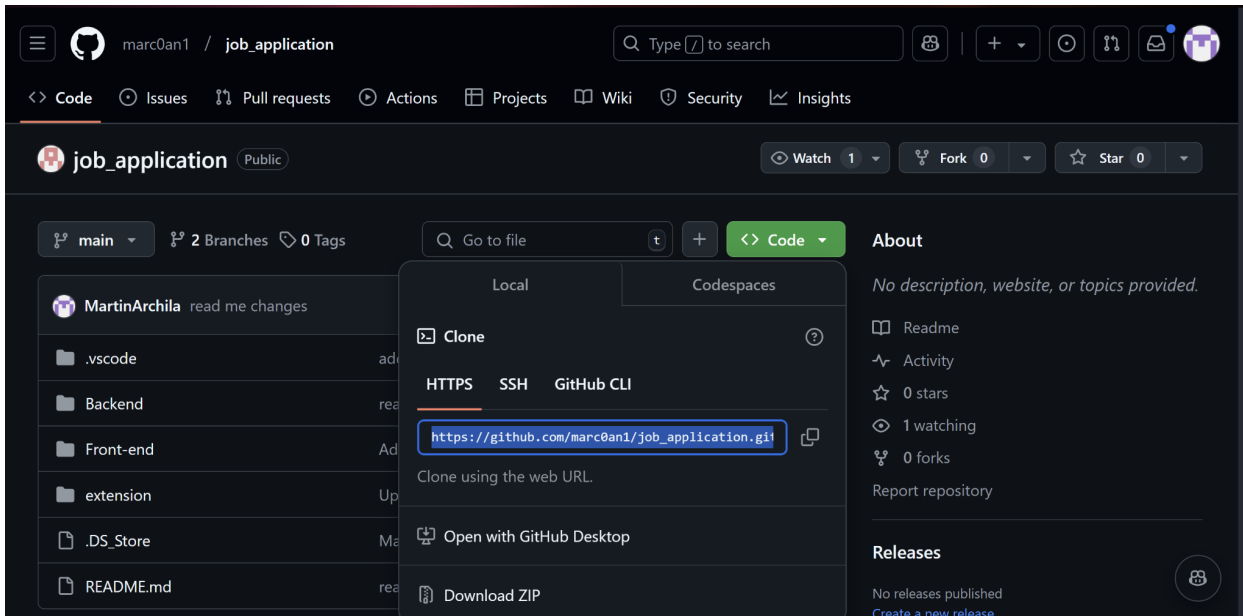


DISCLAIMER:

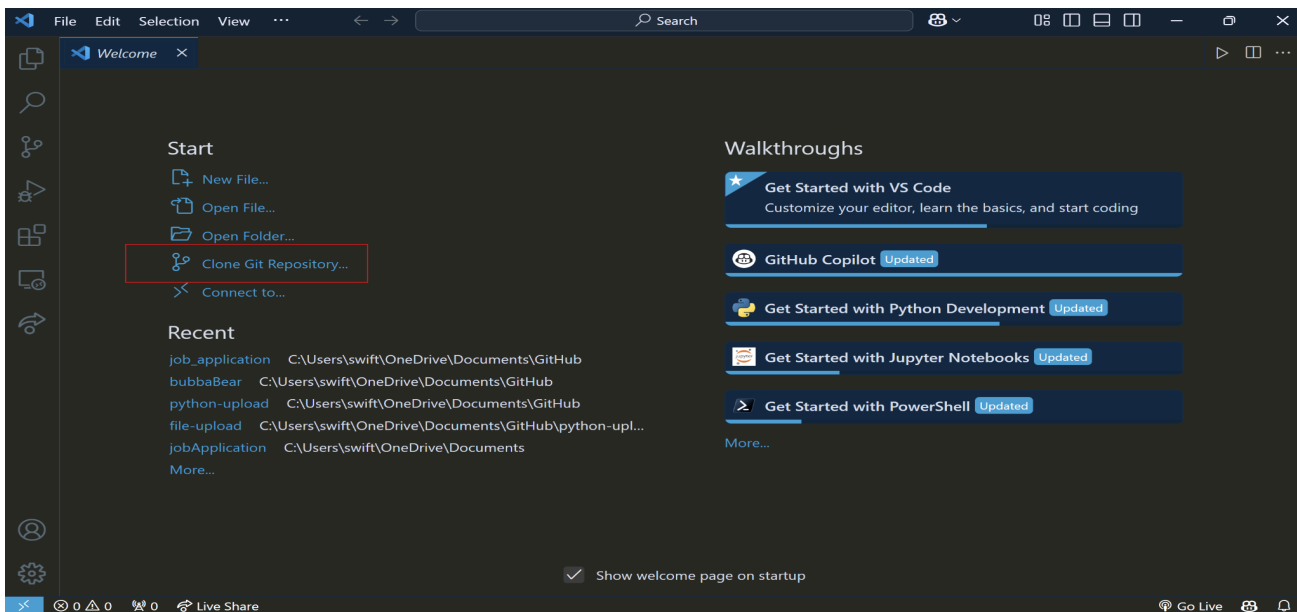
For this purpose will be instructing you through the launching guide for the HTTPS method and cloning the repository to Visual Studio Code as the primary IDE.

Cloning Repository:

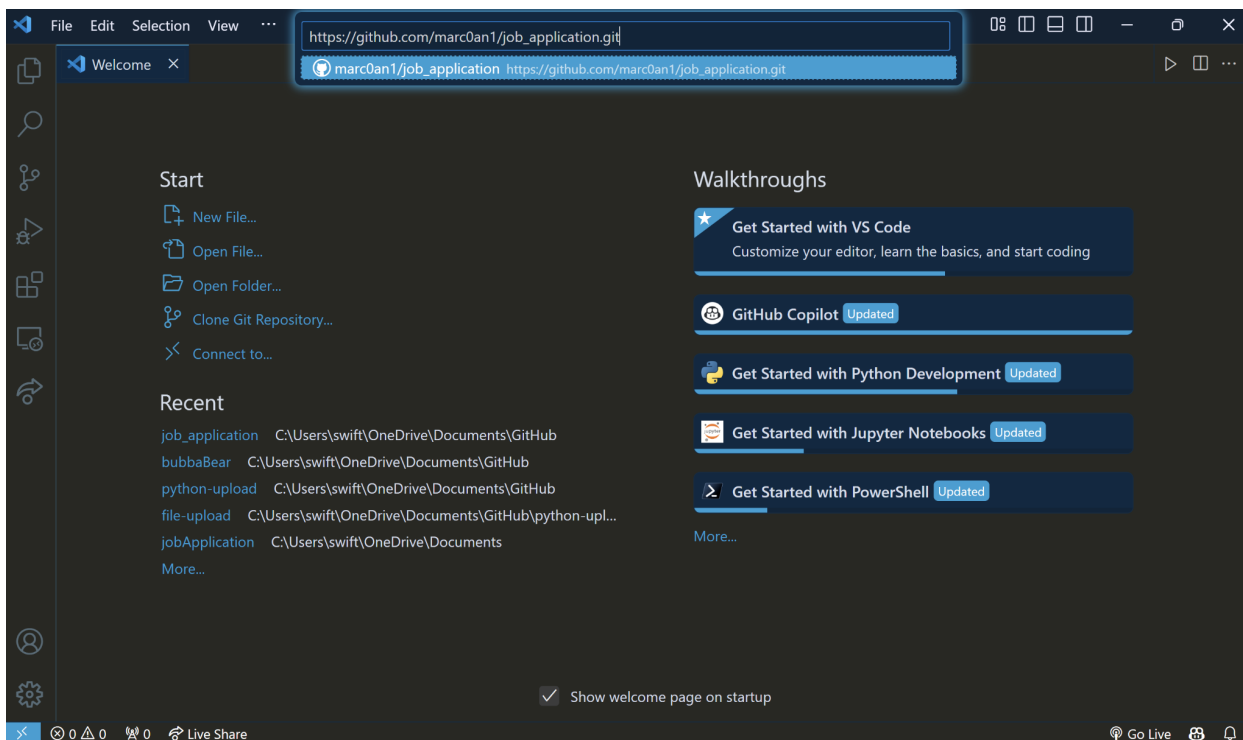
Click the green “Code” button at the top of the GitHub page and select the method of cloning you’d like to use



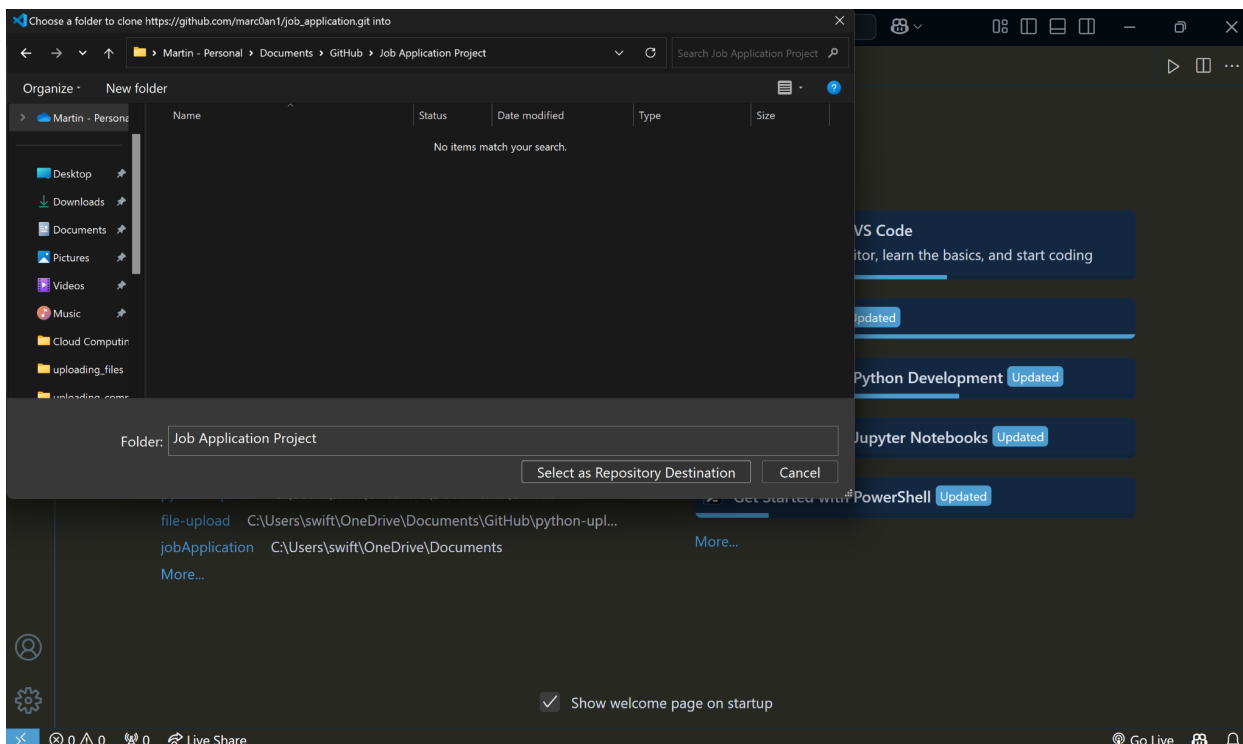
Once you copy the link, open Visual Studio Code to a new window and select “Clone Git Repository”



Select “Github” when prompted and then paste the copied link you got in the first step:



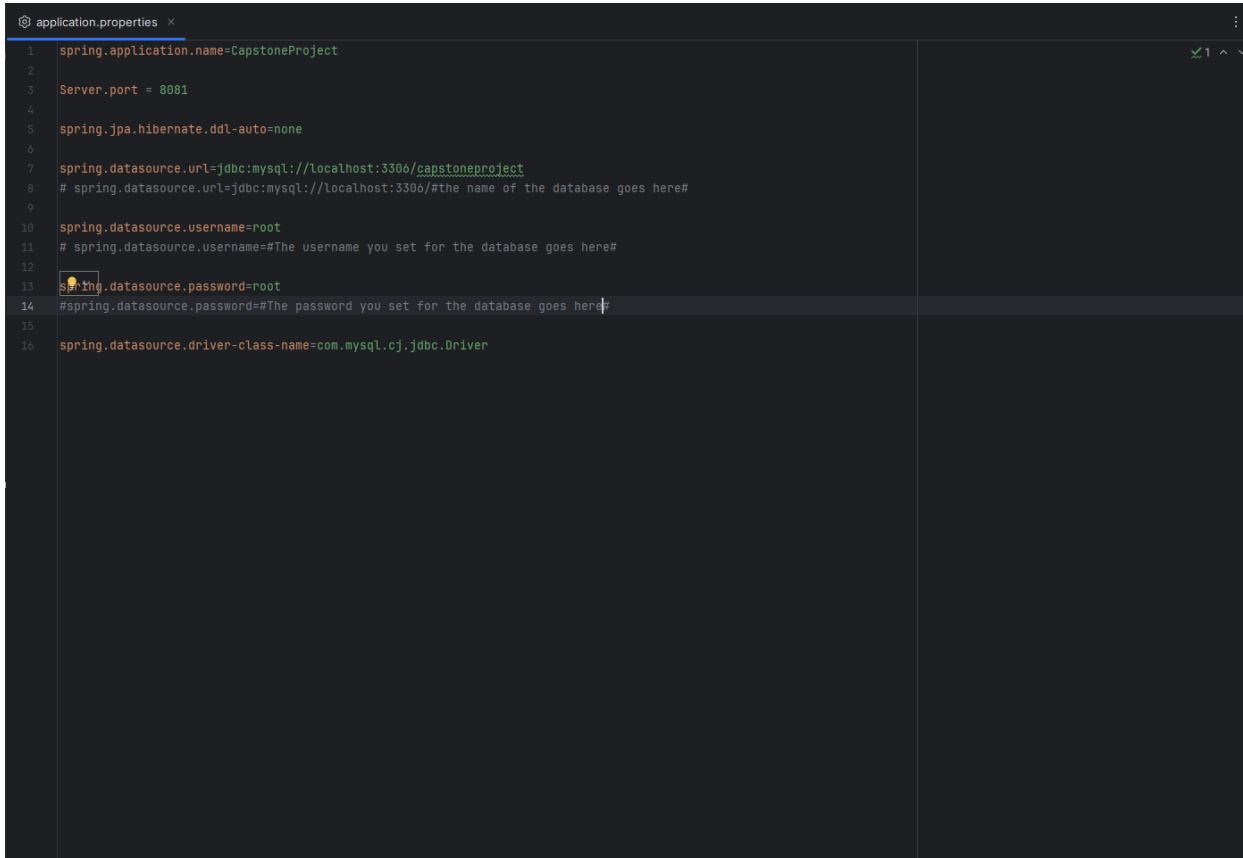
Once you hit Enter, it'll ask you where you'd like to save the repository. Here, I created a new folder where it can be saved:



Once you select the repository destination you're ready to start modifying the code to run.

Database Setup:

Under the application properties file of the backend the command `hibernate.ddl.auto` should be changed accordingly to the necessities if it is the first time that the program is run the command should be set to create. This will automatically create the necessary tables with all its mappings.

A screenshot of a code editor window titled 'application.properties'. The file contains the following configuration:

```
1 spring.application.name=CapstoneProject
2
3 Server.port = 8081
4
5 spring.jpa.hibernate.ddl-auto=none
6
7 spring.datasource.url=jdbc:mysql://localhost:3306/capstoneproject
8 # spring.datasource.url=jdbc:mysql://localhost:3306/#the name of the database goes here#
9
10 spring.datasource.username=root
11 # spring.datasource.username=#The username you set for the database goes here#
12
13 spring.datasource.password=root
14 #spring.datasource.password=#The password you set for the database goes here#
15
16 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

Backend Setup:

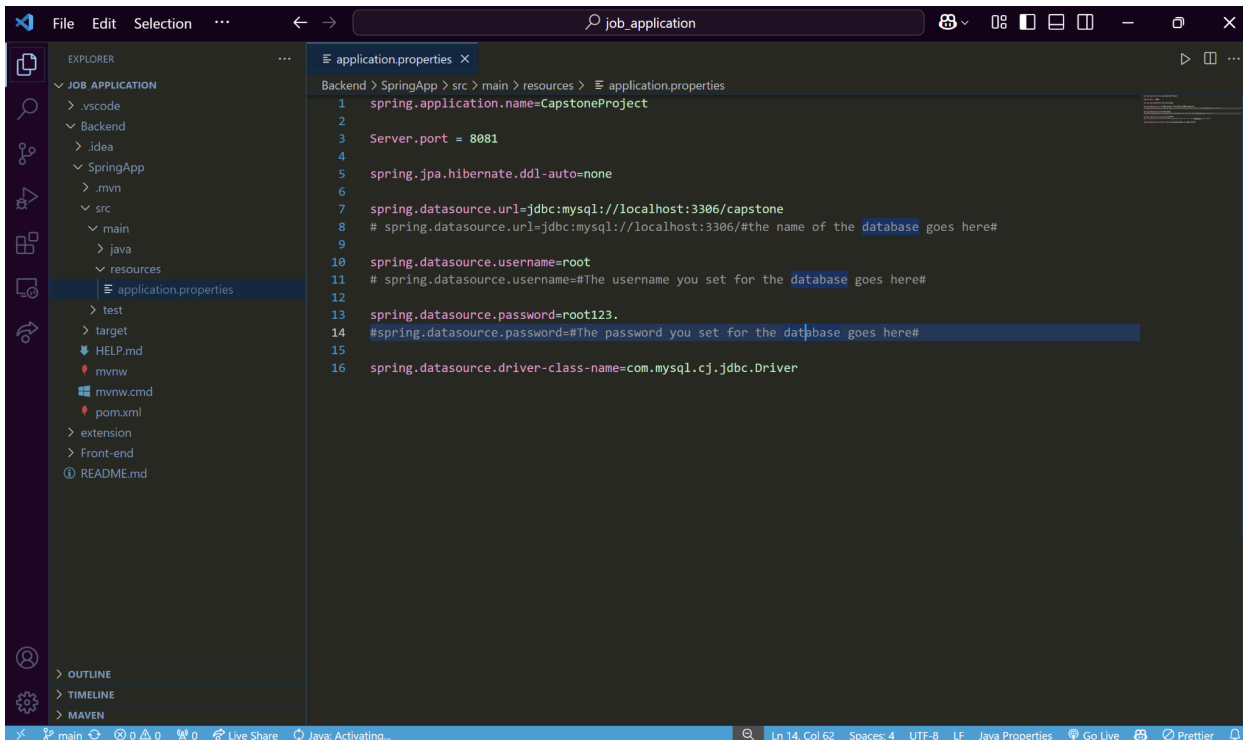
Make sure you've set up the database before continuing forward.

First, you'll want to go to the "application.properties" file to modify some information that is essential for the backend to properly function.

You'll want to change the three fields with comments below them: "url", "username", and "password". For "url", you just have to change "capstone" to the name you gave the database. If you didn't change the name or named it the same don't worry about it. the "3306" might have to be changed if you changed it while creating the database, if not then again don't worry about it.

"Username", here put the username you set to access the database, if you made it the same then don't worry about it.

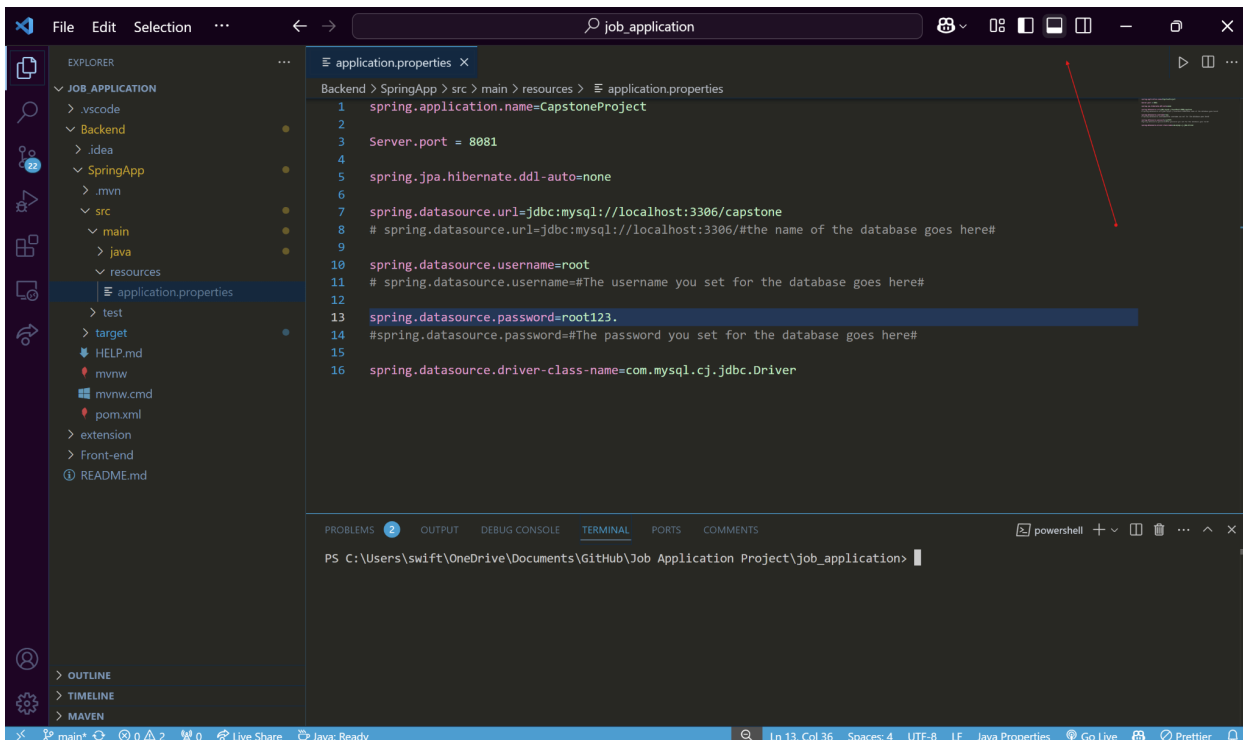
“Password”, here put the password you set to access the database.:



This screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The main editor area shows the file `application.properties` with the following content:

```
Backend > SpringApp > src > main > resources > application.properties
1  spring.application.name=CapstoneProject
2
3  Server.port = 8081
4
5  spring.jpa.hibernate.ddl-auto=none
6
7  spring.datasource.url=jdbc:mysql://localhost:3306/capstone
8  # spring.datasource.url=jdbc:mysql://localhost:3306/#the name of the database goes here#
9
10 spring.datasource.username=root
11 # spring.datasource.username=#The username you set for the database goes here#
12
13 spring.datasource.password=root123.
14 #spring.datasource.password=#The password you set for the database goes here#
15
16 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

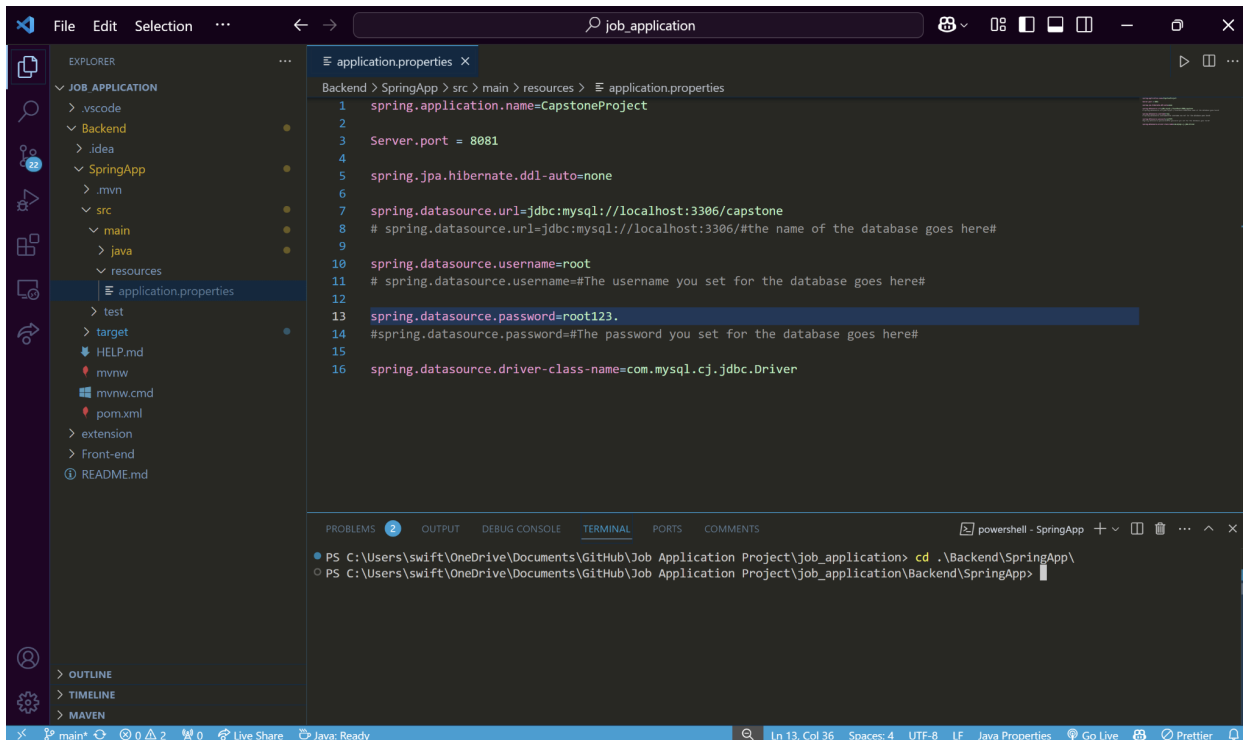
Next, you'll need to create a new Terminal session. You can do so by pressing this icon at the top:



This screenshot shows the same VS Code interface as before, but with a new terminal session opened at the bottom. A red arrow points to the terminal icon in the top-left toolbar. The terminal panel shows a PowerShell prompt at the project directory:

```
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application>
```

You'll want to then `cd` into the backend project using the command below:
“`cd .\Backend\SpringApp\`”



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project structure with folders like .vscode, Backend, SpringApp, and resources. The file application.properties is selected. The main editor displays the content of application.properties, which contains database configuration for a Spring application. The terminal at the bottom shows a PowerShell session where the command `cd .\Backend\SpringApp\` has been executed, changing the current directory to the SpringApp folder within the Backend directory.

```
Backed > SpringApp > src > main > resources > application.properties
1  spring.application.name=CapstoneProject
2
3  Server.port = 8081
4
5  spring.jpa.hibernate.ddl-auto=none
6
7  spring.datasource.url=jdbc:mysql://localhost:3306/capstone
8  # spring.datasource.url=jdbc:mysql://localhost:3306/#the name of the database goes here#
9
10 spring.datasource.username=root
11 # spring.datasource.username=#The username you set for the database goes here#
12
13 spring.datasource.password=root123.
14 #spring.datasource.password=#The password you set for the database goes here#
15
16 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application> cd .\Backend\SpringApp\
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application\Backend\SpringApp>
```

--This is important, if you don't “`cd`” into the backend then the following steps won't work properly--
Once you've confirmed you “`cd`” correctly, you can run the code below to run the Backend:
“`mvn spring-boot:run`”

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows the project structure: JOB_APPLICATION > Backend > SpringApp > src > main > resources > application.properties. The main editor displays the content of application.properties:

```
1 spring.application.name=CapstoneProject
2
3 Server.port = 8081
4
5 spring.jpa.hibernate.ddl-auto=none
6
7 spring.datasource.url=jdbc:mysql://localhost:3306/capstone
8 # spring.datasource.url=jdbc:mysql://localhost:3306/#the name of the database goes here#
9
10 spring.datasource.username=root
11 # spring.datasource.username=#The username you set for the database goes here#
12
13 spring.datasource.password=root123.
14 #spring.datasource.password=#The password you set for the database goes here#
15
16 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

At the bottom, the Terminal pane shows the command prompt output:

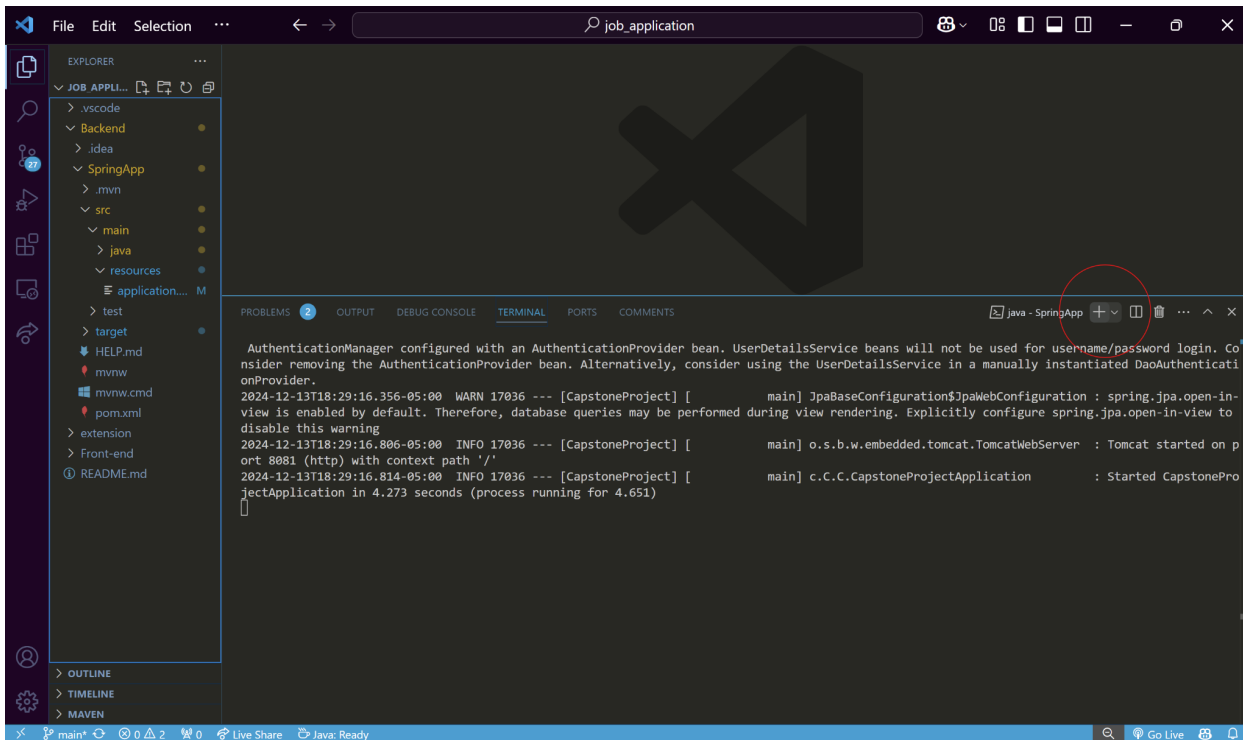
```
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application> cd .\Backend\SpringApp\
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application\Backend\SpringApp> mvn spring-boot:run
```

You'll know it's running correctly if this is the message you get after its done:

```
AuthenticationManager configured with an AuthenticationProvider bean. UserDetailsService beans will not be used for username/password login. Consider removing the AuthenticationProvider bean. Alternatively, consider using the UserDetailsService in a manually instantiated DaoAuthenticationProvider.
2024-12-13T18:29:16.356-05:00 WARN 17036 --- [CapstoneProject] [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2024-12-13T18:29:16.806-05:00 INFO 17036 --- [CapstoneProject] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8081 (http) with context path '/'
2024-12-13T18:29:16.814-05:00 INFO 17036 --- [CapstoneProject] [           main] c.c.C.CapstoneProjectApplication        : Started CapstoneProjectApplication in 4.273 seconds (process running for 4.651s)
```

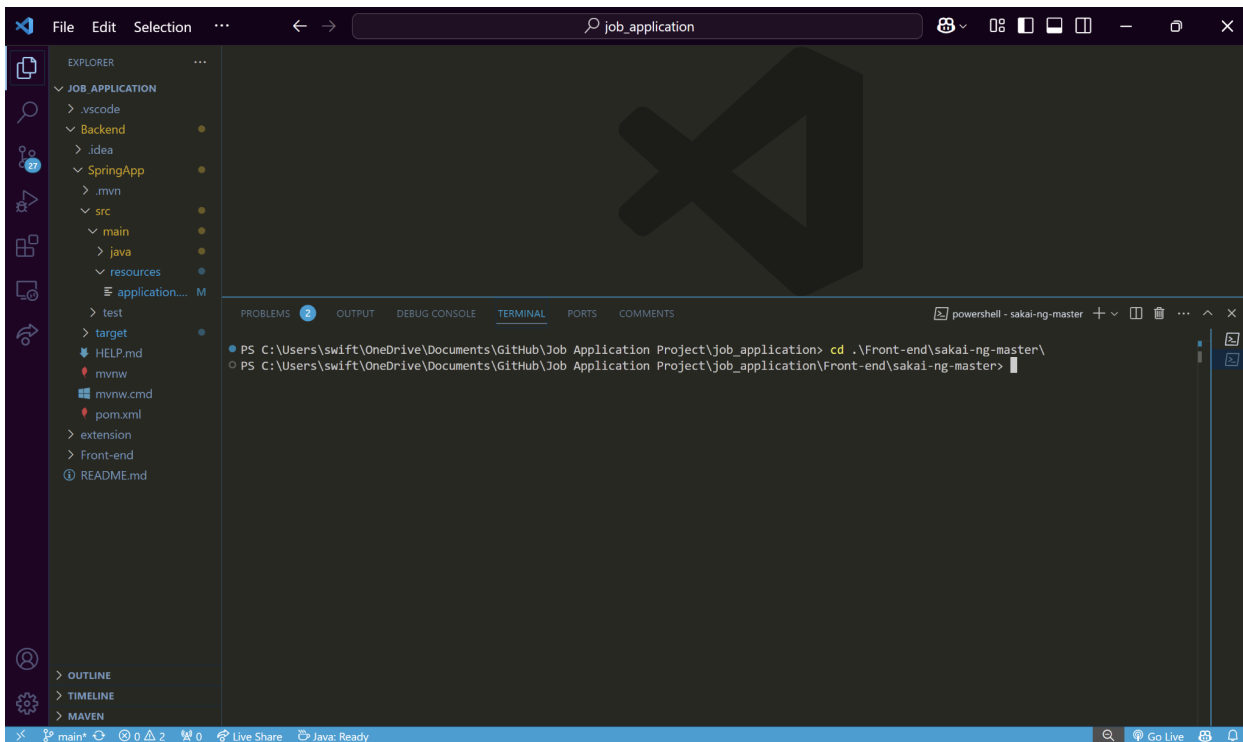
Front-end Setup:

Do not close this terminal session, but instead create a new one using the plus sign on the right of the window:



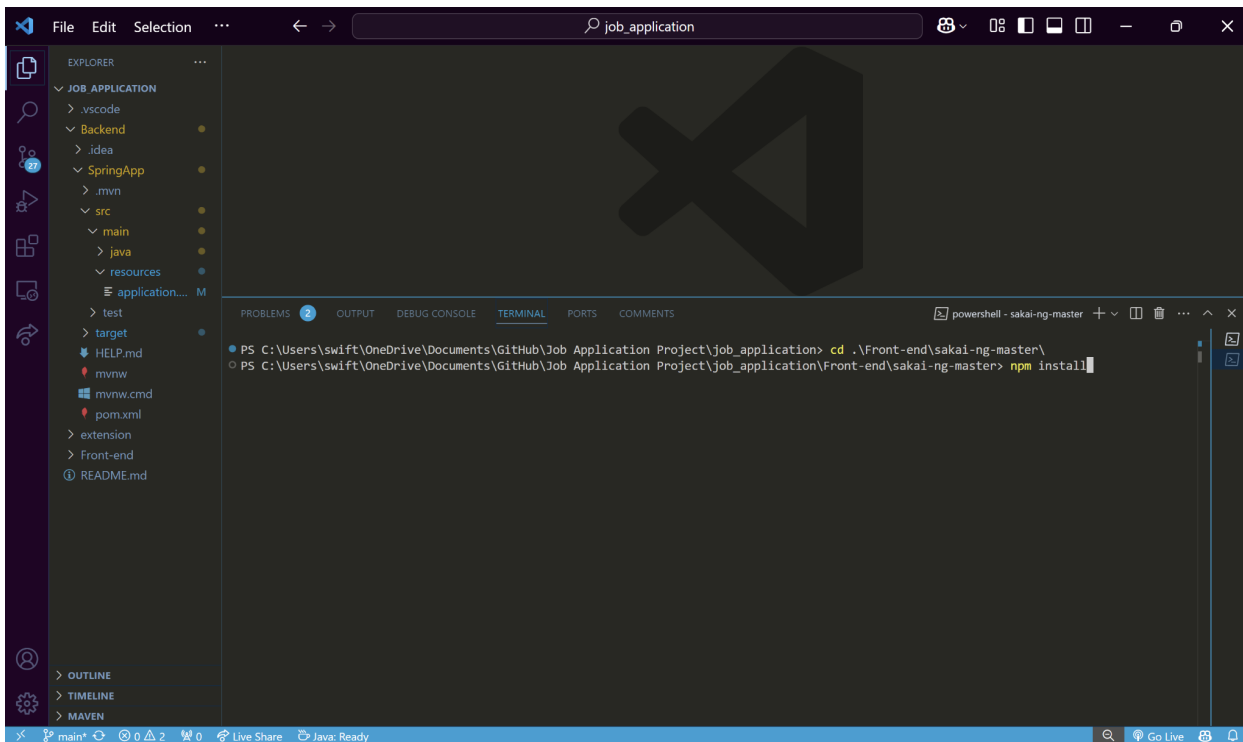
Once you have started a new session, cd into the Frontend using the command below:

“cd .\Front-end\sakai-ng-master”



From here, confirm you’ve “cd” correctly then install npm using the command below:

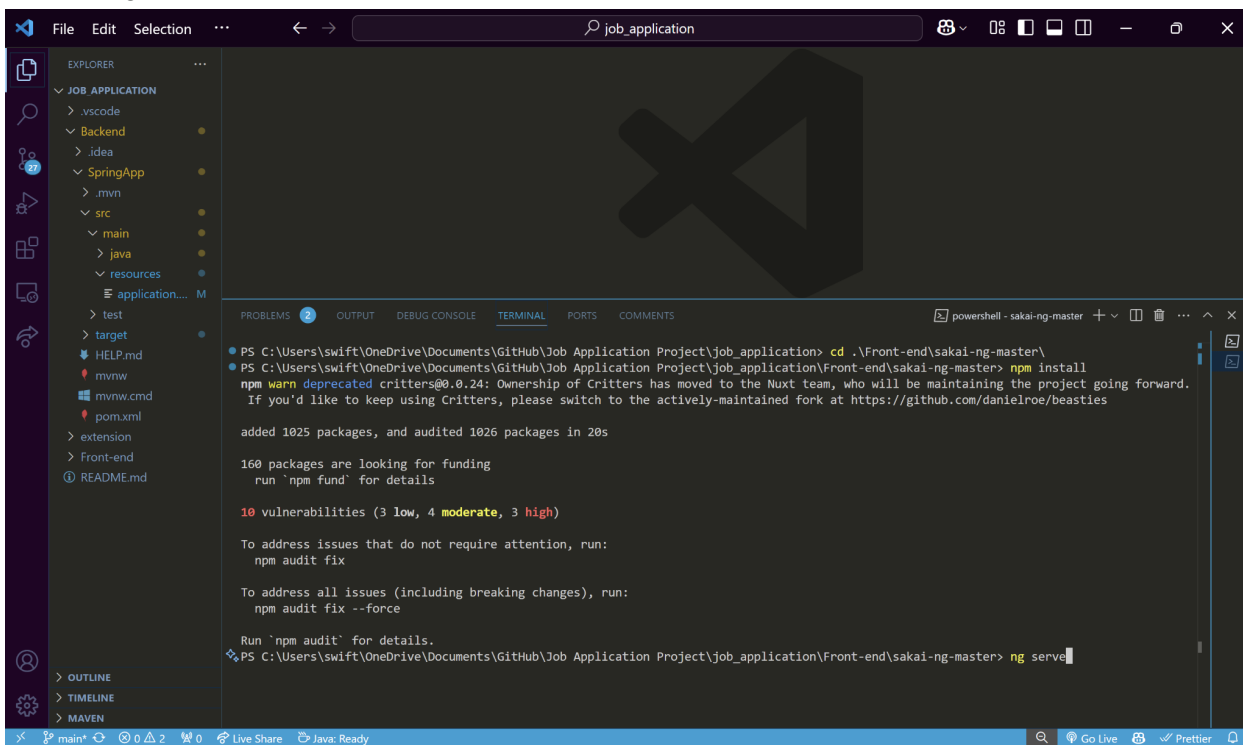
“npm install”



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The main editor area is empty, showing a large dark blue 'X' watermark. The bottom panel contains the TERMINAL tab, which shows the following commands and output:

```
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application> cd .\Front-end\sakai-ng-master\  
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application\Front-end\sakai-ng-master> npm install
```

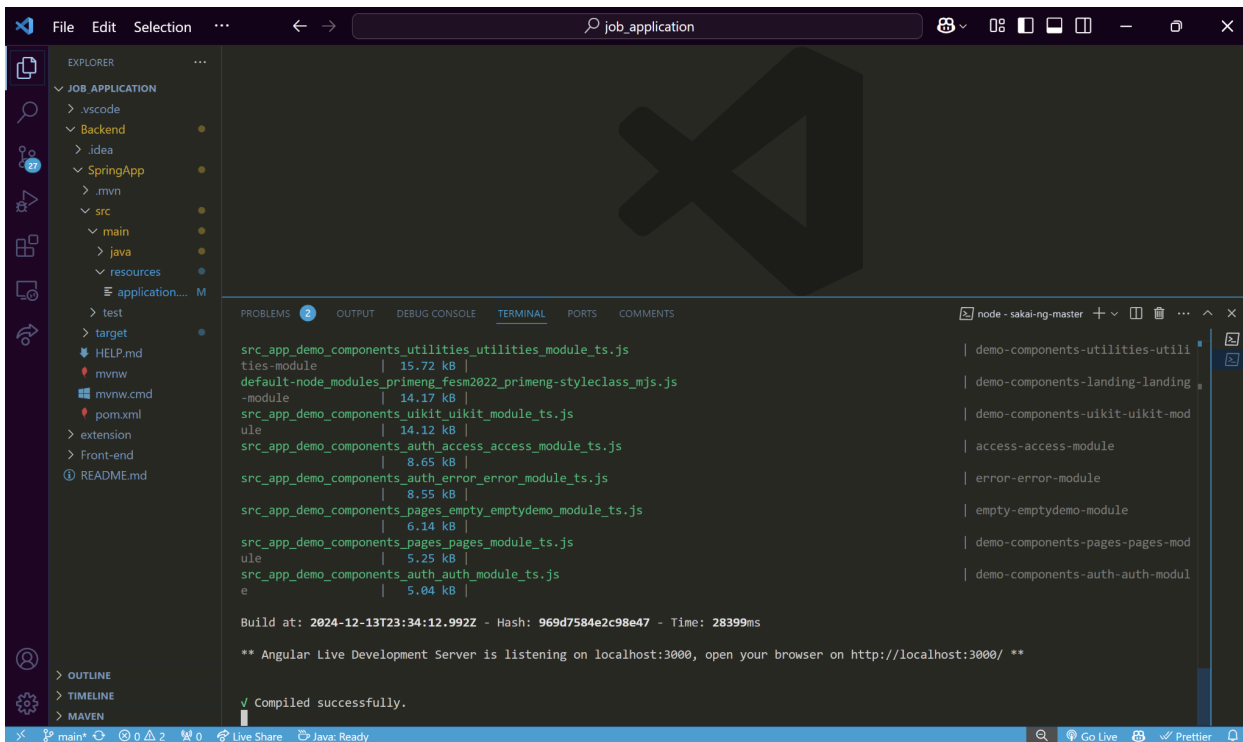
Once npm has been installed correctly, you can run the command below to run the Front-end:
“ng serve”



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The main editor area is empty, showing a large dark blue 'X' watermark. The bottom panel contains the TERMINAL tab, which shows the output of the 'npm install' command and the 'ng serve' command:

```
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application> cd .\Front-end\sakai-ng-master\  
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application\Front-end\sakai-ng-master> npm install  
npm warn deprecated critters@0.0.24: Ownership of Critters has moved to the Nuxt team, who will be maintaining the project going forward.  
If you'd like to keep using Critters, please switch to the actively-maintained fork at https://github.com/danielroe/beasties  
  
added 1025 packages, and audited 1026 packages in 20s  
  
160 packages are looking for funding  
  run `npm fund` for details  
  
10 vulnerabilities (3 low, 4 moderate, 3 high)  
  
To address issues that do not require attention, run:  
  npm audit fix  
  
To address all issues (including breaking changes), run:  
  npm audit fix --force  
  
Run `npm audit` for details.  
PS C:\Users\swift\OneDrive\Documents\GitHub\Job Application Project\job_application\Front-end\sakai-ng-master> ng serve
```

You'll know its running correctly if you get a "localhost" link.



Once you get it, you can either "CTRL + Click" to open up your primary browser to view the site, or copy and paste the link into your preferred browser.

