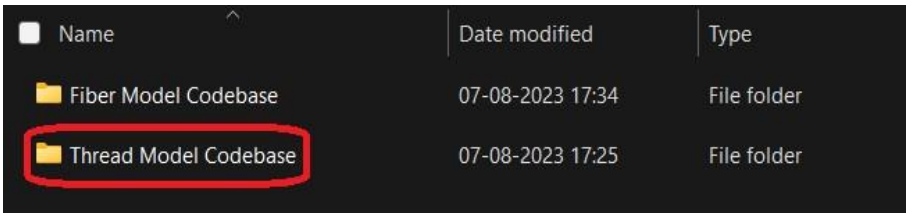
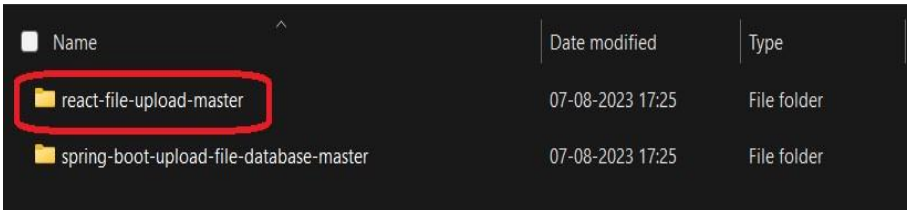
1. **Installation of React-Spring Boot based project.**

   a. At the very first step, extract the project folders named as **Thread Model Codebase** and **Fiber Model Codebase** from the main folder **Codebase**.



   b. Then, clone the React-based front-end project named as **react-file-upload-master** from the **Thread Model Codebase**.



   c. Now, open this project in Visual Studio Code IDE and install all the necessary installations as per the requirement and dependency of the project.

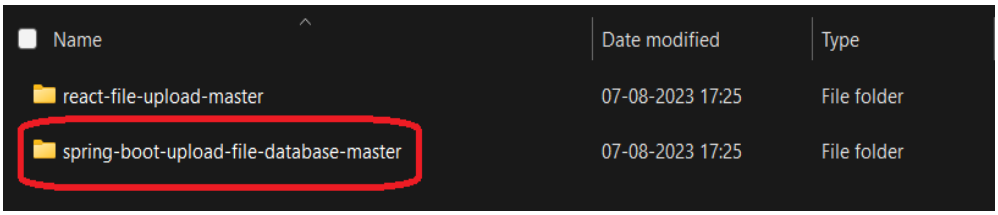
   d. Now once cloned, make sure you run **npm install** download the **node modules** which contains all react dependencies required to run the React project.

   e. The port of the application on which the application shall run is **8081**. On giving the command **npm start** the project will run successfully and will direct you automatically to the link **http://localhost:8081**

```
Compiled successfully!

You can now view react-file-upload in the browser.

  Local:            http://localhost:8081
  On Your Network:  http://192.168.0.229:8081

Note that the development build is not optimized.
To create a production build, use yarn build.

webpack compiled successfully
```

f. But the functionality for uploading the image will not work as it is not yet interacting to the microservice as a Backend API based on Spring Boot framework.

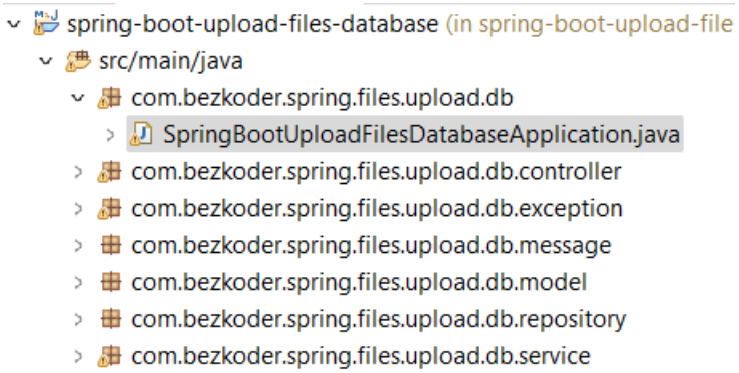
g. Let us now install the Back end-based Spring Boot API Project. For this we will first clone the codebase for the same from the location shown below with the name **spring-boot-upload-file-database-master** for your reference.



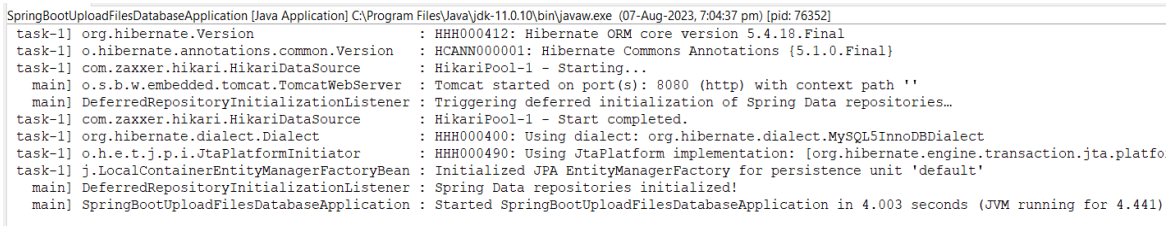
h. Once cloned, import the project as an **existing maven project** in any IDE, IntelliJ, or Eclipse. Now run the command **mvn clean install** in a way to clean the project and reinstall all the dependencies of the project from the repository.

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.2.0:jar (default-jar) @ spring-boot-upload-files-database -
[INFO] Building jar: C:\Users\SURAJ LUMBA\Downloads\spring-boot-upload-file-database-mast
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.2.RELEASE:repackage (repackage) @ spring-boot-upl
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ spring-boot-upload-file
[INFO] Installing C:\Users\SURAJ LUMBA\Downloads\spring-boot-upload-file-database-master\
[INFO] Installing C:\Users\SURAJ LUMBA\Downloads\spring-boot-upload-file-database-master\
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  20.652 s
[INFO] Finished at: 2023-08-07T18:57:45+01:00
[INFO] ------------------------------------------------------------------------
```

i. After this, we will right click on the file shown below in the snap and will click on **Run as Java Application.**

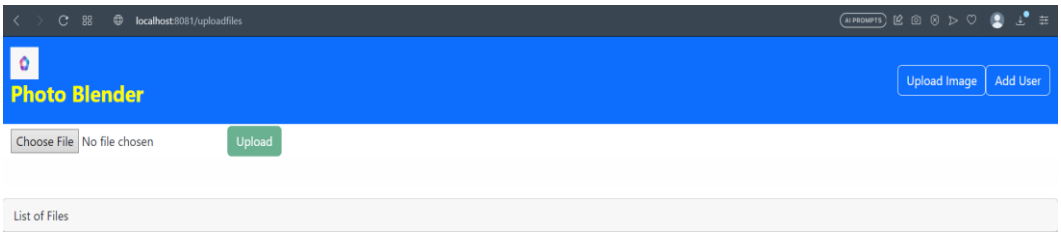


j. After this, the application will run and will be up for the availability of the microservice. The spring boot application will be running on the port **8080. We will get the below shown logs once the application will be up.**



k. Once the application is up, we may go to the front-end application accessibility, that is, from **step e.**

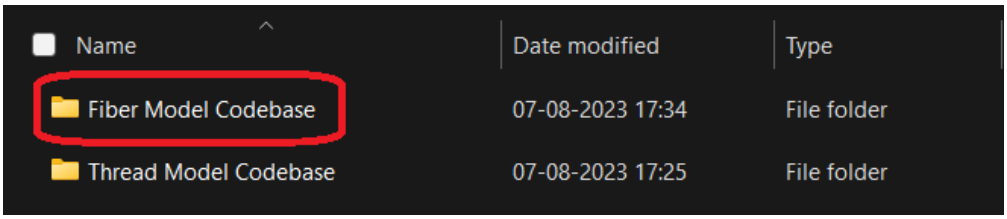
l. Now click on the upload image button as shown below in the snap attached below.



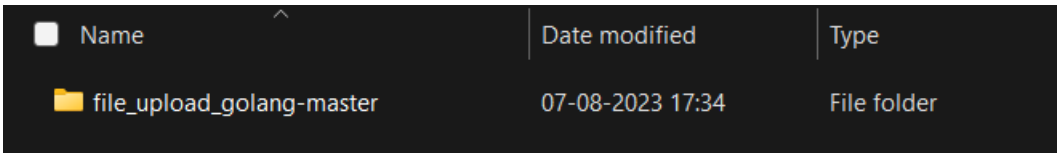
m. Then click on the choose file and select any image file of your choice and press upload. The file then shall be persisted successfully.

2. **Installation of Golang Fiber based project.**

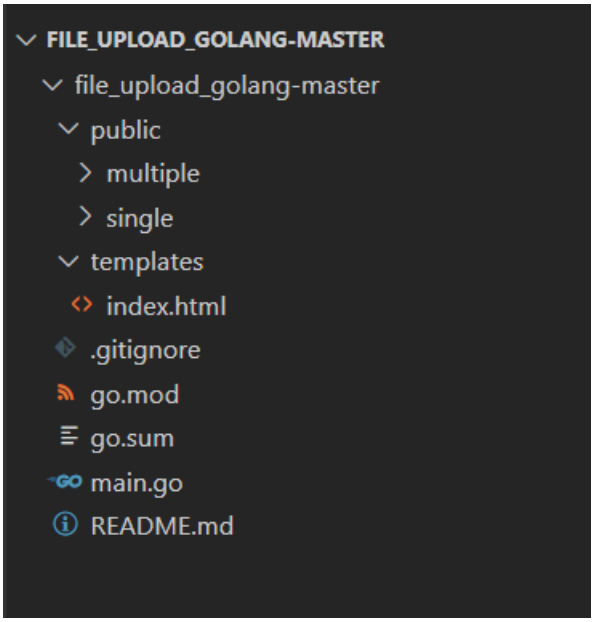
a. Here, from the extracted project folders named as **Thread Model Codebase** and **Fiber Model Codebase** from the main folder **Codebase**. Click on the **Fiber Model Codebase.**



b. Now select the folder name as **file_upload_golang-master** and clone the same in your local memory.

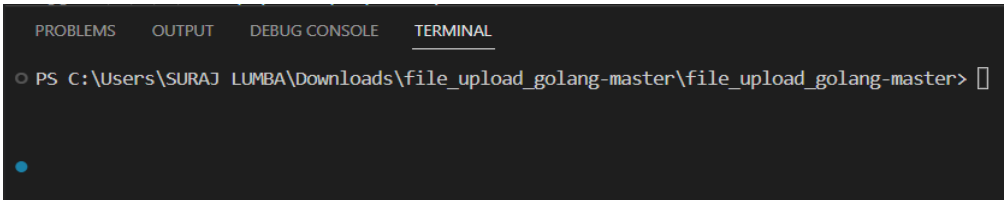


c. After cloning the project, open the same in the Visual Studio Code IDE. **Now make sure all the required libraries as well as the dependencies are all installed.**



d. Above here is the structure of the project. The project running port is set to **8000**.

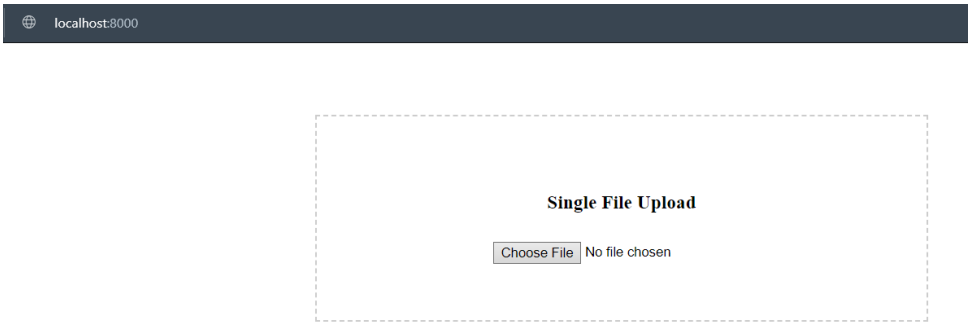
e. Now, open the terminal of the visual studio IDE and point it to this project as shown below:



f. Now, we need to run this Golang project, for which we need to run the **main.go** file.

g. The syntax to run the above file is **go run main.go** which will be executed in the terminal.

h. We can see in the logs that the application is up running and is hosted on the port 8000 with the link **http://localhost:8000**



i. Now we will click on choose File and upload here. Then the file shall be persisted in the database successfully.