innovate with ajay

AI CLASSROOM COURSES · UDAIPUR

Five courses.
One practical way to learn.

Offline, mentor-led courses in websites, mobile apps, full-stack software, AI coding agents and digital marketing — each built around real projects.

Courses at a glance

Course	Duration	Hours	Modules	Fee	Starts
Website Development with AI	3 months	120	12	₹13,000	16 Nov 2026
Mobile App Development with AI	3 months	120	12	₹22,000	19 Nov 2026
Full Stack Development with AI	6 months	240	24	₹44,000	22 Nov 2026
AI Coding Agent Development	6 months	300	12	₹58,000	25 Nov 2026
Digital Marketing with AI	45 days	66	7	₹12,000	28 Nov 2026

Each course follows on the next pages: who it is for, what you build, tools, every module and what it covers, and batch details.

AI CLASSROOM COURSES · BEGINNER PATHWAY

Website Development with AI

Design, develop and launch your first professional website — with AI-assisted workflows and mentor guidance.

3 months	120	12	₹13,000
DURATION	GUIDED HOURS	MODULES	FOUNDING BATCH FEE

Who it is for

Beginners, students, freelancers and small-business owners who want to build and launch websites.

Before you begin

Basic computer skills. Previous coding experience is not required; regular practice is essential.

What you will build

- A portfolio website that shows evidence of your work
- An interactive landing page
- A dynamic business website with an admin panel and enquiry management

What you will be able to do

- Decide what a business actually needs — a website, a single page or a simple form — and explain why
- Build responsive, accessible pages and interactive features with JavaScript and React
- Turn a Next.js site into an application with a Supabase database, login and permissions
- Launch and hand over a site with enquiries, uploads, SEO checks and delivery notes

Tools and technologies

HTML · CSS · JavaScript · React · Next.js · Supabase · Git & GitHub · AI coding assistant

Modules

MODULE 1 · 10 H — Understand the Problem Before the Website

- Website purpose and user journeys
- A mental model of browsers, requests and servers
- Editors, folders and local previews
- Relative paths and directory structure
- Semantic HTML
- Navigation and action semantics
- Context and acceptance criteria
- Small diffs and scope control
- Git commits and change history
- Foundation checkpoint

MODULE 2 · 10 H — Turn Design Choices into Rules

- Colour roles and contrast
- Selectors and the cascade
- The box model
- Content-driven layout
- Flexbox versus Grid
- Responsive layout
- Overflow and flexible sizing
- Keyboard navigation and focus
- Motion and user preferences
- Design checkpoint

MODULE 3 · 10 H — A Portfolio That Shows Evidence, Not Hype

- Audience-first content
- Information architecture
- Low-fidelity wireframes
- Visual hierarchy
- Translating design into HTML
- Reusable styling
- Claim verification
- Image dimensions and compression
- Preview deployment
- Project 1 review

MODULE 4 · 10 H — JavaScript: Predict First, Then Run

- Values and data types
- Naming and intent
- Conditions
- Iteration and termination
- Functions, inputs and outputs
- Return values versus logs
- Objects and modelling
- Arrays and mutation
- Debugging hypotheses
- Logic checkpoint

MODULE 5 · 10 H — Interactive Pages: What Happens After the Click?

- The DOM and events
- Interaction states
- Disclosure interactions
- Validation feedback
- Pending UI and duplicate intent
- Asynchronous operations and loading states
- Failure and retry experience
- Filtering and empty results
- Integration and scope
- Project 2 review

MODULE 6 · 10 H React: The Reasoning Behind Reuse

- The motivation for components
- Dependencies and lockfiles
- Component boundaries
- Props and composition
- Identity in rendering
- State and rendering
- Shared state
- Controlled inputs
- Derived values and unnecessary effects
- Component checkpoint

MODULE 7 · 10 H Next.js: From Pages to an Application

- File-based routing
- Shared layouts
- Links and navigation models
- Not-found handling
- Execution boundaries
- The scope of client boundaries
- Dynamic route parameters
- Consistency across pages
- Scope negotiation
- Readiness review

MODULE 8 · 10 H The Database: Truth Behind the Screen

- Data persistence choices
- IDs and field modelling
- Constraints and field rules
- Published states
- Publishable versus secret configuration
- Row-level access
- Data-fetching integration
- Missing-record handling
- Data freshness and refresh reasoning
- Database checkpoint

MODULE 9 · 10 H After Login: The Real Work of Permissions

- Managed authentication
- Authentication versus authorisation
- Session lifecycle
- Server-side permission checks
- Validated mutations
- Partial updates
- State transitions
- Destructive-action experience
- Negative testing
- Admin checkpoint

MODULE 10 · 10 H Enquiries, Uploads and Imperfect Inputs

- The request-to-storage flow
- Server validation
- Private-inbox permissions
- Enquiry workflows
- Upload constraints
- Storage permissions
- Submission deduplication
- Rate limits and friction
- Uncertain outcomes
- Integration checkpoint

MODULE 11 · 10 H — Difficult Questions Before Launch

- Risk-based test design
- Device and keyboard review
- Search intent and page fundamentals
- Crawlability versus access control
- Performance budgets
- Production builds
- Deployment configuration
- Logs and diagnosis
- Recovery rehearsals
- Release readiness

MODULE 12 · 10 H — Delivery: Responsibility Beyond the Code

- Domains, DNS and hosting
- Transport versus application security
- Repeatable releases
- Change requests
- Effort and handover scope
- Handover ownership
- Case-study evidence
- Independent modification assessment
- Project viva A
- Project viva B and boundaries

Batch details

First class	16 Nov 2026
Days and time	Mon – Fri · 10:00 am – 12:00 pm
Ends	5 Feb 2027 (public holidays may move this later)
Classes	60 classes · 10 h/week · home practice 5-6 hours/week
Format	Udaipur, Rajasthan · offline classroom · Hinglish (Hindi + English) · up to 15 learners

Scope: Business websites and basic content management. CRM/ERP and complex commerce are outside this course — they are covered in Full Stack Development with AI.

AI CLASSROOM COURSES · SPECIALIST BUILD

Mobile App Development with AI

Turn an idea into a working mobile app — from screens and data to testing and release preparation.

3 months	120	12	₹22,000
DURATION	GUIDED HOURS	MODULES	FOUNDING BATCH FEE

Who it is for

Students and builders who want to make Android and iOS apps, especially those with an app idea.

Before you begin

Computer skills and readiness for programming practice. An introductory JavaScript bridge is included.

What you will build

- TaskPocket — a task app
- ExpenseLog — an expense tracker
- An Android-tested booking-request app with a release-preparation pack

What you will be able to do

- Design for thumbs, keyboards and small screens, with clear navigation and state
- Store data that survives the app closing, and handle the network failing
- Add login, privacy boundaries and device features that users can trust
- Test, measure performance, build, sign and prepare a release

Tools and technologies

JavaScript · TypeScript · React Native · Expo · Supabase · Android Studio · AI coding assistant

Modules

MODULE 1 · 10 H — Are You Building an App or Solving a Problem?

- Mobile product fit
- Stack selection
- Runtime and project setup
- Values and basic types
- Conditions and functions
- Arrays, objects and identity
- Asynchronous fundamentals
- Constraint-based AI assistance
- Setup debugging
- Entry-bridge assessment

MODULE 2 · 10 H — Mobile Experience: Thumbs, Keyboards and Small Screens

- Native interface primitives
- Touch targets and placement
- Safe areas and insets
- Keyboard-aware layouts
- Flexible mobile layouts
- Font scaling and accessibility
- Non-colour cues
- Reusable mobile components
- Empty-state onboarding
- Usability checkpoint

MODULE 3 · 10 H — Managing Navigation and State

- Navigation stacks
- Navigation choices
- Route parameters
- Invalid navigation targets
- State ownership
- Draft lifecycle
- Event handling
- Derived state
- Mini-app integration
- TaskPocket review

MODULE 4 · 10 H — What Happens to Data After the App Closes?

- Persistence fundamentals
- Save timing
- Data-versioning fundamentals
- Defensive loading
- Destructive local actions
- Numeric form handling
- Data consistency
- Separating business logic
- ExpenseLog build lab
- Mini-app assessment

MODULE 5 · 10 H — The Internet Arrives—and So Do New Failures

- Network contracts
- Response validation
- Request lifecycle
- Timeouts and retry choices
- Out-of-order reads
- Failure classification
- Cache freshness
- Client trust boundaries
- Integration review
- Network assessment

MODULE 6 · 10 H — Login, Privacy and Account Boundaries

- Managed sign-in
- Authentication feedback
- Session storage
- Session-expiry experience
- Own-user access
- Trusted identity sources
- Account switching
- Managed account recovery
- Profile validation
- Authentication and privacy assessment

MODULE 7 · 10 H — Booking Apps: Design States, Not Just Screens

- Capstone scope
- Booking vocabulary
- Reusable catalogue UI
- Date validation
- Timezone awareness
- Data minimisation
- Confirmation experience
- Server-confirmed submission
- History and discovery
- Capstone integration

MODULE 8 · 10 H — Device Features: Permission Means Trust

- Permission timing
- Graceful degradation
- Image selection and resizing
- Upload lifecycle
- Storage access
- Notification intent
- Notification architecture
- Lifecycle cleanup
- Lifecycle fundamentals
- Device-integration review

MODULE 9 · 10 H — Reliability: Double Taps and Disappearing Networks

- Duplicate-submission handling
- Uncertain writes
- Concurrency boundaries
- Business-state transitions
- Merging and identity
- Offline-read policy
- Error routing
- Privacy-aware diagnosis
- Failure lab
- Integration assessment

MODULE 10 · 10 H — Performance and Testing: From Impressions to Evidence

- Performance testing
- Performance diagnosis
- Virtualised lists
- Measured optimisation
- Test layers
- Integration tests
- Accessibility regression
- Build-mode comparison
- Bug-report quality
- Release prioritisation

MODULE 11 · 10 H — Builds, Signing and App-Store Reality

- Build artifacts
- Application identity
- Signing concepts
- Build diagnostics
- Android acceptance build
- Platform differences
- Listing accuracy
- Data inventories
- Submission versus release
- Submission preparation

MODULE 12 · 10 H — Handover and an Honest Mobile Portfolio

- Scope-change evaluation
- Ownership and support
- Interpreting feedback
- Portfolio narrative
- Reproducible handover
- Independent reliability assessment
- Independent build assessment
- Project defence A
- Project defence B
- Final reflection and delivery

Batch details

First class	19 Nov 2026
Days and time	Mon – Fri · 3:00 pm – 5:00 pm
Ends	10 Feb 2027 (public holidays may move this later)
Classes	60 classes · 10 h/week · home practice 6-8 hours/week
Format	Udaipur, Rajasthan · offline classroom · Hinglish (Hindi + English) · up to 15 learners

Scope: One primary mobile stack. Android is assessed; the iOS workflow depends on access to Apple tools. App-store approval is decided by Google and Apple and is not guaranteed.

AI CLASSROOM COURSES · CAREER TRACK

Full Stack Development with AI

From front end to back end — learn to build, test and deploy complete software applications.

6 months	240	24	₹44,000
DURATION	GUIDED HOURS	MODULES	FOUNDING BATCH FEE

Who it is for

Career-focused learners and early developers who want structured skill from interface to database.

Before you begin

Basic computer skills and regular coding practice. A separate website course is not compulsory.

What you will build

- An assessed dashboard and API
- A four-module CRM/ERP-lite product
- A basic mobile companion app
- One reviewed AI feature

What you will be able to do

- Model a business before writing software, then build interfaces staff can actually use
- Design a NestJS API on PostgreSQL with authentication and authorisation
- Build CRM, tasks, orders and reporting modules whose numbers you can trust
- Test, deploy with CI, add an AI feature only where it helps, and defend what you built

Tools and technologies

TypeScript · React · Next.js · NestJS · PostgreSQL · Testing · GitHub Actions · Docker

Modules

MODULE 1 · 10 H — Model the Business Before Building Software

- Problem framing
- User journeys
- System boundaries
- Vertical slices
- AI-assisted specifications
- Acceptance criteria
- The HTTP mental model
- Reproducible environments
- Version history
- Problem-model checkpoint

MODULE 2 · 10 H — Browser Interfaces: From Meaning to Behaviour

- Semantic structure
- Forms and accessible names
- Cascade reasoning
- Responsive information design
- Design tokens
- UI-state modelling
- Dynamic values
- Functions and reuse
- References and mutation
- Browser checkpoint

MODULE 3 · 10 H — Logic, Asynchronous Work and Failure

- Conditions and validation
- Decision tables
- Collection transformations
- Pure transformations
- Asynchronous operations
- Concurrency fundamentals
- Error handling
- Uncertain outcomes
- Hypothesis-driven repair
- Foundation assessment

MODULE 4 · 10 H — TypeScript and Engineering Habits

- Static types
- Nullability and optional values
- Type escapes
- Union types
- DTOs and domain concepts
- Code organisation
- Dependency assessment
- Review checklists
- Collaborative Git
- Month 1 assessment

MODULE 5 · 10 H — Following Data Through React

- Component design
- Typed props
- Render identity
- State snapshots
- State placement
- Derived data
- The purpose of effects
- Asynchronous lifecycle
- Input state
- Frontend review

MODULE 6 · 10 H Next.js and Page Architecture

- URL design
- Layouts and access intent
- Execution choices
- Client-boundary size
- Data ownership
- Error meaning
- Shareable state
- Navigation persistence
- Caching choices
- Navigation assessment

MODULE 7 · 10 H A Dashboard Staff Can Actually Use

- Information density
- Search contracts
- Ordering semantics
- Pagination fundamentals
- Interaction choices
- Bulk-action experience
- Actionable errors
- Mutation feedback
- Keyboard task completion
- Usability assessment

MODULE 8 · 10 H Frontend Contracts and the Assessed Build

- Contract fixtures
- API-contract alignment
- Explicit UI transitions
- Generated-interface review
- Regression planning
- Testing strategy
- Measured frontend decisions
- First-run experience
- Foundation-build assessment A
- Foundation-build assessment B

MODULE 9 · 10 H NestJS: Who Guards the Server?

- Server runtime
- Architecture scope
- Controllers and services
- Providers
- Module boundaries
- HTTP methods
- Status and error contracts
- Runtime validation
- Input allowlists
- Server checkpoint

MODULE 10 · 10 H PostgreSQL: Taking Responsibility for Data

- Identity and schema
- Relational modelling
- Referential integrity
- Business constraints
- Missing-data semantics
- Precision awareness
- Temporal modelling
- Parameterised queries
- Migrations
- Database review

MODULE 11 · 10 H

Connecting the API to Real Data

- ORM reasoning
- Persisted writes
- Patch semantics
- Deletion policy
- N+1 query awareness
- Index trade-offs
- Pagination bounds
- Stable sorting
- Error translation
- Integration checkpoint

MODULE 12 · 10 H

From Authentication to Authorisation

- Authentication and authorisation
- Reviewed authentication integration
- Session-validity policy
- Resource authorisation
- Permission design
- Browser policy versus authentication
- CSRF awareness
- Secret handling
- Foundation-API assessment A
- Foundation-API assessment B

MODULE 13 · 10 H

CRM: From Records to Relationships

- Domain concepts
- Deduplication choices
- Assignment history
- Qualification rules
- Handling untrusted text
- Related operations
- Audit-trail design
- Query scoping
- Workflow integration
- Module review

MODULE 14 · 10 H

Tasks and Teams: The Status Contract

- Definition of done
- Derived-workflow consistency
- Reference validation
- State transitions
- Concurrent-edit awareness
- Collaboration history
- Scoped team visibility
- Workflow efficiency
- Module integration
- Module assessment

MODULE 15 · 10 H

Orders and Inventory-Lite: Numbers You Can Trust

- Order semantics
- Price snapshots
- Numeric invariants
- Deterministic totals
- Concurrency control
- Isolation reasoning
- Atomicity
- Idempotent reversal
- Scope boundaries
- Module assessment

MODULE 16 · 10 H

Reports, Files and Cross-Module Consistency

- Metric definitions
- Join cardinality
- Time boundaries
- Export permissions
- Safe spreadsheet exports
- Asset metadata
- Upload bounds
- Freshness requirements
- Cross-module integration
- Month 4 assessment

MODULE 17 · 10 H

Testing: What Does a Pass Actually Prove?

- Test oracles
- Focused tests
- Integration testing
- Test portfolios
- Negative authorisation tests
- Deterministic fixtures
- Overfitting to tests
- Coverage limits
- Independent verification
- Quality assessment

MODULE 18 · 10 H

Your First Reliability and Operations Responsibilities

- Background jobs
- Idempotent jobs
- Retry policies
- Backpressure awareness
- Resource protection
- Correlation IDs
- Actionable monitoring
- Liveness and readiness
- Release and data compatibility
- Operations drill

MODULE 19 · 10 H

An AI Feature: Helpful or Unnecessary?

- Feature selection
- Data minimisation
- Grounded generation
- Structured outputs
- Timeout and fallback experience
- Untrusted input
- Usage measurement
- Human review
- Regression evaluation
- AI-integration review

MODULE 20 · 10 H

Integration: When Every Part Works Together

- Compatibility
- Cross-layer state
- Policy consistency
- Authority maps
- Representative fixtures
- Configuration validation
- Dependency-failure isolation
- Scope discipline
- Traceability assessment
- Product-readiness review

MODULE 21 · 10 H — A Basic Mobile Companion: Small Scope, Real Integration

- Scope framing
- Shared concepts
- API reuse
- Mobile authentication integration
- Mobile information hierarchy
- Mobile mutations
- Mobile failure states
- Cross-client consistency
- Mobile-extension review
- Extension assessment

MODULE 22 · 10 H — Deployment, Continuous Integration and Recovery

- Hosting choices
- Continuous integration
- Secret management
- Release sequencing
- Fail-fast settings
- Deployment connectivity
- Recovery practice
- Release operations
- Preproduction testing
- Release assessment

MODULE 23 · 10 H — Product Delivery and Professional Judgement

- Audience-based documentation
- Executable documentation
- Decision records
- Handover clarity
- Operational prioritisation
- Support boundaries
- Accurate product claims
- Honest demonstrations
- Prioritisation
- Handover review

MODULE 24 · 10 H — Final Defence: Can You Explain What You Built?

- Requirement transfer
- Vertical-change assessment
- Debugging assessment
- Concurrency assessment
- Security assessment
- Engineering trade-offs
- Capstone viva A
- Capstone viva B
- Final readiness
- Reflection and next-level planning

Batch details

First class	22 Nov 2026 — Sunday 22 Nov is a special first class; regular classes run Mon – Fri from 23 Nov.
Days and time	Mon – Fri · 5:30 pm – 7:30 pm
Ends	6 May 2027 (public holidays may move this later)
Classes	120 classes · 10 h/week · home practice 8-10 hours/week

Format	Udaipur, Rajasthan · offline classroom · Hinglish (Hindi + English) · up to 15 learners

Scope: A modular monolith and scoped business workflows — not a complete enterprise ERP, accounting system or mobile specialisation. Job placement and interviews are not guaranteed.

AI CLASSROOM COURSES · ADVANCED ENGINEERING

AI Coding Agent Development

Build coding agents, connect tools, automate workflows, and verify what your agents do.

6 months	300	12	₹58,000
DURATION	GUIDED HOURS	MODULES	FOUNDING BATCH FEE

Who it is for

Developers and technical learners who already write code and want to engineer agent systems, not just chat with them.

Before you begin

Working JavaScript/TypeScript, asynchronous APIs, Git, terminal use and backend fundamentals. A practical readiness check precedes enrollment.

What you will build

- A tool-using LLM application
- A reviewable workflow with human approval
- RepoGuide — repository-understanding milestones
- TestFix — a controlled coding agent evaluated on twenty unseen tasks

What you will be able to do

- Build LLM applications with contracts and evidence, where the model proposes and the system enforces
- Design workflows with state, human approval, retrieval and MCP within a context budget
- Let an agent read a repository and propose reviewable patches inside a sandbox
- Evaluate agents on unseen tasks, control safety, privacy and cost, and turn the agent into a product

Tools and technologies

TypeScript · LLM APIs · Tool calling · LangGraph · MCP · Next.js · NestJS · PostgreSQL

Evals & tracing

Modules

MODULE 1 · 25 H LLM Applications: Predictions, Contracts and Evidence

- Narrow task selection
- The automation spectrum
- Engineering readiness
- Model behaviour
- Token budgets
- Output variability
- Reproducibility limits
- Instruction boundaries
- Input trust
- Model-API integration
- Secret boundaries
- Latency perception
- Structured streaming
- Schema versus domain validation
- Abstention design
- Provider-failure handling
- Backoff reasoning
- Usage accounting
- Prompt versioning
- Data-conscious tracing
- Prompt experiments
- Baseline design
- Small-sample interpretation
- LLM-exercise integration
- Foundation assessment

MODULE 2 · 25 H Tools: The Model Proposes, the System Enforces

- Tool-call lifecycle
- Tool-schema design
- Path scope
- Bounded outputs
- Result contracts
- Tool granularity
- Read-first development
- Input validation
- Tool-source trust
- Read permissions and privacy
- Step budgets
- Termination criteria
- Loop detection
- Tool concurrency
- Error feedback
- Fallback safety
- Contract testing
- Run transparency
- User-facing rationale
- Approval scope
- Approval idempotency
- Cancellation semantics
- Tool-use integration
- Capability assessment
- Module review

MODULE 3 · 25 H Workflows, State and Human Approval

- Deterministic versus dynamic control
- Workflow graphs
- Structured state
- State representation
- Step granularity
- State reducers
- Persistence
- Run isolation
- Durable execution
- Replay safety
- Approval placement
- Rejection paths
- Stale approvals
- Time budgets
- Recovery classification
- Progress invariants
- Observable workflows
- Join semantics
- Fallback decisions
- Approval authorisation
- Debug artifacts
- Orchestration integration
- Recovery lab
- Assessment
- Module review

MODULE 4 · 25 H Retrieval, MCP and the Context Budget

- Context selection
- Retrieval choices
- Chunking trade-offs
- Evidence sufficiency
- Retrieval scores
- Source provenance
- Index freshness
- Authorised indexing
- Budget allocation
- Context compression
- Protocol versus trust
- MCP concepts
- Capability enablement
- Connection boundaries
- Audience-bound credentials
- Capability-change review
- Indirect prompt injection
- Layered defence
- Missing-evidence behaviour
- Evidence precedence
- Retrieval evaluation
- Integration exercise
- Optimisation lab
- Assessment
- Module review

MODULE 5 · 25 H Understand the Repository: From File Tree to Behaviour

- Repository orientation
- Untrusted project scripts
- Project detection
- Dependency reproducibility
- Execution entrypoints
- Call-site analysis
- Type dependencies
- Failure attribution
- Clarification
- Specification sources
- Search strategy
- Scope exclusions
- Sensitive-context handling
- Snapshot consistency
- Parsing trade-offs
- Scope discipline
- Test-driven investigation
- Environment dependencies
- Stack interpretation
- Task-specific summaries
- Evidence packaging
- Repository assistants
- Generalisation checks
- Investigation assessment
- Module review

MODULE 6 · 25 H Propose a Patch Without Taking Over

- Bounded action plans
- Minimal patches
- Patch provenance
- Mutation scope
- Patch validation
- Protected evaluation assets
- Dependency changes
- Behaviour-preserving constraints
- Untrusted directives
- Change classification
- Reviewable artifacts
- Content-bound approvals
- Atomic patching strategies
- Protecting user changes
- Scoped recovery
- Evidence-based summaries
- Alternative evaluation
- Search budgets
- Feedback loops
- Replanning boundaries
- Mini-project extension
- Patch critique
- Concurrency lab
- Assessment
- Module review

MODULE 7 · 25 H Sandboxing: Don't Give the Model the Keys to the Machine

- Isolation threat models
- Execution capabilities
- Structured execution
- Filesystem scope
- Egress policy
- Controlled provisioning
- Credential minimisation
- Resource limits
- Memory quotas
- Storage quotas
- Output limits
- Process lifecycle
- Least-privilege execution
- Execution identity
- Immutable inputs
- Tenant isolation
- Artifact-export validation
- Execution-outcome semantics
- Trusted test harnesses
- Safe operational logs
- Execution APIs
- Execution integration
- Concurrency lab
- Assessment
- Safety review

MODULE 8 · 25 H The Test-Fix Loop, Jobs and Review Dashboard

- Failure-feedback packaging
- Bounded repair loops
- Attempt isolation
- Independent tests
- Regression comparison
- Metric integrity
- Nondeterminism analysis
- Job architecture
- Job identity
- Job-state transitions
- Failure recovery
- Race handling
- Status delivery
- Run observability
- Outcome reporting
- Review experience
- Reviewer permissions
- Action boundaries
- Artifact authorisation
- Retention choices
- Product integration
- Fault isolation
- Small-load validation
- Assessment
- Integration review

MODULE 9 · 25 H

Evaluation: Turn the Demo into Evidence

- Success criteria
- Multidimensional scoring
- Fair baselines
- Dataset separation
- Evaluation composition
- Data leakage
- Metric arithmetic
- Attempt accounting
- Evaluation independence
- Specification alignment
- Rubric calibration
- Failure taxonomy
- Coverage versus precision
- Economic evaluation
- Distribution awareness
- Uncertainty fundamentals
- Experiment control
- Controlled experiments
- Component-value tests
- External validity
- Learning from failures
- Reproducible reporting
- Metric critique
- Assessment
- Module review

MODULE 10 · 25 H

Safety, Privacy, Quotas and Unit Economics

- Threat modelling
- Layered mitigation
- Safe adversarial fixtures
- Capability-escalation checks
- Memory isolation
- Retention policy
- Deletion propagation
- Policy composition
- Atomic quota reservation
- Budget uncertainty
- Usage reconciliation
- Cost-quality trade-offs
- Total operating cost
- Usage-based product design
- Trial controls
- Idempotent billing demonstrations
- Eventual-consistency experience
- Operational judgement
- Emergency controls
- Safe incident response
- Continuous review
- Failure-cost lab
- Isolation lab
- Assessment
- Module review

MODULE 11 · 25 H

Productisation: From Agent to Usable Product

- Customer selection
- Customer discovery
- Demand evidence
- Trust and onboarding
- First-use experience
- Onboarding diagnostics
- Compatibility promises
- Interaction design
- Actionable status experience
- Decision support
- Designing for failure
- Accurate product messaging
- Pilot planning
- Consent and authority
- Confidentiality
- Commercial scoping
- Product positioning
- Support operations
- Product deployment
- Operational configuration
- Learning metrics
- Product storytelling
- Integration lab
- Assessment
- Release review

MODULE 12 · 25 H

Capstone: Unseen Tasks, Honest Results and Handover

- Evaluation freeze
- Evaluation governance
- Fair evaluation
- Safety readiness
- Reproducibility
- Held-out execution 1
- Evaluation observation 1
- Held-out execution 2
- Evaluation observation 2
- Held-out execution 3
- Evaluation observation 3
- Held-out execution 4
- Evaluation observation 4
- Held-out execution 5
- Evaluation finalisation
- Final metric reporting
- Final control assessment
- Final approval assessment
- Final reliability assessment
- Post-evaluation transfer task
- Delivery artifacts
- Pilot-proposal defence
- Capstone defence A
- Capstone defence B
- Completion and next steps

Batch details

First class	25 Nov 2026
Days and time	Mon – Fri · 12:30 pm – 3:00 pm
Ends	11 May 2027 (public holidays may move this later)
Classes	120 classes · 12.5 h/week · home practice 8-12 hours/week
Format	Udaipur, Rajasthan · offline classroom · Hinglish (Hindi + English) · up to 15 learners

Scope: A narrow agent built on existing models. Training foundation models, unrestricted shell access and automatic production deployment are excluded, and sales are not guaranteed. Learners without JavaScript/TypeScript and backend basics should start with Full Stack Development with AI.

AI CLASSROOM COURSES · BUSINESS SKILLS

Digital Marketing with AI

Build a practical workflow for content, campaigns and marketing measurement — with AI.

45 days	66	7	₹12,000
DURATION	GUIDED HOURS	MODULES	FOUNDING BATCH FEE

Who it is for

Beginners, freelancers, creators and small-business owners who want practical marketing skills.

Before you begin

Basic computer and browser use, writing skills and comfort with spreadsheets. Coding is not required.

What you will build

- A thirty-day content plan, twelve creatives and three scripts
- One sample video and a landing page
- An SEO audit and two ad drafts
- A marketing dashboard and a client proposal

What you will be able to do

- Define the offer and audience before marketing anything
- Create content that works — posts, scripts and a sample video
- Run SEO and social content that turns into consented leads
- Plan ads as hypotheses, then read analytics to make decisions

Tools and technologies

ChatGPT / Claude · Canva · No-code landing pages · Google Search Console · Meta Ads · Google Ads · GA4

Modules

MODULE 1 · 10 H — Before Marketing: The Offer and the Audience

- Marketing objectives
- Audience specificity
- Offer design
- Research quality
- Positioning
- Customer journeys
- Verified AI research
- Brand-aware prompting
- Channel selection
- Strategy review

MODULE 2 · 10 H — Content That Works, Not Just Content That Looks Good

- Responsible attention
- Benefit-led copy
- Specificity and voice
- Visual hierarchy
- Consistency
- Narrative sequence
- Short-video structure
- Creative verification
- Editing choices
- Creative review

MODULE 3 · 10 H — Search: Relevance Before Rankings

- Search intent
- Traffic quality
- Helpful content
- On-page alignment
- Audit prioritisation
- Indexability versus security
- Local intent
- AI-search fundamentals
- Reputation and relevance
- Search review

MODULE 4 · 10 H — From Social Content to Leads: Consent and Conversion

- Adapting to platform context
- Content planning
- Community handling
- Conversion-focused pages
- Data minimisation
- Permission-based follow-up
- Operational funnels
- Responsible messaging
- Funnel quality assurance
- Journey assessment

MODULE 5 · 10 H — Advertising: Form a Hypothesis Before Spending

- Campaign objectives
- Click quality
- Targeting trade-offs
- Message consistency
- Experiment design
- Controlled testing
- Measurement quality assurance
- Campaign tagging
- Channel campaign planning
- Campaign assessment

MODULE 6 · 10 H — Analytics: From Numbers to Decisions

- Metric hierarchy
- Revenue versus costs
- Attribution versus causality
- Uncertainty
- Data quality
- Measurement limitations
- Diagnosis
- Decision-oriented reporting
- Service scoping
- Portfolio review

MODULE 7 · 6 H — Final Project Clinic and Explain-Back

- Final project correction 1
- Final project correction 2
- Final project correction 3
- Assessment A
- Assessment B
- Completion reflection

Batch details

First class	28 Nov 2026 — Saturday 28 Nov is a special first class; regular classes run Mon – Fri from 30 Nov.
Days and time	Mon – Fri · 8:00 am – 10:00 am
Ends	11 Jan 2027 (public holidays may move this later)
Classes	32 classes · 10 h/week · home practice 5-6 hours/week
Format	Udaipur, Rajasthan · offline classroom · Hinglish (Hindi + English) · up to 15 learners

Scope: An end-to-end practical foundation. Live ad spend is optional and needs approval; simulated results are labelled. Leads and rankings are not guaranteed.