Project Portfolio

# Lightspeed API Integration

## Description:

Developed an application to integrate with the Lightspeed Retail API using OAuth 2.0.

## Steps Completed:

* - Registered the application in the Lightspeed Developer Portal.
* - Generated a client\_id and client\_secret.
* - Used Postman to exchange credentials for an access\_token.
* - Performed authenticated API requests to retrieve data (e.g., customer and sales information).

## Tools:

Postman, REST APIs, JSON, OAuth 2.0

## Next Steps:

* - Create custom dashboards or reports from API data.
* - Build a full-stack web app to visualize and manipulate Lightspeed data.
* - Implement data filtering (e.g., retrieve sales > $50).

# Amazon Alexa Skill – Lightspeed Integration

## Description:

Created a custom Alexa Skill to interface with Lightspeed API using AWS Lambda and Python.

## Steps Completed:

* - Designed a custom Alexa Skill with invocation name, intents, and utterances.
* - Configured AWS Lambda to process Alexa requests and return responses in JSON.
* - Implemented logic in Python to query Lightspeed API and speak responses.

## Tools:

Amazon Alexa Developer Console, AWS Lambda, Python, OAuth

## Next Steps:

* - Add session memory to Alexa Skill for more dynamic interactions.
* - Implement error handling and natural language enhancements.
* - Secure API tokens using AWS Secrets Manager.

# Amazon Alexa Skill – ChatGPT Integration

## Description:

Built a basic Alexa Skill that interacts with OpenAI's ChatGPT via Lambda.

## Steps Completed:

* - Designed intents and utterances in Alexa Developer Console.
* - Configured AWS Lambda function to forward requests to OpenAI's API.
* - Returned ChatGPT responses in Alexa’s spoken format.

## Tools:

Alexa Developer Console, AWS Lambda, OpenAI API

## Next Steps:

* - Improve conversational flow with context tracking.
* - Add memory or personalization features using DynamoDB.
* - Enable dynamic topic selection or GPT-powered summaries.

# DNS Server – Pi-hole Setup

## Description:

Installed and configured Pi-hole as a local DNS server and ad blocker.

## Steps Completed:

* - Installed Pi-hole on an Ubuntu VM within a VMware lab.
* - Set the Pi-hole IP as the DNS resolver for a test VM.
* - Verified DNS query logging and ad blocking capabilities.

## Tools:

Pi-hole, Ubuntu, VMware Workstation

## Next Steps:

* - Integrate with Unbound or DNSCrypt for DNS security.
* - Configure conditional forwarding or split DNS.
* - Add a web-based management dashboard.

# Forward Proxy Server – Squid on pfSense

## Description:

Deployed Squid Proxy Server with SSL Bump capabilities on pfSense.

## Steps Completed:

* - Installed Squid package on pfSense firewall.
* - Created a local Certificate Authority (CA) in pfSense.
* - Configured a client VM to trust the fake CA.
* - Redirected HTTP/HTTPS traffic to go through Squid.
* - Enabled logging and inspected intercepted traffic.

## Tools:

pfSense, Squid Proxy, SSL Bump, Firefox

## Next Steps:

* - Implement URL filtering to block YouTube Shorts.
* - Integrate squidGuard for category-based filtering.
* - Analyze logs using LightSquid or custom scripts.

# Postman API Development

## Description:

Used Postman for testing and developing REST API requests.

## Steps Completed:

* - Set up authentication workflows using OAuth 2.0.
* - Sent GET/POST/PUT requests to Lightspeed API endpoints.
* - Parsed and filtered JSON responses.

## Tools:

Postman, REST APIs, JavaScript (Tests tab)

## Next Steps:

* - Automate workflows using Postman Collections and Monitors.
* - Write pre-request scripts and test assertions.
* - Export Postman collections to generate API documentation.