GenC HR

SPE Java FSE - Angular Stage 2 Plus (GenC Next) - Handbook

Why do we need this Full Stack Engineering Program?

Full Stack Prep-up program engages young talents with a comprehensive learning pathway, giving these millennials an opportunity to become a Full Stack Engineer, understand the corporate environment and groom themselves even before they join us.

Cognizant emphasizes on Learner Autonomy where students take charge of their own learning pathway, with the available tools and resources. More focus is given to “learning” than “teaching”. Get ready to embark your own learning adventure!

Program at a glance

Full Stack Prep-up Internship Program has 3 stages:

- Stage 2 - Best Practices and Foundations of Backend Development
- Stage 3 - Backend Development and Microservices with Spring Boot and Spring Cloud
- Stage 4 - Frontend Development and Cloud Technologies
- Integrated Development Project (IDP)

Program Highlights

- The complete learning journey is formalized using adult learning principles, where problem solving and applying the skills gained are given more importance than conceptual learning.
- Get mentored by Subject Matter Experts, whose motivation and guidance will help you accelerate in the learning journey.
- Higher order framework concepts would be dealt with complete Trainer support in Instructor Led training mode.

cognizant

Service Lines

Service lines can simply be defined as a modern organizational structure strategy for resource planning and allocation for any size of business. Typically, traditional organizational structure models are more vertically aligned -- think of an employee who has several bosses in the hierarchical ladder before being directly under the company's owner or president. Conversely, service lines follow a more horizontal continuum approach, where the company is strategically segmented into more manageable departments. The service line approach tends to focus more on the requirements of customers, which often results in noticeable increases in the customer satisfaction rate.

What is Application Development?

Application development goes through a process of planning, creating, testing, and deploying an information system, also known as the software development lifecycle. Applications are also often developed to automate some type of internal business process or processes, build a product to address common business challenges, or drive innovation.

What is Application Maintenance?

Application maintenance is the continuous updating, analyzing, modifying, and re-evaluating of your existing software applications. Application maintenance must be an ongoing task to ensure your applications are always running to the best of their abilities. Due to evolving customer expectations, the fight to survive in an existing market, and technological advancements, modifying and implementing new strategies is critical in maintaining sustainability and staying competitive. Every competitive business needs to constantly enhance and manage the IT solutions that have been developed in order to stay relevant and meet the wavering needs of users. This is where application maintenance and support come into the picture.

Contrary to popular belief, application maintenance is not just about fixing defects, but modifying a software product after delivery to correct faults, as well as to improve performance. Application maintenance and enhancement to existing applications begin with a thorough study of existing applications to identify areas of improvement.

Tips for Successfully Carrying Out Application Development and Maintenance

Great user experience to end customers through the development and maintenance of modern apps is a must-have. Today, applications (web or mobile) are the most cost-effective and powerful ways to reach out to a vast market and generate revenues. With millions of applications being rolled out every day, it's a good idea to keep in mind a few tips:

- Be as clear as possible as to what your requirements for your application are
- Thoroughly understand the services offered by application development companies and identify the right partner if you're using a partner
- Evaluate the various development platforms and choose the one that best fits the needs of your business

- Make sure to embed processes that focus on continuous improvements and iterations to add new features and/or fix bugs
- When developing your application, make security your top priority
- Regularly update and test your application to deliver improved and better performance, high security, and a bug-free, seamless user experience

What is Digital Engineering (DE)?

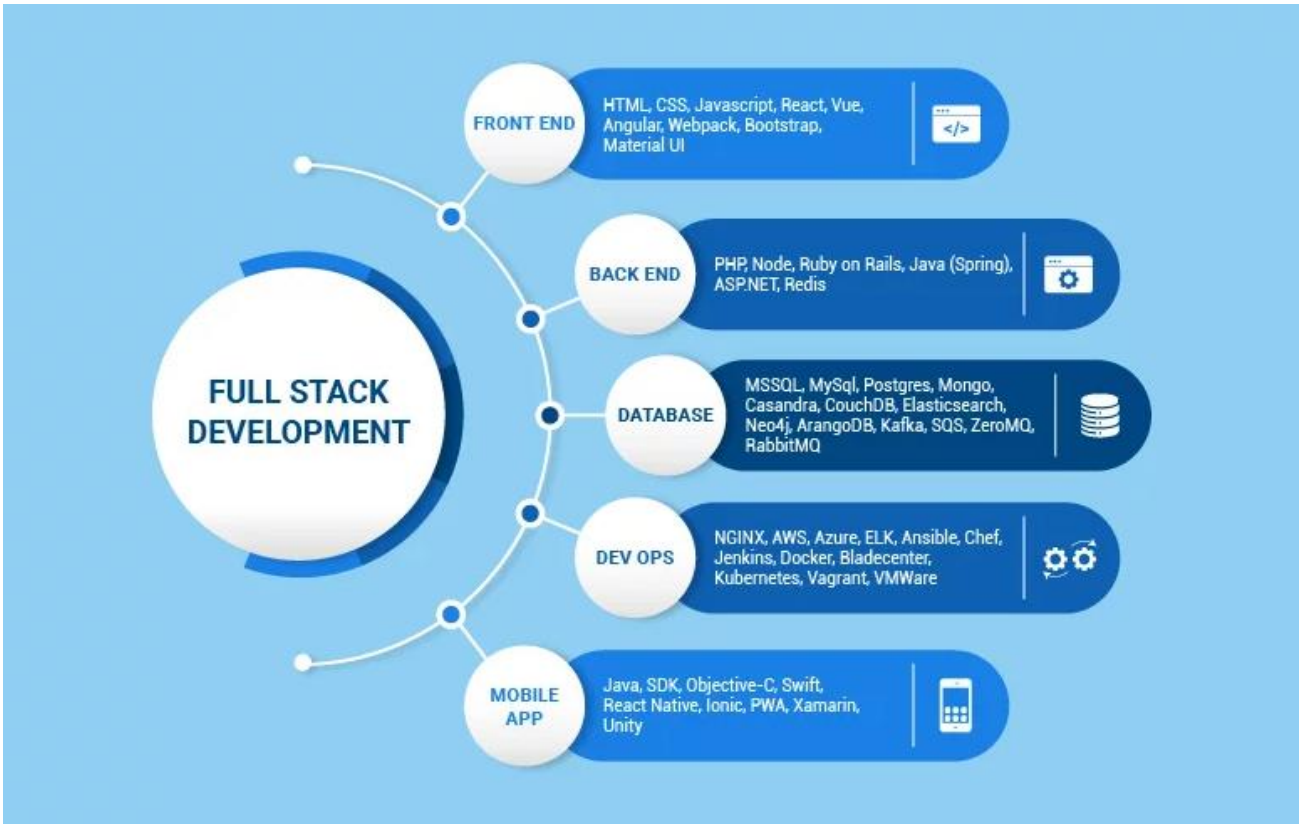
Digital Engineering is the practice in which new applications are conceived and delivered. Encompassing the methodologies, utility, and process of creating new digital products end to end, digital engineering leverages data and technology to produce improvements to applications—or even entirely new solutions.

What are the business benefits of Digital Engineering?

When digital engineering principles are applied to software development, **Software Product Engineering (SPE)** is the result. SPE involves all stages of product creation: design, development, testing and deploying. With SPE, design and engineering teams work together to improve business outcomes. The focus shifts from features and backlogs to better user experiences and performance.

What is Full Stack Development?

Full Stack Development (FSD) is a software development process that includes both the front and back end. To that end, a Full Stack Developer may design and create the front end while simultaneously designing, developing, and debugging databases and the software's backend. There are two significant components to full-stack application development. Development of the Front End and Back End.



Roles and Responsibilities of a Full Stack Developer

A full stack developer is responsible for both the front-end and back-end aspects of a web application. The specific roles and responsibilities can vary depending on the size of the development team and the complexity of the project, but some common responsibilities include:

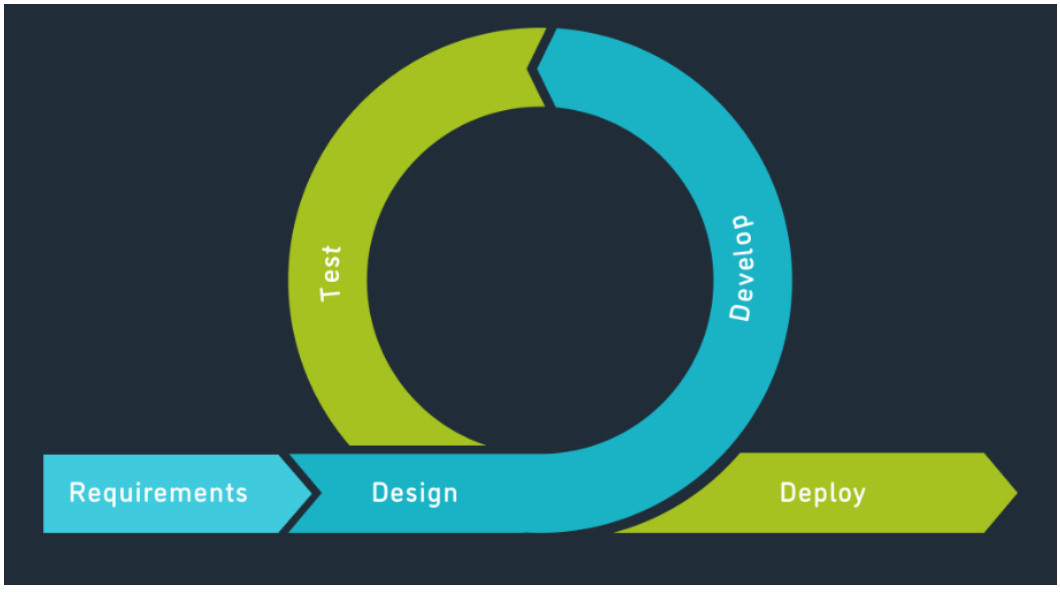
1. Design and develop end-to-end web applications.
2. Implement front-end and back-end components using relevant technologies (e.g., HTML, CSS, JavaScript, React, Node.js, etc.).
3. Write clean, efficient, and well-documented code.
4. Debug and resolve technical issues.
5. Collaborate with the team and other stakeholders to deliver project on time.
6. Stay up-to-date with the latest technologies and industry trends.
7. Write automated tests to ensure code quality and improve application reliability.
8. Develop and maintain databases, servers and application deployment infrastructure.
9. Manage code repositories and version control systems (e.g., Git).
10. Participate in code reviews to ensure high-quality code.
11. Contribute to the architecture and design of applications.
12. Collaborate with designers, product managers, and other stakeholders to understand the requirements and build solutions that meet them.

Integrated Development Project (IDP)

What is Integrated Development Project (IDP)?

Integrated Development Project is an approach wherein the learner experiences the entire software development processes in an incremental fashion as part of the GenC Training. The IDP implementation is purely based on **Agile Software Development** methodologies and inspired from **PBL (Project-Based Learning)** which is learning while doing. It gives learners the opportunity to gain a deeper understanding of a topic through problem-solving using real-world examples and challenges.

Following is the Agile Development Methodology at high-level.

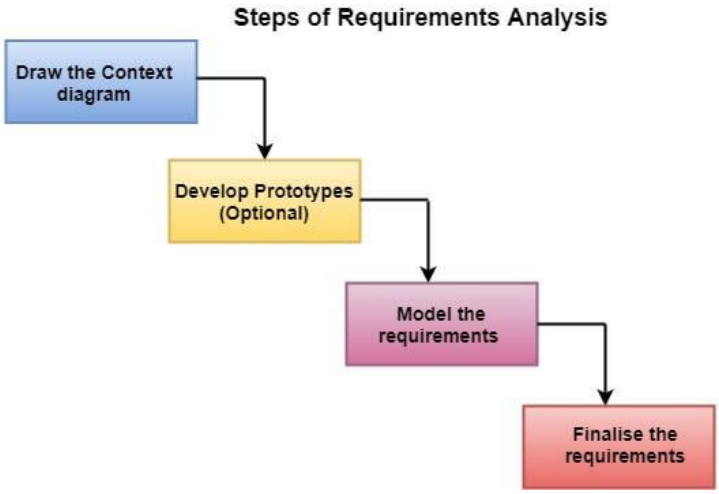


Stages of IDP

Following are the three seminal phases of IPD.

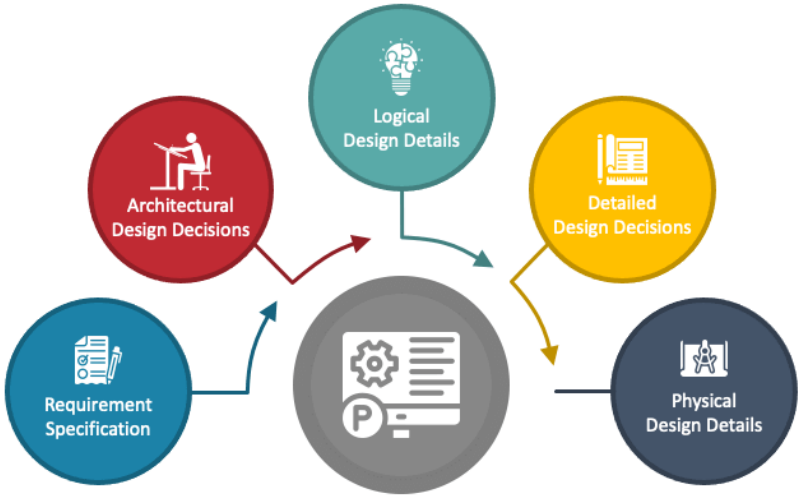
Phase 1: Requirement Analysis

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.



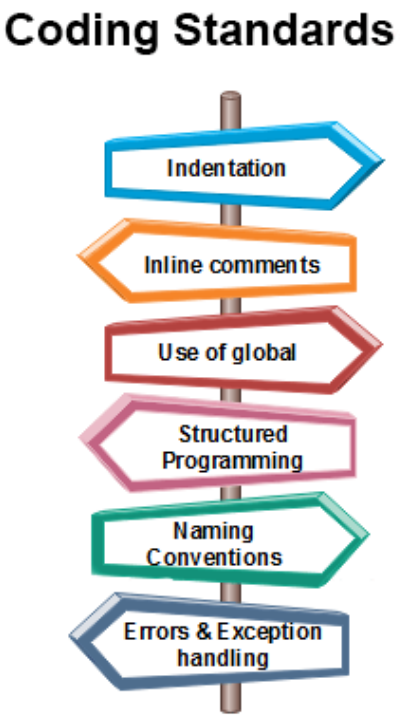
Phase 2: Project Design

Project design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.



Phase 3: Project Development

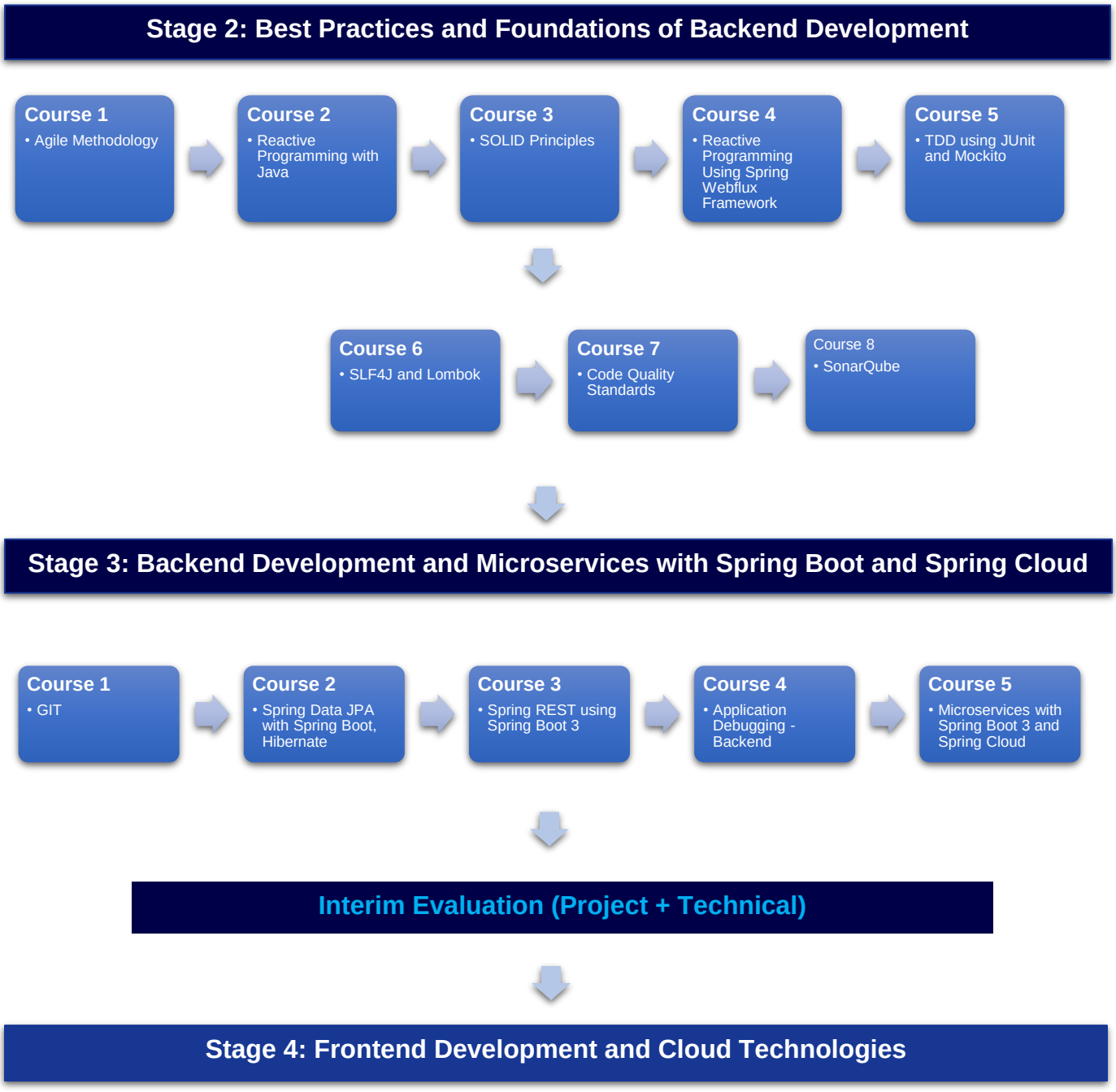
Once the system design phase is over, the next phase is development. In this phase, developers start build the entire system by writing code using the chosen programming language. In this phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

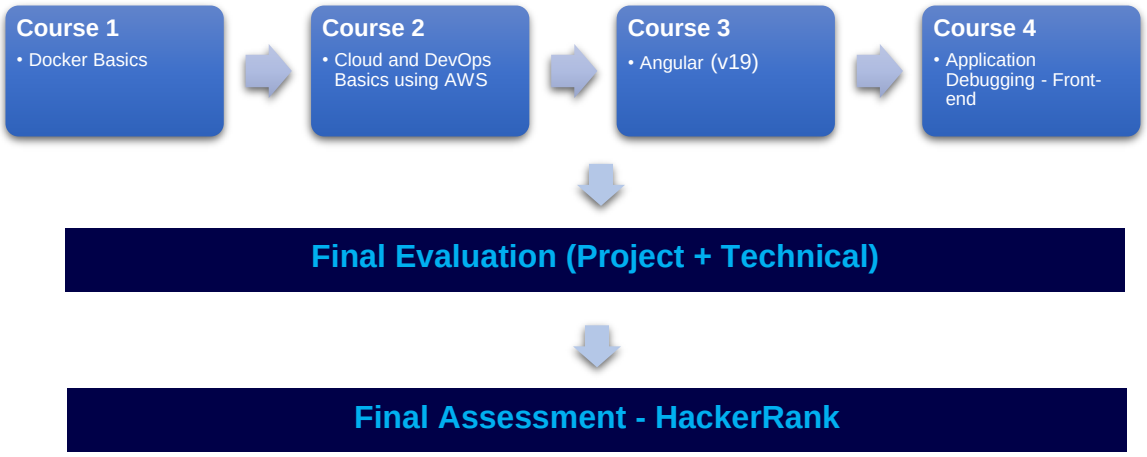


Recommended Program Sequence

The learning journey starts with **5 days of Icebreaker sessions, 1 day of Agile Workshop** followed by a technical learning that contains **3 stages,** and they are the following:

- **Stage 2** - Best Practices and Foundations of Backend Development
- **Stage 3** - Backend Development and Microservices with Spring Boot and Spring Cloud
- **Stage 4** - Frontend Development and Cloud Technologies
- Integrated Development Project (IDP)





All three stages will be executed in the traditional classroom model, and the GenCs will practice through a learning path configured on the **Tekstac** platform.

There will be an integrated project called **IDP (Integrated Development Project)**, which will be executed incrementally and is part of all three stages.

Key Learning and Evaluation Components of the Program

Self-Learning via Udemy

Cognizant has collaborated with **Udemy** to provide world class learning videos for the evolving future of work. These Udemy programs are woven into a learning path, empowering you to plan and learn at your style.

The program also connects you with **Subject Matter Experts (SMEs)** to get the professional guidance on your queries in the learning journey.

The program doesn't ONLY concentrate on the technical skilling, but also on the shaping up of the behavioral skills. **39 hours of Behavioral learning** would be done in ILT mode, with few Self-paced learning modules too.

Continuous Evaluation

GenCs will undergo various evaluations throughout their training. At the conclusion of Stage 3, an **interim evaluation** will be conducted to assess learners on the technical skills covered up to that point and their progress on project deliverables. Towards the end of Stage 4, a **final evaluation** will take place, encompassing the entire scope of the training. After completing the training, a Skill-Based Assessment will be conducted on the **HackerRank** platform to evaluate GenCs' capabilities in programming and database skills.

Program Completion Criteria

RAG as PHS (Performance Health Status)

The program continuously evaluates if you can apply those self-learnt skills to solve a real-time business problem. Depicted below are the two key learning components, which are distributed across the learning journey for the purpose of continuous evaluation.

Interim Evaluation:

During the interim evaluation, the GenC will undergo a video interview on the learning platform. This interview will be conducted by a Tech SME from the BU. The purpose of this evaluation is to assess the GenCs' knowledge and understanding of the skills covered in the training program up to the halfway point. It also encompasses an evaluation of the GenCs' progress in their Integrated Development Project (IDP). The evaluation will involve a technical discussion as well as an assessment of the IDP progression to gauge the GenCs' proficiency in the skills learned thus far.

Final Evaluation:

For the final evaluation, the GenC will participate in a video interview conducted by a Tech SME from the BU. This evaluation aims to assess the GenC's knowledge and expertise in all the skills covered throughout the entire training program. Like the interim evaluation, this assessment will involve a technical discussion via a video interview on the learning platform, along with a project evaluation to assess the GenC's capabilities and their IDP's progress. It serves as a comprehensive evaluation of the GenC's skills and capabilities acquired during the training.

The above evaluation components will attribute to the **Performance Health Status (PHS)** of a GenC. Additional Learning Components like Hands-On, Quizzes, CCs, and ICTs will help you to enhance your expertise level.

Mandatory Hands-On Exercise Completion

- Completion of 100% of the hands-on exercises is mandatory to qualify for the Interim, and Final evaluations.

Icebreaker

Icebreaker session will be conducted for a duration of initial **5 days**. During the session, various topics related to Corporate Induction, Talent Management, Cognizant Agenda on Core Values, Leader Talks, Alumni, BU Mentor connects will be covered. Followed by icebreaker, technical training will kick start.

Following sessions will be covered during the 5 days of icebreaker

- Corporate Induction
- Talent Manager Connect
- Cognizant Agenda Sessions on Core Values

- Leader Talks (Academy) and many more…

Learning Recommendation

A recommended day-wise schedule is provided below for the learning, with the learning content for the day, the practice hands-on and extended hands-on to be done for the day or any other activities are listed.

How and From Where to Learn?

- Udemy learnings are recommended in the Platform to understand the fundamental concepts. In addition to this, you can also learn from any other sources as they are mentioned in this handbook.

Stage 2: Best Practices and Foundations of Backend Development

Overview

Stage 2 of the training focuses on advanced skills essential for software development. It covers Design Patterns and Principles, providing GenCs with a solid understanding of best practices for designing reusable and maintainable software solutions. The module on Data Structures and Algorithms equips GenCs with the knowledge needed to efficiently manipulate and store data, crucial for developing efficient algorithms. Spring Core and Maven are introduced to teach GenCs about dependency injection, aspect-oriented programming, and project management with Maven. The module on TDD using JUnit and Mockito emphasizes the importance of test-driven development in ensuring code quality and reliability. GenCs also learn about SLF4J and Lombok for logging and reducing boilerplate code. Lastly, the module on Code Quality Standards and SonarQube teaches GenCs how to maintain high-quality code by following industry standards and using tools like SonarQube for code analysis and quality assurance.

As part of **Stage 2** of your training, the following skills will be covered.

- Agile Methodology
- Reactive Programming with Java
- SOLID Principles
- Reactive Programming Using Spring Webflux Framework
- TDD using JUnit and Mockito
- SLF4J and Lombok
- Code Quality Standards
- SonarQube

How and From Where to Learn?

- You can learn from the sources as they are mentioned in this learning guide.

Course Overview

In the **Course 1** of the **Stage 1**, learners will be introduced to the basics of **Agile methodology**. Agile is an approach to project management and software development that emphasizes flexibility, collaboration, and customer satisfaction. It involves adaptive planning, iterative development, early delivery, and continuous improvement. Agile methodologies, like Scrum and Kanban, focus on delivering value to the customer and responding to change effectively.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand the principles and values of Agile methodology.
- Describe the benefits of using Agile in software development.
- Explain the differences between Agile and traditional project management approaches.
- Identify the key roles and responsibilities in Agile teams.
- Describe the iterative and incremental nature of Agile development.
- Explain the importance of customer collaboration and feedback in Agile.
- Describe common Agile practices, such as user stories, sprints, and retrospectives.
- Identify common Agile frameworks, such as Scrum, Kanban, and Extreme Programming (XP).
- Explain how Agile principles can be applied in different project environments.

Day 1

Agile Methodology

Key Topics: Introduction to Agile, Agile Manifesto, Scrum Framework, Agile Estimation and Planning, Agile User Stories, Agile Metrics and Reporting

Continuous Learning: Technical Enablement

Agile Crash Course: Agile Project Management; Agile Delivery

- Learn All Sections in this Udemy course.

Course Overview

The **Course 2** of the **Stage 2** provides a comprehensive guide to reactive programming in Java, covering essential features from **Java 8+ and enhancements up to Java 21**. It explores the principles of reactive programming, the Reactive Streams specification, and **Project Reactor**. By the end of this course, participants will have a thorough understanding of reactive programming concepts and be able to implement reactive systems using Java.

Learning Objectives

After completing this course, GenCs will be able to:

- Master functional programming with lambda expressions and method references, utilize the Streams API for data processing, handle null values gracefully with the Optional class, and understand default and static methods in interfaces.
- Learn about syntax enhancements, streamlined IO, new String methods, switch expressions, text blocks, records, pattern matching, finalized records and sealed classes, UTF-8 default, virtual threads preview, finalized virtual threads, and new collections API.
- Differentiate between imperative and reactive programming, understand synchronous vs. asynchronous execution, and learn about blocking vs. non-blocking calls.
- Understand the core components of reactive streams: Publisher, Subscriber, Subscription, Processor, and backpressure handling.
- Work with Mono and Flux, the two core reactive types, and learn how to create, transform, and subscribe to streams.
- Utilize transformation operators (map(), flatMap(), concatMap()), filtering operators (filter(), distinct(), take(), skip()), combining operators (merge(), zip(), concat(), combineLatest), and error handling strategies (onErrorResume(), onErrorContinue(), retry).
- Understand and implement backpressure strategies, including buffering, dropping, and latest strategies.
- Use different schedulers (Schedulers.parallel(), Schedulers.elastic(), Schedulers.single) to manage threading in reactive applications.
- Conduct unit testing with StepVerifier and mock reactive streams.
- Use logging and debugging tools (log(), doOnNext(), doOnError) and understand common pitfalls in reactive programming.

Day 2, 3 - Forenoon

Reactive Programming with Java

Key Topics: Core Java Enhancements

Continuous Learning: Technical Enablement

Modern Java: Mastering Features from Java 8 to Java 21

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 2:** Getting Started Modern Java
  - **Section 5:** Why Java 8
  - **Section 6:** Introduction to Lambda
  - **Section 7:** Lambdas and Functional Interfaces
  - **Section 8:** Constructor and Method References
  - **Section 10:** Streams API
  - **Section 11:** Streams API - Operations
  - **Section 12:** Streams API - Factory Methods
  - **Section 13:** Streams API - Numeric Streams
  - **Section 14:** Streams API - Terminal Operations
  - **Section 15:** Streams API - Parallel Processing
  - **Section 19:** Base Project Setup for Exploring Java 9 and Beyond Features
  - **Section 20:** Local Variable Type Inference (LVTI) using var
  - **Section 21:** Text Blocks
  - **Section 22:** Enhanced Switch
  - **Section 23:** Records
  - **Section 24:** Sealed Classes/Interfaces
  - **Section 25:** Pattern Matching
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Student Score Info
- Vintage Books Emporium
- Animalia
- Area and Volume
- The Perfect Password
- Mystic Line

Day 3 - Afternoon

Reactive Programming with Java

Key Topics: Introduction to Reactive Programming

Continuous Learning: Technical Enablement

 Mastering Java Reactive Programming [From Scratch]

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 1:** Introduction
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Day 4, 5 - Forenoon

Reactive Programming with Java

Key Topics: Reactive Programming with Project Reactor

Continuous Learning: Technical Enablement

 Reactive Programming in Modern Java using Project Reactor

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 4:** Section 4: Getting Started with Project Reactor
  - **Section 7:** Lets create our very first Flux and Mono
  - **Section 8:** Transforming Flux and Mono
  - **Section 9:** Combining Flux and Mono
  - **Section 12:** Exception/Error Handling in Flux and Mono
  - **Section 13:** Implement Exception Handling in Movies Reactive Service
  - **Section 14:** Retry, Repeat using retry(), retryWhen(), repeat()
  - **Section 17:** BackPressure
  - **Section 18:** Explore Data Parallelism in Project Reactor
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the hands-on given in the Udemy course while you learn.

 Do not copy paste the code. Write the code yourself.

Day 5 - Afternoon

Reactive Programming with Java

Key Topics: Advanced Concepts in Reactive Programming

Continuous Learning: Technical Enablement

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 7: [OPTIONAL] -** Threading & Schedulers
  - **Section 14:** Unit Testing With Step Verifier
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the hands-on given in the Udemy course while you learn.

Do not copy paste the code. Write the code yourself.

Day 6 - Forenoon

- This part of the day has been allocated to accommodate the duration of behavioral training and cohort mentor connect sessions.

Stage 2 -> Course 3 -> SOLID Principles

Course Overview

The **Course 3** of the **Stage 2** introduces you to the foundational principles of Object-Oriented Programming (OOP) through the lens of the SOLID principles. These principles, widely regarded as best practices in software design, ensure robust, maintainable, and scalable code. By understanding "What" these principles entail and "Why" they are essential, you will develop the skills necessary to design and implement software that adheres to high standards of quality and efficiency.

Through an exploration of the Single Responsibility Principle, the Open-Closed Principle, the Liskov Substitution Principle, the Interface Segregation Principle, and the Dependency Inversion Principle, you will learn how to create solutions that are flexible and resistant to design flaws.

Learning Objectives

After completing this course, GenCs will be able to:

- Explain what the SOLID principles are and why they are critical for effective software design.
- Design classes and modules that have a single, clear purpose to improve code readability and maintainability.
- Write code that can be extended without modifying the existing implementation, ensuring adaptability to future requirements.
- Ensure subclasses can replace their parent classes without disrupting the application’s functionality.
- Create interfaces that are specific to the needs of clients, avoiding the pitfalls of bloated or unnecessary functionalities.

- Invert dependencies to rely on abstractions rather than concrete implementations, promoting decoupled and testable code.
- Integrate SOLID principles into your development process to create maintainable and scalable applications.

Day 6 - Afternoon, Day 7 - Forenoon

SOLID Principles

Key Topics: SOLID Principles

Continuous Learning: Technical Enablement

Design Patterns in Java

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 2:** SOLID Design Principles
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the hands-on given in the Udemy course while you learn.

Do not copy paste the code. Write the code yourself.

Day 7 - Afternoon

- This part of the day has been allocated to accommodate the duration of behavioral training and cohort mentor connect sessions.

Stage 2 -> Course 4 -> Reactive Programming Using Spring WebFlux Framework

Course Overview

The **Course 4** of the **Stage 2** provides an in-depth exploration of reactive programming using the Spring **WebFlux** framework. It covers fundamental concepts of reactive programming, the differences between reactive and imperative programming, and the core components of Project Reactor. Participants will learn how to create **reactive REST APIs**, handle backpressure, and implement performance tuning strategies. By the end of this course, participants will be equipped to build high-performance, scalable, and responsive applications using Spring WebFlux.

Learning Objectives

After completing this course, GenCs will be able to:

- Grasp the core principles of reactive programming and differentiate it from imperative programming.
- Understand the Reactive Streams specification, including Publisher, Subscriber, Subscription, and Processor.
- Learn about Mono and Flux, the two core reactive data types, and how to handle backpressure and demand.
- Compare blocking (Spring MVC) and non-blocking (WebFlux) approaches and understand their use cases.
- Set up and configure a Spring Boot project with WebFlux.
- Utilize key Reactor operators such as map(), flatMap(), and filter() for data transformation and filtering.
- Develop reactive REST APIs using @RestController and handle various HTTP methods.
- Integrate reactive relational databases using Spring Data R2DBC and ReactiveCrudRepository.
- Work with MongoDB in a reactive manner using ReactiveMongoRepository and stream data with tailable cursors.
- Implement backpressure strategies like buffering, throttling, and windowing, and apply rate limiting in WebFlux.
- Optimize the performance of high-throughput APIs using various tuning techniques.
- Implement SSE in WebFlux, compare WebSockets and SSE, and handle real-time data streaming.
- Implement global exception handling, retry and timeout mechanisms, and the Circuit Breaker pattern with Resilience4J.
- Ensure system resilience and graceful degradation in reactive systems.
- Write unit tests with StepVerifier, perform integration testing using WebTestClient, and simulate streaming and backpressure scenarios.

Day 8

Reactive Programming Using Spring Webflux Framework

Key Topics: Introduction to Reactive Programming, Setting Up Spring WebFlux, Reactive Data Access (MongoDB & R2DBC)

Continuous Learning: Technical Enablement

Reactive Programming with Spring Framework 5

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 2:** Introduction to Reactive Programming
  - **Section 4:** Spring Web Client
  - **Section 5:** Spring Webflux
  - **Section 6:** R2DBC
  - **Section 9:** Reactive MongoDB Application
- Ensure that you learn these topics through self-learning and practice alongside the

course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the hands-on given in the Udemy course while you learn.

Do not copy paste the code. Write the code yourself.

Day 9

Reactive Programming Using Spring Webflux Framework

Key Topics: Handling Backpressure & Performance Optimization

Continuous Learning: Technical Enablement

- Backpressure Mechanism in Spring WebFlux
- Spring WebFlux Optimization

Hands-On

Try out the example programs given in the above links.

Do not copy paste the code. Write the code yourself.

Day 10

Reactive Programming Using Spring Webflux Framework

Key Topics: Event-Driven Programming with SSE & WebSockets

Continuous Learning: Technical Enablement

Reactive Applications with Spring WebFlux Framework

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 16:** Server-Sent Events(SSE) in Reactive Spring WebFlux application
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Go through the below learning.

- Handling Streaming Data with Webflux
- Real-Time Communication with WebSocket in Spring Boot

Hands-On

Try out the example programs given in the above links and the Udemy course.

Do not copy paste the code. Write the code yourself.

Day 11

Reactive Programming Using Spring Webflux Framework

Key Topics: Exception Handling & Resilience Patterns, Testing Reactive Applications

Continuous Learning: Technical Enablement

Reactive Applications with Spring WebFlux Framework

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 9:** Error Handling in Reactive Spring WebFlux applications
  - **Section 18:** Testing the Web Layer in Reactive Spring WebFlux Applications
  - **Section 19:** Unit Testing Service Layer in Reactive Spring WebFlux application
  - **Section 20:** Testing Data Layer in Reactive Spring WebFlux application
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the hands-on given in the Udemy course while you learn.

Do not copy paste the code. Write the code yourself.

Stage 2 -> Course 5 -> TDD using JUnit and Mockito

Course Overview

The **Course 5** of the **Stage 2** provides a comprehensive introduction to modern testing practices essential for building high-quality software. It is designed to equip learners with the skills and knowledge to implement **Test-Driven Development (TDD)** effectively and to use powerful testing

frameworks such as **JUnit** and **Mockito**. Through hands-on practice, learners will gain expertise in testing Java applications, leveraging advanced features of JUnit, and creating mock objects to isolate dependencies. Special focus is placed on testing Spring-based applications and managing external dependencies to ensure robust, maintainable, and scalable software solutions.

Learning Objectives

After completing this course, GenCs will be able to:

- Grasp the principles, benefits, and workflow of TDD in the software development lifecycle.
- Set up and use JUnit for writing and executing unit tests in Java applications.
- Leverage advanced JUnit features to create more expressive and efficient tests.
- Understand the fundamentals of Mockito for mocking objects and isolating dependencies in unit testing.
- Apply JUnit and Mockito to test Spring-based applications, ensuring the reliability of components and configurations.
- Implement effective mocking strategies to simulate external dependencies and test the application in isolation.

Day 12

TDD using JUnit and Mockito

Key Topics: Introduction to Test-Driven Development (TDD), Getting Started with JUnit, Advanced JUnit Features

Continuous Learning: Technical Enablement

Learn Java Unit Testing with Junit & Mockito in 30 Steps

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 2:** Unit Testing with JUnit
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- EB Connection
- CarcoExpress
- Employee Details
- Employee Appraisal

Day 13

TDD using JUnit and Mockito

Key Topics: Mockito Basics, Testing Spring Applications with JUnit and Mockito, Mocking External Dependencies

Continuous Learning: Technical Enablement

Spring Boot Unit Testing with JUnit, Mockito and MockMvc

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 5:** Unit Testing - Mocking with Mockito
  - **Section 7:** Testing Spring Boot MVC Web Apps - Database Integration Testing
  - **Section 8:** Testing Spring Boot MVC Web Apps - MVC Controller Testing
  - **Section 12:** Testing Spring Boot REST APIs
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Employee Address Update
- Computation Service
- Employee Details (Mockito)
- Crisp Placement Training

Code Challenges (For Practice Only)

Attempt the following Code Challenges through the Learning Path for checking your skill level on TDD and JUnit. There will be only 3 attempts and you have to score 70% in order to clear this challenge.

Do not copy paste the code. Write the code yourself.

- Assess-Type-1: Code Challenge - TDD, Junits

Course Overview

The **Course 6** of the **Stage 2** provides an in-depth understanding of SLF4J and Lombok, two powerful libraries that simplify and enhance Java development. You will explore the logging capabilities of SLF4J and its role as a façade for various logging frameworks, along with practical techniques for efficient logging and integration with popular tools. Additionally, the course delves into Lombok's ability to reduce boilerplate code through annotation-based code generation, covering essential features such as getter and setter methods, constructors, and builder patterns. The course equips learners with the skills to streamline code, improve maintainability, and leverage these tools effectively in real-world projects.

Learning Objectives

After completing this course, GenCs will be able to:

- Explain the key features and benefits of SLF4J as a logging façade.
- Compare SLF4J with other logging frameworks, including Log4j and Java Util Logging (JUL).
- Demonstrate how to add and manage SLF4J dependencies using Maven and Gradle.
- Integrate SLF4J with existing logging frameworks and popular libraries like Logback.
- Utilize the Logger interface and its methods to implement effective logging practices.
- Apply best practices for parameterized logging and avoid performance pitfalls such as string concatenation.
- Describe the benefits of Lombok in reducing boilerplate code and improving developer productivity.
- Integrate Lombok seamlessly with popular IDEs like IntelliJ IDEA and Eclipse.
- Leverage Lombok annotations such as @Getter, @Setter, @ToString, and @EqualsAndHashCode for efficient code generation.
- Construct classes effectively using annotations like @NoArgsConstructor, @AllArgsConstructor, @RequiredArgsConstructor, and @Builder.
- Utilize advanced annotations like @Slf4j for logging, @Value for immutability, and @With for immutability-based methods.
- Configure and troubleshoot Lombok in build tools (Maven, Gradle) and resolve common annotation processing issues.
- Debug and analyze Lombok-generated code to ensure alignment with project requirements.

Day 14

SLF4J and Lombok

Key Topics: Introduction to SLF4J, Setting Up SLF4J, Core Concepts of SLF4J, Introduction to Lombok, Key Lombok Annotations, Advanced Lombok Features, Lombok and IDEs

Continuous Learning: Technical Enablement

Java Best Practices for Efficient, Scalable, and Secure Code

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 23:** Logging in Java
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Learning Reference

Practice the sample programs and code snippets provided in the following links.

- Lombok @Log4j, @Slf4j and Other Log Annotations

Stage 2 -> Course 7 -> Code Quality Standards

Course Overview

The **Course 7** of the **Stage 2** is designed to provide a comprehensive understanding of code quality and its significance in modern software development. GenCs will explore best practices, industry standards, and tools to ensure clean, maintainable, and secure code. The curriculum focuses on fostering discipline in coding practices, improving team collaboration through effective code reviews, and leveraging tools to measure and maintain high-quality code standards. By integrating static code analysis, peer review techniques, and industry-approved metrics, learners will gain hands-on experience in writing and evaluating code that meets professional benchmarks.

Learning Objectives

After completing this course, GenCs will be able to:

- Explain the significance of maintaining high standards in code quality.
- Identify the benefits and common challenges in achieving quality assurance in code.
- Implement naming conventions, formatting standards, and effective documentation.
- Organize code using best practices for package and import statements.
- Gain proficiency in configuring and running tools like Checkstyle, PMD, and SpotBugs.
- Analyze and interpret results from static code analysis tools to improve code quality.
- Conduct thorough and constructive code reviews using tools such as GitHub Pull Requests, Gerrit, and Crucible.
- Promote team collaboration by following best practices for peer reviews.
- Calculate and interpret key metrics such as cyclomatic complexity, code duplication, and maintainability index.
- Use tools like SonarQube and IntelliJ IDEA to measure and monitor code quality, and set appropriate thresholds for quality assurance.
- Integrate OWASP Secure Coding Guidelines for Java into development practices.
- Align with ISO/IEC standards for code quality to ensure compliance with industry

benchmarks.

Day 15 - Forenoon

Code Quality Standards

Key Topics: Introduction to Code Quality Standards, Java Coding Conventions, Static Code Analysis Tools, Code Reviews, Code Quality Metrics, Compliance and Industry Standards

Continuous Learning: Technical Enablement

Java Best Practices for Efficient, Scalable, and Secure Code

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 21:** Best Practices of Secure Coding in Java
  - **Section 25:** Best Practices of REST Architecture in Java Applications
  - **Section 26:** Best Practices of Tracking Software Development Performance, Clean Code & others
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Learning Reference

- What is Code Quality?
- Naming Conventions in Java
- Java Static Analysis Tools in Eclipse and IntelliJ IDEA
- 30+ Java Code Review Checklist Items
- Continuous Code Quality using SonarQube

Day 15 - Afternoon

- This part of the day has been added to adjust the behavioral training and mentor connect.

Stage 2 -> Course 8 -> SonarQube

Course Overview

The **Course 8** of the **Stage 2** offers a comprehensive introduction to **SonarQube**, an industry-leading platform for continuous code quality inspection. Designed for developers, architects, and DevOps professionals, the course delves into the critical role of SonarQube in identifying technical

debt and improving software maintainability, reliability, and security. Participants will explore the tool's architecture, installation methods, and key functionalities such as setting up quality gates, scanning codebases, and integrating SonarQube with CI/CD pipelines and popular development tools. By the end of the course, learners will be equipped to effectively use SonarQube to enhance code quality and streamline the development process.

Learning Objectives

After completing this course, GenCs will be able to:

- Define SonarQube, its purpose, and the benefits of continuous code quality inspection.
- Identify and address technical debt using SonarQube.
- Recognize the supported programming languages and the platform's architectural components.
- Perform on-premises and Docker-based installations of SonarQube.
- Configure databases and initiate the SonarQube server.
- Access and navigate the SonarQube web interface.
- Manage projects and dashboards within SonarQube.
- Configure quality profiles and gates to enforce coding standards.
- Detect and resolve issues, code smells, and technical debt.
- Utilize the SonarQube scanner to analyze codebases.
- Integrate SonarQube with CI/CD pipelines, build tools, and IDEs.
- Measure and interpret code coverage, duplication, maintainability, reliability, and security metrics.
- Generate detailed reports to track and communicate code quality improvements.

Day 16

SonarQube

Key Topics: Introduction to SonarQube, Installing and Setting Up SonarQube, Key Concepts and Features, Integrating SonarQube into Development, Metrics and Reporting

Continuous Learning: Technical Enablement

Continuous Code Inspection with SonarQube

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **ALL** Sections
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Walk through the following videos to practice SonarQube.

- SonarQube - Part 1
- SonarQube - Part 2
- SonarQube - Part 3
- SonarQube - Part 4
- SonarQube - Part 5

- SonarQube - Part 6

Code Challenges (For Practice Only)

Attempt the following Code Challenges through the Learning Path for checking your skill level on Code Quality. There will be only 3 attempts, and you have to score 70% in order to clear this challenge.

Do not copy paste the code. Write the code yourself.

- Assess-Type-1: Code Challenge - Code Quality

Integrated Development Project (IDP)

Day 17

IDP - Requirement Analysis and Design

- This day will be used for IDP related activities.

Stage 3 - Backend Development and Microservices with Spring Boot and Spring Cloud

Overview

The **Stage 3** of the training delves into advanced backend development skills with a focus on Java and Spring Boot. GenCs will learn about using GIT for version control to manage source code effectively. They will also gain expertise in debugging backend applications, learning techniques to identify and resolve issues efficiently. The training covers Spring Data JPA with Spring Boot and Hibernate, enabling GenCs to interact with relational databases seamlessly. Additionally, they will master building RESTful web services using Spring Boot 3, a framework highly regarded for Java-based applications. The stage culminates with a deep dive into microservices architecture, where GenCs will learn to design and implement microservices using Spring Boot 3 and Spring Cloud, providing them with the necessary tools for building and deploying microservices-based applications.

As part of the **Stage 3** of your training, the following skills will be covered.

- GIT
- Application Debugging - Backend
- Spring Data JPA with Spring Boot, Hibernate
- Spring REST using Spring Boot 3
- Microservices with Spring Boot 3 and Spring Cloud

How and From Where to Learn?

- You can learn from the sources as they are mentioned in this learning guide.

Course Overview

In the **Course 1** of the **Stage 3**, you will dive into the fundamental skill of version control using **Git**. Git is a distributed version control system that allows you to track changes to your codebase, collaborate with other developers, and manage different versions of your project. You will learn the basics of Git, including setting up a repository, committing changes, branching and merging, resolving conflicts, and working with remote repositories. Understanding Git is essential for any developer, as it provides a solid foundation for managing and collaborating on projects effectively.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand the concept of version control and its importance in managing project files and code changes efficiently.
- Gain a comprehensive understanding of Git, including its features, advantages, and how it differs from other version control systems.
- Learn how to set up Git on your local machine, including configuring user details and initializing a new Git repository.
- Familiarize yourself with basic Git commands such as git add, git commit, and git status to track changes and manage your repository.
- Learn the concepts of branching and merging in Git to manage different versions of your code and collaborate with team members effectively.
- Understand the concept of remote repositories and learn how to work with remote repositories in Git for collaboration and backup purposes.
- Learn how to collaborate with team members using Git, including pushing and pulling changes to and from remote repositories, resolving conflicts, and reviewing code.

Day 18

GIT

Key Topics: Introduction to Version Control, Understanding Git, Setting Up Git, Basic Git Commands, Branching and Merging, Remote Repositories, Collaborating with Git

Continuous Learning: Technical Enablement

The Git & Github Bootcamp

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 2:** Introducing…Git!
    - **Section 3:** Installation & Setup
    - **Section 4:** The Very Basics Of Git: Adding & Committing
    - **Section 5:** Commits In Detail (And Related Topics)

- **Section 6:** Working With Branches
- **Section 7:** Merging Branches, Oh Boy!
- **Section 8:** Comparing Changes With Git Diff
- **Section 14:** Git Collaboration Workflows

- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Git Config
- Clone Repo
- Add, Commit And Push
- Pull And Merge
- Merge - Resolve Conflict
- Git Tags

Stage 3 -> Course 2 -> Spring Data JPA with Spring Boot, Hibernate

Course Overview

In the **Course 3** of the **Stage 3**, you will delve into Spring Data JPA integration with Spring Boot 3, leveraging Hibernate as the JPA provider to streamline data access layer development in Spring applications. This integration simplifies database interactions by abstracting underlying data access technologies and providing a repository-based programming model. You will explore entity mapping using JPA annotations, repository interfaces for CRUD operations, custom query methods for data retrieval, derived queries from method names, and features like pagination, sorting, auditing, and transaction management. By the end of this milestone, you will have a comprehensive understanding of how to harness the power of Spring Data JPA with Spring Boot 3 and Hibernate to build efficient and robust data access layers for your applications.

Learning Objectives

After completing this course, GenCs will be able to:

- Explain the role of Hibernate as a JPA provider in Spring applications.
- Understand the benefits of using Spring Data JPA to abstract data access logic.
- Map entity relationships such as One-to-One, One-to-Many, and Many-to-Many.
- Perform CRUD operations (Create, Read, Update, Delete) using repositories.
- Write native and JPQL queries to retrieve specific data sets.
- Set up and configure auditing features to track entity creation and updates.
- Understand the importance of transaction management in data operations.
- Configure and manage transactions using @Transactional annotations.
- Handle database-related exceptions gracefully using Spring mechanisms.
- Write unit tests for repositories using in-memory databases (H2, HSQL).

- Optimize query performance with caching and batch fetching techniques.

Day 19

Spring Data JPA with Spring Boot, Hibernate

Key Topics: Introduction to Spring Data JPA, Setting Up a Spring Boot Project with Spring Data JPA

Continuous Learning: Technical Enablement

Hibernate and Spring Data JPA: Beginner to Guru

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 2:** Introduction to Spring Data JPA
    - **Section 5:** Hibernate with MySQL
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **Spring Data JPA**.

Day 20

Spring Data JPA with Spring Boot, Hibernate

Key Topics: Entity Mapping, Spring Data Repositories, CRUD Operations with Spring Data JPA

Continuous Learning: Technical Enablement

Hibernate and Spring Data JPA: Beginner to Guru

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 16:** Database Relationship Mapping
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Click on the link provided and delve into the content to enhance your understanding.

- JPA Repositories
- Spring Boot JPA CRUD with Example

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- School Strength
- Loan Lenders
- Sales Statistics
- Account with ATM Card With Logger

Day 21

Spring Data JPA with Spring Boot, Hibernate

Key Topics: Query Methods and Named Queries, Pagination and Sorting, Auditing with Spring Data JPA, Spring Data JPA Projections

Continuous Learning: Technical Enablement

Hibernate and Spring Data JPA: Beginner to Guru

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 13:** Spring Data JPA Queries
    - **Section 14:** Paging and Sorting
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Click on the link provided and delve into the content to enhance your understanding.

- Auditing with JPA, Hibernate, and Spring Data JPA
- Spring Data JPA Projections

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Account Transaction

Day 22, Day 23 - Forenoon

Spring Data JPA with Spring Boot, Hibernate

Key Topics: Spring Data JPA and Spring Boot Integration, Spring Data JPA and Hibernate

Continuous Learning: Technical Enablement

Click on the link provided and delve into the content to enhance your understanding.

- How to use Spring Data JPA in Spring Boot Project
- Spring Data JPA vs Hibernate

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Country - State(OneToMany Bi-Directional)
- Component Mapping

Code Challenges (For Practice Only)

Attempt the following Code Challenges through the Learning Path for checking your skill level on user Spring Data JPA. There will be only 3 attempts and you have to score 70% in order to clear this challenge.

Do not copy paste the code. Write the code yourself.

- Assess-Type-1: Code Challenge - Spring Data JPA with Spring Boot

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **Spring Data JPA**.

Day 23 - Afternoon

- This part of the day has been allocated to accommodate the duration of behavioral training and cohort mentor connect sessions.

Course Overview

The **Course 3** of the **Stage 3** covers **Spring REST using Spring Boot 3**. In this milestone, participants will learn about building RESTful web services using the Spring Framework, specifically focusing on the latest version, Spring Boot 3. Spring Boot is a popular framework for rapidly developing and deploying Java-based web applications.

Learning Objectives

After completing this course, GenCs will be able to:

- Grasp the fundamentals of REST, including statelessness, resource-based interactions, and HTTP methods (GET, POST, PUT, DELETE).
- Set up a Spring Boot project and build REST controllers to handle HTTP requests and return JSON responses.
- Manage path variables, query parameters, request bodies, response headers, and implement exception handling using Spring annotations.
- Use strategies like path/version headers and content negotiation to manage API versions and support multiple media types (e.g., JSON, XML).
- Implement authentication and authorization, including JWT-based token authentication, and manage CORS policies.
- Apply DTOs to decouple data models from entities and customize JSON serialization/deserialization.
- Integrate Actuator to expose metrics and health endpoints, and configure custom monitoring and security.
- Write unit and integration tests with JUnit, Mockito, and MockMvc, and generate API documentation using Swagger/OpenAPI.

Day 24

Spring REST using Spring Boot 3

Key Topics: Introduction to Spring REST and Spring Boot 3, Building a Simple REST Controller

Continuous Learning: Technical Enablement

Building Real-Time REST APIs with Spring Boot - Blog App

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 3:** REST Basics and Key Concepts (For Beginners)
    - **Section 4:** Spring Boot REST API Development Basics - Important Annotations
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Restaurant Menu Service
- Movie Ticket Booking REST Controller App

Day 25

Spring REST using Spring Boot 3

Key Topics: Request and Response Handling, RESTful Resource Representation with DTOs

Continuous Learning: Technical Enablement

Building Real-Time REST APIs with Spring Boot - Blog App

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 6:** Building CRUD REST API's for Post Resource
    - **Section 9:** Using ModelMapper-Map Entity to DTO and Vice Versa
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Online Event Management - Get and Post
- Online Event Management - Put and Delete
- Patient Management - Exception Handling
- Inventory Management System - Get and Post
- Inventory Management System - Put and Delete
- Online Recipe Management - Exception Handling

Day 26

Spring REST using Spring Boot 3

Key Topics: RESTful HATEOAS, Content Negotiation and Media Types

Continuous Learning: Technical Enablement

Click on the link provided and delve into the content to enhance your understanding.

- Building a Hypermedia-Driven RESTful Web Service
- Spring Boot REST Web Services Content Negotiation

Day 27

Spring REST using Spring Boot 3

Key Topics: Spring Boot Actuator for REST Monitoring, Security and Authentication in RESTful APIs

Continuous Learning: Technical Enablement

Click on the link provided and delve into the content to enhance your understanding.

- Spring Boot Actuator
- Authentication and Authorization in Spring Boot 3.0 with Spring Security

Day 28 - Forenoon

Spring REST using Spring Boot 3

Key Topics: Testing RESTful APIs, Documenting RESTful APIs

Continuous Learning: Technical Enablement

Building Real-Time REST APIs with Spring Boot - Blog App

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 18:** REST API Documentation using SpringDoc OpenAPI in Spring Boot 3
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Click on the link provided and delve into the content to enhance your understanding.

- Spring Boot Unit Testing REST APIs Tutorial

Code Challenges (For Practice Only)

Attempt the following Code Challenges through the Learning Path for checking your skill level on Spring REST using Spring Boot. There will be only 3 attempts and you have to score 70% in order to clear this challenge.

Do not copy paste the code. Write the code yourself.

- Assess-Type-1: Code Challenge - Spring REST using Spring Boot

Course Overview

In the **Course 4** of the **Stage 3**, you will delve into the essential skill of debugging Spring Boot applications, particularly focusing on REST APIs. Debugging is a critical aspect of software development, allowing you to identify and fix issues in your code efficiently. You will learn how to set up debugging in a Spring Boot application using popular IDEs like IntelliJ IDEA or Eclipse, enabling you to step through your code, inspect variables, and understand the flow of execution.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand the role of debugging in improving code quality.
- Set up debugging in Spring Boot using IntelliJ IDEA or Eclipse.
- Use breakpoints to pause and control code execution.
- Step through code to analyze method calls and flow.
- Inspect and modify variable states during execution.
- Debug REST API endpoints and HTTP requests.
- Capture and diagnose exceptions and errors.
- Apply logging techniques alongside debugging.
- Develop a systematic approach to debugging.

Day 28 - Afternoon

Application Debugging - Backend

Key Topics: Debugging Fundamentals, Debugging REST Endpoints, Debugging Asynchronous Code

Demo Videos

Watch the following demo videos provided in the platform for getting a complete picture about how to debug backend REST API using Eclipse.

Note: The below given Cognizant Kpoint videos can be accessed even after your Tekstac platform access is revoked.

- Debugging REST API-Part1
- Debugging REST API-LogFiles

Additional Learning

Eclipse Debugging Techniques And Tricks

- Go through the entire course.

Click on the links provided and delve into the content to enhance your understanding.

- Debugging Spring Applications
- Spring Boot – @Async Annotation

Integrated Development Project (IDP)

Day 29

Internal Demo and Rework

- This day can be utilized for Internal Demo & Re-work

Stage 3 -> Course 5 -> Microservices with Spring Boot 3 and Spring Cloud

Course Overview

The **Course 5** of the **Stage 3** covers **Microservices with Spring Boot 3 and Spring Cloud**. This milestone focuses on introducing participants to building microservices-based applications using Spring Boot 3 and Spring Cloud. It covers fundamental concepts of microservices architecture, including modularity, scalability, resilience, and decentralized data management. Participants will learn how Spring Boot 3 simplifies the development of microservices with its auto-configuration and opinionated defaults. Additionally, they will explore how Spring Cloud complements Spring Boot by providing tools for building distributed systems, such as service discovery, circuit breakers, and distributed tracing. The milestone includes practical examples and hands-on exercises to reinforce learning.

Learning Objectives

After completing this course, GenCs will be able to:

- Identify the characteristics, advantages, challenges, and appropriate use cases for microservices.
- Set up service discovery and registration with Netflix Eureka.
- Implement authentication and authorization for RESTful APIs in a microservices environment.
- Configure centralized authentication using OAuth 2.1, OIDC, and JWT-based secure communication.
- Use Spring Cloud Feign for declarative REST clients and explore service orchestration patterns.

- Implement circuit breakers and fallback mechanisms with Spring Cloud Circuit Breaker for fault tolerance.
- Set up API Gateway with Spring Cloud Gateway for routing, filtering, and load balancing.
- Manage dynamic and externalized configurations using Spring Cloud Config.
- Use Spring Boot Actuator for endpoint monitoring and integrate tools like Prometheus and Grafana for metrics.
- Implement role-based access control (RBAC), secure inter-service communication, and enforce security policies.

Day 30

Microservices with Spring Boot 3 and Spring Cloud

Key Topics: Introduction to Microservices Architecture (MSA), Spring Cloud for Microservices

Continuous Learning: Technical Enablement

[NEW] Building Microservices with Spring Boot & Spring Cloud

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 10:** Microservices Introduction
    - **Section 11:** Building Microservices
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Day 31

Microservices with Spring Boot 3 and Spring Cloud

Key Topics: Microservices Communication with Spring Cloud, API Gateway and Edge Services

Continuous Learning: Technical Enablement

[NEW] Building Microservices with Spring Boot & Spring Cloud

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 13:** Microservices Communication
    - **Section 15:** API Gateway using Spring Cloud Gateway
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Register Order App in Spring Cloud
- Access Order App Via Router
- Inter- Microservice Communication UserApp and ProductApp
- Service Fallback

Day 32

Microservices with Spring Boot 3 and Spring Cloud

Key Topics: Spring Cloud Config

Continuous Learning: Technical Enablement

[NEW] Building Microservices with Spring Boot & Spring Cloud

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 16:** Centralized Configurations using Spring Cloud Config Server
    - **Section 17:** Auto Refresh Config Changes using Spring Cloud Bus
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- E-Commerce Platform

Day 33

Microservices with Spring Boot 3 and Spring Cloud

Key Topics: Monitoring and Metrics in Microservices

Continuous Learning: Technical Enablement

Click on the link provided and delve into the content to enhance your understanding.

- Monitoring Spring Boot Microservices with Prometheus
- Monitoring Made Simple: Empowering Spring Boot Applications with Prometheus and Grafana

Day 34 - Forenoon

Microservices with Spring Boot 3 and Spring Cloud

Key Topics: Security Best Practices in Microservices

Continuous Learning: Technical Enablement

Click on the link provided and delve into the content to enhance your understanding.

- Role-Based Access Control with Spring Security
- Learn How to Secure Service-to-Service Microservices

Code Challenges (For Practice Only)

Attempt the following Code Challenges through the Learning Path for checking your skill level on Microservices. There will be only 3 attempts and you have to score 70% in order to clear this challenge.

Do not copy paste the code. Write the code yourself.

- Assess-Type-1: Code Challenge - Microservices

Integrated Development Project (IDP)

Day 34 - Afternoon, 35, 36, 37 - Forenoon

Microservices Implementation

- These days will be spent on the following IDP development activity.
    - Implement Microservices using Spring Cloud

Day 37 - Afternoon

- This part of the day has been allocated to accommodate the duration of behavioral training and cohort mentor connect sessions.

Interim Evaluation (Project + Technical)

Day 38, 39

Interim Evaluation (Project + Technical)

- Interim evaluation will be conducted on these days, and the mode will be a video interview on the Tekstac platform. There will be ONLY one attempt.

Stage 4 - Frontend Development and Cloud Technologies

Overview

Stage 4 deals with the Niche skills such as Angular, Docker, Cloud and DevOps basics using GCP which are seminal while developing/maintaining various software applications. We provide unique learning experience to learners by including diversified learning content and learning methodologies that are based on adult learning principles.

As part of **Stage 4** of your training, the following skills will be covered.

- Docker basics
- Cloud and DevOps Basics using AWS
- Angular (v19)
- Application Debugging - Front-end

How and From Where to Learn?

- You can learn from the sources as they are mentioned in this learning guide.

Stage 4 -> Course 1 -> Docker Basics

Course Overview

The **Course 1** of the **Stage 4** introduces learners to **Docker**, a popular containerization platform used in modern application development. Docker allows developers to package applications and their dependencies into containers, which can run consistently across different environments. The module covers key concepts such as Docker containers, images, volumes, and networking. Learners will understand the benefits of Docker, including improved portability, scalability, and efficiency in resource utilization. They will also learn how to create, manage, and deploy Docker containers, making it easier to build, ship, and run applications in a consistent and isolated environment.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand Containerization Concepts and Docker Architecture
- Develop Proficiency in Working with Docker
- Explore Dockerfile and Image Building
- Manage Container Communication and Persistence
- Use Docker Compose for Multi-Container Applications
- Integrate Docker with REST API and Databases

Day 40

Docker Basics

Key Topics: Introduction, Docker Commands. Docker Run, Docker Images, Docker Compose

Continuous Learning: Technical Enablement

Docker for the Absolute Beginner - Hands On - DevOps

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 1:** Introduction
  - **Section 2:** Docker Commands
  - **Section 3:** Docker Run
  - **Section 4:** Docker Images
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Day 41

Docker Basics

Key Topics: Docker Registry, Docker Engine, Docker Storage, Docker Networking, Container Orchestration

Continuous Learning: Technical Enablement

Docker for the Absolute Beginner - Hands On - DevOps

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 6:** Docker Registry
  - **Section 7:** Docker Engine, Storage and Networking
  - **Section 9:** Container Orchestration - Docker Swarm & Kubernetes
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Course Overview

The **Course 2** of the **Stage 4** training focuses on cloud and DevOps practices using AWS. This milestone covers topics related to cloud computing, including the delivery of computing services over the internet, and AWS-specific services for computing power, storage, content delivery, and more. It also delves into DevOps practices, which aim to automate and integrate software development and IT operations processes for faster deployment and continuous delivery. The milestone includes hands-on exercises and practical scenarios to reinforce learning in these areas.

Learning Objectives

After completing this course, GenCs will be able to:

- Differentiate between cloud and on-premises data centers, including key pros and cons.
- Explain the characteristics of IaaS, PaaS, and SaaS.
- Identify the differences among public, private, hybrid, and community cloud models.
- Understand the core offerings of AWS, Azure, and GCP.
- Recognize AWS services like EC2, S3, RDS, and their functionalities.
- Launch and configure Amazon EC2 instances with AMIs, security groups, and key pairs.
- Manage storage using S3, EBS volumes, and snapshots for backups.
- Manage users, groups, and roles in AWS IAM with policies and MFA.
- Design VPCs with subnets, route tables, and security groups for network security.
- Utilize AWS CodeCommit, CodeBuild, CodeDeploy, and CodePipeline for CI/CD workflows.
- Implement infrastructure as code (IaC) and automate deployments with AWS Lambda and API Gateway.

Day 42

Cloud and DevOps Basics using AWS

Key Topics: Introduction to Cloud Computing, Cloud Service Models, Cloud Deployment Models, Cloud Service Providers, Advantages of Cloud Computing and various services provided by AWS, Introduction to AWS, AWS Compute Services, AWS Storage Services

Continuous Learning: Technical Enablement

Introduction to Cloud Computing on AWS for Beginners [2024]

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course
  - **Section 3:** IT Fundamentals
  - **Section 4:** Cloud Computing Concepts
  - **Section 6:** Amazon EC2, Auto Scaling, and Load Balancing
  - **Section 7:** AWS Storage Services
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Day 43

Cloud and DevOps Basics using AWS

Key Topics: AWS Networking, AWS Database Services, AWS Identity and Access Management (IAM), AWS Serverless Computing

Introduction to Cloud Computing on AWS for Beginners [2024]

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course
  - **Section 5:** AWS Access Control and Networking
  - **Section 8:** AWS Database Services
  - **Section 11:** Containers and Serverless Computing
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Day 44

Cloud and DevOps Basics using AWS

Key Topics: Introduction, AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline

DevOps , CI/CD(Continuous Integration/Delivery for Beginners

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course
  - **Section 1:** The Basics – CI, CD and DevOps
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

AWS CodePipeline Step by Step

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course
  - **Section 1:** Introduction - AWS CodePipeline Basics
  - **Section 2:** Using AWS CodeCommit Repositories as Source Locations
  - **Section 3:** Building Your Code With AWS CodeBuild
  - **Section 4:** Deploying to EC2 Instances With AWS CodeDeploy
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Day 45

- This day has been included to adjust the behavioral training and cohort mentor connect sessions.

Stage 4 -> Course 3 -> Angular (v19)

Course Overview

The **Course 3** of the **Stage 4** provides a comprehensive introduction to building modern, dynamic web applications using **Angular** and **Node.js**. Designed for aspiring web developers, the curriculum covers foundational concepts and advanced topics, empowering learners to create robust and scalable applications. The course emphasizes TypeScript, Angular components, directives, forms, state management, and routing, as well as server-side programming with Node.js. Learners will also explore reactive programming with RxJS and gain hands-on experience with API integrations, testing strategies, and asynchronous programming. By the end of the course, participants will have a holistic understanding of front-end and back-end development using Angular and Node.js.

Learning Objectives

After completing this course, GenCs will be able to:

- Set up and configure the Angular development environment for building web applications.
- Master TypeScript essentials to write and manage Angular code efficiently.
- Develop and utilize Angular components, directives, and pipes to create modular and reusable UI elements.
- Implement Angular forms with validation for capturing and managing user input.
- Leverage dependency injection and services for efficient data sharing and application logic organization.
- Configure and use Angular routing and navigation to enable seamless user experiences in multi-page applications.
- Explore advanced router features for managing application flow and guarding routes.
- Integrate HTTP client functionality to interact with RESTful APIs and handle asynchronous data.
- Understand and apply state management principles for managing complex application states.
- Adopt reactive programming techniques with RxJS for managing asynchronous data streams.
- Write and execute tests to ensure the reliability and quality of Angular applications.
- Gain a foundational understanding of Node.js for server-side programming.
- Work with core Node.js modules to handle file systems, events, and streams.
- Understand asynchronous programming in Node.js to manage non-blocking operations effectively.

Day 46

Angular (v19)

Key Topics: Introduction to Angular and Setting Up Environment, TypeScript Essentials for Angular

Continuous Learning: Technical Enablement

Angular - The Complete Guide (2024 Edition)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course
    - **Section 1:** Getting started
    - **Section 18:** The Basics [Angular < 16]
    - **Section 46:** Bonus: TypeScript Introduction (For Angular 2 Usage)
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **Angular**.

Day 47

Angular (v19)

Key Topics: Angular Components, Directives and Pipes

Continuous Learning: Technical Enablement

Angular - The Complete Guide (2024 Edition)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 21:** Components & Databinding Deep Dive [Angular < 16]
    - **Section 23:** Directives Deep Dive [Angular < 16]
    - **Section 33:** Using Pipes to Transform Output [Angular < 16]
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- My First - Sports Club App
- Boat Ride Entrance Fee
- Show Student DB - ngFor Directive
- Remove Student - (component, ngFor)
- Flight Reservation Form - (Dynamic dropdown using ngFor)
- Change Cases – Pipe
- Book House - Cross component communication

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **Angular**.

Day 48

Angular (v19)

Key Topics: Angular Forms, Dependency Injection and Services, Routing and Navigation, Router Features

Continuous Learning: Technical Enablement

Angular - The Complete Guide (2024 Edition)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course
    - **Section 31:** Handling Forms in Angular Apps [Angular < 16]
    - **Section 25:** Using Services & Dependency Injection [Angular < 16]
    - **Section 27:** Changing Pages with Routing [Angular < 16]
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Hands-On

Complete the following set of hands-on given in the Learning Path at Tekstac.

Do not copy paste the code. Write the code yourself.

- Rent A Home - Routing with Query param

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **Angular**.

Code Challenge (For Practice Only)

Take on the following Code Challenge in the Tekstac Learning Path to assess your skill level in Angular. You will have 3 attempts, and you must score 70% or higher to pass this challenge.

Do not copy paste the code. Write the code yourself.

- Assess-Type-1: Code Challenge - Angular

Day 49

Angular (v19)

Key Topics: HTTP Client and APIs, State Management in Angular, Reactive Programming with RxJS

Continuous Learning: Technical Enablement

Angular - The Complete Guide (2024 Edition)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 34:** Making Http Requests [Angular < 16]
    - **Section 41:** Bonus: Using NgRx For State Management
    - **Section 29:** Understanding Observables [Angular < 16]
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **Angular**.

Day 50

Angular (v19)

Key Topics: Testing Angular Applications, What's new in Angular 19?

Continuous Learning: Technical Enablement

Angular Testing Masterclass (Angular 19)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **ALL** Sections
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Go through the following link to know about the latest features of Angular 19

- What's New in Angular 19: Key Features, Code Examples, and Improvements

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **Angular**.

Day 51

Angular (v19)

Key Topics: Introduction to Node.js, Core Node.js Modules, Asynchronous Programming Basics

Continuous Learning: Technical Enablement

The Complete Node.js Developer Course (3rd Edition)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 2:** Installing and Exploring Node.js
    - **Section 3:** Node.js Module System (Notes App)
    - **Section 4:** File System and Command Line Args (Notes App)
    - **Section 6:** Asynchronous Node.js (Weather App)
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Stage 4 -> Course 4 -> Application Debugging - Front-end

Course Overview

The **Course 4** of the **Stage 4** training focuses on front-end application debugging in Angular or React. In this module, learners will gain essential skills in identifying and resolving issues in their front-end applications. They will learn debugging techniques specific to Angular or React, including using developer tools, logging, and debugging libraries. The module will cover common debugging scenarios such as troubleshooting errors, optimizing performance, and understanding application behavior.

Learning Objectives

After completing this course, GenCs will be able to:

- Familiarize with Chrome DevTools and VS Code Debugger for web development.
- Configure Source Maps and adjust project settings for effective debugging.

- Inspect state, props, and services using breakpoints.
- Debug Observables, Promises, and Async/Await to resolve async issues.
- Monitor Angular Route Guards and React Route Components for smooth navigation.
- Track network requests and profile performance to optimize applications.
- Use memory tools to detect leaks and improve application stability.
- Debug across multiple browsers to handle browser-specific issues.

Day 52 - Forenoon

Application Debugging - Front-end

Key Topics: Debugging Angular Applications with Chrome DevTools

Continuous Learning: Technical Enablement

Devtools Pro: The Basics of Chrome Developer Tools

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **All** Sections
- Ensure that you learn these topics through self-learning and practice alongside the course instructor. It is NOT necessary to cover every topic comprehensively within each section.

Demo Videos

Watch the following demo videos provided in the platform for getting a complete picture about how to debug an/a Angular/React application using Chrome's DevTools and VS Code Debugger.

Note: The below given Cognizant Kpoint videos can be accessed even after your Tekstac platform access is revoked.

- Debugging Angular Part 1
- Debugging Angular Part 2

Integrated Development Project (IDP)

Day 52 - Afternoon, 53, 54 - Forenoon

Integration

These days will be spent on the following IDP activity with the Trainer guidance.

- Integration of Front-end with Web API

Final Evaluation (Project + Technical)

Day 54 - Afternoon, 55, 56, 57 - Forenoon

Final Evaluation (Project + Technical)

- Final Evaluation will be conducted on these days, and the mode will be a video interview on the Tekstac platform.

Final Assessment

Day 57 - Afternoon, 58 - Forenoon

HackerRank Prep & Assessment

- Learners will have HackerRank assessment on this day.

How to learn each day?

Each day has a set of learning objectives. These learning objectives can be met by going through the Udemy courses and by completing the hands-on exercises mentioned in the daily plan.

The below strategies will help you decide the learning approach.

Learning Strategy & Approach

Find below few imaginary profiles. For each of these profiles we have defined a recommended learning approach. This is not an exhaustive list. The approaches below might help invent a new way of learning.

Profile #1

Harry Reacher

Engineering Discipline: Electronics

Skills: Python, Ruby on Rails, nginx

Project: Mining Crime Data to get Route Cause Insights

Learning Approach to Programming Languages: I do not want to waste my time learning. I am more practice oriented. I want to work on the problem immediately

What will work for me?

- Directly complete hands-on exercises
- Refer Internet or Udemy Courses
- If hands on are implemented early, clarify your friends' questions and troubleshoot their issues

Profile #2

Olivia Richards

Engineering Discipline: Computer Science

Skills: Java, C, C++

Project: Library Management System

Learning Approach to Programming Languages: I have interest, but I don't know where to start.

What will work for me?

- Go through the recommended Udemy Course
- Try completing the hands-on exercises
- Get your clarifications solved with help from Tech SME
- Get help from other learners in your batch whom had already completed

Profile #3

Greg Anderson

Engineering Discipline: Civil

Skills: C

Project: Fiber reinforced concrete

Learning Approach to Programming Languages: I am scared of programming languages. I haven't got my hands dirty with coding

What will work for me?

- Go through the recommended Udemy Course
- Implement the coding along with the author of the Udemy Course
- Try completing the hands-on exercises
- Clarify queries with SME
- Troubleshoot programming issues with help from SME or learner from your classroom whom had already completed

FAQs

1. **Who can participate in this program?**

   Ans: Students who have enrolled for Full Internship Program (or) the Cognizant on-boarded GenCs can participate in this program.

2. **Is there any pre-learning I should do?**

   Ans: No. This program is open to all students from any academic discipline.

3. **What is Code Challenge?**

   Ans: A problem statement will be provided to you and you need to solve it using a single skill.

4. **What is Integrated Capability Test (ICT)?**

   Ans: A case study problem statement will be provided to you that you may need to solve using the combination of skills learned in the given stage.

5. **How many attempts are provided for the Coding challenge and ICTs? Is it open all the time for practice?**

   Ans: The coding challenges and ICTs are open from day 1, and a maximum of 3 attempts will be provided.

6. **What if I fail in the Interim evaluation?**

   Ans: Your coach will notify your performance in the Interim evaluation. However, you can continue with the learning.

7. **How many chances will I get in the Final evaluation?**

Ans: You'll get 2 chances in the Final evaluation which covers ALL the skills in the learning journey.

8. Will we be provided with Projects to work on?

Ans: Yes, we will provide the requirement specifications. You will need to implement them by forming a PoD (Pod of Developers). The implementation will be reviewed during both the interim evaluation and the final evaluation.

9. Whom do I reach out in case of any queries?

Ans: Batch Owner is your point of contact.