Documentation

TBPPP

Cloudinary-Photos

Submitted By:

Anurag

2215100004

3AF {C.C.V}

Group Members

Anurag

Praveen (2215200019)

Aakash Chaudhary

(2215100001)

Sourav (2215200023)

Photo Album Web App Documentation

About the Project

The Photo Album Web App is a cutting-edge platform for uploading, managing, and editing images. Built with Next.js, Cloudinary, and ShadCN UI, it offers advanced features like AI-powered tagging, image transformations, and a responsive user interface. The app aims to simplify photo organization while providing an enjoyable user experience.

Mission

To create a user-friendly photo management system leveraging AI for tagging, transformations, and organization, ensuring efficiency and ease of use.

Objectives

1. **Image Upload and Management**:
    - Drag-and-drop and widget-based upload options.
    - Image storage and retrieval via Cloudinary APIs.
2. **AI-Powered Features:**
    - Auto-tagging images based on content.
    - Image transformations like cropping, filtering, and background removal.
3. **Favorites and Albums:**
    - Mark favorite photos and organize them into albums.
4. **Responsive Design:**
    - Intuitive and mobile-friendly user interface.
5. **Deployment:**
    - Ensure global accessibility with a robust backend.

Technology Stack

Frontend

- **Next.js**:
    - Why: Powerful React-based framework for server-side rendering.
    - Use Case: Build server components and dynamic pages.
- **ShadCN UI:**
    - Why: Simplifies UI creation with pre-built components.
    - Use Case: Customize buttons, sidebars, and modal dialogs.
- **Tailwind CSS:**
    - Why: Rapid and consistent styling.

- Use Case: Ensure responsive layouts.

Backend

- **Cloudinary:**
  - Why: Centralized media management.
  - Use Case: Handle image uploads, storage, and transformations.
- **Next.js Server Actions:**
  - Why: Secure server-side logic.
  - Use Case: Manage dynamic image fetching and API interactions.

Deployment

- **Frontend Hosting: Vercel**
  - Why: Optimized for Next.js apps.
  - Use Case: Deploy the user interface.
- **Backend Hosting: Vercel Functions**
  - Why: Provide serverless API solutions.
  - Use Case: Deploy API endpoints for interacting with Cloudinary.

Agile Work Plan

Sprint 1: Setup and Foundation (Week 1)

Goal: Establish the foundational structure for the application.

Tasks:

- Configure Next.js, Tailwind CSS, and ShadCN UI.
- Set up a GitHub repository and environment variables.
- Create the initial project skeleton.

Assigned to: Person A (Sourav)

Deliverables:

- Functional app skeleton hosted on a local server.

Sprint 2: Image Upload and Display (Week 2)

Goal: Enable users to upload and view images dynamically.

Tasks:

- Configure Cloudinary for media storage.
- Build drag-and-drop image upload functionality.
- Display uploaded images in a gallery format.

Assigned to: Person B (Praveen)

Deliverables:

- Image upload functionality with a dynamic gallery display.

Sprint 3: Favorites and Albums (Week 3)

Goal: Provide organizational tools for users.

Tasks:

- Implement favorites functionality using Cloudinary tags.
- Build album creation and management features.
- Create a sidebar for navigating albums and favorites using ShadCN components.

Assigned to: Person C (Anurag)

Deliverables:

- Fully functional favorites and album management system.

Sprint 4: Image Editing (Week 4)

Goal: Add advanced image editing features.

Tasks:

- Enable image transformations like cropping, blurring, and resizing.
- Integrate AI-powered background removal and auto-tagging.

Assigned to: Person D (Aakash)

Deliverables:

- Advanced image editing features integrated into the app.

Sprint 5: Responsive Design (Week 5)

Goal: Ensure a seamless user experience across devices.

Tasks:

- Use Tailwind CSS to build responsive layouts.
- Test UI components on various screen sizes and devices.
- **Assigned to:** Person A

Deliverables:

- Fully responsive and mobile-friendly design.

Sprint 6: Testing and Optimization (Week 6)

Goal: Ensure the app is stable and optimized for deployment.

Tasks:

- Test for bugs and fix issues.
- Optimize performance for server-side and client-side operations.

Assigned to: Person B & Person C

Deliverables:

- Bug-free and optimized application.

Sprint 7: Deployment (Week 7)

Goal: Launch the application and ensure global accessibility.

Tasks:

- Deploy the frontend and backend to Vercel.
- Monitor deployment for any issues.

Assigned to: Person D

Deliverables:

- Fully deployed Photo Album web app.

Workflow Overview

1. **Upload Images**: Users upload images via drag-and-drop or widgets.
2. **Gallery Display**: Uploaded images appear dynamically in a gallery.
3. **AI Features**: Images are auto-tagged for easy filtering.
4. **Image Transformations**: Users can edit photos (crop, resize, etc.).
5. **Organize Albums**: Photos can be grouped into albums.
6. **Favorites**: Users can mark and view favorite photos.

Collaboration Notes

- **Communication**: Use Slack for daily updates and troubleshooting.
- **Version Control**: Maintain feature-specific branches on GitHub.
- **Sprint Reviews**: Conduct reviews at the end of each sprint to showcase progress and gather feedback.
- **Documentation**: Update project wiki with technical details and resolved challenges.