

**Implementation of ABCBooks Web-Based Application**



**Done By:**

**Bharath Parthasarathy N01609484**

**HUMBER COLLEGE**

**Cloud Solution Architecture**

**Executive Summary**

This technical report outlines the implementation details of the web-based application developed for ABCBooks, focusing on integrating various AWS services to achieve the desired functionalities. The application utilizes Amazon DynamoDB for data storage, AWS Lambda for serverless computing, Amazon API Gateway for handling API requests, and Amazon Cognito for user authentication and authorization. The report discusses the architecture, implementation steps, and achieved objectives.

**1. Introduction**

ABCBooks aims to enhance its book collection management system by introducing a web-based application. The new application is designed to provide functionalities such as displaying, adding, deleting, and updating book entries. Furthermore, security is ensured by integrating Amazon Cognito for user authentication.

**2. Architecture Overview**

The architecture of the ABCBooks web-based application comprises three main components:

A diagram of a computer process

Description automatically generated

1. **Static Frontend:** Hosted on Amazon S3, the front-end is developed using HTML, CSS, and JavaScript to provide a user-friendly interface for interacting with the application.
2. **Serverless Backend:** This component is built using AWS Lambda, Amazon API Gateway, and Amazon Cognito. AWS Lambda functions, written in Python, perform CRUD operations on the DynamoDB table. API Gateway acts as a proxy to trigger Lambda functions based on incoming HTTP requests. Amazon Cognito handles user authentication and authorization.
3. **DynamoDB:** The persistence layer of the application, where book data is stored in a NoSQL database.

**3. Implementation Details**

**Frontend Development**

The frontend is developed using HTML, CSS, and JavaScript to create a responsive and intuitive user interface. The interface allows users to view the list of books, add new books, delete existing ones, and update book details.

**Backend Development**

**Lambda Functions**

Python-based Lambda functions are created to perform CRUD operations on the DynamoDB table. Each function is designed to handle a specific operation:

**List Books:** Retrieves all books from the DynamoDB table.

**Add Book**: Adds a new book entry to the database.

**Delete Book:** Deletes a book from the database.

**Update Book**: Updates non-key attributes of a book.

**API Gateway**

API Gateway is configured to expose RESTful endpoints corresponding to each Lambda function. Integration between API Gateway and Lambda functions allows seamless invocation of functions in response to incoming HTTP requests.

**Amazon Cognito**

Amazon Cognito is utilized to manage user authentication and authorization. A user pool is created to store user accounts, and appropriate authentication flows are configured to ensure secure access to the application.

**DynamoDB Configuration**

A DynamoDB table is created to store book data. Proper indexes and keys are defined to optimize query performance. Conditional expressions are utilized to ensure data consistency during write operations.

**4. Project Objectives Achieved**

Upon completion of the project, the following objectives were achieved:

**Frontend User Interface**: A responsive frontend interface was developed to interact with AWS services.

**Lambda Functions Deployment**: Python-based Lambda functions were created and deployed to perform operations on DynamoDB.

**DynamoDB Operations**: CRUD operations were successfully implemented using DynamoDB, leveraging conditional expressions.

**API Gateway Endpoints**: RESTful endpoints were created and configured in API Gateway to proxy Lambda functions.

**Route 53 :** Using route 53 we redirected our network traffic into the public Domain Name System (DNS) and Using ACM generated a SSL certificate to protect our DNS.

**CloudFront:** We attached the certificate to our web domain to make sure the data is protected and make our website highly available.

**Amazon Cognito Integration**: User authentication and authorization were implemented using Amazon Cognito, ensuring the security of the application.

**Serverless Architecture Implementation**: The entire application was deployed using a serverless architecture on AWS, ensuring scalability and cost-effectiveness.

**5. Application Source Code (https://github.com/BharathParthasarathy/cloudsolutionArchitect.git)**

**6. Conclusion**

The implementation of the web-based application for ABCBooks demonstrates the effective utilization of AWS services to achieve desired functionalities. By leveraging Amazon DynamoDB, AWS Lambda, Amazon API Gateway, and Amazon Cognito, a scalable, secure, and cost-efficient solution was developed to manage the book collection. Future enhancements may include additional features such as search functionality and advanced user management capabilities.