Slackbot Content Pipeline – Assignment Report

This document presents the detailed technical report for the **Slackbot Content Pipeline** project, developed as part of the AI Engineer assignment by **Ahtesham Shaikh**. The application is built in Python using Flask and simulates a content generation workflow that processes Slack-like event data and produces automated PDF reports. The project demonstrates backend API handling, text processing, and report automation.

System Architecture
The Slackbot Content Pipeline follows a modular architecture comprising: **Flask App (main.py)**: Handles incoming HTTP requests and routes them to appropriate modules. **Slack Bot Logic (slack_bot.py)**: Simulates Slack event triggers and interactions. **Content Generator (content_generator.py)**: Mocks AI-generated outlines and keyword analysis. **PDF Generator (pdf_generator.py)**: Uses ReportLab to create structured, branded PDF reports. **Test Script (test_request.py)**: Simulates API testing locally without Slack API dependency.

Component	Technology Used
Backend Framework	Flask (Python)
Bot Integration	Slack Bolt (mocked locally)
Data Processing	Python text processing & rule-based logic
Report Generation	ReportLab PDF Engine
Version Control	Git & GitHub
Deployment (optional)	Render (Docker-based, Python 3.10)

Workflow
1. The Flask server starts locally using python main.py.
2. The test script (test_request.py) sends a mock POST request with keyword or topic data.
3. The server processes the input, generates structured content (mocked AI output).
4. The output is formatted and converted into a PDF report stored in the **outputs/** directory.
5. Each PDF includes topic summary, trends, and recommendations.

Key Features End-to-end Python-based Slackbot simulation. Automated PDF report generation. Clear modular separation of logic and presentation. Reusable Flask route structure for integration with real Slack APIs. Compatible with both local and containerized execution.

Limitations & Future Enhancements
This mock Slackbot currently runs offline for demonstration purposes. Future versions may include: Real Slack API integration with slash commands. Cloud deployment with Supabase and Redis for persistent storage. LLM-based keyword clustering and semantic content generation. Email automation using SendGrid API.

Conclusion
The Slackbot Content Pipeline successfully demonstrates a scalable and modular approach to building a content generation backend. Although deployed locally, it reflects a deploy-ready structure suitable for real Slack workspace automation and report delivery.