GenC Program

SPE DotNet FSE - React (Stage 2 Plus) Handbook

Why do we need this Full Stack Engineering Prep-up 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 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics
- Stage 4 - Full Stack Development: Front-end Development and Version Control
- 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.
- Learner Autonomy is implemented via Flipped Classroom, where the learning platform offers world class learning resources, and students would not be constrained by tutelage of an instructor.
- 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.

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)?

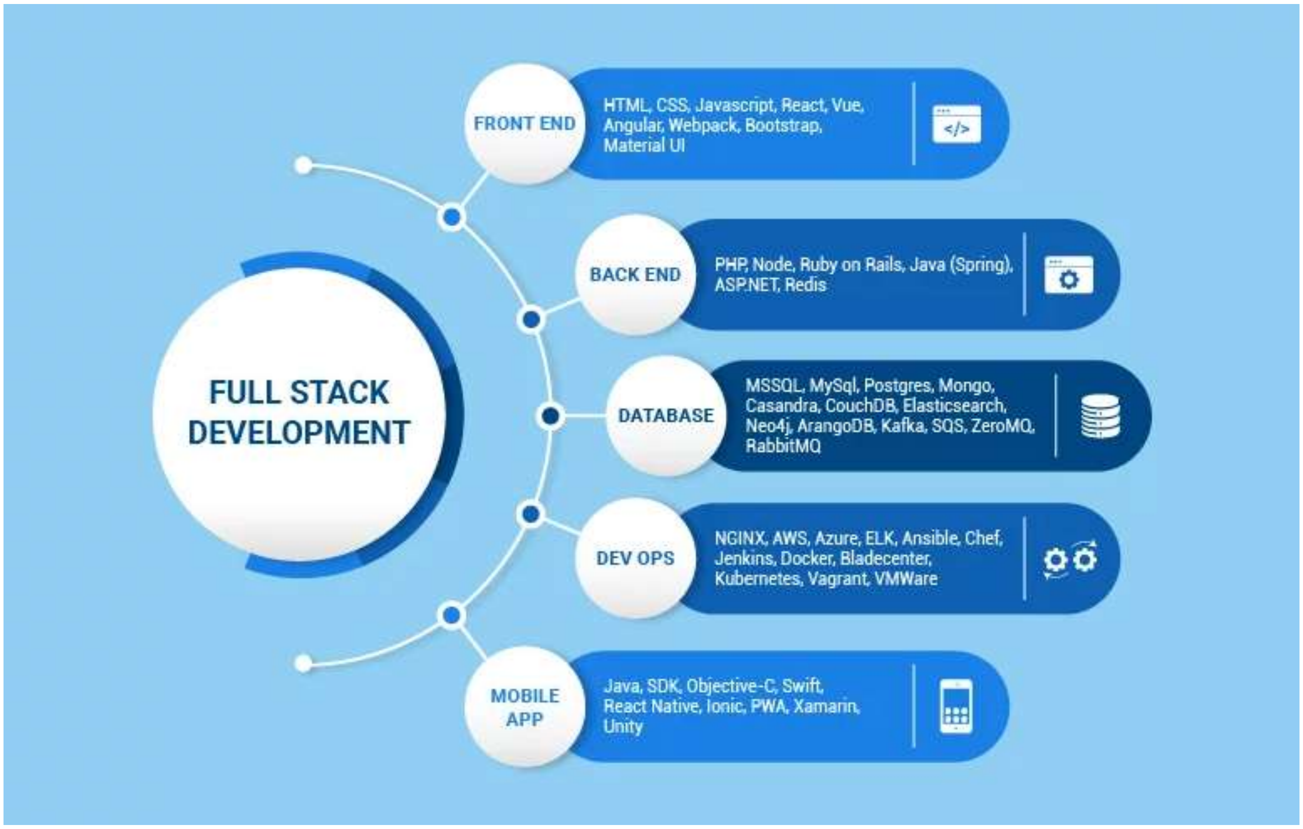
Digital Engineering is the practice in which new applications are conceived and delivered. Encompassing the methodologies, utility, and process of creating new digital products end to end, digital engineering leverages data and technology to produce improvements to applications—or even entirely new solutions.

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.


FULL STACK DEVELOPMENT
FRONT END
HTML, CSS, Javascript, React, Vue, Angular, Webpack, Bootstrap, Material UI
BACK END
PHP, Node, Ruby on Rails, Java (Spring), ASP.NET, Redis
DATABASE
MSSQL, MySql, Postgres, Mongo, Casandra, CouchDB, Elasticsearch, Neo4j, ArangoDB, Kafka, SQS, ZeroMQ, RabbitMQ
DEV OPS
NGINX, AWS, Azure, ELK, Ansible, Chef, Jenkins, Docker, Bladecenter, Kubernetes, Vagrant, VMWare
MOBILE APP
Java, SDK, Objective-C, Swift, React Native, Ionic, PWA, Xamarin, Unity

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.

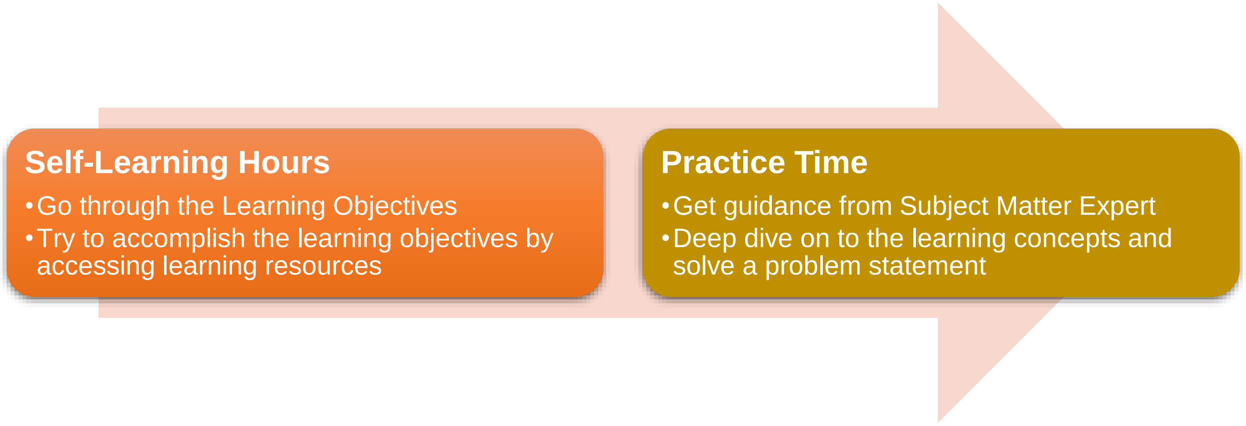
Learning Journey thru Flipped Classroom

The flipped classroom inverts the traditional learning experience. Lectures are shared outside of class time for individual review as homework, and classroom time is reserved for class discussion and interactive projects. The principal goals of flipping are as follows:

- To make the classroom an active learning environment
- To enable students to learn at their own pace, and
- To give the instructor more time to teach each student individually, rather than the class as a whole.

The learning path is set in the GEN C Learn Platform, which you can login with SSO.

Flipped Classroom



Self-Learning Hours

- Go through the Learning Objectives
- Try to accomplish the learning objectives by accessing learning resources

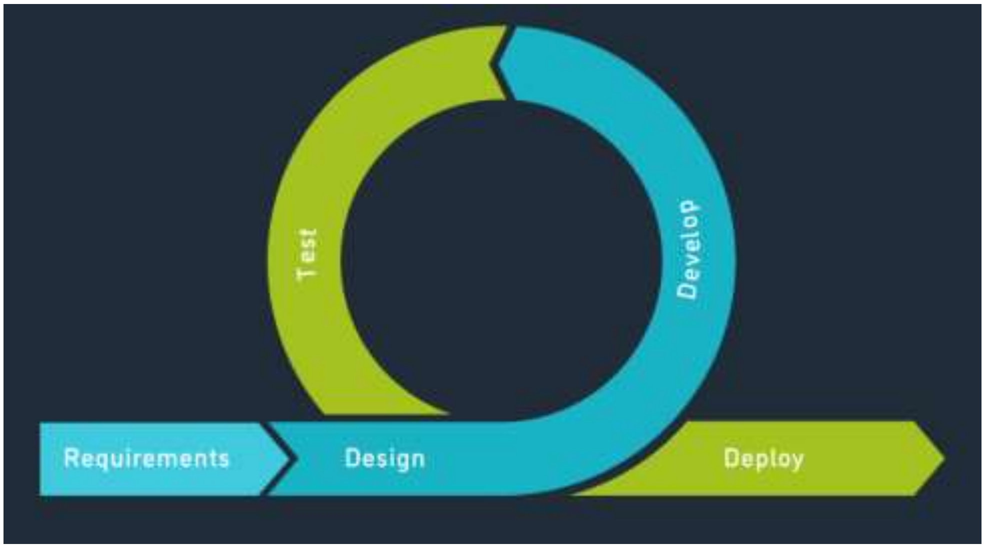
Practice Time

- Get guidance from Subject Matter Expert
- Deep dive on to the learning concepts and solve a problem statement

What is Integrated Development Project (IDP)?

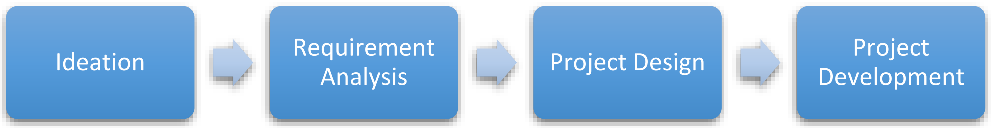
Integrated Development Project is an approach wherein the learner experiences the entire software development processes in an incremental fashion as part of the GenC Training. The IDP implementation is purely based on **Agile Software Development** methodologies and inspired from **PBL (Project-Based 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.



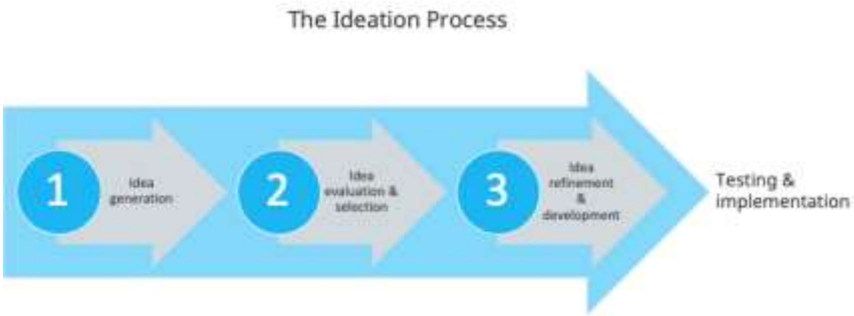
Stages of IDP

Following are the four seminal phases of IPD.



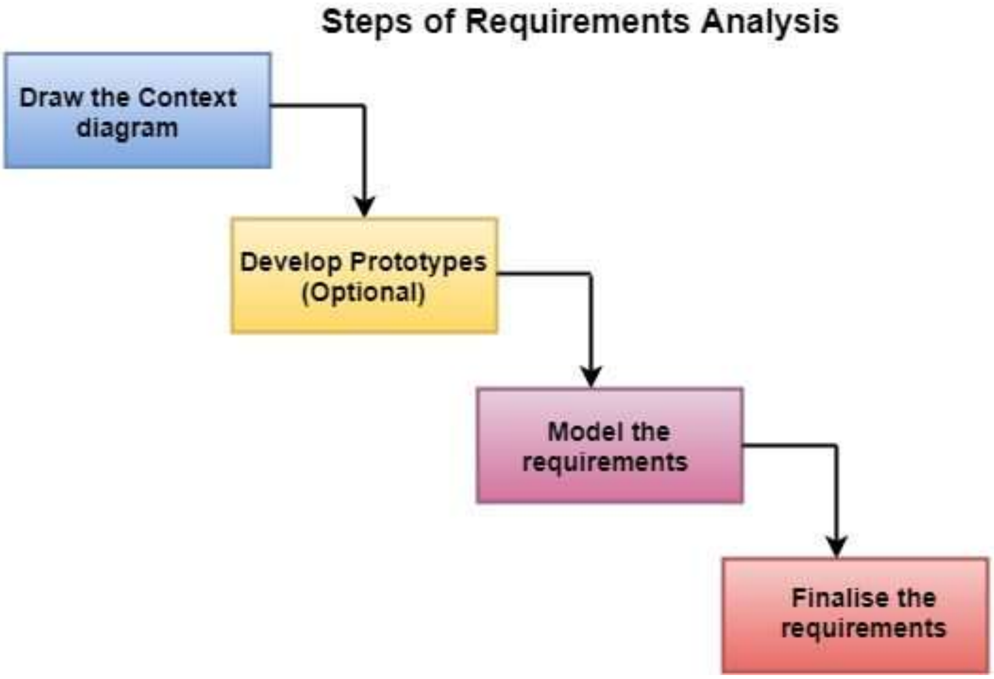
Phase 1: Ideation

Ideation is the creative process of generating, developing, and communicating ideas. It’s important to note that these ideas don’t have to be completely new. You can ideate to solve specific problems, look into new ways of implementing a solution, or even collect feedback and evaluate ideas.



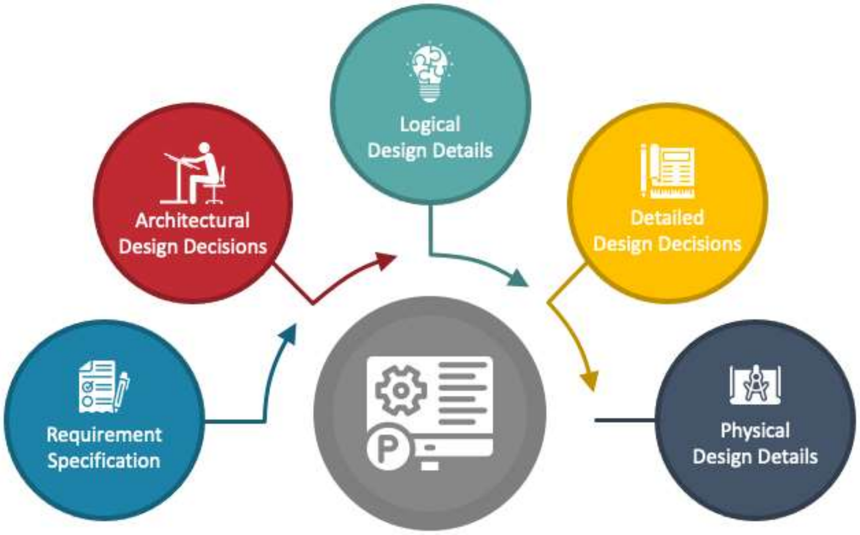
Phase 2: Requirement Analysis

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.


Steps of Requirements Analysis
Draw the Context diagram
Develop Prototypes (Optional)
Model the requirements
Finalise the requirements

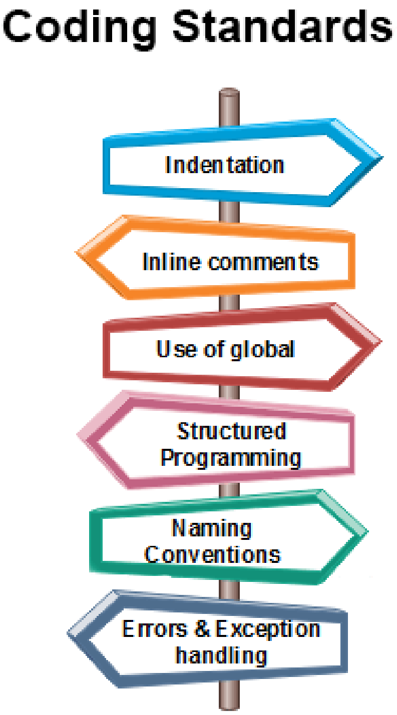
Phase 3: Project Design

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


Logical Design Details
Architectural Design Decisions
Detailed Design Decisions
Requirement Specification
Physical Design Details

Phase 4: Project Development

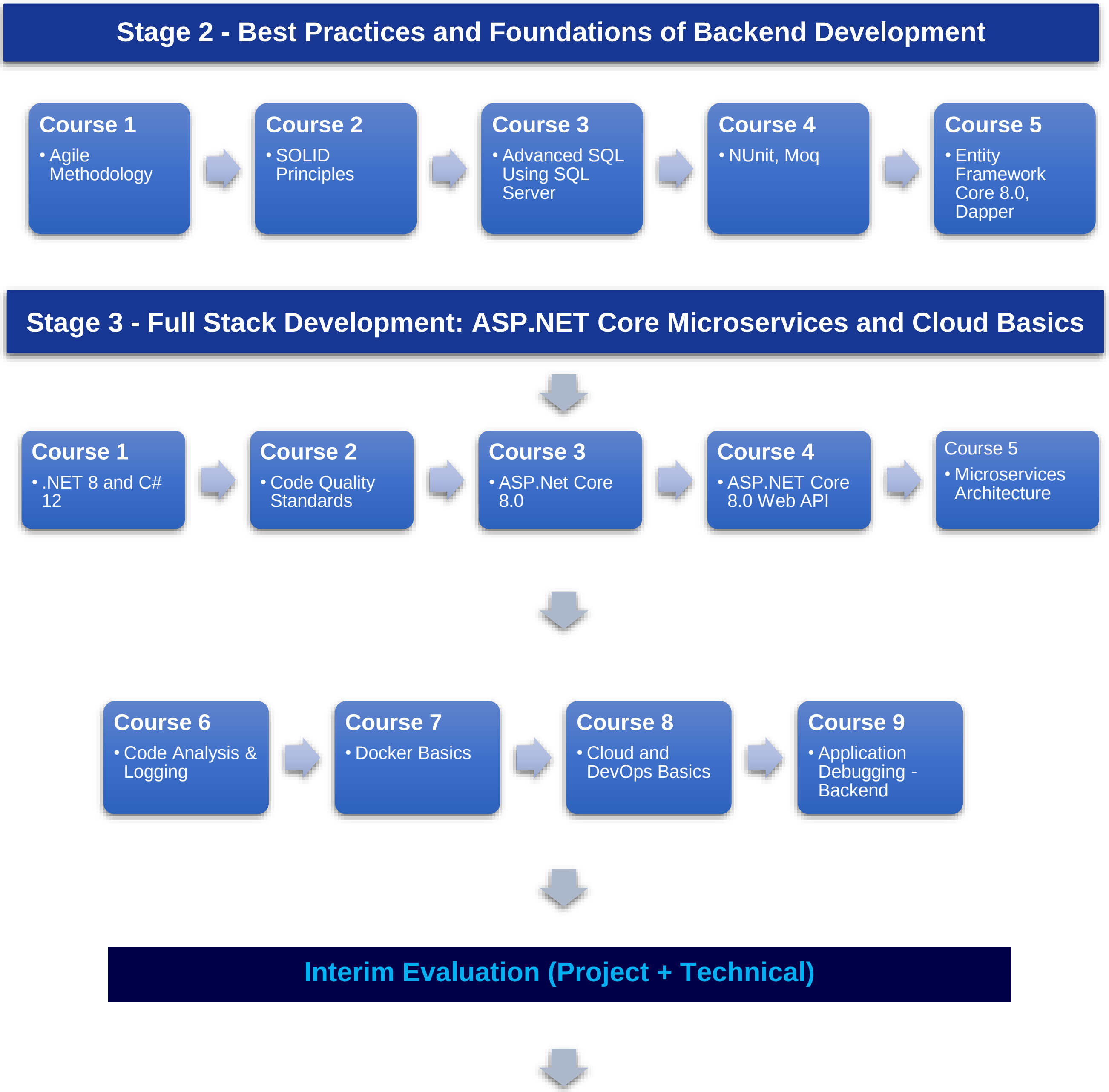
Once the system design phase is over, the next phase is development. In this phase, developers start build the entire system by writing code using the chosen programming language. In this phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of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



Recommended Program Sequence

The learning journey starts with **5 days of Icebreaker sessions** 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 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics
- Stage 4 - Full Stack Development: Front-end Development and Version Control
- Integrated Development Project (IDP)



Stage 2 - Best Practices and Foundations of Backend Development

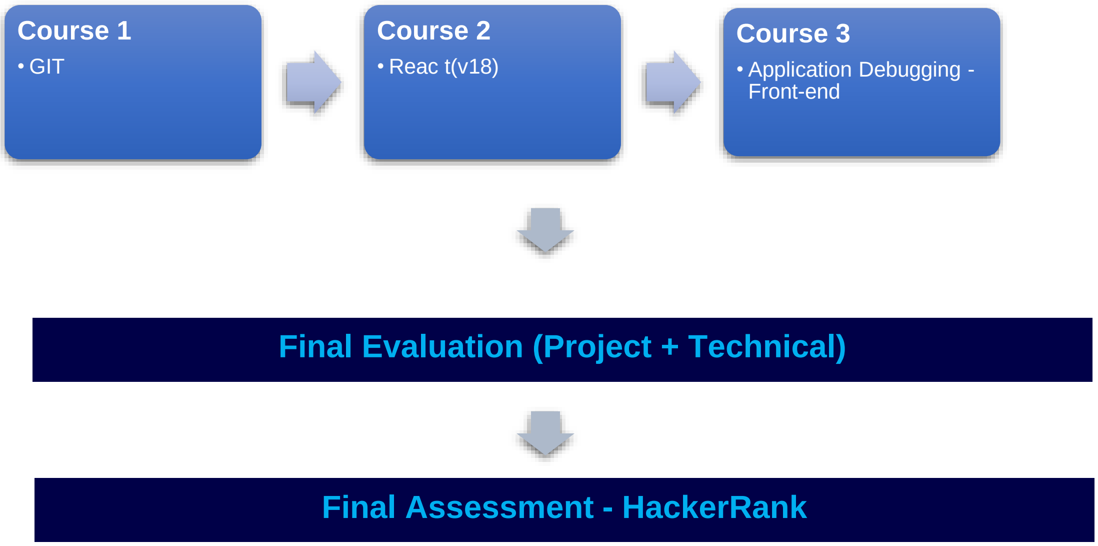
- **Course 1**
  - Agile Methodology
- **Course 2**
  - SOLID Principles
- **Course 3**
  - Advanced SQL Using SQL Server
- **Course 4**
  - NUnit, Moq
- **Course 5**
  - Entity Framework Core 8.0, Dapper

Stage 3 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics

- **Course 1**
  - .NET 8 and C# 12
- **Course 2**
  - Code Quality Standards
- **Course 3**
  - ASP.Net Core 8.0
- **Course 4**
  - ASP.NET Core 8.0 Web API
- Course 5
  - Microservices Architecture
- **Course 6**
  - Code Analysis & Logging
- **Course 7**
  - Docker Basics
- **Course 8**
  - Cloud and DevOps Basics
- **Course 9**
  - Application Debugging - Backend

Interim Evaluation (Project + Technical)

Stage 4 - Full Stack Development: Front-end Development and Version Control



Course 1
- GIT

Course 2
- Reac t(v18)

Course 3
- Application Debugging - Front-end

Final Evaluation (Project + Technical)

Final Assessment - HackerRank

All the 3 stages would be executed in the **Flipped classroom model** through Learning paths configured on the **Tekstac** platform.

There will be an integrated project called **IDP** (Integrated Development Project) which will be executed in an incremental fashion and is part of all the 3 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 are able to 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 GenC's knowledge and understanding of the skills covered in the training program up to the halfway point. It also encompasses an evaluation of the GenC's 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 GenC's 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. Similar to 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 Exercises 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…

Recommended Day-wise Schedule

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. Few days might be interleaved to accommodate the extension due to Behavioral Training.

Stage 2 - Best Practices and Foundations of Backend Development

Overview

Stage 2 deals with best practices and foundations of backend development. 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 2 of your training, the following skills will be covered.

- Agile Methodology
- SOLID Principles
- Advanced SQL Using SQL Server
- NUnit, Moq
- Entity Framework Core 8.0, Dapper

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 reliable sources as they are mentioned in this handbook.

Stage 2 -> Course 1 -> Agile Methodology

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.

Stage 2 -> Course 2 -> SOLID Principles

Course Overview

The **Course 2** of the **Stage 2** provides a comprehensive introduction to the SOLID principles of Object-Oriented Programming (OOP), designed to enhance your software design skills and enable you to create scalable, maintainable, and robust applications. By exploring each principle in detail, you will understand the "what" and "why" behind these foundational guidelines for OOP and learn how to apply them effectively in real-world scenarios. This course is essential for anyone aiming to improve their software development practices and deliver high-quality code that adheres to industry standards.

Learning Objectives

After completing this course, GenCs will be able to:

- Explain the purpose and importance of SOLID principles in Object-Oriented Programming.
- Design classes that focus on a single responsibility to improve modularity and reduce coupling.
- Develop code that is open for extension but closed for modification, ensuring flexibility without sacrificing stability.

- Ensure class hierarchies are designed to allow seamless substitution of derived classes for base classes without altering functionality.
- Create specific and focused interfaces to prevent unnecessary dependencies and enhance class usability.
- Structure code to depend on abstractions rather than concrete implementations, promoting decoupled and flexible architectures.

Day 2

SOLID Principles

Key Topics: Solid Principles

Continuous Learning: Technical Enablement

Design Patterns in C# and .NET

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 0:** undefined
    - **Section 1:** The 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.

Stage 2 -> Course 3 -> Advanced SQL Using SQL Server

Course Overview

The **Course 3** of the **Stage 2** focuses on **Advanced SQL Using SQL Server**, covering various key concepts. Participants will start by understanding the basics and gradually delve into advanced concepts. They will learn how to work with views and indexes for efficient data retrieval and management. The module also covers stored procedures and user-defined functions, which are essential for complex data processing tasks. Additionally, participants will explore triggers and cursors for more granular control over database operations. Exception handling and transaction management, crucial for ensuring data integrity and reliability, will also be covered in this milestone.

Learning Objectives

After completing this course, GenCs will be able to:

- Learn how to connect to SQL Server and execute basic SQL queries.
- Learn about SQL Server-specific features like window functions and common table expressions.
- Understand the concept of views and how to create and manage them in SQL Server.
- Learn about indexes and their importance in optimizing query performance.
- Learn how to create and use stored procedures in SQL Server for reusable code execution.
- Understand the concept of user-defined functions and how to create and use them in SQL Server.
- Explore the use of triggers for automatically enforcing constraints or executing actions in response to data modifications.

- Understand the concept of cursors and how to use them to process individual rows returned by a query.
- Learn how to handle exceptions and errors in SQL Server using try-catch blocks. Understand the importance of error handling in maintaining data integrity and application reliability.
- Understand the concept of transactions and their properties (ACID).
- Learn how to manage transactions in SQL Server to ensure data consistency and recoverability.

Day 3

Advanced SQL Using SQL Server

Key Topics: Getting Started, Advanced Concepts

Continuous Learning: Technical Enablement

70-461, 761: Querying Microsoft SQL Server with Transact-SQL

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 25: Session 5 - Objective 9:** Implement aggregate queries
    - **Section 26: Session 5 - Objective 9d:** Ranking functions
    - **Section 28: Session 5 - Objective 9b:** Grouping sets
    - **Section 33: Session 6 - Objective 7e:** WITH statement
    - **Section 34: Session 6 - Objective 7b:** PIVOTing and UNPIVOTing
    - **Section 35: Session 6 - Objective 7d:** CTE statement
- 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 solution. Write the query yourself.

- Flight Booking Details – DDL
- Alter Flight Details – DDL
- Flight Count By Source - DRL
- Bill Count Based On Status - Pivot
- Weekend Flights - DRL
- Time Change - DRL
- Car Sales - Rank
- Flight Bookings - CTE
- Doctor and Updated Doctor - Merge

Day 4

Advanced SQL Using SQL Server

Key Topics: Views and Indexes, Stored Procedures and User-Defined Functions

Continuous Learning: Technical Enablement

70-461, 761: Querying Microsoft SQL Server with Transact-SQL

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 16: Session 3 - Objectives 2 and 3:** Views
  - **Section 47: Session 7** - Indexes
  - **Section 21: Session 4 - Objective 11 -** Create and alter stored procedure
  - **Section 36: Session 6 - Objective 14:** Functions
- 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 solution. Write the query yourself.

- Create View Flight Booking
- Passenger Details - Drop View
- Update Flight Name - Function
- Booking Information - Function
- Booking Amount - Procedure
- Passenger Info - Procedure
- Booking – Procedure

Day 5 - Forenoon

Advanced SQL Using SQL Server

Key Topics: Triggers and Cursors, Exception Handling, Transactions

Continuous Learning: Technical Enablement

70-461, 761: Querying Microsoft SQL Server with Transact-SQL

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 22: Session 4 - Objective 18a -** Implement try/catch/throw
  - **Section 17: Session 3 - Objective 5:** Create and alter DML triggers

  - **Section 46: Session 7 -** Manage transactions
- 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 links in order to understand more about Cursors.

- Cursors

Hands-On

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

❗ Do not copy paste the solution. Write the query yourself.

- March Booking - Cursor
- Insert - Trigger
- Update Flight Date – Trigger
- Passenger Info - Exception Handling
- Delete Flight Details- Exception Handling
- Passenger Info - Transaction Control

Technical Quiz

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level on SQL Server.

- SQLServer

Code Challenge (For Practice Only)

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on Advanced SQL Programming using SQL Server. You have to secure 70% in order to clear this challenge.

❗ Do not copy paste the solution. Write the query yourself.

- Assess-Type-1: Code Challenge - SQL Server

Stage 2 -> Course 4 -> NUnit, Moq

Course Overview

The **Course 4** of the **Stage 2** focuses on unit testing frameworks like **NUnit** and mocking frameworks like **Moq**. Participants will start by understanding the importance of unit testing in software development and how it contributes to building robust and maintainable code. They will then delve into NUnit, a popular unit testing framework for .NET applications, learning how to write and execute unit tests for their code. Additionally, participants will explore Moq, a mocking framework that helps in creating mock objects for testing. They will learn how to use Moq to isolate components for unit testing and simulate interactions between objects. Overall, this milestone aims to equip participants with the skills and knowledge needed to effectively test their code using NUnit and Moq.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand the concept and benefits of automated testing.
- Learn how to set up and use NUnit in Visual Studio for writing and executing unit tests.
- Understand the principles of TDD and how to apply them using NUnit.
- Learn about the characteristics of good unit tests, including readability and maintainability.
- Learn how to set up and tear down test fixtures using NUnit.
- Understand the concepts of dependency injection and mocking using frameworks like Moq.
- Learn best practices for writing trustworthy tests, organizing tests, and achieving code coverage.

Day 5 - Afternoon, 6, 7 - Forenoon

NUnit, Moq

Key Topics: Getting Started, Fundamentals of Unit Testing, Core Unit Testing Techniques, Breaking External Dependencies

Continuous Learning: Technical Enablement

Unit Testing for C# Developers

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 1:** Getting Started
  - **Section 2:** Fundamentals of Unit Testing
  - **Section 3:** Core Unit Testing Techniques
  - **Section 5:** Breaking External Dependencies
- 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.

- Test Case - Validations On String Concatenation
- Test Case - Validations On Vowel Checker
- Test Case - Validations On Array Search
- Test Case - Validations on Array Multiplication
- Test Case - Validations on List

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 5 -> Entity Framework Core 8.0, Dapper

Course Overview

The **Course 5** of the **Stage 2** provides an in-depth understanding of two key .NET data access frameworks: Entity Framework Core (EF Core) and Dapper. Participants will learn to implement data access operations efficiently using modern ORM techniques (EF Core) and lightweight object mapping (Dapper).

Learning Objectives

After completing this course, GenCs will be able to:

- Gain a clear understanding of when and why to use EF Core and Dapper for data access.
- Install EF Core, create projects, and configure DbContext and entities effectively.
- Define and manage one-to-one, one-to-many, and many-to-many relationships.
- Implement Create, Read, Update, and Delete operations using both EF Core and Dapper.
- Use LINQ with EF Core and raw SQL queries to fetch and manipulate data efficiently.
- Manage transactions and work with stored procedures for complex operations.
- Identify key differences between EF Core and Dapper and choose the right tool based on use cases.
- Prevent SQL injection attacks by using parameterized queries in both frameworks.
- Use Dapper Plus for bulk insert, update, and delete operations.
- Utilize logging and debugging tools to troubleshoot issues in EF Core and Dapper applications.

Day 8

Entity Framework Core 8.0, Dapper

Key Topics: Overview of EF Core 8 and .NET 8 Integration, Setting up EF Core in a .NET 8 Project

Continuous Learning: Technical Enablement

Entity Framework Core - A Full Tour

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 1:** Introduction
    - **Section 2:** Environment Setup

    - **Section 3:** Getting Started with Entity Framework Core
- 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.

- Create a Model with Database Table in .NET 8 using EF Core

Day 9

Entity Framework Core 8.0, Dapper

Key Topics: Creating a Simple Database Model, Performing Basic CRUD Operations, LINQ Queries in EF Core 8

Continuous Learning: Technical Enablement

Entity Framework Core - A Full Tour

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 4:** Using Entity Framework Core to Query a Database
    - **Section 5:** Using Entity Framework Core to Manipulate Data
- 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.

- A Comprehensive Guide to Entity Framework Core in .NET 8
- CRUD Operations in Entity Framework Core

Hands-On

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

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

- Car Repository - Insert
- Car Repository - Eager Loading
- Car Repository - Lazy Loading

Day 10

Entity Framework Core 8.0, Dapper

Key Topics: EF Core Migrations and Database Updates, Handling Relationships and Data Loading, Performance Optimizations and Best Practices

Continuous Learning: Technical Enablement

Entity Framework Core - A Full Tour

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 6:** Handling Database Changes and Migrations
    - **Section 7:** Interacting With Related Records
- 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.

- EF Core Performance Optimisations

Technical Quiz

Attempt the following technical quiz in the Learning Path at Tekstac for checking your knowledge level on Entity Framework Core.

- Entity Framework

Code Challenge (For Practice Only)

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on Entity Framework Core. You have to secure 70% in order to clear this challenge.

> ❗ Do not copy paste the solution. Write the query yourself.

- Assess-Type-1: Code Challenge - Entity Framework

Day 11

Entity Framework Core 8.0, Dapper

Key Topics: Getting Started, Database Providers with Dapper, Querying Data with Dapper, Executing Non-Query Commands with Dapper

Continuous Learning: Technical Enablement

Dapper - Getting Started

- 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.

Learn about database providers from the below.

- Database Providers with Dapper

Day 12 - Forenoon

Entity Framework Core 8.0, Dapper

Key Topics: Using Parameters with Dapper, Using Relationships with Dapper, Bulk Operations with Dapper

Continuous Learning: Technical Enablement

- Using Parameters with Dapper
- Using Relationship with Dapper
- Bulk Operations with Dapper

ICT (Integrated Capability Test) (For Practice Only)

Take up the following extended integrated practice task in order to check your skill level after completing the Stage 1 of your training. Unlike Code Challenge, the coverage of this practice will be Stage 2 Skills. You have to score a minimum 70% in order to complete this activity successfully.

- Dotnet Assess-Type-2: Integrated Capability Test (ICT)

IDP - Demo & Re-work

Day 12 - Afternoon, Day 13 - Forenoon

IDP - Internal Demo & Re-work

- This day will be used for project demo & re-work

At this point, the GenC will demonstrate the project work completed and the BU Mentor/Evaluator will sign off the work on agreeable conditions.

Stage 3 - Full Stack Development: ASP.NET Core Microservices and Cloud Basics

Overview

Stage 3 training is designed to equip participants with advanced skills in modern .NET development and related technologies. The training covers topics such as the latest features in .NET 8 and C# 12, building web applications and APIs with ASP.NET Core 8, implementing microservices architecture, using Docker for containerization, understanding cloud computing and DevOps practices, and mastering application debugging for backend development. Through a combination of theoretical concepts and hands-on exercises, participants will gain a deep understanding of these technologies, enabling them to build scalable, efficient, and maintainable applications in a modern development environment.

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

- .NET 8 and C# 12
- Code Quality Standards
- ASP.Net Core 8.0
- ASP.NET Core 8.0 Web API
- Microservices Architecture
- Code Analysis & Logging
- Docker Basics
- Cloud and DevOps Basics
- Application Debugging - Backend

How and From Where to Learn?

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

Offline Hands On (Additional Practice)

We strongly recommend that you try out the offline hands-on exercises given in the Learning Path, as they will help you to gain a better understanding of the concepts covered in the course and provide you with a more practical and comprehensive learning experience.

Course Overview

The **Course 1** of the **Stage 3** provides a comprehensive introduction to the .NET platform and its latest advancements, equipping learners with the knowledge and skills to build modern, robust applications. Starting with an overview of the .NET platform and its evolution from the .NET Framework to the current .NET 6 and .NET 8, participants will explore the key features and improvements in .NET 8. The course also delves into C# 12, highlighting its new language features, syntax enhancements, and their practical benefits when developing .NET 8 applications. By the end of this course, learners will have a strong foundation in leveraging the latest capabilities of .NET and C# to create high-performance and maintainable solutions.

Learning Objectives

After completing this course, GenCs will be able to:

- Explain the architecture and components of the .NET platform.
- Trace the evolution of .NET from the .NET Framework to .NET 6 and .NET 8.
- Identify the key features and improvements introduced in .NET 8.
- Evaluate the benefits of the enhancements for modern application development.
- Describe the new language features and syntax improvements in C# 12.
- Demonstrate how to utilize these enhancements effectively in .NET 8 applications.
- Develop applications using the improved capabilities of .NET 8 and C# 12.
- Optimize solutions by leveraging modern features for better performance and maintainability.

Day 13 - Afternoon

.NET 8 and C# 12

Key Topics: Introduction to .NET 8, C# 12 Features

Continuous Learning: Technical Enablement

.NET 8 Microservices: DDD, CQRS, Vertical/Clean Architecture

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 3:** Why .NET and What is new .NET 8 and C#12?
- 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.

- What's new in .NET 8
- What's new in C# 12

Course Overview

The **Course 2** of the **Stage 3** is designed to equip learners with essential coding best practices that ensure clarity, efficiency, and maintainability in software development. By mastering key topics such as naming conventions, indentation, and memory management, participants will develop the skills necessary to write clean, optimized, and professional code. The course also emphasizes collaborative practices like peer reviews and introduces tools for automated code analysis to foster quality assurance in development projects.

Learning Objectives

After completing this course, GenCs will be able to:

- Utilize consistent and meaningful names for variables, functions, classes, and files to enhance code clarity and readability.
- Structure code with appropriate indentation and formatting techniques to improve its visual structure and maintainability.
- Create clear and concise comments and documentation to support code understanding and future development.
- Implement practices that make code intuitive and accessible for team members and future developers.
- Write efficient algorithms and manage resources effectively to optimize application performance.
- Apply techniques to manage memory efficiently, ensuring the stability and scalability of applications.
- Engage in collaborative code reviews to identify and rectify potential issues, ensuring high-quality outputs.
- Use industry-standard tools to analyze and validate code quality, performance, and compliance with coding standards.

Day 14 - Afternoon

Code Quality Standards

Key Topics: Coding Standards, Performance, Code Reviews

Continuous Learning: Technical Enablement

- C# Coding Standards
- Performance
- Code Reviews

Offline Hands-On (Additional Practice)

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

Day 15 - Forenoon

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

Stage 3 -> Course 3 -> ASP.Net Core 8.0

Course Overview

The **Course 3** of the **Stage 3** provides a comprehensive introduction to **ASP.NET Core 8**, a modern, open-source, cross-platform framework for building web applications. Designed for aspiring developers, the course emphasizes practical skills to design, develop, and deploy dynamic web applications using the latest features of ASP.NET Core. Participants will gain a solid understanding of the MVC (Model-View-Controller) architecture, explore robust techniques for handling user input and state management, and learn to integrate Entity Framework (EF) Core 8 for seamless data operations. By the end of the course, learners will be equipped to create scalable, maintainable, and deployable web applications using industry best practices.

Learning Objectives

After completing this course, GenCs will be able to:

- Describe the key features and benefits of ASP.NET Core 8.
- Set up and configure a new ASP.NET Core project.
- Explain the principles of the Model-View-Controller (MVC) design pattern.
- Develop and integrate Models, Controllers, and Views effectively.
- Create and manage forms to collect and process user input.
- Utilize routing to enable intuitive navigation and URL structures.
- Explore techniques for session and application state management.
- Implement secure state management solutions.
- Configure and connect ASP.NET Core applications with databases using EF Core 8.
- Perform CRUD (Create, Read, Update, Delete) operations with EF Core.
- Implement Dependency Injection (DI) for modular and testable code.
- Understand the process of publishing and deploying ASP.NET Core applications to various hosting environments.
- Apply the skills and concepts learned to design and develop a fully functional web application, ready for deployment.

Day 15 - Afternoon, 16

ASP.NET Core 8

Key Topics: Introduction to ASP.NET Core 8, Creating a New ASP.NET Core Project, Understanding MVC Architecture

Continuous Learning: Technical Enablement

The complete ASP.NET Core 9 course for busy developers

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 1:** Building your first ASP.NET Core 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

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

- Interrogation Panel - Data Transferring
- Interrogation Panel - Component
- Interrogation Panel - Razor Page

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **ASP.NET Core**.

Day 17

ASP.NET Core 8

Key Topics: Working with Models, Building Controllers, Designing Views in ASP.NET Core

Continuous Learning: Technical Enablement

- Get started with ASP.NET Core MVC
- Add a controller to an ASP.NET Core MVC app
- Add a view to an ASP.NET Core MVC app
- Add a model to an ASP.NET Core MVC app

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **ASP.NET Core**.

Day 18

ASP.NET Core 8

Key Topics: Routing in ASP.NET Core, Working with Forms and User Input, State Management, Integrating EF Core 8 with ASP.NET Core

Continuous Learning: Technical Enablement

- Routing in ASP.NET Core
- ASP.NET Core Blazor forms overview
- Session and state management in ASP.NET Core

Continuous Learning: Technical Enablement

Asp.Net Core 9 (.NET 9) | True Ultimate Guide

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 5:** Routing [MVC and Web API]
    - **Section 17:** Tag Helpers [MVC]
    - **Section 18:** Entity Framework Core [MVC and Web API]
- 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.

- Interrogation Panel - DB Initialize

Day 19 - Forenoon

ASP.NET Core 8

Key Topics: Dependency Injection (DI) in ASP.NET Core, Publishing and Deployment

Continuous Learning: Technical Enablement

- Dependency injection in ASP.NET Core
- Host and deploy ASP.NET Core

Code Challenge (For Practice Only)

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on ASP.NET Core. You have to secure 70% in order to clear this challenge.

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

- Assess-Type-1 Code Challenge - ASP.Net Core, MVC with Entity Framework

Course Overview

The **Course 4** of the **Stage 3** provides an in-depth understanding of Web APIs and ASP.NET Core, enabling learners to design, develop, and secure robust API solutions. Starting with the foundational concepts of Web APIs and ASP.NET Core, the course delves into building RESTful APIs, exploring advanced API features, and integrating SOAP services. Additionally, learners will gain expertise in API security, exception handling, and best practices for API documentation and testing. By the end of the course, participants will be equipped to create scalable, secure, and well-documented APIs tailored to real-world applications.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand the fundamental principles of Web APIs and the ASP.NET Core framework.
- Design and implement RESTful APIs using ASP.NET Core, adhering to industry standards.
- Explore advanced API features, including pagination, versioning, and filtering.
- Integrate and consume SOAP services in .NET applications.
- Ensure API security through authentication, authorization, and secure communication practices.
- Effectively handle exceptions and implement error-handling mechanisms in APIs.
- Create comprehensive API documentation using tools like Swagger/OpenAPI.
- Conduct rigorous testing of APIs to ensure reliability and performance.

Day 19 - Afternoon, 20

ASP.NET Core 8 Web API

Key Topics: Introduction to Web APIs and ASP.NET Core, Building RESTful APIs with ASP.NET Core

Continuous Learning: Technical Enablement

ASP.NET CORE WEB API | The Complete Guide

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 1:** Before You Get Started
  - **Section 2:** Introduction to Web API
  - **Section 3:** Building Your First ASP.NET Core API
  - **Section 5:** Controller Action Return Types
- 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.

- Implement Entity Framework A Code First Approach in .Net 8 API

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.

- Online Bookstore API - Insert

Offline Hands-On (Additional Practice)

- Try out the offline hands-on exercises given in the learning path on **ASP.NET Core Web API**.

Day 21

ASP.NET Core 8 Web API

Key Topics: Advanced API Features

Continuous Learning: Technical Enablement

Build ASP.NET Core Web API - Scratch To Finish (.NET8 API)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 8:** Securing our ASP.NET Core API - Authentication and Authorization - JWT Tokens
- 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.

- Routing in ASP.NET Core Web API
- Filters in ASP.NET Core Web API

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.

- Online Bookstore API – Retrieve
- Online Bookstore API - Modify
- Online Bookstore API - Remove

Day 22

ASP.NET Core 8 Web API

Key Topics: Consuming and Creating SOAP Services

Continuous Learning: Technical Enablement

- SOAP Web Service in .NET Core

Day 23

ASP.NET Core 8 Web API

Key Topics: API Security and Exception Handling

Continuous Learning: Technical Enablement

Build ASP.NET Core Web API - Scratch To Finish (.NET8 API)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 10:** Advanced Functionality in ASP.NET Core Web 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.

Day 24, 25 - Forenoon

ASP.NET Core 6 Web API

Key Topics: Securing Your API, API Versioning, Documenting Your API with Swagger/OpenAPI

Continuous Learning: Technical Enablement

Build ASP.NET Core Web API - Scratch To Finish (.NET8 API)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 8:** Securing our ASP.NET Core API - Authentication and Authorization - JWT Tokens
  - **Section 11:** Versioning in ASP.NET Core Web API
- 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.

Asp.Net Core 8 (.NET 8) | True Ultimate Guide

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
    - **Section 27:** Swagger / Open API
- 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.

Code Challenge (For Practice Only)

Attempt the following Code Challenge through the Learning Path at Tekstac for checking your skill level on ASP.NET Core Web API. You have to secure 70% in order to clear this challenge.

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

- Assess-Type-1: Code Challenge - Web API.Net

Stage 3 -> Course 3 -> Microservices Architecture

Course Overview

The **Course 3** of the **Stage 3** introduces the principles and practices of microservices architecture with a focus on ASP.NET Core. It covers the key advantages, challenges, and differences between monolithic and microservices architectures. The learners will gain hands-on experience setting up ASP.NET Core Web API projects, implementing service communication methods, ensuring secure authentication, and managing data consistency across services. Additionally, the module delves into advanced topics like monitoring, health checks, logging, and automated deployment using modern DevOps practices such as Docker, Kubernetes, and CI/CD pipelines.

Learning Objectives

After completing this course, GenCs will be able to:

- Learn the core concepts, benefits, and challenges of microservices and compare them with monolithic architectures.
- Create and configure microservices-based ASP.NET Core Web API projects from scratch.
- Use communication methods such as HTTP, messaging, and gRPC for efficient service interactions.
- Understand how service discovery works and register services effectively in a microservices environment.
- Implement data access using Entity Framework Core and manage data consistency and transactions across services.
- Implement JWT-based authentication and secure API endpoints within microservices.
- Set up logging strategies and health checks to monitor service health and performance effectively.
- Understand blue-green and rolling update deployments to reduce downtime during microservices deployment.
- Set up CI/CD pipelines using Azure DevOps or GitHub Actions for seamless integration and delivery.
- Automate microservices deployment with Docker containers and Kubernetes for scalable orchestration.

Day 25 - Afternoon

Microservices Architecture

Key Topics: Introduction to Microservices Architecture

Continuous Learning: Technical Enablement

Creating .Net Core Microservices using Clean Architecture

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 1:** Introduction
  - **Section 2:** Developing Catalog Microservice using Clean Architecture
- 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.

Additional Learning

Walkthrough the following resources in order to understand the basics of DDD (Domain-Driven Design) and 12-Factor App methodology.

- 12-factor-app
- Domain Driven Design

Day 26

Microservices Architecture

Key Topics: Microservice Communication and Data Management

Continuous Learning: Technical Enablement

Creating .Net Core Microservices using Clean Architecture

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 7:** Async Communication between Rabbit MQ, Basket and Ordering Microservice
- 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.

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

- Creating a simple data-driven CRUD microservice

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.

- Health Actuator

Day 27, 28 - Forenoon

Microservices Architecture

Key Topics: Security, Monitoring, and Deployment

Continuous Learning: Technical Enablement

Creating .Net Core Microservices using Clean Architecture

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 9:** Identity Server Introduction
  - **Section 10:** Securing Catalog and Basket Microservice
  - **Section 11:** Securing Service to Service Communication
  - **Section 26:** Deploying Microservices to Kubernetes and AKS
- 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.

- Info Actuator

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

- Make secure .NET Microservices and Web Applications

Course Overview

The **Course 4** of the **Stage 3** provides an in-depth understanding of code analysis and logging in modern .NET applications, equipping learners with essential skills to enhance code quality, maintainability, and traceability. The course begins with a comprehensive introduction to code analysis, covering enabling, configuring, and interpreting code metrics and diagnostics using code analyzers. Participants will gain insights into common code analysis rules and their application in code reviews to ensure compliance with best practices.

The second part of the course focuses on logging in .NET 8, offering a step-by-step exploration of logging configurations, levels, and categories. Learners will also be introduced to log4net, a popular logging framework, and its integration into .NET applications for effective monitoring and debugging. By the end of the course, participants will be equipped to implement robust code analysis and logging mechanisms, contributing to high-quality, maintainable software.

Learning Objectives

After completing this course, GenCs will be able to:

- Explain the importance of code analysis in software development.
- Identify key metrics and diagnostics for evaluating code quality.
- Enable and configure code analysis features in .NET projects.
- Interpret and address diagnostics provided by code analyzers.
- Apply common code analysis rules to identify and resolve issues.
- Utilize code analysis outputs to enhance the code review process.
- Advocate for adherence to coding standards and best practices.
- Set up and configure logging in .NET 8 applications.
- Differentiate between logging levels and categories and apply them effectively.
- Implement logging in controllers and services for traceability.
- Filter and customize log configurations to meet project requirements.
- Understand the basics of log4net and its integration with .NET applications.
- Configure log4net for basic logging scenarios.
- Leverage log4net for advanced logging and monitoring needs.

Day 28 - Afternoon, 29

Code Analysis & Logging

Key Topics: Overview of Code Analysis, Enabling and Configuring Code Analysis, Code Metrics, Code Analyzers and Diagnostics, Common Code Analysis Rules, Code Analysis in Code Reviews, Introduction to Logging in .NET 8, Configuring Logging in .NET 8, Logging Levels and Categories, Logging in Controllers and Services, Log Filtering and Configuration, Introduction to log4net, Basic Logging with log4net

Walk through the following articles and try out the examples given in them.

Code Analysis

- Overview of .NET source code analysis
- Configure live code analysis for .NET
- Code metrics - Cyclomatic complexity
- Code quality rules
- About .NET source code analysis in Visual Studio with Roslyn analyzers

Logging

- Logging in .NET Core and ASP.NET Core
- How to Implement Log4Net in ASP.NET Core Application

Stage 3 -> Course 5 -> Docker Basics

Course Overview

The **Course 5** of the **Stage 3** module 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 the fundamental differences and use cases for Docker containers and virtual machines.
- Analyze the advantages and limitations of Docker compared to traditional virtual machine technologies.
- Distinguish between Docker Community Edition (CE) and Docker Enterprise Edition (EE), including their features and appropriate usage scenarios.
- Demonstrate proficiency in essential Docker commands such as run, ps, stop, rm, and pull for managing containers and images.
- Effectively use the docker run command to create and configure containers with various options, including naming, background execution, and port publishing.
- Understand the structure of Docker images, including layers, parent images, and base images, and how they function in the container lifecycle.
- Describe the purpose and benefits of Docker Compose, and create and manage multi-container applications using Compose files.

- Explain the architecture of Docker, including the Docker Engine, CLI, REST API, and Docker daemon.
- Identify and use Docker volumes and storage drivers for persistent data management within containers.
- Define container orchestration and evaluate the role of Kubernetes in managing containerized applications at scale.

Day 30

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 31

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.

Day 32

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

Stage 3 -> Course 6 -> Cloud and DevOps Basics

Course Overview

The **Course 6** of the **Stage 3** covers Cloud and DevOps using Azure involve leveraging Microsoft Azure cloud services and DevOps practices to build, deploy, and manage applications efficiently. Azure offers a wide range of cloud services, including computing, storage, databases, networking, and more, enabling organizations to scale their applications as needed.

DevOps practices focus on collaboration between development and operations teams, emphasizing automation, continuous integration/continuous deployment (CI/CD), and monitoring. With Azure DevOps services, teams can automate build and release pipelines, manage code repositories, and track work items, facilitating faster delivery of high-quality software.

Learning Objectives

After completing this course, GenCs will be able to:

- Define cloud computing, its characteristics, and the differences between CapEx and OpEx.
- Compare and contrast IaaS, PaaS, and SaaS, and identify their use cases.
- Describe the various cloud deployment models, including public, private, and hybrid clouds.
- Demonstrate proficiency in using the Azure portal, including account creation and resource management.
- Create, configure, and manage Azure resources effectively using both the Azure portal and command-line tools.
- Design and implement secure networking solutions within Azure, including virtual networks and load balancers.
- Create and manage different types of Azure storage and database solutions, including SQL and NoSQL databases.
- Understand and implement Azure's messaging and event solutions, such as Service Bus and Event Hubs.
- Apply Azure security features, including identity management, RBAC, and MFA.
- Utilize Azure DevOps for CI/CD pipelines, version control with Git, and testing strategies to enhance software development workflows.

Day 33

Cloud and DevOps Basics

Key Topics: Introduction to the Cloud, Introduction to Azure, Azure Basic Concepts, Azure Compute

Continuous Learning: Technical Enablement

Microsoft Azure: From Zero to Hero - The Complete Guide

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 2:** Introduction to the Cloud
  - **Section 3:** Introduction to Azure
  - **Section 4:** First Look at Azure
  - **Section 5:** Azure Basic Concepts
  - **Section 7:** Azure Compute
- 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 34

Cloud and DevOps Basics

Key Topics: Azure Networking, Data in Azure, Messaging in Azure, Azure Active Directory

Continuous Learning: Technical Enablement

Microsoft Azure: From Zero to Hero - The Complete Guide

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 8:** Azure Networking
  - **Section 9:** Data in Azure
  - **Section 10:** Messaging in Azure
  - **Section 11:** Identity Management with Azure AD (Entra)
- 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 35

Cloud and DevOps Basics

Key Topics: Introduction to Azure DevOps, Azure Repos, Azure Boards, Azure Pipelines Basics, Advanced Azure Pipelines, Azure Release Pipelines, Azure Test Plans

Continuous Learning: Technical Enablement

Azure Devops Crash Course -Build CI/CD release pipelines

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 1:** Introduction to Azure Devops
  - **Section 2:** Deployment Basics/What are CI/CD Pipelines
  - **Section 3:** Setting up Build CI/CD Pipelines for Projects using Azure
  - **Section 4:** Setting up Release CI/CD Pipelines for Projects using Azure
  - **Section 5:** Introduction to Azure Repositories and Release components
  - **Section 6:** Adding Test Automation Jobs to the Release Pipelines

- 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 3 -> Course 7 -> Application Debugging -Backend

Course Overview

The **Course 7** of the **Stage 3** provides a comprehensive introduction to Web API debugging, equipping learners with essential skills to troubleshoot and optimize API functionalities effectively. Through hands-on activities and real-world scenarios, students will explore powerful debugging tools like Visual Studio and Postman to identify and resolve issues in API calls. The course progresses from fundamental debugging concepts to advanced techniques, ensuring learners are well-prepared to handle diverse debugging challenges in modern web development.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand the fundamental principles of Web API debugging and its role in web application development.
- Utilize Visual Studio's debugging tools to identify and fix common API issues.
- Implement best practices for handling exceptions in API calls.
- Use Postman effectively for API testing and debugging.
- Analyze and resolve issues in API calls through Postman's advanced debugging features.
- Apply advanced debugging techniques to troubleshoot complex API scenarios and optimize performance.

Day 36 - Forenoon

Application Debugging - Backend

Key Topics: Overview of Web API Debugging, Debugging in Visual Studio, Handling Exceptions, Using Postman for API Testing, Debugging API Calls with Postman, Advanced Debugging Techniques

Learn and Practice

Walk through the following articles and try out the examples given in them.

- Debug ASP.NET Core with the Visual Studio debugger
- Handle errors in ASP.NET Core web APIs
- How To Use Postman With ASP.NET Core Web API Testing
- Remote Debugging ASP.NET Core Application on a Remote IIS

IDP – Project Activities

Day 36 - Afternoon, 37 - Forenoon

Review, Demo

- You will conduct an internal demo on the completed modules and rework it if required.

Interim Evaluation

Day 37 - Afternoon, 38, 39 - Forenoon

Interim Evaluation (Project + Technical)

- Interim evaluation will be conducted on this day, and the mode will be a video interview on the Tekstac platform.

Day 39 - Afternoon

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

Stage 4 - Full Stack Development: Front-end Development and Version Control

Overview

Stage 4 deals with the Niche skills such as React, GIT 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.

- GIT
- React (v18)
- 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 -> GIT

Course Overview

The **Course 1** of the **Stage 4** training focuses on **Git**, a distributed version control system widely used in software development. In this module, learners will explore the fundamentals of Git, including version control concepts, repository management, branching strategies, and collaboration workflows. They will learn how to use Git commands for tracking changes, resolving conflicts, and managing project history. Understanding Git is essential for modern application development as it enables teams to efficiently collaborate on code, track changes, revert to previous versions, and maintain a well-organized codebase.

Learning Objectives

After completing this course, GenCs will be able to:

- Understand the purpose, benefits, and types of version control systems.
- Explain Git as a Distributed Version Control System (DVCS) and its core components.
- Install and configure Git with basic settings (username, email).
- Create, initialize, and clone repositories using git init and git clone.
- Stage and commit changes with git add and git commit.
- Monitor repository status with git status and view history using git log.
- Create and manage branches, and merge changes while resolving conflicts.
- Explore branching strategies (feature, release, Git Flow).
- Link, push, and pull changes between local and remote repositories.
- Collaborate using workflows (feature branching, Gitflow) and manage pull requests effectively.

Day 40

GIT

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

Learn and Practice

The Git & Github Bootcamp

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope. course
  - **Section 2:** Introducing…Git!
  - **Section 3:** Installation and Setup
  - **Section 4:** The very Basics of Git: Adding & Committing
  - **Section 6:** Working With Branches
  - **Section 7:** Merging Branches, Oh Boy!

    - **Section 8:** Comparing Changes with Git Diff
    - **Section 12:** Fetching & Pulling
    - **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.

Offline Hands-On (Additional Practice)

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

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.

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

Stage 4 -> Course 2 -> React (v18)

Course Overview

The **Course 2** of the **Stage 4** This course provides a comprehensive introduction to modern front-end development using React and back-end development with Node.js. Designed for aspiring developers, the curriculum emphasizes building dynamic, responsive, and efficient web applications. Students will learn the fundamentals of React, including JSX, components, state management, and routing, as well as advanced topics such as hooks, context API, Redux, and testing. Through hands-on projects and real-world scenarios, learners will gain the technical expertise and problem-solving skills needed for a successful career in web development.

Learning Objectives

After completing this course, GenCs will be able to:

- Set up a development environment for React projects and understand the basics of JSX.
- Develop modular and reusable components using props, state, and event handling.
- Utilize React Hooks to manage state and side effects effectively.
- Implement navigation using React Router and build forms with controlled components.
- Leverage Context API for simple state management across components.
- Implement advanced state management solutions using Redux.
- Handle data fetching and asynchronous operations effectively within React applications.
- Write and execute unit tests to ensure the reliability and functionality of React components.
- Understand the fundamentals of Node.js and its core modules.

- Explore the basics of asynchronous programming in Node.js to handle back-end functionality.

Day 41

React (v18)

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

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 1:** Getting Started
  - **Section 2:** JavaScript Refresher
  - **Section 32:** React + TypeScript
- 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 **React**.

Day 42

React (v18)

Key Topics: JSX and Rendering Elements, Components and Props

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 3:** React Essentials – Components, JSX, Props, State & More
  - **Section 14:** An Alternative Way Of Building Components: Class-based Components
- 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.

- Stock securities-component
- Display Current Date and Time
- Share market-statistics
- One Spot Furniture -component
- Telecom-statistics
- Apple Inc. Market - state
- Stock Securities-Props
- Stock Securities-double and array props
- FunTimeTicket – PropTypes
- Book House - Cross component communication

Offline Hands-On (Additional Practice)

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

Day 43

React (v18)

Key Topics: State and Event Handling, React Hooks, Component Lifecycle and Effects

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 4:** React Essentials - Deep Dive
  - **Section 11:** Handling Side Effects & Working with the useEffect() Hook
- 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.

- Quote For the Day - Event
- 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

Offline Hands-On (Additional Practice)

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

Day 44

React (v18)

Key Topics: React Router and Navigation, Forms and Controlled Components

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 22:** Building a Multi-Page SPA with React Router
  - **Section 17:** Working with Forms & User Input
  - **Section 18:** Handling Forms via Form Action
- 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.

- Course Details with Form_Final
- Student Details using Map_Final
- Sort Applicants List_Final
- Search for Course availability
- Tech Valley - Styles
- Stock Securities - Report - keys and styles - Final
- Style table using External CSS – Final
- Router - Final

Offline Hands-On (Additional Practice)

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

Day 45

React (v18)

Key Topics: Context API for State Management

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 10:** React's Context API & useReducer - Advanced State Management
- 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 **React**.

Code Challenge (For Practice Only)

Take on the following Code Challenge in the Tekstac Learning Path to assess your skill level in React. 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 - React

Day 46

React (v18)

Key Topics: Advanced State Management with Redux

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 20:** Diving into Redux (An Alternative To The Context API)
  - **Section 21:** Advanced Redux
- 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 **React**.

Day 47

React (v18)

Key Topics: Asynchronous Operations and Data Fetching

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - **Section 15:** Sending Http Requests (e.g., Connecting to a Database)
- 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 **React**.

Day 48

React (v18)

Key Topics: Testing in React

Continuous Learning: Technical Enablement

React - The Complete Guide 2025 (incl. Next.js, Redux)

- Walkthrough the following Udemy course sections and focus on the corresponding topics within our training curriculum's technical scope.
  - Section 31: Testing React Apps (Unit Tests)
- 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 49

React (v18)

Key Topics: Introduction to Node.js, Core Node.js Modules

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)
- 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 50

React (v18)

Key Topics: 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 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.

Day 51 - Forenoon

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

Stage 4 -> Course 3 -> Application Debugging - Front-end

Course Overview

The **Course 3** of the **Stage 4** training focuses on front-end application debugging in 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 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 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 51 - Afternoon

Application Debugging - Front-end

Key Topics: Debugging Basics, Debugging React Applications, Advanced Debugging Techniques

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.

IDP- Project Activities

Day 52, 53

IDP - Sprint 2 Development (Integration of front-end with backend)

- These days will be spent on integrating various aspects of the project such as front-end and back-end

Final Evaluation

Day 54, 55, 56

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

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 are the entry criteria for Stage 1 Delta Assessment?**

Ans: The eligibility criterion for the Stage 1 Delta Assessment is 100% hands-on completion, along with an attempt and submission in both CC and ICT.

7. **What skills are covered in the Stage 1 Delta Assessment?**

   Ans: The delta skills of Stage 1 are covered in this assessment. Maximum THREE attempts will be provided to pass with a minimum score of 70%.

8. **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.

9. **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.

10. **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.

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

    Ans: Batch Owner is your point of contact.