PROJECT PROPOSAL, EXECUTIVE SUMMARY, PROBLEM DEFINITION

Fasiha Rohail - bsse23041

Abdul Samad Sohail - bsse23041

Information Technology University

Software development & construction

Zunnurain Hussain

1/4/2026

1. Executive Summary

This project presents an end-to-end deployment of an **In-Document AI Semantic Editor** on **Amazon Web Services (AWS)**. Unlike standard AI note-takers that restrict AI output to a side-chat panel, this application bridges the gap between AI generation and document editing. By integrating the **Groq API inference engine**, the system enables real-time, in-place text modification allowing users to rewrite, summarize, or expand content directly within their workspace. The architecture utilizes an **Application Load Balancer (ALB)** for secure traffic management and **PM2** for high availability on an **EC2 Ubuntu** environment.

2. Introduction & Domain Relevance

The domain of **Cloud-Native AI Orchestration** is rapidly evolving from passive transcription to active content co-creation. Traditional AI assistants often act as external research partners, requiring users to copy-paste text between a chat window and their document. This project targets the **Productivity and Collaboration** sector by automating the transition from AI output to document integration, reducing the cognitive load on users.

3. Problem Statement & Justification

Current AI note-taking and writing assistants suffer from a **workflow fragmentation** problem:

● **The "Chat-Silo" Limitation:** Most applications deliver AI insights as chat responses. Users are forced to manually transfer this text into their documents, which disrupts creative flow and increases the risk of formatting errors.

● **Infrastructure Latency:** For in-document editing to feel natural, AI inference must be nearly instantaneous. Standard cloud APIs often introduce 5–10 second delays, making in-place editing feel sluggish.

This project justifies the use of **AWS ALB** and **Groq LPU** to provide a secure, low-latency environment that enables truly integrated, in-document AI editing.

4. Aim & Measurable Objectives

The **Aim** of this project is to develop a load-balanced AWS infrastructure that supports an AI editor capable of **direct, in-place text modification**.

Measurable Objectives:

1. **Direct Editing Workflow:** Implement a frontend using **Vanilla JS** that can replace selected document text with AI-generated content without page refreshes.
2. **Low-Latency Infrastructure:** Achieve sub-second response times for text transformations using the **Groq SDK**.
3. **Secure Traffic Distribution:** Use an **AWS ALB** to route and protect the backend on Port 8081, ensuring no direct public access to the EC2 IP.
4. **Operational Resilience:** Maintain 24/7 service uptime using **PM2** for process management and monitoring.

5. Technical Stack Overview

●**Frontend:** HTML5/CSS3 and JS for a responsive, interactive editor, React

● **Backend:** Node.js and Express.js for handling API requests and JWT verification, Bcrypt,CORSE, Dynamo DB.

● **Cloud Hosting:** AWS EC2 and Application Load Balancer.

● **AI Engine:** Groq API , SDK

● **Process Management:** PM2 for monitoring and auto-restart capabilities.

6. Preliminary Architecture Design

The implementation follows a **Zero-Copy Workflow**:

1. **Request:** The user highlights text in the View (HTML) and selects a transformation.
2. **Processing:** The Controller (Express) sends the text to the Model (Groq SDK) with specific instructions for in-place replacement.
3. **Update:** The backend returns only the modified text, which the Frontend inserts directly into the document structure, bypassing the need for a separate chat window.