Compliance Questionnaire Assistant

Project Proposal

David O’ Connor, 08566496

Higher Diploma in Computer Science, SETU Waterford

Project Type: Work-based
Project Category: Web App

The Compliance Questionnaire Assistant (CQA) will be a web app that allows ServisBOT staff to upload security and compliance documentation as the input to a RAG process, and quickly answer questions with AI-generated answers.

When companies send ServisBOT security questionnaires as part of vendor onboarding, we spend significant time searching through our documentation. This process is effort intensive, task dependent, and takes up valuable time. The goal of the CQA is therefore to free up engineer time. This will be accomplished by allowing users to send queries to the CQA, in both single-question and CSV questionnaire mode. The CQA will then reply to the user with AI-generated answers, including references to source documentation.

This project will use RAG, which is a way of improving AI output by allowing LLMs to reference a user-managed knowledge base before generating responses. In this case, the knowledge base will consist of ServisBOT documentation which has been chunked, embedded, and stored in a vector database.

Proposed Technology List:
- AWS Lambda, S3, API Gateway, SQS, Aurora, CloudFront, DynamoDB, ECS
- Bedrock Knowledge Base
- JavaScript, TypeScript

Proposed Tools/Frameworks:
- React, Material UI, Tailwind
- AWS CDK in TypeScript
- Express, Joi, ESLint, Prettier, Mocha

The project process I intend to use involves first building an MVP. This will consist of a frontend, a backend API, and the ability to use these to query the Knowledge Base and receive answers. Subsequently, additional releases will deliver extra functionalities as they are developed. These functionalities will include CSV questionnaire upload, document management, authentication, and potentially a Slack integration.

The other stakeholders I am collaborating with are the ServisBOT SecOps team, who will be using this app. I will also be collaborating with the Director of Engineering and Lead Engineer on the ServisBOT team, who will be helping to guide this project.

The data used will be ServisBOT security and compliance documentation, in PDF, Markdown, and possibly other formats. No personal, private, or sensitive data will be used.

As this project will be built for ServisBOT, the finished code will be stored within the ServisBOT GitHub account. This will allow other ServisBOT engineers to further develop this application in the future. Upon discussion with ServisBOT, SETU personnel can likely be given access to the repo to see my commit history, etc.