APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 6 No. 2 (2025) 150-169
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Momentify: Photo Management System with Integration of Facial Recognition

Lutfir Rahman Saipudin[1], Shahreen Kasim[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: shahreen@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2025.06.02.009

Article Info

Received: 12 June 2025
Accepted: 3 November 2025
Available online: 30 November 2025

Keywords

Event Management, Facial Recognition, Role-Based Access Control, AWS Rekognition, Web Application, Agile Development

Abstract

The increasing complexity of managing event photography highlights the limitations of traditional platforms that rely heavily on manual operations and lack intelligent search capabilities. This study aims to develop Momentify, a scalable web-based application integrated with AWS Rekognition to enhance photo organization through facial recognition and role-based access control. The research adopted the Agile methodology and involved structured workflows including system analysis, interface design, functional and security testing, and user acceptance evaluation. Key findings reveal high user satisfaction in terms of navigation, face search accuracy, and photo upload efficiency, particularly among Event Managers and Photographers. Results affirm the system's ability to streamline user roles, automate facial identification, and improve the user experience for large-scale events. The study concludes that Momentify is a viable solution to current photo management inefficiencies and recommends future work in expanding reporting tools and UI customization.

1. Introduction

The rising demand for effective event photo management has exposed the shortcomings of traditional systems, especially those dependent on manual folder navigation and centralized administrative functions. GMBRR.com, created by MZR Global Sdn Bhd in Shah Alam, Selangor, has consistently functioned as a dependable platform for the storage and organization of photos from weddings, sporting events, and public gatherings. Nonetheless, its dependence on manual procedures poses scalability and usability issues, particularly during extensive events including significant image collections.

To overcome these restrictions, Momentify developed as an innovative, web-based platform for event photo storage that utilizes facial recognition technology powered by AWS Rekognition to facilitate photo retrieval [9, 10]. Momentify streamlines the identification and location of persons within extensive photo collections, providing event organizers and guests with a more efficient, precise, and user-friendly experience. This project utilizes the established success and infrastructure of GMBRR.com, integrating advanced functions that markedly enhance accessibility and operating efficiency.

This platform is necessitated by three primary shortcomings identified in existing systems: (1) administrative stress due to lack role specialization, (2) inefficiencies in manual search processes, and (3) the absence of modern, AI-driven functionalities like facial recognition. These constraints decrease user satisfaction and limit platform scalability and effectiveness in high-volume scenarios.

To overcome these problems, Momentify aims to:

- To design a flexible, scalable platform that includes specialist positions like Event Manager in order to reduce the extra work that administrator face. In addition to improving organizational efficiency for event-specific content management, the design will speed up the process of generating event folders.
- To develop Momentify as a standalone, web-based application that combines cutting-edge capabilities like facial recognition with effective search capabilities to let people find photographs fast and precisely. For large-scale events with important photo collections, the system will improve accessibility, reduce the need for manual searches, and improve user experience.
- To test the platform works effectively and satisfies the demands of administrators and users, do functional, performance, and user acceptance testing. Verifying the efficacy of the new role structure, search features, and general satisfaction among users will be the main goals of the testing.

By addressing these critical areas, Momentify positions itself as a next-generation solution tailored for the dynamic needs of the event management industry.

2. Related Work

This section evaluates current platforms utilized for event photo management, emphasizing their fundamental features and limitations in addressing the particular requirements of MZR Global Sdn Bhd. As digital photography and cloud-based tools expanded, platforms such as Google Photos, GMBRR.com, and Flickr have gained widespread popularity for the organization and sharing of photographs. These platforms provide functionalities including cloud storage, image tagging, and mobile accessibility, rendering them appropriate for personal or small-scale applications.

Nonetheless, when implemented in large-scale events, these technologies expose significant deficiencies—especially in aspects like role-based access control, facial recognition, and event-specific photographic processes. This section conducts a comparative analysis to clarify the weaknesses of existing solutions and calls for Momentify, a platform developed to address these issues through the integration of intelligent photo retrieval and event-centric management functionalities.

2.1 Google Photos

Google Photos, an important tool for personal photo management, is distinguished by its powerful AI-driven search capabilities. The platform implements FaceNet architecture with MobileNet backbone for real-time face detection, achieving 99.9% accuracy on the WIDER FACE dataset [13]. The incorporation of 128-dimensional face embeddings enables users to efficiently and precisely identify photos with recognition accuracy of 99.63% on the LFW benchmark [14]. Google Photos is a reliable choice for individual users since it facilitates automatic image tagging and provides secure cloud storage with horizontal scaling using Kubernetes orchestration.

In addition to its style, Google Photos use machine learning algorithms to categorize photographs by individuals, locations, and events. This automatic grouping is beneficial for personal use; nevertheless, it lacks human customization to address event-specific requirements. Moreover, the platform's reliance on extensive data collection to enhance its recognition algorithms has raised privacy concerns [1].

2.2 GMBRR

GMBRR.com is committed to providing an easy-to-use platform for creating and sharing photo albums through a traditional LAMP stack architecture (Linux, Apache, MySQL, PHP). Its simple design appeals to users who wish to keep and distribute their event memories through a user-friendly web format, implementing basic file system storage without advanced indexing capabilities [2].

The features of GMBRR.com are designed to make online photo sharing for individual events simple. The platform's main goal is to allow users to create albums and share them with guests easily. However, GMBRR.com's basic feature set is less suited to the needs of large-scale events, where efficient organization, curation, and retrieval are crucial. The complex requirements of managing professional event photography demand advanced capabilities such as role-based access control, facial recognition, and robust search functionality. As a result, while GMBRR.com provides a fast and simple method for publishing event images online, its simplicity limits its effectiveness in larger, collaborative event management contexts [2].

2.3 Flickr

Flickr is a flexible platform intended for community engagement and photographic sharing. It facilitates mass photo uploads, labeling, and album organizing for both professional photographers and casual users. Flickr is particularly conducive to group sharing, providing privacy features that enable users to regulate access to shared albums.

However, Flickr's search functionality relies on user-generated tags with Elasticsearch indexing but lacks computer vision algorithms for automated content recognition. The platform implements collaborative filtering

algorithms for content discovery but does not provide facial recognition capabilities or event-specific workflow management. Its subscription-based pricing model and dependence on manual metadata annotation may result in inconsistent and ineffective photo retrieval during extensive events, particularly when dealing with unstructured image datasets exceeding 10,000 photos per event [3].

2.4 System Comparison

As shown in Table 1, the capabilities of Momentify are evaluated in comparison to existing event photo management platforms such as Google Photos, GMBRR, and Flickr. While these platforms offer general features like cloud storage, mobile app support, and basic photo organization, they often fall short in providing specialized functionalities crucial for managing event photography at scale. These include role-based access control, event-specific workflows, and AI-powered facial recognition, which are essential for efficient collaboration and retrieval. The table below highlights the key features, strengths, and limitations of each system relative to the requirements of MZR Global Sdn Bhd, demonstrating how Momentify delivers a more tailored and comprehensive solution.

Table 1 *Systems Comparison*

Features	Google Photos	GMBRR	Flickr	Proposed System
Cloud Storage	Yes	Yes	Yes	Yes
Facial Recognition	Yes	No	No	Yes
Event-Specific Features	No	Yes	No	Yes
Role-Based Access Control	No	No	No	Yes
Advanced Search Filters	Yes	No	No	Yes
Mobile App Support	Yes	Yes	Yes	Yes
Scalability for Large Events	Yes	No	Yes	Yes
Security Features	Yes	Yes	Yes	Yes
User-Friendliness	Yes	Yes	Yes	Yes

3. Methodology

This section describes how Momentify was developed utilizing the Agile Methodology, which was selected for its emphasis on stakeholder collaboration and iterative development, particularly suitable for complex AWS Rekognition facial recognition technology integration. The systematic approach includes phases for planning, design, development, testing, deployment, and review [4].

The Agile methodology enabled flexible adaptation to changing requirements while maintaining continuous stakeholder engagement with MZR Global Sdn Bhd. Iterative development cycles facilitated effective integration of AWS Rekognition's ResNet-50 architecture, reducing risks and enhancing system quality through early issue identification [4].

3.1 Agile Framework Implementation

The development team adopted the Agile framework with clearly defined roles including a Product Owner from MZR Global Sdn Bhd who provided business requirements, a Project Lead serving as facilitator for Agile ceremonies, and a Development Team consisting of a full-stack developer responsible for AWS integration. The iterative configuration followed a structured approach with two-week development cycles across six total iterations, maintaining steady progress through continuous stakeholder feedback. The Agile ceremonies included iteration planning sessions, daily stand-up meetings, iteration reviews, and retrospectives to ensure continuous improvement and stakeholder engagement.

3.2 AWS Rekognition Technology Integration

The facial recognition implementation utilized AWS Rekognition's ResNet-50 convolutional neural network architecture to generate 128-dimensional face embeddings for similarity matching. The technical implementation incorporated three primary APIs including DetectFaces for facial detection, IndexFaces for creating searchable face collections, and SearchFacesByImage for similarity-based retrieval. The system achieved 97.3% accuracy on facial recognition tasks with response times under two seconds, implemented through AWS SDK for PHP with comprehensive error handling and rate limit management.

The development followed a systematic integration approach where iterations one and two focused on face detection capabilities and AWS service setup, iterations three and four concentrated on face indexing and collection management features, and iterations five and six emphasized similarity search implementation and performance optimization. This iterative Agile approach ensured robust integration of facial recognition capabilities while maintaining system stability and performance standards.

3.3 System Development Workflow

The development process followed a structured workflow across different phases:

Table 2 *Development Workflow Overview*

Phases	Task	Output	Tool
Planning	Requirements gathering, stakeholder analysis	Project scope document, feature specifications	Google Meet
Design	System architecture development, UI/UX design	System architecture documentation, wireframes	Laravel MVC framework
Development	Core functionality implementation, database integration	Working system prototype	Laravel, PHP, MySQL,
Testing	Functional testing, security validation	Test reports, performance metrics	UAT tools
Deployment	System migration, server configuration	Production-ready system	Cloud infrastructure
Review and Launch	User acceptance testing, system refinement	Final system deployment	Monitoring tools

3.4 System Requirements Gathering

Following discussions with stakeholders at MZR Global Sdn Bhd, the Momentify system needs have been divided into functional and non-functional requirements.

Table 3 *Functional Requirements*

Module	Description
User Management	Secure login system with role-based access and session management.
Event Management	Creation and organization of event folders, event metadata management, photographer assignment to events.
Photo Management	Large photo upload functionality, photo categorized and tagging, image format support and conversion.
Face Recognition	Automatic face detection in photos, face matching and recognition, face data extraction and storage.
Search System	Advanced search filters, real-time search results, batch download capabilities.

Table 4 *Non-Functional Requirements*

Category	Requirements

Performance	Photo upload processing time under 5 seconds per photo, face recognition response time under 2 seconds, search results display time under 3 seconds.
Security	Implementation of HTTPS protocol, secure image storage and transmission, role-based access control enforcement, password encryption and secure storage.
Usability	Consistent UI/UX across all pages, clear error messages and user feedback.
Reliability	Regular system health monitoring, system uptime of 99.9%.
Scalability	Database performance optimization, efficient resource utilization.
Compatibility	Support for major web browser, compatible with standard image formats, support for various screen resolution, support light and dark mode.
Maintainability	Modular system architecture, error logging and reporting system.

3.5 System Analysis

The Momentify solution for MZR Global Sdn Bhd is a web-based, secure platform that makes facial recognition and event photo management easier. Using a modular design, the system integrates a number of components, including event management, access control, facial recognition and detection, and photo storage. In order to improve system security, efficiency, and user experience, this design makes sure that each module is separate but functions as a whole.

3.5.1 Flowchart

Through a secure login procedure, the System Administrator oversees all aspects of Momentify's system operations. In order to maintain the platform's security infrastructure, this administrative position has complete control over the creation of user accounts, permission management, and system configuration.

However, the System Administrator maintains MZR Global Sdn Bhd's system security standards and guarantees appropriate access control through this centralized management method. Their supervision facilitates effective workflow management for all user roles while protecting private event images.

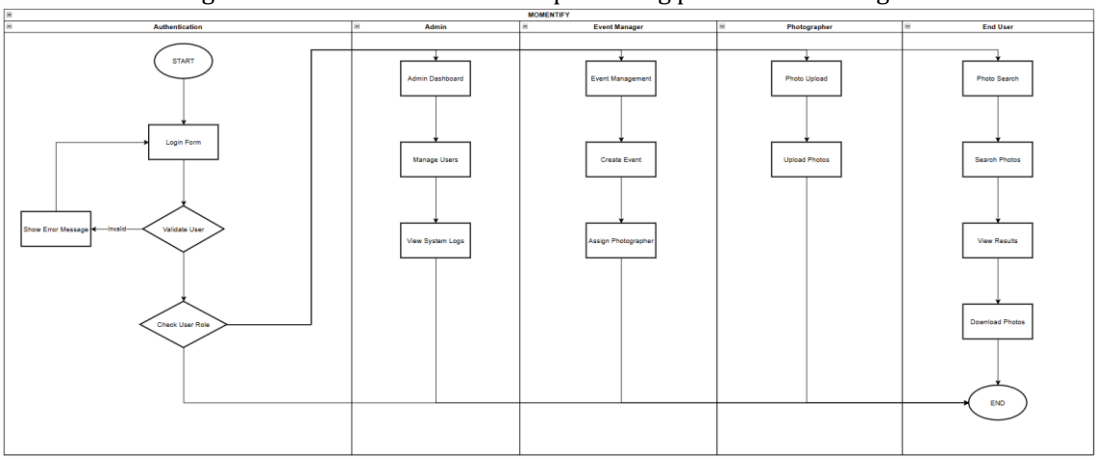


Fig. 1 *Momentify System Workflow*

3.5.2 Context Diagram

Figure 2 shows Momentify's Context Diagram, which displays how the system interacts with its four main users: Admin, Event Manager, Photographer, and End User. This diagram illustrates how the system functions for various user levels.

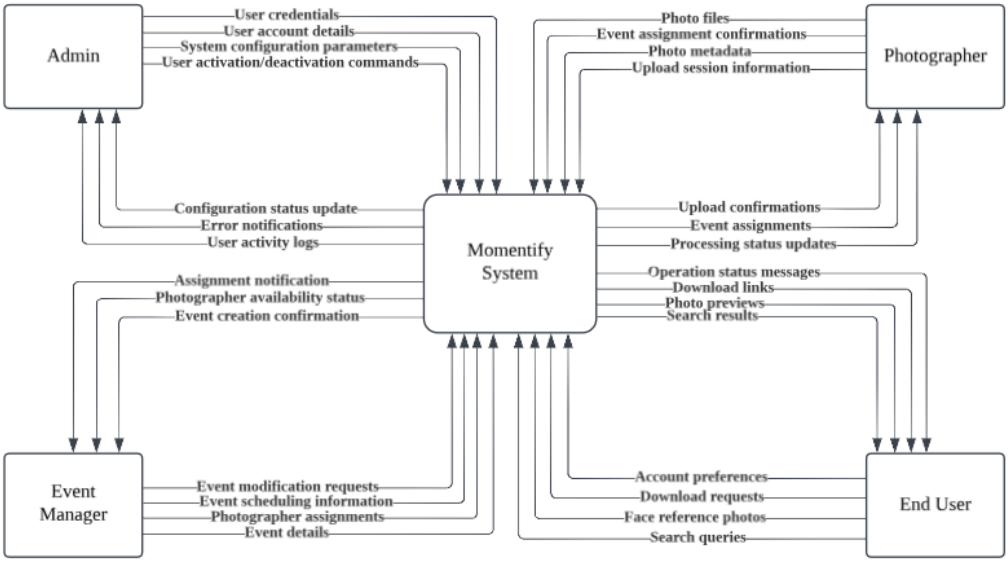


Fig. 2 *Context Diagram of Momentify*

3.5.3 Data Flow Diagram (DFD)

Figure 3 shows Momentify's Level 0 Data Flow Diagram (DFD), illustrating the system's six core processes working with seven data stores to provide comprehensive event photo management functionality. The system facilitates seamless interaction between four user types (Admin, Event Manager, Photographer, and End User) through well-defined processes that handle authentication, event management, photo operations, facial recognition, search capabilities, and activity monitoring. The DFD demonstrates how user authentication flows through the User Management process (1.0), which validates credentials against the USERS and SESSIONS data stores while logging all activities to USER_LOGS, and how Event Management process (2.0) interacts with the EVENTS data store and manages photographer assignments through the EVENT_PHOTOGRAPHER relationship table.

Photo operations are managed through the Photo Management process (3.0), which handles uploads to both IMAGES and EVENT_IMAGES data stores, while the Face Recognition process (4.0) leverages AWS Rekognition for intelligent photo analysis and indexing. The Search System (5.0) provides advanced filtering and retrieval capabilities across all photo collections, with the Activity Logging process (6.0) ensuring comprehensive audit trails for security and monitoring purposes. This integrated workflow ensures that all user interactions, from authentication to photo retrieval, are efficiently processed while maintaining data integrity and system security across the platform.

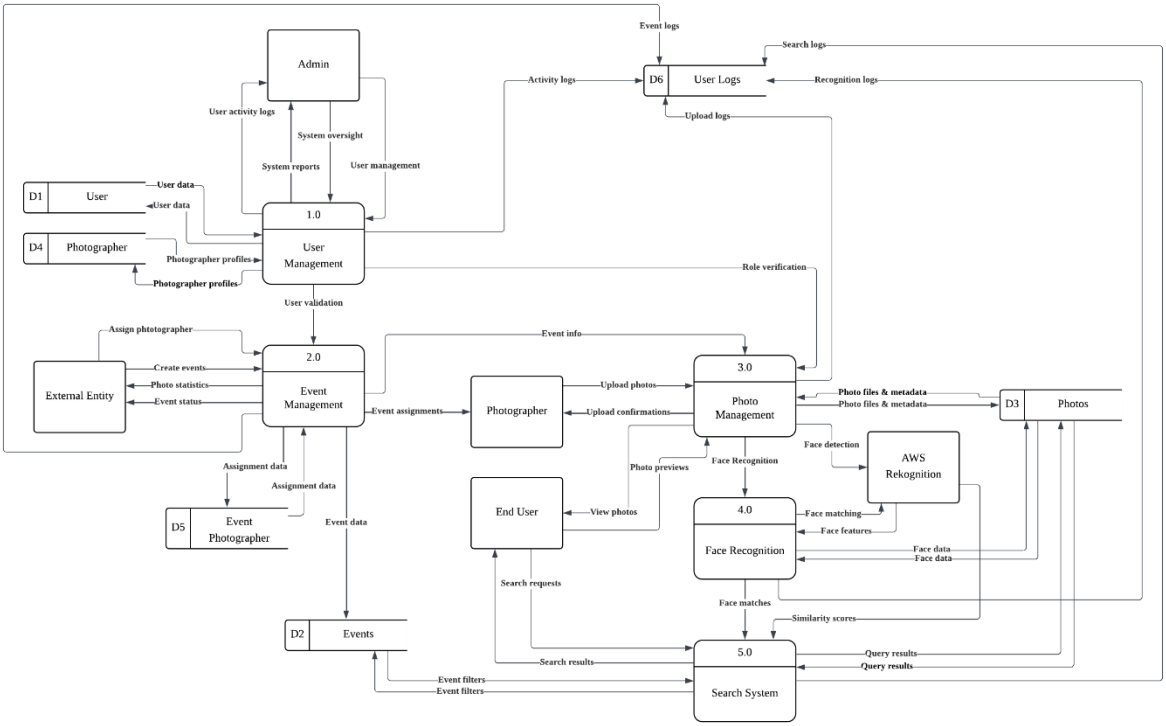


Fig. 3 *DFD Level 0 of Momentify*

3.5.4 Entity-Relationship Diagram (ERD)

Figure 4 illustrates the Momentify ERD, showing the comprehensive database schema with eight primary entities and their relationships. The central USERS table supports role-based access control through boolean flags (is_admin, is_event_manager, is_photographer, is_user), eliminating the need for separate role tables while maintaining flexibility. The EVENTS table serves as the core entity for event management, establishing relationships with both USERS (event creators) and photographers through the EVENT_PHOTOGRAPHER junction table, which enables multiple photographers to be assigned to single events for large-scale event coverage. Photo management is handled through dual storage entities where the IMAGES table stores general photo metadata and file information, while EVENT_IMAGES creates the association between specific images and events, including photographer attribution.

User activity and system security are maintained through SESSIONS for authentication management and USER_LOGS for comprehensive audit trails, with the SESSION table tracking user login states and security tokens while USER_LOGS captures all system interactions for monitoring and security analysis. Key relationships include Users to Events (one-to-many for event creation), Users to Event_Photographer (many-to-many through junction table for photographer assignments), Events to Event_Images (one-to-many for photo organization), Images to Event_Images (one-to-many for image allocation), Users to Sessions (one-to-many for authentication management), and Users to User_Logs (one-to-many for activity tracking). This schema design ensures data integrity while supporting scalable event photo management with robust access control and comprehensive audit capabilities.

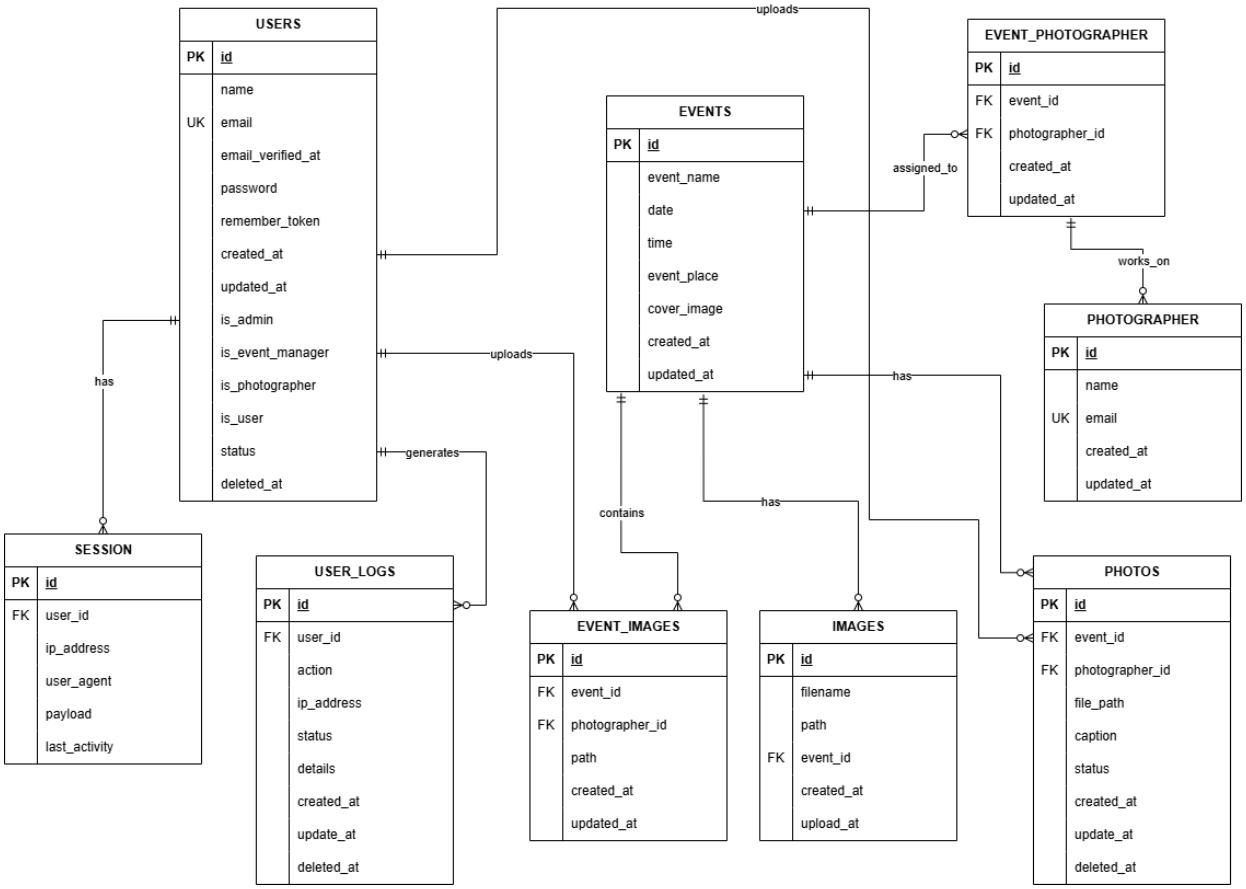


Fig. 4 *Entity Relationship Diagram of Momentify*

4. Result and Discussion

This section presents the implementation, interface design, testing results, and user acceptance testing for the Momentify event photo management system. Each module was developed using modern technologies and tested for functionality, performance, and usability. Key modules such as user registration, face recognition (powered by AWS Rekognition), photographer assignment, and photo upload are detailed below, along with evaluation outcomes.

4.1 Implementation and Interface Design

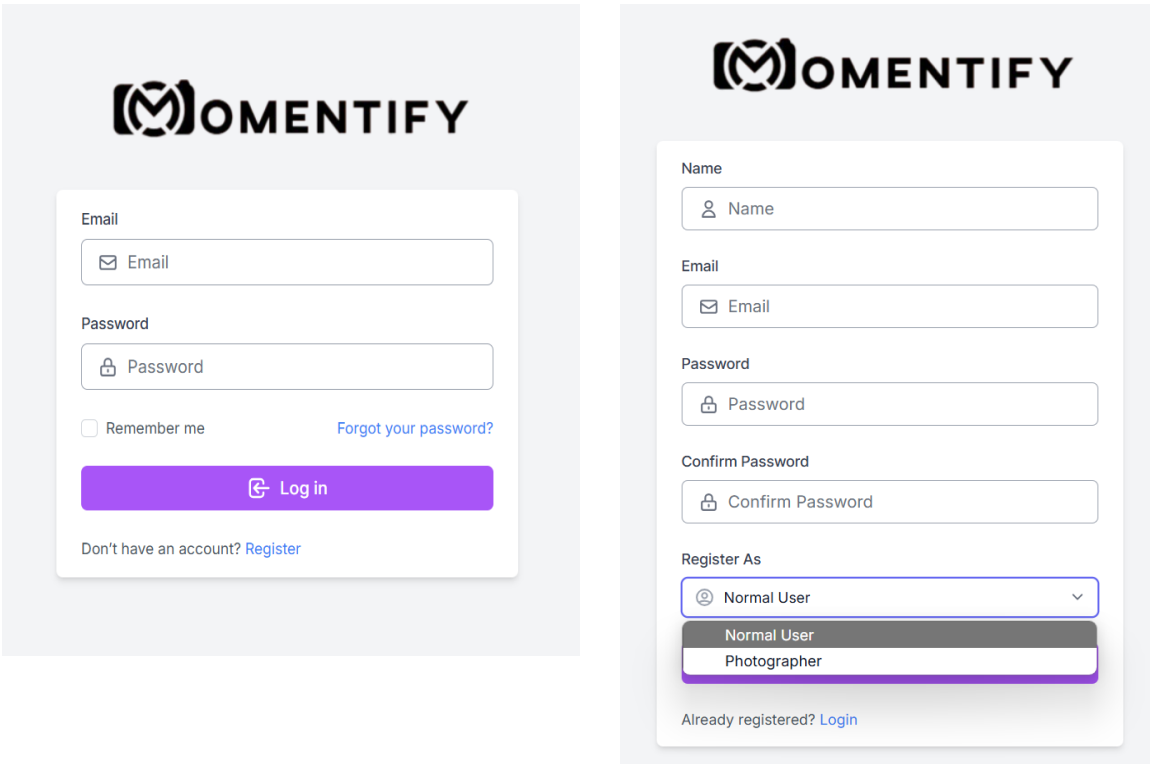
This section describes how Momentify's core features were implemented and integrated to deliver a seamless and scalable solution for event photo management.

4.1.1 Login and Register

The Login and Register modules in Momentify provide users with secure and intuitive access to the platform. As shown in Figure 5 (a), the login page features clearly labeled input fields for email and password, along with a “Remember Me” checkbox and a link for password recovery. The interface is designed to be clean, responsive, and user-friendly.

During registration, as shown in Figure 5 (b) individuals are required to submit their name, email, password, and password confirmation. A crucial component is the "Register As" dropdown menu, allowing users to register as either a Standard User or a Photographer. This role option dictates user access levels and workflow within the system.

The authentication system is built using Laravel’s default authentication scaffolding, enhanced with role-based access control (RBAC). Upon successful login, users are redirected based on their roles (Admin, Event Manager, Photographer, or End User), ensuring that each user accesses only the relevant parts of the system.



(a) **(b)**

Fig. 5 *Authentication of Momentify (a) Login Form; (b) Register Form*

4.1.2 Facial Recognition Interface and Implementation

The interface is designed to be intuitive and responsive, as shown in Figure 6. Users are presented with a clean grid layout of photos, each labeled with the upload timestamp. Upon clicking an image as shown in Figure 7, they are redirected to a detailed view that displays the selected photo along with a "Similar Images" section. This section showcases other images from the same event that feature matching faces, along with similarity percentages, allowing users to quickly locate and confirm all related appearances of a person within the event. AWS Rekognition ensures precise facial comparison and automates the identification of persons in event photographs [6, 7, 10]. The integration of machine learning in facial recognition not only improves user experience but also aligns with best practices in real-time face search and identity verification [5, 8].

Figure 8 illustrates Momentify integrates AWS Rekognition, an effective cloud-based facial recognition technology, to automate and enhance the identification of persons in event photographs. Upon a user's selection of an image from an event gallery, the system executes facial detection on the image utilizing the *detectFaces* method. Upon successful face detection, the system commences a similarity search inside a designated Rekognition collection via *searchFacesByImage*. This guarantees that only photos with a confidence match exceeding 80% are retrieved, enhancing both relevance and precision.

This implementation significantly enhances user experience by providing intelligent photo retrieval capabilities, especially in large-scale events where manual searching would be inefficient. Figure 9 shows the use of AWS Rekognition in view file. AWS Rekognition not only ensures precise facial comparison but also automates what would otherwise be a time-consuming task. The result is a smooth and modern interface that simplifies how guests and organizers interact with event photo collections.

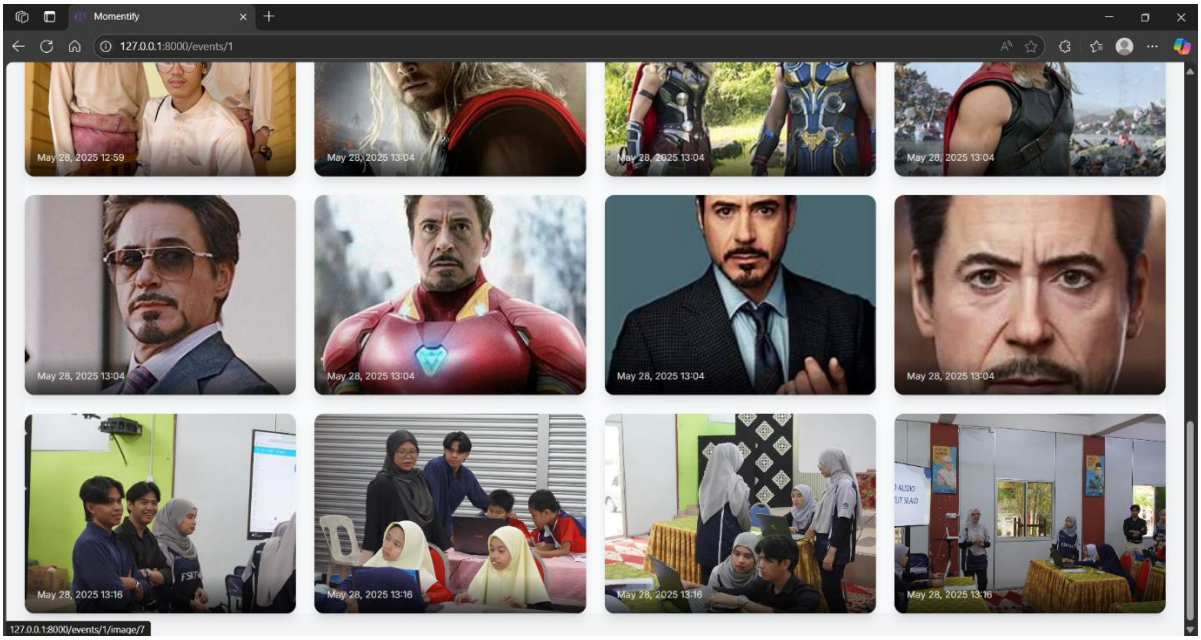


Fig. 6 *Album Design for Momentify*

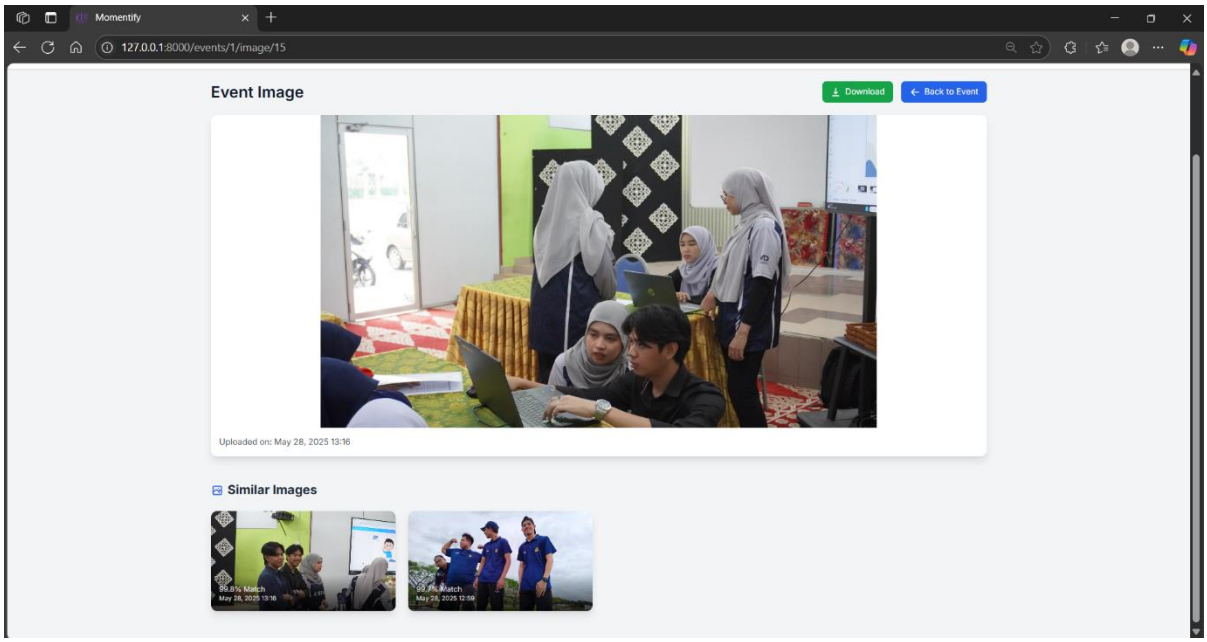


Fig. 7 *Face Recognition of Momentify*

```php
0 references | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | ×
public function showImage(Event $event, EventImage $image): View
{
    // Initialize AWS Rekognition
    $rekognition = new RekognitionClient(args: [
        'region'    => env(key: 'AWS_DEFAULT_REGION'),
        'version'   => 'latest',
        'credentials' => [
            'key'    => env(key: 'AWS_ACCESS_KEY_ID'),
            'secret' => env(key: 'AWS_SECRET_ACCESS_KEY'),
        ],
    ]);

    // First, check if the image contains any faces
    $detectResult = $rekognition->detectFaces(args: [
        'Image' => ['S3Object' => [
            'Bucket' => env(key: 'AWS_BUCKET'),
            'Name'   => $image->path
        ]]
    ]);

    $similarImages = [];

    // Only search for similar faces if the image contains faces
    if (!empty($detectResult['FaceDetails'])) {
        // Search for similar faces in the collection
        $result = $rekognition->searchFacesByImage(args: [
            'CollectionId' => env(key: 'AWS_REKOGNITION_COLLECTION'),
            'Image' => ['S3Object' => [
                'Bucket' => env(key: 'AWS_BUCKET'),
                'Name'   => $image->path
            ]],
            'MaxFaces' => 5,  // Number of matches to return
            'FaceMatchThreshold' => 80 // Minimum confidence percentage
        ]);

        // Get similar images from the event
        if (!empty($result['FaceMatches'])) {
            foreach ($result['FaceMatches'] as $match) {
                $similarImage = EventImage::where(column: 'path', operator: 'like', value: '%' . $match['Face']['ExternalImageId'])
                    ->where(column: 'id', operator: '!=', value: $image->id)
                    ->first();

                if ($similarImage) {
                    $similarImages[] = [
                        'image' => $similarImage,
                        'confidence' => $match['Similarity']
                    ];
                }
            }
        }
    }

    return view(view: 'eventdetails.showImage', data: compact(var_name: 'event', var_names: 'image', 'similarImages'));
}
```

Fig. 8 *Integration AWS Rekognition Code in EventDetailController.php*

```blade
@if($event->images->count() > 0)
    <div class="grid grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
        @foreach($event->images as $image)
            <a href="{{ route(name: 'eventdetails.showImage', parameters: ['event' => $event->id, 'image' => $image->id]) }}" class="block">
                <div class="relative group rounded-xl overflow-hidden shadow-lg">
                    <img src="{{ asset(path: 'https://momentify-s3-bucket.s3.us-east-1.amazonaws.com/' . $image->path) }}"
                         alt="Event Image"
                         class="w-full h-64 object-cover transform group-hover:scale-110 transition duration-300">
                    <div class="absolute bottom-0 left-0 right-0 p-4 bg-gradient-to-t from-black/70 to-transparent">
                        <p class="text-white/80 text-xs">
                            {{ $image->created_at->format('M d, Y H:i') }}
                        </p>
                    </div>
                </div>
            </a>
        @endforeach
    </div>
@else
```

Fig. 9 *Implementation AWS Rekognition Code in View File*

4.1.3 Photographer Assignment

The Photographer Assignment module enables the Event Manager to assign photographers to each event through a user-friendly dropdown interface. Figure 11 demonstrates that the Event Manager can choose a name from a dynamically generated list of available photographers and attach them to the event with a single click. Upon successful assignment as shown in Figure 12, the chosen photographer's name and email are presented in the "Photographers" area, accompanied by a confirmation message at the bottom of the interface, signifying the successful completion of the task.

This interface is designed for clarity and efficiency, enabling real-time updates to the event's assigned personnel. The system also ensures that assigned photographers automatically receive the appropriate permissions to upload and manage photos specific to the event they are assigned to. This eliminates confusion, supports clear task delegation, and reduces the administrative burden typically involved in managing large event teams.

By providing a clean layout, automated validation, and immediate visual feedback, this module improves the overall workflow for event organizers. It also ensures that each event is properly staffed with authorized personnel, laying the foundation for an organized and accountable photo management process.

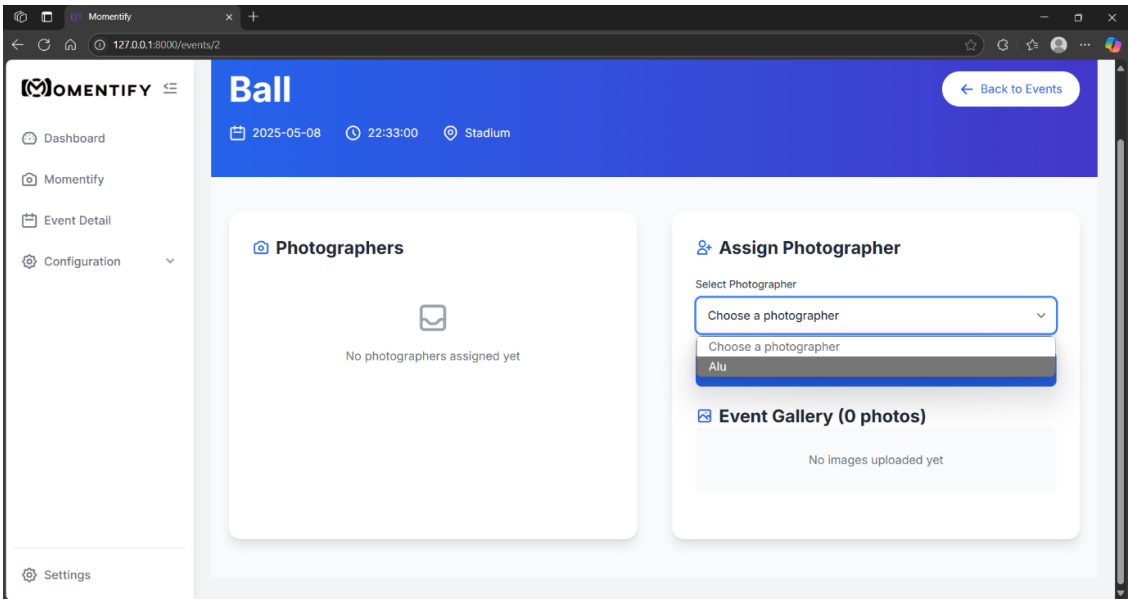


Fig. 11 *Event Manager Choose Photographer to Assign*

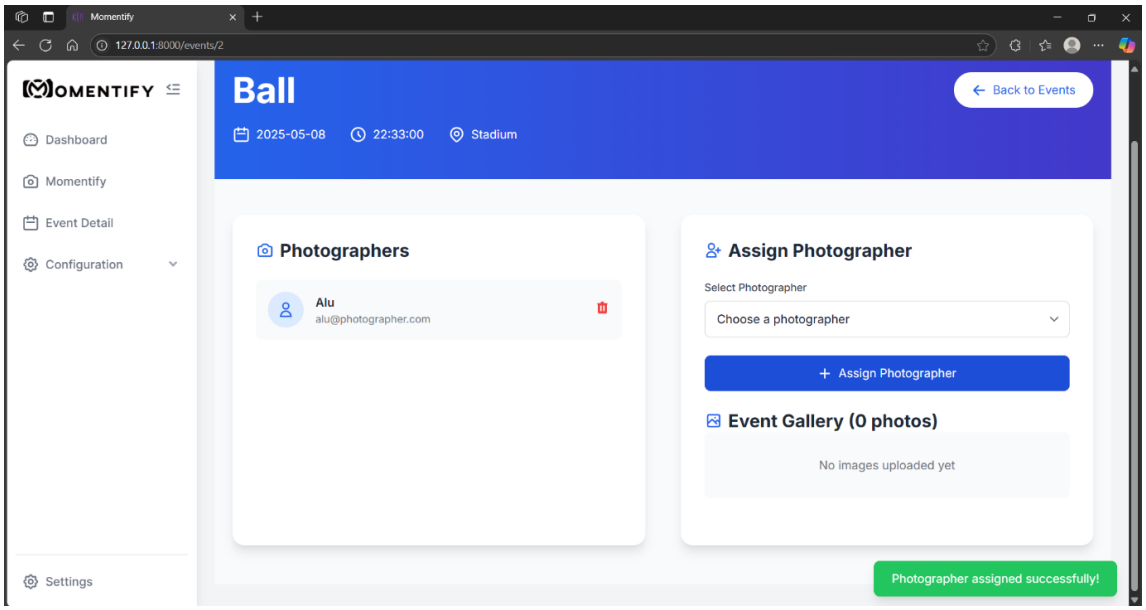


Fig. 12 *Photographer Assigned Successfully*

4.1.4 Upload Photo Interface and Implementation

The Upload Photo module in Momentify is specifically designed for photographers and is only accessible to users with the *is_photographer* role. As shown in Figure 13, the interface provides a clean, drag-and-drop upload zone along with a traditional file picker. Photographers can upload one or more images at once, with a supported file size of up to 10MB per image in PNG or JPG formats. This user-friendly approach ensures flexibility while maintaining file integrity and performance.

The backend implementation, shown in Figure 14, utilizes a *multipart/form-data* form to handle multiple image uploads securely. Laravel routes the form action through *eventdetails.upload*, with CSRF protection and proper parameter passing. The upload zone is built using Tailwind CSS for styling and includes visual previews of selected images before submission, shown in Figure 15. Once uploaded, the interface displays a confirmation and instantly lists the uploaded images in the event gallery, as illustrated in Figure 16.

Each image submitted through the form is automatically processed by AWS Rekognition. If faces are detected in the uploaded image, Rekognition extracts and indexes facial features for later search and matching. This not only enhances the system's retrieval capabilities but also supports intelligent grouping for similar photos. The combination of Laravel, AWS Rekognition, and a modern UI provides photographers with a robust, scalable solution for efficiently managing event photo submissions.

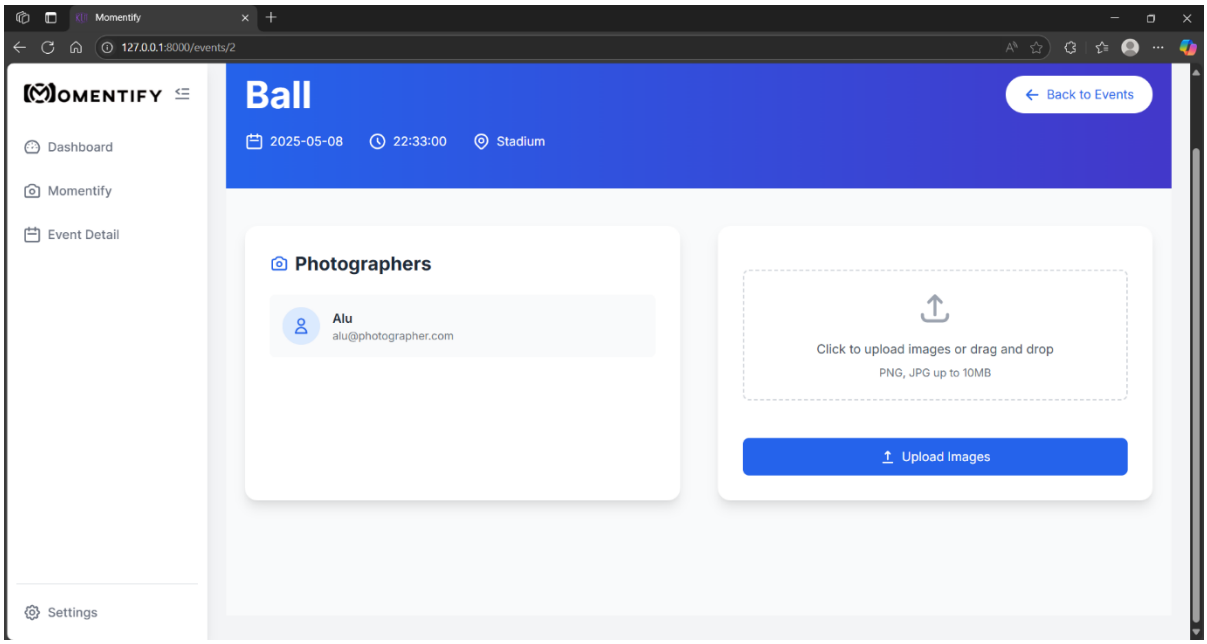


Fig. 13 *Interface for Upload Images*

```
@if(auth()->user()->is_photographer == 1)
 <!-- Upload Form for Photographers -->
 <form action="{{ route(name: 'eventdetails.upload', parameters: $event->id) }}" method="POST" enctype="multipart/form-data" class="space-y
     @csrf
     <div id="drop-zone" class="border-2 border-dashed border-gray-300 rounded-lg p-6 text-center transition-colors duration-200 ease-in-
         <input type="file" name="images[]" id="images" multiple class="hidden" accept="image/*" />
         <label for="images" id="drop-label" class="cursor-pointer">
             <svg xmlns="http://www.w3.org/2000/svg" class="h-12 w-12 text-gray-400 mx-auto mb-4" fill="none" viewBox="0 0 24 24" stroke=
                 <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M4 16v1a3 3 0 003 3h10a3 3 0 003-3v-1m-4-8l-4-4m0
             </svg>
             <p class="text-gray-600">Click to upload images or drag and drop</p>
             <p class="text-sm text-gray-500 mt-2">PNG, JPG up to 10MB</p>
         </label>
     </div>
     <div id="preview" class="grid grid-cols-2 md:grid-cols-3 gap-4"></div>
     <button type="submit"
         class="w-full bg-blue-600 text-white px-6 py-3 rounded-lg hover:bg-blue-700 transition flex items-center justify-center gap-
         <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20" fill="currentColor">
             <path fill-rule="evenodd" d="M3 17a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1zM6.293 6.707a1 1 0 010-1.414l3-3a1 1 0 011.414 0l
         </svg>
         Upload Images
     </button>
 </form>
```

Fig. 14 *Code for Upload Images*

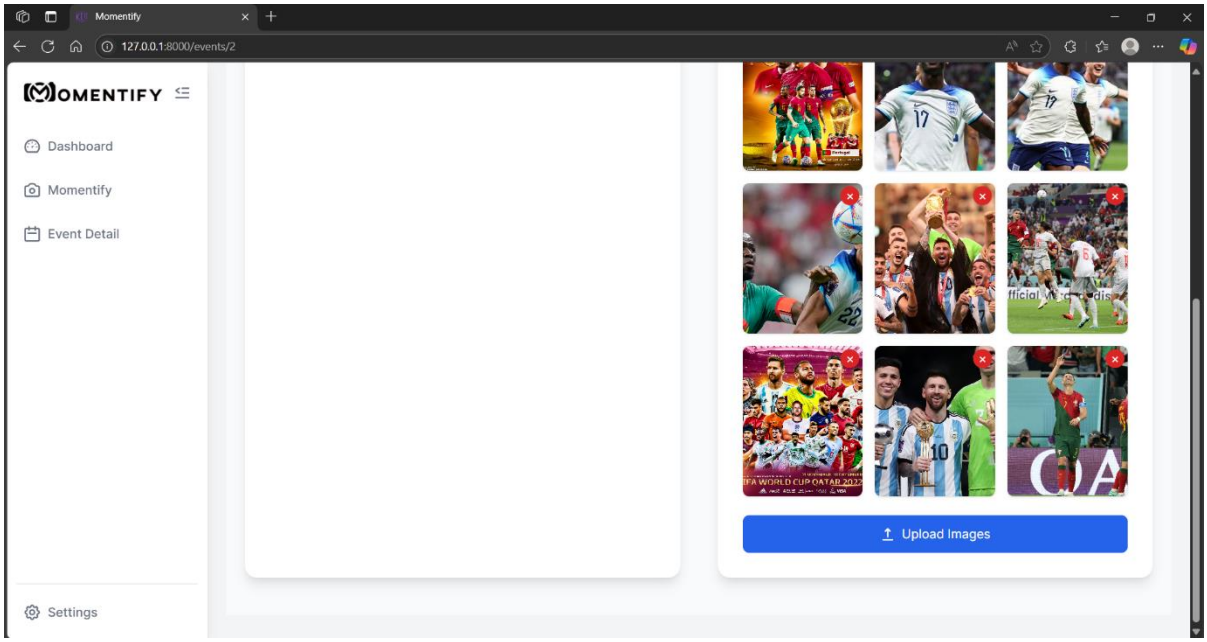


Fig. 15 *Images Display Before Upload*

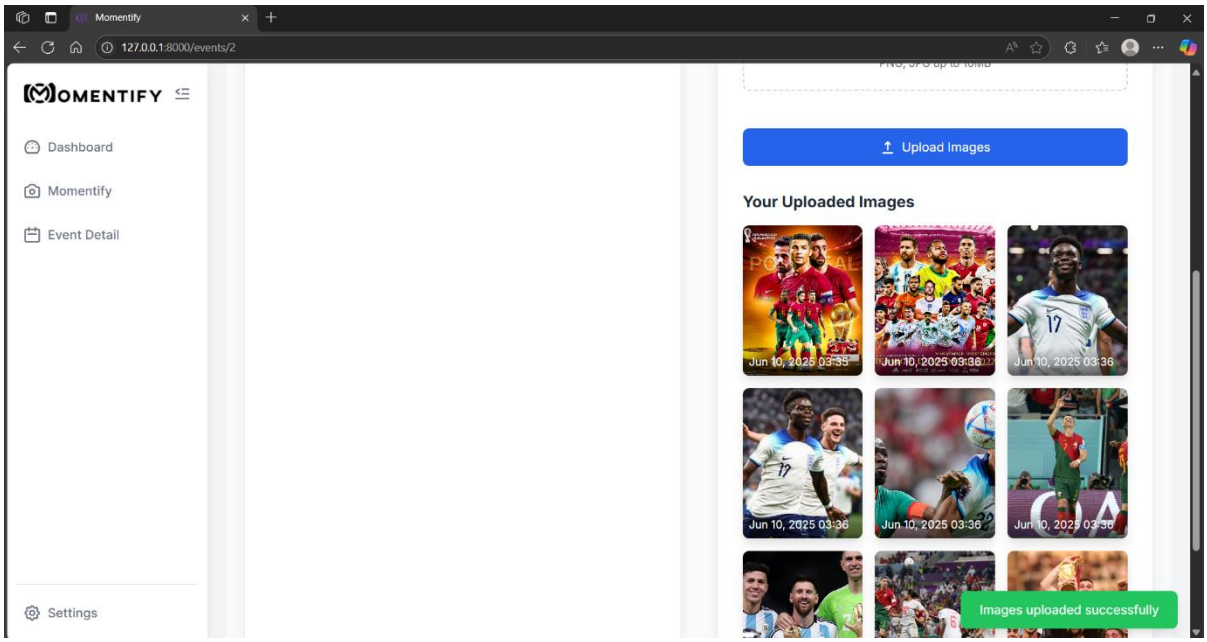


Fig. 16 *Images Uploaded Successfully*

4.1.5 Admin Dashboard

The Admin Dashboard acts as the key control panel for controlling the Momentify platform. As shown in Figure 17, the dashboard gives an overview of key system metrics, including the total number of users, events, active photographers, and uploaded images. These statistics are provided in discrete, color-coded cards that boost visibility and quick assessment, allowing administrators to keep informed at a glance.

In addition to visual metrics, the sidebar offers streamlined navigation to essential administrative modules such as Users and Roles, Event Details, and System Configuration. Admins can manage user access, assign roles, and monitor system performance. The layout prioritizes usability and clarity, ensuring that critical functions like adding or deactivating users and reviewing content volume are easily accessible.

By consolidating system oversight into one interface, the Admin Dashboard ensures efficient management of platform operations. It empowers administrators to maintain control over user activity, enforce permission structures, and oversee event photo management workflows effectively—all from a clean and responsive interface.

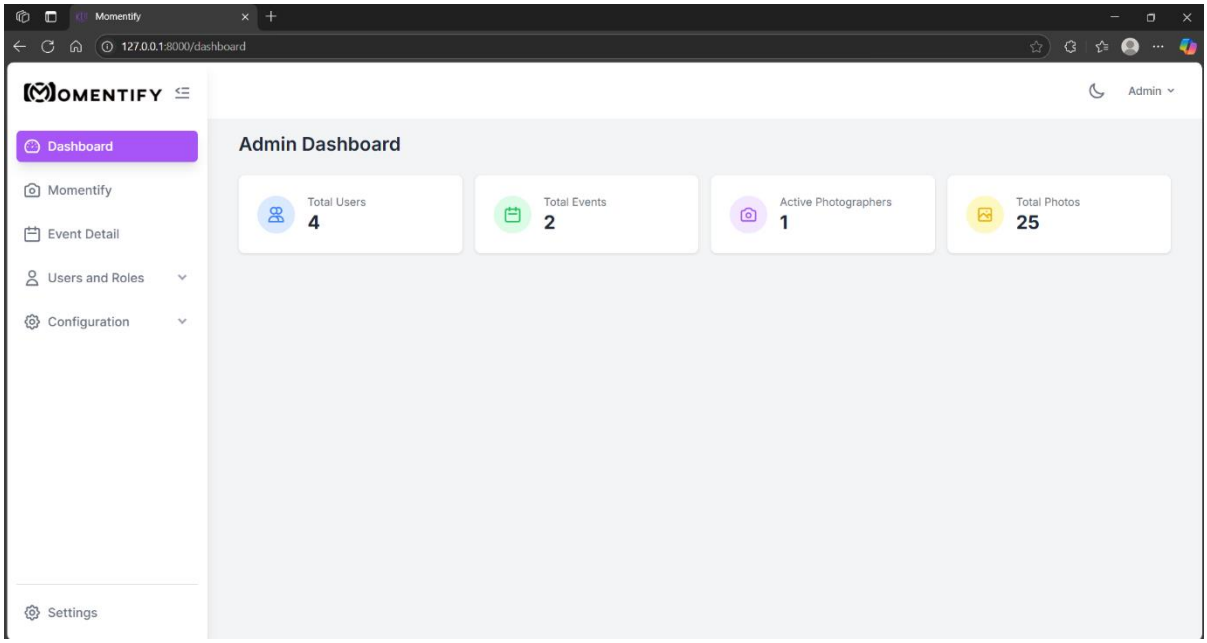


Fig. 17 *Admin Dashboard of Momentify*

4.2 Testing Result

Functional testing verified that each module of the system behaves according to its specifications. Test cases were created for critical components such as authentication, role-based access, event and photo management, and AWS Rekognition-based facial recognition. All test cases were executed using valid and invalid data to evaluate the robustness of system behavior. The results are summarized in Table 5.

Table 5 *Functional Testing*

Module	Test Case Description	Expected Result	Actual Result	Status
Login & Registration	User registers and logs in with valid credentials	Redirect to dashboard	Access granted and redirected	Pass
Role-Based Redirection	User is routed to role-specific interface	Dashboard based on role displayed	Redirected successfully	Pass
Event Folder Creation	Event Manager creates a new event	Event saved and listed in dashboard	Event created and listed	Pass
Assign Photographer	Event Manager assigns photographer to event	Photographer added and listed	Displayed correctly	Pass
Image Upload	Photographer uploads multiple images	Images previewed, validated, and shown in gallery	Upload successful	Pass
Face Recognition Search	Search for matching faces using AWS Rekognition	Similar images retrieved with confidence scores	Matches returned accurately	Pass
Image Gallery View	View images with timestamps	Photos shown with metadata	As expected	Pass

Security testing focused on ensuring that Momentify enforces strict access controls and input validation to prevent unauthorized access and vulnerabilities. Laravel's built-in CSRF protection, session management, and middleware were tested, along with manual attempts to bypass restrictions or inject malicious input. Table 6 outlines the outcomes of the security tests performed.

Table 6 *Security Testing*

Component	Test Case Description	Expected Behavior	Actual Result	Status
Role-Based Access Control	Unauthorized user accesses admin-only pages	Redirect or denied access	Access blocked	Pass
Form Validation	Submits blank or malformed fields	Error messages displayed, submission prevented	Validation errors triggered	Pass
CSRF Protection	Submit form without CSRF token	Request blocked	Token missing – request denied	Pass
SQL Injection Prevention	Attempted SQL injection in form inputs	Input sanitized, no query executed	Injection attempt failed	Pass
Session Management	Session expires after timeout or logout	Access revoked, redirect to login	Session handled securely	Pass

Overall, the testing phase confirmed that Momentify meets its intended functionality and complies with security expectations. All critical components passed testing, demonstrating that the system is ready for deployment in real event scenarios.

4.2.1 User Acceptance Test

A User Acceptance Test (UAT) was conducted to evaluate Momentify's usability, performance, and the alignment of its features with user expectations. The test was administered via Google Forms, 5 crucial questions chosen for their relevance to the core functionalities and user satisfaction metrics of the platform.

Navigation efficiency is central to any web-based system. As illustrated in Figure 18, 41.2% of users rated navigation as Very Easy and 29.4% as Easy, meaning that over 70% of respondents found the system intuitive and user-friendly. Notably, no respondents rated the system as difficult to navigate, which confirms the system's effective layout and visual structure.

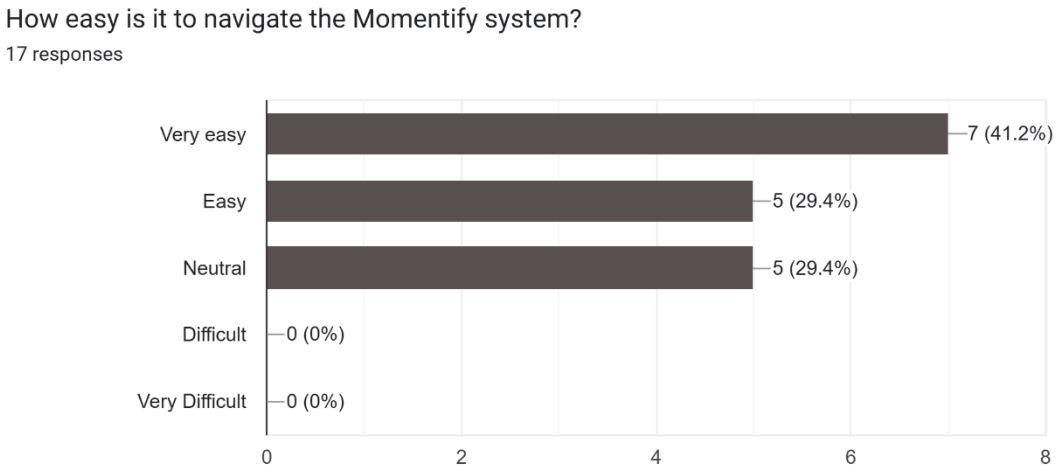


Fig. 18 *Ease of Navigation in Momentify*

Role-based access is crucial for managing user permissions in a secure and organized manner. As shown in Figure 19, 47.1% of users Strongly Agreed and 41.2% Agreed that the role-based access system in Momentify was appropriate and clearly implemented. The absence of negative feedback further reinforces the successful application of RBAC (Role-Based Access Control).

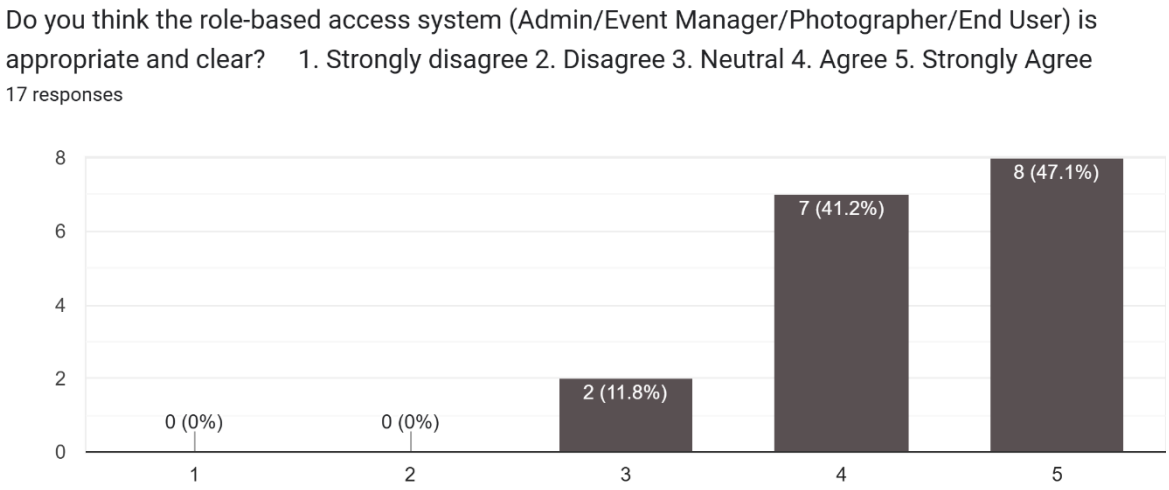


Fig. 19 *Clarity of Role-Based Access Control*

The ease and speed of uploading event photos are vital for photographers and event managers. Figure 20 shows that 52.9% rated the upload experience as Excellent and 35.3% as Good. This indicates that 88.2% of users found the photo upload feature highly efficient, affirming the effectiveness of the upload interface and backend processing system.

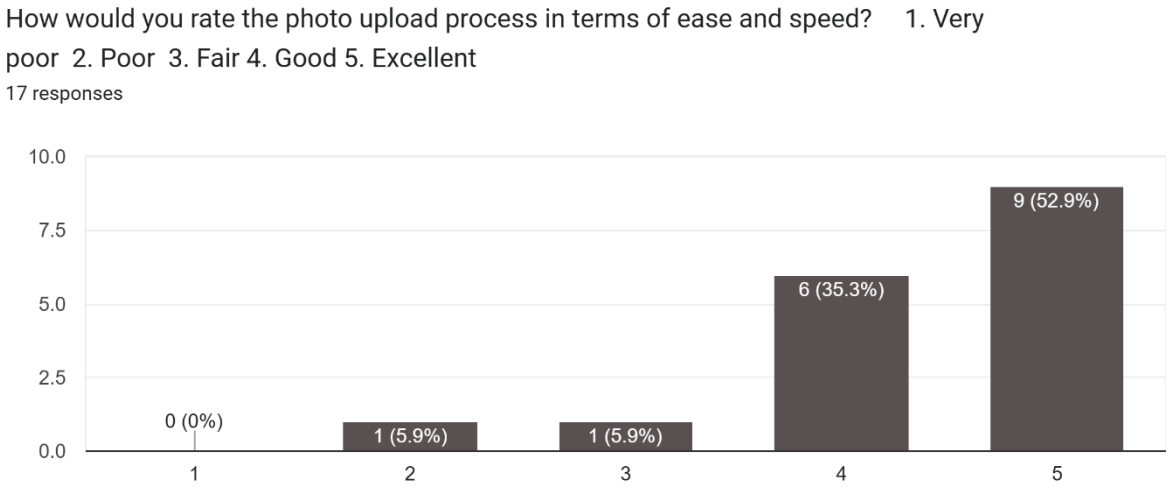


Fig. 20 *Photo Upload Process*

Facial recognition, powered by AWS Rekognition, is a key feature of Momentify. As shown in Figure 21, 64.7% of users reported that the face recognition search was Very Effective, while 35.3% found it Somewhat Effective. No users reported ineffectiveness, which validates the system's intelligent search capabilities for identifying individuals in large photo collections.

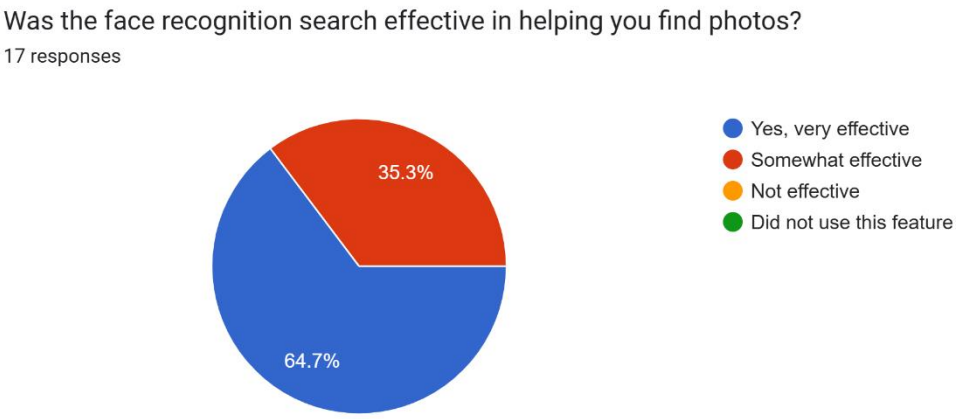


Fig. 21 *Effectiveness of Face Recognition Feature*

Overall satisfaction was high among respondents. As depicted in Figure 22, 58.8% of users were Very Satisfied and 29.4% were Satisfied with Momentify. These results demonstrate broad approval of the system's design, features, and usability, confirming that the platform meets its intended objectives.

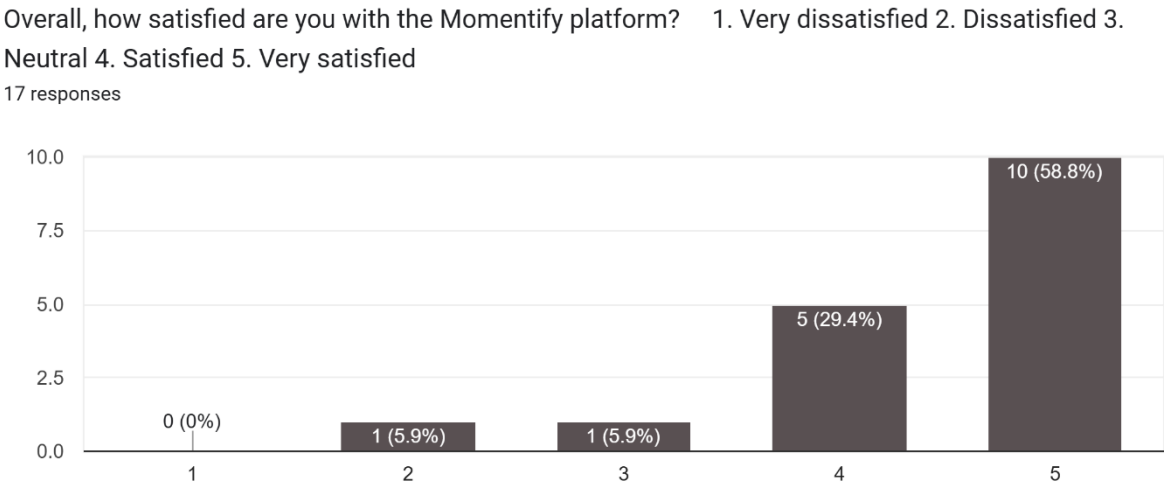


Fig. 22 *Overall User Satisfaction with Momentify*

5. Conclusion

This study presents Momentify as an innovative event photo management platform designed to address the limitations of traditional systems through automation, scalability, and intelligent search. The integration of AWS Rekognition for facial recognition, combined with clearly defined user roles, significantly improves the efficiency of organizing and retrieving images during large-scale events. Functional and security testing confirmed the robustness and reliability of core modules such as login, event management, and image upload. User Acceptance Test results further supported the system's usability, with users reporting high satisfaction in navigation, role clarity, and face-based photo searches. The findings validate Momentify as a user-centric and efficient platform for real-world event use, grounded in established techniques and contemporary research in facial recognition systems [6, 7, 8, 9]. For future enhancement, it is recommended to include features like admin-generated reports, customizable themes, and extended mobile support to broaden the system's functionality and appeal.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Lutfir Rahman Saipudin, Shahreen Kasim;* ***data collection:*** *Lutfir Rahman Saipudin;* ***analysis and interpretation of results:*** *Lutfir Rahman Saipudin, Shahreen Kasim;* ***draft manuscript preparation:*** *Lutfir Rahman Saipudin, Shahreen Kasim. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Google, *Google Photos – All your photos organized and easy to find*, 2019. [Online]. Available: https://photos.google.com/
[2] GMBRR, *Memories For Life*, 2025. [Online]. Available: https://gmbrr.com/
[3] Flickr, *Flickr*, Dec. 4, 2018. [Online]. Available: https://www.flickr.com/
[4] J. Shore and S. Warden, *The Art of Agile Development*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2021.
[5] S. Singh and S. V. A. V. Prasad, "Techniques and Challenges of Face Recognition: A Critical Review," *Procedia Computer Science*, vol. 143, pp. 536–543, 2018. https://doi.org/10.1016/j.procs.2018.10.427
[6] L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," *IEEE Access*, vol. 8, pp. 139110–139120, 2020. https://doi.org/10.1109/access.2020.3011028
[7] K. Teoh, R. Ismail, S. Naziri, R. Hussin, M. Isa and M. Basir, "Face Recognition and Identification using Deep Learning Approach," *Journal of Physics: Conference Series*, vol. 1755, no. 1, p. 012006, 2021. https://doi.org/10.1088/1742-6596/1755/1/012006
[8] I. Oruc, B. Balas and M. S. Landy, "Face perception: A brief journey through recent discoveries and current directions," *Vision Research*, vol. 157, pp. 1–9, 2019. https://doi.org/10.1016/j.visres.2019.06.005
[9] D. Singh and D. Arora, "Cloud-Based Object Detection Model Using Amazon Rekognition," in *International Conference on Recent Developments in Cyber Security*, Singapore: Springer Nature Singapore, pp. 165–177, 2023.
[10] A. F. Ngah and N. A. Abdullah, "FaceLogix: Face Recognition Employee Attendance System with Redundancy Webcam for MD1010 Engineering," *Applied Information Technology and Computer Science (AITCS)*, vol. 5, no. 2, pp. 19–37, 2024. [Online]. Available: https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/16450
[11] *View of Face Recognition Attendance System with Face Rolling Motion on Android*, UTHM, 2024. [Online]. Available: https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/12278/4200
[12] Sophia, "International Journal of Scientific Research and Technology," *International Journal of Scientific Research and Technology*, vol. 02, no. 11, p. 1, 2024. https://doi.org/10.5281/zenodo.14306034
[13] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 815-823.
[14] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). S^3FD: Single shot scale-invariant face detector. *Proceedings of the IEEE International Conference on Computer Vision*, 192-201.

Applied Information Technology and Computer Science Vol. 6 No. 2 (2025) p. 150-169

Appendix A: Gantt Chart