Table of Content

1. Introduction
   1. Overview

* Use-Case Diagram
  1. Technologies Used

1. Setup and Installation
   1. Prerequisites

* Version of Express.js, node.js & MongoDB
* Modules/Middleware Used
  1. Installation Guide
  2. Running the Application

1. Architecture and Design
   1. System Architecture

* Structure Diagram
  1. Database Schema
* ER Diagram
* Data Dictionary
  1. System Overview
* State Diagram
* Assumption and Limitation (1 album for each user. Max 2 raw photos)
* Brief Description of each Process

1. User Management
   1. Sign Up
   2. Login and Logout
   3. User Profile Edit
   4. Un-subscription
2. Photo Management
   1. Photo Upload
   2. Photo Retrieval and Display
3. APIs and Routes
   1. RESTful API Endpoints
   2. Route Handlers
4. Security Control
   1. Password Hashing
   2. Protection of Photo from Direct Access
5. Future Enhancements (???)

# Copilot Suggestion for Section 6

**Overview:**

* Description of what RESTful APIs are and their importance in a web application.
* Explanation of how your photo-sharing application uses RESTful APIs to handle HTTP requests and responses.

**Common HTTP Methods:**

* **GET:** Retrieve data from the server.
* **POST:** Send new data to the server.
* **PUT:** Update existing data on the server.
* **DELETE:** Remove data from the server.

**API Endpoint Structure:**

* **Base URL:** The root URL for all API endpoints, e.g., https://yourapp.com/api/v1.
* **Endpoint Path:** The specific resource paths, e.g., /photos, /users, /comments.

**Example Endpoints:**

1. **User Authentication:**
   * **POST /api/v1/register:** Register a new user.
   * **POST /api/v1/login:** Authenticate a user and return a token.
2. **Photo Management:**
   * **GET /api/v1/photos:** Retrieve a list of all photos.
   * **POST /api/v1/photos:** Upload a new photo.
   * **GET /api/v1/photos/:id:** Retrieve a single photo by ID.
   * **PUT /api/v1/photos/:id:** Update details of a specific photo.
   * **DELETE /api/v1/photos/:id:** Delete a specific photo.

# Data Dictionary

Collection: user\_profile

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| login\_id | String | Unique identity used to enter into the application |
| display\_name | String | Name of the user used to show in application |
| password | String | Hash value of secret key used by the user to log-in to system |
| profile\_photo | String | Filename of an image file used by the user as his/her profile picture |

Collection: photo\_collection

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| login\_id | String | Used as identity of album folder owned by the user |
| filename | String | Filename of a photo uploaded by the user |
| ISO | String | Technical parameters of a photo |
| aperture | String |
| shutter | String |
| EV | String |
| likeCount | Integer | Total number of user response like a photo |
| likeBy | Array of string | Login ID of user(s) who response like a photo |
| raw1 | Raw photo object | raw photo used to compose the photo in this record |
| raw2 | Raw photo object |

Object: raw photo

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| filename | String | Filename of a raw photo uploaded by the user |
| ISO | String | Technical parameters of a photo |
| aperture | String |
| shutter | String |
| EV | String |