Project Vault

Alexander Lin – Project Vault

This document serves as a deep technical and philosophical breakdown of Alexander Lin’s major projects, creative experiments, and engineering approach. It captures not just what he's built, but how he thinks when he's building.

Featured Projects

📁 SharePoint Data Extractor (Internship Project)

Tech Stack: Python, Microsoft Graph API, Microsoft Azure, Custom GUI

A fully modular Python-based application designed to extract data from Microsoft SharePoint lists and export them to CSV format. Created during an internship to automate a manual, error-prone data entry system used by faculty and IT.

- Used Microsoft Graph API for secure data access
- Designed a GUI for non-technical users
- Modular architecture with clear separation between data retrieval, file writing, and UI
- Deployed internally to save dozens of hours per week in manual entry

> *"I was super proud of this shit. It was clean, fast, and modular as hell."*

🎧 AudioReader (iOS Audiobook App)

Tech Stack: Swift, iCloud API, Google AI NotebookLM (for voice generation)

A lightweight audiobook reader built in Swift, originally designed to let Alexander listen to AI-generated lectures based on his class notes. Supported multiple file types including .mp3, .wav, and .m4b.

- Custom UI with functional playback controls
- Leveraged personal AI tools (NotebookLM) to generate content
- Discontinued after discovering more robust apps in the App Store

> *"Honestly, I didn’t vibe with Swift. It was a solo grind and wasn’t super fun. Still glad I built it."*

🧠 The Oracle Chatbot (This Project)

Tech Stack: Oracle Cloud Infrastructure, Oracle APEX, Oracle GenAI, RAG, CLOB/VECTOR, REST APIs

A fully cloud-native chatbot built to answer questions about Alexander Lin’s background, skills, personality, and work. Ingests résumé data, personal documents, and personality profiles through Oracle GenAI with automated document ingestion and semantic embeddings.

- RAG-powered with OCI-native embeddings
- Designed and deployed within Oracle APEX
- Hooks into a WordPress site for public access

> *"I made a f*cking chatbot to talk about myself. How awesome is that?"*

Design Philosophy

Alexander builds tools like a tactician. He doesn’t jam everything into `Main()` and hope for the best. His style is modular, orchestration-based, and forward-compatible.

- **Modularity:** Each class or function serves a discrete purpose
- **Futureproofing:** Systems are designed to be expanded without rewrites
- **Readability > Cleverness:** Code should make sense later, even to someone else

> *"Adding new functionality should feel like slotting in a puzzle piece, not detonating a landmine."*

Debugging in Action

When the SharePoint Extractor started returning the wrong page, Alexander diagnosed the problem as architectural bloat.

Solution:

- Refactored the extraction logic into two separate classes:
  - `Extractor`: handles data retrieval
  - `Navigator`: handles SharePoint navigation and access

> *"Modularity saved my ass again. I just told myself: 'Alright, shitass. Break it into parts.' And it worked."*

In-Progress & Future Projects

- **AI-Enhanced Tooling in OCI:** Building more Oracle-based GenAI chatbots for:
  - Personal task automation
  - Development standards
  - Real estate + financial Q&A
- **Neural Networks (Coming Soon):** Studying models and inference architecture

---

Side Projects & Experimental Builds

👾 Digital Pirate Adventures (Text Game)

A sophomore-year passion project. A terminal-based, choose-your-own-adventure game blending swashbuckling pirate aesthetics with cybersecurity metaphors. Entirely written in pirate speak and packed with technical and narrative creativity.

- Navigate VPN sails through DDoS storms
- Arm your logic bomb swivels and exploit cannons
- Seek the Pirate Bay, with an optional side quest to the Library of Genesis
- Fully modular and object-oriented, coded before taking formal OOP coursework
- Featured class inheritance: `Galleon()`, `Frigate()`, and `Sloop()` subclasses of `Ship()`
  - `Sloop` had broadsides only
  - `Frigate` added swivel guns
  - `Galleon` had swivel guns and crew management systems
- Included alternate endings depending on side quests

> *"I built this entire game off raw instinct and freshman knowledge. Looking back, it was OOP-heavy and fully modular—and I didn't even know what 'inheritance' formally meant yet."*
>
> *"It never got released, but that game is still one of the most fun things I've ever built."*

Final Word

Alexander Lin doesn't just build software. He builds machines with purpose. Systems that breathe. Games that laugh. Tools that teach.

He believes in structure, clarity, and modularity—not just because it's clean, but because it keeps things alive.