

1. Register User

- **URL:** /register/
- **Method:** POST
- **Request Format:** multipart/form-data
- **Request Body:**
 - **id** (string): The user's ID. (Required)
 - **image** (file): The user's image file. (Required)
- **Response Body:**
 - **Success (201 Created):**

```
{
  "message": "Data inserted successfully"
}
```

- **Errors:**
 - **400 Bad Request:**

```
{
  "error": "ID and image are required"
}
```

- **409 Conflict:**

```
{
  "error": "User already exists"
}
```

- **500 Internal Server Error:**

```
{
  "error": "Error message describing the issue"
}
```

2. Get Registered Users

- **URL:** /register/
- **Method:** GET
- **Response Body:**
 - **Success (200 OK):**

```
{
  "message": "Hello, world!",
  "data": [
    {
      "id": "user1",
      "timestamp": "2023-07-23T12:34:56"
    },
    {
      "id": "user2",
      "timestamp": "2023-07-23T12:34:56"
    }
  ]
}
```

- **Errors:**
 - **500 Internal Server Error:**

```
{
  "error": "Error message describing the issue"
}
```

3. Verify User

- **URL:** /verify/

- **Method:** POST
- **Request Format:** multipart/form-data
- **Request Body:**
 - **id** (string): The user's ID. (Required)
 - **image** (file): The image file to verify. (Required)
- **Response Body:**
 - **Success (200 OK):**

```
{
  "message": "Image received successfully!",
  "recognition": "Recognized User ID",
  "verified": true
}
```

- **Errors:**
 - **400 Bad Request:**

```
{
  "error": "Image and ID are required",
  "verified": false
}
```

```
{
  "error": "Invalid User",
  "verified": false
}
```

- **500 Internal Server Error:**

```
{
  "error": "Error message describing the issue",
  "verified": false
}
```

4. Delete User

- **URL:** /register/
- **Method:** DELETE
- **Request Format:** multipart/form-data
- **Request Body:**
 - **id** (string): The user's ID. (Required)
- **Response Body:**
 - **Success (204 No Content):**

```
{
  "message": "Data deleted successfully",
  "files_deleted": ["list_of_deleted_files"]
}
```

- **Errors:**
 - **400 Bad Request:**

```
{
  "error": "ID is required"
}
```

- **404 Not Found:**

```
{
  "error": "User not found"
}
```

Models

- **User Model:**
 - `id` (string): User ID (Primary Key)
 - `timestamp` (datetime): Timestamp of registration
-

Setup Instructions

1. Install Required Packages

Ensure you have all the required packages installed by using the following command:

```
pip install -r requirements.txt
```

2. Configure MySQL Database in `settings.py`

Add the following configuration to your `settings.py` file to set up the MySQL database connection:

```
# settings.py

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'libraryregistration',
        'USER': 'your_mysql_user',
        'PASSWORD': 'your_mysql_password',
        'HOST': 'localhost', # Or the address of your MySQL server
        'PORT': '3306',      # Default MySQL port
    }
}

# Ensure you have 'django.db.backends.mysql' installed
# You can install it using: pip install mysqlclient
```

3. Create the `users` Table in MySQL

Execute the following SQL command in your MySQL database to create the `users` table:

```
CREATE DATABASE IF NOT EXISTS libraryregistration;

USE libraryregistration;

CREATE TABLE users (
    id VARCHAR(255) PRIMARY KEY,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

4. Make and Apply Migrations

Run the following commands to create and apply the necessary migrations:

```
python manage.py makemigrations
python manage.py migrate
```