ACADEMIC TRAINING PROPOSAL / 6-DAY COMPLETE PROGRAM

AI Agents & Automation

Design, connect and test useful AI-assisted workflows

DURATION	TRAINING	APPLICATION
6 days · 36 hours	**18 hours · mornings**	**18 hours · afternoons**

A practical learning opportunity

Students encounter AI assistants every day, but many have not yet learned to turn a task into a dependable process. This program bridges that gap. Learners work through a small workflow from problem definition and prompt design to integration, testing and human oversight. The emphasis is on the decisions that make an automation genuinely useful—not on collecting tools or flashy demos.

The full program gives learners time to make mistakes, investigate them and improve a working system. Starting with agent design and grounded answers, participants add integrations, approvals and practical tests before presenting a documented automation. The final output is a defensible prototype rather than a black-box demonstration.

What participants will take away

FINAL PARTICIPANT OUTCOME

A documented AI-assisted workflow that combines bounded knowledge or tools, at least one integration, repeatable tests, a human checkpoint and an operator guide.

Who this is designed for

SUITABLE PARTICIPANTS	STARTING POINT
Computer science, IT, engineering, innovation clubs, and students interested in practical workflow building.	Comfort using browser-based tools, files and spreadsheets. Coding is not required for the guided no-code route.

Why it matters to the department

Fits technical skill-development, innovation-cell activities and applied computing initiatives. Student prototypes can seed mini-projects or practical demonstrations without requiring production-system access.

This proposal is intended for academic review. Cohort-specific examples and delivery arrangements can be agreed with the department once there is interest.

02 / ACADEMIC DESIGN

Learning outcomes and evidence

The outcomes below describe work a participant can show at the end of the program. They are deliberately scoped to the available instructional time.

PARTICIPANTS WILL BE ABLE TO...	OBSERVABLE EVIDENCE
Choose suitable agent patterns	A task analysis showing why an agent or conventional automation fits.
Design bounded instructions and knowledge	A documented assistant evaluated against prepared source questions.
Build event-driven workflows	A testable trigger-to-output flow with structured fields.
Connect tools responsibly	An integration diagram including access and approval boundaries.
Test reliability and handle failure	A test log covering normal, ambiguous and failing scenarios.
Present and hand over a prototype	A demonstration supported by workflow map and runbook.

How the workshop runs

SESSION	TEACHING APPROACH	PARTICIPANT EXPERIENCE
Morning · 3 hours	Concepts, worked examples, instructor demonstrations and guided questioning.	Understand the method, challenge assumptions and see a complete example.
Afternoon · 3 hours	Hands-on labs, structured iteration, peer critique and feedback.	Apply the method to a case, improve an artifact and leave with evidence of learning.

Illustrative daily timing: 09:30–12:30 training; 12:30–13:30 lunch (outside contact hours); 13:30–16:30 practice. The essential commitment is six instructional hours per day, split equally between learning and application.

What we will not overpromise

Depth includes grounded assistants, connectors and evaluation, but does not promise production deployment or access to restricted institutional systems. Multi-agent design is introduced conceptually where useful, not treated as a compulsory build.

03 / SINGLE-PAGE TIMETABLE

6-day teaching and practice schedule

Each day deliberately links the morning lesson to an afternoon artifact. The complete timetable is kept together on this page for ease of academic review.

DAY & FOCUS	MORNING TRAINING · 3 HOURS	AFTERNOON PRACTICE · 3 HOURS	DAILY EVIDENCE
DAY 01 **Discover the right workflow**	Agent patterns, use-case selection, boundaries, inputs/outputs and human oversight.	Interview a mock user, map a low-risk process and write assistant success criteria.	Workflow map + specification
DAY 02 **Create a grounded assistant**	Prompt structure, approved reference material, retrieval concepts and evaluation questions.	Build a reference-aware assistant and test answer quality against a short source pack.	Assistant + 10-case check
DAY 03 **Automate predictable steps**	No-code triggers, actions, structured data, branching and simple logs.	Connect a form/sheet event to processing and a reviewed output with synthetic data.	End-to-end workflow
DAY 04 **Add tools and approvals**	API concepts, permissions, connector limits, exceptions and human-in-the-loop design.	Extend the workflow with a guided data lookup or API, plus an approval checkpoint.	Integrated prototype
DAY 05 **Make the system trustworthy**	Failure handling, evaluation sets, prompt injection risks, monitoring and fallback design.	Test edge cases, document errors and improve the workflow using a small rubric.	Failure-test report
DAY 06 **Demonstrate and hand over**	Operational boundaries, documentation, handover and deployment choices.	Complete capstone, run a live demonstration and produce an operator-ready guide.	Capstone + runbook

Daily learning rhythm

Explore the idea → see a worked demonstration → practise with feedback → check the output → record what improved. The afternoon closes with an artifact, not only a completed exercise.

Progression across the week

Days 1–2 develop the foundation; Days 3–4 extend it through guided application; Day 5 strengthens quality and judgment; Day 6 integrates the learning into a demonstrable capstone. Individual exercises may be completed in teams if appropriate to the cohort.

04 / LEARNING BY DOING

Hands-on studio: what students will actually do

The following activities show the level of work expected in the afternoon sessions. Case material and examples can be aligned with the department while preserving the core learning outcomes.

01 Source-grounded question answering

Students receive a compact, approved policy or product information pack. They build an assistant that distinguishes supported answers from unknowns, then score ten responses for completeness and faithfulness.

Evidence: prompt specification, source pack and evaluation sheet.

02 A real workflow, safely simulated

A mock enquiry or event-registration record triggers extraction and classification. A human approves the generated response before any external action. Students follow every field through the flow.

Evidence: workflow diagram, sample executions and approval rule.

03 Integrations without blind trust

Groups add a structured lookup or API call and intentionally provoke one permission, timeout or formatting problem. They decide which failures should trigger retry, review or halt.

Evidence: integration notes and exception handling.

04 Reliability clinic

Using a shared test set, students compare initial and revised behavior, check claims against a source, and record what the automation should not be trusted to do.

Evidence: before/after test log and runbook.

Capstone brief

END-OF-PROGRAM CHALLENGE

A campus-service or small-business workflow built with synthetic data: user input, one or more AI-assisted steps, a connected tool, a human review gate and traceable outputs. Teams submit a workflow map, a minimum eight-case test log and a one-page operator guide.

Examples of suitable practice contexts

- Categorise enquiries from a sample spreadsheet and prepare draft replies for approval.
- Answer common questions from an approved, short reference document, while showing the source passage.
- Take a simulated form submission, prepare a structured summary and route it to a human reviewer.

05 / EVIDENCE AND ACADEMIC FIT

Assessment that a department can inspect

Assessment is practical and formative. Facilitators look at the quality of the work and the reasoning behind it, not just whether a tool produced an answer. The weights are suggested for workshop evaluation and can be adapted to the cohort.

CRITERION	WEIGHT	WHAT REVIEWERS LOOK FOR
Workflow design and task fit	20%	Justification, clear boundaries and useful outputs
Implementation and integration	30%	Demonstrably working workflow and handoff
Reliability and responsible design	30%	Structured testing, error response, source checking and approvals
Documentation and communication	20%	Diagram, operator guide and live explanation

Evidence retained by learners

- Completed daily artifacts or short case submissions.
- A final project, demonstration or decision portfolio appropriate to the program.
- A brief explanation of what was checked, improved and left unresolved.

Tools, access and responsible practice

An institution-approved AI interface; one accessible no-code automation platform; forms, spreadsheets and approved sample documents. APIs or coding extensions are optional, subject to access.

Use synthetic or approved sample data. No live student records, unattended external messages or production credentials are needed. Every workflow includes a human review or stop condition.

Institutional value

Fits technical skill-development, innovation-cell activities and applied computing initiatives. Student prototypes can seed mini-projects or practical demonstrations without requiring production-system access.

Suggested academic discussion

A short conversation with the relevant department can confirm the student level, preferred case domain and expected final artifacts. That alignment is enough to decide whether the program fits the institution’s learning priorities.