

PROJECT REPORT

Shobhit Institute of Engineering & Technology, Meerut
(NAAC 'A' Grade Accredited Deemed-to-be University)

NH-58, Modipuram, Meerut - 250 110

PhotoStream – Cloud-Based Photo Gallery

SESSION: 2025-2026
[T01BCCA0396]

Submitted By:

Name: - Gupta Karan

Course: - BCA 2nd Year

Roll No: - MRT24UGBCA114

Enroll No: - 20240110548

Name: - Mayank Raj

Course: - BCA 2nd Year

Roll No: - MRT24UGBCA127

Enroll No: - 20240110640

Submitted To:
Priyanka Saini
(Assistant Professor)



Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **Gupta Karan** and **Mayank Raj**, students of **BCA 2nd Year, Section B**, have successfully completed the Minor Project titled “PhotoStream – Cloud-Based Photo Gallery” under the guidance of **Ms. Priyanka Saini**, Department of Computer Science and Engineering, during the academic session **2025–2026**.

Project Title: PhotoStream – Cloud-Based Photo Gallery

Date of Completion: Nov 10, 2025

Name of the External Examiner

Signature

Name of the Internal Examiner

Signature

ACKNOWLEDGMENT

We express our heartfelt gratitude to everyone who contributed to the successful completion of our project.

We sincerely thank **Ms. Priyanka Saini, Assistant Professor, Department of Computer Science & Engineering**, for her continuous support, motivation, and valuable guidance throughout the development of this project. Her insights and suggestions played a crucial role in shaping the project and improving the overall quality of our work.

We are also thankful to our Head of Department, respected faculty members, seniors, and classmates who provided support and suggestions at various stages of this project.

TABLE OF CONTENTS

<u>SL.NO</u>	<u>DESCRIPTION</u>	<u>PAGE NO</u>
	Certificate	i
	Acknowledgement	ii
1.	Introduction	1
2.	Problem Statement	2
3.	Project Architecture	3
4.	Technology used in the Project	4
5.	Project Screenshots	5-6
6.	Conclusion	7
7.	References	8

INTRODUCTION

PhotoStream – Cloud-Based Photo Gallery is a modern web application designed to store, organize, and access images securely from anywhere. The project aims to build a **Django-based photo gallery system** where users can upload photos, categorize them, and view them through an intuitive and responsive interface.

In today's digital world, users need a secure and always-available place to store and manage their memories. This cloud-enabled platform makes it possible for users to manage their entire photo collection online without worrying about physical storage or device loss.

Key Features of PhotoStream:

1. Image Upload

Users can upload photos with titles, descriptions, tags, and categories.

2. Category-wise Sorting

Photos can be categorized for easy browsing.

3. Responsive UI

Optimized for mobile, tablet, and desktop viewing.

4. Date-wise Filtering

Users can view transactions by day, week, or month to analyze their spending patterns.

5. Cloud Storage Integration

Secure storage of photos with high availability.

6. User-Friendly Dashboard

Displays photo collections, search tools, and filters.

7. Secure Access

Only authorized users can upload/delete photos.

8. Scalable Architecture

Built with Django's modular MVC structure.

9. Multi-Media Support

Users can upload and view both high-quality Images and Videos.

10. Video Playback

Integrated HTML5 video player for seamless playback of uploaded video content directly in the dashboard.

11. Smart Categorization

Separate collections for "Photos" and "Videos" for better organization.

PROBLEM STATEMENT

Title: Efficient Cloud-Based Image Management with PhotoStream Gallery Application

Problem:

Users often struggle to manage scattered photos stored across devices, leading to disorganization, limited accessibility, and storage issues. PhotoStream aims to provide a centralized, secure, and cloud-based solution for uploading, organizing, and accessing images efficiently through a structured gallery with categorization and smart retrieval features, users struggle with:

1. Disorganized photos spread across devices.
2. Limited device storage.
3. Difficulty finding photos quickly.
4. Lack of centralized, secure access.
5. No categorization or tagging support.
6. No cross-device synchronization.

Objective:

To develop a secure, scalable, cloud-based photo gallery application that provides a central platform for uploading, organizing, and viewing images through an intuitive web interface.

Key Goals:

- Provide centralized cloud storage for photos
- Enable easy upload, organization, and retrieval
- Offer category-based albums and smart filtering
- Ensure secure user authentication and privacy
- Deliver a responsive, user-friendly interface

Impact:

- Enhances photo organization and accessibility.
- Reduces device storage dependency.
- Protects personal images through secure cloud storage.
- Improves user convenience with cross-platform access.
- Supports efficient searching and memory management.

FEASIBILITY STUDY

1. Technical Feasibility

The project is technically feasible as it uses Python (Django), which is open-source and has robust libraries for file handling. The frontend uses standard HTML5/CSS3 which is supported by all modern browsers.

2. Operational Feasibility

The user interface is designed to be intuitive (similar to Google Photos or Instagram), requiring no special training for users to upload or view media.

3. Economic Feasibility

The project is built using open-source technologies (VS Code, SQLite, Django), incurring zero licensing costs, making it highly economically viable.

PROJECT ARCHITECTURE

1. Client Interface (Frontend Pages)

- Users interact with the system using responsive web pages created with HTML, CSS and basic JavaScript.
- Main pages include: Home/Gallery View, Upload Photo Page, Category-wise View, and Login/Signup Page.

2. Django Templates & URL Routing

- Django templates render dynamic content such as photo lists, categories and user-specific galleries.
- The urls.py file maps user requests (URLs) to appropriate views like gallery view, upload view, login view, etc.

3. Views and Business Logic (Django Views)

- Django views handle the core logic of the application:
 - processing image upload forms
 - saving photo details
 - filtering photos by category or user
- Views also manage redirects, success messages and error handling.

4. Database & Models Layer

- Django models define the structure of data stored in the database, such as:
 - Photo model (image file, title, description, category, upload date, user)
 - Category model (category name, description)
- SQLite (or any configured RDBMS) is used to store metadata of uploaded photos securely.

5. Media Storage (Local / Cloud Storage)

- Uploaded images are stored in a dedicated media directory or cloud storage based on configuration.
- MEDIA_ROOT and MEDIA_URL settings in Django manage how files are stored and accessed.
- Templates load these image URLs to display photos in the gallery.

6. User Authentication & Access Control

- Django's authentication system manages user registration, login and logout.
- Only authenticated users can upload, manage or delete their photos.
- Each photo entry is linked to the user who uploaded it to maintain privacy and ownership.

7. Admin Panel (Django Admin Interface)

- The Django admin panel allows the administrator or faculty to:
 - view all uploaded photos
 - manage categories
 - manage users
- It provides a quick way to monitor and control the overall application data

8. Overall Workflow

- User logs into PhotoStream
- User uploads photos with title, description and category
- System stores image file in media storage and details in database
- User can later view, filter or search photos in the gallery based on categories or other fields

TECHNOLOGY USED

we are developing a web project built using **Django**, along with **HTML, CSS, and basic JavaScript** for the interface.

Frontend

- HTML5 – Structure of the web pages
- CSS3 – Styling and responsive layout
- JavaScript – Dynamic UI interactions & validation

Backend

- Python (**Django Framework**) – Core logic, routing, models, authentication

Database

- SQLite3 (default Django database)

Tools & Platforms

- GitHub (Version control)
- Python pip packages
- Cloud storage (optional)

Hardware Requirements

- Minimum 2 GB RAM
- Dual-core processor
- 100 MB storage
- Stable internet connection

WORK-FLOW

1. Authentication Module

- **User Action:** Enters credentials via Login or registers via Signup.
- **System Process:** Verifies credentials against the database.
- **Outcome:** Grants access to private routes or prompts error; ensures session security.

2. Dashboard & Data Retrieval

- **Trigger:** Successful login.
- **System Process:** Asynchronously fetches image metadata from the database and visual assets from media storage.
- **Outcome:** Renders the main dashboard with a responsive grid gallery of all user uploads.

3. Asset Upload Workflow

- **User Action:** Selects file(s) and inputs metadata (Title, Description, Album).
- **System Process:**
 - **Database:** Saves metadata (text/tags).
 - **Storage:** Saves the physical image file to a media server/bucket.
- **Outcome:** Displays updates with a success message and the new image appears in the feed.

4. Organization & Album Management

- **Feature:** Users can create custom albums (e.g., Work, Travel).
- **Action:** Move or copy existing photos into specific albums for better categorization.

5. Exploration & Viewing

- **Modes:** "View All" (Master stream) or "Album View" (Categorized).
- **Detail View:** Clicking a thumbnail triggers a detailed modal or page view showing full-resolution images and descriptions.

6. Search & Query Engine

- **Input:** User types keywords or selects date ranges.
- **Filtering:** System filters results by:
 - Upload Date
 - Album Assignment
 - Keywords (Title/Description)
- **Outcome:** Dynamic refresh of the gallery showing only matching assets.

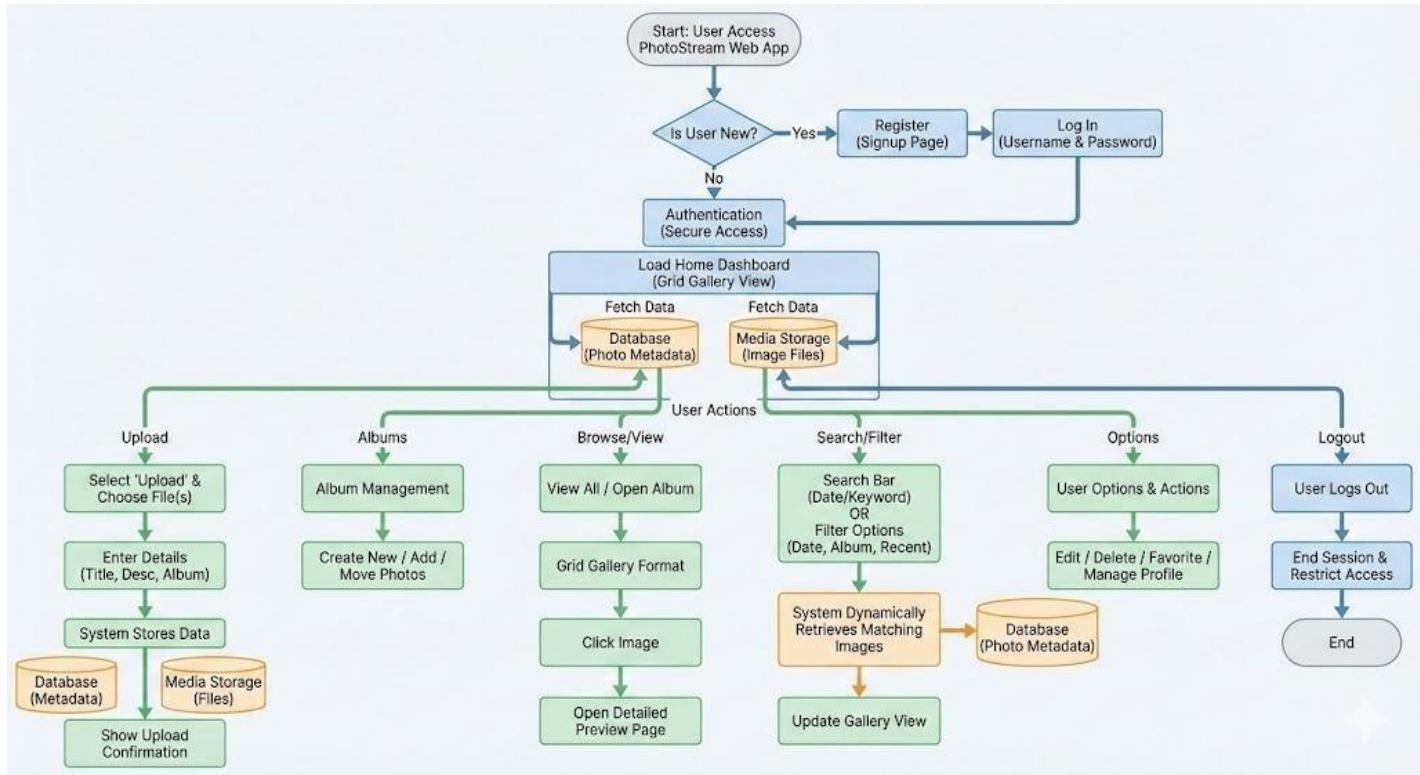
7. Asset Management

- **CRUD Operations:** User can Edit details, Delete images (removes from DB and Storage), or Toggle "Favorite" status.

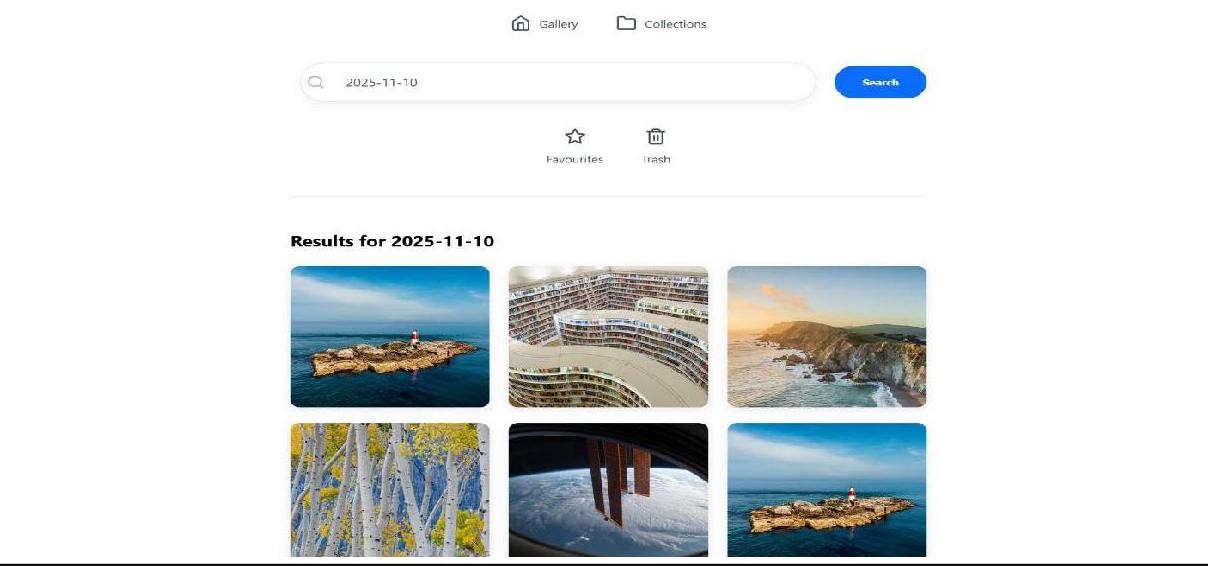
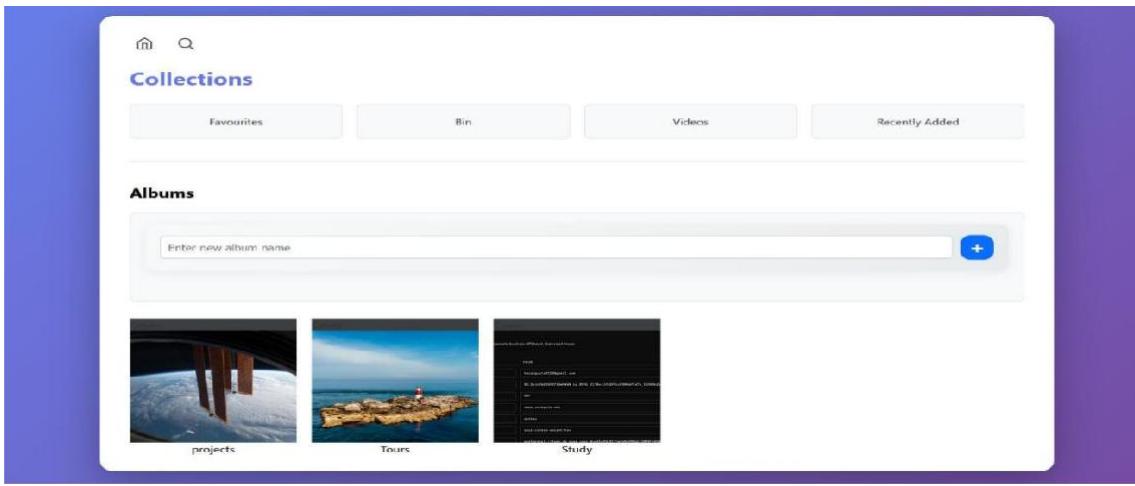
8. Session Termination

- **Action:** User clicks Logout.
- **Outcome:** Server destroys the session token and redirects to the public landing page.

FLOW DIAGRAM



PROJECT SCREENSHOTS



Login

Username:

Password:

Don't have an account? [Sign Up](#)

[Forgot Password?](#)

Sign Up

- A user with that username already exists.

Username:

Email:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Already have an Account! [login here](#)

Forgot Your Password?

Enter your email address below, and we'll email instructions for setting a new one.

Email:

Password Reset Request

Hello karangupta,

You are receiving this email because you (or someone else) requested a password reset for your account on our PhotoGallery website.

If you did not request a password reset, please ignore this email and your password will remain unchanged.

© 2025 PhotoGallery. All rights reserved.

CONCLUSION

The **PhotoStream – Cloud-Based Photo Gallery** project provides a modern and efficient way to store and organize photos online. Using Django, the system ensures secure storage, easy navigation, and responsive performance.

Key Highlights:

- User-friendly and intuitive interface
- Efficient cloud-based storage.
- Organized and categorized image browsing
- Django's powerful backend ensures scalability
- Easy to extend with future enhancements

Future Enhancements:

- AI-based image tagging
- Face recognition albums
- Cloud database integration (AWS/GCP)
- Multi-user collaboration
- Photo editing tools

REFERENCES

1. Django Documentation – <https://docs.djangoproject.com>
2. Python Official Documentation – <https://www.python.org/doc/>
3. GitHub – PhotoStream Project Repository & System Flow Logic
https://github.com/karangupta12-stack/Django_Gallery_Application
4. StackOverflow – Community-driven Programming Questions & Solutions
<https://stackoverflow.com>
5. W3Schools – HTML, CSS & JavaScript Tutorials <https://www.w3schools.com>