**Data-Drive-System\_GDrive APIS**

**Packages I have used:-**

* **golang.org/x/crypto/bcrypt (password encryption and decryption)**
* **github.com/dgrijalva/jwt-go (for authentication)**
* **github.com/spf13/viper (for environment variable)**
* **github.com/go-sql-driver/MySQL**
* **github.com/natefinch/lumberjack (to print logs)**
* **gorm.io/driver/MySQL**
* **gorm.io/gorm**

**API endpoints:-**

**[ http://localhost:8080/g-drive ]**

## **1. User Authentication**

### **1.1 Create User**

* **Endpoint: /signup**
* **Method: POST**
* **Description: Creates a new user.**
* **Response:**
  + **200 Ok→ User registered successfully**
  + **400 Bad Request → Missing or invalid fields**

### **1.2 Login User**

* **Endpoint: /login**
* **Method: POST**
* **Description: Authenticates a user and returns a JWT token.**
* **Response:**
  + **200 OK → Returns JWT token**
  + **401 Unauthorized → Invalid credentials**

## **2. File Management**

### **2.1 Upload File**

* **Endpoint: /uploadFile**
* **Method: POST**
* **Description: Uploads a new file or updates an existing file.**
* **Response:**
  + **200 OK → File uploaded successfully**
  + **400 Bad Request → Invalid file**

### **2.2 Get Files by User ID**

* **Endpoint: /getFilesByUserId**
* **Method: GET**
* **Description: Retrieves all files for a given user.**

### **2.3 Delete File by User ID**

* **Endpoint: /deleteFileByUserId**
* **Method: DELETE**
* **Description: Deletes a file by marking it as deleted (delete\_flag = 1).**
* **Query Parameters:**
  + **file\_id (Required)**
* **Response:**
  + **200 OK → File deleted successfully**
  + **404 Not Found → File does not exist**

## **3. Folder Management**

### **3.1 Create Folder by User ID**

* **Endpoint: /createFolderByUserId**
* **Method: POST**
* **Description: Creates a new folder under a specified parent folder.**
* **Response:**
  + **201 Created → Folder created successfully**
  + **400 Bad Request → Invalid data**

### **3.2 Get Folders**

* **Endpoint: /getFolders**
* **Method: GET**
* **Description: Retrieves all folders in a hierarchical structure.**

### **3.3 Delete Folder by User ID**

* **Endpoint: /deleteFolderByUserId**
* **Method: DELETE**
* **Description: Deletes a folder and marks it as deleted (delete\_flag = 1).**
* **Response:**
  + **200 OK → Folder deleted successfully**
  + **404 Not Found → Folder does not exist**

### **Notes:**

* **All POST, GET, and DELETE requests require authentication via a JWT token in the headers.**
* **delete\_flag = 1 is used instead of actual deletion for data integrity.**