Website Setup Documentation

Tech Stack:

1. **Database:**
    a. MySQL
    b. SQL
    c. MySQL Workbench
2. **Server Side:**
    a. Java JDK (18.0.1)
    b. Spring Boot (3.3.1)
    c. Maven (3.9.8)
    d. IntelliJ IDEA 2023.3.2 (Community Edition)
3. **Client Side:**
    a. Javascript
    b. React.js (18.3.1)
    c. Node.js (20.9.0)
    d. VSCode (1.90.2)

MVP:

1. **Add**
    a. Product, Test, Instrument, Material
2. **Update**
    a. Product, Test, Instrument, Material
3. **Remove**
    a. Product, Test, Instrument, Material
4. **Search**
    a. Product, Test, Instrument, Material

Post MVP:

1. Add a next page
    a. Ex:
        i. Rule: limit is 20 results per page
        ii. 100 results → 5 pages
2. Website converts CSV test sheets to graph automatically
3. Export searched data
4. Import
5. Booking a material feature

    a. History of booking
        i. New table and database
6. Question mark bubble for extra info

Cloning Repository from Github:

1. Go to repository
2. Click on <> Code button (green)
3. Click on Download Zip
4. Open Zip file to create a Nara_DB_Lab_Laptop
5. **Client-side (Front-end):**
    a. Open VSCode
    b. Open Nara_DB_Lab_Laptop in VSCode
    c. Access terminal in VSCode (Look at navigation bar at top left and click on "Terminal")
    d. In terminal type: ls
        i. Output: blah\blah\blah\NARA_DB_Lab_Laptop-main
            1. Output should look something like this
            2. Note: On Windows, the file path would look like the top one, but in MacOS the slash is the opposite way → blah/blah/blah/NARA_DB_Lab_Laptop-main
    e. In terminal type: cd .\NaraDBWebsite\my-app
        i. Note: If on MacOS, look at the above comment about the slash direction
    f. In terminal type: npm install react-scripts --save
        i. Note: there are 2 dashes (-) before save
    g. In terminal type: npm start
        i. This should open a react page on chrome
        ii. CONGRATS, you are finished cloning the front-end!
6. **Server-side (Back-end):**
    a. Open IntelliJ
    b. Open Nara_DB_Lab_Laptop in IntelliJ
    c. Find and open the ServerSideNhrlDBApplication.java file in IntelliJ
    d. There will be an error message on top that says "Project JDK is not defined"
    e. Click on Setup SDK
    f. Click on 17 Oracle OpenJDK 17.0.12
        i. This is the one NARA installed on the lab laptop
    g. Go to terminal on IntelliJ

        i. Look at side navigation bar on bottom left of screen

h. In terminal type: mvn -v

    i. Correct output: Apache Maven 3.9.8

        1. If you get an output that looks something like this, then move to step i (skip the Output Error steps and move to the next step)

    ii. Output Error: The JAVA_HOME environment variable is not defined correctly, this environment variable is needed to run this program

        1. In terminal type: java -version

            a. Output: java version "17.0.12" 2024-07-16 LTS

            b. The output should look something like this

        2. In terminal type: [System.Environment]::SetEnvironmentVariable("JAVA_HOME", "C:\Program Files\Java\jdk-17")

            a. Note: notice that in the end of the file path, I typed in jdk-17. This is because my java version from above was 17.0.12. Make sure you pick the jdk number that corresponds to your java version. This part is ESSENTIAL to get correct.

            b. If this command doesn’t work, look at the maven-error-sol.txt in Google Drive and follow the steps for Regular Command Line

        3. In terminal type: echo $env:JAVA_HOME

            a. Output: C:\Program Files\Java\jdk-17

        4. In terminal type: mvn -v

        5. Output: Apache Maven 3.9.8

            a. The output should look something like this

i. Find and open application.properties file in IntelliJ

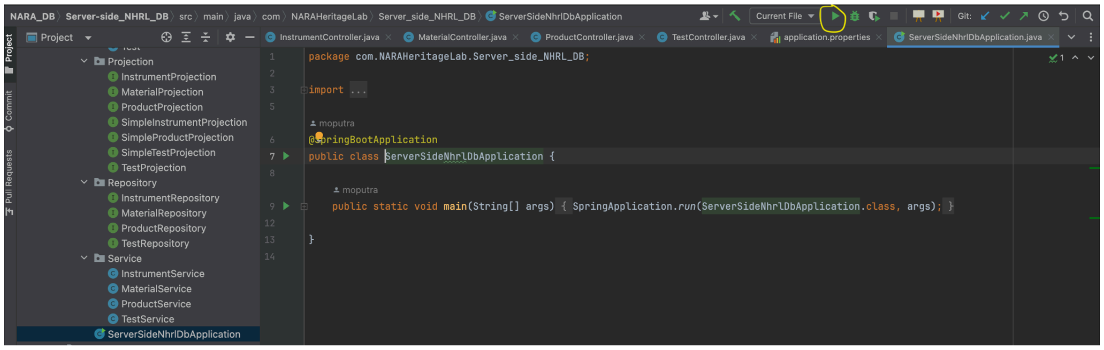
    i. Make sure that the code follows this structure:

```
spring.application.name=Server-side_NHRL_DB
spring.datasource.url=jdbc:mysql://localhost:3306/{Connection_Name}
spring.datasource.username=root
spring.datasource.password={Password}

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

```
9. spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8
   Dialect
```

- ii. Note: DO NOT include the {} in your code, they are just placeholders indicating what it needs to be replaced by
- iii. Note: for the localhost:3306, ensure that the number matches the one you see on MySQL Workbench. If the numbers don't match, change the code above to match the number in Workbench

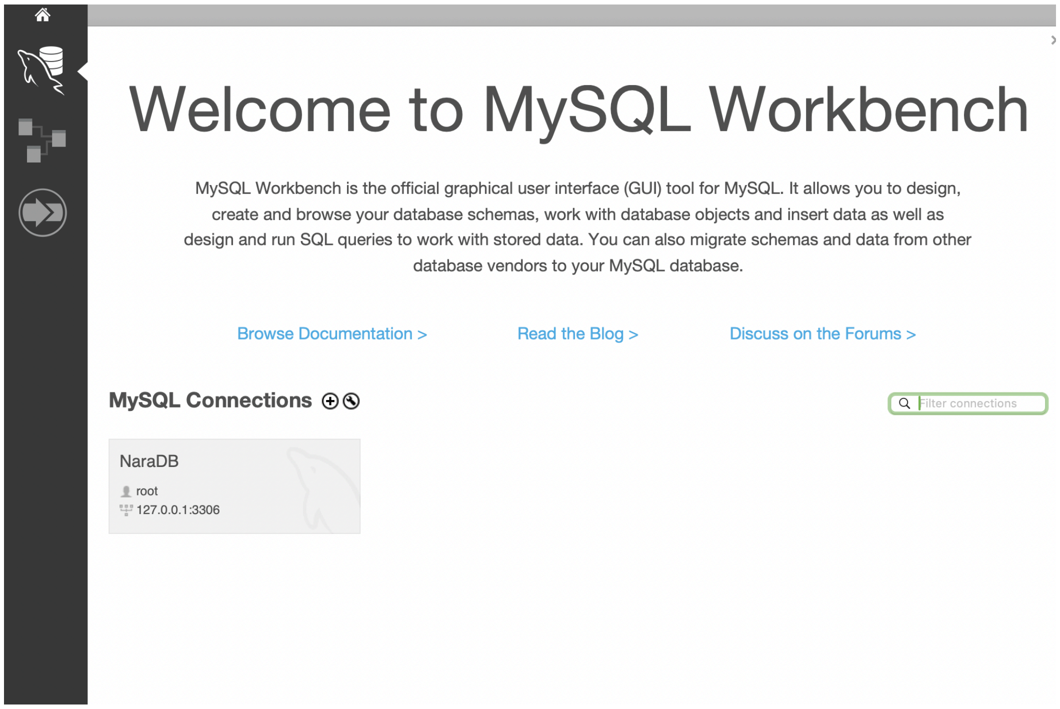
j. Find and open pom.xml file in IntelliJ
- i. Right-click on the code
- ii. Click + Add as Maven Project
- iii. Wait a few seconds till the dependencies load

k. Go back to ServerSideNhrlDBApplication.java in IntelliJ
- i. The icon next to the file name should turn from orange to blue and there should also be a green arrow icon
- ii. Click on the green run icon on top of the screen (look at circled part in image below)
- iii. If the code runs without crashing then, CONGRATS, you are finished cloning the server side!
- iv. 

```
package com.NARAHeritageLab.Server_side_NHRL_DB;

import ...

moputra
@SpringBootApplication
public class ServerSideNhrlDbApplication {

    moputra
    public static void main(String[] args) { SpringApplication.run(ServerSideNhrlDbApplication.class, args); }

}
```

Common Errors:

1. **MySQL Connection:**
   a. Make sure it looks EXACTLY like this

b. 

Welcome to MySQL Workbench

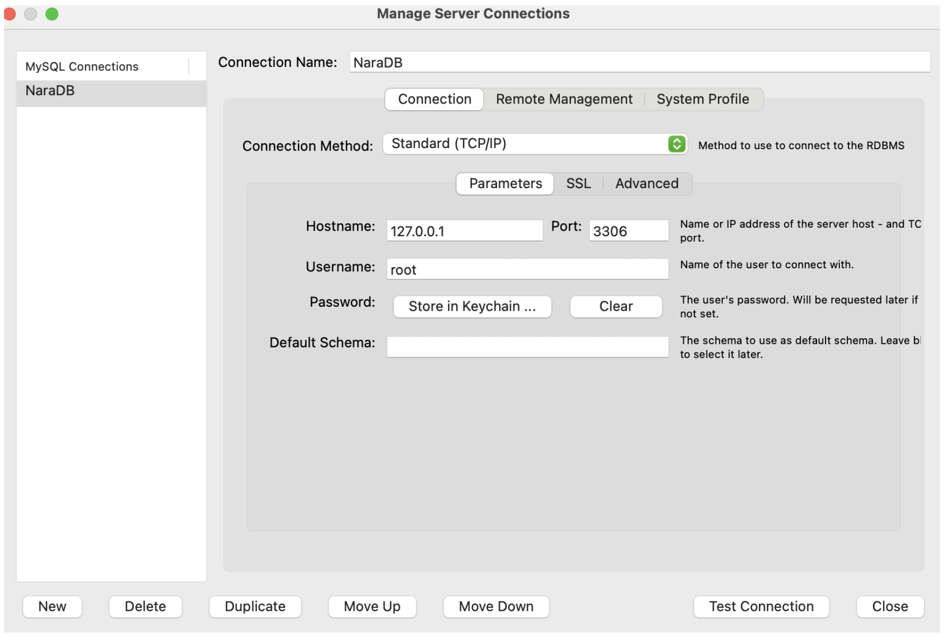
MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Browse Documentation > Read the Blog > Discuss on the Forums >

MySQL Connections

Filter connections

NaraDB
root
127.0.0.1:3306

c. If it does NOT look like this, follow these steps:
   i. Right-click on the grey box
   ii. Click on Edit Connection…
   iii. Copy this image, then click Test Connection, then save
   iv. Note: make sure that application.properties files have the same Connection Name and Port (Look above for instructions)
   v. 

Manage Server Connections

MySQL Connections
NaraDB

Connection Name: NaraDB

Connection | Remote Management | System Profile

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Parameters | SSL | Advanced

Hostname: 127.0.0.1 Port: 3306 — Name or IP address of the server host - and TC port.

Username: root — Name of the user to connect with.

Password: Store in Keychain ... Clear — The user's password. Will be requested later if not set.

Default Schema: — The schema to use as default schema. Leave b to select it later.

New | Delete | Duplicate | Move Up | Move Down | Test Connection | Close

Testing Endpoints:

1. Use Postman
2. Look at API Documentation in Google Drive to understand each endpoint