



AI-Assisted SDLC & Integration Engineering Bootcamp

Accelerated Future Skills for Structured AI-Enabled Software Development

5-DAY INTENSIVE HANDS-ON PROGRAMME

ENGINEERING-GRADE AI SKILLS

Programme Overview

What It Is

A 5-day intensive bootcamp equipping engineers, developers, fresh graduates and technical professionals with structured AI-enabled software development skills — across the full Software Development Life Cycle (SDLC)

What Makes It Different

Unlike generic AI coding courses focused on fast code generation, this programme emphasises **structured and responsible AI-assisted development** — covering requirements, architecture, testing, integration, security, traceability and deployment within a controlled engineering workflow.

Who Should Attend

Junior developers, fresh graduates, electrical/electronics/automation engineers moving into software, IT support, digital transformation and process improvement teams., Technicians, Process Specialist

How This Programme Differs

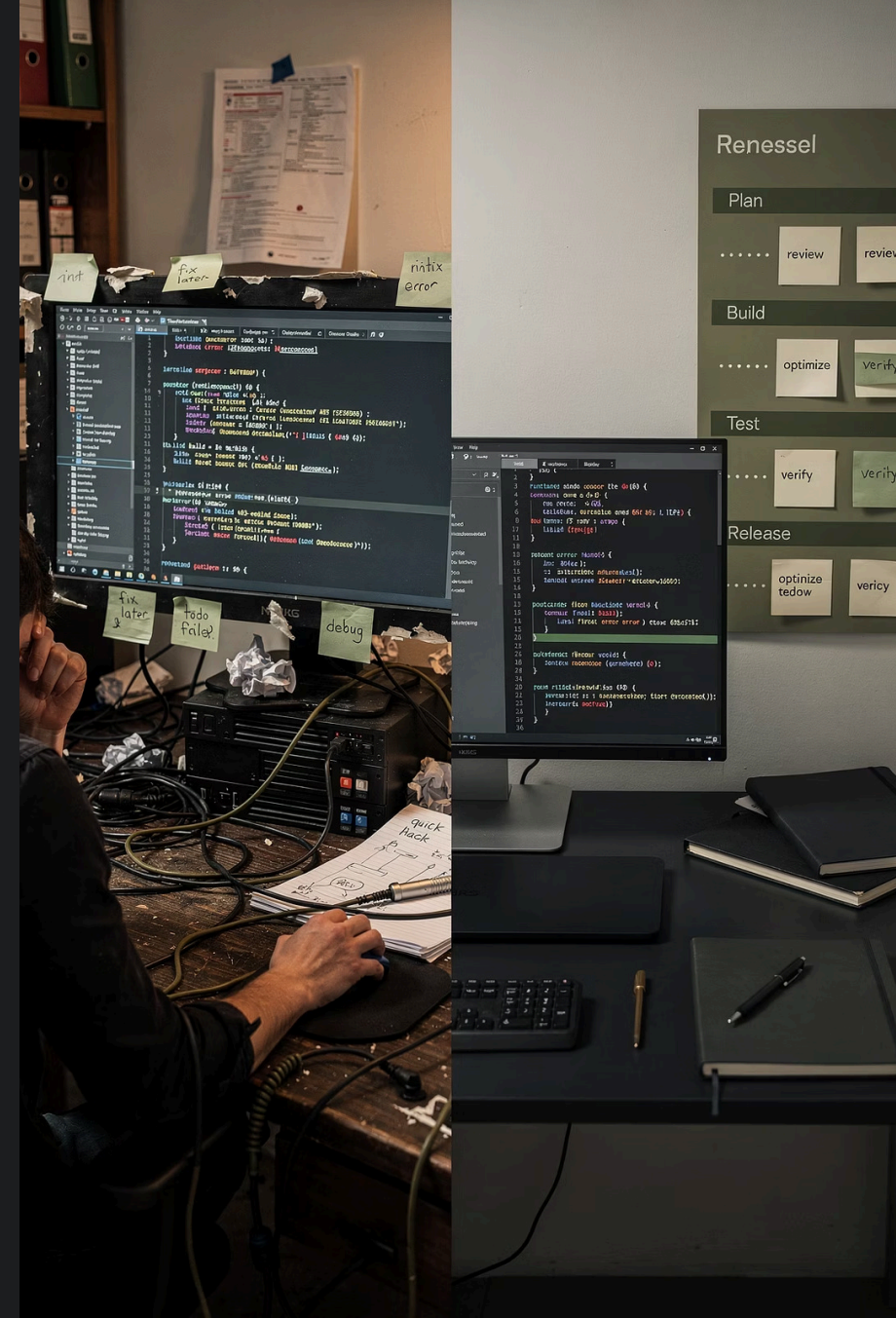
Most market AI coding programmes focus on fast app creation through free-form prompting. This bootcamp takes a fundamentally different approach.

Typical Market Programmes

- Exploratory, prompt-and-generate, trial-and-error
- AI used mainly as a code generator
- Minimal testing, traceability or security
- Outcome: a quick prototype or app demo

This Programme

- Guided, structured and engineering-oriented
- AI used across requirements, design, coding, testing, integration and deployment
- Explicit SDLC stages, traceability, versioning and security
- Outcome: structured prototype with intent, architecture, tests and deployment plan



Programme Learning Outcomes

Participants will leave with 13 practical, future-ready competencies spanning the full AI-assisted development lifecycle.

#	Competency	Description
1	AI Foundations	Explain AI, ML, Deep Learning, Generative AI, LLMs, Agentic AI, RAG and fine-tuning; compare tools used in modern engineering workflows.
2	Structured Development	Convert requirements into structured intent and acceptance criteria; apply structured prompting and intent engineering for controlled AI-assisted development.
3	Testing & Quality	Generate unit tests, functional tests and edge-case scenarios; use validation loops to test, diagnose, fix and improve AI-generated outputs.
4	Integration & Deployment	Connect applications to databases, APIs and automation workflows; apply security, traceability, CI/CD concepts and observability practices.
5	Tool Proficiency	Set up and operate AI coding assistants (e.g. CLINE, Copilot); configure LLM backends and manage AI-assisted dev environments.
6	Intent Engineering	Write structured prompts with clear intent, constraints and acceptance criteria to guide AI code generation reliably.
7	Architecture Thinking	Produce basic architecture diagrams and system designs for AI-assisted applications.
8	Code Review & Validation	Review, validate and improve AI-generated code using structured checklists and human oversight practices.
9	API & Workflow Integration	Integrate AI-assisted applications with external APIs, databases and automation workflows.
10	RAG Implementation	Build knowledge-enabled applications using Retrieval-Augmented Generation techniques.
11	Security & Governance	Apply security checklists, responsible AI practices, privacy controls and change management to AI-assisted projects.
12	Traceability & Versioning	Maintain traceability maps, version control and audit trails across the AI-assisted development lifecycle.
13	Deployment Readiness	Apply CI/CD concepts, observability, monitoring and release readiness practices to AI-assisted software projects.

Pre-Work & Prerequisites

Entry Requirements

- Basic computer literacy
- Basic familiarity with programming concepts
- Basic Python exposure helpful but not mandatory
- Willingness to use AI tools for technical work
- No deep AI development experience required

Required Pre-Work (5–10 hrs)

AI Tool Familiarisation (1–2 hrs)

Explore ChatGPT, Claude, Gemini, Copilot or similar. Try summarising, explaining concepts and generating simple scripts.

Python & SDLC Review (2–4 hrs)

Review Python basics (variables, functions, CSV/JSON) and SDLC stages: requirements, design, implementation, testing, deployment.

Dev Environment Setup (2–4 hrs)

Install VS Code and CLINE (or equivalent AI extension), configure LLM backend/API key, and understand PLAN and ACT workflows.

Day 1: AI Foundations , Tools and technologies for AI-Assisted Development

Learning Objective: Participants will build a strong foundation in AI concepts, tools and technologies required for AI-assisted software development. They will understand the differences between AI, Machine Learning, Generative AI, LLMs, Agentic AI, RAG and fine-tuning, and learn how these technologies support accelerated future skills in software development and integration.

7 hours of structured learning

Time	Topic
9:00 AM – 10:30 AM	AI, ML, Deep Learning, Generative AI, Predictive AI
10:30 AM – 12:00 PM	LLMs, Agentic AI, RAG and Fine-Tuning
12:00 PM – 12:30 PM	AI Tools and technologies for developers and engineers
12:30 PM – 1:30 PM	LUNCH
1:30 PM – 3:00 PM	AI Tools and technologies for developers and engineers (Continue)
3:00 PM – 4:00 PM	AI-Assisted Development & Future SDLC Skills
4:00 PM – 5:00 PM	Lab: Tool Comparison, Environment Check & AI Capability Mapping

Day 1 Deliverables

#	Deliverable
1	AI Tool Comparison Summary
2	Configured Dev Environment
3	AI Landscape Map
4	AI Capability Map

Day 2: Structured Intent & Architecture

Learning Objective: Participants will learn how to convert problems into structured intent, acceptance criteria, architecture and controlled AI-assisted development workflows.

7 hours of structured learning

Time	Topic
9:00 AM – 10:30 AM	From Problem Statement to Structured Intent
10:30 AM – 11:30 AM	Requirements & Acceptance Criteria
11:30 AM – 12:30 AM	Basic Architecture Thinking for AI-Assisted Systems
12:30 PM – 1:30 PM	LUNCH
1:30 PM – 2:45 PM	Structured Prompting & Intent Engineering
2:45 PM – 4:00 PM	Rules, Workflows & Controlled AI Generation
4:00 PM – 5:00 PM	AI-Assisted Code Generation using Templates

Day 2 Deliverables

#	Deliverable
1	Structured Intent Document
2	Acceptance Criteria Sheet
3	Basic Architecture Diagram
4	Initial Code Scaffold

Day 3: AI-Assisted Testing, Validation & Quality Engineering

Learning Objective: Participants will learn how to use AI to generate tests, expand coverage, model scenarios, validate system behaviour and improve software quality.

7 hours of structured learning.

Time	Topic
9:00 AM – 10:30 AM	Test-First Ai Assisted Development
10:30 AM – 11:30 PM	Unit Test Generations and Edge Cases
11:30 PM – 12:30 PM	Scenario Modelling & Functional Testing
12:30 PM – 1:30 PM	LUNCH
1:30 PM – 2:45 PM	Validation Loops: Generate, Test, Diagnose & Fix
2:45 PM – 4:00 PM	Resilience & Failure Path and Security testing
4:00 PM – 5:00 PM	Quality Review and test coverage improvement

Day 3 Deliverables

#	Deliverable
1	Unit Test Suite
2	Functional scenario list and test cases
3	Validation Loop Record
4	Quality review checklist
5	Improved Code Components

Day 4: Integration, Data, APIs, RAG, Versioning, Traceability & Security

Learning Objective: Participants will learn how to connect AI-assisted software components with data, databases, APIs and external workflows while maintaining traceability, versioning and security controls.

Time	Topic
9:00 AM – 10:00 AM	Data & Database Handling for AI-Assisted Applications
10:00 AM – 11:00 AM	API Integration & Workflow Connectivity
11:300 AM – 12:30 PM	RAG: Knowledge-Enabled Applications
12:30 PM – 1:30 PM	LUNCH
1:30 PM – 2:45 PM	Versioning, Traceability & Change Management
2:45 PM – 3:45 PM	Security, Privacy & Responsible AI Practices
3:45 PM – 5:00 PM	Integration Readiness Review & Capstone Build Planning

Day 4 Deliverables

#	Deliverable
1	Data, Database Structrure and Integration Component
2	API Integration Flow
3	RAG Component (if applicable)
4	Security & Privacy Checklist
5	Traceability Map
6	Final Capstone Build Plan

Day 5: Deployment, CI/CD, Observability, Capstone Build & Demonstration

Learning Objective: Participants will consolidate all learning by completing, testing, deploying or preparing deployment for an AI-assisted software or integration prototype.

Time	Topic
9:00 AM – 10:00 AM	Deployment & CI/CD Concepts
10:00 AM – 11:00 AM	Infrastructure, Environment & Release Readiness
11:00 AM – 12:30 PM	Observability, Monitoring & AI Feedback Loops
12:30 PM – 1:30 PM	LUNCH
1:30 PM – 3:30 PM	Capstone Build Sprint
3:30 PM – 4:30 PM	Final Presentation & Demonstration
4:30 PM – 5:00 PM	Feedback, Reflection & Future Skills Roadmap

Day 5 Deliverables

#	Deliverable
1	Completed Capstone Prototype or Semi working Prototype
2	Structured Intent Document
3	Architecture Diagram
4	Test Suite and validation notes, Data/API Integration
5	Deployment Plan
6	Future Improvement Roadmap

Capstone Project & Programme Summary

Capstone Requirements

Each team builds a prototype including at least **6 of 13 components**: structured intent, acceptance criteria, AI-generated code, unit/functional tests, validation loop evidence, data/API integration, RAG component, UI/workflow trigger, security checklist, traceability map, deployment plan, observability loop, and human review point.

Evaluation Criteria

#	Criteria	Weight
1	Problem Clarity & Requirements	15%
2	AI-Assisted Dev Workflow	15%
3	Test Generation & Validation	15%
4	Technical Integration	15%
5	Security, Traceability & Governance	10%
6	Presentation & Communication	30%
	Total	100%

Programme Promise

This programme prepares participants to become effective AI-enabled engineers who can direct, test, validate, integrate and operationalise AI-assisted development outputs – with quality, reliability, governance and human oversight.

- ✔ Participants leave ready for real-world AI-enabled engineering, software and digital transformation roles.



www.eliteindigo.com



Scan to register /
contact us

WhatsApp

6012 705 5594
(Aisha/Azli)