****

**Internship Report – Kavtech**

**Name:** Ahmed Shafique

**Department:** AI Intern

**Internship start and end date:** 24- June to 01-August

**Acknowledgement:**

I would like to express my sincere gratitude to my mentors Hassan Bin Aslam and Specially Hamad Nasir for their constant guidance, support, and encouragement throughout my internship at Kavtech. Thanks to Sana Akram for mentoring me in script-based prompt engineering. The experience gained has greatly enriched my technical and professional skills.

**Declaration:**

I hereby declare that this internship report is the result of my own work and a true reflection of my learning and contributions at Kavtech during my internship from 24 June to 1 August. All the information presented in this report has been obtained through my personal experience under the guidance of my mentors.

**Executive Summary:**

This report summarizes the six-week internship at Kavtech where I worked as an AI Intern. My internship journey included work with the YLOPO AI Voice Team and the CAMMI Project Team. Throughout the internship, I explored a range of AI technologies, AWS services, and participated in real-world projects involving LLMs, Twilio, LiveKit, CrewAI, and prompt engineering.

**Table of Contents:**

1. Introduction
2. Objectives
3. Weekly Activities
4. Projects
5. Tools
6. Research and Development Work
7. Challenges and Solutions
8. Skills Acquired
9. Learning Outcomes and Reflections
10. Suggestions for Improvement
11. Conclusion

**Introduction:**

Kavtech Solutions is a forward-thinking technology company working on cutting-edge projects including MDB, IDX, YLOPO, and CAMMI. The company has a well-defined HR structure, training programs, and policies that create a professional growth-oriented environment. Projects range from real estate data platforms to generative AI systems for business ideation. The technical orientation included exposure to IDX’s ETL pipelines, metadata management, and API integration.

**Objectives:**

* Learn prompt engineering and LLM usage.
* Contribute to AI Voice Assistant development.
* Gain hands-on AWS cloud deployment knowledge.
* Research advanced image and text generation models.
* Apply CrewAI workflows for structured agentic prompts.

**Weekly Activities:**

* **Week1:**

Orientation with Kavtech’s organizational structure and understanding of ongoing projects including MDB, and YLOPO.

* Week2:

Worked with Sana Akram on designing script-based prompts for YLOPO's Digital Assistant on VAPI.

* Week3:

Developed STT-LLM-TTS agents using LiveKit and Twilio also using GROQ LLMs for lightweight prompt processing. Guided by Hassan Bin Aslam.

* Week4:

Moved to CAMMI project. Learned Python libraries like docx, regex, boto3. Worked on visual and text generation model’s R&D using Hugging Face models and other available models.

* Week5:

Studied and implemented AWS services: IAM, Lambda, API Gateway, Step Functions deployment framework.

* Week6:

Integrated CrewAI with LLaMA 3.1 8B to create agentic prompts generating history-rich business ideas. Worked on refining user prompts on frontend of CAMMI.

**Projects:**

* **Dynamic Intent, Script Based Prompt for Buyer/Seller Lead Flow:**

We're building a smart conversation prompt for a real estate chatbot. It needs to quickly understand if someone wants to buy a home, sell theirs, or get a quick cash offer and then guide the conversation the right way. If the person changes their mind or has doubts, the bot should handle that smoothly too. We'll use VAPI to make sure everything flows naturally during testing.

* **Multimodal AI Voice Agents:**  
  In this project, I set out to build something that feels futuristic but is very real today: a real-time AI voice agent that can talk to people naturally over the phone not just respond, but understand, think, and speak in a human-like way. This was the idea of my mentor Hassan Bin Aslam.

At the heart of the project is LiveKit, which helps the AI and the user talk to each other in real time. You can think of it like the communication link it keeps the voice flowing clearly and smoothly so the conversation feels natural, without delays or interruptions.

To make the agent truly conversational, I brought in three powerful tools:

* **Deepgram** handles the listening part turning spoken words into text instantly and accurately.
* **GROQ LLM** is the brain it processes what was said, thinks through the response, and generates a smart reply in real time.
* **Cartesia** gives the agent a voice turning the text back into natural-sounding speech that feels more like you're speaking to a person than a machine.

But I didn’t stop there. I wanted this agent to do things, not just sit there. So, I integrated Twilio, which gives the agent the ability to receive and make real phone calls. That means:

* Someone can call the AI agent and talk to it, and it’ll answer just like a real assistant.
* Or the AI agent can call someone on its own maybe to remind them of a meeting, follow up on a support ticket, or start a sales conversation.

Look at the code on GitHub: https://github.com/AhmedShafique313/LiveKit-Development.git

* **CAMMI – Chat Agent Marketing Manager Interface:**

**Overview:**

I worked on research and development for generative AI models that create both text and images. I learned how the CAMMI workflow is built on the backend, then built my own backend using Python and AWS Lambda (using only free‑tier resources). I also focused on fine‑tuning prompts and using the CrewAI agentic framework and Llama 3.1, to make the text output more historically accurate. On top of that, I improved prompts shown on the CAMMI user dashboard so behind‑the‑scenes adjustments would produce better front‑end results.

Later, I added a new feature: if a user gives a URL instead of typing a prompt, the system should automatically gather the important text from that web page, think about what’s useful, then generate a complete GTM (go‑to‑market) output from it. I used a tool stack that included hyperpower for scraping, DeepSeek for understanding what’s relevant, and Llama to structure the final output.

**Research & Prompt optimization:**

I created tables summarizing how different prompts performed when generating text or images.

Text Generation Models:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Description** | **Parameters** | **Input Modalities** | **Output Modalities** | **Context Length** | **Token Count** |
| Llama 3.1-8B | The Meta Llama 3.1 collection of multilingual large language models (LLMs) is a collection of pretrained, and instruction tuned generative models in 8B, 70B and 405B sizes (text in/text out) | 8B, 70B and 405B | Multilingual Text | Multilingual Text and code | 128K | 15T |
| deepseek-r1-distill-llama-70b | We introduce DeepSeek-R1, which incorporates cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1 across math, code, and reasoning tasks. | 671B | Multilingual Text | Multilingual Text and code | 128K | 15T |
| Queen3 | Qwen 3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models. The flagship model, Qwen3-235B-A22B, achieves competitive results in benchmark evaluations of coding, math, general capabilities, etc., when compared to other top-tier models such as DeepSeek-R1, o1, o3-mini, Grok-3, and Gemini-2.5-Pro. | 4B, 8B, 14B, 30B, 32B, 235B | Multilingual Text | Multilingual Text and code | 40K | 2T |

Image Generation Models:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **Providers** | **Logo Generation** | **Visual Representation** | **Pricing** |
| Flux | Black Forest Labs | A logo of a solar company  AI-generated content may be incorrect. | Fail | Free |
| Sea Art | Seart.ai |  | A diagram of a business | Paid |
| Imagen 4 | Google | A logo with a sun and green leaves  AI-generated content may be incorrect. | A powerpoint presentation with solar energy  AI-generated content may be incorrect. | Free + Paid |
| Napkin | Napkin.ai | Fail |  | Paid |
| ChatGPT | OpenAI |  |  | Paid |

From R&D I found the best model for both the text, logo and visual generation. Here is the GPT pricing table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **GPT Plans** | **Free** | **Plus** | **Pro** | **Team** | **Enterprise** |
| Messaging and Interaction | Unlimited | Unlimited | Unlimited | Unlimited | Unlimited |
| Chat History | Unlimited | Unlimited | Unlimited | Unlimited | Unlimited |
| GPT 4-O |  | 5 times | Unlimited | Flexible | Flexible |
| GPT 4.5 |  | ☑️ | ☑️ | ☑️ | ☑️ |
| OpenAI o4-mini | Limited | Standard | Unlimited | Unlimited | Unlimited |
| OpenAI o4 mini-high |  | Standard | Unlimited | Flexible | Flexible |
| Response time | Limited | Fast | Fast | Fast | Fast |
| Context Window | 8K | 32K | 128K | 32K | 128K |
| Memory | Limited | Expanded | Expanded | Expanded | Expanded |
| GPT Agents |  | Limited | Unlimited | Unlimited | Unlimited |
| File Upload | Limited | Limited | Unlimited | Unlimited | Unlimited |
| Image Generation | Limited | Unlimited | Unlimited | Unlimited | Unlimited |
| Sora |  | Limited | Expanded | Limited |  |
| Codex Agent Research Preview |  | ☑️ | Expanded | ☑️ |  |
| Admin Roles |  |  |  | ☑️ | ☑️ |
| Unified Billing |  |  |  | ☑️ | ☑️ |
| Bulk Member Management |  |  |  | ☑️ | ☑️ |

* + I explored the CAMMI architecture in detail and understood how the backend processes prompts, calls LLM engines, and handles output.
  + Then I built my own backend using Python and deployed it in AWS Lambda, keeping everything within the **AWS free tier** so I didn’t incur costs. When a file upload on S3 bucket the lambda function automatically triggered and perform and action to convert the .xml file into the .docx file.
  + The backend code handles incoming prompts, sends them to the AI, collects responses, and returns structured output for CAMMI.

Input URL feature in CAMMI:

I added a new feature that allows users to simply input a URL instead of writing a prompt manually. Once the URL is entered, the code uses a tool like Hyperpower (or similar services such as Crawl4AI or Bright Data) to scrape and fetch the content from the web page. After retrieving the data, DeepSeek is used to analyse and understand which parts of the page are the most important this includes identifying useful content like headings, product descriptions, feature lists, and key paragraphs. Once the relevant information is extracted, it is passed through a structured pipeline where a CrewAI agent system, powered by Llama, organizes the content into a clear and complete Go-To-Market (GTM) summary. The final document is well-formatted and easy to understand, giving users a clean and ready-to-use output with minimal input.

**Tools:**

* Python: Boto3, Regix, Requests, Json, Groq, Genai, CrewAI
* LLMs: Hugging Face Transformers & Diffusers, Llama, GROQ, Imagen
* AWS: Lambda, IAM, S3, Step Function & API Gateway
* Voice: Twilio, LiveKit, Deepgram, Cartesia
* Frameworks: LiveKit, AWS, CrewAI

**Research and Development Work:**

The CAMMI project involved in-depth exploration of text and image generation models. Evaluation metrics focused on creativity, consistency, and relevance. Tools like Hugging Face, CrewAI, and Llama 3.1 were integrated and used to optimize the generation. AWS was used for deploying microservices with IAM roles and step functions for managing task orchestration.

**Challenges and Solutions:**

* Managing the same agent as Inbound and Outbound calls. This is done by integration with Twilio SIP.
* Complexity in understanding stateful/stateless AWS services was solved through guided AWS documentation and SAM templates.
* Selecting suitable LLMs required trial experiments with Groq, OpenAI, and LLaMA for quality and latency evaluation.

**Skills Acquired:**

* Technical: Prompt Engineering, AI Integration, Python Scripting, LLMs Integrations with frontend
* Soft: Communication, reporting, agile collaboration, problem-solving
* Research: Evaluation and benchmarking of AI models also integration of backend AI models with frontend.

**Learning Outcomes and Reflections:**

This internship solidified my interest in AI development and deployment also LLM integration with frontend. I learned to work within real teams, follow agile practices, and debug complex cloud-hosted systems. It has helped me grow technically and professionally, preparing me for future roles in AI product development.

**Suggestions for Improvements:**

* While AWS is widely used, integrating GCP can provide added flexibility and overcome certain AWS limitations.
* While APIs are helpful, they have limitations. So, it's better to download free, open-source models (like from Hugging Face or Ollama), fine-tune them, apply RAG, and use prompt engineering to get accurate, customized results without any cost, and deploy them.

**Conclusion:**

My internship at Kavtech was a truly eye-opening experience. I got the chance to work on real AI systems, explore the latest models, and deploy scalable services using AWS. More than just technical skills, I gained a solid foundation in AI development and came to really value the power of teamwork and the guidance of great mentors.