**Image Gallery Application**

**Project Overview**

The Image Gallery Application is a web-based platform that allows users to upload images, manage them, and view a gallery of uploaded images. The application includes a backend API for handling image storage and metadata management, while the frontend provides a user-friendly interface for uploading, editing and viewing images.

**Technical Design Document**

**Project Architecture**

* **Frontend:** React.js
* **Backend:** ASP.NET Core API
* **Database:** SQL Server (for storing image metadata)
* **Cloud Storage:** Cloudinary

With five models including AppUser, Comments, Images, Passwords and Category

**App Use**

* **Home Page:** Displays a gallery of all uploaded images.
* **Upload Page:** Users can drag and drop images or select files from their local storage to upload.
* **Image Details Page:** Users can view image details, including title, description, and comments. They can also edit these details and manage comments pictures uploaded by that specific user who is logged in.
* **Management Page:** Provides a view for managing all uploaded images, including editing and deleting.

**How the App Runs**

**Backend (ASP.NET Core API)**

1. **Image Upload:** Handles image uploads, storing the image in cloud storage and saving metadata in the database.
2. **Image Management:** Provides endpoints for CRUD operations (Create, Read, Update, Delete) on images and associated metadata.
3. **Comment Management:** Manages comments associated with each image, including adding, updating, and deleting comments.
4. **Authentication:** Manages user authentication during login.

**Frontend (React.js)**

1. **Image Upload Component:** Allows users to upload images, which are then processed by the backend API.
2. **Image Display:** Fetches and displays images from the backend, providing an interactive gallery.
3. **Comments Section:** Enables users to post and manage comments on images.
4. **User Interface:** Provides a responsive and intuitive interface for easy navigation and interaction.

**App Features**

* **Image Upload:** Drag-and-drop functionality and file explorer upload.
* **Image Gallery:** Displays all uploaded images with titles and descriptions.
* **Image Management:** Edit and delete image details.
* **Comments:** Add, edit, and delete comments associated with images.
* **Authentication:** Secure login with JWT token.
* **Error Handling:** Displays user-friendly error messages for validation and upload errors.
* **Success Feedback:** Notifies users of successful uploads and operations.

**Technologies**

* **Frontend:** React.js, Axios (for API communication)
* **Backend:** ASP.NET Core API, C#, Entity Framework Core (for database interaction)
* **Database:** SQL Server
* **Cloud Storage:** (Cloudinary)
* **Authentication:** JWT token