

GenAI for Cloud Engineer

Program Introduction and Rationale

Generative AI tools such as GitHub Copilot, ChatGPT, and Amazon Q are reshaping modern web development by automating code generation, debugging, and documentation. This program focuses on configuring and using these tools effectively within standard development environments like Visual Studio Code. It covers prompt engineering techniques, real-time coding assistance, and AI-powered explanations to enhance development efficiency.

This program also explores the use of Amazon Q for integrated code suggestions, command-line completions, and interaction with AWS tools. A comparison of popular AI assistants, including free options like Google Gemini, provides insight into selecting the right tool for specific development needs. The content emphasizes practical strategies to incorporate GenAI responsibly while maintaining human oversight and code quality.

Key outcomes:

- Apply prompt engineering techniques to generate, debug, and explain code efficiently.
- Use GitHub Copilot and ChatGPT to build RESTful APIs
- Handling data access using copilot
- Using Amazon Q in Visual Studio Code
- Apply best practices through Amazon Q
- Using Amazon Q's AWS-aware coding capabilities.
- Adopt responsible practices for using GenAI tools

Prerequisites:

- Ability to write code in modern programming languages (Program will use Node.js).
- Familiarity with core programming logic, data structures, and algorithms.
- Experience developing RESTful services (preferably with Express.js)
- Exposure to using GitHub as a version control system.
- Basic understanding of IDEs, especially Visual Studio Code.
- Familiarity with AWS and storage service

Course : GenAI for Cloud Engineer

Indicative Design and Content Coverage

Duration: 8 Hours (Virtual)

Module	Function	Duration in Hrs
Module 1: ChatGPT & Effective Prompting	<ul style="list-style-type: none"> Orientation and agenda setting Capabilities & limits of ChatGPT Prompt engineering basics (intent, context, examples) Live demos: converting vague requests into precise prompts Rules for effective written prompts to Copilot Prompting via comments and inline instructions, context Activity : AI-Powered chatbots (ChatGPT, Gemini) 	1
Module 2: GitHub Copilot & Agents	<ul style="list-style-type: none"> Revisiting GitHub Copilot capabilities. Customizing GitHub Copilot's output with guidelines and coding standards. Install & configure VS Code and required extensions Configure workspace, repos, and basic Git flow Copilot Chat walked through (chat UI + code context) Agents: @vscode, @terminal, @workspace — Demo Practical Git tips, navigating codebases Develop Node JS programs, generate tests, and refactor with AI 	2
Module 3: GenAI for Backend Developers	<ul style="list-style-type: none"> Generating DB models and seeding scripts instantly Avoiding bias in AI-generated logic. Generate optimized query endpoints using Copilot. Activity : Use copilot and ChatGPT to build a Node.js + Express backend project 	1.5
Module 4: Amazon Q for Developers	<ul style="list-style-type: none"> Introduction to Amazon Q Developer Configuring IDE Using Amazon Q for chat, inline suggestions and scanning Working with error deduction and debug complex code Demo using index workspace, run scans Interaction with AWS tools using Amazon Q Activity : Retrieve AWS Service Data using Amazon Q 	1.5
Module 5: Capstone Project	<p>Objective</p> <p>create a customer service-based REST API using Node.js + Express, while leveraging GitHub Copilot, ChatGPT, and Amazon Q for code generation, debugging, AWS integration, and best practice recommendations.</p> <p>Task Overview</p> <ul style="list-style-type: none"> Use Copilot to implement Get/Post endpoints of REST API Use Google Gemini (or another free tool) to write the search function. Use Amazon Q's code scanning to detect and fix the issue. Use Amazon Q for code snippet to upload files to S3. Auto-generate API documentation from code comments using Amazon Q 	2

Learning Pedagogy

The pedagogic model of the program is focused on learning in remote virtual learning mode. Expert mentors shall work with students through the program. Learning is in an environment that combines the convenience of online access with the intensity of mentoring.

The model combines the following elements:

1. Virtual Instructor-led Live connects: These work on a fixed schedule with recorded versions available to people who miss them.
 - Sessions that provide context.
 - Sessions with expert-led demonstrations that provide step-by-step guidance on critical tasks.
 - Sessions with hands-on workshops where participants actively work through practice exercises and challenges.
 - Sessions that involve group discussions.
 - Sessions that explain best practices.
 - Sessions that explain common pitfalls/issues.
 - Sessions that discuss success stories, case studies and real-world scenarios that provide insight into the practical challenges and solutions.
2. In addition, on-demand participant support will be provided via team collaboration tools such as Slack.