AI Masterclass by Finer Circle

This Masterclass is a comprehensive specialisation track designed to take you from **0 to 1 (Literacy)** and **1 to 10 (Mastery)**. We have stitched together three distinct advanced modules into one cohesive journey. You will move from understanding the mathematical foundations of Neural Networks to training your own models, and finally, to building autonomous Agentic workflows that act as your virtual teammates.

Phase 1: The Foundations (AI Literacy, Architecture & Development)

This is the most extensive phase of the program, designed to transform you from a "User" of AI to a "Builder" of AI. Before you can test an AI system, you must understand its anatomy. We strip away the magic and focus on the engineering: from the math of Neural Networks to the architecture of modern GenAI apps. You will build your first models, design application architectures, and master the data intelligence required to fuel them. By the end, you will be able to look at any AI output and explain the *architectural* reasons behind its behaviour.

⏱ **Duration:** 45 Hours

Module 1: The Mechanics of Intelligence

Get under the hood of the "Black Box." We break down the jargon and map out the entire AI landscape.

- **The Learning Hierarchy:** A rigorous breakdown of the relationships between Artificial Intelligence, Machine Learning, Deep Learning, and Generative AI.
- **The Three Learning Paradigms (Deep Dive):**
    - **Supervised Learning:** Training machines using labelled data (Input → Output) for tasks like Classification (Spam vs. Not Spam) and Regression (Predicting values).
    - **Unsupervised Learning:** Finding hidden structures, clusters, and associations in unlabeled data (e.g., Customer Segmentation) without human guidance.
    - **Reinforcement Learning:** Understanding learning via environment interaction using reward/penalty feedback loops (e.g., Robotics, Game AI).

- **Neural Network Architecture:**
    - **Deconstructing the Network:** Understanding the role of Input Layers, Hidden Layers (where the mathematical functions live), and Output Layers.
    - **The Training Loop:** A conceptual walkthrough of how networks "learn" by adjusting Weights and Biases through repeated iterations.
    - **Common Failures:** Understanding critical failure modes like Overfitting (memorising noise), Underfitting (oversimplifying patterns), and Non-determinism (why the same input produces different outputs).

Module 2: Applied Supervised Learning (Hands-On First App)

Your first engineering milestone. You will build and train a functional AI model from scratch.

- **The Exercise:** Building a custom Text Classifier using *MachineLearningForKids* to categorise product reviews (Positive vs. Negative vs. Neutral).
- **Iterative Refinement:** Testing the model with tricky edge cases (e.g., double negatives like "I don't hate this product") and retraining the model to fix classification errors.
- **Performance Metrics (Measuring Success):**
    - **Confusion Matrix:** A deep dive into analysing True Positives, True Negatives, False Positives, and False Negatives.
    - **The Three Metrics:** Calculating Accuracy (Overall correctness), Precision (Trustworthiness of positives), and Recall (Ability to find all positives) to scientifically validate your model.

Module 3: The Language of GenAI (Transformers & Vectors)

How do computers understand text? We explore the architectures that power ChatGPT and other LLMs.

- **Vectorisation & Embeddings:** How unstructured inputs (Text/Images) are converted into multi-dimensional numerical vectors so machines can "read" and process meaning.
- **Transformer Architecture:**
    - **Encoders:** The component that converts human language into vector representations.
    - **Decoders:** The component that converts vector calculations back into human-readable output (Generation).
    - **Attention Layers:** How models use "Vector Distance" to understand context (e.g., associating "Bank" with "Money" vs. "River").

Module 4: GenAI Application Architectures

GenAI is not just "Prompting." We explore the four distinct ways to architect an AI solution.

- **The Evolution of Architecture:**
    1. **Direct Prompting:** Simple Input/Output apps (e.g., ChatGPT wrappers) for rapid prototyping.
    2. **Fine-Tuned Models:** Training specific models for niche domains (Medical/Legal) to change the model's behaviour.
    3. **RAG (Retrieval Augmented Generation):** Connecting models to private data (Vector DBs) to solve knowledge cutoffs without retraining.
    4. **Agentic Architecture:** Systems where AI plans and executes multi-step workflows using tools.
- **Component Deep Dive:** Understanding Vector Databases (Long-term memory), Guardrails (Safety layers), and LLM Ops (Model management).

Module 5: Hands-On: Building a GenAI Application

Put architecture into practice. We build a functional GenAI application from scratch.

- **The Project:** Building a Conversational Agent for Customer Support that can handle email responses autonomously.
- **The Development Lifecycle:**
    - **Problem Definition:** Scoping the AI's role and limitations.
    - **Model Selection:** Choosing between Foundation Models (GPT-4, Llama 3) vs. Fine-Tuned models based on cost/performance.
    - **Development:** Integrating the LLM with an application logic layer.
    - **Feedback Loops:** Implementing mechanisms to capture user corrections and improve the model over time.
- **Integration Challenges:** Handling non-deterministic outputs in a traditional software stack.

Module 6: Data Intelligence for QA

AI is only as good as the data feeding it. We teach you a rigorous framework for assessing data quality.

- **The "Never Trust Data" Principle:** A systematic workflow for QAs: Understand Context → Assess Structure → Check Characteristics → Verify Business Rules.
- **Data Quality Dimensions:**
    - **Completeness:** Identifying and handling missing values (e.g., analyzing blank Status fields in E-commerce data).
    - **Uniqueness:** Validating distinct identifiers (e.g., Transaction IDs) to prevent data duplication.

    - **Validity & Accuracy:** Ensuring formats (Currency/Dates) and values match real-world expectations.
- **Hands-On Analysis:** Deep dive into a raw E-commerce dataset (1,050 rows) to identify gaps, irrelevant columns, and define Service Level Objectives (SLOs).

Phase 2: AI in Testing (The Practitioner's Toolkit)

Theory meets practice. In this phase, you transition from "learning" to "doing." This is where you learn to apply AI to your daily work. We provide a rigorous framework to evaluate AI use cases (so you don't waste time on bad ideas) and then dive into engineering.

⏱ **Duration:** 24 Hours

Module 7: Strategic AI Implementation (The Framework)

Stop guessing. We provide a rigorous framework to decide precisely when to use AI, and when to skip it.

The "Atomic Steps" Methodology: A technique for breaking down complex QA workflows (e.g., Failure Analysis, Bug Triage) into individual atomic actions to pinpoint precisely where AI adds value.

- **The 7-Filter Evaluation Framework:** A rigorous checklist to approve/reject AI use cases:
    - **Desirable:** Does it solve a real problem (Time/Cost/Quality)?
    - **Technically Feasible:** Do the right algorithms/models exist?
    - **Data & Infrastructure:** Is compliant, high-quality data available?
    - **Ethical & Responsible:** Assessing privacy risks and explainability.
    - **Right People:** Does the team have the skills to maintain it?
    - **Governance:** Does it meet GDPR/Policy rules?
    - **Testable:** Can the output be validated efficiently?

Module 8: GenAI Engineering & API Configuration

Get your hands dirty with code. We set up the environment for professional AI development.

- **Environment Setup:** Setting up Python virtual environments, managing dependencies via requirements.txt, and securing API Keys (Grok/OpenAI).
- **Controlling the Brain (LLM Configuration):**
    - **Temperature:** Configuring randomness (0 for deterministic testing vs. 2 for creative variety).

  - **Max Tokens:** Managing output length to prevent hallucinations and rate limit exhaustion.
  - **System vs. User Prompts:** Designing the "Persona" (System Prompt) and the "Task" (User Prompt) for optimal responses.
- **Model Benchmarking:** Strategies for comparing models such as Llama 3 (8B vs 70B), Mixtral, and Gemma to strike a balance between cost and performance.

Module 9: RAG Architecture (Retrieval Augmented Generation)

Solve the "Knowledge Cutoff" problem. Learn to build systems that rely on your private data, not just the model's training.

- **The RAG Concept:** Supplementing LLMs with external, private data (Company Docs, Jira) without the cost of fine-tuning.
- **The RAG Pipeline Deep Dive:**
  - **Ingestion:** Using LangChain loaders to process PDFs, HTML, and JSON documents.
  - **Chunking:** Strategies for splitting text (Recursive Character vs. Sentence) and managing "Overlap" to preserve context.
  - **Embeddings:** Converting chunks into vectors using models like HuggingFace.
  - **Vector Stores:** Storing and retrieving vectors using FAISS (Facebook AI Similarity Search).
- **Testing RAG:** Validating the Retrieval accuracy (Did we get the correct doc?) separately from the Generation quality (Did the AI answer correctly?).

Phase 3: Agentic AI (The Future of Automation)

This is the frontier of AI. You will move beyond simple "Chat" interfaces to building **Autonomous Agents** - AI systems that can plan, reason, use tools, and execute complex workflows without constant supervision. You will create "Crews" of agents that act as researchers, writers, and analysts, and learn how to govern them safely using advanced guardrails.

⏱ **Duration:** 24 Hours

Module 10: The Agentic Framework (CrewAI)

Understanding the shift from linear automation to autonomous reasoning.

- **Agents vs. Automation:** Why "Agents" (who plan and reason) are different from "Scripts" (which simply follow rules).
- **Core Building Blocks (CrewAI):**
  - **Agents:** Defining specific Personas (Role, Goal, Backstory).
  - **Tasks:** Defining distinct Actions, Rules, and Expected Outputs.
  - **Tools:** Empowering Agents with non-AI capabilities (Web Scraping via **SerperDev**, File I/O, Custom Python Scripts).

Module 11: Designing & Building Complex Workflows

Hands-on construction of multi-agent systems that solve real-world problems.

- **Workflow Patterns:** Designing Sequential (A → B → C) and Hierarchical (Manager → Worker) flows.
- **Hands-On Project 1: Content Planner Crew:**
  - **The Goal:** A 3-agent pipeline (Planner → Writer → Editor) that researches topics and produces formatted blog posts.
  - **The Tech:** Using Llama 70B for planning and lower-cost models for writing.
- **Hands-On Project 2: MLB Stats Crew:**
  - **The Goal:** A complex crew (Researcher → Statistician → Writer) that uses tools to scrape real-time game data, calculates statistics, and generates summaries.
  - **Error Handling:** Seeing how Agents behave when data is missing ("hallucinating" game IDs) and how to fix it.
- **Function Calling:** Integrating custom code to allow Agents to perform mathematical calculations or API calls accurately.

Module 12: Governance, Safety & Human-in-the-Loop

AI is powerful, but it needs control. Learn to build safe, reliable, and unbiased systems.

- **Guardrails:** Implementing safety layers using Llama Guard to filter offensive content, PII, and off-topic queries (Input/Output validation).
- **Human-in-the-Loop:** Designing workflows that pause for human feedback or approval (e.g., reviewing an article plan) before proceeding to execution.
- **Bias & Security:** Using tools like IBM AI Fairness 360 for bias detection and understanding security risks like Prompt Injection (unauthorised actions) vs. Jailbreaking (unauthorised information extraction).