

## Angular

- Core Concepts (Modules, Components)
- Templates and Data Binding
- Directives (ngIf, ngFor, etc.)
- Services and Dependency Injection
- Routing and Navigation
- Forms (Template-driven & Reactive)
- HttpClient for API Interaction
- Pipes
- State Management with NgRx
- Change Detection Strategy (OnPush)
- Lazy Loading Modules
- Content Projection (ng-content)
- Dynamic Components
- Interceptors

## RxJs

- Observables and Observers
- Operators (map, filter, tap)
- Combination Operators (mergeMap, switchMap)
- Subjects (Subject, BehaviorSubject)
- Error Handling
- Higher-Order Mapping Operators
- Advanced Error Handling
- Schedulers
- Multicasting
- Custom Operators

## Node JS

- Node.js Fundamentals (Event Loop)
- Modules (CommonJS and ES6)
- File System (fs) Module
- Asynchronous Programming (Callbacks, Promises, Async/Await)
- HTTP Module
- Streams in Depth
- Worker Threads
- Clustering
- Data Validation and Sanitization
- Implementing Authentication with JWT
- Event Emitter
- WebSockets

## Express JS

- Routing
- Middleware
- Request and Response Objects
- Error Handling Middleware
- Building a RESTful API.
- Modular Routers
- Route-level Middleware
- Data Validation and Sanitization
- API Security Middleware

## Chart.js

- Setup & Installation (CDN vs. npm/yarn)
- Creating a Basic Chart (using <canvas>)
- The Configuration Object (type, data, options)
- Structuring the data Object (labels, datasets)
- Using the options Object for Basic Customization
- Making Charts Responsive
- Configuring Tooltips, Titles, and Legends
- Customizing Scales (Axes), Ticks, and Grid Lines
- Creating Bar and Line Charts
- Creating Pie and Doughnut Charts
- Creating Scatter and Bubble Charts
- Creating Radar and Polar Area Charts
- Combining Chart Types (Mixed Charts)
- Applying Colors and Styling (Borders, Backgrounds)
- Customizing Fonts and Text Elements
- Implementing Chart Animations
- Handling Events and User Interactions (onClick, onHover)
- Updating Chart Data and Options Dynamically
- Using Community Plugins to Extend Functionality
- Performance Optimization with Large Datasets
- Integration with a Framework (e.g., React, Angular, Vue)

## Devops

- Project scaffolding, building, packaging, deployment
- Preparing Docker
- Deployment Automation using Azure Pipelines
- Sonarcube rules enabling, issue tracking and closures
- Deployment in AKS in PODs
- Monitoring and Debugging.
- Automated Testing