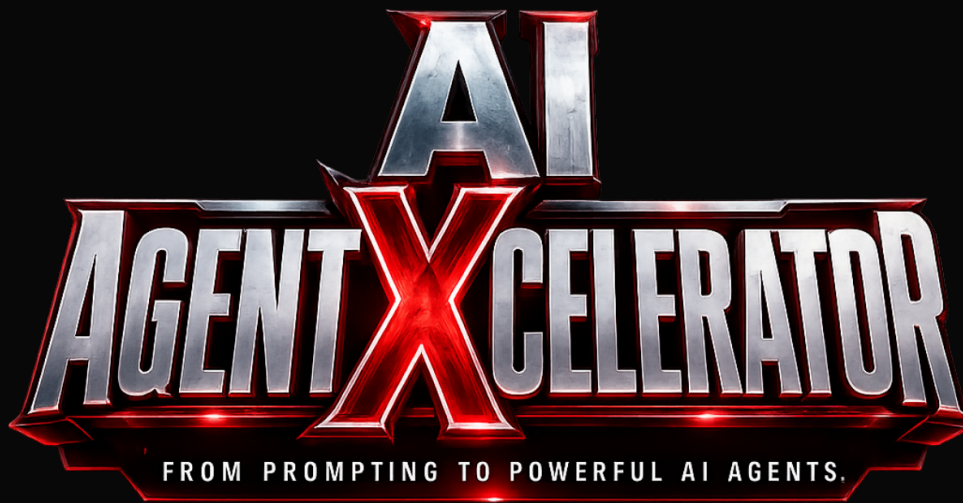




AI AGENT XCELLERATOR

21-Day Agentic AI Product Bootcamp

Build. Deploy. Automate. Sell.

FROM PROMPTING TO POWERFUL AI AGENTS

Organized By:	SIN School of AI
Duration:	21 Days Intensive
Program Type:	Agentic AI, Full-Stack Product Development, Automation & Sales Execution
Program Brand:	AI Agent Xcellerator Program
Learning Model:	65% Practical 20% Strategy & Theory 15% AI Workflow & Review
Outcomes:	Deployed AI Products, RAG Agents, n8n Workflows, Mobile App, GTM Assets & Capstone Defense

Program Positioning

The AI Agent Xcellerator Program is designed as a 21-day intensive, product-building bootcamp for students who need more than prompt engineering. The program trains learners to build, deploy, automate, analyse, and sell agentic AI products using one modern production stack instead of jumping randomly across tools.

The program is positioned around a blunt but important standard: students do not pass because they watched AI demos. They pass when they can ship working agentic products, explain the architecture, secure the data layer, automate a workflow, demonstrate a mobile/web experience, and defend the commercial use case.

Important Rule: AI tools are allowed for speed, but every AI-generated output must be verified. Blind copy-paste from AI tools, untested agent actions, hallucinated RAG answers, and unverified client outreach are treated as workflow failures, not skills.

Learning Model	Meaning inside the Program
65% Practical Implementation	Daily labs, working code, deployed products, RAG pipelines, n8n workflows, mobile companion screens, GitHub repositories, demos, and capstone build reviews.
20% Strategy, Architecture & Theory	Agentic AI concepts, workflow redesign, tool calling, RAG design, security, product thinking, business intelligence, big data fundamentals, and deployment discipline.
15% AI Workflow, Business Communication & Review	Plan-first AI coding, prompt discipline, testing, AI usage audit, pitch narrative, outreach assets, portfolio packaging, and technical viva.

Mandatory Prework

- Git and GitHub basics: repositories, commits, branches, pull requests, README structure.
- Command-line basics: folders, navigation, package installation, environment variables.
- JavaScript/TypeScript fundamentals: functions, objects, async/await, imports, JSON.
- Python syntax: variables, loops, functions, modules, file handling, virtual environments.
- HTTP/API basics: GET/POST requests, headers, JSON payloads, status codes, API keys.
- SQL fundamentals: tables, rows, joins, filters, simple aggregation, and data integrity.

Final Student Output

- | | |
|---|--|
| <ul style="list-style-type: none"> • 1 agent strategy memo • 1 Python CLI research helper • 1 Gradio demo and API client script • 1 Next.js/Supabase SaaS skeleton • 1 secured database with RLS and Edge Function • 1 coding-agent comparison log • 1 ingest-and-query RAG pipeline | <ul style="list-style-type: none"> • 1 deployed RAG assistant with source references • 1 MCP-connected tool and multi-agent delegation demo • 1 n8n business workflow automation • 1 BI/data-agent design and big-data architecture sketch • 1 Expo/React Native mobile companion • 1 sales, marketing, outreach, and GTM pack • 1 final capstone product, pitch deck, demo video, handoff docs, and portfolio case study |
|---|--|

AI Agentic Development Tool Stack

The stack is intentionally opinionated. The program is not a random survey of every AI tool. It focuses on a production-ready path that students can actually build with in 21 days.

Tool Category	Tools / Platforms	How Students Use Them
Core Web Product Stack	Next.js, React, TypeScript, Vercel AI SDK, Vercel	Build full-stack AI applications, route handlers, server functions, dashboards, authenticated workspaces, and AI-powered interfaces.
Backend, Data & Auth	Supabase, Postgres, Auth, Storage, Realtime, Edge Functions, pgvector	Create secure user-scoped data models, APIs, vector storage, RAG permissions, real-time events, and production-ready backend foundations.
Agentic Coding Assistants	Cursor, Freebuff, Google Antigravity, GitHub Copilot, ChatGPT, Claude, Gemini	Plan features, generate draft code, compare agent outputs, debug errors, write tests, produce documentation, and verify changes manually.
Python & Fast Demo Layer	Python, Gradio, REST APIs, Python client scripts	Build CLI helpers, wrap functions into shareable demos, expose simple APIs, and prototype intelligence before integrating into full-stack apps.
RAG & Knowledge Agents	Supabase Vector, hybrid search, embeddings, LangChain/LlamaIndex awareness	Ingest documents, retrieve context, cite sources, enforce permissions, and build RAG assistants for business use cases.
Tool Calling & Interoperability	Typed tool schemas, MCP, A2A, webhooks, API connectors	Connect agents to external tools, control multi-step loops, validate inputs, and demonstrate multi-agent delegation.
Automation & Operations	n8n, HTTP nodes, triggers, custom code, cron, email/CRM/webhook workflows	Automate lead enrichment, support workflows, research ops, internal notifications, and business process execution.
BI, Data Agents & Big Data	Microsoft Fabric, Power BI Copilot, data agents, lakehouse concepts	Design governed analytics, conversational data workflows, KPI definitions, and big-data architecture sketches.
Mobile Companion Stack	Expo, React Native, Supabase React Native, EAS	Create mobile companion apps that reuse the same backend and consume agent/RAG endpoints.
Presentation & Documentation AI	Canva AI, Gamma, Tome, Notion AI, ChatGPT/Claude/Gemini	Create executive decks, README files, demo scripts, handoff documents, outreach content, and portfolio case studies.

AI Ethics, Verification & Security Policy

- Every AI-generated code block must be executed, tested, and explained by the student.
- Every tool call must use typed schemas, explicit inputs, bounded loops, and safe failure behavior.
- Every RAG answer must be backed by retrieval evidence, citations, or source references.
- Every database feature must respect authentication, row-level security, and environment-secret hygiene.
- Every external outreach or client-facing message must be human-verified before use.
- Every final project must include an AI Usage Audit Sheet: tool used, prompt/task, output received, verification performed, correction made, and final decision.

21-Day Milestone Roadmap

Phase	Days	Milestone	Final Output
Phase 1	Day 1-3	Agentic AI Foundations, Python, Gradio & Toolchain Setup	Agent strategy memo, Python CLI helper, Gradio demo, client script, working local development environment
Phase 2	Day 4-7	Full-Stack Agentic App Foundation	Next.js app skeleton, Supabase auth/CRUD, Edge Function, RLS security review, coding-agent comparison log
Phase 3	Day 8-10	RAG Agents, Tool Calling & Interoperability	Document ingestion pipeline, deployed RAG assistant, MCP-connected tool, multi-agent delegation demo
Phase 4	Day 11-14	Automation, BI, Big Data & Mobile Integration	n8n workflow, BI/data-agent design, big-data architecture sketch, Expo mobile companion
Phase 5	Day 15-17	Sales, Marketing & Business Creation	ICP, lead-scoring rubric, sales-research agent brief, campaign pack, product strategy document, commercial offer
Phase 6	Day 18-19	Deployment, Release Discipline & Client Outreach	Staging deployment, production checklist, release runbook, outreach list, discovery script, target-account notes
Phase 7	Day 20	Ideathon + Agentic Product Hackathon	Hackathon prototype, pitch deck, architecture board, data plan, automation flow, demo narrative
Phase 8	Day 21	Capstone Defense, Client Selling & Portfolio Packaging	Deployed capstone, demo video, outreach assets, handoff docs, portfolio case study, individual viva

Detailed 21-Day Curriculum

Day	Focus Area	Practical Work	Deliverable	AI/Agent Tool Layer
1	Agentic AI Foundations & Market Reality	Difference between generative AI and agentic systems; workflow redesign; risk, governance, and product mindset; install core stack.	Agent strategy memo + working local toolchain	Cursor, Freebuff, Antigravity, GitHub, LLM assistant
2	Python Core for Agent Builders	Variables, functions, loops, dictionaries, modules, files, API calls, JSON parsing, structured logs.	CLI research helper in Python	ChatGPT/Claude/Gemini for explanation; manual code testing
3	Gradio Demo Layer	Wrap Python helper into a Gradio interface, publish/share internally, call it through a Python client script.	Gradio demo + client script	Gradio, Python client, API-style invocation
4	Next.js Product Shell	App Router, layouts, navigation, server/client components, landing page, dashboard, workspace area.	Base Next.js app skeleton	Cursor planning, TypeScript scaffolding, Vercel preview
5	API Connectivity & Supabase Auth	Route handlers, server functions, REST/GraphQL concepts, login, user-scoped CRUD around leads/tickets/knowledge items.	Authenticated full-stack CRUD app	Next.js, Supabase, Vercel AI SDK basics
6	Supabase Backend Security	Postgres data model, Storage, Realtime, Edge Functions, webhooks, RLS policies, security review.	Secure multi-user backend + Edge Function	Supabase RLS, environment variables, API secrets
7	Coding-Agent Workflow Comparison	Run the same feature task in Cursor, Freebuff, and Antigravity; compare context handling, speed, testability, and failure modes.	Tool decision log + polished week-one app	Cursor rules, Freebuff agents, Antigravity artifacts
8	RAG Fundamentals	Embeddings, vector database, retrieval, chunking, hybrid search, permissions-aware RAG, ingest pipeline.	Ingest-and-query pipeline	Supabase Vector, pgvector, hybrid search
9	First Real RAG Agent	Build a web-based RAG assistant with retrieval, tool calling, source references, schema discipline, and step limits.	Deployed RAG assistant for a vertical use case	Vercel AI SDK, typed tool schemas, source references
10	MCP, A2A & Tool Interoperability	Connect an external tool through MCP; build a simple multi-agent delegation demo; review security concerns.	MCP-connected tool + multi-agent demo	MCP, A2A concepts, Cursor/Antigravity integration
11	n8n Workflow Automation	Triggers, HTTP nodes, custom code, AI step, data enrichment, write-back to app database or CRM-style table.	Working n8n sales/support/research ops workflow	n8n AI nodes, webhooks, Supabase/API integration
12	Business Intelligence with Data Agents	Fabric concepts, data-agent thinking, conversational analytics, governed metrics, operations monitoring.	BI design + mock data-agent interaction flow	Microsoft Fabric, Power BI Copilot/data-agent concepts

Day	Focus Area	Practical Work	Deliverable	AI/Agent Tool Layer
13	Big Data Fundamentals for AI Products	Big data Vs, batch vs stream, lakehouse, distributed workloads, data readiness for AI products.	Big-data architecture sketch	Lakehouse thinking, Fabric/Azure-style architecture
14	Mobile Companion Development	Expo/React Native app, Supabase auth, consume one agent/RAG endpoint from the web stack.	Mobile companion with sign-in + one agent interaction	Expo, React Native, Supabase, EAS basics
15	Sales Systems with Agentic AI	ICP definition, lead qualification rubric, sales-research agent brief, CRM-style workflow, human validation.	ICP + lead scoring logic + sales-research agent brief	Agent-assisted lead research; manual claim verification
16	Marketing Systems & Brand Execution	Campaign engine, offer positioning, landing-page copy, outbound content, performance view.	Campaign pack + agent-assisted content workflow	ChatGPT/Claude/Gemini, Canva/Gamma, content QA
17	Business Creation & Product Strategy	Problem selection, product wedge, pricing logic, ROI framing, governance choices, customer journey.	Product strategy document + commercial offer	AI-assisted strategy drafting; human market judgment
18	Deployment & Production Hardening	Vercel deployment, Docker/Node hosting discussion, Supabase production checklist, n8n deployment, secrets.	Staging deployment + release runbook	Vercel, Supabase, n8n, Expo EAS deployment thinking
19	Client Outreach Sprint	Lead list, personalised outreach, discovery-call assets, target-account notes, manual validation of claims.	Outreach list + sequence + discovery script	Agent-assisted research; human approval before sending
20	Ideathon + Agentic Hackathon	Team brief to prototype: problem, architecture, data plan, automation flow, UI, deployment, pitch narrative.	Hackathon prototype + pitch deck + architecture board	Full stack, RAG, n8n, mobile, BI, AI deck tools
21	Final Demo, Selling & Portfolio Packaging	Live deployed demo, security explanation, web/mobile/automation/analytics demonstration, pricing and roadmap.	Deployed capstone + demo video + handoff docs + portfolio case study	Final AI usage audit + technical viva

Challenge Structure: Solo vs Team

Blunt Rule: not every challenge should be team-based. The first technical challenge must stay solo. If everything is team-based, weak students can hide while strong students carry the build. Team challenges begin only after individual fundamentals are tested.

Challenge	Mode	Total Points	Reason for Mode
Challenge 1: Agentic App Foundation Sprint	Solo	100	Tests individual ability to set up the app, backend, security basics, and agent-assisted coding workflow. No hiding inside teams.
Challenge 2: RAG Agent Build Battle	Team of 2	100	RAG work benefits from pairing retrieval engineering with product UX while keeping accountability high.
Challenge 3: Automation + Mobile Integration Challenge	Team of 3	150	Integration work needs app, automation, data, and mobile coordination similar to real delivery squads.
Challenge 4: Build-and-Sell Agentic Hackathon	Team of 3-4	200	The final product sprint requires architecture, build, automation, GTM, and pitch execution.
Final Capstone Defense	Hybrid: Team Product + Individual Viva	300	Teams can build together, but each student must individually defend code, architecture, agent logic, AI usage, and business case.

Challenge 1: Agentic App Foundation Sprint

Mode: Solo Individual Sprint

Objective: Build a secure app foundation and prove that the student can use AI coding agents without losing control of the codebase.

AI Rule: AI may assist planning, debugging, and boilerplate generation. The student must explain the code, tests, security choices, and corrections made.

Deliverables:

- Next.js app shell
- Supabase auth and user-scoped CRUD
- RLS policy screenshot or policy notes
- One Edge Function or API route
- Cursor/Freebuff/Antigravity comparison log
- Short security review

Evaluation Area	Points
Next.js structure and route design	16
Supabase data model and auth	18
RLS/security implementation	16
API route or Edge Function correctness	14
Agent-assisted workflow transparency	12
Code readability and basic testing	12
Documentation and explanation	12

Challenge 2: RAG Agent Build Battle

Mode: Team of 2

Objective: Turn a document or business knowledge base into a working RAG assistant with retrieval, source references, and controlled tool behavior.

AI Rule: AI may help write ingestion scripts and UI copy. Retrieval accuracy, source references, and hallucination controls must be manually tested.

Deliverables:

- Document ingestion pipeline
- Vector/hybrid search setup
- RAG assistant UI
- At least 10 test questions with expected evidence
- Source-reference or retrieval log
- Short failure-mode report

Evaluation Area	Points
Document ingestion and chunking quality	16
Vector/hybrid retrieval relevance	18
Agent answer quality and citation discipline	18
Tool schema and step-control design	14
UI clarity and usability	12
Testing and failure-mode analysis	12
Team explanation and demo quality	10

Challenge 3: Automation + Mobile Integration Challenge

Mode: Team of 3

Objective: Connect the web app, automation workflow, and mobile companion into one coherent agentic system.

AI Rule: AI may assist integration debugging and content drafting. The team must verify API calls, workflow triggers, mobile auth, and data write-back manually.

Deliverables:

- n8n workflow with trigger, AI step, enrichment, and write-back
- Expo/React Native mobile companion screen
- Supabase-backed agent/RAG endpoint
- Architecture diagram
- Integration test checklist
- AI usage and verification log

Evaluation Area	Points
n8n workflow correctness	24
API and write-back reliability	20
Mobile companion functionality	22
Architecture clarity	18
Security and secret handling	16
Testing discipline	20
Team coordination and demo	30

Challenge 4: Build-and-Sell Agentic Hackathon

Mode: Team of 3-4

Objective: Solve a real business brief by building a scoped agentic product, deployment plan, automation flow, pitch narrative, and GTM asset pack.

AI Rule: AI can accelerate code, deck writing, and outreach drafts. It cannot invent market claims, fake metrics, or unverified client statements.

Deliverables:

- Deployed prototype
- RAG/agent/tool-calling component
- Automation workflow
- Data/BI or mobile component
- Pitch deck and architecture board
- ICP and outreach sequence
- Release checklist and handoff notes

Evaluation Area	Points
Problem framing and product wedge	24
Engineering completeness	35
Agent/RAG/tool quality	32
Automation and integration depth	26
Deployment and release readiness	24
Commercial clarity and GTM assets	29
Pitch, demo, and Q&A; defense	30

Final Capstone Evaluation

Mode: Hybrid - Team Product + Individual Viva

The final capstone is built as a team agentic product, but each student is evaluated individually through code explanation, architecture explanation, agent behavior explanation, AI usage explanation, contribution defense, and business-case viva.

Evaluation Area	Points
Problem clarity, user journey, and product wedge	25
Full-stack app structure and backend architecture	35
Supabase data model, RLS, auth, and secret hygiene	30
RAG quality, retrieval evidence, and answer reliability	35
Tool calling, MCP/A2A awareness, and agent-control design	30
n8n automation workflow and business-process integration	25
Mobile companion or responsive mobile path	20
BI/data-agent design, KPI logic, or big-data architecture	25
Deployment, release runbook, and handoff documentation	25
GTM plan, ICP, outreach assets, pricing, and ROI argument	25
AI usage audit, responsible workflow, and verification	25
Final presentation, demo quality, and individual technical viva	60

Total Scoring System

Component	Points
Challenge 1: Agentic App Foundation Sprint	100
Challenge 2: RAG Agent Build Battle	100
Challenge 3: Automation + Mobile Integration Challenge	150
Challenge 4: Build-and-Sell Agentic Hackathon	200
Final Capstone Defense	300
Total	850 Points

Top 12 Industry-Standard Updates Added

1. **Agentic AI Foundations Beyond Prompting:** students learn planning, tool use, autonomy limits, workflow redesign, and governance.
2. **Production Full-Stack Stack:** Next.js, Supabase, Vercel AI SDK, and Vercel are used as the core build path.
3. **Secure Supabase Backend:** Auth, Postgres, Edge Functions, and row-level security are treated as required product skills.
4. **RAG with Permissions and Hybrid Search:** retrieval is built with evidence, source references, and security thinking.
5. **Tool Calling and Typed Schemas:** agents are designed with controlled inputs, step limits, validation, and safe failure behavior.
6. **MCP and A2A Awareness:** students understand modern interoperability patterns for tools and multi-agent systems.
7. **n8n Business Automation:** workflows connect AI to operational triggers, enrichment, routing, and write-back actions.
8. **BI and Data-Agent Thinking:** analytics is taught as governed conversational decision support, not only dashboards.
9. **Expo Mobile Companion:** students extend the same agentic system into a mobile interface without native-platform overload.
10. **Deployment and Release Discipline:** staging, production checklist, secrets, runbooks, and handoff documents are mandatory.
11. **Client Outreach and GTM Execution:** the final week includes ICP, lead research, outreach, pitch, pricing, and ROI framing.
12. **AI Usage Audit and Viva:** students must show how AI was used, verified, corrected, and defended individually.

Certification Eligibility

- Minimum 90% attendance.
- Completion of all core assignments and daily deliverables.
- Submission of all challenge deliverables.
- Working GitHub repository with README, screenshots, setup instructions, and environment notes.
- Deployed final product or reviewable staging link.
- AI usage audit sheet submission.
- Final capstone presentation and demo.
- Individual technical viva clearance.

Final Program Outcome

By the end of 21 days, students will not simply claim that they learned Agentic AI. They will have a proof-of-work portfolio containing a full-stack AI product, secure backend, RAG assistant, automation workflow, mobile companion, BI/data-agent design, deployment runbook, sales and marketing assets, AI usage audit, and a final capstone project that can be shown to institutions, recruiters, founders, and industry mentors.

Reference Notes for AI/Agent Tool Layer

- Harvard Online - Agentic AI Foundations: <https://harvardonline.harvard.edu/course/agentic-ai-foundations>
- Stanford HAI - AI Index 2026: <https://hai.stanford.edu/ai-index/2026-ai-index-report>
- Microsoft Learn - AI Agents for Beginners: <https://learn.microsoft.com/en-us/shows/ai-agents-for-beginners/>
- Next.js Documentation: <https://nextjs.org/docs>
- Supabase AI and Vector Documentation: <https://supabase.com/docs/guides/ai>
- Vercel AI SDK Agent Guide: <https://vercel.com/kb/guide/how-to-build-ai-agents-with-vercel-and-the-ai-sdk>
- n8n Documentation: <https://docs.n8n.io/>
- Expo Documentation: <https://docs.expo.dev/>
- Gradio Quickstart: <https://www.gradio.app/guides/quickstart>
- Model Context Protocol: <https://modelcontextprotocol.io/docs/getting-started/intro>
- Microsoft Fabric Data Agents: <https://learn.microsoft.com/en-us/fabric/data-science/concept-data-agent>

Document Review & Update Rule

Agentic AI tools change fast. This syllabus should be reviewed quarterly. The stable foundation is the build discipline: full-stack product delivery, secure data handling, verified agent behavior, deployment readiness, and commercial execution. Tools may change; the standard does not.

AI Agent Xcellerator Program

Build. Deploy. Automate. Sell.

Organized By: SIN School of AI