1. Project Overview

Students will build a Cloud Compliance Automation Tool using cutting-edge cloud and AI technologies. The system will automate the lifecycle of compliance enforcement across multi-cloud environments (AWS, Azure, GCP) using Generative AI, Machine Learning, and Decision Modeling. The end goal is a working prototype deployable on public clouds and suitable for enterprise adoption.

2. Objectives

- Translate human-readable policies (e.g., ISO 27001) into deployable cloud configurations (e.g., Terraform).
- Generate validation test cases using past misconfiguration data.
- Score compliance using Multi-Criteria Decision Modeling (MCDM) techniques.
- Deploy and validate on AWS, Azure, and GCP with dashboards and reporting.

3. Functional Requirements

FR1: The system shall accept compliance policies in natural language.

FR2: The system shall translate these policies into Terraform or JSON configuration files.

FR3: The system shall generate test cases based on historical misconfigurations.

FR4: The system shall deploy and validate controls across AWS, Azure, and GCP.

FR5: The system shall compute a composite compliance score using MCDM (e.g., AHP or ELECTRE).

FR6: The system shall display results in a dashboard with filtering and export features.

4. Non-Functional Requirements

NFR1: The tool shall be cloud-agnostic and modular.

NFR2: The system shall support extensibility for new compliance frameworks.

NFR3: The latency for policy-to-template translation shall be under 5 seconds.

NFR4: The tool shall ensure data privacy and comply with academic ethical use standards.

NFR5: System performance should scale up to 100 concurrent policy validations.

5. Technology Stack

- LLM/GenAI: GPT-4 / LLaMA
- IaC: Terraform / CloudFormation
- ML Model: Decision Trees / Transformers
- Scoring Engine: AHP / ELECTRE
- Cloud Platforms: AWS, Azure, GCP
- Interface: Streamlit / ReactJS
- Database: Firebase / MongoDB / Postgres

6. Dataset Sources

- CloudMisconfig: Historical cloud misconfigurations
- CIS-CAT: Center for Internet Security benchmark results
- ComplianceForge: Policy templates
- AWS Config Logs: Cloud compliance logs

7. Milestones & Timeline

Week 1-2: Literature review, dataset acquisition

Week 3-5: Module 1: LLM-based policy translator

Week 6-8: Module 2: Misconfiguration-based test case generator

Week 9-10: Module 3: MCDM scoring engine

Week 11-12: System integration, dashboard design

Week 13: Final report, deployment, and code review

8. Team Deliverables

- Code Repository: Modular implementation with README.md files

- Architecture Diagram: Show inter-module and cloud-service interactions

- Validation Logs: At least 5 test cases per platform

- Presentation: Technical walkthrough and results

- Final Report: Academic format with sections for Introduction, Methodology, Results, and Discussion

9. Assessment Criteria

Functionality and Correctness: 30%

Innovation and Use of AI/ML: 20%

Cloud Platform Integration: 20%

Documentation and Reporting: 15%

Presentation and Demo: 15%