

INDIAN INSTITUTE OF TECHNOLOGY JAMMU

AI & LLM SUMMER SCHOOL 2026

INTERNSHIP REPORT

Applied Generative AI, LLM Agents and Workflow Automation

Internship Training, Projects and Technical Outcomes

SUBMITTED BY

Krishna Tayal

Programme: AI (Generative AI and Large Language Models)

Cohort: SS26-C1 | College Offline

SUBMISSION PROFILE

Academic report and evidence portfolio

Submission summary

This report consolidates the training, implementation, testing, documentation and professional evidence completed during AI & LLM Summer School 2026. It is organized for direct academic review, with technical artifacts, screenshots, repository links, project posts, source excerpts and credential verification grouped into dedicated appendices.

Submission field	Details
Student	Krishna Tayal
Programme	AI & LLM Summer School 2026
Cohort	SS26-C1 College Offline
Institution	Indian Institute of Technology Jammu
Technical focus	Generative AI, LLM agents, workflow automation and applied AI

PORTFOLIO EVIDENCE AT A GLANCE

Evidence area	Included record
Final capstones	2 complete systems: n8n customer support automation and an agentic customer success manager
Applied portfolio	15 documented systems across automation, agents, interfaces, multimodal pipelines and product research
Technical learning	8 executed Python day notebooks, 5 digital assignments and multimodal AI pipeline evidence
Professional publication	13 project-specific LinkedIn posts supported by repositories, posters, screenshots and demonstrations
Verified credentials	14 course credentials: 4 n8n, 2 Hugging Face, 1 LangChain and 7 Anthropic / Claude certificates

Credential completeness - 14 of 14 verified files included

Appendix F contains a complete visual index, fourteen individual full-size certificate pages and credential verification or evidence access for academic review.

STUDENT DECLARATION

I, Krishna Tayal, declare that this internship report, entitled “Applied Generative AI, LLM Agents and Workflow Automation,” is an original record of the training, implementation, testing and documentation work carried out by me during AI & LLM Summer School 2026, conducted by the Indian Institute of Technology Jammu.

The report consolidates the technical work completed during the programme, including Python notebooks, AI pipelines, automation workflows, intelligent agents, conversational interfaces, integrations, capstone projects, demonstrations and professional project publications.

All external libraries, platforms and technical references used during implementation have been acknowledged in the appropriate sections. The report has been prepared for academic review and certification, and the published project packages have been checked to exclude confidential API keys, access tokens and private credentials.

Name	Krishna Tayal
Programme	AI & LLM Summer School 2026
Cohort	SS26-C1 AI (GenAI and LLMs) College Offline

STUDENT SIGNATURE

A handwritten signature in blue ink that reads "Krishna Tayal". The signature is written in a cursive style and is positioned above a horizontal line.

Krishna Tayal

Declared and signed for academic submission

ACKNOWLEDGEMENT

I express my sincere gratitude to the faculty coordinators, mentors and organizing team of AI & LLM Summer School 2026 at the Indian Institute of Technology Jammu for designing a practice-oriented programme in generative AI, large language models, automation and intelligent agents.

The structured learning sessions, assignments, certifications and capstone challenges provided an opportunity to move from foundational Python and AI concepts to complete automation systems. The programme encouraged experimentation with n8n, Hugging Face, LangChain, CrewAI, Gradio, Slack, Telegram, Vapi, ElevenLabs, Claude and modern agent frameworks.

I am thankful to my peers and the wider learning community for technical discussions, feedback and shared problem-solving during the programme. I also acknowledge the open-source maintainers and documentation teams whose libraries and tools made it possible to design, test and document the projects described in this report.

Finally, I thank my family and teachers for their encouragement, patience and support throughout the internship and submission period.

ABSTRACT / EXECUTIVE SUMMARY

This internship report presents the work completed during AI & LLM Summer School 2026 in the area of generative AI, large language models, workflow automation and intelligent agents. The central technical theme was the design of an AI-powered customer support and customer success automation suite. The solution combines event-driven n8n workflows with a multi-agent decision layer, local and cloud model integrations, secure human approval controls, structured data storage and professional reporting.

The first major project, AI Customer Support Automation Platform, uses five independent n8n workflows with twenty-five nodes to perform ticket intake, duplicate prevention, AI classification, priority assignment, skill-based routing, escalation, customer communication and SLA analytics. The second major project, AI Customer Success Manager, uses six specialist agents and eight tools to evaluate SaaS account health, identify churn risk, recommend interventions, retain session memory and request human approval before side-effecting actions.

The internship also produced twelve supporting applied projects spanning Python notebooks, Hugging Face multimodal pipelines, Telegram and Slack interfaces, Vapi and ElevenLabs integrations, reusable n8n templates, Clay AI enrichment, CrewAI collaboration, civic technology, student-success product design and automated professional publishing.

The work demonstrates how AI systems can be made useful in operational settings by combining model reasoning with deterministic scoring, schema validation, audit trails, human oversight and reproducible offline demonstrations. The final outcome is a documented portfolio of deployable workflows, tested agent components, public repositories, recorded demonstrations and professional project evidence.

Keywords

Generative AI, large language models, n8n, OpenAI Agents SDK, Gemini, Hugging Face, workflow automation, customer support, customer success, human-in-the-loop, Python, GitHub.

TABLE OF CONTENTS

Section	Page
Student Declaration	3
Acknowledgement	4
Abstract / Executive Summary	5
Chapter 1. Introduction	7
Chapter 2. Internship Overview	9
Chapter 3. Training and Learning Phase	11
Chapter 4. Project Work	13
Chapter 5. Weekly Work Log	29
Chapter 6. Challenges Faced	30
Chapter 7. Skills Developed	31
Chapter 8. Learning Outcomes	32
Chapter 9. Self Assessment	33
Chapter 10. Conclusion and Future Scope	34
Appendix A. Evidence and Repository Links	35
Appendix B. LinkedIn Project Posts	36
Appendix C. Project Screenshots	37
Appendix D. Selected Source Code	52
Appendix E. Deliverables and References	53
Appendix F. Certificates and Credentials	54

CHAPTER 1

Introduction

Chapter overview

This chapter introduces the programme context, internship purpose, objectives, scope, duration and structure of the report.

1.1 Background

Generative AI and large language models have moved from experimental chat interfaces into workflow automation, decision support, knowledge retrieval and multi-agent applications. Organizations increasingly require systems that can combine AI reasoning with business rules, databases, APIs, auditability and human oversight. Customer support and customer success are suitable domains because they involve high volumes of text, repeated decisions, diverse stakeholder needs and measurable service outcomes.

AI & LLM Summer School 2026 provided a structured environment to learn these technologies through notebooks, platform assignments, integrations, bots, certifications and final capstone projects. The programme emphasized building working artifacts, documenting them clearly and presenting outcomes through GitHub, Google Drive, demonstrations and professional communication.

1.2 Purpose of the Internship

The purpose of the internship was to develop practical competence in AI application engineering and automation. The work connected foundational programming with low-code orchestration, model APIs, multimodal pipelines, intelligent agents, interface development, deployment and technical documentation.

1.3 Internship Objectives

- Strengthen Python programming, notebook execution and reproducible experimentation.
- Understand generative AI, LLM prompting, transformers, structured outputs and model limitations.
- Design event-driven business automations using n8n workflows, webhooks, schedules and integrations.
- Build a multi-agent application with specialist roles, tools, handoffs, memory and human approval.
- Develop secure bots and interfaces for Telegram, Slack and Gradio.
- Apply version control, testing, documentation, presentation and professional publishing practices.

1.4 Scope of Work

The internship scope covered training assignments and two integrated capstone systems. Supporting work included Python notebooks, Hugging Face pipelines, n8n templates and integrations, voice-agent configuration, research enrichment, bot interfaces, AI course certificates, project documentation, demonstrations, GitHub publication and LinkedIn project posts.

1.5 Internship Duration

The core internship followed a six-week structure during Summer 2026, followed by a final packaging, publication and documentation phase. The learning plan progressed from orientation and tool training to requirement analysis, development, testing, deployment and reporting.

1.6 Organization of the Report

Chapters 2 and 3 describe the internship plan and training phase. Chapter 4 presents the system design, implementation and results. Chapter 5 records the weekly work log. Chapters 6 to 9 discuss challenges, skills, learning outcomes and self-assessment. Chapter 10 concludes the report and identifies future improvements. The appendices provide repository links, LinkedIn posts, screenshots, selected code and reference material.

CHAPTER 2

Internship Overview

Chapter overview

This chapter summarizes the internship details, responsibilities, expected deliverables and six-week work plan.

2.1 Internship Details

Field	Details
Programme	AI & LLM Summer School 2026
Host institution	Indian Institute of Technology Jammu
Student	Krishna Tayal
Mode	College - Offline
Core duration	Six-week structured internship, Summer 2026
Primary domain	Generative AI, LLMs, workflow automation and intelligent agents
Report theme	Applied Generative AI, LLM Agents and Workflow Automation

2.2 Internship Objectives

The operational objective was to produce working, reviewable AI systems rather than isolated demonstrations. Each project was expected to have a clear problem statement, architecture, implementation, validation evidence, deployment path and submission package.

2.3 Roles and Responsibilities

- Analyze requirements and translate them into workflow and agent components.
- Develop Python notebooks, automation workflows, agent tools and interface code.
- Configure model, voice, messaging and Google Workspace integrations securely.
- Create diagrams, documentation, presentations and recorded demonstrations.
- Validate outputs, test error paths and maintain human control over sensitive actions.
- Publish versioned repositories and communicate projects through LinkedIn posts.

2.4 Expected Deliverables

Deliverable group	Expected output
Training artifacts	Python notebooks, certifications, tool assignments and course evidence
Automation portfolio	n8n integrations, bots and ten reusable workflow templates
AI pipeline portfolio	Text, image, audio and video pipeline demonstrations

Deliverable group	Expected output
Capstone 2	AI Customer Support Automation Platform using n8n
Capstone 3	AI Customer Success Manager using the Agents SDK
Professional evidence	GitHub repositories, Drive packages, presentations, videos and LinkedIn posts

2.5 Weekly Timeline

Week	Phase	Activities	Approx. hours
Week 1	Orientation and training	Programme orientation, Python/Colab, AI tools, certificates and setup	30
Week 2	Requirement analysis	Stakeholders, problem statements, workflow mapping and project architecture	30
Week 3	Development	n8n workflows, Telegram bot, Vapi, ElevenLabs and Hugging Face pipelines	32
Week 4	Advanced development	Agents SDK system, health scoring, tools, memory and approval controls	34
Week 5	Testing and deployment	Validation, Docker/local deployment, GitHub publication and demos	30
Week 6	Documentation	Reports, presentations, screenshots, LinkedIn posts and final packaging	24

Total engagement

The six-week plan represents approximately 180 hours of guided learning, independent implementation, testing, documentation and project communication.

CHAPTER 3

Training and Learning Phase

Chapter overview

This chapter records the orientation, tools, platforms, technical concepts and documentation practices studied before and during project implementation.

3.1 Orientation Program

The orientation phase introduced the internship structure, submission expectations, collaborative learning process and safe use of AI tools. Early assignments emphasized exploring multiple AI platforms, comparing their capabilities, recording outputs and developing confidence with prompt-driven workflows.

3.2 Tools Learned

Category	Tools / platforms
Programming and notebooks	Python, Google Colab, Jupyter notebooks
Automation	n8n, webhooks, schedules, Data Tables and workflow templates
Models and pipelines	Hugging Face Transformers, Gemini, Ollama, structured outputs
Agents	OpenAI Agents SDK, LangChain concepts, CrewAI setup
Interfaces and bots	Gradio, Slack Bolt, Telegram, aiogram
Voice and enrichment	Vapi, ElevenLabs, Clay AI
Engineering	Git, GitHub, Docker Compose, PostgreSQL, SQLite
Documentation	Google Drive, PowerPoint, diagrams, recorded demos, LinkedIn

3.3 Software and Platforms Used

Development took place across Google Colab, local Python environments, n8n, GitHub and Google Drive. Docker Compose was used for the self-contained n8n demonstration environment. SQLite supported agent memory, PostgreSQL supported structured support-platform persistence, and JSON/CSV files supported reproducible offline evidence.

3.4 Technical Concepts Studied

- Prompt design, LLM response control and schema-constrained JSON outputs.
- Transformer pipelines for text, image, audio and video tasks.
- Event-driven workflow design using triggers, branches, retries and scheduled jobs.

- Multi-agent orchestration using manager-as-tools and decentralized handoffs.
- Deterministic health scoring combined with language-model explanation and planning.
- Human-in-the-loop controls, session memory, audit logging and least-privilege secrets.
- Software packaging, tests, reproducible offline demos and submission integrity.

3.5 Documentation Reviewed

Implementation decisions were supported by platform documentation for n8n workflows and integrations, Hugging Face pipelines, the OpenAI Agents SDK, Gemini model access, Slack Bolt, Telegram aiogram, Docker, PostgreSQL and GitHub. Project-specific READMEs, architecture notes, validation reports and setup guides were maintained alongside the source code.

The screenshot shows a Jupyter Notebook interface with a dark theme. At the top right, it says 'SOURCE EVIDENCE'. The notebook title is 'Hugging Face Multimodal Pipelines' with a subtitle 'Executed notebook evidence'. Below this, there's a section 'PROJECT SOURCE' containing Python code for text processing using NLTK. The code includes functions for lowercasing, punctuation removal, and tokenization. Below the code, the output of the execution is shown, displaying the processed text and a message about downloading NLTK data.

```
#1. Text Lowercasing
text1 = "Hello, World world"
print(text1.lower())
text2 = "NLP Is FuN"
print(text2.lower())
text1 = "hello!!! How are you?"
print(text1.translate(str.maketrans('', '', string.punctuation)))
text2 = "NLP Is FuN"
print(text2.translate(str.maketrans('', '', string.punctuation)))
text1 = "NLP 101 is easy"
print(re.sub(r'\d+', '', text1))
text2 = "I scored 95 marks"
print(re.sub(r'\d+', '', text2))
text1 = "NLP is amazing"
print(word_tokenize(text1))
text2 = "I love AI. Added more content"
print(word_tokenize(text2))
text1 = "NLP is amazing"
```

```
hello, world world
nlp is fun
hello How are you
NLP Is FuN
NLP is easy
I scored marks
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
```

Krishna Tayal · SS26 AI/LLM Summer School

Figure 3.1 - Executed Hugging Face pipeline evidence used during the training phase.

CHAPTER 4

Project Work

Chapter overview

This chapter presents the problem statement, objectives, architecture, workflow, technology stack, implementation modules, algorithms, integrations, deployment and final results of the internship project.

4.1 Project Introduction

Problem Statement

Customer support and customer success teams often work with disconnected channels, inconsistent priority decisions and fragmented customer data. Manual triage slows acknowledgement, duplicate tickets consume time, critical cases can be overlooked, and account health reviews depend on subjective judgment. Fully autonomous AI is also inappropriate for high-impact actions such as escalation or scheduling without human approval.

Objectives

- Centralize request intake and prevent duplicate support records.
- Classify tickets by category, priority, sentiment and summary using structured AI output.
- Route tickets and customer-success work to the most relevant specialist.
- Calculate a transparent account health score and recommend interventions.
- Preserve human approval for sensitive actions and maintain an audit trail.
- Provide reproducible demonstrations, documentation and public project evidence.

Scope

The implemented scope includes website/webhook intake, AI triage, skill-based routing, human escalation, customer updates, SLA analytics, customer-health assessment, product and support analysis, success planning, approved scheduling/escalation, session memory and audit logging. Supporting modules include Telegram and Slack interfaces, multimodal pipelines, voice-agent integrations and research-enrichment evidence.

4.2 System Design

Overall Architecture

The solution follows a layered architecture. n8n receives and validates events, the AI triage and Agents SDK layers perform classification and specialist reasoning, tools read or update operational data, human approval protects sensitive actions, and the persistence layer records accounts, tickets, metrics and audit events.

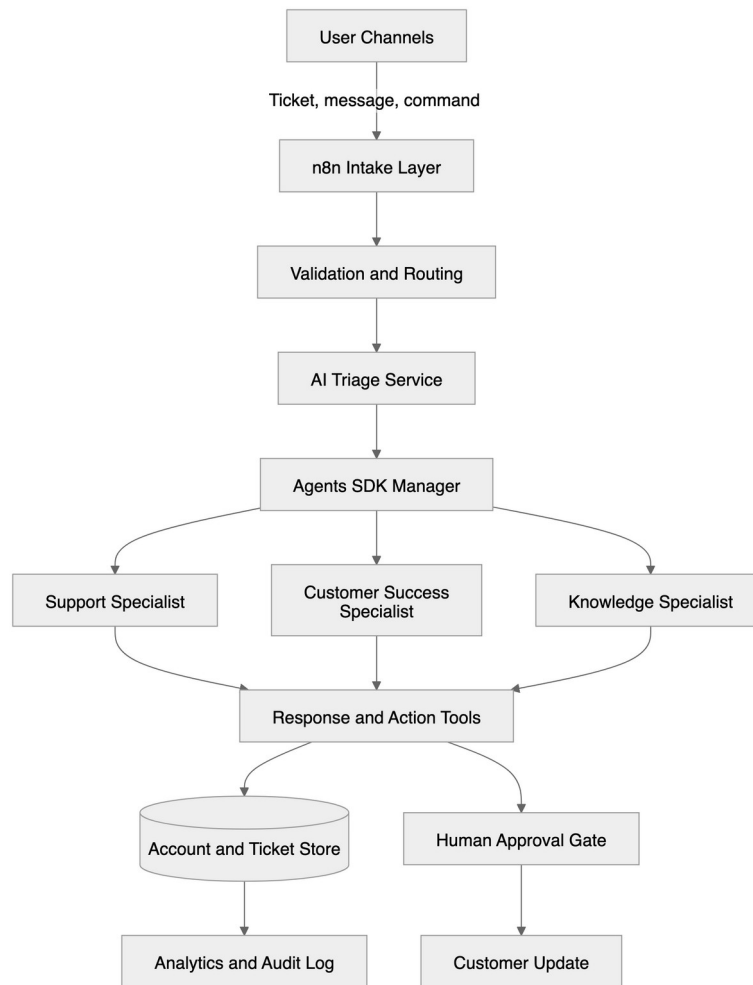


Figure 4.1 - Overall architecture of the AI support and success automation suite.

Workflow Diagram

The end-to-end workflow begins with request validation and branches according to support, success-risk or knowledge needs. A specialist agent prepares an action, approval is requested when required, and the final outcome is communicated and recorded.

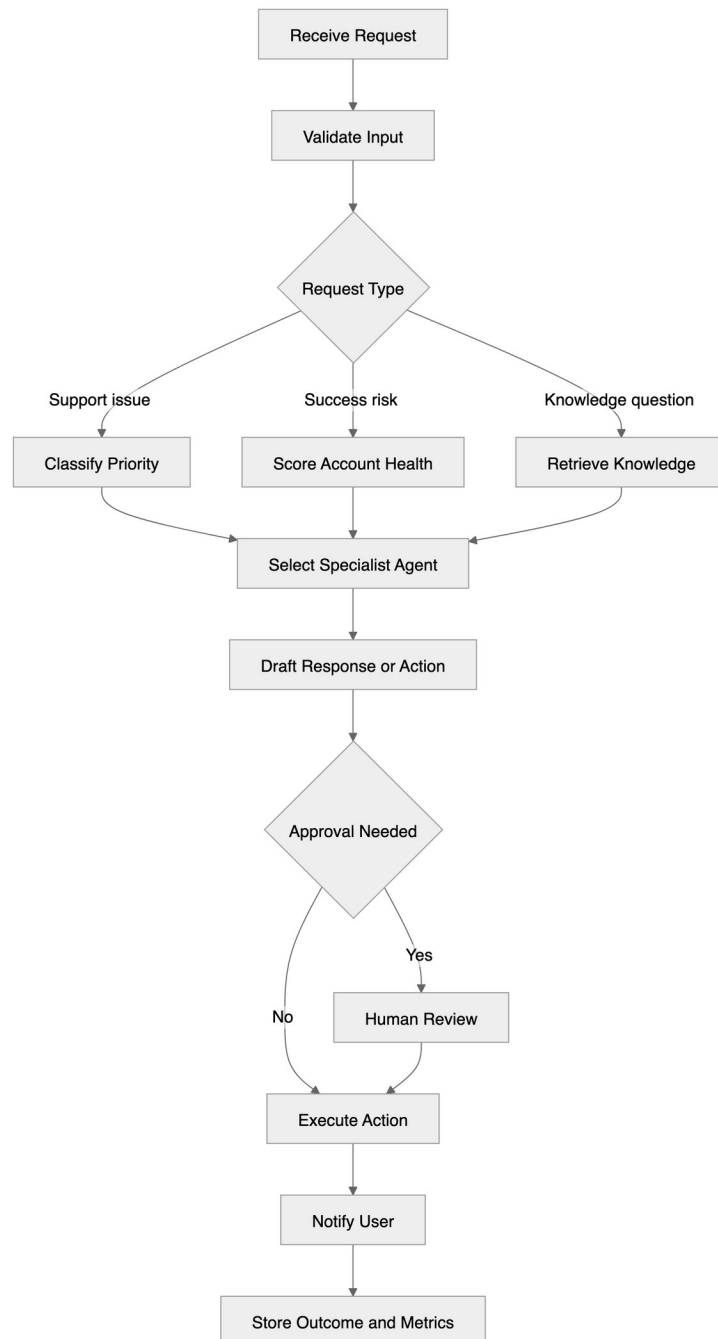


Figure 4.2 - End-to-end request triage, approval and execution workflow.

Use Case Diagram

The principal actors are customers, support executives, customer-success managers and administrators. The platform provides each actor with a controlled set of interactions while keeping workflow management and auditing separated from customer-facing operations.

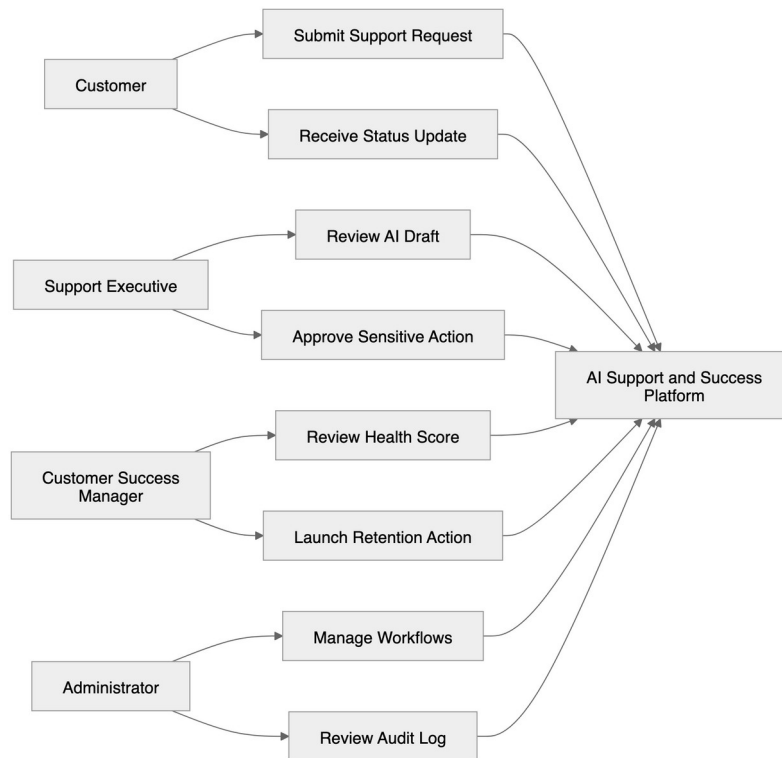


Figure 4.3 - Use-case map for the main actors and system functions.

Database and Entity Design

The logical data model connects customer accounts with interactions, tickets and health scores. Agent actions can require approvals, health scores can trigger success plays, and every operational side effect is represented by an audit event. The n8n demonstration uses PostgreSQL for support records, while the agent project uses SQLite for session memory and JSONL for the append-only side-effect log.

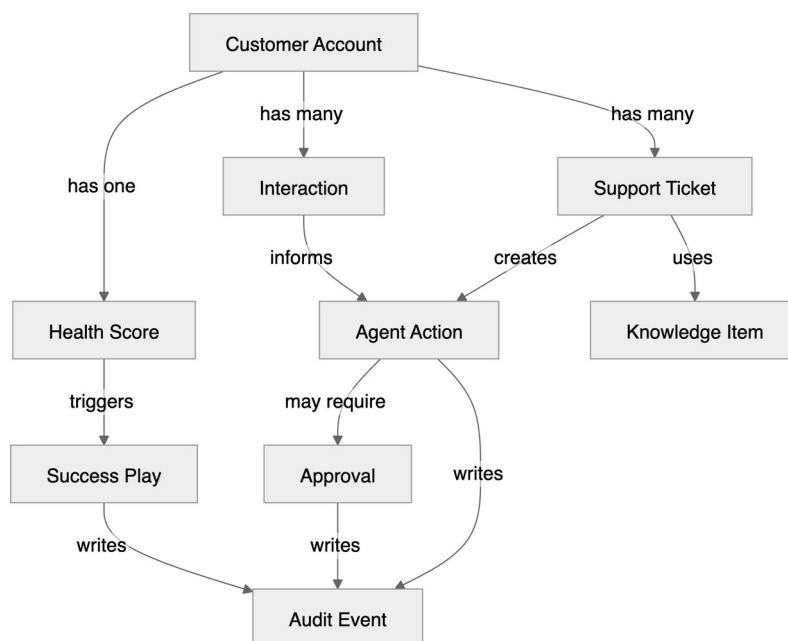


Figure 4.4 - Logical entity and relationship model.

Component / Class Design

The Agents SDK project is divided into data models, scoring logic, tools, agent definitions, data access, session handling, validation and command-line interfaces. Pydantic models define `CustomerAccount`, `HealthAssessment`, `InterventionPlan` and `AccountHealthReport`. This separation keeps deterministic business logic independent from model reasoning and presentation code.

4.3 Technology Stack

Layer	Technology
Programming languages	Python, JavaScript expressions, SQL, JSON
Automation framework	n8n with webhooks, schedules, branching, retries and Data Tables
Agent framework	OpenAI Agents SDK with LiteLLM model adapter
Models	Gemini for agent runs; local Ollama qwen2.5:3b for structured triage
Libraries	Pydantic, pytest, aiogram, Gradio, Slack Bolt, Transformers
Databases	PostgreSQL, SQLite, JSON/CSV and n8n Data Tables
Cloud / integrations	Google Drive, Google Sheets, Google Calendar, Vapi, ElevenLabs

Layer	Technology
Version control	Git and GitHub
Deployment tools	Docker Compose, local n8n runtime and command-line validation

4.4 System Implementation

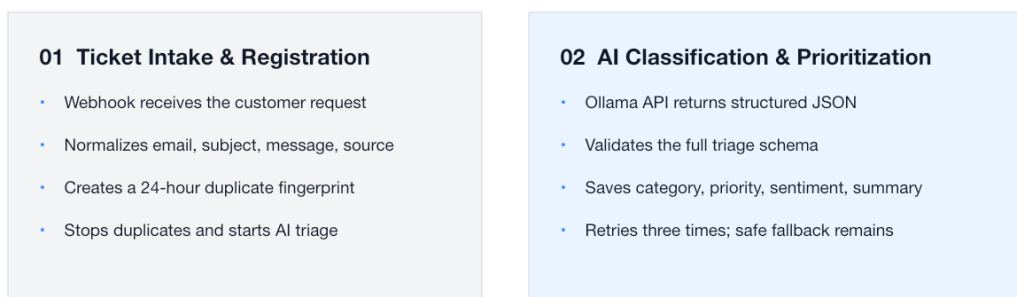
Module 1 - n8n Customer Support Automation

The n8n capstone is implemented as five independent workflows with exactly twenty-five nodes. Ticket Intake and Registration receives a support request, normalizes input and creates a twenty-four-hour duplicate fingerprint. AI Classification and Prioritization requests structured triage from a local model, validates the schema and saves category, priority, sentiment and summary fields.

Agent Assignment and Escalation selects an available agent using skill and workload signals, branches on risk and creates a manager-review event when needed. Customer Communication and Feedback updates ticket state and requests feedback after resolution. SLA Monitoring and Analytics checks overdue work every six hours and produces KPI snapshots for Google Sheets, Google Drive and PostgreSQL.

WORKFLOW 01-02

Intake protects data quality before AI triage begins



Outcome: every accepted ticket has a single record and a consistent triage result before routing.

06

Figure 4.5 - Intake and AI classification stages of the n8n capstone.

Module 1 output

Five importable workflow exports, twenty-five nodes, database schema, integration guide, validation evidence, dashboard/API demo, presentation and narrated video.

Module 2 - AI Customer Success Manager

The Agents SDK capstone coordinates six specialists: Customer Health, Product Expert, Support Assistant, Escalation Manager, Success Planner and Analytics. Eight tools provide customer profile, usage, support history, product knowledge, health scoring, success planning, follow-up scheduling and escalation.

Two orchestration patterns were implemented. In the manager pattern, the AI Customer Success Manager retains control and calls specialists as tools. In the handoff pattern, a triage agent transfers a focused request directly to the appropriate specialist. SQLiteSession preserves multi-turn context, while Pydantic schemas validate health, intervention and final report objects.

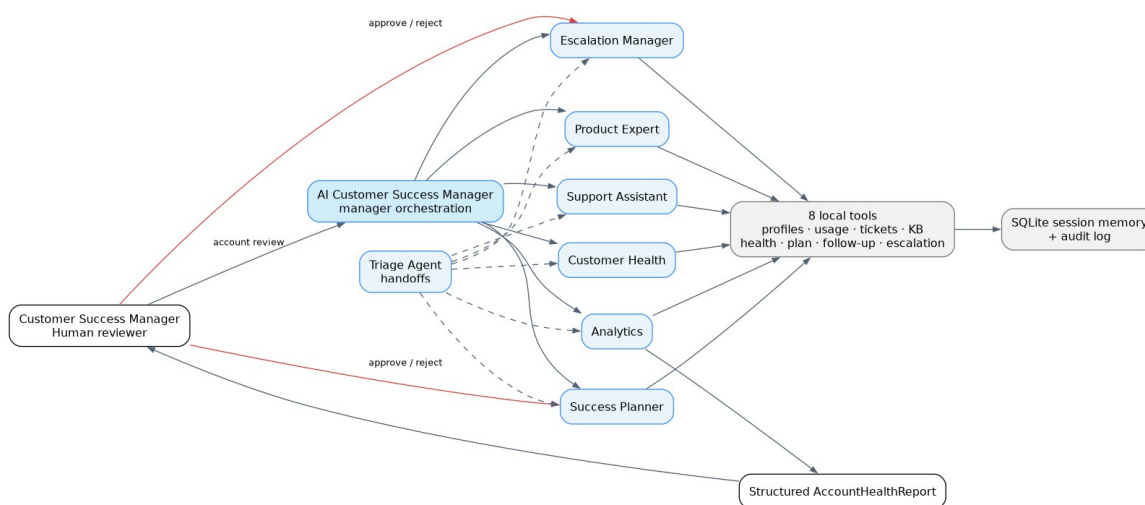


Figure 4.6 - Multi-agent architecture with tools, memory, outputs and human review.

Scheduling and escalation tools declare mandatory approval. The command-line interface exposes the exact pending action, records the user decision and resumes the same run state. This approach provides useful autonomy without removing accountability.

Memory preserves context; approval preserves accountability

SQLite session memory

The SDK stores each turn by session ID, enabling multi-turn account reviews without manual transcript replay.

Human-in-the-loop

Scheduling and escalation declare `needs_approval=True`. The CLI exposes arguments, records the choice, and resumes safely.

8

Figure 4.7 - Session memory and human-approval design.

Module 2 output

Six specialist agents, eight tools, two orchestration patterns, SQLite memory, structured reports, approval controls, tests, documentation, presentation and narrated demo.

Module 3 - Interfaces, Pipelines and Integrations

The supporting interface module demonstrates how the core AI and automation concepts can be exposed through different channels. A Telegram command bot provides message-driven automation. A Gradio and Slack implementation shares a secure reply engine and defines application-mention and direct-message handlers. Hugging Face notebooks demonstrate text, image, audio and video pipeline patterns. Vapi and ElevenLabs evidence covers voice-agent and conversational configuration, while Clay AI supports structured company research and outreach enrichment.

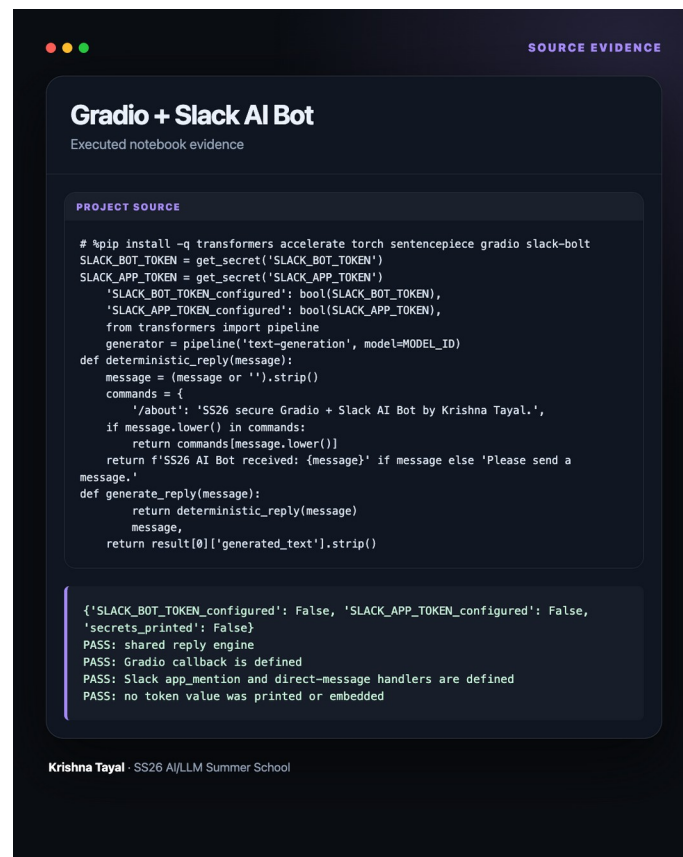


Figure 4.8 - Secure Gradio and Slack bot implementation evidence.

Supporting Applied Project Portfolio

Beyond the two final capstones, the internship produced a connected portfolio of automation, conversational AI, multimodal, research and product-design projects. The summaries below identify the practical problem addressed, the implementation contribution and the evidence prepared for evaluation. This section converts the supporting work from a list of links into a reviewable engineering record.

Portfolio review 1 of 4

Project	Problem / objective	Implementation contribution	Evaluation evidence
Telegram AI Command Bot	Provide a lightweight chat interface for structured AI requests.	An n8n Telegram trigger validates commands, preserves sender context, routes supported requests to Gemini and formats concise chat responses.	Importable workflow, demo request/response, validation notes, GitHub repository and LinkedIn evidence.
Vapi Voice Assistant	Connect natural voice conversations to repeatable business actions.	Vapi conversation context is normalized by n8n and routed to meeting, pricing, support, lead-capture or general-answer branches with human escalation.	Assistant configuration, workflow export, setup guide, screenshots and published project post.
ElevenLabs Conversational Workflow	Turn caller messages into dependable structured actions.	A blocking webhook accepts payload variants, normalizes them to a stable JSON contract and routes lead, lookup and general-response paths.	Two importable workflow configurations, agent screenshot, documentation and LinkedIn evidence.

Portfolio review 2 of 4

Project	Problem / objective	Implementation contribution	Evaluation evidence
10 Reusable n8n Templates	Reduce the time required to start common automation tasks.	Ten documented patterns cover monitoring, APIs, data transformation, notifications and repeatable business workflow structures.	JSON template collections, execution summaries, representative screenshots and project publication.
Automated LinkedIn Publishing	Create a repeatable path from scheduled content to authenticated publishing.	A scheduled n8n flow prepares the post payload, uses OAuth-based LinkedIn access and records publishing status and failure information.	Workflow documentation, content assets and published activity on the student LinkedIn profile.
Gradio + Slack AI Bot	Expose one secure generation function through web and workplace-chat interfaces.	The notebook connects a Gradio UI and Slack event handlers to Hugging Face text generation while reading credentials from secret storage.	Credential-safe executed notebook, event-handling code, output screenshot and dedicated project post.

Portfolio review 3 of 4

Project	Problem / objective	Implementation contribution	Evaluation evidence
Telegram + AI Pipelines Bot	Make multimodal model actions available through simple chat commands.	Telegram commands select text, image, audio or video processing; validation prepares inputs and returns progress, results and safe errors.	Executed notebook, multimodal command evidence, screenshots and project publication.
Hugging Face Multimodal Pipelines	Compare reusable inference patterns across four media categories.	A Colab notebook applies task-specific preprocessing and pretrained pipelines to text, image, audio and video inputs, then records model outputs.	Rerun notebook with student-owned metadata, saved outputs, Drive evidence and LinkedIn post.
Clay AI Company Research	Convert incomplete company records into structured research data.	A two-part enrichment workflow gathers company context, normalizes required fields and produces a reviewable CSV for outreach analysis.	Structured dataset, enrichment screenshot, assignment evidence and professional project post.

Portfolio review 4 of 4

Project	Problem / objective	Implementation contribution	Evaluation evidence
CrewAI Career Roadmap Team	Demonstrate role-based collaboration between specialist agents.	Research, skill-gap analysis, learning-path and synthesis agents work through explicit tasks and combine results into a sequenced roadmap.	Executed notebook, final-output evidence, architecture explanation and LinkedIn publication.
SuccessAI Student Success Platform	Support students before academic risk becomes failure.	The concept combines explainable support signals, personalized planning, AI mentoring and advisor workflows while keeping intervention human-led.	Six prototype views, ten-slide capstone presentation, ethics roadmap, evidence PDF and project post.
Nadi Rakshak	Create public accountability for visible river-pollution reports.	The product concept combines geo-tagged citizen evidence, transparent confidence scoring, human review, a public heatmap and Reported-to-Resolved states.	Problem research, product narrative, risk matrix, data schemas, interface evidence and hackathon presentation.

Engineering Quality, Safety and Evaluation

The portfolio was evaluated as a set of engineering artifacts rather than only as model demonstrations. Particular attention was given to boundary validation, recoverable failure behavior, secure credentials, deterministic business rules, human oversight and reproducibility.

Quality control	Implementation	Reason
Input contracts	Webhook and bot payloads are normalized before downstream use.	Reduces connector-specific errors and makes tests repeatable.
Structured model output	Schemas constrain categories, scores and reports; invalid output enters a safe fallback.	Prevents malformed model text from silently driving operations.
Deterministic decisions	Duplicate fingerprints, health scores and routing conditions are calculated outside the LLM.	Keeps measurable business logic explainable and testable.
Human approval	Scheduling and escalation actions pause before side effects.	Preserves accountability for consequential customer actions.
Secret management	Tokens and keys are loaded from environment variables, Colab Secrets or platform stores.	Keeps public notebooks and repositories credential-safe.
Reproducibility	Offline demos, sample data, validation notes, screenshots and checksums accompany major projects.	Allows evaluation even when an external API or account is unavailable.

Evaluation boundary

The submitted systems demonstrate correct functional behavior and engineering controls using representative data. They are not presented as production load-tested services; latency, concurrency, cost and service-level performance remain future evaluation work.

4.5 Algorithms and Models Used

Customer Health Scoring

The account health score is deterministic and transparent. It combines login recency, feature adoption, seat utilization, NPS sentiment and support burden. The score is calculated before an LLM explains the risks or recommends an intervention, which prevents the model from inventing the numerical result.

Signal	Weight	Purpose
Login recency / activity	25%	Penalizes long periods without product activity
Feature adoption	25%	Measures adoption of available product capability
Seat utilization	20%	Measures active seats relative to licensed seats
NPS sentiment	15%	Normalizes customer sentiment into the score
Support burden	15%	Accounts for open and high-priority tickets

Scores below 40 are classified as critical, 40-59 as high risk, 60-74 as medium risk and 75 or above as low risk. Recommended actions are generated from explicit drivers such as login inactivity, low adoption, underused seats, severe support load, negative NPS or an approaching renewal.

Ticket Triage and Routing

Ticket triage uses a strict JSON schema for category, priority, sentiment, confidence and summary. Invalid or unavailable model output follows a deterministic fallback. The routing stage combines classification with agent skill and workload signals. Critical, low-confidence and policy-sensitive cases are escalated rather than resolved automatically.

Duplicate Prevention and SLA Monitoring

A normalized fingerprint prevents equivalent requests from creating duplicate records within a twenty-four-hour window. Scheduled SLA monitoring checks overdue work every six hours and produces auditable KPI snapshots. Retry policies are applied to critical API calls and workflow handoffs.

4.6 APIs and Integrations

API / integration	Role in the solution
n8n webhooks	Receive external support requests and trigger workflow execution
Ollama API	Local structured ticket triage using qwen2.5:3b
Gemini via LiteLLM	Language-model provider for the live Agents SDK path
Google Calendar	Create review events for approved escalations
Google Sheets and Drive	Store KPI snapshots and timestamped archives
Telegram / aiogram	Message and command interface
Slack Bolt / Socket Mode	Mention and direct-message interface
Vapi and ElevenLabs	Voice and conversational-agent configuration

4.7 Deployment

The n8n capstone provides a self-contained Docker Compose demonstration environment with a dashboard, API, database and imported workflows. The dashboard runs locally, while n8n exposes the production-style intake webhook. Optional Google OAuth credentials can be added through n8n without storing secrets in workflow exports.

The Agents SDK project supports both live and offline execution. The offline demo runs the same data and scoring layer without a network key and produces JSON and Markdown reports. The live CLI uses a configurable Gemini model, persistent SQLite session and approval flow. Both projects are

published in public GitHub repositories and packaged in public Google Drive folders with ZIPs, presentations, README files and videos.

4.8 Final Output, Performance and Results

Output area	Validated result
n8n capstone	5 workflows, 25 nodes, local dashboard/API/database, deck and demo video
Agents capstone	6 specialists, 8 tools, memory, structured reports, approvals, 4 passing tests
Automation portfolio	Telegram bot, Vapi, ElevenLabs, 10 n8n templates and LinkedIn automation
AI pipeline portfolio	Text, image, audio and video pipeline evidence
Applied research and interfaces	Nadi Rakshak, SuccessAI, Clay AI, CrewAI and secure Gradio/Slack work
Professional publication	Public GitHub repositories, Drive packages and 13 project-specific LinkedIn posts

Functional validation focused on correctness, reproducibility, schema compliance, error handling and secure approval behavior. The projects are demonstrations rather than production load-tested services; future benchmarking can measure throughput, concurrency, cost and service-level performance under realistic traffic.

CHAPTER 5

Weekly Work Log

Chapter overview

This chapter provides a week-wise record of activities, learning outcomes and approximate engagement hours.

Week	Activity	Key learning	Hours
Week 1	Orientation, AI tools, Python and Colab setup, foundational assignments	Prompting discipline, notebook execution, documentation habits	30
Week 2	Stakeholder analysis, capstone selection, workflow and architecture design	Requirement decomposition and system thinking	30
Week 3	Built n8n support workflows, Telegram bot, Vapi and ElevenLabs evidence	Triggers, branches, retries, API integration and secret handling	32
Week 4	Built multi-agent customer-success project and Hugging Face pipelines	Agents, tools, handoffs, structured outputs, scoring and memory	34
Week 5	Testing, Docker/local deployment, validation and GitHub publication	Reproducible builds, tests, deployment checks and version control	30
Week 6	Presentations, demos, screenshots, LinkedIn posts and final report	Technical storytelling, visual evidence and submission quality	24

Week-wise Reflection

The work progressed from broad tool exploration to increasingly structured engineering. Early weeks focused on fluency and setup. Middle weeks required architectural choices and implementation. Final weeks emphasized validation, security, communication and packaging. This progression made the relationship between coding, automation, AI reasoning and professional delivery clear.

CHAPTER 6

Challenges Faced

Chapter overview

This chapter describes the technical, communication and time-management challenges encountered and the methods used to resolve them.

Challenge	Impact	Resolution
Integration reliability	Different services use different payloads, credentials and error behavior.	Normalized input schemas, isolated connectors and added deterministic fallbacks.
Structured AI output	Model responses can be malformed or invalid JSON.	Used strict schemas, validation and safe fallback classifications.
Human control	Autonomous scheduling or escalation can create operational risk.	Marked side-effecting tools as approval-required and recorded audit events.
State and memory	Multi-turn agent work requires context without exposing secrets.	Used SQLite sessions for context and excluded runtime state from repositories.
Submission consistency	Artifacts existed across notebooks, Drive, GitHub and presentations.	Created standardized folders, README files, link indexes and checksums.
Time management	Several platforms and deliverables had to be completed in parallel.	Used a weekly plan, prioritized capstones and reused shared documentation assets.

Problem-Solving Approach

1. Reproduce the issue with the smallest possible input.
2. Separate model behavior from deterministic business logic.
3. Validate data at system boundaries before the next workflow stage.
4. Prefer recoverable fallbacks and human review for uncertain cases.
5. Document the final command, configuration and evidence after resolution.

CHAPTER 7

Skills Developed

Chapter overview

This chapter summarizes the technical and professional capabilities strengthened during the internship.

7.1 Technical Skills

Skill area	Capability developed
Programming	Python functions, data models, notebooks, JSON handling and CLI workflows
Databases	PostgreSQL schema use, SQLite session memory, JSONL audit logs and CSV evidence
Artificial intelligence	LLM prompting, structured outputs, transformers, agent tools and health scoring
Automation	n8n triggers, webhooks, sub-workflows, branching, retries, schedules and integrations
Cloud and APIs	Google Workspace integrations, Gemini, Vapi, ElevenLabs, Slack and Telegram
Version control	Git repository structure, GitHub publication, validation and secret-safe packaging
Tools	Docker Compose, Gradio, Hugging Face, LangChain concepts, CrewAI and diagramming

7.2 Professional Skills

- Communication: converted technical implementation into README files, presentations, demos and public posts.
- Teamwork: used peer discussion and feedback to refine project scope and evidence.
- Leadership: owned the final structure, testing and delivery of both capstones.
- Presentation: prepared project decks, narrated demonstrations and visual explanations.
- Time management: balanced training, implementation, certification and publication work.
- Critical thinking: separated evidence, assumptions, business rules and model-generated output.

CHAPTER 8

Learning Outcomes

Chapter overview

This chapter connects academic concepts with industry-oriented engineering practices, ethics, collaboration and career insights.

8.1 Academic Concepts Applied

The internship applied programming fundamentals, data structures, APIs, databases, software modularity, testing and system design. AI concepts included transformer pipelines, prompt design, classification, sentiment, structured generation, deterministic scoring and agent orchestration.

8.2 Industry Exposure

The projects reflected common industry requirements: multiple channels, external integrations, persistent state, auditability, safe fallbacks, approval controls, deployment instructions and measurable operational outcomes. Packaging projects for public review also demonstrated the importance of documentation and reproducibility.

8.3 Best Practices Learned

- Keep credentials in secret stores or environment configuration, never in source files.
- Validate untrusted input and model output before downstream execution.
- Use deterministic logic for scores and high-impact decisions.
- Place human approval before side effects that affect customers or operations.
- Maintain tests, README files, architecture diagrams and reproducible demos.
- Treat communication and evidence quality as part of engineering quality.

8.4 Professional Ethics

Responsible AI requires transparency about what a model can and cannot decide. The projects keep sensitive actions under human control, minimize data shared with models, avoid embedding credentials and preserve an audit trail. These choices reduce privacy, security and accountability risks.

8.5 Team Collaboration and Career Insights

The programme showed that AI application development combines software engineering, automation, product thinking and communication. The experience strengthened my interest in AI product development, workflow automation and full-stack systems that integrate models with reliable operational tools.

CHAPTER 9

Self Assessment

Chapter overview

This chapter reflects on strengths, areas for improvement and the overall internship experience.

9.1 Strengths

- Strong willingness to explore unfamiliar platforms and connect them into working systems.
- Ability to move from a broad problem statement to modular workflows and agent roles.
- Attention to documentation, submission packaging and professional project presentation.
- Security awareness around credentials, approval controls and public repositories.
- Persistence in testing, rerunning and validating artifacts across multiple environments.

9.2 Areas for Improvement

- Develop deeper production benchmarking for concurrency, latency, cost and reliability.
- Improve automated integration testing across n8n, model providers and messaging platforms.
- Strengthen frontend engineering for richer operational dashboards and user interfaces.
- Study evaluation methods for LLM quality, hallucination detection and prompt regression testing.
- Gain more experience with cloud deployment, observability and production security controls.

9.3 Reflection on the Internship Experience

The internship transformed a broad interest in AI into a clearer engineering workflow: understand the user problem, define measurable objectives, design the system, separate deterministic and model-driven behavior, test failure paths, protect sensitive actions and communicate the result. Building complete capstone packages was especially valuable because it required technical depth and disciplined presentation at the same time.

CHAPTER 10

Conclusion and Future Scope

Chapter overview

This chapter summarizes the internship contribution and identifies practical improvements and career directions.

10.1 Conclusion

AI & LLM Summer School 2026 provided a practical pathway from AI foundations to integrated applications. The internship resulted in an AI-powered support and success automation suite, supported by Python notebooks, multimodal pipelines, bots, voice integrations, data enrichment, documentation, public repositories, demonstrations and professional posts.

The most important learning was that useful AI applications require more than a model call. Reliable systems need clean input, deterministic rules, structured outputs, error handling, memory, secure credentials, human approval, persistence, testing and clear documentation. The capstone projects demonstrate these principles through n8n workflow automation and an Agents SDK application.

10.2 Future Improvements

- Deploy the systems on managed infrastructure with HTTPS, authentication and observability.
- Add production CRM, help-desk and analytics connectors with least-privilege access.
- Create automated evaluation datasets for triage quality and agent recommendations.
- Add load tests, cost dashboards, tracing and service-level monitoring.
- Expand the interface layer with a web dashboard and role-based operational views.
- Improve retrieval with vector search and governed knowledge-base updates.

10.3 Recommendations

Future programme batches can benefit from shared evaluation rubrics, reference payloads for integrations, small production-style datasets and scheduled architecture reviews. A final demonstration checklist that covers security, validation, deployment and evidence would help students present more consistent projects.

10.4 Career Plans

I plan to continue developing skills in AI product engineering, agentic systems, automation, full-stack development and cloud deployment. The portfolio created during this internship provides a foundation for internships and projects that require practical integration of AI models with real software systems.

APPENDIX A

Evidence and Repository Links

Chapter overview

This appendix provides the primary submission, Google Drive and GitHub links for evaluator access.

Resource	Link
Master Summer School 2026 evidence folder	Open Drive
Capstone 2 - n8n Customer Support Automation	Open Drive
Capstone 3 - AI Customer Success Manager	Open Drive
AI First Hackathon - Nadi Rakshak	Open Drive
Python and Colab work	Open Drive
Hugging Face pipelines	Open Drive
Capstone submission-links document	Open document

GitHub Repositories

Repository	GitHub
SS26 AI/LLM Summer School Portfolio	Open GitHub
AI Customer Support Automation - n8n Capstone	Open GitHub
AI Customer Success Manager - Agents SDK Capstone	Open GitHub
n8n Telegram AI Command Bot	Open GitHub

APPENDIX B

LinkedIn Project Posts

Chapter overview

This appendix lists the individual LinkedIn project publications prepared as professional evidence of the internship work.

Project publication	LinkedIn
AI Customer Support Automation - n8n Capstone	Open post
AI Customer Success Manager - Agents SDK Capstone	Open post
Telegram AI Command Bot - n8n	Open post
Vapi + n8n Voice Assistant	Open post
ElevenLabs Conversational AI Agent + n8n	Open post
10 Reusable n8n Templates	Open post
Gradio + Slack AI Bot	Open post
Telegram + AI Pipelines Bot	Open post
Nadi Rakshak - AI First Hackathon	Open post
Hugging Face Multimodal AI Pipelines	Open post
Clay AI Company Research	Open post
CrewAI Career Roadmap Team	Open post
SuccessAI - Student Success Platform	Open post

Publication approach

Each project was published as a separate media post so evaluators can review the project context, technical contribution and visual evidence independently.

APPENDIX C

Project Screenshots

Chapter overview

This appendix contains selected implementation and output screenshots from the internship portfolio.

The evidence gallery below uses screenshots and rendered project artifacts from the capstone, automation, bot, pipeline, voice-agent, research and hackathon project folders.

WORKFLOW 01-02

Intake protects data quality before AI triage begins

01 Ticket Intake & Registration

- Webhook receives the customer request
- Normalizes email, subject, message, source
- Creates a 24-hour duplicate fingerprint
- Stops duplicates and starts AI triage

02 AI Classification & Prioritization

- Ollama API returns structured JSON
- Validates the full triage schema
- Saves category, priority, sentiment, summary
- Retries three times; safe fallback remains

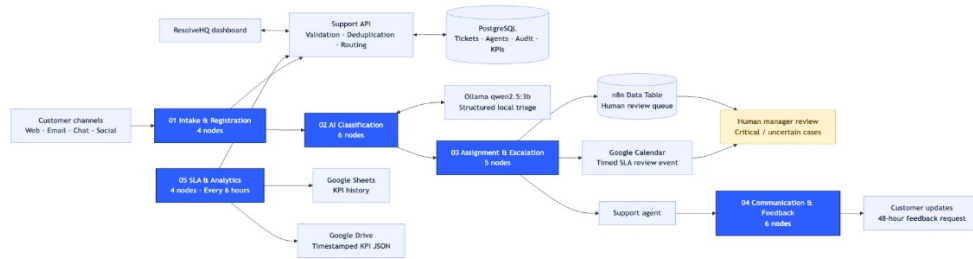
Outcome: every accepted ticket has a single record and a consistent triage result before routing.

06

Figure C.1 - n8n customer-support workflow implementation overview.

ARCHITECTURE

Five n8n workflows connect the complete ticket lifecycle



Ollama, PostgreSQL, Calendar, Sheets, Drive, and one n8n Data Table share an auditable lifecycle.

04

Figure C.2 - Capstone 2 architecture connecting the five n8n workflows and supporting services.

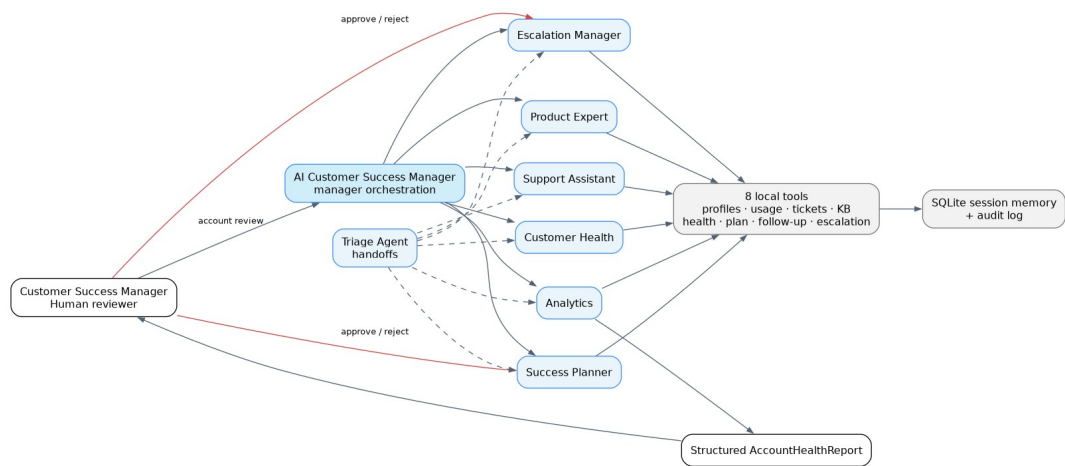


Figure C.3 - AI Customer Success Manager multi-agent system architecture.

Memory preserves context; approval preserves accountability

SQLite session memory

The SDK stores each turn by session ID, enabling multi-turn account reviews without manual transcript replay.

Human-in-the-loop

Scheduling and escalation declare `needs_approval=True`. The CLI exposes arguments, records the choice, and resumes safely.

8

Figure C.4 - Session memory, approval controls and auditability in the Agents SDK capstone.

Validation covers behavior, architecture, and packaging



TESTS

Three unit tests pass for score bounds, risk classification, and structured output.



SDK OBJECTS

Six specialists, eight tools, six handoffs, and six manager tools construct successfully.

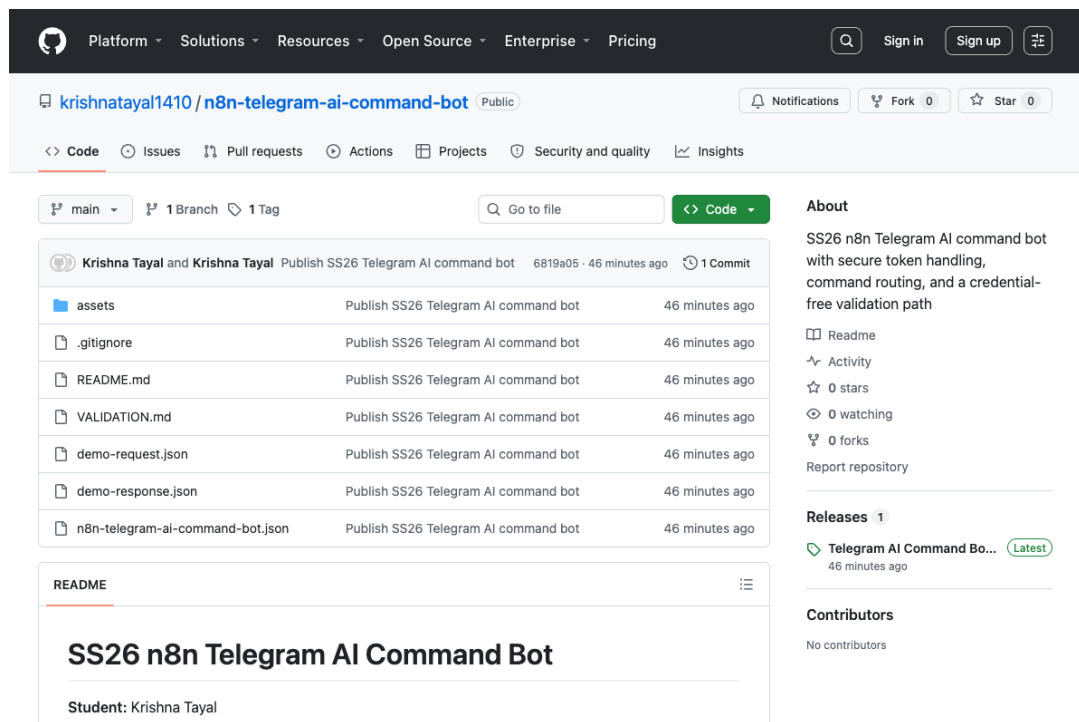


SAFE DEMO

Offline JSON and Markdown reports run without credentials or network access.

10

Figure C.5 - Capstone 3 validation coverage for behavior, SDK construction and safe offline execution.



The screenshot shows the GitHub interface for a new public repository named `n8n-telegram-ai-command-bot` by user `krishnatayal1410`. The repository was created 46 minutes ago and contains 1 commit. The file list includes `assets`, `.gitignore`, `README.md`, `VALIDATION.md`, `demo-request.json`, `demo-response.json`, and `n8n-telegram-ai-command-bot.json`. The README file is selected, showing the title **SS26 n8n Telegram AI Command Bot** and the author **Student: Krishna Tayal**. The right sidebar provides details about the repository, including its description, activity, and release information.

About

SS26 n8n Telegram AI command bot with secure token handling, command routing, and a credential-free validation path

Releases 1

Telegram AI Command Bo... (Latest) 46 minutes ago

Contributors

No contributors

Figure C.6 - Public GitHub repository for the n8n Telegram AI command bot.

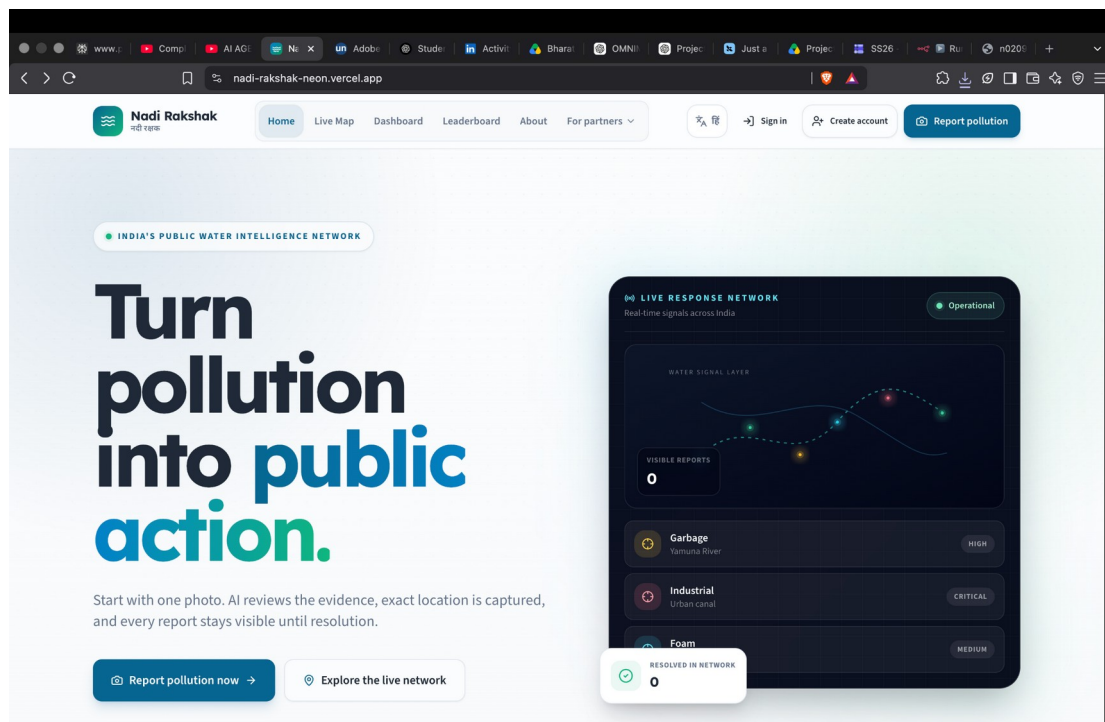


Figure C.7 - Nadi Rakshak hackathon project interface.

SOURCE EVIDENCE

Hugging Face Multimodal Pipelines

Executed notebook evidence

PROJECT SOURCE

```
#1. Text Lowercasing
text1 = "Hello, World world"
print(text1.lower())
text2 = "NLP Is FuN"
print(text2.lower())
text1 = "hello!!! How are you?"
print(text1.translate(str.maketrans('', '', string.punctuation)))
text2 = "NLP Is FuN"
print(text2.translate(str.maketrans('', '', string.punctuation)))
text1 = "NLP 101 is easy"
print(re.sub(r'\d+', '', text1))
text2 = "I scored 95 marks"
print(re.sub(r'\d+', '', text2))
text1 = "NLP is amazing"
print(word_tokenize(text1))
text2 = "I love AI. Added more content"
print(word_tokenize(text2))
text1 = "NLP is amazing"
```

```
hello, world world
nlp is fun
hello How are you
NLP Is FuN
NLP is easy
I scored marks
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
```

Krishna Tayal · SS26 AI/LLM Summer School

Figure C.8 - Hugging Face multimodal pipeline execution evidence.

SOURCE EVIDENCE

Gradio + Slack AI Bot

Executed notebook evidence

PROJECT SOURCE

```
# %pip install -q transformers accelerate torch sentencepiece gradio slack-bolt
SLACK_BOT_TOKEN = get_secret('SLACK_BOT_TOKEN')
SLACK_APP_TOKEN = get_secret('SLACK_APP_TOKEN')
'SLACK_BOT_TOKEN_configured': bool(SLACK_BOT_TOKEN),
'SLACK_APP_TOKEN_configured': bool(SLACK_APP_TOKEN),
from transformers import pipeline
generator = pipeline('text-generation', model=MODEL_ID)
def deterministic_reply(message):
    message = (message or '').strip()
    commands = {
        '/about': 'SS26 secure Gradio + Slack AI Bot by Krishna Tayal.',
    }
    if message.lower() in commands:
        return commands[message.lower()]
    return f'SS26 AI Bot received: {message}' if message else 'Please send a message.'
def generate_reply(message):
    return deterministic_reply(message)
    message,
    return result[0]['generated_text'].strip()
```

```
{'SLACK_BOT_TOKEN_configured': False, 'SLACK_APP_TOKEN_configured': False,
'secrets_printed': False}
PASS: shared reply engine
PASS: Gradio callback is defined
PASS: Slack app_mention and direct-message handlers are defined
PASS: no token value was printed or embedded
```

Krishna Tayal · SS26 AI/LLM Summer School

Figure C.9 - Gradio and Slack AI bot implementation evidence.

SOURCE EVIDENCE

Telegram + AI Pipelines Bot

Executed notebook evidence

PROJECT SOURCE

```

from aiogram.filters import Command
from aiogram.filters import Command
@dp.message(Command('start'))
async def command_start(message: types.Message):
    "This handler will reieve the message with/start command"
    await message.reply("Hi!\nI,m SS26 Bot!\nPowered by SummerSchool26")
@dp.message(Command('help'))
async def cmd_help(message: types.Message):
    "This will receive Message with /help command"
    help_text="""
    Refer to these commands:
    await message.reply(help_text)
#other message
#@dp.message()
#async def other_messages(message: types.Message):
    #"This will receive other messages"
    #await message.reply(message.text)
context = []

```

Requirement already satisfied: aiogram in /usr/local/lib/python3.12/dist-packages (3.29.1)

Requirement already satisfied: aiofiles<26.0,>=23.2.1 in /usr/local/lib/python3.12/dist-packages (from aiogram) (25.1.0)

Requirement already satisfied: aiohttp<3.15,>=3.9.0 in /usr/local/lib/python3.12/dist-packages (from aiogram) (3.14.1)

Requirement already satisfied: certifi>=2023.7.22 in

Krishna Tayal · SS26 AI/LLM Summer School

Figure C.10 - Telegram bot integrated with AI media pipelines.

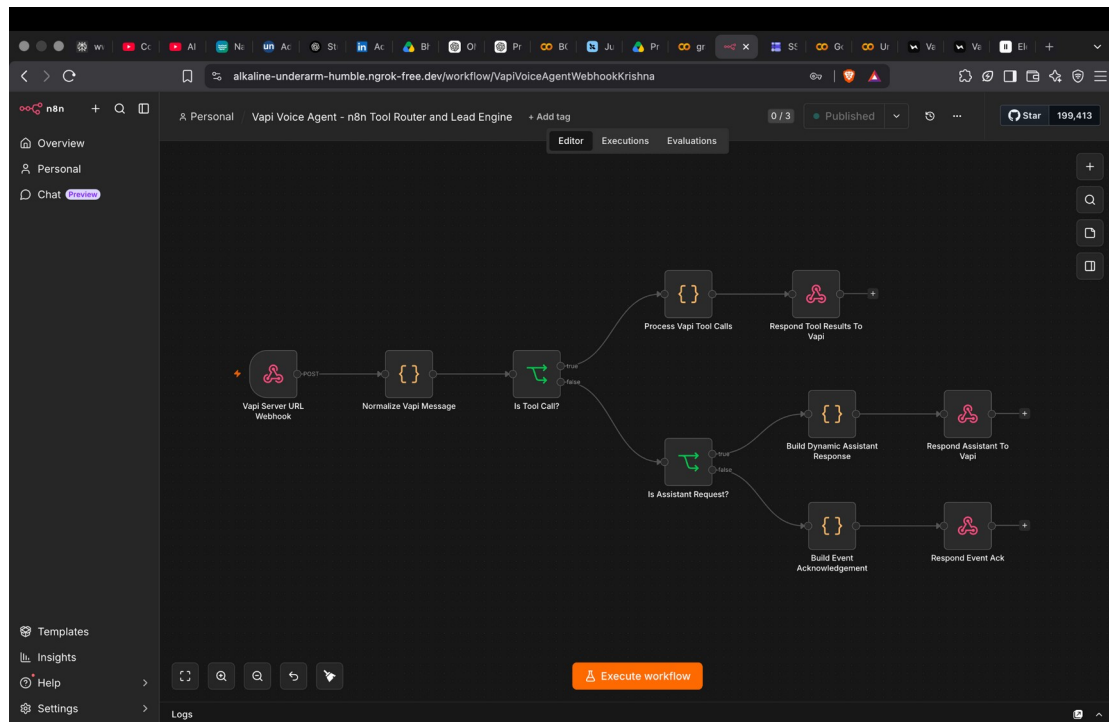


Figure C.11 - Vapi and n8n workflow integration evidence.

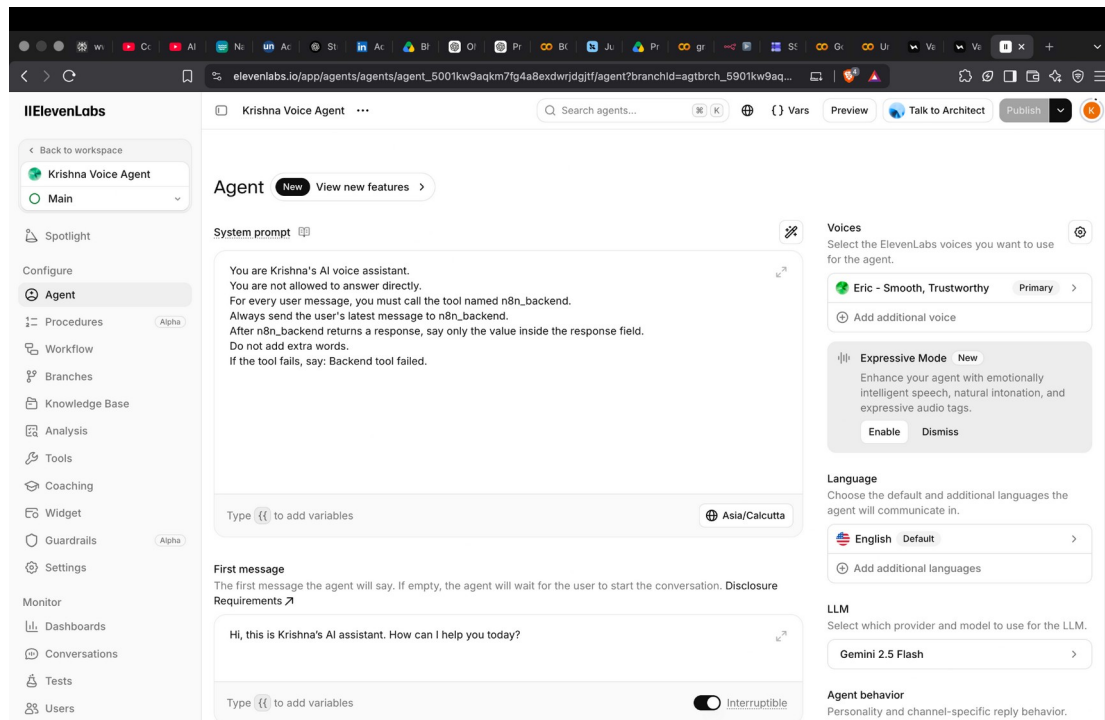


Figure C.12 - ElevenLabs conversational-agent configuration evidence.

The image is a screenshot of a presentation slide. At the top left, there are three colored circles (red, yellow, green). At the top right, the text "SOURCE EVIDENCE" is displayed in red. The main title "Clay AI Company Research" is in a large, bold, white font. Below it, "Submission dataset preview" is written in a smaller white font. The slide features a dark blue background with a lighter blue rectangular area in the center containing a list of research fields. The list is titled "PROJECT SOURCE" in red. The fields are listed in a single paragraph, separated by commas. At the bottom left, there is a red vertical bar. At the bottom right, the text "AI-enriched research fields preserved in CSV" is written in white. The footer at the very bottom reads "Krishna Tayal SS26 AI/LLM Summer School" in white.

SOURCE EVIDENCE

Clay AI Company Research

Submission dataset preview

PROJECT SOURCE

"Company Name", "Website", "Industry", "Country", "Company Size", "Enrich Company", "Description", "Url", "Employee Count", "Industry (2)", "Founded", "Country (2)", "Size", "Business Description", "Use AI business Description", "AI Business Challenge", "Use AI company Name", "Use AI ai Business Challenge", "AI Solution Suggestion", "Use AI company Name (2)", "Use AI ai Solution", "AI Cold Email Draft", "Use AI subject", "Use AI body", "AI Lead Score", "Use AI ai Lead Score", "Use AI reason", "Website Summary", "Main Products or Services", "Target Customers or Market", "Business or Technology Focus"
"Microsoft", "microsoft.com", "Software", "USA", "Large", "Microsoft", "Every company has a mission. What's ours? To empower every person and every organization to achieve more. We believe technology can and should be a force for good and that meaningful innovation contributes to a brighter world in the future and today. Our culture doesn't just encourage curiosity; it embraces it. Each day we make progress together by showing up as our authentic selves. We show up with a learn-it-all mentality. We show up cheering on others, knowing their success doesn't diminish our own. We show up every day open to learning our own biases, changing our behavior, and inviting in differences. Because impact matters.
Microsoft operates in 190 countries and is made up of approximately 228,000 passionate employees worldwide.", "https://www.linkedin.com/company/microsoft/", 233541, "Software Development", "US", "10,001+ employees", "Microsoft is a large software company that develops and provides software, services, and solutions.", "Microsoft is a large software company that develops and provides software, services, and solutions.", "Response", "Microsoft", "Microsoft could use AI to prioritize support and

AI-enriched research fields preserved in CSV

Krishna Tayal SS26 AI/LLM Summer School

Figure C.13 - Clay AI enriched company-research dataset evidence.

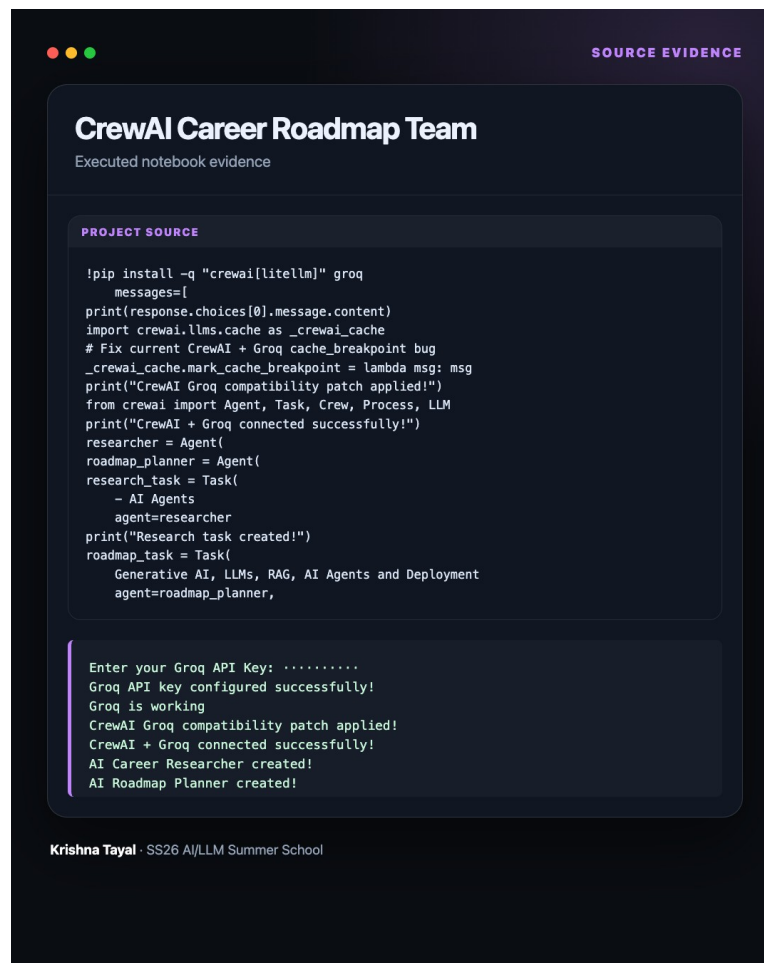


Figure C.14 - CrewAI multi-agent career-roadmap project evidence.

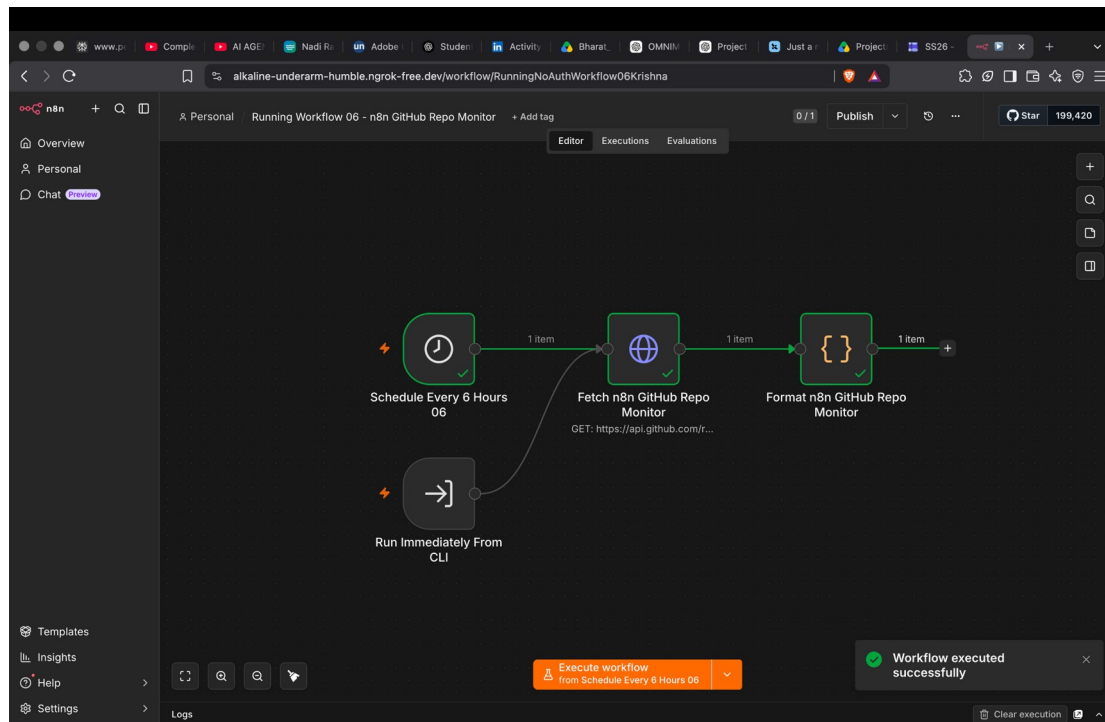


Figure C.15 - Reusable n8n GitHub repository monitor template executed successfully.

APPENDIX D

Selected Source Code

Chapter overview

This appendix presents compact code extracts that demonstrate deterministic scoring and human approval controls in the Agents SDK project.

```
def assess_health(account):
    activity = clamp(100 - account.days_since_login * 4)
    adoption = account.feature_adoption * 100
    seat_utilization = clamp(
        account.active_seats / account.licensed_seats * 100
    )
    sentiment = (account.nps + 100) / 2
    support = clamp(
        100 - account.open_tickets * 5
        - account.high_priority_tickets * 15
    )

    score = round(
        activity * 0.25 + adoption * 0.25
        + seat_utilization * 0.20
        + sentiment * 0.15 + support * 0.15
    )
```

Listing D.1 - Deterministic customer health-score calculation.

The score is calculated from explicit business signals before an LLM produces narrative analysis. This preserves explainability and makes unit tests deterministic.

```
@tool(needs_approval=True)
def schedule_follow_up(account_id, purpose, days_from_now):
    account = get_account(account_id)
    event = append_audit_event(
        "follow_up_scheduled",
        {
            "account_id": account_id,
            "company": account.company,
            "purpose": purpose,
            "days_from_now": days_from_now,
            "owner": account.owner,
        },
    )
    return json.dumps(event, indent=2)
```

Listing D.2 - Approval-gated follow-up scheduling tool.

The approval decorator pauses the side-effecting tool call until a human reviews the exact request. Approved actions are written to an audit log.

APPENDIX E

Deliverables and References

Chapter overview

This appendix summarizes the submitted artifacts and the main technical references used during implementation.

Deliverable	Contents
Capstone 2 package	Source workflows, database schema, documentation, diagrams, deck, demo video and ZIP
Capstone 3 package	Python package, tests, data, documentation, diagrams, deck, demo video and ZIP
Training evidence	Python/Colab notebooks, AI tool assignments and course certificates
Integration evidence	Vapi, ElevenLabs, Telegram, Slack, Gradio, Hugging Face and Clay AI artifacts
Publication evidence	GitHub repositories, Google Drive folders and LinkedIn project posts
Editable diagrams	Mermaid source, SVG, PNG and Excalidraw files in the report package

Technical References

Reference	Link
OpenAI Agents SDK documentation	Open reference
OpenAI Agents SDK handoffs	Open reference
OpenAI Agents SDK human-in-the-loop	Open reference
OpenAI Agents SDK sessions	Open reference
Google Gemini models documentation	Open reference
n8n documentation	Open reference
Hugging Face Transformers documentation	Open reference

APPENDIX F

Certificates and Credentials

Chapter overview

This appendix reproduces verified course-completion credentials issued to Krishna Tayal and provides public verification links where available.

The certificate images below are reproduced from the original files or official public verification pages. Each entry is identified by issuer, title and issue period for straightforward evaluator review.

Credential	Issuer	Issued
n8n Quickstart	n8n Academy	23 June 2026
Essentials: Your First Workflows	n8n Academy	3 August 2026
Integrations: APIs & Connected Workflows	n8n Academy	3 August 2026
In Practice: AI, Testing & Best Practices	n8n Academy	3 August 2026
Fundamentals of Agents	Hugging Face	5 August 2026
The LLM Course - Fundamentals of LLMs	Hugging Face	5 August 2026
Quickstart: LangChain Essentials - Python	LangChain Academy	5 August 2026
Introduction to Subagents	Anthropic	July 2026
Introduction to Claude Cowork	Anthropic	July 2026
Introduction to Agent Skills	Anthropic	July 2026
Claude Platform 101	Anthropic	July 2026
Claude Code in Action	Anthropic	July 2026
Claude Code 101	Anthropic	July 2026
AI Fluency: Framework & Foundations	Anthropic	July 2026

COMPLETE CERTIFICATE VISUAL INDEX

14 OF 14 VERIFIED COURSE CREDENTIALS | INDEX 1 OF 3

Every thumbnail below corresponds to a full-size certificate reproduction on the following pages, with issuer details and evidence access retained for academic review.

 1. n8n Quickstart n8n Academy 23 June 2026	 2. Essentials: Your First Workflows n8n Academy 3 August 2026	 3. Integrations: APIs & Connected Workflows n8n Academy 3 August 2026
 4. In Practice: AI, Testing & Best Practices n8n Academy 3 August 2026	 5. Fundamentals of Agents Hugging Face 5 August 2026	 6. The LLM Course - Fundamentals of LLMs Hugging Face 5 August 2026

COMPLETE CERTIFICATE VISUAL INDEX

14 OF 14 VERIFIED COURSE CREDENTIALS | INDEX 2 OF 3

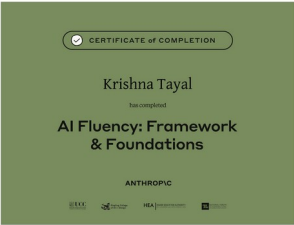
Every thumbnail below corresponds to a full-size certificate reproduction on the following pages, with issuer details and evidence access retained for academic review.

 7. Quickstart: LangChain Essentials - Python LangChain Academy 5 August 2026	 8. Introduction to Subagents Anthropic July 2026	 9. Introduction to Claude Cowork Anthropic July 2026
 10. Introduction to Agent Skills Anthropic July 2026	 11. Claude Platform 101 Anthropic July 2026	 12. Claude Code in Action Anthropic July 2026

COMPLETE CERTIFICATE VISUAL INDEX

14 OF 14 VERIFIED COURSE CREDENTIALS | INDEX 3 OF 3

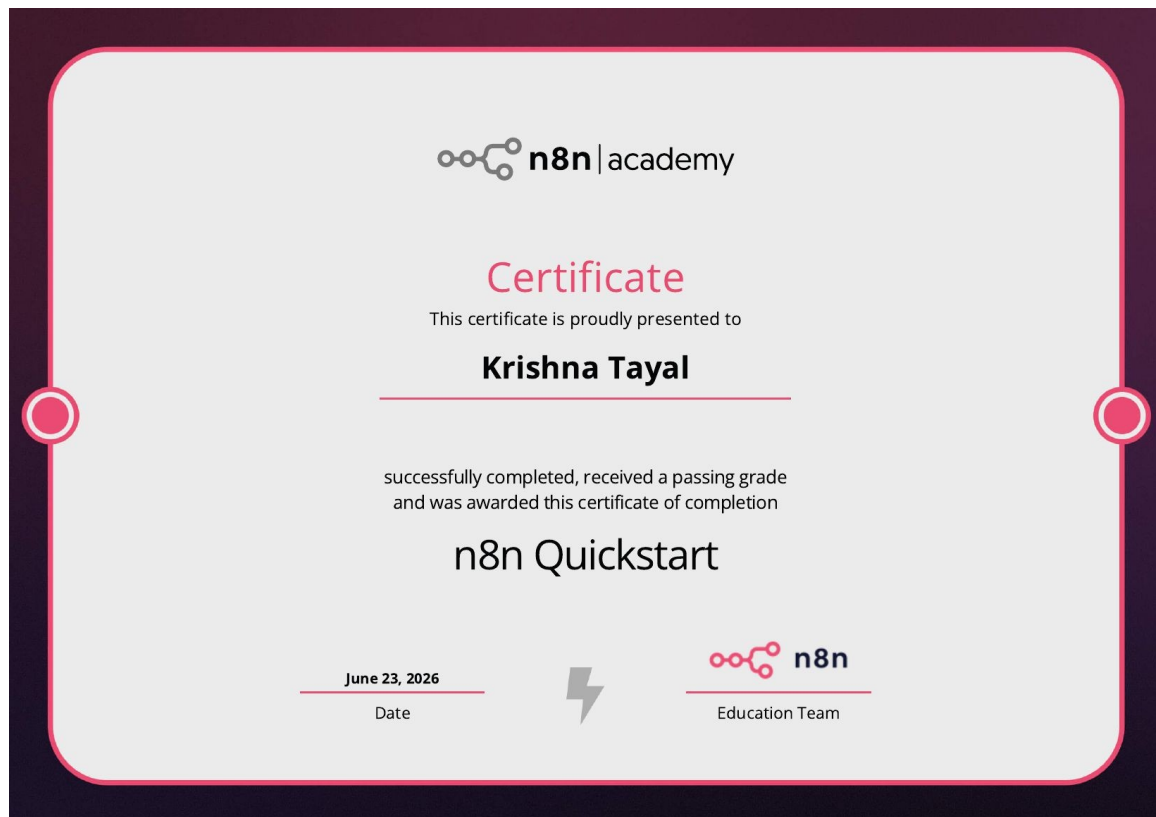
Every thumbnail below corresponds to a full-size certificate reproduction on the following pages, with issuer details and evidence access retained for academic review.

 A green certificate thumbnail with a white border. At the top, it says 'CERTIFICATE of COMPLETION' with a checkmark icon. Below that, the name 'Krishna Tayal' is listed with 'has completed' underneath. The course title 'Claude Code 101' is prominently displayed. At the bottom, the 'ANTHROPIC' logo is visible. 13. Claude Code 101 Anthropic July 2026	 A green certificate thumbnail with a white border. At the top, it says 'CERTIFICATE of COMPLETION' with a checkmark icon. Below that, the name 'Krishna Tayal' is listed with 'has completed' underneath. The course title 'AI Fluency: Framework & Foundations' is prominently displayed. At the bottom, the 'ANTHROPIC' logo is visible, along with several partner logos including MIT, Google, and others. 14. AI Fluency: Framework & Foundations Anthropic July 2026
--	---

CERTIFICATE F.1

n8n Quickstart

n8n Academy | 23 June 2026



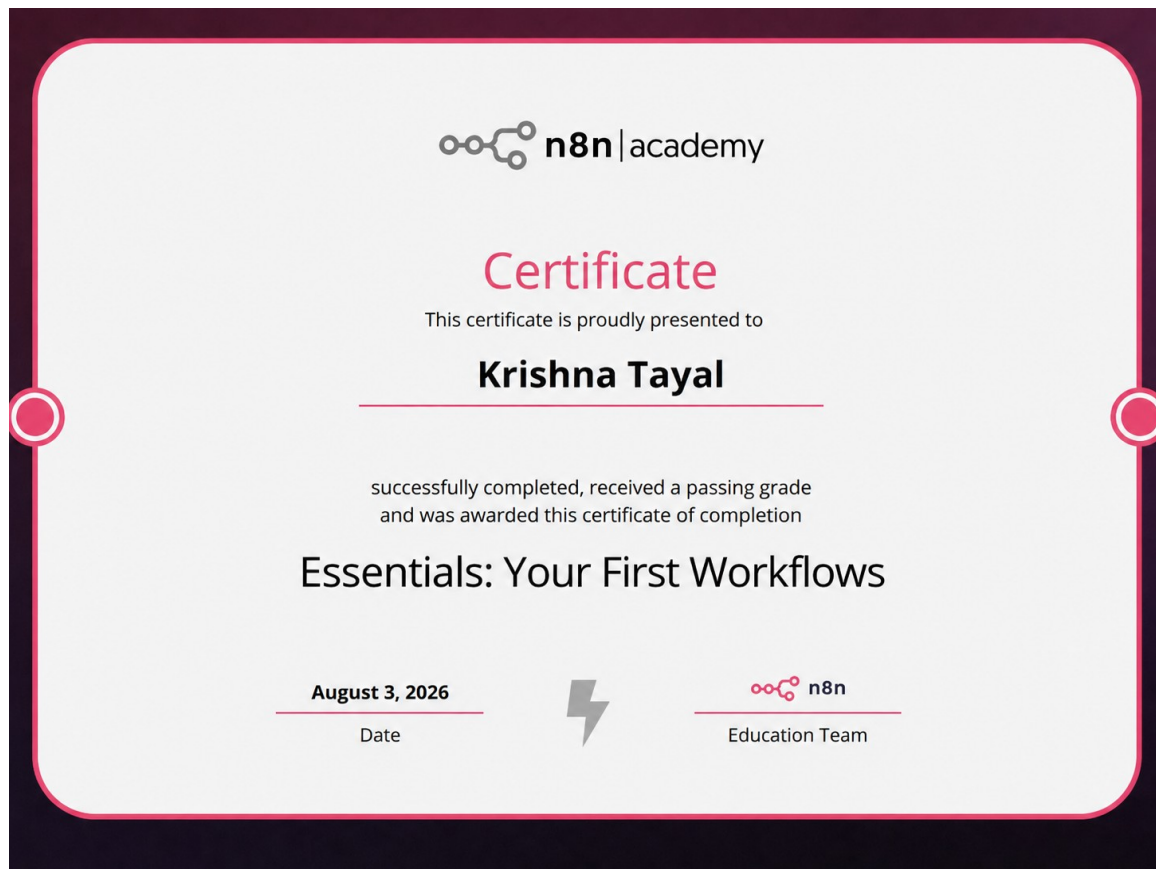
Certificate F.1 - n8n Quickstart, issued by n8n Academy.

[Open credential verification or evidence record](#)

CERTIFICATE F.2

Essentials: Your First Workflows

n8n Academy | 3 August 2026



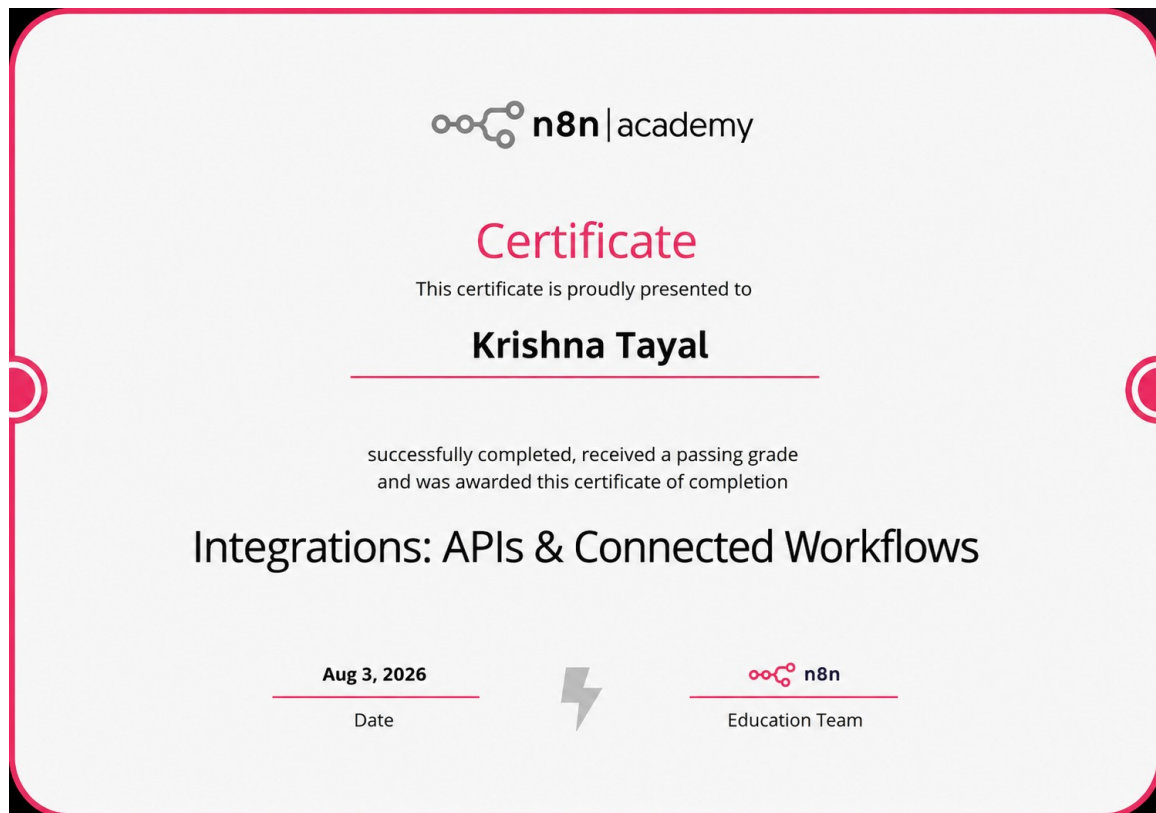
Certificate F.2 - Essentials: Your First Workflows, issued by n8n Academy.

[Open credential verification or evidence record](#)

CERTIFICATE F.3

Integrations: APIs & Connected Workflows

n8n Academy | 3 August 2026



Certificate F.3 - Integrations: APIs & Connected Workflows, issued by n8n Academy.

[Open credential verification or evidence record](#)

CERTIFICATE F.4

In Practice: AI, Testing & Best Practices

n8n Academy | 3 August 2026



Certificate F.4 - In Practice: AI, Testing & Best Practices, issued by n8n Academy.

[Open credential verification or evidence record](#)

CERTIFICATE F.5

Fundamentals of Agents

Hugging Face | 5 August 2026



Certificate F.5 - Fundamentals of Agents, issued by Hugging Face.

[Open credential verification or evidence record](#)

CERTIFICATE F.6

The LLM Course - Fundamentals of LLMs

Hugging Face | 5 August 2026



Certificate F.6 - The LLM Course - Fundamentals of LLMs, issued by Hugging Face.

[Open credential verification or evidence record](#)

CERTIFICATE F.7

Quickstart: LangChain Essentials - Python

LangChain Academy | 5 August 2026

Course Completed

This Certificate is Awarded To

Krishna Tayal

For Successfully Completing The Course

Quickstart: LangChain Essentials - Python

 **LangChain Academy**

Issued: 2026-08-05

2028-08-04

Certificate ID: d3jzhz7cim

Certificate F.7 - Quickstart: LangChain Essentials - Python, issued by LangChain Academy.

[Open credential verification or evidence record](#)

CERTIFICATE F.8

Introduction to Subagents

Anthropic | July 2026



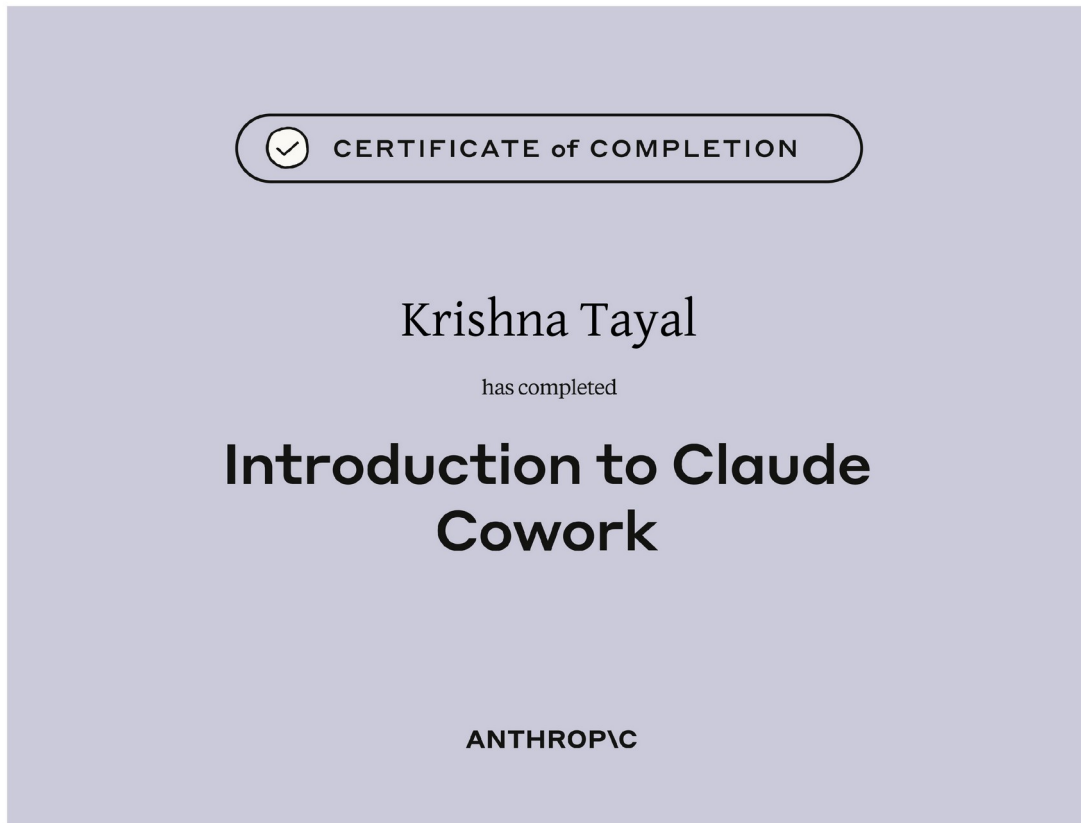
Certificate F.8 - Introduction to Subagents, issued by Anthropic.

[Open credential verification or evidence record](#)

CERTIFICATE F.9

Introduction to Claude Cowork

Anthropic | July 2026



Certificate F.9 - Introduction to Claude Cowork, issued by Anthropic.

[Open credential verification or evidence record](#)

CERTIFICATE F.10

Introduction to Agent Skills

Anthropic | July 2026



Certificate F.10 - Introduction to Agent Skills, issued by Anthropic.

[Open credential verification or evidence record](#)

CERTIFICATE F.11

Claude Platform 101

Anthropic | July 2026



Certificate F.11 - Claude Platform 101, issued by Anthropic.

[Open credential verification or evidence record](#)

CERTIFICATE F.12

Claude Code in Action

Anthropic | July 2026



Certificate F.12 - Claude Code in Action, issued by Anthropic.

[Open credential verification or evidence record](#)

CERTIFICATE F.13

Claude Code 101

Anthropic | July 2026



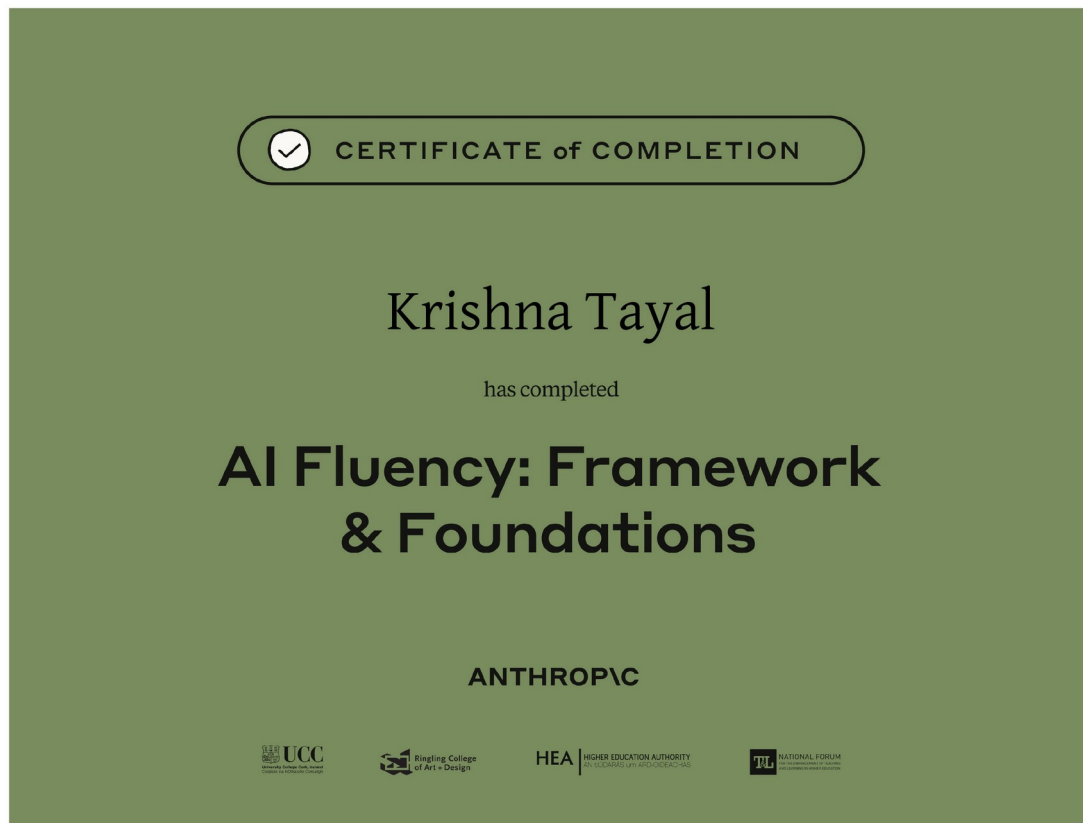
Certificate F.13 - Claude Code 101, issued by Anthropic.

[Open credential verification or evidence record](#)

CERTIFICATE F.14

AI Fluency: Framework & Foundations

Anthropic | July 2026



Certificate F.14 - AI Fluency: Framework & Foundations, issued by Anthropic.

[Open credential verification or evidence record](#)

END OF INTERNSHIP REPORT