Weekly Report: MD. Ariful Islam Shakil

Week-06 (April 28 – May 01)

1. AWS Cloud Lecture

AWS Cloud

In our lecture, I learned that Amazon Web Services (AWS) is a cloud computing platform that provides on-demand resources like computing power, storage, and databases. I understood how AWS helps developers deploy and scale applications without managing physical hardware. It offers a range of services that can be integrated into our projects easily.

Amazon DynamoDB

From the session, I came to know that DynamoDB is a fully managed NoSQL database service offered by AWS. It is optimized for high-speed and low-latency operations. I understood the concept of key-value and document data models, and how it supports horizontal scaling automatically. It’s ideal for building serverless applications or storing flexible, unstructured data.

Amazon S3 (Simple Storage Service)

We also learned about Amazon S3, which is an object storage service used for storing files like images, videos, documents, etc. The lecture explained how S3 stores data as objects in buckets and allows secure access using features like pre-signed URLs. It’s commonly used for static file hosting, backups, and integration with other AWS services.

2. Digital Resource Management Project

FastAPI Backend Development

Over the last week, I worked extensively with **FastAPI**, focusing on building a robust backend system to support employee data management and storage. This involved several critical components that demonstrate both practical understanding and real-world backend implementation skills.

CRUD Operations with FastAPI (Employee Management System)

I successfully implemented **Create**, **Read**, **Update**, and **Delete (CRUD)** operations using FastAPI. The endpoints are RESTful and designed for clarity and ease of use. These operations include:

- **Creating employee records** via structured JSON data using POST requests.
- **Retrieving all employees or specific records** by ID using GET endpoints.

- **Updating existing records** with partial or full data using PUT.
- **Deleting employees** with DELETE endpoints.

Data Validation and Structuring

To ensure data integrity and clean API requests, I utilized **Pydantic models** for input validation. These models help prevent runtime errors and enforce type safety when creating or updating employee records.

Integration with DynamoDB

I integrated **Amazon DynamoDB** as the primary backend NoSQL database to store employee records. Key operations such as insert, update, delete, and scan were implemented using DynamoDB's Python SDK (boto3), ensuring scalable and fast data access.

AWS Services Integration

I explored AWS service usage in the backend, including:

- **Amazon S3**: Configured AWS S3 for storing image files and other static assets. I familiarized myself with the S3 integration and presigned URL generation another of my DRMS project.
- **IAM Roles and Access Configuration**: Managed AWS credentials securely and granted proper permissions for accessing DynamoDB and S3 via IAM roles.

React Frontend for Employee Management

I also developed the React frontend interface to interact with the API and perform CRUD actions:

Employee List and Form UI

- Used useEffect to fetch and display employees on page load.
- Created a dynamic form to add or update employee information.
- Added dropdowns and input validations for user-friendly data entry.

Functional Integration

- Connected to the FastAPI endpoints using Axios via a custom EmployeeServices class.
- Implemented button triggers and form handlers for:
  - Creating new employee records
  - Editing existing ones
  - Deleting employees with confirmation and state refresh