CS 331 – Software Engineering Lab

Assignment 4

I. Selected Software Architecture Style

Microservices Architecture

I-A. Justification Based on Component Granularity

- The system is divided into **independent services**, each handling a **single responsibility**.

- Major services are:

  - **Frontend Service** – user interface and interaction

  - **Backend API Service** – authentication, job management, uploads

  - **Worker Service** – OCR, document parsing, Excel generation

- Each service has:

  - Its own codebase

  - Independent runtime and configuration

- Communication between services is done using:

  - **REST APIs**

  - **Message queue (AWS SQS)**

- Components are **loosely coupled** and **independently deployable**, which is a core property of Microservices Architecture.

I-B. Why Microservices is the Best Choice for This Project

- **Scalability**

    - OCR and document processing can scale independently using worker services.

    - Frontend and backend scale based on user traffic.

- **Maintainability**

    - Clear separation of concerns makes the system easy to modify and debug.

    - Changes in one service do not affect others.

- **Performance**

    - Asynchronous job processing prevents blocking user requests.

    - Heavy OCR tasks are handled in the background.

- **Fault Isolation**

    - Failure in document processing does not crash the entire system.

- **Extensibility**

    - New document types or exporters can be added without redesigning the system.

II. Application Components of the Project

1. Frontend Application

- User interface for upload and download

- Job status display

- API communication with backend

---

2. Backend API Service

- Authentication and user management

- Document upload handling

- Job creation and tracking

- Communication with queue and worker

---

3. Worker Service

- Background job processing

- OCR execution using AWS Textract

- Document parsing and data extraction

- Excel file generation

---

4. Storage Component (AWS S3)

- Stores uploaded documents

- Stores generated Excel files

---

5. Queue Component (AWS SQS)

- Manages asynchronous job processing

- Decouples backend from heavy processing tasks

---

6. OCR Service (AWS Textract)

- Extracts text from invoices and HR documents

---

7. Database

- Stores user details

- Stores job metadata and status

- Stores file paths and results

---

8. Job Management Component

- Tracks job lifecycle (pending, processing, completed)

- Links users to their documents

---

9. Export/Conversion Component

- Converts extracted text into structured Excel format

---

10. Configuration & Utility Components

- Environment configuration

- Common utilities and shared logic