

Weekly Report: Oudarja Barman Tanmoy

Alpha AI

Week-07 (May 05 - May 09)

Initial Setup and Version Control Integration: The DRMS project repository was initialized and prepared for structured development. Below initial step of development are given :

1. A placeholder .gitkeep file was pushed to the frontend directory to preserve the folder's presence in Git and support modular development from the beginning.
2. The initial commit also included scaffolding necessary for setting up the backend and frontend as separate layers.
3. The initial commit also included scaffolding necessary for setting up the backend and frontend as separate layers.

Backend Development Progress :

a) Employee Feature Module (CRUD with DynamoDB):

1. A complete backend feature to manage employee metadata was implemented.

2. This included full CRUD (Create, Read, Update, Delete) operations using FastAPI for API endpoints and DynamoDB for storage.
3. The model included attributes like employee_id, name, and additional metadata.
4. Endpoint design followed RESTful principles and supported interaction with structured data.

b) Image Management Module (S3 Integration) :

1) Upload and Delete Functionality:

- I. Developed robust backend endpoints for uploading employee-related images to AWS S3.
- II. Used pre-signed URLs to securely handle image uploads and enable time-limited access.
- III. Implemented delete functionality that allows secure removal of images from the S3 bucket and corresponding metadata cleanup from DynamoDB.

2) Image Query API (Metadata Retrieval):

- I. Built APIs to retrieve image metadata based on dynamic parameters such as employee_id and tags.
- II. Supported advanced querying using nested filters to allow targeted searches across employee-associated records.

- III. This served as the foundation for frontend search and filtering capabilities.

Frontend Development with ReactJS :

a) React App Initialization and Employee Form:

1. Initialized the frontend using Create React App.
2. Developed EmployeeForm.js, a dynamic form to add employee details.
3. Integrated with the backend API to fetch employee data from DynamoDB and render it on the UI.
4. Form fields and design followed a responsive layout pattern.

b) Image Search UI Module:

1. Implemented a full-fledged image search interface.
2. Enabled filtering based on multiple tags and associated employees using Axios queries.
3. Integrated scroll-to-top behavior for better UX and applied modern styling for clean design.
4. Conditional rendering was added to adapt the UI based on the presence or absence of search results.

Bug Fixes and Enhancements

a) Axios Query Fix for Multi-Tag Search:

1. Fixed a critical bug where multiple tags in search queries were not being properly serialized.
2. Implemented `paramsSerializer` using the `qs` package to correctly handle query arrays and ensure accurate filtering.
3. This fix ensured reliable communication between frontend filters and backend API responses.