QA TABLE OF CONTENTS

Foundations of Generative AI for QA Professionals

Duration: 16 Hrs

Training Description:

This entry-level course is designed for QA professionals to understand the fundamentals of Generative AI (GenAI) and how it complements and enhances QA workflows. Participants will explore how Large Language Models (LLMs)—such as GPT, Claude, Gemini, or other provider-agnostic AI tools—can streamline test case generation, error detection workflows, and defect reporting. Through hands-on practice and practical examples, participants will learn to leverage LLM-based tools to improve testing efficiency, enhance code quality, and automate repetitive QA processes.

Target Audience:

- Junior QA engineers, testers, and QA analysts seeking to understand and adopt Generative AI in their workflows.
- QA teams exploring entry-level automation of repetitive tasks such as test case generation or documentation review.
- QA employees with little/no AI experience who focus on manual testing or process optimization.

List of Tools, Frameworks, and Concepts Covered:

- **Generative AI Tools**: LLM-based APIs (e.g., GPT, Claude, Gemini) and conversational interfaces for automation.
- Testing Tools (AI-Integrated): Testim.io, Mabl, Selenium (basic AI-driven scenarios).
- **NLP and Automation** Concepts: Text summarization, text generation for test cases, LLM-based natural language models.
- Basic use **of Python or language-agnostic integration libraries** for testing and querying LLM-powered outputs.
- Hands-on practice with Prompt Engineering to customize LLM outputs effectively for QA-specific use cases.

What **Participants Can Expect After Completing This Training**:
Understand the role of Generative AI and LLMs in QA workflows and how they reduce repetitive manual processes.
Learn how to leverage LLMs to generate test cases, review documentation, and assist in testing tasks.
Acquire practical skills in basic AI-powered testing tools for automation.
Gain hands-on experience in crafting prompts and workflows for QA-relevant scenarios using flexible LLM-based tools.

Syllabus Breakdown for Course 1: Generative AI Foundations for QA

Module 1: Introduction to Generative AI for QA (2 Hours)

Objective:
Understand the basics of Generative AI and its relevance to QA.

1. **Content**:
- Overview of Generative AI: What it is and why it's important for QA workflows.
- Common applications in QA workflows: Generating test cases, bug reporting, and automating defect classification.
- Introduction to working with LLMs: How models like GPT, Claude, or Gemini enable automation in QA processes.

2. **Demo**:
- Use an LLM API to generate a structured test case from a plain-text testing requirement (e.g., login functionality).

3. **Hands-On Practice**:
- Participants provide sample testing requirements in plain text, build test cases using an LLM API, and refine the generated output to increase relevance (e.g., edge case scenarios for logins).

Module 2: Generating Test Cases and Scenarios Using LLMs (4 Hours)

Objective:

Learn how to generate comprehensive test scenarios and cases using Generative AI tools for dynamic workflows.

1. **Content:**
   - Writing effective prompts to create positive and negative test cases using LLMs.
   - Parameterizing test cases for various scenarios (e.g., variations in user input and environment setups).
   - LLM-powered Regression Testing for iterative updates.

2. **Demo:**
   - Generate multiple sets of test cases for an e-commerce cart feature, ensuring both functional and edge-case coverage using an LLM API.

3. **Hands-On Practice:**
   - Create test cases based on provided API documentation (e.g., testing a payment gateway).
   - Iterate prompts to refine test coverage and assess edge cases, such as invalid inputs or latency-related failures.

Module 3: Automated Review of QA Documentation (3 Hours)

Objective:
Enable QA professionals to automate the review and enhancement of technical and testing documentation.

1. Content:
- Summarizing Software Requirement Specification (SRS) documents to extract key testing criteria.
- Automating defect report analysis to identify common trends or patterns (e.g., grouping common bugs or categorizing defects by severity).
- Generating acceptance criteria and user stories dynamically from project narratives.

2. Demo:
- Use an LLM to analyze an SRS document and generate acceptance criteria and actionable testing scenarios.

3. Hands-On Practice:
- Summarize a sample specifications document provided by the instructor, extract actionable test criteria using LLM-provided insights, and cross-verify with functional requirements.

Module 4: Introduction to API Testing Using AI (3 Hours)

Objective:
Learn to automate API testing workflows using LLM-powered tools for payload generation, assertions, and validation.

1. **Content:**
   - Basics of REST API testing using tools like Postman.
   - Using LLMs to generate API test cases, including crafting payloads for requests and validating response assertions.
   - Best practices for automating response validation and dynamic payload generation with LLMs.

2. **Demo:**
   - Test an open API endpoint by generating payloads dynamically with an LLM and generating validation logic for JSON-based responses (e.g., response codes, key checks).

3. **Hands-On Practice:**
   - Automate testing for a sample API endpoint by querying LLM-generated payloads and validating edge cases like unexpected response structures or failure scenarios.

Module 5: Best Practices for Integrating LLMs into QA Workflows (4 Hours)

Objective:
Define efficient, ethical, and practical frameworks for incorporating Generative AI into QA tasks.

1. **Content:**
- Do's and Don'ts for leveraging LLMs in QA workflows: Benefits, pitfalls, and common usage mistakes.
- Understanding limitations of AI in QA: How to handle AI-generated false positives and negatives.
- Ethical considerations such as avoiding reliance on biased data, ensuring test results are inclusive, and monitoring performance improvements transparently.

2. **Hands-On Practice:**
- Analyze and compare two workflows: a traditional manual QA process vs an LLM-enhanced QA process. Identify performance,
- coverage improvements, and areas where human oversight remains necessary.

Advanced Generative AI for QA: Scalability and Automation

Duration: 20 Hrs

Training Description:
This advanced course builds upon foundational Generative AI knowledge to help QA professionals integrate Large Language Models (LLMs) such as GPT, Claude, Gemini, or other APIs for scalable and automated testing workflows. Participants will learn advanced testing techniques powered by LLMs, such as automated test pipeline orchestration, performance and load testing with AI tools, debugging AI-generated test workflows, and incorporating AI into CI/CD pipelines for continuous testing. The training emphasizes hands-on practice with real-world applications for enterprise-level QA scalability and advanced automation tasks across various LLM providers and tools.

Training Duration:
20 Hours (Delivered over 3 days, 6-7 hours/day)

Target Audience:

- Mid-level to senior QA professionals already familiar with basic Generative AI techniques, seeking to expand their expertise with advanced workflows and automation.
- QA engineers focusing on end-to-end testing workflows, aiming to scale their testing strategies with AI.
- Teams implementing CI/CD pipelines and DevOps practices for continuous testing and monitoring of software quality.

List of Tools, Frameworks, and Technologies Covered:

- **LLM APIs** (Provider-Agnostic): Integration of APIs such as GPT, Claude, Gemini, and other large language models for automation workflows.
- **QA Tools & Frameworks**: Selenium, Cypress, TestNG, and JUnit for orchestrating AI-driven functional and regression testing.
- Performance Testing Tools: BlazeMeter, LoadNinja, or JMeter augmented with AI for predictive load simulation.

- **CI/CD Automation Tools**: Jenkins, GitHub Actions, or CircleCI integrated with LLMs for continuous testing pipelines.
- **Monitoring and Observability**: Grafana, Prometheus, and other tools for real-time analysis of test workflows and performance metrics.

What Participants Can Expect After Completing This Training:

1. Learn advanced techniques for automating and scaling testing tasks with LLM-powered APIs (GPT, Claude, Gemini, etc.).
2. Integrate LLMs into test pipeline orchestration tools (e.g., Selenium or Cypress), generating dynamic test cases and automated validation workflows.
3. Employ LLMs for intelligent load and performance testing of enterprise applications to simulate real-world scenarios.
4. Build CI/CD pipelines for continuous testing workflows using AI-powered automation.
5. Debug and enhance LLM-powered workflows while adopting ethical best practices for minimizing bias and ensuring reliability.

Module 1: Advanced Test Case Generation and Orchestration Using LLMs (5 Hours)

Objective:
Learn how to create and orchestrate advanced test cases using LLM APIs across varying workflow scenarios.

Topics Covered:
Generating complex, parameterized test cases for edge cases and corner scenarios using LLMs.
Orchestrating regression tests, smoke tests, and integration tests dynamically with AI-based tools.
Leveraging multi-step task generation with LLMs to automate full test suites.

Demo:
Generate parameterized REST API tests dynamically using LLM APIs and validate their assertions.

Hands-On Practice:
Create matrix-based end-to-end test coverage for a login and authentication flow using LLM-generated input and output combinations for a range of scenarios (positive, negative, and boundary cases).

Module 2: Load and Stress Testing Using AI Tools (5 Hours)

Objective:
Simulate real-world application loads using AI to identify bottlenecks and analyze system resilience under high traffic.

Topics Covered:
Introduction to performance testing tools (e.g., BlazeMeter, LoadNinja) combined with LLMs for realistic test data generation.
AI-generated synthetic traffic patterns for load, stress, and spike testing.
Metrics and analysis: Capturing real-time performance insights under LLM-provided scenarios.

Demo:
Simulate sudden traffic spikes on an e-commerce website using BlazeMeter and AI-generated traffic patterns.

Hands-On Practice:
Participants simulate application traffic for an on-demand video streaming service, integrating LLM-generated payloads to create realistic usage patterns.

Module 3: Debugging and Enhancing LLM-Powered Test Pipelines (5 Hours)

Objective:
Understand how to debug and enhance LLM-powered QA workflows in CI/CD pipelines or standalone systems.

1. **Topics Covered:**

- Identifying gaps in AI-generated test cases (e.g., false positives, missed edge cases).
- Debugging integration failures in AI-driven pipelines and analyzing root causes.
- Using error output from LLM-generated test cases to iterate and improve prompts for higher-quality results.

2. Demo:

- Debug an AI-generated Selenium-based test that fails due to misconfigured locators, and fine-tune the generation pipeline to adjust for organizational testing standards.

3. Hands-On Practice:

- Participants analyze and fix pipeline issues in a Jenkins-driven CI environment powering an integration test suite, with failures logged via LLM analysis for remediation recommendations.

Module 4: CI/CD Testing Automation with AI-Powered Orchestration (5 Hours)

Objective:
Integrate AI-augmented QA workflows into continuous testing pipelines for DevOps workflows.

1. Topics Covered:

- Embedding LLM APIs into CI/CD pipelines, such as Jenkins, GitHub Actions, or CircleCI, to generate and validate test cases on the fly.
- Automate pipeline-based defect identification and regression testing using AI.
- Creating dynamic AI-assisted test scenarios that evolve based on real-time data, to ensure continuous improvement in coverage.

2. Demo:

- CI/CD pipeline generating AI-based regression suites for an e-commerce order management system and running seamlessly in Jenkins.

3. Hands-On Practice:

- Participants integrate an AI-assisted Cypress testing workflow into a GitHub Actions pipeline to continuously execute sanity testing during every build deployment.

Module 5: Ethical Considerations and Limitations in AI-Augmented Testing (2 Hours)

Objective:

Learn how to adopt ethical best practices and understand the limitations of AI augmentation in QA testing workflows.

1. Topics Covered:

- Addressing bias in LLM-generated test scenarios and ensuring inclusive testing practices.
- Handling common failure points: Adjusting LLMs to account for false positives, incomplete test cases, and hallucinated outputs.
- Defining clear acceptance criteria for AI-generated outputs to maintain quality and reliability.

2. Demo:

- Analyze two LLM-powered QA workflows: One biased towards common OS/browser configurations and another revised for broader compatibility.

3.Hands-On Practice:

- Review pre-generated API test outputs provided by an LLM and identify areas of bias or gaps in behavior validation to ensure compliance with ethical standards.