AI Agent & Automation Engineering Program



Duration: 5 months : 3 classes per week : 2 hours per class

Module 1: Python Programming Fundamentals for AI Agents

1. Python Basics

- Introduction to Python
- Variables and data types
- Type conversion
- Input and output
- Operators
- Conditional statements
- Loops

2. Python Data Structures

- Strings and string operations
- Lists and list methods
- Tuples
- Sets
- Dictionaries and dictionary methods
- Indexing and slicing
- Nested data structures

3. Python Functions

- Defining and calling functions
- Parameters and arguments
- Return statements
- Default arguments
- Keyword arguments
- Lambda functions
- List comprehension
- Dictionary comprehension

4. Essential Python Programming

- Modules and packages
- Import statements
- File handling

TuxAcademy

- Reading and writing text files
- Working with CSV files
- JSON handling
- Exception handling
- Basic OOP: classes and objects
- Pandas(Basic)



5. Python Development Environment

- PyCharm / VS Code
- Installing packages using pip
- Virtual environments
- Introduction to Git and GitHub
- Working with environment variables

Lab Activity:

- Build a student record management program.
- Read and write JSON data.
- Create a Python application that processes customer records.
- Build a simple file-processing application.

Module 2: LLM Fundamentals and Prompt Engineering

- Introduction to Generative AI
- What are Large Language Models?
- How LLMs generate responses
- Tokens and tokenization
- Context windows
- Model parameters
- Temperature and top-p
- Hallucinations and limitations
- Introduction to OpenAI, Gemini and Claude
- Prompt engineering
- Zero-shot and few-shot prompting
- Role-based prompting
- Structured prompting
- JSON output generation

Lab Activity: Build an AI text summarizer and structured information extraction application.

Module 3: Working with AI APIs

- Understanding APIs
- Client-server architecture
- REST APIs
- HTTP methods
- GET and POST requests
- HTTP status codes

TuxAcademy

- Headers and authentication
- JSON request and response
- API keys and environment variables
- OpenAI API
- Gemini API
- Anthropic API
- Streaming responses
- Conversation history
- Structured outputs
- Pydantic fundamentals
- Function calling and tool calling
- API error handling and rate limits

Lab Activity:

- AI chatbot using API
- AI information extraction system
- AI application using structured outputs

Module 4: LangChain and LLM Application Development

- Introduction to LangChain
- LangChain architecture
- Chat models
- Prompt templates
- Output parsers
- Runnables and LCEL
- Chains
- Document loaders
- Text splitters
- Model integration
- Tool creation
- Tool binding
- Function calling
- Agent fundamentals

Lab Activity: Build a tool-enabled AI assistant.

Module 5: Retrieval-Augmented Generation (RAG)

RAG Fundamentals

- What is RAG?
- RAG architecture
- RAG versus fine-tuning
- Retrieval and generation

TuxAcademy

Document Processing

- PDF and Word document loading
- Text extraction
- Text splitting
- Chunking strategies
- Metadata extraction



Embeddings

- What are embeddings?
- Text embeddings
- Semantic similarity
- Cosine similarity
- Embedding models

Vector Databases

- Introduction to vector databases
- ChromaDB
- Qdrant
- Collections and points
- Vector storage
- Similarity search
- Metadata filtering

Advanced RAG

- Hybrid search
- Reranking
- Query transformation
- Retrieval optimization
- RAG evaluation

Lab Activity:

- PDF Question Answering System
- Company Knowledge Assistant
- Document Research Assistant

Module 6: AI Agent Development with LangGraph

Core specialization module

Understanding AI Agents

- What are AI agents?
- Chatbots versus AI agents
- Agent architecture
- Agent execution cycle
- Reasoning and planning
- Autonomous versus supervised agents
- Agent use cases

TuxAcademy

Agent Components

- LLMs as decision-makers
- Tools and tool calling
- Custom Python tools
- Tool selection
- Agent memory
- Short-term and long-term memory
- Conversation history



LangGraph

- Introduction to LangGraph
- Graph architecture
- Nodes and edges
- State management
- Conditional edges
- Cycles and loops
- Agent workflows
- Tool execution nodes
- State persistence
- Checkpointing
- Human-in-the-loop
- Interruptions and approvals
- Error handling and retries

Advanced Agent Patterns

- Planning and execution
- Reflection and self-correction
- Dynamic tool selection
- Supervisor patterns
- Agent handoffs

Lab Activity:

- Personal AI Assistant
- AI Research Agent
- AI Travel Planning Agent
- Human-Approved Business Assistant

Module 7: Model Context Protocol (MCP)

- What is MCP?
- Why MCP is needed
- MCP architecture
- Hosts, clients and servers
- MCP versus API integration
- MCP versus function calling

TuxAcademy

- Tools, resources and prompts
- Official MCP Python SDK
- Creating custom MCP servers
- Creating MCP clients
- Connecting agents with MCP servers
- File system integration
- Database integration
- External API integration
- MCP security and permissions



Lab Activity:

- Custom File Management MCP Server
- Database MCP Server
- MCP-powered AI Assistant

Module 8: Multi-Agent Systems

- Single-agent versus multi-agent systems
- Agent collaboration
- Agent communication
- Task delegation
- Supervisor architecture
- Sequential and parallel execution
- Shared state
- Agent orchestration

CrewAI

- Agents
- Roles and goals
- Tasks
- Tools
- Crews
- Processes
- Sequential workflows
- Hierarchical workflows

LangGraph Multi-Agent Systems

- Supervisor agents
- Sub-agents
- Agent handoffs
- Collaborative workflows

Lab Activity:

- Multi-Agent Research System
- AI Content Creation Team
- Business Analysis Agent Team

TuxAcademy

Module 9: AI Automation with n8n

Core specialization module



Automation Fundamentals

- Traditional automation versus AI automation
- Rule-based versus AI-powered workflows
- Workflow triggers and actions
- Event-driven automation
- Workflow orchestration

n8n Fundamentals

- Introduction to n8n
- Installation and setup
- Cloud versus self-hosted
- Workflows and nodes
- Manual triggers
- Schedule triggers
- Webhook triggers
- Conditional logic
- Switch nodes
- Loops
- Data transformation
- Expressions
- Error handling

API Integrations

- HTTP Request node
- REST API integration
- Authentication
- Webhooks
- JSON processing
- Connecting external services

AI Automation

- AI Agent node
- LLM integration
- AI tools
- Memory integration
- RAG integration
- Multi-step AI workflows
- Human approval
- Error handling and retries

Business Integrations

- Gmail

TuxAcademy

- Google Sheets
- Google Drive
- Microsoft 365 overview
- CRM systems
- Databases
- Slack and communication tools
- Email automation
- Document automation



Other Automation Platforms

- Make: scenarios, modules and integrations
- Zapier: triggers, actions and multi-step workflows

Lab Activity:

- AI Email Processing System
- Automated Customer Support Workflow
- AI Excel Report Generator
- Automated Lead Management System
- Document Processing and Approval System

Module 10: AI Application Development

Streamlit

- Streamlit fundamentals
- Application layouts
- Input components
- Chat interfaces
- File uploads
- Session state
- Connecting AI agents
- Building agent dashboards

FastAPI

- Introduction to FastAPI
- REST API development
- GET and POST endpoints
- Pydantic validation
- Connecting agent functions
- Async endpoints
- Streaming responses
- API documentation

Gradio

- Introduction
- ChatInterface
- Interactive AI applications

TuxAcademy

Lab Activity:

- AI Agent Web Application
- Agent Backend API
- Interactive AI Chatbot



Module 11: Deployment, Testing and Production Engineering

Git and GitHub

- Repository management
- Branches and commits
- Project documentation
- Managing dependencies

Docker

- Containerization
- Images and containers
- Dockerfile
- Docker Compose
- Environment variables
- Containerizing AI applications

Deployment

- Streamlit Community Cloud
- Render
- GitHub integration
- Environment variables and secrets
- Deployment logs
- Cloud deployment fundamentals
- AWS, Azure and Google Cloud overview

Testing and Monitoring

- Pytest
- Postman
- Unit and integration testing
- Agent testing
- RAG evaluation
- LangSmith
- Execution traces
- Application logging
- Token and cost monitoring

Security

- API key management
- Authentication and authorization
- Prompt injection
- Tool permissions

TuXAcademy

- Data privacy
- Rate limiting
- Human approval for sensitive actions



Module 12: Industry Capstone Projects

Lab Activity 1: Enterprise Knowledge Assistant

- Document ingestion
- RAG
- Vector database
- Agent integration
- Streamlit interface
- Deployment

Lab Activity 2: Multi-Agent Research Assistant

- Research agent
- Analysis agent
- Report generation agent
- Agent orchestration
- Report export

Lab Activity 3: AI Business Automation

- Email processing
- Customer classification
- RAG-based information retrieval
- AI-generated responses
- Human approval
- Email integration
- Database storage

Lab Activity 4: Production-Ready AI Agent

- LangGraph agent
- Custom tools
- MCP integration
- FastAPI backend
- Streamlit frontend
- Docker
- Cloud deployment
- Testing and monitoring

TuxAcademy