Mentor Guide for Extream.AI Project

1. Overview of Extream.AI Project

Project Description:

Extream.AI is an agentic automation platform that empowers enterprises to deploy intelligent, no-code AI agents capable of automating complex workflows across both legacy and modern systems.

Mission and Goals:

- Democratize AI-driven automation
- Enable no-code creation of intelligent agents
- Integrate modern GenAI capabilities with enterprise workflows

Key Innovation Areas:

- Agentic Automation
- LLM Prompt Orchestration
- Multi-Agent Systems
- Drag-and-Drop No-Code Platforms
- AI Governance, Compliance, and Observability

2. Mentor Objectives & Responsibilities

Time Commitments by Role (M1–M6):

- GenAI & Prompt Intelligence: 6–8 hrs/week
- Agent & Automation Architect: 6–8 hrs/week
- No-Code Experience: 4–6 hrs/week
- DevOps & Integration: 5–6 hrs/week
- Governance & Security: 4–6 hrs/week

Core Responsibilities:

- Participate in sprint planning, retrospectives, and design reviews
- Offer hands-on code, architecture, and UX feedback
- Guide documentation and prompt tuning reviews
- Ensure best practices in agent design, integration, and governance

3. Tech Stack and Tools by Role

GenAI & Prompt Intelligence

Languages & Libraries: Python, LangChain, LlamaIndex
LLM Tools: OpenAI API, Hugging Face Transformers, Ollama
Vector Stores: Pinecone, FAISS, ChromaDB
Evaluation & Monitoring: LangSmith, Weights & Biases
Dev Tools: JupyterLab, VSCode, LangChain Assistant

Agent & Automation Architect

Agent Frameworks: AutoGen, CrewAI, LangGraph
Orchestration: AWS Step Functions, Temporal.io
MicroBOT Design: FastAPI, RabbitMQ
Local Sandbox: Docker, Postman
Diagramming: Mermaid.js, Miro

Platform & No-Code UX Engineer

Frontend: React, TypeScript, React Flow, Retool
Backend: Node.js, Flask
Testing: Cypress, Storybook
No-Code Tools: Blockly, Retool Playground
Accessibility: axe-core, Lighthouse

Integration & DevOps

CI/CD: GitHub Actions, Argo CD
Infrastructure as Code: Terraform
Containerization & Orchestration: Docker, Kubernetes (EKS/GKE)
Monitoring & Observability: Prometheus, Grafana, Loki, ELK Stack
Chaos Engineering: KubeMonkey
Dev Tools: GitHub CLI, Docker Desktop

AI Governance & Security

Policy Enforcement: OPA (Open Policy Agent), AWS GuardDuty
Auditing & Provenance: Apache Atlas, SQLite
Compliance: NIST AI RMF, GDPR, Microsoft RAIL Guidelines
Security Drills: Red Team Toolkit, Threat Modeling Templates

4. Reference Materials & Learning Links

GenAI & Prompt Intelligence:

- LangChain: https://docs.langchain.com
- OpenAI: https://platform.openai.com/docs
- Prompt Engineering: https://learnprompting.org, OpenAI Cookbook, LangChain Hub

Agent Architectures:

- ReAct, AutoGPT Whitepapers
- AutoGen Documentation, CrewAI Examples
- State Machines: https://xstate.js.org

No-Code Platforms:

- Blockly Tutorials, React DnD Tutorial
- React Flow Docs, Retool Playground

DevOps/Integration:

- GitHub Actions: https://docs.github.com/en/actions
- CI/CD with Docker & Kubernetes
- Kubernetes Basics, Terraform Modules

Governance & Security:

- NIST AI RMF: https://www.nist.gov/itl/ai-risk-management-framework
- GDPR: https://gdpr.eu
- Microsoft RAIL Guidelines

5. Development Environment Setup

Recommended Tools:

- VSCode (with GitLens, Dev Containers, LangChain Assistant)
- Docker Desktop, Postman, GitHub CLI
- JupyterLab, MLflow, Locust

Setup Guidelines:

- Dev containers using `devcontainer.json`
- `docker-compose.yaml` with API, Vector DB, UI
- Local sandbox for prompt/agent testing

6. Code Check-in & Review Procedures

Branching Strategy:

- GitFlow: feature branches → dev → main

PR Requirements:

- Link PR to Jira/GitHub Issue (e.g., EXT-123)
- Automated test pass (unit/integration)
- Reviewed by ≥2 aligned members (e.g., Siddharth + TBD)

Commit Rules:

- Semantic messages (e.g., feat:, fix:)
- Atomic commits only

Merge Protocol:

- Squash merge into `dev` after approval
- Weekly dev → main promotions

7. Project Tracking Components

KPIs by Role:

- GenAI: Prompt success rate, LLM latency, feedback loop coverage
- Agents: Decision log completeness, BOT execution success rate
- Platform: UI load time, accessibility (WCAG 2.1)
- DevOps: Deployment frequency, rollback success
- Governance: Audit trail gaps, threat model coverage

Tools:

- Task Tracking: GitHub Projects/Jira
- Documentation: MkDocs + GitHub Pages
- Logging: Loki + Grafana

Governance Artifacts:

- Prompt audit logs (JSON in S3)
- Agent decision trees (Mermaid.js)
- Failover test reports (/docs/compliance)

8. Mentor-to-Role Alignment

Role	Assigned Member(s)	Mentor Name	Time Commitment
GenAI & Prompt Intelligence	Siddharth P	jseetharaman@kanchiuniv.ac.in	6–8 hrs/week
Agent & Automation Architect	Siddharth P	jseetharaman@kanchiuniv.ac.in	6–8 hrs/week
Platform & No-Code UX	A Swathy, S Ramya Sri	jseetharaman@kanchiuniv.ac.in	4–6 hrs/week
Integration & DevOps	K Mithun Kumar	jseetharaman@kanchiuniv.ac.in	5–6 hrs/week
AI Governance & Security	V Balakrishnan/ B. Saravanan	jseetharaman@kanchiuniv.ac.in	4–6 hrs/week

9. Guidelines for Open Roles

Mentors must help identify and onboard:

- Prompt Tuning Engineer (GenAI)
- Backup DevOps Engineer
- Compliance/Failsafe Contributor (AI Governance)

10. Best Practices & Tips

- Follow a documentation-first approach
- Enable safe rollbacks & test coverage for all features
- Maintain modular, reusable, and readable code
- Encourage peer code reviews and knowledge-sharing sessions

Mentor Action Plan

Per Role:

GenAI Mentors:

- Weekly prompt review sessions
- Validate vector store indexing strategies

Agent Architects:

- Diagram state machines before implementation
- Simulate MicroBOT failure scenarios

Platform Engineers:

- Integrate accessibility tools (axe-core)
- Deliver ≥5 visual workflow templates by M4

DevOps Mentors:

- Enforce IaC (Terraform)
- Mandate chaos testing (e.g., KubeMonkey)

Governance Strategists:

- Map GDPR/SOC2 triggers (e.g., data deletion)
- Conduct quarterly threat modeling workshops

Key Reminders:

1. Use Role Pairing for TBD positions
2. Governance Checkpoints:
   - Code merges require governance sign-off
   - Monthly "red team" security drills
3. Student Onboarding:
   - Provide GitPod or Docker sandbox
   - Assign micro-projects in M1 (e.g., prompt chain triage bot)

Delivery Timeline:

- Draft Guide: M2 W3
- Feedback Loop: M2 W4–M3 W1
- Finalization: Pre-M3 surge

Tool Installation Checklist by Role

Role	Tools to be Installed
GenAI & Prompt Intelligence	Python, JupyterLab, VSCode (LangChain Assistant), LangChain, LlamaIndex, OpenAI CLI, Hugging Face Transformers, Ollama, Pinecone CLI, FAISS, ChromaDB
Agent & Automation Architect	AutoGen, CrewAI, LangGraph, FastAPI, Docker, Postman, RabbitMQ, Mermaid.js plugin, Miro Desktop App
Platform & No-Code UX	Node.js, React, TypeScript, Flask, Cypress, Storybook, Blockly, Retool, React Flow, axe-core browser extension, Lighthouse
Integration & DevOps	Docker Desktop, GitHub CLI, GitHub Actions, Argo CD, Terraform CLI, Prometheus + Grafana Stack, ELK Stack, KubeMonkey, Kubernetes CLI (kubectl), Helm
AI Governance & Security	OPA CLI, Apache Atlas (Local Docker), SQLite Browser, Red Team Toolkit, Threat Modeling Templates, GDPR/NIST document toolkits

Skills-to-Tool Matrix

Skill Area	Tools Mapped
Prompt Engineering	LangChain, OpenAI API, LangChain Assistant, LlamaIndex
LLM Orchestration	LangGraph, CrewAI, AutoGen
Agent Simulation	Docker, FastAPI, RabbitMQ
Frontend Dev	React, TypeScript, React Flow, Cypress, axe-core
No-Code Workflow	Retool, Blockly, Miro
Backend Dev	Node.js, Flask
CI/CD Automation	GitHub Actions, Argo CD
Infrastructure as Code	Terraform
Monitoring & Logging	Prometheus, Grafana, Loki, ELK
Security & Compliance	OPA, GuardDuty, Apache Atlas, Threat Modeling Templates
Accessibility Auditing	Lighthouse, axe-core
Chaos Engineering	KubeMonkey

Let’s democratize automation responsibly! ✨
—Extream.AI Core Team