

Skill	Topic	Sub Topics
HTLM5	HTML5 - Introduction	Need and Benefits of HTML, Setup, Browser, BOM and DOM, DOCTYPE, Character Encoding, <script> , <link> , , HTML5 Document, Comments
HTLM5	HTML5 - Getting Started	Visual Studio Code features, Google Chrome Developer tools, Inspect document
HTLM5	HTML5 - Elements & Attributes	Formatting Tags, List, Table, Form & Input Tags, Images, Styles, placeholder, inline and block elements, id vs class attributes
HTLM5	HTML5 - Navigation	Navigation tags, hyper link, reference to intermediate section
HTLM5	HTML5 - Events	onblur, onchange, onclick, form related events, load events, key events, mouse events, ondblclick, onbeforeunload, oncanplay
HTLM5	HTML5 - Web Forms 2.0	<input> element in HTML5, <output>, placeholder attribute, autofocus, required
CSS3	CSS3 - Introduction	Need and Benefits of CSS, Setup, CSS Syntax, CSS Comments, Including CSS in HTML Documents(Inline styles, Embedded styles, External style sheets)
CSS3	CSS3 - Selectors	CSS3 Selectors - Universal Selector, Element Type Selector, Id Selectors, Class Selectors, Grouping Selectors
CSS3	CSS3 - Styling	CSS Color, CSS Background, CSS Fonts, CSS Text, CSS Links, CSS Lists, CSS Tables
CSS3	CSS3 - Box Model	CSS Box Model, Margin, padding, border, Outline, Visibility vs. Display, Multiple Columns
JavaScript	JavaScript Fundamentals	JavaScript Basics, DOM
JavaScript		Variables, Datatypes, Type Conversions, Operators, Comparisons, Interactions, Loops, Functions, Function expression, Control Flow, Conditionals
JavaScript	Javascript Deep Dive	Array, String, Date, Math, Number, Boolean, Functions, JSON, Regular Expression
JavaScript		Objects, Arrays and Functions, JSON
SQL Scripting	Introduction to Databases	Introduction to SQL/NoSQL, Advantages and Disadvantages
	Connecting to the Database, Normalization Forms	Table and Views

	Basic CRUD operations, Functions, Expressions and Clauses	DDL, DML, Query and Operations, Clauses, Aggregate Functions, String Functions, GROUP BY, Having, Rank Functions
	Joins and Subqueries	Joins, Sub queries
C#	C# Concepts	DataTypes And Variables, Value types , Reference types
		Functions / Methods, Decision Making
		Looping constructs, Arrays
		Inheritance and Polymorphism
		Advanced C# topics(Lambda Expressions, Access Modifiers, Lists, Arraylists vs Lists vs Arrays, Structs, Enums, Math Class, Random Class, Regular Expressions, DateTime)
		Linq, Exception Handling
		File IO, Read, Write, StreamReader, StreamWriter
		C# 7 features
Data access	ADO.NET	ADO.NET Architecture
		SqlConnection, SqlCommand, SqlDataReader
		DataAdapter, Data Set, Data Views
NUnit, Mock framework	Introduction	Overview, Installation of the environment
	Assertions	Equality, Condition
	Constraints and Attributes	Equal, Same as, Condition
	Running test	Visual studio support
	Mocking framework	Moq to mock database or any dependency, SetUp, Returns, IsAny
C# additional topics	Asynchronous programming	Async, await, task, named and optional parameters
	Multithreading	Thread usage, Threadpool, Multithreading, Thread.Wait, Thread.Sleep, Monitor in thread safety
	Other application types	Windows forms - stateful and stateless, Windows service

ASP.Net Core	Benefits of .Net Core	Problems in ASP.Net, Advantages of .Net core - Separation of concerns, reusability of business logic thru Controller, ease of unit testing, Console application creation in .Net core
	.Net Core components and its importance	dependency injection, registering services, middleware, ConfigureServices method, Configure method, appSettings.json, launchsettings.json
	MVC components in Web page rendering	folder structure of the MVC application, webpage rendering thru route, controller, action method, action results, razor engine rendering
	Data binding concepts in .Net core	2 way binding, Tag helpers, Use Ajax to render partial view
	Exception filter	Custom filter creation for exception handling
	Simple usage of Entity Framework	Scaffolding, parameter binding to Create methods
	Knowledge on using log4net for logging	Usage of middleware setting for log4Net
Code analysis		Minimum recommended ruleset, categories in the ruleset, most frequently seen Csharp code analysis rulesets, how to view the the code analysis resultset in error list in Visual studio, different actions for the ruleset
ASP.Net Core WebAPI	REST Concepts	REST features, Comparison with Web Service, HttpRequest and Response, ActionVerbs, HttpStatusCode
	Tools	Postman for testing, Swagger, Routing
	Attributes	AllowAnonymous, FromBody, ActionFilter, Custom exception filter
	Operations & Security	Update, Delete operation, CORS, Security using Bearer & JWT
		Concept of creating independent, loosely coupled services - Microservices
	Reference WebApi in ASP.Net core web application	Introduction to EF core, reason why EF core is introduced while EF available, use EF core in WebApi core to fetch data, SP usage to fetch data, multiple tables using LINQ
Cloud concepts	Introduction to Cloud concepts basics and its offerings	Cloud Introduction, Difference between on-premise server and cloud offerings - Server, Storage, Service, PAAS, SAAS. Google cloud, Microsoft cloud - Azure, Amazon cloud

CI/CD (Azure)	CI/CD Introduction	What is CI, What is CD, Various environments in CI/CD, Manual and Automated triggers in CICD pipeline
	Use CI/CD on Cloud	Use the Cloud environment and perform CI/CD operation Creating a build pipeline from the DEV branch Creating a deployment pipeline to release the core to different environments. Practice in DEV/ QA/ Stage Add YAML Creating Branches in Git Repositories
Container Basics	Containers	Container Introduction, Container architecture, Overview of Docker, Docker architecture, Docker file
	Kubernetes Basics	What, Why and Features of Kubernetes Kubernetes Cluster, Master, Node Kubernetes Architecture, Deploy an App
Cloud Services	Azure	Creating Azure App Service Web Apps Implement Azure Functions Develop solutions that use blob storage Implement secure cloud solutions Implement API Management Develop event-based solutions Develop message-based solutions Instrument solutions to support monitoring and logging Integrate caching and content delivery within solutions
Microservices	Microservices	Domain driven design, difference between Monolith & Microservices, Microservice key pattern - CQRS, Event driven architecture, Service to service communication, service registry, service discovery
		Cloud Management - DevOps project - CICD pipeline, Azure cloud repository, Kubernetes service on Azure, Deployment file usage, Azure command line commands(Create secret, kubernetes browser opening, admin setup)
		CD pipeline to use Deployment file to deploy to Azure Kubernetes service cluster(Service Discovery, Load Balancing), logging from application using ApplicationInsights
Angular	Angular basics, components, forms	SPA basics Works inside a browser, no page reload, example multi page application (MPA) sites, example SPA sites, technologies to develop SPA (HTML, CSS, JavaScript, jQuery, ajax, AngularJS, Angular, ReactJS, Ember.js) Traditional page life cycle in Servlet/ASP.NET/ASP.Net MVC, initial request, single time loading of HTML resources, AJAX, JSON Limitations of implementing SPA using JavaScript, Limitations of implementing SPA using jQuery, how to implement using angular, better performance since UI resources loads only once, gives better

		<p>user experience, clear separation of view, action and web service call, compatible with all major browsers</p> <p>Angular introduction Angular JS is version 1, called as Angular from version 2, how Angular JS works, key differences between Angular JS and Angular, Angular JS uses JavaScript, Angular uses TypeScript, Angular is command line based, Angular has a compilation step</p>
		<p>ECMAScript 6 European Computer Manufacturers Association (ECMA) standards organization, ECMAScript, scripting language specification, let, const, arrow functions, classes, TypeScript ES6 is related to TypeScript and Angular</p> <p>Typescript static typing, superset of JavaScript, typescript code compiles to JavaScript, browser compatible, open source, primary language of angular Install typescript npm package, write typescript program, compile and run Data Types (boolean, number, string, array, tuple, enum, any, void) interfaces, class and modules Defining functions with parameters' data type and return data type defined, using for .. of to iterate through arrays</p>
		<p>Angular Create new angular application in command line (ng new), execute the application (ng serve), understanding app component and app component html, importance of app module Interpolation, ng directives(ngIf, ngFor), pipes - builtin for date, Upper and lower case, currency</p> <p>Component ng generate Component, component.ts, component.html, @component decorator, selector, templateUrl, style, two way binding using [(ngModel)], click event and event handling, difference between normal and square brackets, Routing: router-outlet, routerLink, routerLinkActive</p> <p>Basics of Karma testing ng test, TestBed, fixture, component, debugElement, nativeElement, detectChanges(), it(), expect(), toEqual(), toBe()</p>

		<p>Template driven forms template reference variable, [(ngModel)], FormModule, Validation Properties (invalid, touched, dirty, errors.required, errors.minlength, json pipe, binding ngModel with radio buttons using [value], binding ngModel with select drop down using [value], [selected], (change) and \$event.target.value, karma testing with createNewEvent() and dispatchNewEvent()</p> <p>Reactive forms ReactiveFormsModule, FormControl, [formControl], formControlName, set form control values, FormGroup, Validators (required, minlength, maxlength), read value from FormGroup, definition of FormGroup in <form>, nested form groups, FormBuilder, Dynamic Forms, FormArray, formArrayName, FormBuilder.array(), FormBuilder.control(), push new form control</p>
		<p>Component interaction app selector with passing parameters, @Input decorator, filter() method, defining router link with parameters in app-routing.module.ts, defining router link in HTML template with parameter, ActivatedRoute, paramMap, router.navigate() method</p> <p>Guard generate guard, define protected routes in app-routing.module.ts with mapping to guard, define authentication service to retain the status of authentication, implement redirection in guard based on authentication status</p> <p>Services generate service, dependency injection, HttpClientModule, HttpClient, HttpHeaders, Observable, get(), post(), put(), delete(), subscribe, error handling</p>
Web Technologies	Node Js	<p>INTRODUCTION TO NODE WORLD</p> <ul style="list-style-type: none"> • Introduction to Node event driven programming with Typescript <p>MAKING A WEB SERVER</p> <ul style="list-style-type: none"> • Webserver basics • How to set up a router? • Serving static files / pages <p>HTTP OBJECT</p> <ul style="list-style-type: none"> • Introduction to HTTP Object • Processing POST Data • Handling File uploads

		<ul style="list-style-type: none"> • Using Node as HTTP Client • Call REST api • Implementing download Throttling <p>Develop REST API</p> <ul style="list-style-type: none"> • Introduction to REST development • Create server • Create REST api for get, put, post, del • Handle events, errors <p>ALL ABOUT SECURITY</p> <ul style="list-style-type: none"> • Basic Authentication methods • Password cryptography • Setting up HTTPS • How to safeguard your site from cross-site forgeries <p>WRITING CUSTOMIZED NODE MODULES</p> <ul style="list-style-type: none"> • Writing a functional module • Extending the module's API • Deploying a module to npm • Unit Testing • Socket.io
Web Technologies	React	<p>Getting Started with React</p> <ul style="list-style-type: none"> • What is React? • How it is different? • History • Current Standing • Big Picture <p>REACT Components</p> <ul style="list-style-type: none"> • Functional components • Building components • Rendering components • Using properties • State machine • Component as state machine • Library as stateful components

- Component Lifecycle
- Forms
- Composite Components
- Nesting Components
- One-way Flow
- Communicating with parent
- Rendering Collections

Styles

- Traditional styling
- Inline styling
- Applying Class
- Applying Style
- Styled Component
- Creating Component
- Using Component

Routing

- Static routing
- The SAP problem
- Install router support
- Browser Router
- Hash Router
- Dynamic Routing

React Hooks:

- useState
- useEffect
- Attaching effect to a state
- Custom Hooks

Testing in React:

- Pure Components
- Testing Stateful Components

Intro to Redux:

- Action
- Reducer
- Store