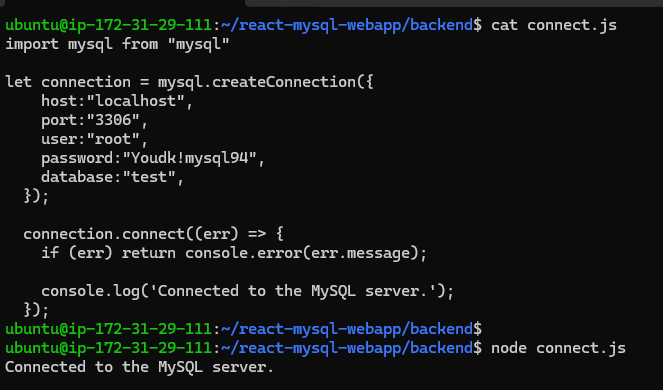


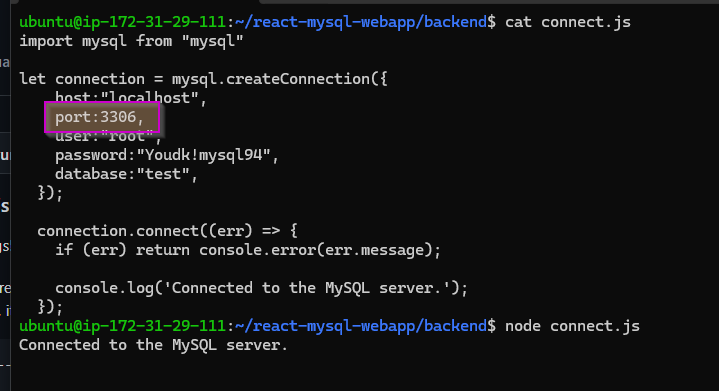
1. Curl -s <https://deb.nodesourcce.com/setup_21.x> | sudo bash
   1. Sudo apt install nodejs -y
   2. Note: seems like need to run with the URL will be more complete, else, hit an error later that npm is not found
2. Push changes to the git repository
   1. Make sure that node\_modules or other unnecessary files are excluded in .gitignore file
   2. Make sure the remote repository is correct
      1. git status
      2. git branch
      3. If need to change URL
         1. git remote -v
         2. git remote set-url origin <new-url>
         3. git remote -v
   3. Add, commit and push changes
      1. Git add.
      2. Git commit -m “xxxx”
      3. git push origin main
3. Clone the git repository
   1. git clone xxxx
   2. cd webappdemo/client   
      sudo npm install  
      sudo npm run build
   3. cd ../backend/ && sudo npm install
4. Setup nginx to point to our newly build react app
   1. Sudo cp -R /home/ubuntu/webappdemo/client/build /var/www
   2. sudo vim /etc/nginx/sites-available/default
      1. server {  
          listen 80 default\_server;  
          listen [::]:80 default\_server; # We want the root folder to point at index.html  
          root /var/www/build;  
          index index.html index.htm index.nginx-debian.html;  
           
          server\_name \_;  
           
          location / {  
          # First attempt to serve request as file, then  
          # as directory, then fall back to displaying a 404.  
          try\_files $uri /index.html $uri/ =404;  
          }  
            
          location /api {  
          proxy\_pass http://localhost:8081;  
          }  
         }
   3. ESC + :wq to save and exit
   4. Sudo systemctl restart nginx
      1. Check if pointing correctly to the react website now
5. Install pm2
   1. Sudo npm install pm2 -g
   2. pm2 start index.js
   3. pm2 status



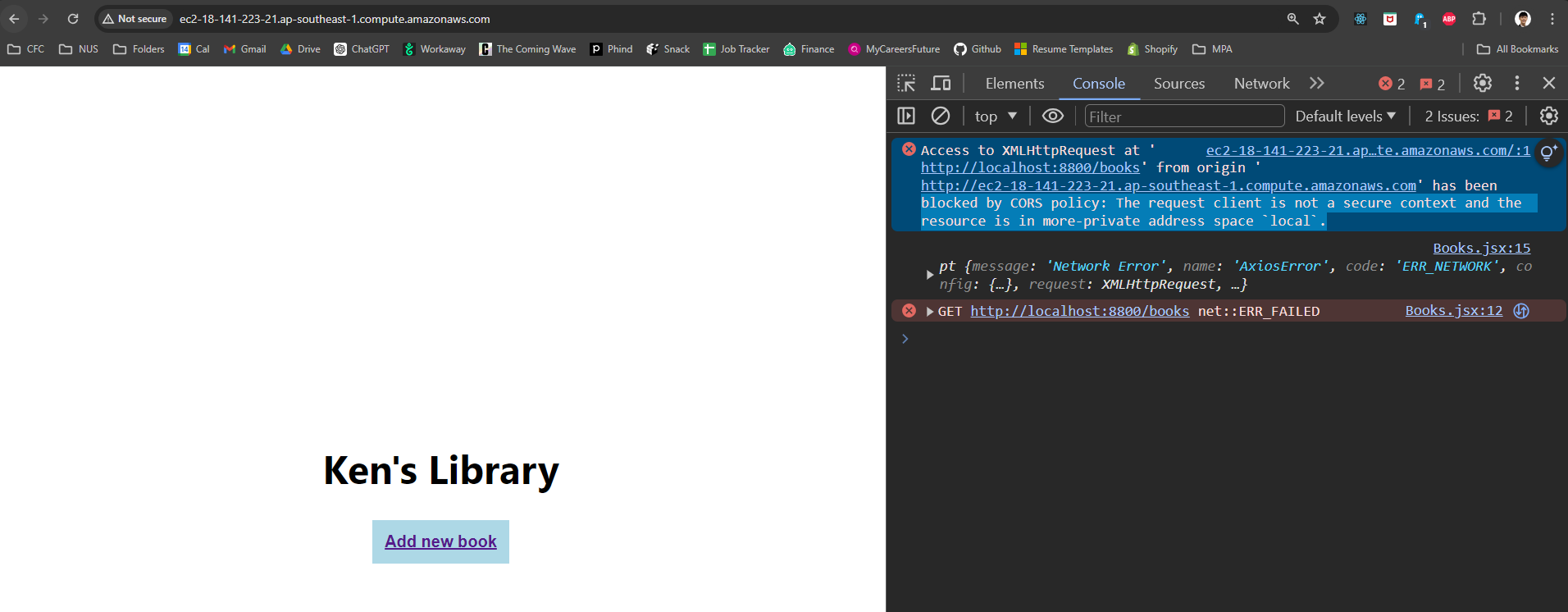
<https://www.mysqltutorial.org/mysql-nodejs/connect/>

Debugging why the mysql server is not connected to react website (or, to rephrase, why the react website unable to pull the mysql database)

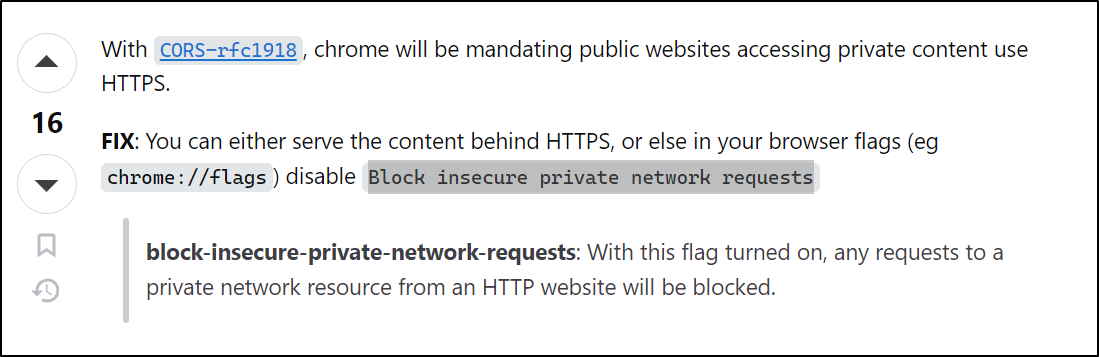


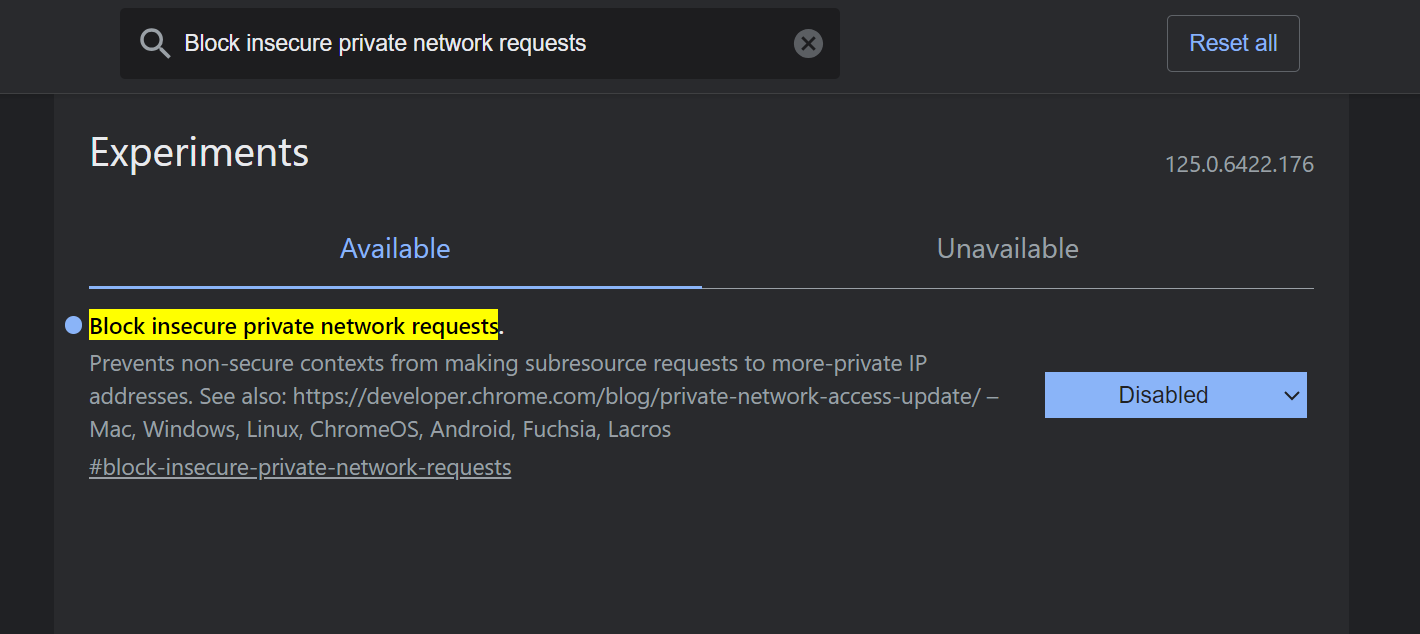


1. Git pull xxxxx
   1. If any conflict in changes, can run
   2. git checkout filename



Access to XMLHttpRequest at 'http://localhost:8800/books' from origin 'http://ec2-18-141-223-21.ap-southeast-1.compute.amazonaws.com' has been blocked by CORS policy: The request client is not a secure context and the resource is in more-private address space `local`.





<https://dev.to/codewithbernard/how-to-run-react-js-build-locally-4nd7>

1. Build locally and push it to github and pull into the EC2
   1. Npm run build
   2. Test if the build is working as expected
      1. Npm install -g serve
      2. npx serve -s build

# Creating mysql database on EC2 (instead of using RDS)

SSH into your EC2 instance:

sudo apt update  
sudo apt upgrade

sudo apt install mysql-server

sudo mysql\_secure\_installation

* VALIDATE PASSWORD COMPONENT >> Yes
* Levels of password validate policy >> 2 strong
* Remove anonymous users >> Yes
* Disallow root login remotely >> No
* Remove test database and access >> No
* Reload privilege tables >> No

Login to mysql using

sudo mysql

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'Your\_Secure\_Password';

CREATE USER 'admin'@'%' IDENTIFIED BY 'Your\_Secure\_Password';

GRANT ALL PRIVILEGES ON \*.\* TO 'admin'@'%' WITH GRANT OPTION;

FLUSH PRIVILEGES;

EXIT;

Try again to login and create the necessary tables (example in next page)

Try to connect from local nodejs and react app

Example from the library MySQL Database creation

CREATE DATABASE test;

CREATE TABLE `test`.`books` (

`id` INT NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(45) NOT NULL,

`desc` VARCHAR(255) NOT NULL,

`cover` VARCHAR(45) NULL,

PRIMARY KEY (`id`));

INSERT INTO `test`.`books` (`id`, `title`, `desc`, `cover`) VALUES ("1", "title1", "desc1", "cover1.png");

INSERT INTO `test`.`books` (`id`, `title`, `desc`, `cover`) VALUES ("2", "title2", "desc2", "cover2.png");

use test;

SELECT \* FROM books;

ALTER TABLE `test`.`books` ADD COLUMN `price` INT NOT NULL AFTER `desc`;

INSERT INTO `test`.`books` (`id`, `title`, `desc`, `price`, `cover`) VALUES ("3", "title3", "desc3","3", "cover3.png");

use test;

SELECT \* FROM books;