Extream.AI Developer Guide & Student Developer Onboarding Manual

1. Purpose of this Guide

This Developer Guide outlines all necessary instructions to set up local development environments tailored to the Extream.AI platform. It includes role-based scopes, setup commands, tools, folder organization, contribution workflows, and learning resources for contributors across GenAI, agent architecture, platform UX, DevOps, and AI governance domains.

2. Scope

This guide applies to developers and students contributing to:

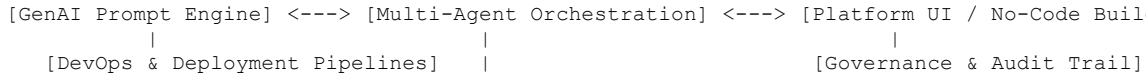
- Prompt engineering and GenAI toolchains
- Multi-agent orchestration frameworks
- No-code AI platforms and UX
- CI/CD pipelines, deployment, and monitoring
- AI risk governance and audit trail generation

3. Introduction & Project Overview

3.1 What is Extream.AI?

Extream.AI is a modular platform integrating Generative AI, autonomous agents, and no-code UX to enable rapid AI automation for enterprises. The system orchestrates microbots, manages prompt intelligence, and ensures AI governance across workflows.

3.2 Architecture Diagram



3.3 Key Terms & Concepts

- **Prompt Chain:** Sequential LLM prompt workflows
- **Microbot:** Autonomous AI agents performing tasks
- **Vector DB:** Vector databases for embeddings and similarity search
- **LangChain, LangGraph:** Frameworks for prompt orchestration and agent interaction
- **Audit Trail:** Immutable logs of AI decisions for governance
- **Chaos Testing:** Fault injection and resilience testing

4. Roles, Scopes, and Environment Setup

4.1 GenAI & Prompt Intelligence Engineer

Scope: Build intelligent prompt chains, manage LLM APIs, vector store embeddings, and evaluation workflows.
Setup:

```
sudo apt install python3 python3-pip
pip install virtualenv

virtualenv extream_genai_env
source extream_genai_env/bin/activate

pip install langchain llama-index openai chromadb faiss-cpu pinecone-client
jupyterlab
pip install langsmith  # Optional observability tool

jupyter lab
```

4.2 Agent & Automation Architect

Scope: Develop autonomous microbots, integrate with LangGraph/CrewAI/AutoGen, build decision trees.
Setup:

```
pip install autogen crewai langgraph fastapi

sudo apt install docker.io
sudo snap install postman

docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3-
management  # Optional
```

4.3 Platform & No-Code UX Engineer

Scope: Develop React/TypeScript frontend, Retool workflows, Blockly configurations, accessibility.
Setup:

```
sudo apt install nodejs npm
npm install -g yarn

git clone https://github.com/extream-ai/platform-ui.git
cd platform-ui
npm install

npx storybook dev
```

```
npx cypress open
```

Retool Playground for no-code preview available at https://retool.com/playground

4.4 Integration & DevOps Engineer

Scope: Setup GitHub Actions, Terraform modules, monitoring, chaos testing.
Setup:

```
sudo apt install docker.io gh

curl -LO "https://dl.k8s.io/release/$(curl -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 |
bash

brew install argocd

sudo apt install terraform

helm repo add grafana https://grafana.github.io/helm-charts
helm install monitoring-stack grafana/k6-prometheus
```

4.5 AI Governance & Security Strategist

Scope: Track LLM decisions, build audit trails, enforce compliance, simulate threats.
Setup:

```
bash
CopyEdit
sudo apt install python3 sqlite3
pip install pandas jupyterlab matplotlib openpolicyagent

docker run -d -p 21000:21000 apache/atlas

git clone https://github.com/clxend/red-team-toolkit.git
```

5. Directory Structure Recommendation

```
extream-ai/
├── prompt-engineering/
├── agent-automation/
├── platform-ui/
├── devops/
├── governance/
├── vector-dbs/
```

6. Git Workflow

```
git checkout -b feat/<role>/<ticket-id>
git add .
git commit -m "feat: add vector db integration [EXT-123]"
git push origin feat/<role>/<ticket-id>
```

7. Troubleshooting Tips

- Use `docker ps` and `docker logs` for container health checks
- Enable LangChain debug mode: `langchain.debug=True`
- Use `kubectl get pods` and `helm ls` to monitor infrastructure
- Re-authenticate OpenAI CLI if LLM calls fail: `openai api key set <key>`

8. Student Developer Onboarding Manual

8.1 Development Goals & Milestones

MVP Features:

- Intelligent prompt chains using LangChain
- Autonomous agent integration via AutoGen
- React-based no-code UX builder
- CI/CD pipelines with Terraform & ArgoCD
- AI audit trail & governance dashboards

Student Contribution Opportunities:
Students can select tasks aligned with their interests and skill levels. Look for issues tagged "good first issue" or "student-friendly" in the project repositories.

8.2 Role-Based Learning Paths & Skills

Role	Skills to Learn	Suggested Resources	Starter Task Example
GenAI & Prompt Intelligence	Python, LangChain, LLM APIs, Vector DBs	LangChain docs, OpenAI API tutorials	Build a summarization prompt chain
Agent & Automation Architect	Python async, FastAPI, Docker, RabbitMQ	AutoGen docs, Docker tutorials	Create a microbot for automated email replies

Role	Skills to Learn	Suggested Resources	Starter Task Example
Platform & No-Code UX Engineer	React, TypeScript, Storybook, Cypress	React docs, Retool playground	Implement a new React workflow UI component
Integration & DevOps Engineer	GitHub Actions, Terraform, Kubernetes	Terraform tutorials, Kubernetes basics	Deploy monitoring stack via Helm
AI Governance & Security	Python, OpenPolicyAgent, Apache Atlas	OPA docs, Apache Atlas intro	Create a compliance audit report

8.3 Example Projects & Starter Tasks

Example Task: Build a LangChain prompt that generates quizzes from an article.

- Uses OpenAI API
- Supports multiple-choice questions
- Includes unit tests using PyTest

8.4 Coding, Documentation & Testing Standards

- Follow PEP8 for Python; ESLint + Prettier for JS/TS
- Write clear docstrings and update README as needed
- Mandatory unit tests for new features (PyTest, Jest, Cypress)
- CI pipelines enforce linting and tests before merges

8.5 Git & Contribution Workflow

- Branch naming: `feat/genai/EXT-123`
- Use pull request templates with checklists
- Follow code review best practices
- Use issue tagging and assignments

8.6 Security, Ethics & Governance Guidelines

- Adhere to data privacy best practices
- Avoid prompt injections; ensure safe prompt design
- Use sandboxed environments for testing agents
- Maintain audit trails for AI decisions
- Follow ethical AI usage and red team toolkit protocols

8.7 Community & Support

- Join Slack/Discord at [link to community]
- Weekly office hours with role leads
- Monthly virtual hackathons and feature sprints

- Mentorship program pairing students with senior developers

8.8 Evaluation & Progress Tracking

- Self-assessment checklists per role
- Milestone badges for tasks, projects, code reviews
- Regular feedback sessions with mentors
- Recognition in team meetings for significant contributions

8.9 FAQs and Common Issues

- Common setup errors and solutions
- Debugging LLM calls or vector database problems
- Contact points for urgent assistance

9. Contacts & Role Leads

Role	Contact Person
Prompt Chains	Siddharth P
Agents & MicroBots	Siddharth P
UX Workflow Builder	A Swathy / S Ramya Sri
DevOps Pipelines	K Mithun Kumar
AI Audit/Governance	V Balakrishnan/ B. Saravanan