FIT5225 - BirdTag AWS-powered Serverless Media Storage

Assignment 3 Team Report

Group Number – 178
Submission Date: 10th June 2025
Git Repository: https://git.infotech.monash.edu/fit5225-group178-birdtag/birdtag-aws

Lucid Chart Link: https://lucid.app/lucidchart/40b7c937-6688-4256-a6ad-0e397bd3983a/edit?viewport_loc=-1184%2C-1071%2C6666%2C3520%2CGPOmuZ498OgT&invitationId=inv_612ce153-d5d3-43bc-9808-e1d95ddb8bc6

Team Members and Contributions

Role	Name	Student ID	Contribution
AWS Infrastructure & DevOps	Anmol Salwan	34560203	AWS S3, DynamoDB, IAM Roles, Lambda Triggers, SNS Setup, ECR integration, CloudWatch logging setup.
Backend Integration	Glenn Varghese George	34448543	Lambda logic development, integration of YOLO-based ML model, thumbnail generation, DynamoDB integration, metadata handling.
API & Query Engine	Aditya Mehrotra	35242868	API Gateway configuration, query API endpoints (search, delete, manual tagging), HTTP methods implementation, security integrations.
Frontend & Authentication	Varun Kashyap	34407693	AWS Cognito authentication setup, React frontend implementation, user interface development for media uploads, querying, and preview.

Project Introduction

BirdTag is an innovative AWS-powered solution aimed at providing scalable, efficient, and reliable serverless storage and automated tagging of various media types such as images, audios, and videos. By leveraging AWS infrastructure and a sophisticated machine learning model, BirdTag enables efficient identification and tagging of bird species, streamlining media organization and retrieval.

Detailed Architecture Overview

BirdTag's architecture is a fully serverless design, leveraging AWS services optimized for scalability, high availability, and ease of management:

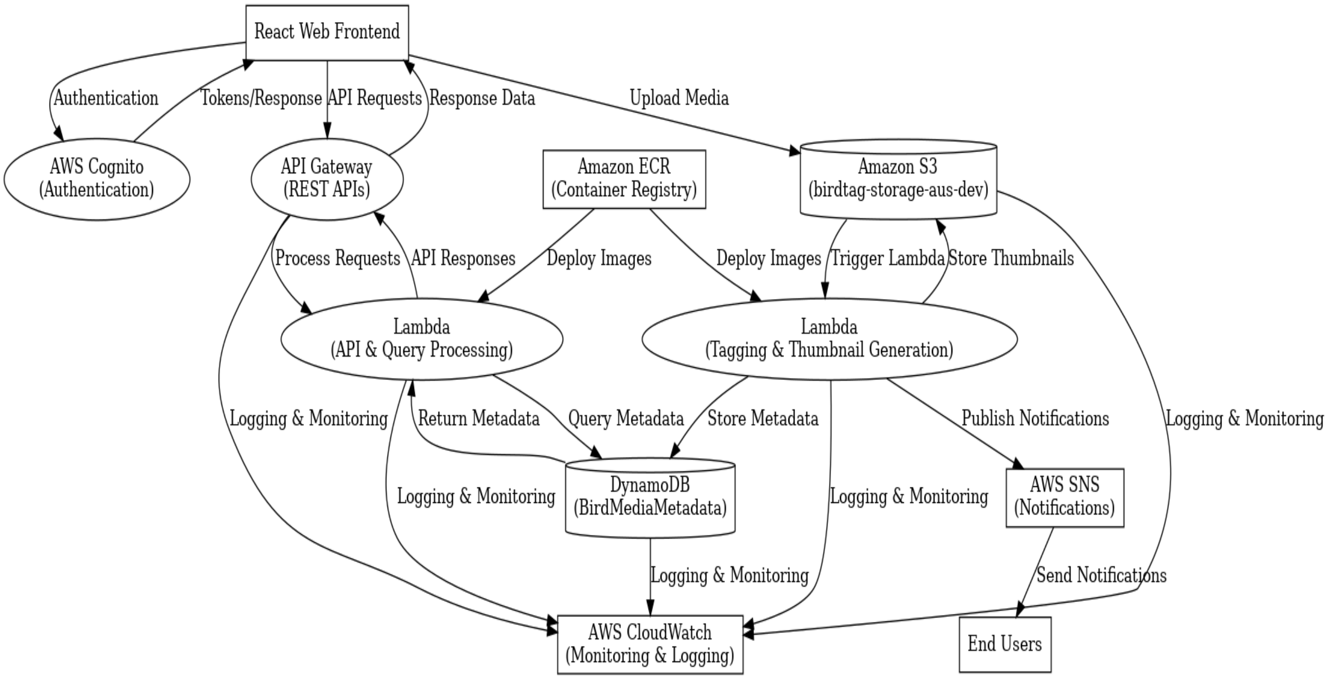
- AWS S3 (birdtag-storage-aus-dev): Secure and scalable storage for media files, model files, and generated thumbnails.
- AWS DynamoDB (BirdMediaMetadata): Highly available and performant NoSQL database storing metadata including tags, file types, and media locations.
- AWS Lambda: Serverless compute platform executing backend logic for tagging media, generating thumbnails, and metadata management.
- AWS API Gateway: Provides secure, RESTful APIs to access and manage media resources effectively.
- AWS Cognito: Robust user authentication and authorization management.
- AWS SNS: Event-driven notification system delivering timely tag-based alerts.
- Amazon ECR: Container registry hosting containerized Lambda functions with embedded machine learning models.
- AWS CloudWatch: Monitoring, logging, and observability for system operations.

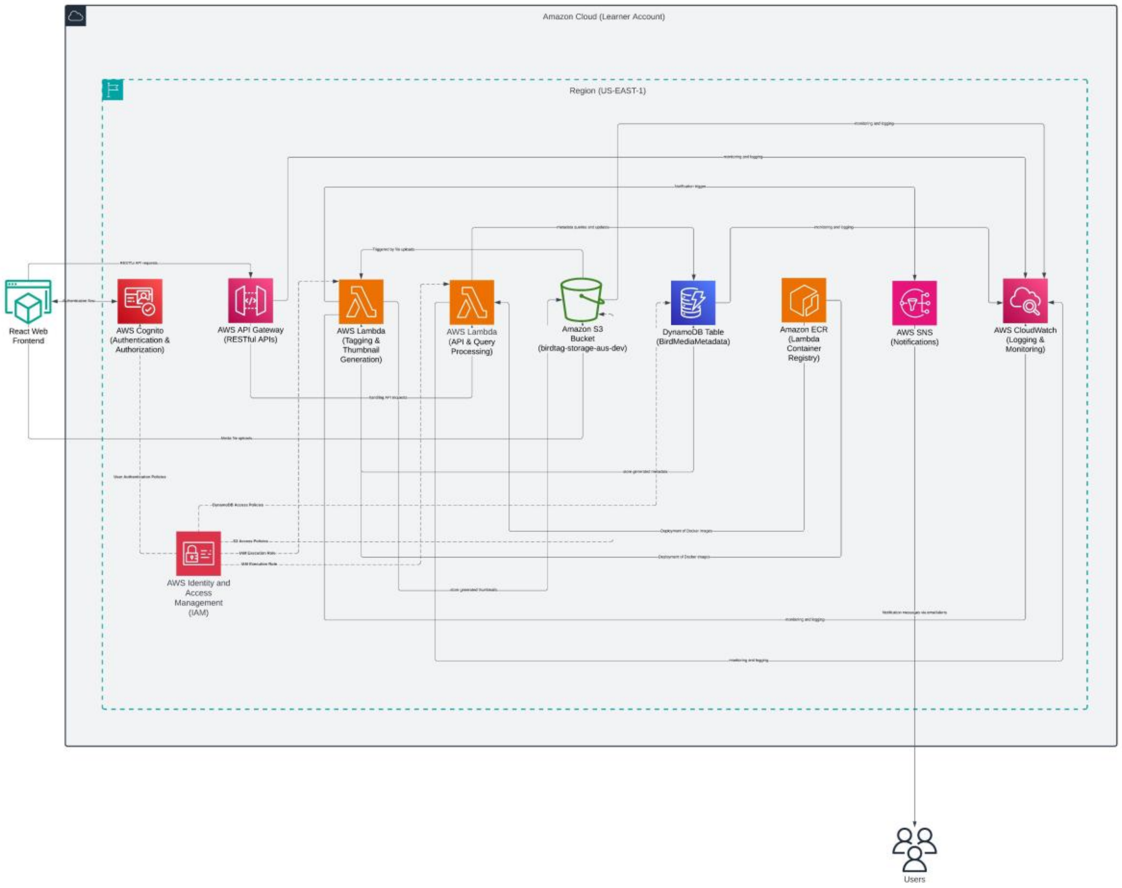
While the architecture diagram visually depicts the processing flow for a single media type (e.g., images), the design and implementation are fully generalized to support all required formats, including audio and video files. The same serverless workflow-comprising upload to Amazon S3, automatic Lambda-based processing (tagging, metadata extraction, and optional thumbnail or waveform generation), metadata storage in DynamoDB, and subsequent API-based querying-applies seamlessly to all supported media. This ensures consistent, scalable, and extensible architecture regardless of the type of media ingested by the BirdTag system.

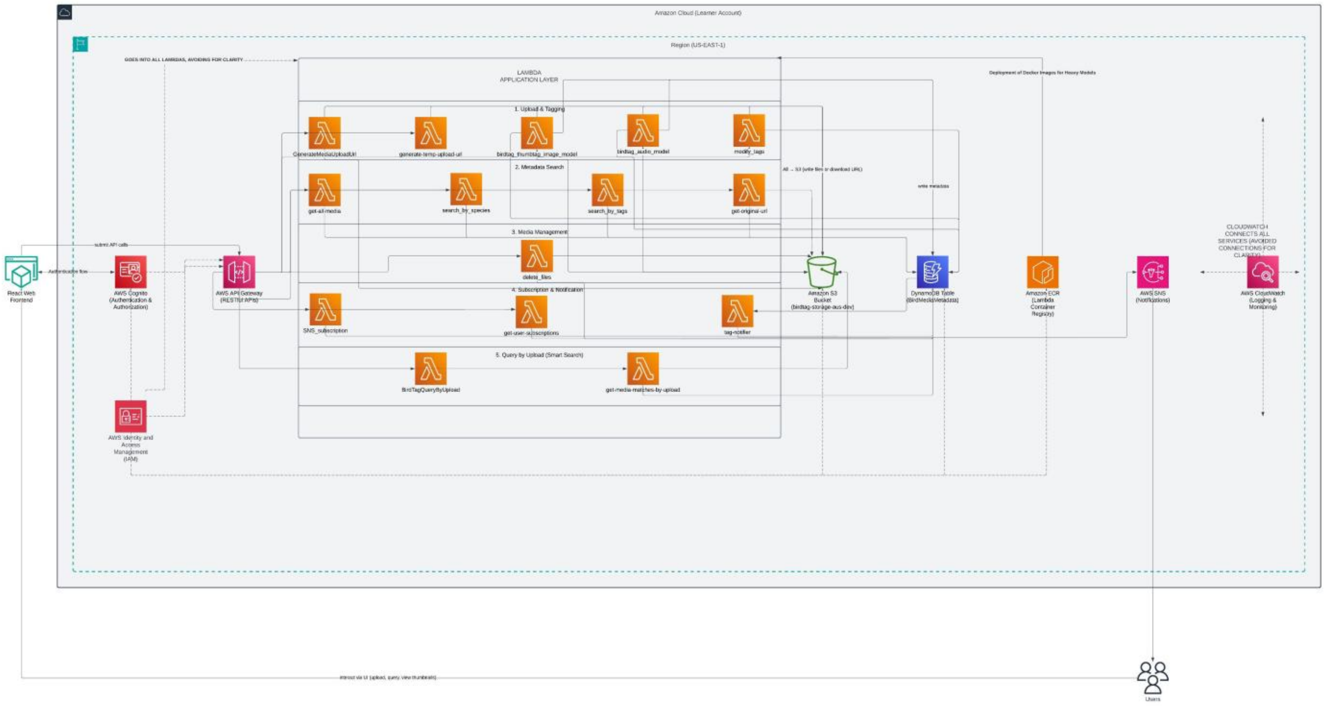
The ***first diagram*** *provides a high-level overview of the implemented architecture, illustrating the primary components and their interactions within the BirdTag system. It focuses on the broader system flow from the frontend interface through to backend processing and storage.*

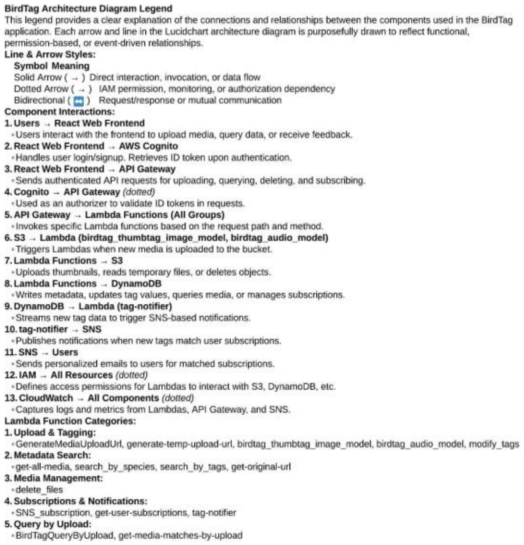
The ***second diagram*** *presents a more detailed, service-level architecture. It maps out individual AWS services, Lambda functions, and integration points, offering a comprehensive view of how media processing, metadata storage, querying, and notification mechanisms are orchestrated in the deployed solution.*

Architecture Diagram:




Amazon Cloud (Learner Account)
Region (US-EAST-1)
React Web Frontend
AWS Cognito (Authentication & Authorization)
AWS API Gateway (RESTful APIs)
AWS Lambda (Tagging & Thumbnail Generation)
AWS Lambda (API & Query Processing)
Amazon S3 Bucket (birdtag-storage-aus-dev)
DynamoDB Table (BirdMediaMetadata)
Amazon ECR (Lambda Container Registry)
AWS SNS (Notifications)
AWS CloudWatch (Logging & Monitoring)
AWS Identity and Access Management (IAM)
Users
RESTful API requests
Authentication flow
Triggered by file uploads
metadata queries and updates
Notification trigger
monitoring and logging
monitoring and logging
monitoring and logging
handling API requests
Media file uploads
store generated metadata
User Authentication Policies
DynamoDB Access Policies
S3 Access Policies
IAM Execution Role
IAM Execution Role
Deployment of Docker images
Deployment of Docker images
store generated thumbnails
Notification messages via email/alerts
monitoring and logging
monitoring and logging


Amazon Cloud (Learner Account)
Region (US-EAST-1)
GOES INTO ALL LAMBDAS, AVOIDING FOR CLARITY
LAMBDA
APPLICATION LAYER
Deployment of Docker Images for Heavy Models
1. Upload & Tagging
GenerateMediaUploadUrl
generate-temp-upload-url
birdtag_thumbtag_image_model
birdtag_audio_model
modify_tags
2. Metadata Search
get-all-media
search_by_species
search_by_tags
get-original-url
All → S3 (write files or download URL)
write metadata
3. Media Management
delete_files
4. Subscription & Notification
SNS_subscription
get-user-subscriptions
tag-notifier
5. Query by Upload (Smart Search)
BirdTagQueryByUpload
get-media-matches-by-upload
submit API calls
Authentication flow
React Web Frontend
AWS Cognito (Authentication & Authorization)
AWS API Gateway (RESTful APIs)
AWS Identity and Access Management (IAM)
Amazon S3 Bucket (birdtag-storage-aus-dev)
DynamoDB Table (BirdMediaMetadata)
Amazon ECR (Lambda Container Registry)
AWS SNS (Notifications)
CLOUDWATCH CONNECTS ALL SERVICES (AVOIDED CONNECTIONS FOR CLARITY)
AWS CloudWatch (Logging & Monitoring)
Interact via UI (upload, query, view thumbnails)
Users



BirdTag Architecture Diagram Legend

This legend provides a clear explanation of the connections and relationships between the components used in the BirdTag application. Each arrow and line in the Lucidchart architecture diagram is purposefully drawn to reflect functional, permission-based, or event-driven relationships.

Line & Arrow Styles:

Symbol	Meaning
Solid Arrow (→)	Direct interaction, invocation, or data flow
Dotted Arrow (→)	IAM permission, monitoring, or authorization dependency
Bidirectional (↔)	Request/response or mutual communication

Component Interactions:

1. **Users → React Web Frontend**
   - Users interact with the frontend to upload media, query data, or receive feedback.
2. **React Web Frontend → AWS Cognito**
   - Handles user login/signup. Retrieves ID token upon authentication.
3. **React Web Frontend → API Gateway**
   - Sends authenticated API requests for uploading, querying, deleting, and subscribing.
4. **Cognito → API Gateway** *(dotted)*
   - Used as an authorizer to validate ID tokens in requests.
5. **API Gateway → Lambda Functions (All Groups)**
   - Invokes specific Lambda functions based on the request path and method.
6. **S3 → Lambda (birdtag_thumbtag_image_model, birdtag_audio_model)**
   - Triggers Lambdas when new media is uploaded to the bucket.
7. **Lambda Functions → S3**
   - Uploads thumbnails, reads temporary files, or deletes objects.
8. **Lambda Functions → DynamoDB**
   - Writes metadata, updates tag values, queries media, or manages subscriptions.
9. **DynamoDB → Lambda (tag-notifier)**
   - Streams new tag data to trigger SNS-based notifications.
10. **tag-notifier → SNS**
    - Publishes notifications when new tags match user subscriptions.
11. **SNS → Users**
    - Sends personalized emails to users for matched subscriptions.
12. **IAM → All Resources** *(dotted)*
    - Defines access permissions for Lambdas to interact with S3, DynamoDB, etc.
13. **CloudWatch → All Components** *(dotted)*
    - Captures logs and metrics from Lambdas, API Gateway, and SNS.

Lambda Function Categories:

1. **Upload & Tagging:**
   - GenerateMediaUploadUrl, generate-temp-upload-url, birdtag_thumbtag_image_model, birdtag_audio_model, modify_tags
2. **Metadata Search:**
   - get-all-media, search_by_species, search_by_tags, get-original-url
3. **Media Management:**
   - delete_files
4. **Subscriptions & Notifications:**
   - SNS_subscription, get-user-subscriptions, tag-notifier
5. **Query by Upload:**
   - BirdTagQueryByUpload, get-media-matches-by-upload

LUCID CHART LINK

Implementation Details

AWS Infrastructure & DevOps (Anmol Salwan):

Anmol provisioned comprehensive AWS infrastructure components. He configured the dedicated S3 bucket (birdtag-storage-aus-dev) ensuring secure storage with encryption and appropriate access policies. DynamoDB table (BirdMediaMetadata) was set up with a schema optimized for querying efficiency. He also created secure IAM roles and policies, ensuring Lambda functions operate with necessary and minimum privileges. AWS SNS was integrated to provide tag-based notifications effectively. Anmol further established ECR integration for efficient Lambda deployments and set up detailed CloudWatch logging for effective monitoring and debugging.

Backend Integration (Glenn Varghese George):

Glenn managed the backend infrastructure, implementing Lambda functions to handle key logic, such as integrating the YOLO machine learning model for automated bird species tagging. He developed functions for automatic thumbnail generation upon image upload and efficiently managed the storage and retrieval of media metadata in DynamoDB.

API & Query Engine (Aditya Mehrotra):

Aditya developed robust and secure API endpoints using AWS API Gateway. His contributions include endpoints for querying media based on tags, managing tagging operations (adding, deleting tags), and handling deletion requests. Aditya meticulously secured API endpoints using proper authorization and validation techniques to ensure data integrity and security.

Frontend & Authentication (Varun Kashyap):

Varun set up AWS Cognito to provide secure user authentication and authorization, ensuring a seamless and safe user experience. He developed an intuitive React-based frontend, offering users an efficient interface to upload media, execute queries, manage tags, and preview content with ease. Varun emphasized responsiveness and accessibility in UI design, enhancing the overall user interaction experience.

User Guide

Getting Started with BirdTag:
1. User Authentication: Register or log in securely via AWS Cognito-enabled React frontend.
2. Media Upload: Easily upload media files (images, audios, videos) using the frontend interface.
3. Automated Processing: Uploaded media undergoes automated tagging and thumbnail creation.
4. Tag-based Queries: Execute queries via intuitive frontend to retrieve specific media files using tags.
5. Media Management: Utilize provided APIs for tag management (manual addition, deletion) and resource deletion.
6. Receive Notifications: Opt-in to tag-based notifications through AWS SNS.

This guide provides step-by-step instructions to help users effectively interact with the BirdTag platform -a serverless media storage and bird species detection system hosted on AWS.

---

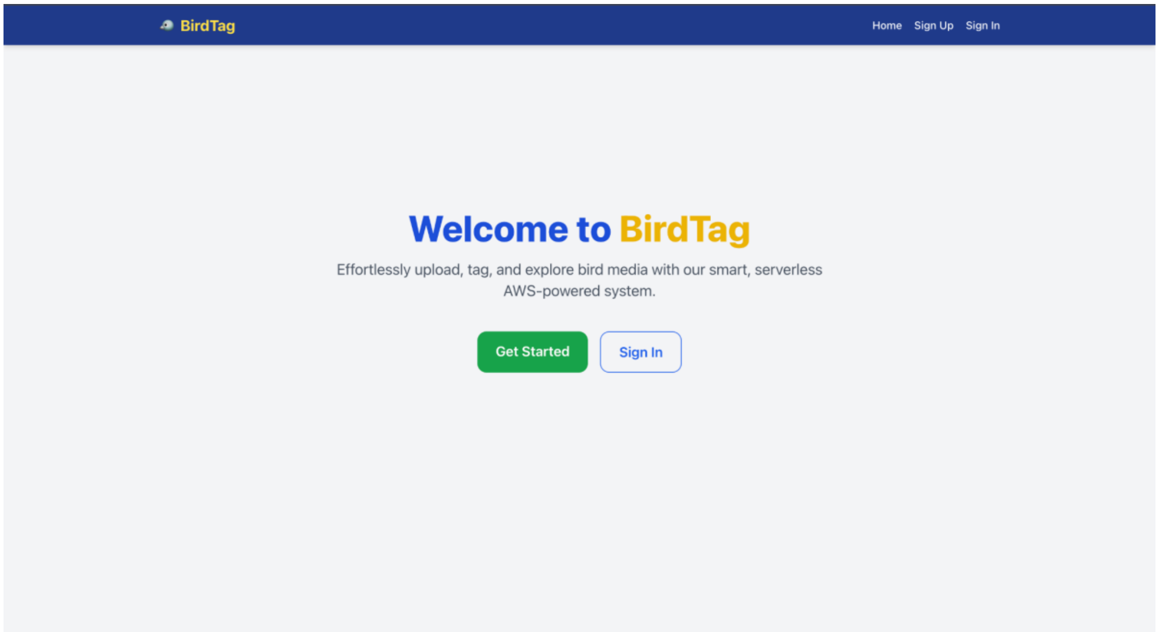
1. Authentication and Authorization

BirdTag uses **AWS Cognito** to securely manage user sign-up and login. Users must create an account and log in to access features like media uploads, queries, and personalized notifications.

---

2. Home Page / Landing Page

Upon visiting BirdTag, users are greeted with a **Home Page** offering clear options to either **Log In** or **Sign Up**.

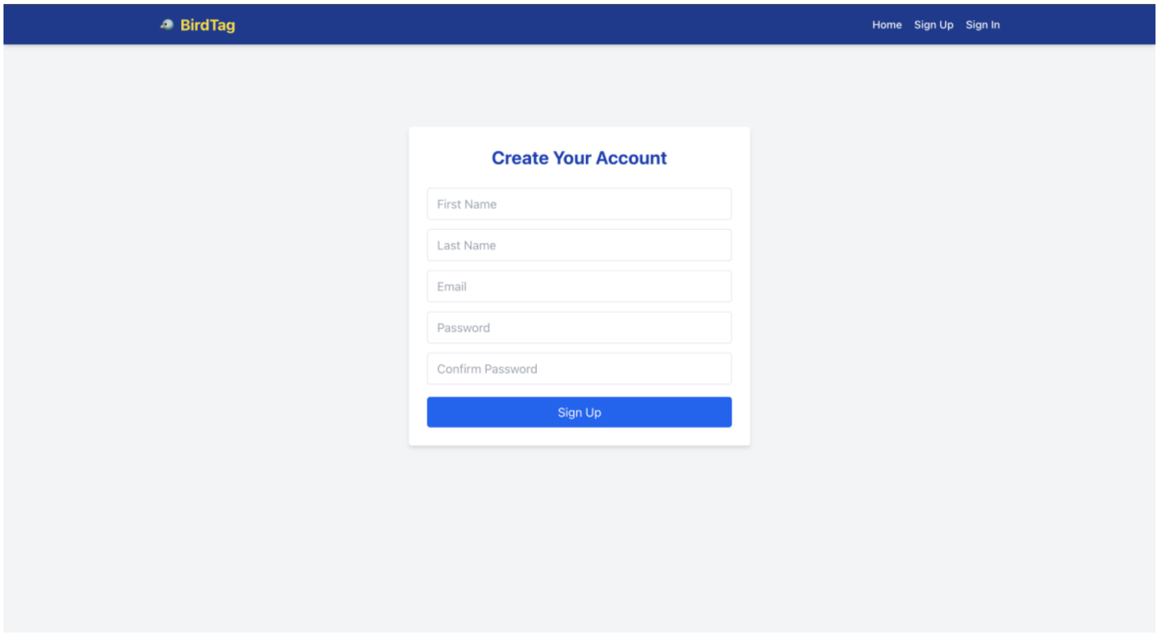


Welcome to BirdTag

Effortlessly upload, tag, and explore bird media with our smart, serverless AWS-powered system.

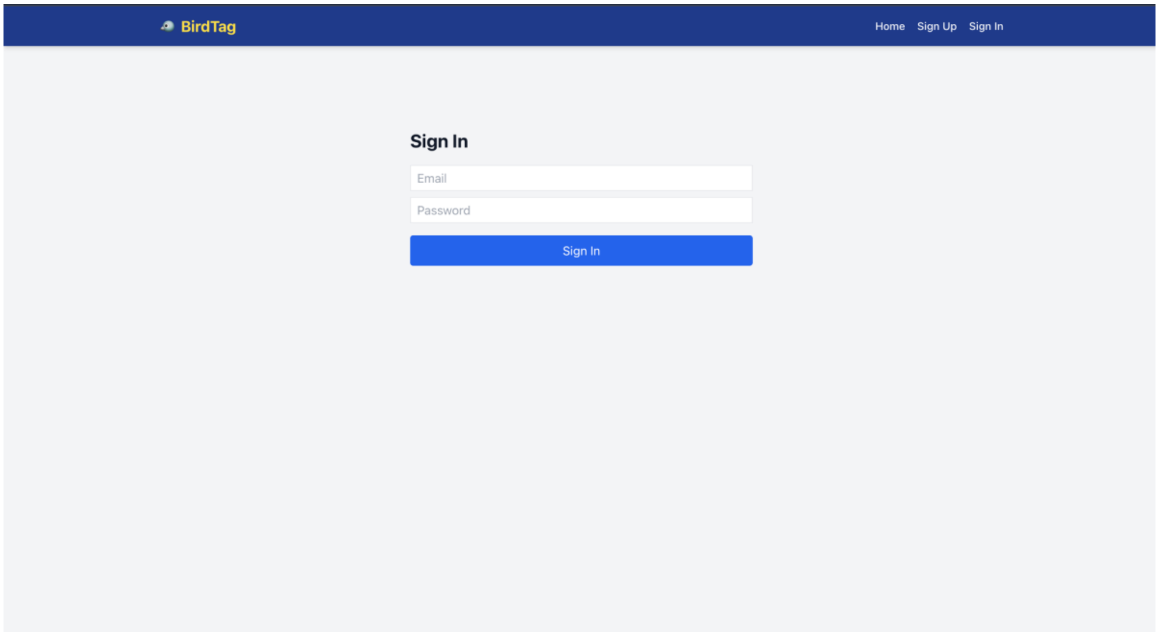
3. Sign-Up Page

New users can sign up using their email and a secure password. Upon successful sign-up, Cognito stores the user in a secure user pool.



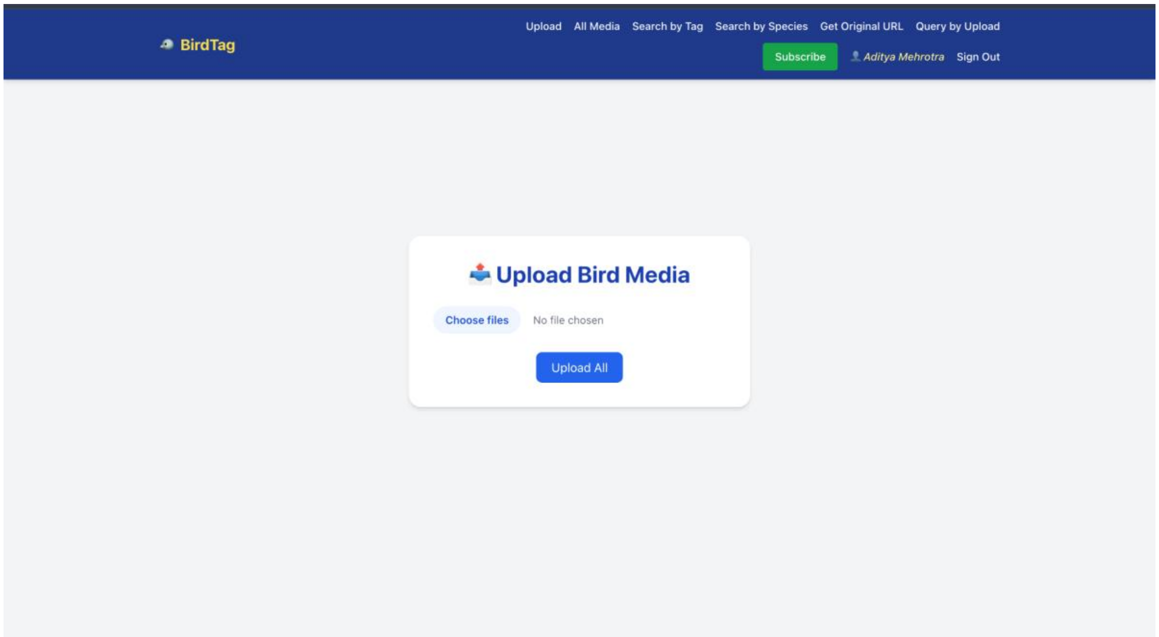
4. Login Page

Existing users can log in using their registered credentials. This generates a secure Cognito token for subsequent requests.



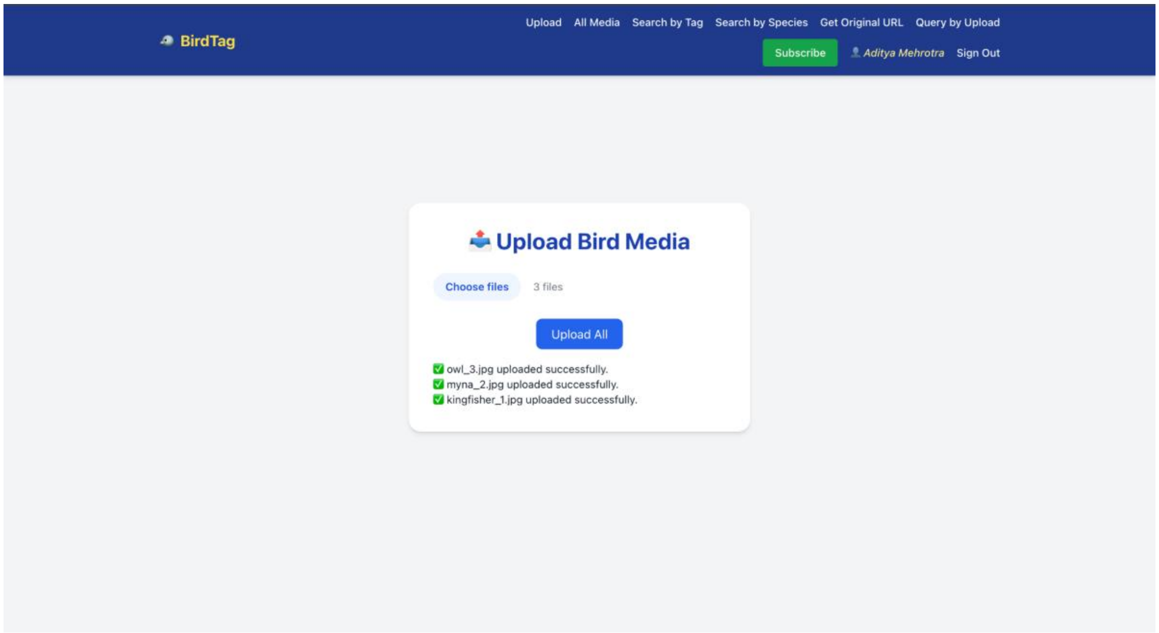
5. Post-Login Landing Page

After a successful login, users are redirected to the **Dashboard**, which provides access to media upload, queries, and notifications.



6. Uploading Media to S3

Users can upload images, videos, or audio files to the platform. Upon upload, a Lambda function is triggered to generate tags and store metadata in DynamoDB. For image uploads, a thumbnail is generated and displayed alongside detected bird species tags shown under All Media Files.



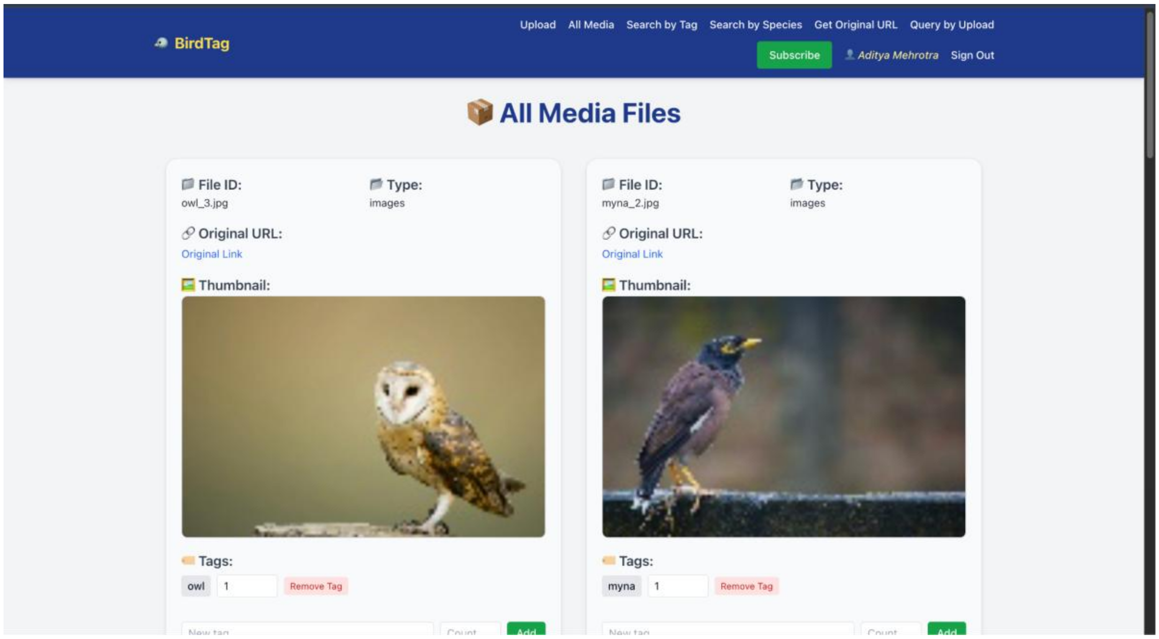
Upload Bird Media

3 files

✅ owl_3.jpg uploaded successfully.
✅ myna_2.jpg uploaded successfully.
✅ kingfisher_1.jpg uploaded successfully.

7. Thumbnail and Tagging Output

When the image and Thumbnails are retrieved a **pre-signed S3 URL** is generated for the user to be able to view them, ensuring secure media storage.



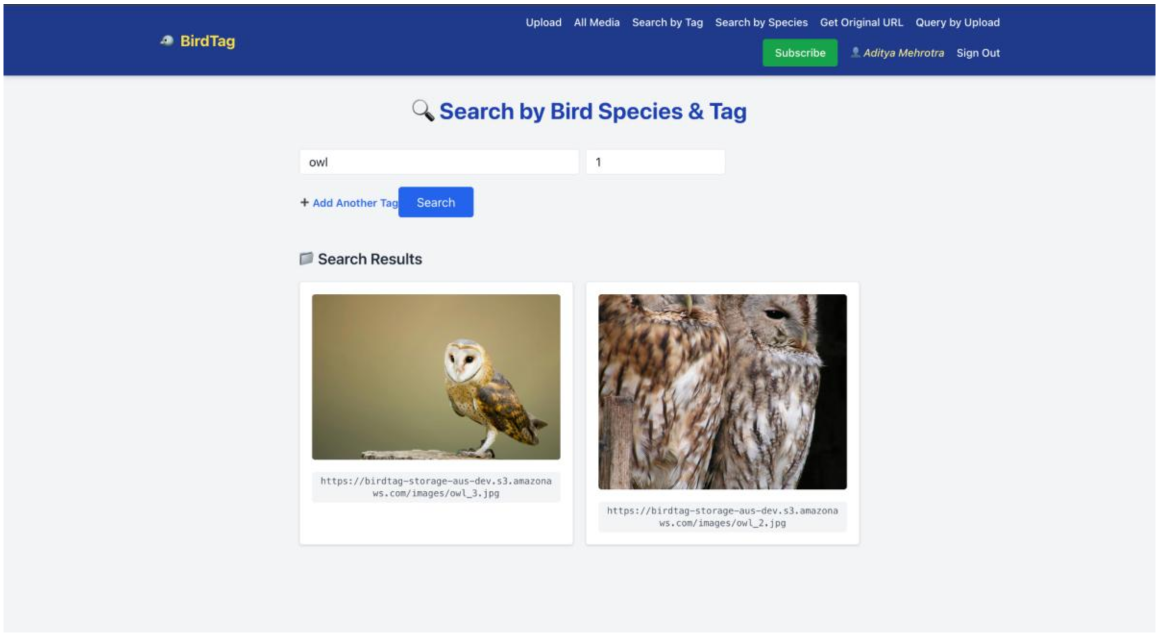
All Media Files

File ID: owl_3.jpg
Type: images
Original URL: Original Link
Thumbnail:
Tags: owl 1

File ID: myna_2.jpg
Type: images
Original URL: Original Link
Thumbnail:
Tags: myna 1

8. Querying Files - Search by Tags

Users can search files using a specific tag (e.g., "crow", "sparrow"). A Lambda function queries DynamoDB and returns media files where the tag exists.



🔍 Search by Bird Species & Tag

owl | 1

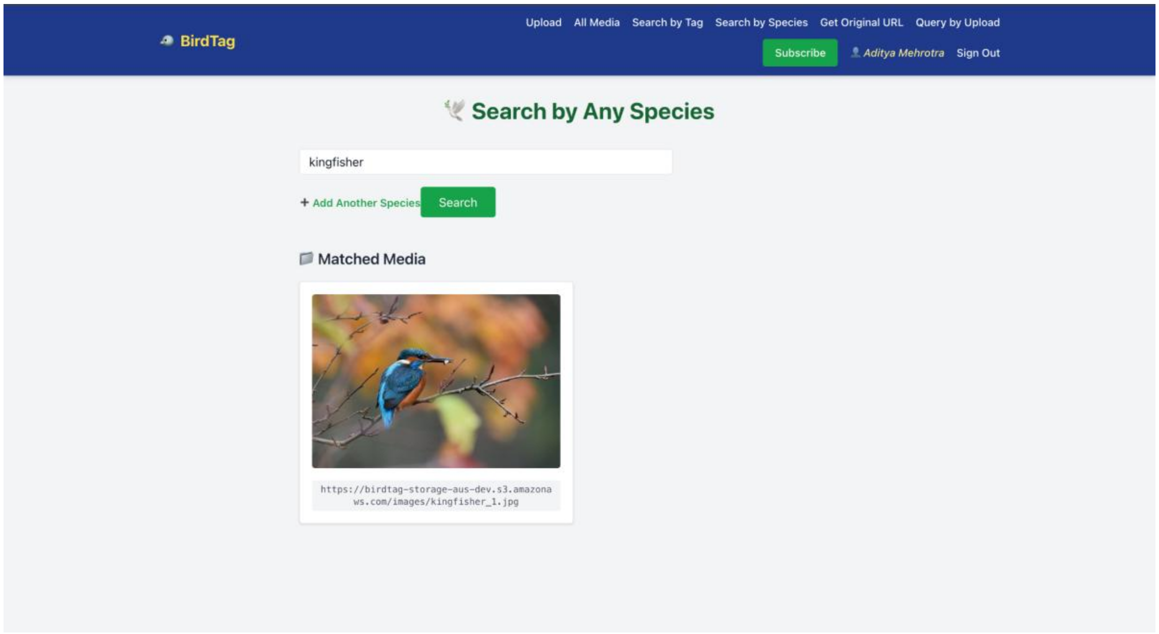
Search Results

https://birdtag-storage-aus-dev.s3.amazonaws.com/images/owl_3.jpg

https://birdtag-storage-aus-dev.s3.amazonaws.com/images/owl_2.jpg

9. Querying Files - Search by Species Count

Users can search for media by entering the **name of a bird species** (e.g., "crow", "pigeon"). The system returns **all matching files** (images, videos, or audios) where the tag metadata includes **at least one occurrence** of the specified species.



Search by Any Species

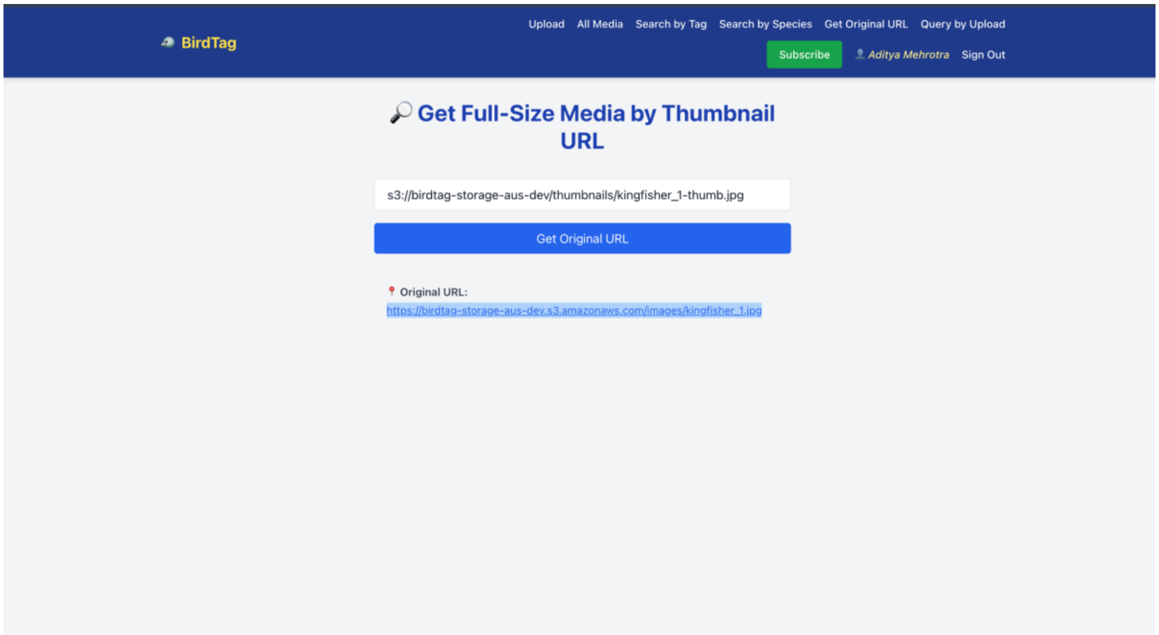
kingfisher

Matched Media

https://birdtag-storage-aus-dev.s3.amazonaws.com/images/kingfisher_1.jpg

10. Retrieve Original File from Thumbnail

If a user has only the thumbnail URL, this feature allows them to retrieve the original full-size file via a DynamoDB lookup.



🔎 Get Full-Size Media by Thumbnail URL

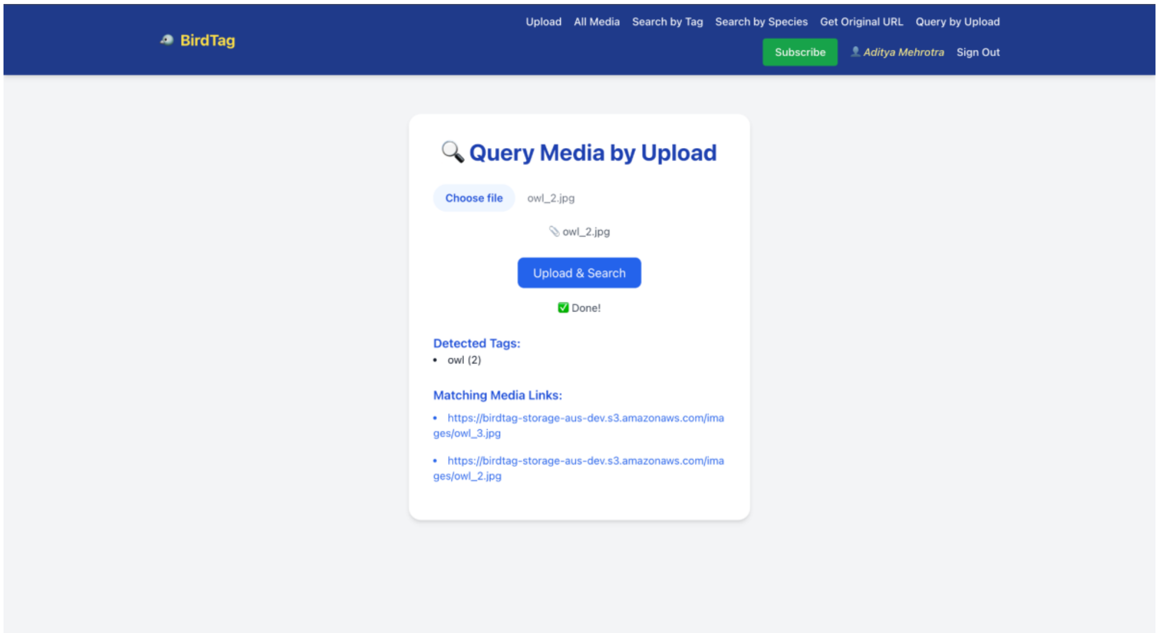
s3://birdtag-storage-aus-dev/thumbnails/kingfisher_1-thumb.jpg

📍 Original URL:
https://birdtag-storage-aus-dev.s3.amazonaws.com/images/kingfisher_1.jpg

---

11. Find Similar Files Based on a File’s Tags

Users can input a file ID, and the system will **find other files** that share overlapping tag information (e.g., same birds detected).



🔍 Query Media by Upload

owl_2.jpg

📎 owl_2.jpg

✅ Done!

Detected Tags:

- owl (2)

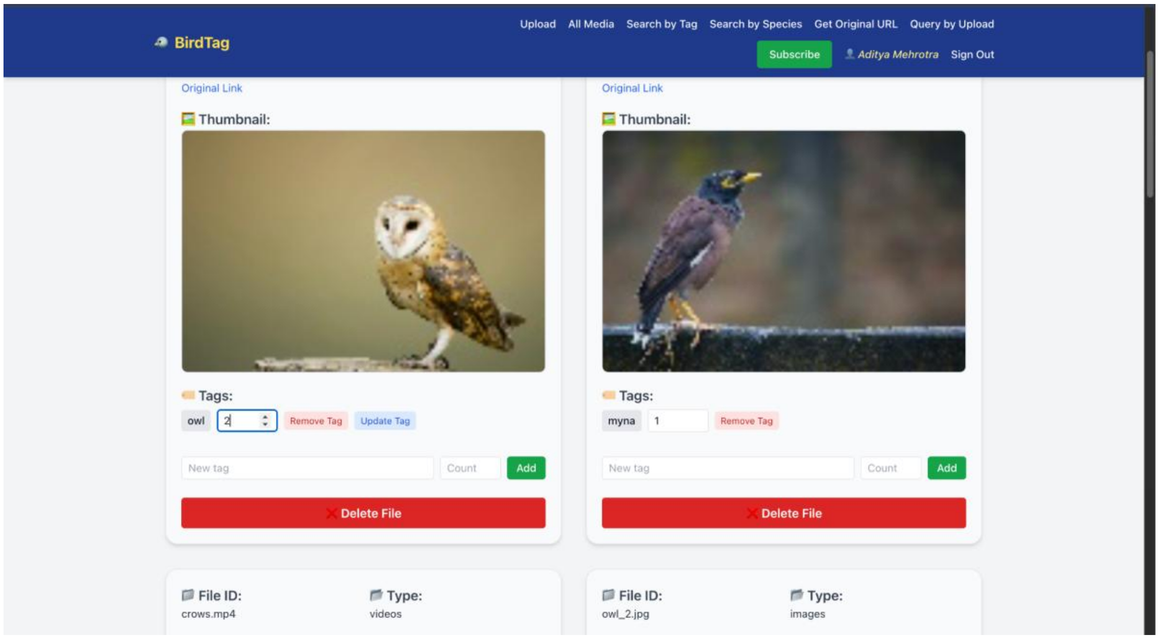
Matching Media Links:

- https://birdtag-storage-aus-dev.s3.amazonaws.com/images/owl_3.jpg
- https://birdtag-storage-aus-dev.s3.amazonaws.com/images/owl_2.jpg

---

12. Manual Tag Management and Deletion

Users can manually **add or remove tags** to/from a file, and also **delete one or more files** from the system. Deletion removes metadata from DynamoDB and deletes media from S3.



Tags: owl 2

Tags: myna 1

File ID: crows.mp4

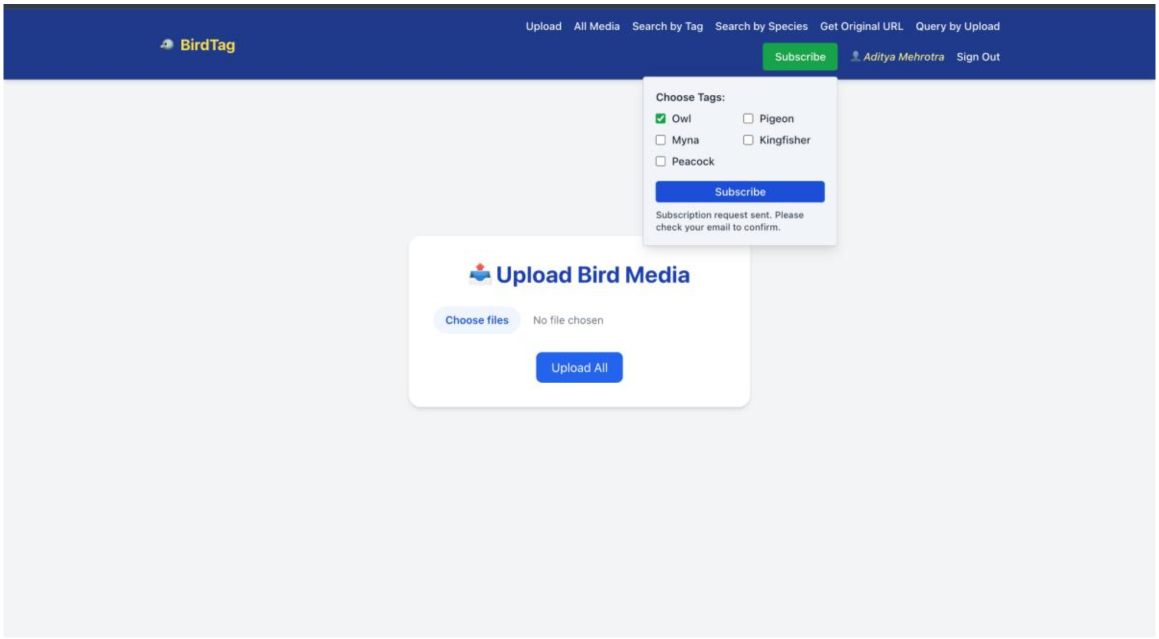
Type: videos

File ID: owl_2.jpg

Type: images

13. Tag-based Notifications

Users can **subscribe to specific bird species** (tags) and receive **email notifications** if a newly uploaded file is tagged with that species.

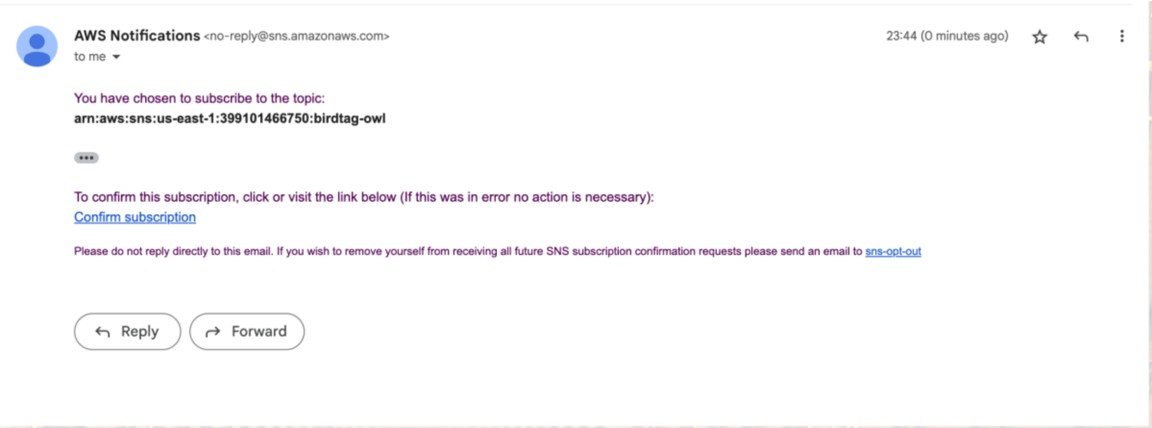


Choose Tags:

☑ Owl
☐ Pigeon
☐ Myna
☐ Kingfisher
☐ Peacock

Subscribe

Subscription request sent. Please check your email to confirm.



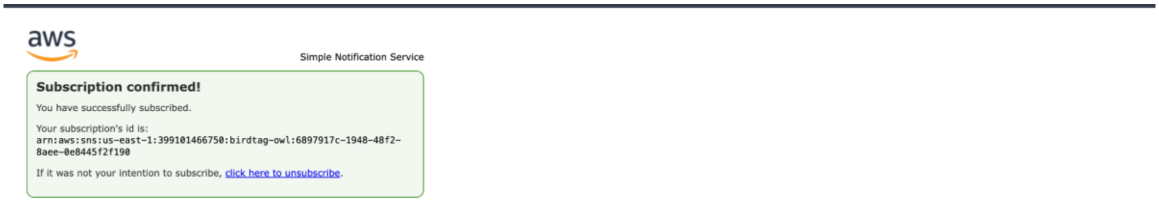
AWS Notifications <no-reply@sns.amazonaws.com>
to me

23:44 (0 minutes ago)

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:399101466750:birdtag-owl

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out



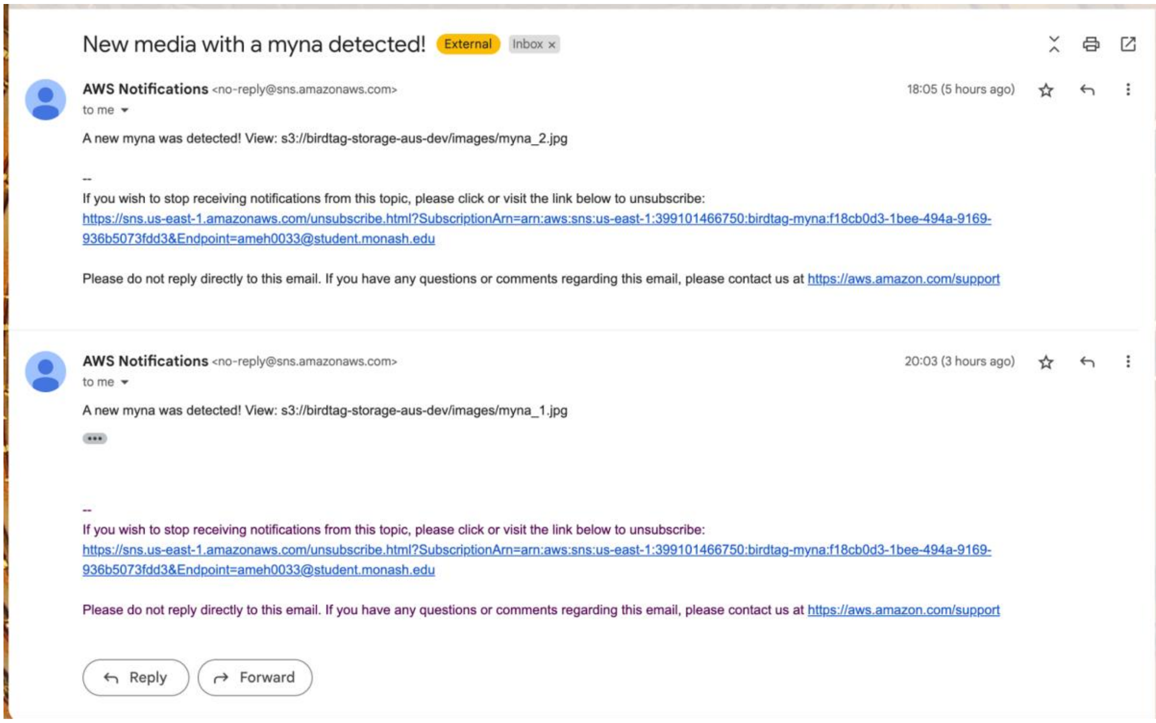
Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:us-east-1:399101466750:birdtag-owl:6897917c-1948-48f2-8aee-0e8445f2f190

If it was not your intention to subscribe, click here to unsubscribe.



New media with a myna detected! External Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 18:05 (5 hours ago)
to me

A new myna was detected! View: s3://birdtag-storage-aus-dev/images/myna_2.jpg

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:399101466750:birdtag-myna:f18cb0d3-1bee-494a-9169-936b5073fdd3&Endpoint=ameh0033@student.monash.edu

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

AWS Notifications <no-reply@sns.amazonaws.com> 20:03 (3 hours ago)
to me

A new myna was detected! View: s3://birdtag-storage-aus-dev/images/myna_1.jpg

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:399101466750:birdtag-myna:f18cb0d3-1bee-494a-9169-936b5073fdd3&Endpoint=ameh0033@student.monash.edu

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Repository

The complete codebase and documentation are maintained at:
https://git.infotech.monash.edu/fit5225-group178-birdtag/birdtag-aws

Conclusion and Future Work

The BirdTag project has successfully delivered a highly scalable, secure, and fully serverless solution, harnessing the power of AWS services. Future enhancements could include support for additional media types, optimizing the ML model for faster inference times, and further refining the frontend UI for improved usability. The implemented solution demonstrates significant practical value and establishes a solid foundation for continued innovation and scalability.

Acknowledgment of AI Tools

AI tools served as accelerators for learning, debugging, and productivity, but the core technical implementation, architecture, design, and integration were carried out by the BirdTag team members. We affirm that the usage of AI was within the guidelines of academic integrity and ethical responsibility.