STACK ROUTE

AN NIIT VENTURE

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 equips developers to set up and configure these tools within their environments and apply them effectively in real-world projects. Participants will learn to craft precise prompts for code generation, debugging, and explanation, while leveraging Copilot to streamline coding tasks and boost productivity.

Key outcomes:

- Apply prompt engineering techniques to generate, debug, and explain code efficiently.
- Set up and configure GitHub Copilot, ChatGPT, Google Gemini, and Amazon Q
- Streamline coding workflows by leveraging Copilot and Amazon Q
- 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

Duration: 8 Hours

Indicative Design and Content Coverage

Module	Function	Duration in Hrs
Module 1: ChatGPT & Effective Prompting	• Need of AI-assisted coding • Real-life examples: AI in everyday life (Netflix, Google Maps, ChatGPT, Alexa) • GitHub & OpenAI account creation • Installing VSCode + extensions • Connect to ChatGPT/alternative • Capabilities & limits of ChatGPT • Understand Prompts, its types, components of an effective prompt • Key strategies to write effective prompts • Prompting via comments and inline instructions, context • Common pitfalls to avoid. • Compare ChatGPT & Gemini • Prompt ChatGPT to Generate a function, convert and explain Activity Prompts for financial analysis, risk management, investment analysis, regulatory compliance, fraud detection using ChatGPT	1
Module 2: GitHub Copilot & Agents	• Copilot overview and architecture • Install GitHub Copilot in IDEs • Enable and test Copilot Chat functionality • Generate complex algorithms through prompting • Application using context utilization techniques • Generate API integration code	2

	• Understanding @vscode, @terminal, @workspace agents • Codebase navigation strategies Activity Build an application that calculates loan interest and manages customer transactions using copilot and its agents	
Module 3: GenAI for Backend Developers	• GenAI tools for Node.js ecosystem enhancement • Express.js development acceleration techniques • Generate Express app boilerplate using AI • Package.json optimization with AI suggestions • Folder structure generation for scalable architecture • Set up middleware structure with AI assistance • Generate RESTful routes • Create authentication middleware using AI • Create input validation and sanitization • Generate logging and CORS middleware • Create database connection configuration • Build model relationships and validations Activity Build an "Event Management API" using only AI assistance including user authentication, event CRUD operations, real-time notifications, and basic analytics - demonstrating mastery of GenAI-assisted Node.js development	2.5
Module 4: Amazon Q for Developers	• Introduction to Amazon Q Developer • Configuring IDE • Amazon Q capabilities and benefits for developers • AWS account configuration and IAM permissions • Amazon Q subscription and access setup • Understanding pricing tiers and usage limits • Amazon Q sidebar navigation and features • Chat interface overview and interaction methods • Basic chat functionality with simple queries • Working with code suggestion capabilities • Integration with VSCode features • Understanding context and conversation continuity • Chat history management and organization • Enable workspace indexing in Amazon Q • Understanding indexed content scope and limitations • Available AWS services within Amazon Q • CloudFormation template generation and management • Lambda function development assistance • Using Amazon Q to explain unfamiliar code patterns • Generate documentation Activity Build a Node.js expense tracker using Amazon Q to generate functions, explain code, and index the workspace for intelligent assistance. Use Q’s scanning feature to detect and fix errors or security concerns, ensuring a clean and secure project.	2.5

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.