Technical Solution Overview

# Objective:

The goal of this migration is to transfer scanned images stored in an Oracle database (as BLOBs) to AWS S3, perform CRUD operations through IBM API Connect, and update the Oracle table with the corresponding S3 URL and the upload status.

# Key Components:

1. Oracle Database:  
 - Stores images as BLOB data.  
 - The Oracle table will be updated with the S3 URL and upload status after each image upload.  
  
2. AWS S3:  
 - Destination for storing scanned images in cloud storage.  
 - Unique URL generated for each uploaded image.  
  
3. Spring Boot Application:  
 - Responsible for querying Oracle, uploading images to S3, and updating Oracle with S3 URL and upload status.  
  
4. IBM API Connect (APIC):  
 - Manages the API layer for image operations such as upload, retrieval, update, and deletion.

# Flow Overview:

1. Extract Image from Oracle: The Spring Boot application will connect to Oracle, retrieve the image BLOBs, and prepare the images for upload.  
  
2. Upload Image to S3: Images are uploaded to AWS S3 using the AWS SDK. Each upload will generate a unique URL for the image stored in the S3 bucket.  
  
3. Update Oracle: After a successful image upload, the S3 URL and the status (e.g., success or failure) are updated in the Oracle table for the respective image.  
  
4. CRUD Operations via IBM API Connect: Expose Create, Read, Update, and Delete operations for image management.

# Technical Design Summary:

1. Extract Image Data: The Spring Boot application connects to the Oracle database to retrieve image data (BLOB type) using a SELECT query.  
  
2. Upload to S3: Once the image data is retrieved, it is uploaded to AWS S3 using the AWS SDK. A unique URL is generated for each image stored in S3.  
  
3. Update Oracle: The Oracle table will be updated with the following:  
 - S3 URL: The URL pointing to the image in S3.  
 - Upload Status: Whether the upload was successful or failed (e.g., "Success", "Failure").  
  
4. CRUD via IBM API Connect: Expose the necessary APIs for image operations (via APIC).

# Possible Status Codes:

| \*\*Status Code\*\* | \*\*Description\*\* |  
|-----------------|------------------------------------------------------|  
| `200 OK` | Image successfully uploaded to S3 and Oracle updated with the S3 URL. |  
| `201 Created` | A new image has been uploaded to S3 and metadata added to Oracle. |  
| `400 Bad Request` | The image data or metadata was invalid or incomplete. |  
| `404 Not Found` | The requested image was not found in Oracle or S3. |  
| `500 Internal Server Error` | Server encountered an error while processing the image upload or Oracle update. |  
| `502 Bad Gateway` | The Spring Boot service or APIC failed to communicate with the AWS SDK or Oracle. |  
| `503 Service Unavailable` | The service is temporarily unavailable (e.g., AWS S3 or Oracle is down). |

# Possible Error Codes:

| \*\*Error Code\*\* | \*\*Description\*\* | \*\*Action Required\*\* |  
|----------------|----------------------------------------------------------------------|----------------------------------------------------------------|  
| `1001` | Image extraction from Oracle failed (e.g., BLOB read error). | Check Oracle connection, ensure BLOB data is correct. |  
| `1002` | S3 Upload failed due to invalid file format or size. | Validate image format and size before upload. |  
| `1003` | S3 upload timeout or connection failure. | Check AWS S3 service status and retry the upload. |  
| `1004` | Failed to update Oracle with S3 URL (e.g., database connection error). | Check Oracle database connection and retry the update. |  
| `1005` | Invalid metadata in API request. | Ensure proper image metadata is passed during the API call. |  
| `1006` | S3 URL retrieval failed (image not found in S3). | Check if the image exists in S3, and validate URL generation. |  
| `1007` | Image deletion from S3 failed (e.g., missing permissions). | Check S3 permissions for delete operation. |  
| `1008` | Timeout during API call to APIC or S3. | Investigate network connectivity or timeout configuration. |

# Security Considerations:

1. IAM Roles and Permissions: Ensure that the Spring Boot application and APIC are configured with proper IAM roles to access AWS S3 and Oracle.  
  
2. API Security: Use OAuth or similar secure authentication methods for the API operations managed by IBM API Connect.

# Error Handling:

Implement robust logging for failed uploads and Oracle updates. Retry mechanism for transient errors (e.g., network timeouts). Alerts or notifications for critical errors (e.g., failure to upload to S3, unable to update Oracle).