



ONE MONTH SUMMER TRAINING REPORT

On

AI & Prompt Engineering

Submitted by

Mausham Kumar

Organization: VaultofCodes.in

Duration: 01 July 2025 – 01 August 2025

Under the Guidance of

[Anmoldeep Singh]

School of Computer Science & Engineering
Central University of Rajasthan, Ajmer

(July–August, 2025)

DECLARATION

I hereby declare that I have completed my Summer Internship training at **Vaultof-Codes.in** (AI and prompt engineering) from **01 July 2025** to **01 August 2025** under the guidance of my mentor. I declare that I have worked with full dedication during this internship and my learning outcomes fulfill the requirements of training for the award of the degree of **Bachelor of Technology (B.Tech.) in CSE**, Central University of Rajasthan, Ajmer.

Date: _____

(Signature of Student)

Name of Student: Mausham Kumar
Enrollment No.: 2022BTCSE015

ACKNOWLEDGEMENT

The success and final outcome of my Summer Internship on **AI & Prompt Engineering** required a lot of guidance and support. I am extremely privileged to have received this opportunity from **VaultofCodes.in**, and I am deeply grateful for the guidance provided throughout the internship. All that I have accomplished is due to such supervision and encouragement.

I sincerely thank **VaultofCodes.in** for providing me the opportunity to work on real-world projects and for giving me valuable exposure to AI and Prompt Engineering practices. I am especially thankful to my mentor(s) and the engineering team for their constant support, technical guidance, and feedback, which helped me complete the internship successfully.

I also express my gratitude to my university and faculty members for their encouragement, which has been instrumental in shaping my learning and project work.

Date: _____

(Signature of Student)

Name of Student: Mausham Kumar
Enrollment No.: 2022BTCSE015

ABSTRACT

The internship at **VaultofCodes** involved a comprehensive exploration of **Artificial Intelligence** and **Prompt Engineering** with a focus on developing a Streamlit-based diagnostic assistant for healthcare professionals in remote areas. Through practical exercises and the implementation of a real-world project, I gained hands-on experience with large language models (Google Gemini) and multi-agent orchestration using CrewAI. The work included building an intuitive user interface, designing modular AI agents for symptom analysis, vital sign assessment, lab report interpretation, and medical history evaluation, and integrating these agents to deliver an AI-driven differential diagnosis. In

addition to developing the core functionality, I implemented structured diagnostic summaries and a downloadable report feature to support clinicians in decision making. The internship also emphasized best practices in API integration, data handling, and ethical considerations when working with medical data. This experience provided valuable skills in Python, Streamlit, LLM integration, and prompt engineering, while deepening my understanding of how AI systems can assist healthcare delivery in underserved regions.

Overall, the internship in AI and Prompt Engineering served as a significant stepping stone in my professional growth, equipping me with practical skills and insights into building scalable, modular, and human-centered AI applications. The knowledge gained during this period will undoubtedly contribute to my future endeavors in the field of artificial intelligence and healthcare technology.

Contents

Declaration	1
Acknowledgement	2
0.1 Introduction	5
0.1.1 Motivation	5
0.1.2 Problem Definition	5
0.2 Specification of Company	6
0.3 Reports	7
0.3.1 Offer Letter	7
0.3.2 Completion Certificate	8
0.4 Reason for Choosing AI and Prompt Engineering	9
0.5 Learning Outcomes	10
0.6 Gantt Chart: Internship Timeline & Project Phases	11
0.7 Project Details & How It Works	12
0.7.1 Overview	12
0.7.2 Agents (Modular)	12
0.7.3 Sample Input & Output (CKD Example)	13
0.7.4 Disclaimer	14
0.8 Future Improvements	15
0.9 Bibliography / References	16

0.1 Introduction

0.1.1 Motivation

In recent years, Artificial Intelligence (AI) has become one of the most powerful tools in solving complex real-world problems. Among its many applications, healthcare is one of the most impactful domains where AI has the potential to save lives, reduce the burden on medical professionals, and improve access to timely treatment. However, healthcare delivery in many parts of the world still faces critical challenges. Rural and remote regions often struggle with a lack of experienced medical professionals, inadequate diagnostic facilities, and insufficient infrastructure. Patients in such regions usually depend on primary healthcare workers who may not always have the expertise to interpret complex clinical data such as lab reports, vital signs, or disease histories. This leads to delays in diagnosis and treatment, often worsening patient outcomes.

The motivation behind this project was to explore how AI, combined with effective prompt engineering, could assist in addressing these challenges. By developing a modular diagnostic assistant powered by large language models (LLMs), the goal was to provide a reliable, explainable, and easily accessible support system for healthcare workers. Unlike traditional diagnostic tools, an AI-driven assistant can analyze multiple aspects of patient information—symptoms, vitals, lab values, and medical history—and synthesize them into a structured differential diagnosis. This project also aimed to create a solution that is simple enough to be used by non-technical users, yet powerful enough to provide meaningful insights. The idea of using AI for healthcare in underserved regions is driven not only by the rapid advancements in LLMs but also by the pressing need to ensure that technology benefits communities that are often left behind in the digital revolution.

0.1.2 Problem Definition

The central problem addressed in this project is the absence of accessible, intelligent, and interpretable diagnostic tools for clinicians and healthcare workers in low-resource environments. Traditional diagnostic systems require advanced laboratory infrastructure, expert knowledge, and significant time. In contrast, many patients in rural areas cannot afford such delays, nor do healthcare workers always have access to specialists.

This project attempts to solve the following issues:

- **Fragmented Data:** Patient data often comes in the form of unstructured text, inconsistent lab reports, and subjective medical histories. Clinicians need a system that can standardize this information for easier interpretation.
- **Lack of Decision Support:** Healthcare professionals in remote areas frequently lack immediate support from specialists. An AI-based system can provide initial diagnostic suggestions to guide treatment or referrals.
- **Accessibility:** Existing AI tools are often designed for advanced hospital setups, requiring expensive infrastructure or internet connectivity. A lightweight, Streamlit-based web app is far more feasible for small clinics.
- **Explainability:** Many AI models provide results without justification, which limits trust. This project emphasizes clear, human-readable explanations of the diagnostic reasoning, ensuring transparency and building clinician confidence.

0.2 Specification of Company

Who We Are & What We Do

VaultofCodes is a rapidly growing **edtech startup** that focuses on empowering students to build real, job-ready skills through live training, hands-on projects, and certified internships. With more than four years of experience in the education technology sector, VaultofCodes has established itself as one of the fastest-growing coding communities in India.

About the Company

Over the years, VaultofCodes has earned the trust of more than **30,000 learners** nationwide. The organization is recognized as an official **Google for Education Partner**, is listed on **AICTE's National Internship Portal**, and is also a verified partner on Internshala. In 2024, VaultofCodes was awarded the title of **Best EdTech Startup**, demonstrating its rapid growth and strong presence in the technology education ecosystem.

At VaultofCodes, the mission is simple: *to simplify the journey from learning to earning*. This is achieved by creating programs that are practical, engaging, and aligned with real-world industry requirements. Through coding bootcamps, training modules, and internship opportunities, the company helps students not only acquire knowledge but also apply it in meaningful ways that prepare them for professional careers.

Core Offerings

The company provides a range of offerings tailored to meet the needs of learners at different stages of their education journey:

- **Training & Internship Programs:** Hands-on opportunities to apply theoretical knowledge to real-world projects.
- **Test Series:** Carefully curated practice tests to help learners prepare for competitive exams and technical assessments.
- **Study Material:** Updated and relevant resources to strengthen conceptual understanding.
- **Free Content:** Accessible educational resources that support self-paced learning.
- **Coding Practice:** Community-driven challenges and projects to encourage collaborative learning.

Contact Information

- **Phone:** 01169655581
- **Email:** vaultofcodes@gmail.com
- **Website:** <https://vaultofcodes.in>

0.3 Reports

0.3.1 Offer Letter

The following is the official **Offer Letter** provided by VaultofCodes for the internship program.


VaultofCodes

OFFER LETTER

Dear Mausham Kumar ,

Congratulations! We are delighted to offer you a position in our Internship Program as a AI & Prompt Engineering Intern at VaultofCodes. Your application and skills have demonstrated qualities we value, and we are confident that you will be a valuable asset to our team.

Program Overview:
This Internship Program is designed to provide hands-on experience through real-world projects, enhancing your practical knowledge in AI & Prompt Engineering. Throughout the program, you will take on a variety of tasks as assigned by the Manager-HR and uphold VaultofCodes' standards in all activities. This experience will allow you to further develop essential leadership and management skills while working towards meaningful project outcomes.

Terms of the Internship:
We believe in setting clear guidelines and expectations to maintain professionalism and accountability:

- **Attendance and Participation:** Active participation and consistent attendance are mandatory to qualify for the completion certificate and any recommendations. Unexcused absences may affect your evaluation.
- **Project Submissions:** Timely submission of all assigned projects is required to receive the final certificate. Those who do not submit projects or meet performance standards may receive a modified certificate, excluding certain details.
- **Professional Conduct:** Interns are expected to represent VaultofCodes with integrity and adhere to professional standards during interactions. Any breach of conduct will be addressed formally and may affect your standing in the program.
- **Feedback and Evaluation:** Your performance will be evaluated throughout the internship, and feedback will be provided to support your growth.

Internship Details:
Location: Online/Remote
Duration: 1 Month, starting from 01/07/2025
At the successful completion of the program, you will receive a certificate summarizing your achievements and a letter of recommendation based on your performance and contributions.

We look forward to welcoming you to VaultofCodes and to supporting you on this journey. Should you have any questions regarding these terms, please feel free to reach out.

Regards,


Founder, VaultofCodes.in



0.3.2 Completion Certificate

The following is the official **Completion Certificate** awarded by VaultofCodes after the successful completion of the internship program.



CERTIFICATE OF INTERNSHIP

THIS CERTIFICATE IS AWARDED TO

Mausham Kumar

For successful completion of internship as a AI & Prompt Engineering intern at
VaultofCodes.in for a duration of 1 month from 01/07/2025 to 01/08/2025

We found the candidate sincere, hardworking, technically sound and result oriented. We
take this opportunity to appreciate his/her work and contributions to thank and wish all
the best for your future endeavors

Founder



Google for Education
Partner

Manager

AICTE Corporate ID:
CORPORATE6511252d7c3271695622445

0.4 Reason for Choosing AI and Prompt Engineering

Artificial Intelligence and Prompt Engineering have emerged as transformative technologies that are reshaping industries, research, and education. The decision to pursue this internship project in AI and Prompt Engineering was influenced by both academic interest and the growing relevance of these fields in solving real-world problems.

Bridging the Gap in Healthcare

One of the main reasons for choosing this area was its potential to address pressing challenges in healthcare. Remote and underserved areas often face a shortage of doctors, diagnostic infrastructure, and timely access to treatment. AI systems, when applied responsibly, can support healthcare professionals by analyzing patient data such as symptoms, vitals, lab results, and medical history to provide preliminary insights. Prompt engineering ensures that the outputs from large language models (LLMs) are accurate, context-aware, and aligned with medical reasoning. Together, these tools can bridge gaps where human expertise is limited.

Relevance in Modern Technology

AI is no longer confined to academic research it has become an integral part of industry practices ranging from automation and natural language processing to robotics and healthcare. Similarly, prompt engineering has emerged as a specialized skill for effectively communicating with advanced LLMs like Google Gemini. Crafting precise prompts directly influences the quality, safety, and reliability of AI outputs. As these technologies advance, professionals equipped with knowledge in prompt design and AI system integration will be in high demand.

Skill Development and Career Growth

From a learning perspective, this project provided an opportunity to strengthen multiple technical skills:

- Building web-based AI applications using Python and Streamlit.
- Designing modular multi-agent systems with CrewAI for structured reasoning.
- Integrating Google Gemini through LiteLLM for high-quality inference.
- Practicing prompt engineering strategies to guide model outputs.

These skills are not only academically valuable but also align directly with industry requirements, making them crucial for future career growth.

Ethical and Practical Considerations

Another motivation for choosing AI and Prompt Engineering was the chance to explore ethical issues such as responsible deployment of AI in healthcare, data privacy, explainability, and user trust. Prompt engineering plays a critical role here by ensuring that models provide outputs that are safe, transparent, and easy for humans to interpret.

0.5 Learning Outcomes

The internship on **AI and Prompt Engineering** at VaultofCodes was a valuable experience that provided both technical and professional growth. Over the course of the training, I was able to transform theoretical knowledge into practical skills by working on a real-world project: the *AI Diagnostic Assistant for Remote Areas*. The following are the key learning outcomes achieved during this internship.

1. Technical Skills Acquired

- **Streamlit Development:** Gained hands-on experience in building interactive and user-friendly web applications using Streamlit, enabling rapid prototyping and deployment of AI-driven solutions.
- **CrewAI Orchestration:** Learned how to design modular, multi-agent workflows where each agent performs a specialized task such as analyzing symptoms, vitals, labs, or medical history.
- **Large Language Model Integration:** Successfully integrated Google Gemini API through LiteLLM, learning how to handle API responses, control model parameters, and maintain reliability in outputs.
- **Prompt Engineering:** Understood the importance of carefully designing prompts to ensure accuracy, context-awareness, and explainability in AI-generated responses.
- **Python Programming:** Improved proficiency in Python, covering libraries, modular coding practices, environment setup, and error handling.

2. Applied Project Experience

- Developed a diagnostic assistant capable of generating structured medical summaries and follow-up recommendations.
- Tested the system with real-world use cases such as Chronic Kidney Disease (CKD) detection from patient symptoms, vitals, and lab results.
- Implemented export functionality for downloading diagnostic summaries, improving usability for healthcare workers.
- Practiced iterative testing and debugging to ensure smooth multi-agent collaboration within the app.

3. Professional Growth

- Improved ability to work with project deadlines and milestone-based planning using a Gantt chart.
- Learned to document the system effectively through README files, code comments, and structured reports.
- Developed awareness of ethical considerations in AI, including data privacy, responsible deployment, and the importance of disclaimers in medical tools.

0.6 Gantt Chart: Internship Timeline & Project Phases

The internship at VaultofCodes was conducted from **01 July 2025 to 01 August 2025**. The entire duration was divided into weekly milestones to ensure structured progress on the project *AI Diagnostic Assistant for Remote Areas*. The following table presents the phases of the internship:

Week / Duration	Tasks and Project Phases
Week 1 (01 Jul – 07 Jul)	Orientation, understanding project requirements, literature review on AI in healthcare, setup of development environment, initial Streamlit app skeleton.
Week 2 (08 Jul – 14 Jul)	Development of agents for symptom analysis and vital signs assessment, basic prompt engineering experiments, early integration testing.
Week 3 (15 Jul – 21 Jul)	Implementation of agents for lab report interpretation and medical history analysis, CrewAI multi-agent orchestration, debugging workflows.
Week 4 (22 Jul – 28 Jul)	Integration of Google Gemini model through LiteLLM, refining prompt strategies, generating structured diagnostic summaries, UI improvements.
Final Days (29 Jul – 01 Aug)	Testing the app with real-world case examples (e.g., CKD), documentation of code and project, preparation of final report and presentation.

Visual Timeline: The following simplified Gantt-style timeline highlights the phases of work carried out during the internship:

Task / Phase	01–07 Jul	08–14 Jul	15–21 Jul	22–28 Jul	29 Jul–01 Aug
Requirement Gathering & Setup	_____				
Symptom & Vitals Agents		_____			
Labs & History Agents			_____		
Gemini Integration & UI				_____	
Testing & Documentation					_____

This timeline ensured that each phase of the project was completed in sequence, providing adequate time for development, testing, and documentation.

0.7 Project Details & How It Works

This section explains the design, architecture, and functionality of the project **AI Diagnostic Assistant for Remote Areas**. The system was developed as part of the internship at VaultofCodes and integrates multiple AI agents with Google Gemini to assist healthcare professionals by providing structured diagnostic support.

0.7.1 Overview

The project is built as a **Streamlit web application** that collects patient data (symptoms, vital signs, lab reports, and medical history) and processes it through a **multi-agent pipeline**. Each agent in the pipeline specializes in a specific medical reasoning task.

The overall workflow can be summarized as follows:

1. **Input Stage:** The user (clinician or healthcare worker) enters patient data into the web form.
2. **Preprocessing:** Data is structured and normalized to ensure consistency for further analysis.
3. **Multi-Agent Analysis:** Different agents process symptoms, vitals, labs, and history independently.
4. **Synthesis:** Outputs from all agents are combined by the *Diagnostic Synthesizer Agent* to generate a comprehensive differential diagnosis.
5. **Recommendations:** The system provides follow-up advice, recommended tests, or next steps.
6. **Output Stage:** A structured diagnostic summary is displayed on the interface and can be downloaded as a report.

The application ensures that every step of reasoning is modular, traceable, and human-readable.

0.7.2 Agents (Modular)

The project uses **CrewAI** to design modular agents that act like a virtual team of medical specialists. Each agent is powered by Google Gemini and has a clearly defined role:

- **Input Collector:** Structures raw patient input (symptoms, vitals, labs, history).
- **Preprocessor:** Cleans and normalizes the data for consistency.
- **Symptom Interpreter:** Analyzes reported symptoms and maps them to possible conditions.
- **Vital Sign Analyzer:** Evaluates abnormalities in blood pressure, pulse, temperature, etc.
- **Lab Result Analyzer:** Interprets key lab results (e.g., creatinine, hemoglobin, BUN).

- **History Analyzer:** Examines chronic conditions (e.g., diabetes, hypertension) that may affect diagnosis.
- **Diagnostic Synthesizer:** Integrates all findings to provide a differential diagnosis.
- **Recommendation Agent:** Suggests further tests, treatments, or referrals.
- **Explanation Agent:** Translates the medical reasoning into clear, human-readable language.

This modular approach improves transparency and makes the system adaptable to new agents (e.g., for radiology images or genetic data) in the future.

0.7.3 Sample Input & Output (CKD Example)

To demonstrate the functionality of the system, a sample case was tested for **Chronic Kidney Disease (CKD)**.

Input Data:

- Symptoms: swelling in legs, fatigue, reduced urine output, nausea.
- Body Temperature: $98.7^{\circ}F$
- Blood Pressure: 145/95 mmHg
- Pulse Rate: 88 bpm
- Lab Results: Creatinine 3.5, BUN 52, Hb 9.8
- Medical History: diabetes, hypertension

Expected Output (Generated Summary):

- **Probable Diagnosis:** Chronic Kidney Disease (Stage 3–4), Hypertensive Nephropathy.
- **Abnormal Findings:** High creatinine, high BUN, low hemoglobin, elevated BP.
- **Recommendations:** Further renal function tests, strict blood pressure management, referral to a nephrologist.
- **Disclaimer:** Diagnosis is AI-assisted; clinical confirmation required.

This example validates the system’s ability to combine multiple inputs and provide structured outputs that align with real medical reasoning.

0.7.4 Disclaimer

It is important to note that the **AI Diagnostic Assistant** is designed as a decision-support tool and not as a replacement for professional medical judgment. The outputs generated are based on patterns recognized by the model and should always be interpreted by a qualified healthcare provider.

The disclaimer emphasizes:

- The tool provides **probable diagnoses**, not final conclusions.
- Clinical correlation and additional medical tests are always necessary.
- Ethical use of AI in healthcare requires strict adherence to patient privacy and data protection.
- VaultofCodes and the developer assume no responsibility for misuse of this prototype in real-world clinical settings.

This ensures the project follows responsible AI practices while highlighting its potential to support healthcare professionals, especially in underserved and remote regions.

0.8 Future Improvements

While the **AI Diagnostic Assistant for Remote Areas** demonstrates the successful application of AI and prompt engineering in healthcare, there remain multiple opportunities for enhancement and expansion. These improvements will not only strengthen the system’s technical capabilities but also make it more practical and impactful in real-world clinical environments. The following points summarize the key future improvements:

- **Support for Detailed Lab Reports and Medical Images:** Currently, the system processes only basic text-based lab values. In the future, functionality can be extended to allow clinicians to upload detailed lab reports (PDF/CSV formats) as well as diagnostic images (X-rays, MRI scans). This will enable multimodal reasoning and provide more comprehensive diagnostic insights.
- **Visualization of Diagnosis Trends and Graphs:** To improve interpretability, the system could generate graphical visualizations of vital signs, lab values, and disease progression trends. Graphs and charts will help healthcare providers identify changes over time and support evidence-based decision-making.
- **User Authentication and Data Security:** Since medical information is highly sensitive, future versions of the system should include secure login features, role-based access control, and encryption methods. This will ensure compliance with healthcare data protection standards such as HIPAA.
- **Offline Capability for Remote Areas:** Given that many rural and underserved regions lack stable internet connectivity, the system can be enhanced to support offline functionality. Lightweight local models or caching mechanisms could allow the assistant to run even without continuous access to cloud-based APIs.
- **Multilingual Support:** To make the diagnostic assistant accessible to a wider audience, especially in linguistically diverse regions, multilingual capabilities can be integrated. This includes support for regional languages, speech-to-text input, and text-to-speech output for better communication with non-English-speaking patients.
- **Mobile-Friendly Progressive Web App (PWA):** Converting the Streamlit application into a mobile-friendly PWA would make it accessible on smartphones and tablets without requiring installation. This will improve usability for healthcare workers in the field and support portable deployment.

In conclusion, these improvements will move the project closer to becoming a robust, scalable, and globally usable healthcare tool. With the integration of advanced features such as image analysis, data security, multilingualism, and offline functionality, the diagnostic assistant has the potential to make a meaningful impact in both rural and urban healthcare ecosystems.

0.9 Bibliography / References

The following references, documentation, and resources were consulted during the internship and project development:

- **GitHub Repository:** <https://github.com/maushamkumarray1/ai-diagnostic-assistant>
- **CrewAI Documentation:** Official documentation and guides for building and orchestrating multi-agent AI workflows.
- **LiteLLM GitHub:** Lightweight interface for integrating LLMs such as Google Gemini. <https://github.com/BerriAI/litellm>
- **Google Gemini API / AI Studio:** Google's AI platform used for model integration, inference, and experimentation. <https://aistudio.google.com/>
- **Python Official Documentation:** The official Python documentation used as a reference for programming best practices. <https://docs.python.org/3/>
- **Streamlit Documentation:** Documentation for rapid UI development with Streamlit. <https://docs.streamlit.io/>

These references provided essential knowledge for understanding the concepts, implementing the application, and ensuring best practices during the internship.