# Graphical user interface, application, Word Description automatically generated

# Screen Shot GUI

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface, website

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated

# Docker

8761

## -javaagent:lombok.jar

Place the Lombok.jar in STS folder and and add the above line in .ini file of sts

The serve command requires to be run in an Angular project, but a project definition could not be found. (just check you directory. You must be inside the anugular project to run ng serve, npm update)

## Angular

<https://loizenai.com/angular-8-springboot-jwt-authentication/>

##### Technologies

– Angular 6  
– RxJS 6

<http://localhost:4200/home>

1. ng help
2. To update : npm update
3. To run : ng serve

Graphical user interface, application, Word

Description automatically generated

# MySQL

1. Go to mysql client terminal
2. Enter password : pass@123
3. Command : create database pixogram;
4. Command : show databases;
5. Command : use pixogram;
6. Command to delete database : drop database dbName;

spring.datasource.url=jdbc:mysql://localhost:3306/pixogram

spring.datasource.username=root

spring.datasource.password=pass@123

spring.jpa.generate-ddl=true

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Command **:**  CREATE TABLE roles (id bigint(11) primary key AUTO\_INCREMENT

, name varchar(60) );

**Command** : CREATE TABLE user\_roles (user\_id bigint(20) primary key, role\_id bigint(20) );

**Command** : CREATE TABLE users (id bigint(20) primary key AUTO\_INCREMENT

, email varchar(50) UNIQUE, name varchar(50), password varchar(100) , username varchar(50) UNIQUE);

Command : show tables;

INSERT INTO roles(name) VALUES('ROLE\_USER');

INSERT INTO roles(name) VALUES('ROLE\_PM');

INSERT INTO roles(name) VALUES('ROLE\_ADMIN');

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

## Postman

## http://localhost:8080/api/test/user

A screenshot of a computer

Description automatically generated

## <http://localhost:8080/api/auth/signin>

## Request

{

"username": "anushkhathakur",

"password" : "anushkhathakur"

}

## Response:

{

"accessToken": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhbnVzaGtoYXRoYWt1ciIsImlhdCI6MTYyODM0MzM0NSwiZXhwIjoxNjI4NDI5NzQ1fQ.0j7fq0QKKjCCV78mETE6XaWRzRLVmXhPmbz0GAI0Sni8vbNvTb\_w24Np152W-9mb8OPpQ7zMBLSEYnkzNwz7bA",

"tokenType": "Bearer"

}A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot, indoor, computer

Description automatically generated

## Request

{

"username": "wrongUser",

"password" : "anushkhathakur"

}

## Response

{

"timestamp": "2021-08-07T13:36:58.645+0000",

"status": 401,

"error": "Unauthorized",

"message": "Error -> Unauthorized",

"path": "/api/auth/signin"

}

A picture containing text, screenshot, computer, indoor

Description automatically generated

## <http://localhost:8080/api/auth/signup>

## Request

{

"name": "Anushkha Thakur",

"username": "anushkhathakur",

"email": "anushkhathakur@gmail.com",

"role": ["user"],

"password": "anushkhathakur"

}

## Response

User registered successfully!

A picture containing text, screenshot, computer, computer

Description automatically generated

A picture containing text, screenshot, computer, computer

Description automatically generated

# User Interface

<http://localhost:4200/home>

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

http://localhost:4200/signup

Graphical user interface, application, Word

Description automatically generated

## Postman media service

http://localhost:9001/uploadSingleFile

No need to add any headers ,

Just add select file and put one json in another form data

## request

{

"userId": "2",

"mediaTitle": "mediaTitle",

"desc": "desc",

"fileName":"fileName"

}

## Response

{

"message": "Uploaded the file successfully: "

}

Graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence