# 📌 Social App API Documentation

This document provides an overview of the REST API endpoints for the **Social App**, built using **FastAPI** and **MySQL**.

## 🔗 Base URL

http://localhost:8000/

## 📖 Authentication

Currently, no token-based authentication is implemented. All endpoints rely on passing user identifiers (user\_id, email, etc.) as query parameters.

## 📂 Endpoints

### 🧑 User Management

#### 🔹 Sign Up

**POST** /sign-up - **Parameters**: name, email, password - **Response**:

{  
 "id": 1  
}

#### 🔹 Sign In

**GET** /sign-in - **Parameters**: email, password - **Response**: User ID if authenticated, 404 if not found

#### 🔹 Get User Data

**GET** /get-user\_data - **Parameters**: user\_id - **Response**: JSON object containing user data

#### 🔹 Delete User

**DELETE** /delete-user - **Parameters**: user\_id - **Response**:

"user with id = X deleted successfully"

### 🔔 Notifications

#### 🔹 Send Notification

**POST** /send-notification - **Parameters**: receiver\_id, content, friend\_request\_id (optional) - **Response**: { "id": <notification\_id> }

#### 🔹 Delete Notification

**DELETE** /delete-notification - **Parameters**: notification\_id

#### 🔹 Mark Notification as Read

**POST** /mark-notification-as-read - **Parameters**: notification\_id

#### 🔹 Get Notifications

**GET** /get-notifications - **Parameters**: user\_id

### 👥 Friends

#### 🔹 Send Friend Request

**POST** /send-friend-request - **Parameters**: sender\_id, receiver\_id

#### 🔹 Add Friend

**POST** /add-friend - **Parameters**: sender\_id, receiver\_id

#### 🔹 Delete Friend

**DELETE** /delete-friend - **Parameters**: sender\_id, receiver\_id

#### 🔹 Show Friends

**GET** /show-friends - **Parameters**: user\_id - **Response**: List of friend IDs

### 👤 Followers

#### 🔹 Follow User

**POST** /follow - **Parameters**: sender\_id, receiver\_id

#### 🔹 Unfollow User

**DELETE** /unfollow - **Parameters**: sender\_id, receiver\_id

#### 🔹 Get Followers

**GET** /get-followers - **Parameters**: user\_id

#### 🔹 Get Followings

**GET** /get-followings - **Parameters**: user\_id

### 📝 Posts

#### 🔹 Create Post

**POST** /create-post - **Parameters**: user\_id, content

#### 🔹 Delete Post

**DELETE** /delete-post - **Parameters**: post\_id

#### 🔹 Get All Posts

**GET** /get-all-posts

#### 🔹 Get User Posts

**GET** /get-user-posts - **Parameters**: user\_id

#### 🔹 Share Post

**POST** /share-post - **Parameters**: user\_id, post\_id

#### 🔹 Delete Post Share

**DELETE** /delete-post-share - **Parameters**: user\_id, post\_id

#### 🔹 Like Post

**POST** /like-post - **Parameters**: user\_id, post\_id

#### 🔹 Delete Post Like

**DELETE** /delete-post-like - **Parameters**: user\_id, post\_id

### 💬 Comments

#### 🔹 Create Comment

**POST** /create-comment - **Parameters**: user\_id, post\_id, content

#### 🔹 Delete Comment

**DELETE** /delete-comment - **Parameters**: comment\_id, user\_id, post\_id

#### 🔹 Get Post Comments

**GET** /get-post-comments - **Parameters**: post\_id

#### 🔹 Like Comment

**POST** /like-comment - **Parameters**: user\_id, comment\_id, comment\_user\_id, post\_id

#### 🔹 Delete Comment Like

**DELETE** /delete-comment-like - **Parameters**: user\_id, comment\_id, comment\_user\_id, post\_id

### 🤖 AI Chat

#### 🔹 Send Prompt

**POST** /send-prompt - **Parameters**: user\_id, content - **Response**:

{  
 "response": "AI generated reply"  
}

#### 🔹 Delete Message

**DELETE** /delete-message - **Parameters**: user\_id, message\_id

## ⚠️ Error Handling

* **400**: Bad Request (e.g., record not found, already exists)
* **404**: Not Found (e.g., user not found)
* **500**: Internal Server Error (e.g., database errors)

## 📌 Notes

* All database connections use MySQL (mysql.connector).
* IDs are auto-generated via GetNewID function.
* Some queries use raw SQL (⚠️ potential SQL injection risk).