Skills	Topics	Sub Topics
GIT Basics	Introduction to Version Control	Version Control Concepts-Definition and Purpose, Benefits of Version Control, Types of Version Control Systems
	Understanding Git	What is Git-Introduction to Git, Git as a Distributed Version Control System (DVCS); Git Components-Working Directory, Staging Area, Repository
	Setting Up Git	Installing Git-Download and Installation Steps, Configuring Basic Settings (username, email); Creating a Git Repository-Initializing a New Repository, Cloning an Existing Repository;
	Basic Git Commands	git init and git clone - Initializing a New Repository, Cloning an Existing Repository; git add-Staging Changes, Using Wildcards; git commit-Committing Changes, Adding Commit Messages; git status-Checking the Status of Your Repository; git log-Viewing Commit History, Options for Customizing Log Output;
	Branching and Merging	Introduction to Branching-Creating and Managing Branches, Switching Between Branches; Merging Changes-Merging Branches, Handling Merge Conflicts; Branching Strategies-Feature Branching, Release Branching, Git Flow Workflow;

	Remote Repositories	Adding Remote Repositories-Linking Local and Remote Repositories, Multiple Remotes; git pull and git push-Pulling Changes from a Remote Repository, Pushing Changes to a Remote Repository; Handling Remote Branches-Tracking and Creating Remote Branches;
	Collaborating with Git	Forking and Pull Requests-Forking a Repository, Creating and Managing Pull Requests; Git Collaboration Workflows-Centralized Workflow, Feature Branch Workflow, Forking Workflow, Gitflow Workflow
CI/CD Fundamentals	Introduction to CI/CD	What is CI/CD?, Benefits of CI/CD, CI/CD Pipeline Overview
	Continuous Integration (CI)	Source Code Management (SCM), Automated Builds, Automated Testing in CI, Code Quality and Static Analysis, Build Triggers and Notification Systems
	Continuous Delivery (CD)	Concepts of Continuous Delivery, Deployment Strategies, Artifact Management, Environment Management, Automated Deployment Process
	Continuous Deployment	Understanding Continuous Deployment, Release Automation, Monitoring and Logging Post-Deployment, Feature Flags and A/B Testing
HTLM5	Introduction	Need and Benefits of HTML, Setup, Browser, BOM and DOM, DOCTYPE, Character Encoding, <script> , <link>, , HTML5 Document, Comments

	Getting Started	Visual Studio Code features, Google Chrome Developer tools, Inspect document
	Elements & Attributes	Formatting Tags, List, Table, Form & Input Tags, Images, Styles, placeholder, inline and block elements, id vs class attributes
	Navigation	Navigation tags, hyper link, reference to intermediate section
	Events	onblur, onchange, onclick, form related events, load events, key events, mouse events, ondblclick, onbeforeonload, oncanplay
	Web Forms 2.0	<input> element in HTML5, <output>, placeholder attribute, autofocus, required
	Web Storage	Session storage, local storage and Delete web storage
	Web SQL Database	openDatabase, transaction, transaction
	Geo location	Geolocation Methods, Location Properties, Handling Errors, Position Options
CSS3	Introduction	Need and Benefits of CSS, Setup, CSS Syntax, CSS Comments, Including CSS in HTML Documents(Inline styles, Embedded styles, External style sheets)
	Selectors	CSS3 Selectors - Universal Selector, Element Type Selector, Id Selectors, Class Selectors, Grouping Selectors
	Styling	CSS Color, CSS Background, CSS Fonts, CSS Text, CSS Links, CSS Lists, CSS Tables

	Box Model	CSS Box Model, Margin, padding, border, Outline, Visibility vs. Display, Multiple Columns
	Advanced	Media Queries, RWD
JavaScript	Introduction to JavaScript	What is JavaScript?, Setting Up JavaScript Development Environment
	JavaScript Basics	Syntax and Statements, Data Types, Operators
	Control Flow	Conditional Statements, Loops, Error Handling
	Functions and Scope	Function Basics, Scope and Closures, Higher-Order Functions
	Objects and the Document Object Model (DOM)	Objects in JavaScript, JavaScript Prototypes, The DOM, Event Handling
	Arrays and Array Methods	Array Basics, Array Methods
	Asynchronous JavaScript	Understanding Asynchronous Programming, Callbacks and Callback Hell, Promises, Async/Await Syntax
	JavaScript ES6+ Features	Let and Const, Template Literals, Destructuring, Rest and Spread Operators, Modules, Default Parameters
	JavaScript in Web Development	Working with Forms, AJAX and Fetch API
	Debugging and Testing JavaScript	Debugging Tools, Testing JavaScript Code
	Introduction to JavaScript Frameworks and Libraries	Overview of Popular JavaScript Frameworks, Using jQuery for DOM Manipulation

Bootstrap 5	Introduction to Bootstrap 5	Overview of Bootstrap Framework, Setting Up Bootstrap 5, Bootstrap Structure and Files
	Bootstrap Grid System	Fundamentals of Responsive Grid Layout, Column Layouts and Grid Classes, Alignment and Reordering in Grid, Responsive Flexbox Utilities
	Bootstrap Components	Typography, Forms, Buttons, Navbars and Navigation, Cards and Media Objects
	Bootstrap Utilities and Helpers	Spacing Utilities, Colors and Backgrounds, Display and Visibility, Borders, Shadows, and Rounded Corners, Positioning Utilities
	Advanced Bootstrap 5 Features	Icons with Bootstrap Icons, Bootstrap 5 JavaScript Plugins, Customization with Sass
jQuery	Introduction to jQuery and its Features	What is jQuery?, Features of jQuery
	Basic Components of jQuery	jQuery Library Inclusion, The $() Function, jQuery Methods
	DOM Manipulation and Events in jQuery	DOM Manipulation, Working with Events, Event Helpers
ANSI SQL Using MySQL	Introduction to ANSI SQL and MySQL	Overview of ANSI SQL, Introduction to MySQL, Importance of Standard SQL
	Data Retrieval with SELECT Statement	Basic SELECT syntax, Retrieving specific columns, Filtering data with WHERE clause, Sorting results with ORDER BY
	Filtering and Sorting Data	Using logical operators in WHERE clause, Combining conditions with AND, OR, Sorting data using multiple columns

	Aggregate Functions and Grouping	Introduction to aggregate functions (SUM, AVG, COUNT, etc.), GROUP BY clause for grouping results, HAVING clause for filtering grouped data
	Joins and Subqueries	INNER JOIN, LEFT JOIN, RIGHT JOIN, Self-joins and cross joins, Subqueries in SELECT, WHERE, and FROM clauses, Correlated Subqueries
	Data Modification with INSERT, UPDATE, DELETE	Inserting data into tables, Updating existing records, Deleting records from tables
	Creating and Modifying Tables	Creating tables with CREATE TABLE, Modifying table structure with ALTER TABLE, Deleting tables with DROP TABLE
	Indexes and Constraints	Creating indexes for better query performance, Defining primary and foreign key constraints, Ensuring data integrity with UNIQUE and CHECK constraints
Core Java (v21)	Introduction to Java	Overview of Java Platform and Editions, Installation and Setup of Java Development Environment (JDK 21), Introduction to Java Development Kit (JDK), Java Runtime Environment (JRE), and JVM, Writing, Compiling, and Running Java Programs
	Basics of Java Programming	Data Types and Variables, Operators, Control Statements
	Object-Oriented Programming in Java	Classes and Objects, Encapsulation, Inheritance, Polymorphism, Abstraction and Interfaces
	Exception Handling	Introduction to Exceptions, Handling Exceptions, Custom Exceptions

	Java Collections Framework	Introduction to Collections, List Interface and Implementations, Set Interface and Implementations, Map Interface and Implementations, Queue Interface and Implementations, Stream API
	Functional Programming in Java	Lambda Expressions, Functional Interfaces, Method References, Optional Class, Streams and Parallel Streams
	Java I/O and File Handling	Java I/O Streams, File Handling with java.nio.file Package, Serialization and Deserialization
	Multithreading and Concurrency	Introduction to Multithreading, Thread Lifecycle and Thread Control, Synchronization, Concurrency Utilities (java.util.concurrent package)
	Application Debugging Using IntelliJ IDEA	What is Debugging?, Importance of Debugging, Overview of Debugging Tools, Setting Up IntelliJ IDEA, Debugging Environment Overview, Setting Breakpoints, Running the Debugger, Inspecting Variables and Expressions, Using Watches, Exception Breakpoints, Thread Debugging, Memory and Performance Profiling
	Introduction to Reactive Programming	Reactive Programming Fundamentals, Reactive Streams and Backpressure, Project Reactor Basics, Concurrency in Reactive Programming
	Working with Java Modules	Introduction to Modular Programming, Creating and Using Modules, Modularity in Java 17

	Java Networking	Basics of Networking in Java, TCP and UDP Communication, HTTP Client
	Reverse Engineering Concepts	Introduction to Reverse Engineering, Decompilation in Java, Java Bytecode Analysis, Reflection and Introspection in Java, Debugging and Code Analysis Techniques, Code Obfuscation and Deobfuscation, Working with Legacy Code and APIs, Software Security through Reverse Engineering
	Introduction to Java Database Connectivity (JDBC)	JDBC Overview, Executing SQL Queries, Handling Transactions in JDBC
	Java 21 Features	Sealed Classes, Pattern Matching for switch, Sequenced Collections, Virtual Threads
Design Principles	SOLID Principles	What & Why, The SOLID Principles of Object-Oriented Programming - The Single Responsibility Principle, The Open-Closed Principle, The Liskov Substitution Principle, The Interface Segregation Principle, The Dependency Inversion Principle ;
Spring with Maven	Introduction to Spring Framework	Overview of the Spring Framework, Inversion of Control (IoC) and Dependency Injection (DI), Spring modules: Core, AOP, Data Access, ORM, MVC, etc., Benefits of using Spring in Java applications
	Setting up a Spring Project with Maven	Introduction to Maven build tool, Creating a new Maven project, Adding Spring dependencies in the pom.xml file, Configuring Maven for building and managing dependencies

Spring IoC Container	Understanding the IoC container, Configuring the Spring IoC container using XML, Defining beans and their dependencies, ApplicationContext and BeanFactory
Spring Bean Configuration	Using annotations for bean configuration, Component scanning and Stereotype annotations, Java-based configuration with @Configuration, Mixing XML and Java-based configurations
Dependency Injection in Spring	Constructor injection, Setter injection, Autowiring dependencies, Qualifiers for resolving autowiring conflicts, Using @Resource and @Inject annotations
Spring AOP (Aspect-Oriented Programming)	Introduction to AOP concepts, Creating aspects and advice, Pointcuts and Joinpoints, AOP proxying mechanisms, Integrating AOP with Spring applications
Spring MVC and ORM	Overview of MVC and ORM, Configuration, Controller Layer, Model Layer, View Layer, Form Handling, Querying, Validation, Exception Handling
Spring Boot (Introduction)	Overview of Spring Boot, Simplifying Spring configuration with Boot, Creating a Spring Boot application, Auto-configuration and convention over configuration
Reactive Programming with Spring WebFlux	Overview of Reactive Web Framework and Non-blocking I/O, Creating and Configuring Reactive Controllers, Handling HTTP Requests and Responses with Mono and Flux, Integrating with NoSQL Databases using Reactive Repositories, Writing Unit and

		Integration Tests for Reactive Services
Spring REST with Spring Boot 3	Introduction to Spring REST and Spring Boot 3	Overview of RESTful architecture, Introduction to Spring REST, Benefits of using Spring Boot for RESTful services, Setting up a Spring Boot project for REST, What's New in Spring Boot 3?
	Building a Simple REST Controller	Creating a basic REST controller, Defining request mappings, Handling HTTP methods (GET, POST, PUT, DELETE), Returning JSON responses
	Request and Response Handling	Handling path variables and query parameters, Request body and form data processing, Customizing response status and headers, Exception handling in REST controllers
	RESTful Resource Representation with DTOs	Introduction to Data Transfer Objects (DTOs) Mapping entities to DTOs, Customizing JSON serialization and deserialization, Managing versioning and backward compatibility
	RESTful CRUD Operations	Implementing Create, Read, Update, and Delete operations, Utilizing HTTP methods for CRUD operations, Validating input data with annotations, Optimistic locking for concurrent updates
	RESTful HATEOAS	Understanding HATEOAS (Hypermedia as the Engine of Application State), Adding links to resources, Building and consuming hypermedia-driven APIs

	Content Negotiation and Media Types	Configuring content negotiation Supporting different media types (JSON, XML), Using the Accept header for content negotiation, Producing and consuming custom media types
	Spring Boot Actuator for REST Monitoring	Integrating Spring Boot Actuator, Monitoring and managing RESTful services, Exposing custom metrics, Securing and customizing Actuator endpoints
	Security and Authentication in RESTful APIs	Securing RESTful endpoints with Spring Security, Implementing authentication and authorization, Token-based authentication (JWT), Handling Cross-Origin Resource Sharing (CORS)
	Testing RESTful APIs	Unit testing REST controllers with JUnit and Mockito, Integration testing for REST services, Using Spring Test and MockMvc, Test coverage and best practices
	Documenting RESTful APIs	Introduction to API documentation tools (Swagger, Springdoc), Documenting REST APIs with Swagger/OpenAPI, Generating API documentation, Best practices for API documentation
TDD using JUnit and Mockito	Introduction to Test-Driven Development (TDD)	Definition and principles of TDD, Benefits of TDD in software development, TDD lifecycle: Red-Green-Refactor, TDD vs. traditional development
	Getting Started with JUnit	Overview of JUnit framework, Writing and running simple JUnit tests, Annotations in JUnit (e.g., @Test, @Before, @After), Assertions for verifying expected outcomes, Test fixture setup and teardown

	Advanced JUnit Features	Parameterized tests, Test suites and categories, Test execution order, Exception testing, Timeout and performance testing
	Mockito Basics	Introduction to Mockito, Mocking and stubbing, Verifying interactions, Argument matching, Handling void methods
	Testing Spring Applications with JUnit and Mockito	Overview of Spring testing, Testing controllers, services, and repositories, Integration testing with Spring Boot, Mocking dependencies in Spring tests
	Mocking External Dependencies	Mocking databases and repositories, Mocking external services (e.g., RESTful APIs), Mocking file I/O and network interactions, Strategies for testing code with external dependencies
Spring Data JPA with Spring Boot, Hibernate	Introduction to Spring Data JPA	Overview of Spring Data JPA, Relationship between JPA and Spring Data JPA, Advantages of using Spring Data JPA in Spring Boot
	Setting Up a Spring Boot Project with Spring Data JPA	Creating a Spring Boot project, Adding Spring Data JPA dependencies, Configuring the application properties for database connection
	Entity Mapping	Introduction to JPA entities, Mapping entities to database tables, Defining primary keys and relationships, Using annotations like @Entity, @Table, @Id, @GeneratedValue, etc.

Spring Data Repositories	Overview of Spring Data repositories, Creating repositories for entities, Derived queries from method names, Using Query DSL with @Query annotation, Custom query methods
CRUD Operations with Spring Data JPA	Implementing basic CRUD operations, Using JpaRepository methods for common operations, Executing custom queries with the repository
Query Methods and Named Queries	Defining query methods in repositories, Using keywords in query methods, Named queries with @NamedQuery and @NamedQueries, Executing dynamic queries
Pagination and Sorting	Implementing pagination with Page and Pageable, Sorting query results, Combining pagination and sorting
Auditing with Spring Data JPA	Enabling entity auditing, Using @CreatedBy, @LastModifiedBy, @CreatedDate, and @LastModifiedDate, Configuring auditing properties
Spring Data JPA Projections	Creating projections for specific data subsets, Interface-based and class-based projections, Using @Value and constructor expressions, Controlling the fetched data with projections
Spring Data JPA and Spring Boot Integration	Leveraging Spring Boot auto-configuration, Customizing data source configuration, Externalizing configuration with application properties, Managing multiple data sources

	Spring Data JPA and Hibernate	Introduction to Spring Data JPA and Hibernate, Hibernate-specific Features - Leveraging Hibernate-specific annotations, Configuring Hibernate dialect and properties, Batch processing with Hibernate;
Logging, and Continuous Code Quality	SLF4J, Lombok	SLF4J vs. Log4J vs. Lombok, SLF4J - Env Setup, Sample Logging, SLF4j - error messages, warning levels, parameterized logging, different appenders Lombok - Configuring Project using Lombok Logging using Lombok annotations
	Continuous Code Quality using SONAR	Clean as you code - SONAR Intro, benefits Using SonarQube with Maven - Update connect URLs, Proxies in POM Sonar architecture - how it works? SonarQube client and Server, Sonar Database Running Maven Sonar Goals for Static Code Analysis Use SonarQube to find the following: Finding Duplicate Code, Cyclomatic Complexity, Sphagetti Design, Lack of Unit Tests, Improper Coding Standards, Potential Bugs, Insufficient Comments,Interpreting Sonar Logs
Angular (v16)	Introduction to Angular and Setting Up Environment	Overview of Angular, Setting Up the Development Environment-Installing Angular CLI and Creating a New Angular Project, Angular Project Structure and Files, Running and Building Angular Applications
	TypeScript Essentials for Angular	TypeScript Basics, Using TypeScript in Angular

	Angular Components	Creating Components, Component Interaction - Data Binding: Property Binding, Event Binding, and Two-way Binding, Component Lifecycle Hooks (ngOnInit, ngOnChanges, etc.), Parent-Child Component Communication (@Input and @Output)
	Directives and Pipes	Built-in Directives - Structural Directives: *ngIf, *ngFor, and *ngSwitch, Attribute Directives: ngClass, ngStyle, ngModel, Custom Directives, Pipes - Built-in Pipes (e.g., date, uppercase, currency), Creating Custom Pipes
	Angular Forms	Template-driven Forms - Basics of Template-driven Forms and Form Validation, Binding Data with ngModel and Handling Form Submission Reactive Forms - Setting Up Reactive Forms with FormBuilder, Form Control, FormGroup, and FormArray, Reactive Form Validation (Built-in Validators, Custom Validators)
	Dependency Injection and Services	Introduction to Dependency Injection, Creating and Using Services, Hierarchical Dependency Injection
	Angular Routing and Navigation	Setting Up Routing - Configuring Routes in app-routing.module.ts, Route Parameters and Query Parameters, Nested Routes and Lazy Loading Modules
	Router Features	Router Guards: CanActivate, CanDeactivate, Resolve, Router Events and Navigation Lifecycle, Passing Data Between Routes

	HTTP Client and APIs	Using Angular HTTP Client - Setting Up HTTPClientModule and Making HTTP Requests, GET, POST, PUT, DELETE Requests with HTTP Client , Handling API Responses - Observables and Promises in Angular, Error Handling and Retry Strategies, Interceptors for Modifying Requests and Responses
	State Management in Angular	Introduction to State Management, Using Services for State Management, NgRx for Advanced State Management
	Reactive Programming with RxJS	RxJS Observables and Operators, Reactive Patterns in Angular
	Testing Angular Applications	Unit Testing with Jasmine and Karma, Component and Service Testing, End-to-End (E2E) Testing
	Introduction to Node.js	Overview of Node.js and its role in web development, Node.js architecture: Single-threaded, non-blocking I/O, Node.js Module System - Understanding modules and require(), Exporting and importing modules
	Core Node.js Modules	File System Module (fs) - Reading and writing files, Basic asynchronous operations with fs (e.g., readFile and writeFile), HTTP Module - Creating a basic HTTP server, Handling simple HTTP requests and responses
	Asynchronous Programming Basics	Callbacks and Promises - Using callbacks for asynchronous operations, Introduction to Promises and .then() syntax, Async/Await Basics - Using async/await for cleaner asynchronous code

React (v18)	Introduction to React and Setting Up the Environment	Key Concepts: Components, State, and Props, Differences Between Class and Functional Components, Introduction to React v18 Features (e.g., Automatic Batching, Concurrent Rendering), Installing Node.js and npm, Creating a React Application with create-react-app, Understanding the Project Structure
	JSX and Rendering Elements	Writing JSX and Embedding Expressions, JSX Syntax and Rules, Differences Between JSX and HTML, Rendering Elements with ReactDOM.createRoot, Understanding React’s Virtual DOM, Conditional Rendering with Ternaries and && Operators
	Components and Props	Defining and Exporting Components, Understanding Component Reusability and Nesting, Passing Data with Props, Prop Types and Default Props, Parent-to-Child Data Flow
	State and Event Handling	Using useState Hook to Manage Local State, Differences Between Props and State, Event Handling Syntax in React, Passing Event Handlers as Props, Using Synthetic Event
	React Hooks	Benefits of Hooks and Why They Replaced Class Components, Basic Rules of Using Hooks, useState for State Management, useEffect for Side Effects and Lifecycle, useRef for DOM Manipulation and Persistent Values

	Component Lifecycle and Effects	Executing Side Effects on Mount and Update, Cleanup Functions and Component Unmounting, Using Dependency Arrays for Conditional Execution, Common Pitfalls and Best Practices with useEffect
	React Router and Navigation	Installing React Router and Configuring Routes, Basic Routing with BrowserRouter, Route, and Link, Nested Routes and Route Parameters, Redirects and Programmatic Navigation, useNavigate and useParams Hooks for Navigation
	Forms and Controlled Components	Building Controlled Components with State, Managing Form Data and Submitting Forms, Basic Validation in Controlled Components, Integrating Form Libraries (e.g., Formik or React Hook Form)
	Context API for State Management	Understanding Prop Drilling and When to Use Context, Creating and Providing Context with createContext and Provider, Accessing Context with useContext, Best Practices for Using Context API in Larger Applications
	Advanced State Management with Redux	Installing Redux and React-Redux Libraries, Understanding Redux Concepts: Store, Actions, and Reducers, Dispatching Actions and Reading State with useSelector and useDispatch, Best Practices for Managing Complex State
	Asynchronous Operations and Data Fetching	Using fetch API to Make HTTP Requests, Error Handling and Displaying Loading States, Benefits of React Query for Caching and Synchronization, Optimistic Updates and Error Boundaries

	Testing in React	Setting Up Jest and Writing Simple Unit Tests, Testing Components, Props, and State, Testing User Interactions with React Testing Library, Overview of E2E Testing with Cypress (Basic Introduction)
	Introduction to Node.js	Overview of Node.js and its role in web development, Node.js architecture: Single-threaded, non-blocking I/O, Node.js Module System - Understanding modules and require(), Exporting and importing modules
	Core Node.js Modules	File System Module (fs) - Reading and writing files, Basic asynchronous operations with fs (e.g., readFile and writeFile), HTTP Module - Creating a basic HTTP server, Handling simple HTTP requests and responses
	Asynchronous Programming Basics	Callbacks and Promises - Using callbacks for asynchronous operations, Introduction to Promises and .then() syntax, Async/Await Basics - Using async/await for cleaner asynchronous code
Microservices with Spring Boot 3 and Spring Cloud	Introduction to Microservices Architecture (MSA)	Overview of monolithic vs. microservices architecture, Advantages and challenges of microservices, Characteristics of microservices, Use cases and scenarios suitable for microservices
	Spring Cloud for Microservices	Introduction to Spring Cloud, Features and components of Spring Cloud, Configuring microservices with Spring Cloud, Service discovery and registration with Spring Cloud Netflix Eureka

	Spring Security for Microservices	Overview of Spring Security, Securing microservices using Spring Security, Authentication and authorization in a microservices environment, Configuring security for RESTful APIs
	Centralized Authentication and Authorization	Implementing centralized authentication with OAuth 2.1/OIDC, Configuring Authorization Servers and Resource Servers, Using JSON Web Tokens (JWT) for secure communication, Single Sign-On (SSO) in a microservices architecture
	Microservices Communication with Spring Cloud	Inter-service communication patterns, Using Spring Cloud Feign for declarative REST clients, Service orchestration and choreography, Circuit Breaker pattern with Spring Cloud Circuit Breaker
	API Gateway and Edge Services	Introduction to API Gateways, Configuring API Gateway with Spring Cloud Gateway, Implementing edge services for routing and filtering, Load balancing and resilience patterns in an API Gateway
	Fault Tolerance and Resilience	Implementing fault tolerance with Spring Cloud Hystrix, Circuit Breaker and fallback mechanisms, Retrying and fallback strategies for resilience, Handling transient faults in microservices
	Spring Cloud Config	Externalized configuration in microservices, Configuring microservices using Spring Cloud Config, Dynamic configuration updates and refresh, Managing configuration properties for different environments

	Monitoring and Metrics in Microservices	Introduction to microservices monitoring, Using Spring Boot Actuator for monitoring endpoints, Integrating monitoring tools (Prometheus, Grafana), Application and system-level metrics in microservices
	Security Best Practices in Microservices	Role-based access control (RBAC) in microservices, Securing communication between microservices, Securing sensitive data in microservices, Implementing security policies and practices
Docker	Introduction to Docker	Container Introduction, Container architecture, Overview of Docker, Docker architecture - Docker file, Working with containers, Container Communication, Docker Compose, open source; create, deploy and run applications in containers, package application, virtual machine, image, container
	Hosting App using nginx, using command line and Dockerfile	nginx, pull, run, listing images, container name, detaching the process, port number, volumes, listing containers, listing non running containers, starting and stoping a container, Dockerfile, FROM, COPY, ENTRYPOINT, build, remove images and containers
	Hosting a MySQL database with schema creation using docker	docker-compose.yml, docker-compose up command, mapping MySQL data file to local folder, schema creation script execution definition, defining port, password definition, docker compose up, executing mysql client on the mysql server container

	Hosting a REST API Microservice using docker	Defining Dockerfile for REST API, building Dockerfile from docker-compose.yml, using depends_on in docker compose to define dependencies, using links to establish connectivity between REST API service and MySQL server, modify connection properties in REST API to connect to the docker instance
Cloud Computing using GCP	Introduction to Cloud Computing	Traditional IT Deployment, Virtualization, Service-Oriented Architecture (SOA), Cloud vs. On-Premises Data Centers, Pros and Cons of Cloud Computing
	Cloud Service Models	Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)
	Cloud Deployment Models	Public Cloud Model, Private Cloud Model, Hybrid Cloud Model, Community Cloud Model
	Cloud Service Providers	AWS, Azure, GCP
	Advantages of Cloud Computing and Various Services Provided by GCP	Benefits of Cloud Computing Various Services Available in GCP: Compute, Storage, Database, App Engine, Kubernetes Engine (GKE), Cloud Functions
	Introduction to GCP	Overview of Cloud Computing, Introduction to GCP Services, Creating a GCP Account, GCP Global Infrastructure

	GCP Compute Services	Google Compute Engine: Launching and Configuring Instances, Machine Types and Customization, Firewall Rules and SSH Keys Google Kubernetes Engine (GKE): Cluster Setup and Management, Deploying and Managing Containers Google App Engine: Standard and Flexible Environments, Deploying Applications Google Cloud Functions: Creating and Deploying Functions, Integrating with Other GCP Services
	GCP Storage Services	Google Cloud Storage: Bucket Creation and Configuration, Uploading and Downloading Objects, Storage Classes (Standard, Nearline, Coldline, Archive) Google Persistent Disk: Creating and Attaching Disks, Snapshots and Backups Google Filestore: Setting Up File Shares, Managing Storage Capacity
	GCP Networking	Virtual Private Cloud (VPC): Subnets, Route Tables, and Firewall Rules, VPC Peering and Shared VPC Cloud Load Balancing: HTTP(S) Load Balancer, Network Load Balancer, Internal Load Balancing Cloud CDN (Content Delivery Network): Configuring CDN, Integrating with Load Balancers
	GCP Database Services	Cloud SQL: Creating and Managing Database Instances, High Availability Configurations Cloud Spanner: Globally Distributed Databases, Scalability and Consistency

		Cloud Firestore: NoSQL Document Databases, Creating and Querying Collections BigQuery: Data Warehousing, Running Queries and Analyzing Data
	GCP Identity and Access Management (IAM)	Users, Groups, and Roles, IAM Policies and Permissions, Multi-Factor Authentication (MFA), Service Accounts and Key Management
	GCP Serverless Computing	Google Cloud Functions: Creating and Deploying Functions, Event-Driven Integrations Google Cloud Run: Deploying Containerized Applications, Managing Scalability and Traffic API Gateway: Creating and Managing APIs, Securing API Access
GCP DevOps	Introduction to DevOps in GCP	Overview of DevOps Practices, GCP Cloud Services for DevOps, Infrastructure as Code (IaC) with Terraform or Deployment Manager, Continuous Integration (CI), Continuous Deployment (CD)
	GCP Cloud Source Repositories	Introduction to Cloud Source Repositories, Setting Up Repositories, Basic Version Control with Git, Repository Workflows, Integration with Cloud Build and Cloud Deploy
	GCP Cloud Build	Introduction to Cloud Build, Configuring Build Triggers and Projects, Build Configuration with cloudbuild.yaml, Managing Build Artifacts, Customizing Build Environments and Steps

	GCP Cloud Deploy	Introduction to Cloud Deploy, Setting Up Deployment Pipelines, Defining Deployment Targets and Strategies, Managing Rollbacks and Revisions, Integrations with Cloud Build and Artifact Registry
	GCP Cloud Build Pipelines	Introduction to Cloud Build Pipelines, Creating and Managing Pipelines, Pipeline Execution and Artifact Management, Integrating Various GCP Services within Pipelines, Pipeline Monitoring and Troubleshooting