**CAR RENTAL SYSTEM**

**ReadMe**

1. **Prerequisites:**

* Before you start, make sure you have the following tools and knowledge:
  + Java Development Kit ([JDK](https://docs.oracle.com/en/java/javase/17/install/installation-jdk-macos.html" \l "GUID-2FE451B0-9572-4E38-A1A5-568B77B146DE)), [Nodejs](https://nodejs.dev/en/learn/how-to-install-nodejs/)
  + Integrated Development Environment (IDE): [IntelliJ IDEA](https://www.jetbrains.com/help/idea/installation-guide.html), [Visual Studio Code](https://code.visualstudio.com/docs/setup/setup-overview).
  + Tools: Apache [Maven](https://maven.apache.org/install.html), [Github](https://github.com/git-guides/install-git), [Docker](https://docs.docker.com/engine/install/), [Docker Desktop](https://docs.docker.com/desktop/install/mac-install/), [Postman](https://www.postman.com/downloads/).
  + Database: [MySQL](https://www.mysql.com/downloads/), MySQL Workbench

1. **Run and Test Backend Application:**
2. **Clone source code:**

* Refer to the Github repository

<https://github.com/luan-tran-89/cs490>

* You can use the github command or Github Desktop to checkout the source code. Please refer the link [Cloning a repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository)

1. **Access MySQL database and create a schema**

* Open MySQL Workbench and create a conenction

A screenshot of a computer

Description automatically generated

* Create the car-rental schema:

**CREATE SCHEMA car-rental;**

**A screenshot of a computer

Description automatically generated**

1. **Setup Zookeeper and Kafka (Ignore this step if using docker-compose)**

* You can refer the links:
  + [Apache Kafka Installation on Mac](https://medium.com/@Ankitthakur/apache-kafka-installation-on-mac-using-homebrew-a367cdefd273)
  + [Install Apache Kafka on Windows](https://www.conduktor.io/kafka/how-to-install-apache-kafka-on-windows/)
* Start Zookeeper and Kafka: refer the [link](https://kafka.apache.org/quickstart)

1. **Run code in local:**

* Open the source code in IntelliJ, you will see the project structure as below:

**A screenshot of a computer

Description automatically generated**

* To run the code in your local, you need to update your datasource in the application.properties in each module.
  + To initialize the databases in the first time, you neen to update:

**spring.jpa.hibernate.ddl-auto=create**

**spring.sql.init.mode=always**

A screenshot of a computer program

Description automatically generated

* Reload the maven and start projects by order:
  + Service-discovery
  + Api-gateway
  + Others services: auth-service, user-service, car-fleet-service, rental-service, and payment-service
* To start a application, you can open the xxxApplication file and right click then choose Run.

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

* After the application started successfullyA screenshot of a computer

  Description automatically generated

1. **Check the Eureka:**

* **Access the URL: http://localhost:8888/**

**A screenshot of a computer

Description automatically generated**

1. **Test the service**

* Open the [swagger ui](http://localhost:8889/swagger-ui/index.html) and you can see the documentation of each service

**A screenshot of a computer

Description automatically generated**

* Try to call the login token API:
  + Enter the body
  + Execute the request, you will receive the reponse as below:

A screenshot of a computer

Description automatically generated

1. **Deploy applications to Docker:**
2. **Build the project by using maven**

* Go to the project and open terminal, and run the command

**mvn clean install**

A screenshot of a computer

Description automatically generated

1. **Create docker images:**

* Create a docker hub by using the link

https://hub.docker.com/

* Go to each service folder and run the related commands:

docker build --tag <profile\_name>/service-discovery:0.0.1 .

docker build --tag <profile\_name>/api-gateway:0.0.1 .

docker build --tag <profile\_name>/auth-service:0.0.1 .

docker build --tag <profile\_name>/user-service:0.0.1 .

docker build --tag <profile\_name>/car-fleet-service:0.0.1 .

docker build --tag <profile\_name>/rental-service:0.0.1 .

docker build --tag <profile\_name>/payment-service:0.0.1 .

* Push the images to docker hub

docker push <profile\_name>/service-discovery:0.0.1

docker push <profile\_name>/api-gateway:0.0.1

docker push <profile\_name>/auth-service:0.0.1

docker push <profile\_name>/user-service:0.0.1

docker push <profile\_name>/car-fleet-service:0.0.1

docker push <profile\_name>/rental-service:0.0.1

docker push <profile\_name>/payment-service:0.0.1

**A screenshot of a computer

Description automatically generated**

1. **Run docker-compose to deploy applications to the local docker**

* Open terminal in the / Car-Rental-Portal folder and run the command:

docker-compose up -d

**A black rectangle with white text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. **Test the service by using Postman:**

* Open the Postman application and import the /Car-Rental-Portal/ Car-Rental-postman.json

**A screenshot of a computer

Description automatically generated**

* Add global variables as below:

**A screenshot of a computer

Description automatically generated**

* Try to call APIs by Postman

A screenshot of a computer

Description automatically generated

1. **Additional**

* If you want to set up to add images to AWS S3, please do some steps below:

1. **Create an AWS account:**

https://aws.amazon.com/vi/resources/create-account/

1. **Create a AWS S3 bucket:**

**A screenshot of a bucket

Description automatically generated**

* Disable Block all public access

**A screenshot of a computer

Description automatically generated**

1. **Add Bucket policy after creating Bucket successfully:**

* Go to the Permissions tab of thebucket:

A screenshot of a computer

Description automatically generated

* Update the bucket policy to allow Get/Delete/Put objects

**A screenshot of a bucket policy

Description automatically generated**

1. **Generate Access Key**

* Open Security credentials page

**A screenshot of a computer

Description automatically generated**

* Create access key in the Access keys

**A screenshot of a computer

Description automatically generated**

* Enter description tag and Create access key

**A screenshot of a computer

Description automatically generated**

* After creating successfully, please keep the **Access key** and **Secret access** key

**A screenshot of a computer

Description automatically generated**

1. Update the ../resources/application.yml in the car-fleet-service

* Enter the accessKey, secretKey, and bucketName to the file

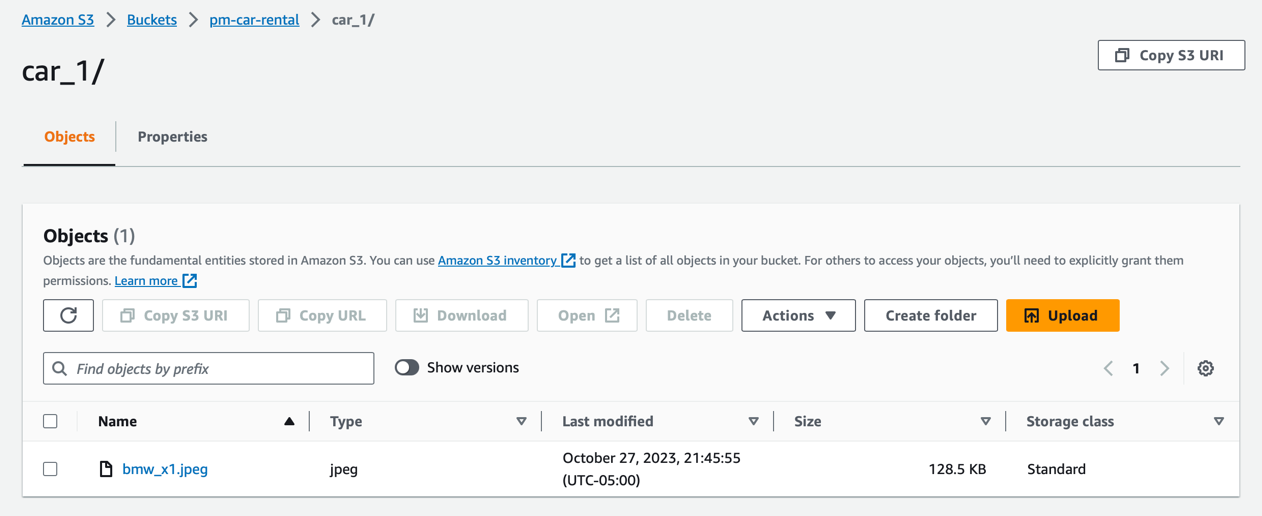
A screenshot of a computer

Description automatically generated

1. **Test the API by using Postman:**

A screenshot of a computer

Description automatically generated



**A screenshot of a computer

Description automatically generated**

* Click to the Object URL and see the image has been updated to AWS S3.

A car parked on a road

Description automatically generated

1. **Setting Up & Running the React Application:**
2. **Installing Necessary Software**

Install Node.js:

* **Go to the Node.js website:** [**https://nodejs.org**](https://nodejs.org)
  + Download the LTS (Long Term Support) version.
  + Follow the installation instructions and complete the installation process.
  + Install Visual Studio Code (VS Code):
* **Go to the VS Code website:** [**https://code.visualstudio.com/**](https://code.visualstudio.com/)
  + Download the version appropriate for your operating system (Windows, MacOS, Linux).
* **Follow the installation instructions and complete the installation process.**

1. **Setting Up the Project**

* Download/Clone the React Application:
* If the project is shared with you as a zip file, unzip it in a location you prefer.
* **Open the Project in VS Code**:
  + Open VS Code.
  + Go to File > Open Folder (or similar menu options) and navigate to where you have unzipped or cloned the project. Select the folder and click Open.
* **Open a Terminal in VS Code:**
  + In VS Code, you can open a terminal by going to the top menu and selecting Terminal > New Terminal.

1. **Installing Dependencies**

* In the terminal that you just opened inside VS Code, type the following commands. This will install all the necessary dependencies for the project, including Axios, Material-UI, Yup, and JWT.
* Run the command

**npm install**

* Wait for the installation process to complete.

**4. Running the Project**

- Once all the dependencies are installed, in the same terminal, type the following command:

npm start