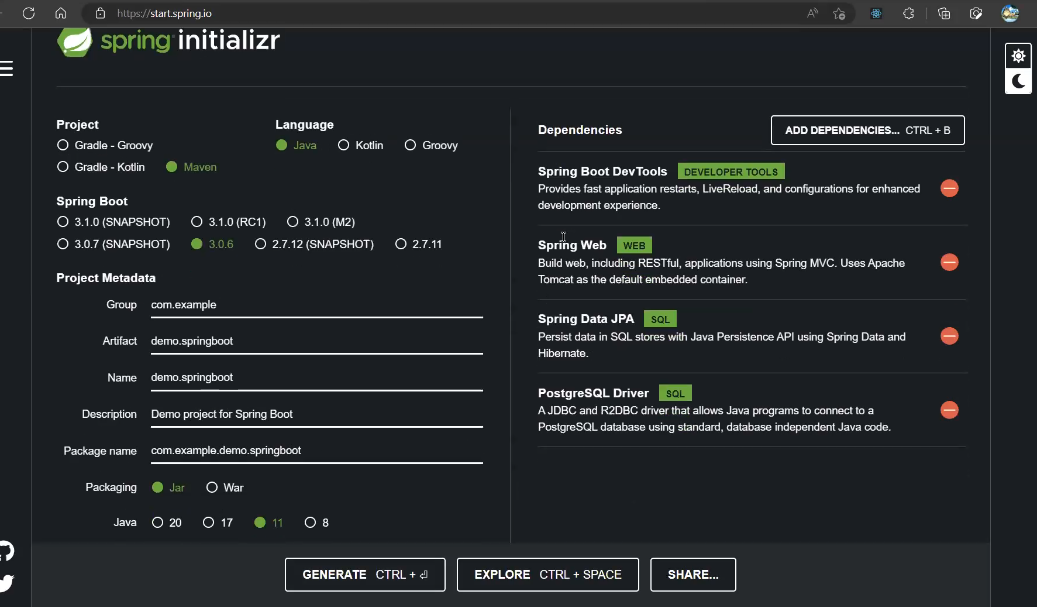
MY SPRINGBOOT NOTES:

# ##How to Install Spring

1. Go to Spring Initializr <https://start.spring.io/>
2. Fill up project details (include Spring Web, Spring Data JPA, PostgreSQL Driver, and Spring Boot Dev Tools)
3. Generate
4. Unzip file
5. Open Eclipse
6. Import File as Maven/Gradle project
7. Wait for the dependencies to install
8. Setup properties for DB
9. Launch project

****

//Open folder sa Eclipse,   
//File → Import → Maven → Existing maven projects   
//browse click demo.springboot extracted folder  
//Punta sa src → Main → java → Application tapos run   
//Lalabas dapat yung SPRING na maangas

**Database (application.properties):**

spring.main.banner-mode=off

logging.level.org.springframework=ERROR

spring.jpa.hibernate.ddl-auto=none

# Information below are needed for connecting to DB server in this case PostgreSQL

spring.datasource.initialization-mode=always

spring.datasource.platform=postgres

spring.jpa.properties.hibernate.default\_schema = springsample

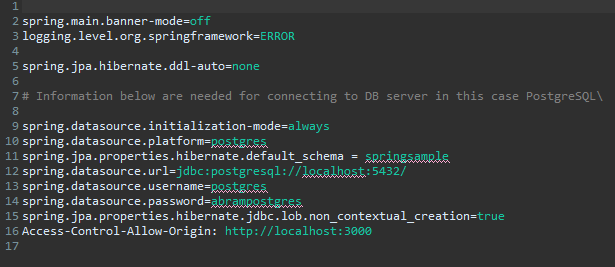
spring.datasource.url=jdbc:postgresql://localhost:5432/

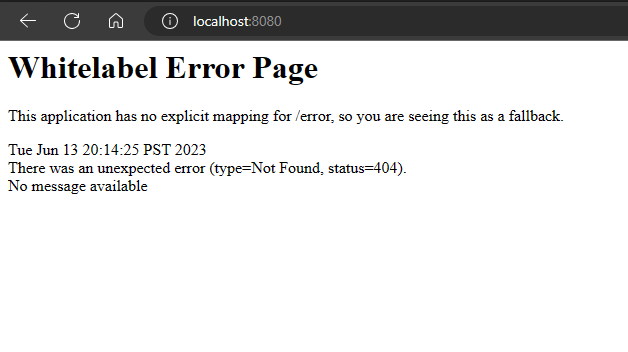
spring.datasource.username=postgres

spring.datasource.password=abrampostgres

spring.jpa.properties.hibernate.jdbc.lob.non\_contextual\_creation=true

//copy paste this to application.properties  
//change the password to your own password (abrampostgres)  
//change the schema to the name of your database (ex: springsample)  
//in database this is under postgres -> schema -> springsample



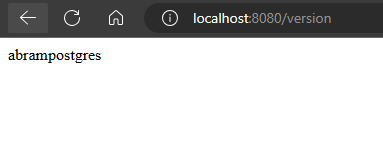
//Type in the browser: [http://localhost:8080](http://localhost:8080/)  
//And this will pop up

# Parts of Spring

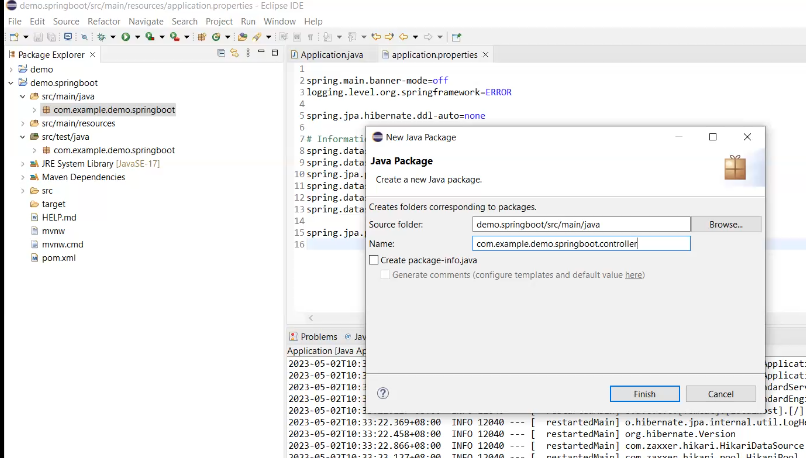
* Controller - for the API part. Mapping URL Paths
* Service - responsible for Logic
* Model - database structure
* Repository - includes all the methods such as sorting, paginating data, and CRUD operations.

**SpringBoot Derived Query Methods:**<https://www.baeldung.com/spring-data-derived-queries>

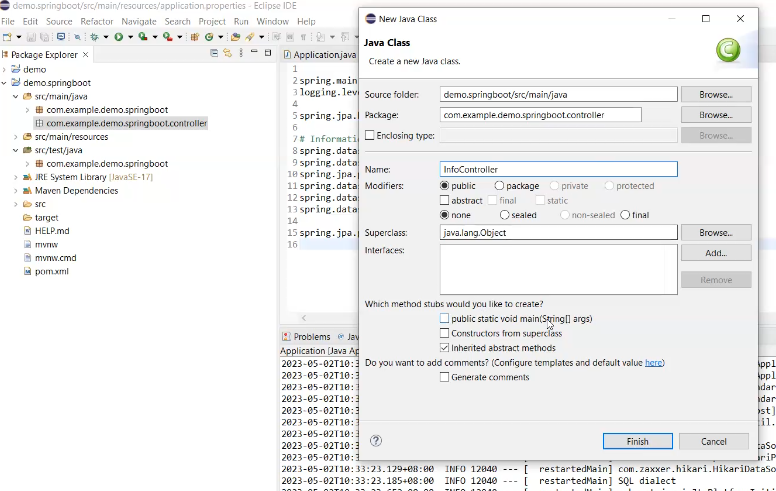
//edit the API springboot to your desired application  
//for the demo, we are retrieving students  
//also try <http://localhost:8080/version> (edited na ung api, so may lalabas)



//right click, then select new package  
//name it “ “.controller - to to have a controller

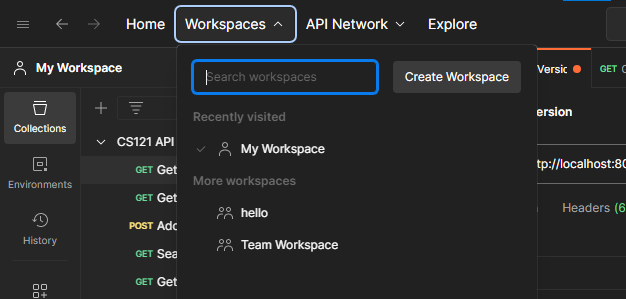


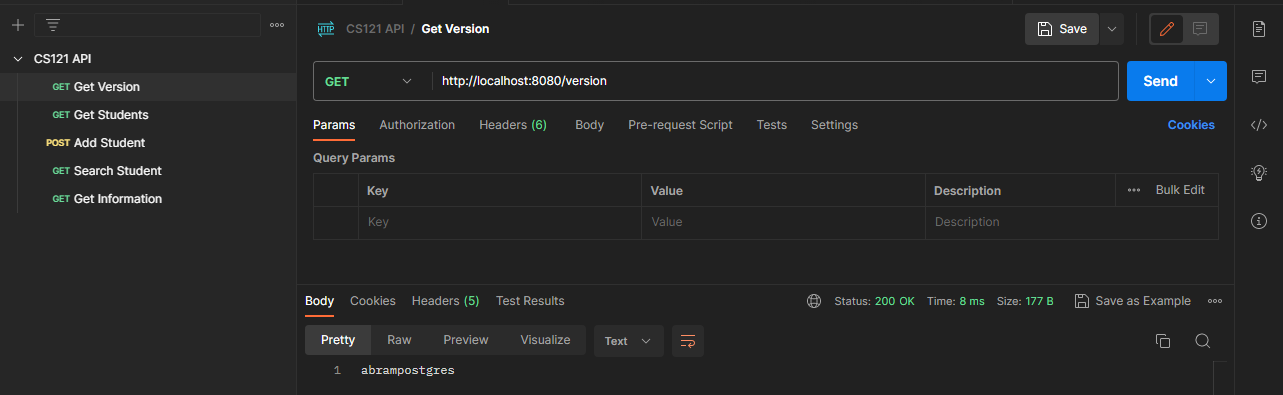
//next is to make a class inside it. Lets make InfoController

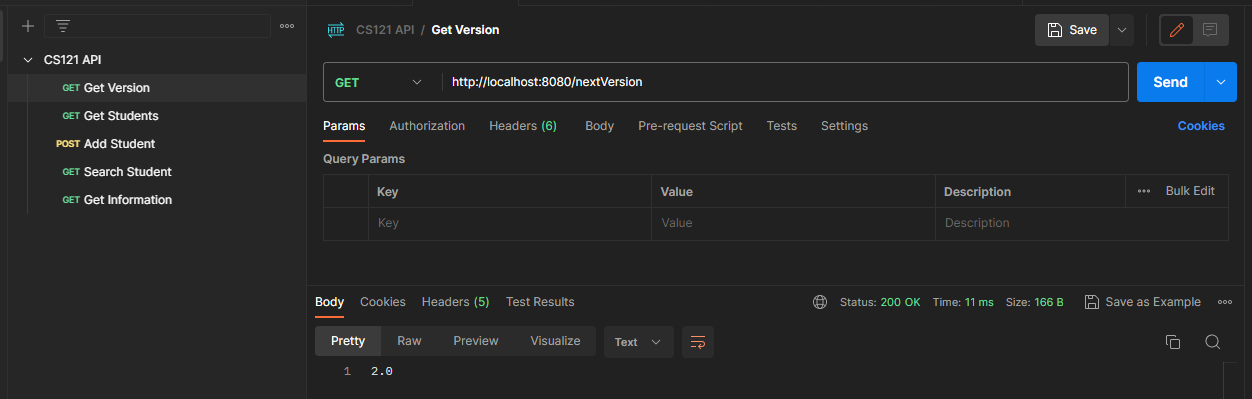


# Postman

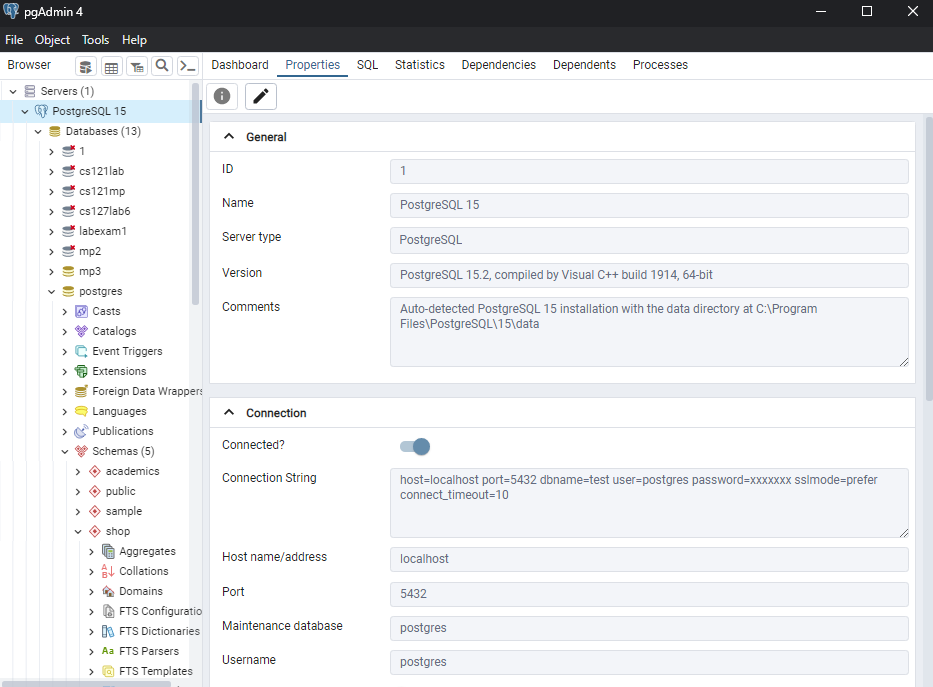
1. Create a workspace
2. Go to collections and hit ‘+’
3. Name your collections
4. Add some requests change whether GET or POST



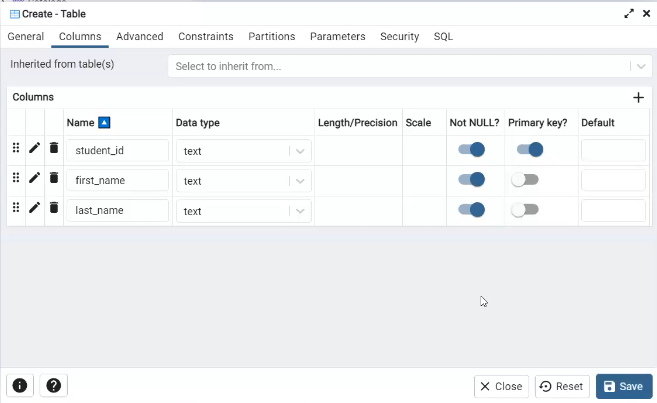


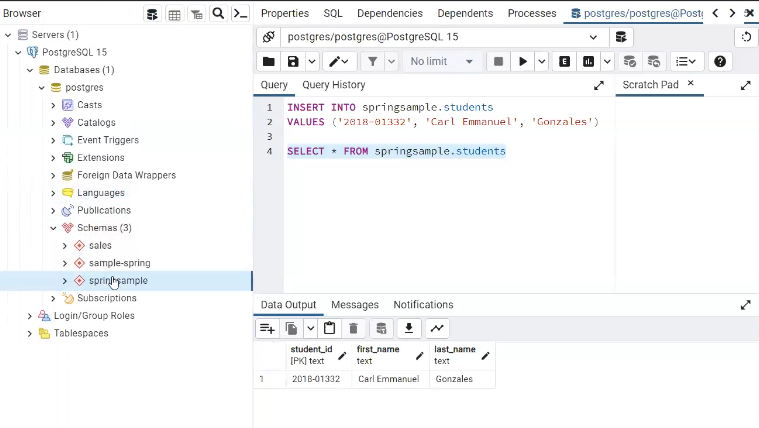


# PgAdmin4



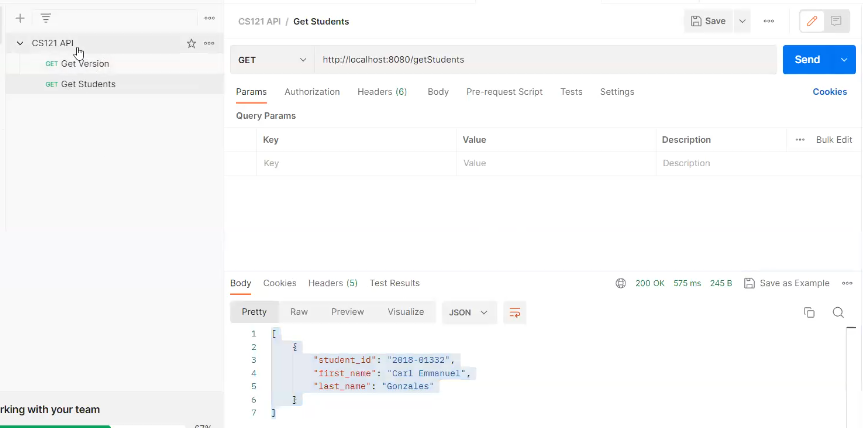
//this is the properties of the database na nilagay natin sa application.properties sa eclipse  
//gawa ka lang ng new schema so ayun nga (springsample)  
//gawa ka din ng table under ng schema nayun  
//tas try mo mag insert ng values don



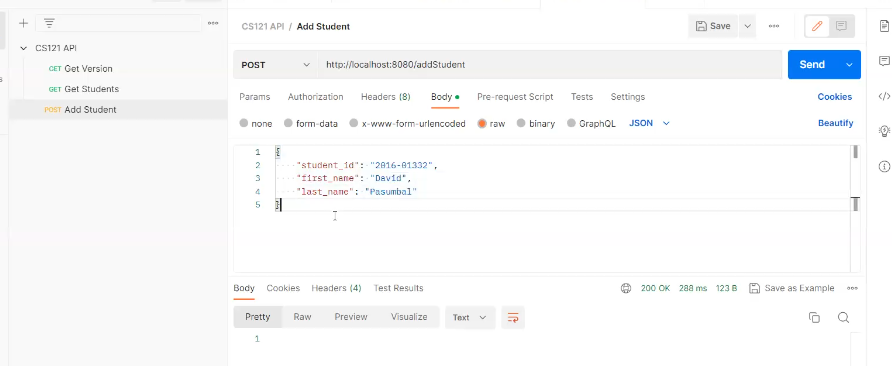


# =====================================================================================

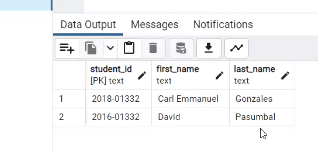
# Continue..



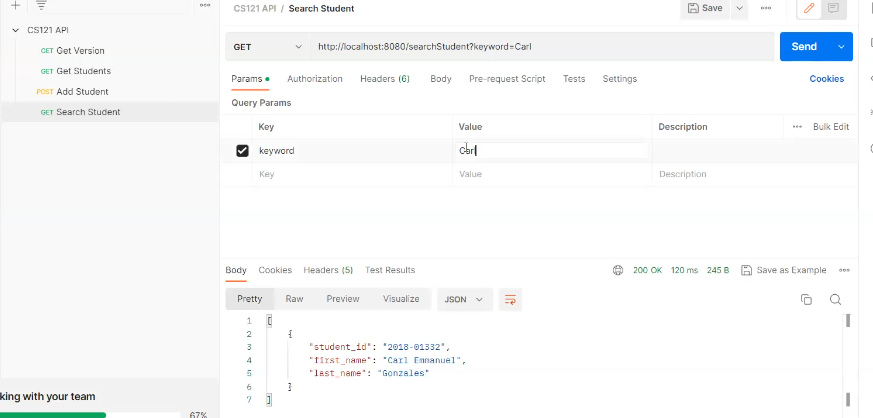
//mag getStudents ka, eto lalabas (note na nag eedit tayo sa eclipse para maretrieve)



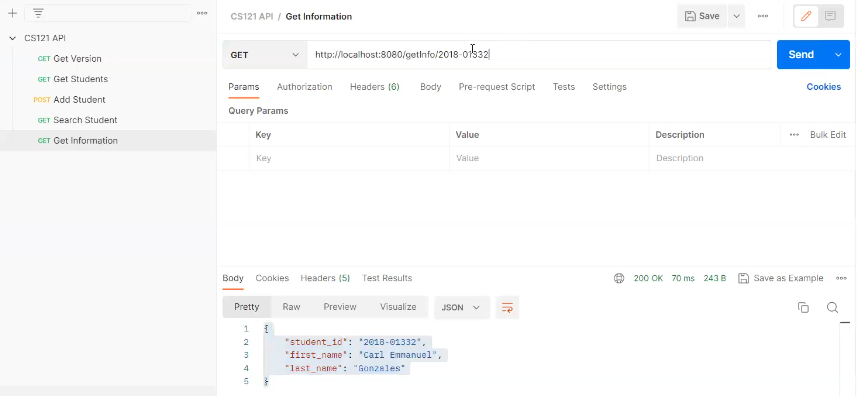
//add request ka uli tas POST naman  
//punta ka sa body, tas type mo ung iadd mo sa database  
//pagka click send, makikita mo na nag add na sya sa dababase



//try naman natin mag searchStudent  
//keyword lang ung ilalagay, pero mahahanap na natin ung student  
//note: sa StudentService.java – naka indicate kung findByFirstNameContains or findByLastNameContains or etc.



//getInfo naman   
//findById

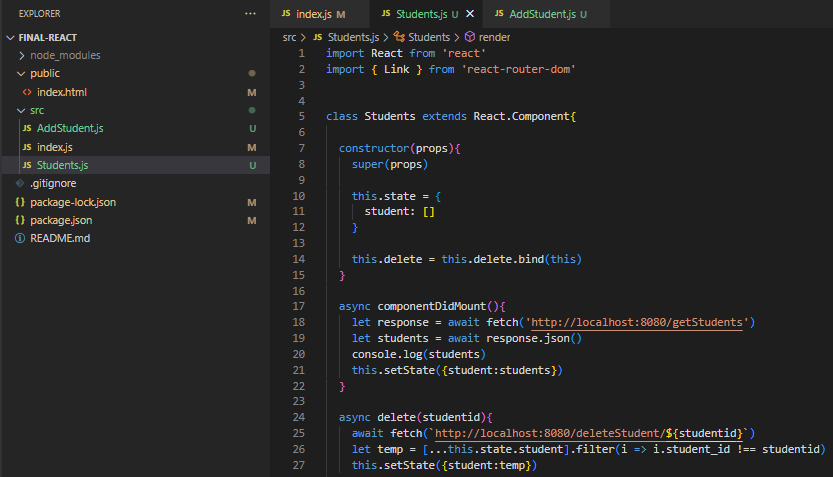


# =====================================================================================

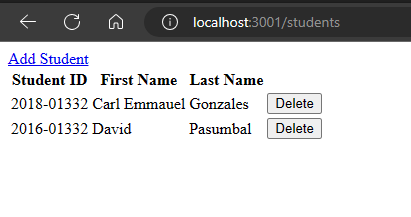
# ReactJS

//if wala pang reactJS project, mag create using - npx create-react-app <file name>  
//ex: npx create-react-app sample-react

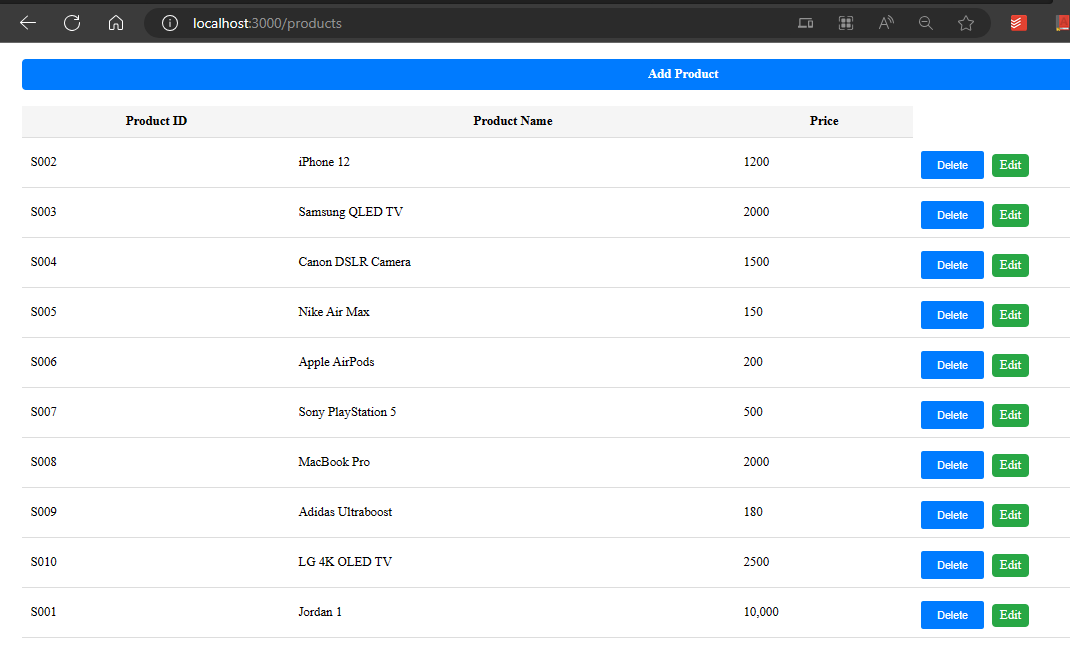
//setup the codes needed, connect to the api and database  
//run npm start



//after you run the project, localhost:3000 you will get a blank page  
//add students to it - localhost:3000//students



# My project

****