Full Stack Asp.net MVC - Course Plan

1. Getting Ready
    a. Get to know each other
    b. Purpose of learning
    c. Proper way of learning
2. Tool Installation
    a. Installing Visual Studio
    b. Installing SQL Server
    c. Installing Git & Tortoise Git
    d. Installing Docker
    e. Installing VS code
3. Version Control (2 Classes)
    a. Why we need version control
    b. One step vs Two step version control
    c. Github overview
    d. Using git through SSH
    e. Understanding Gitflow
    f. Use case of tortoise git
    g. Using git commands
4. C# Overview (3 Classes)
    a. Linq
    b. Delegates & Events
    c. Threading
    d. C# 7 & 8 & 9 new features
    e. Collections
    f. Reflection
5. Asp.net Core MVC Nuts & Bolts (2 Classes)
    a. Understand project types and structure
    b. Model - View - Controller in depth
    c. Using NuGet
    d. Using configuration & routing
6. Understanding Razor (1 Class)
    a. Concept of layout and child page
    b. Tag helpers & Html helpers
    c. Partial view
    d. Sections
7. Dependency Injection Configuration (2 Classes)
    a. Using service collection
    b. Using Autofac
    c. Using serilog

8. Working with SQL Server (2 Class)
    a. Creating and managing database
    b. Working with tables
    c. Working with stored procedures
    d. Permission control
    e. Backup and restore database
    f. Concept of Dynamic SQL
9. Frontend development (3 Classes)
    a. Understand bootstrap grid
    b. Understanding SASS
    c. Implementing bootstrap theme
10. Working with TypeScript (2 Classes)
    a. Installation & configuration
    b. Important TypeScript language features
11. Data access with Ado.net (1 Class)
    a. Ado.net configuration
    b. Good practices
    c. CRUD example with Ado.net
    d. Using datatables
12. Working with Entity Framework (2 Classes)
    a. Understand ORM
    b. Code first overview
    c. Create Entity
    d. Apply Migrations
    e. Annotation and Relations
    f. Fluent API
    g. DbContext
    h. Data Seeding
    i. Using good practices
13. Creating Data Access Layer (3 Classes)
    a. Implement Repository Pattern
    b. Implement UnitOfWork Pattern
    c. Apply good practices
14. Implementation of CRUD example (3 Classes)
    a. Application of service layer
    b. Integration of Business Objects
    c. Using View Model to work with controller and service
15. Using AutoMapper for object cloning (1 Class)
    a. Installation and configuration
    b. Use of Automapper to map Entity and Business Objects
16. Understanding Security Issues (1 Class)
    a. SQL Injection
    b. CSRF

    c. Script Injection
    d. DDoS
    e. Importance of SSL Certificate & Https
    f. Database Backup, Error Logging, Encrypting to minimize damage

17. Implementing Asp.net Core Identity (3 Classes)
    a. Configure Identity
    b. Convert Page based code to MVC
    c. Configure Authentication
    d. Customize Service classes
    e. Move Identity Code in separate project
    f. Configure Authorization
    g. Role based authorization
    h. Policy based authorization
    i. Claim based authorization
18. Working with Web API (2 Classes)
    a. Why we need Web API
    b. Understanding Web API structure
    c. Working with Postman
    d. Implement JWT Authorization
    e. Apply JWT to restrict access
    f. Implementing CORS
19. Working with Worker Service (1 Class)
    a. Create and configuring worker service
    b. Deploying worker service
20. Dockering Asp.net Core project (2 Classes)
    a. Why we need docker
    b. Creating docker image
    c. Creating docker container
    d. Common docker commands
    e. Working with docker hub
    f. Deploy Asp.net Core MVC project in docker
    g. Using docker-compose
21. Testing Asp.net Project (3 Classes)
    a. Understand Unit Tests
    b. Unit Test good practices
    c. Using Automoq
    d. Understanding code coverage
    e. Assertion using Shouldly
22. Object Oriented Principles & Patterns (3 Classes)
    a. Understanding SOLID principles
    b. Understanding other 5 important principles
    c. Understanding creational patterns
    d. Use case, class & sequence diagrams

23. Create Web App with Angular (2 Classes)
    a. Creating Angular app using Visual studio & npm
    b. Angular project structure
    c. Connecting Angular App with Web API
    d. Create new component & module
    e. Apply good practices
24. Working in AWS (3 Classes)
    a. Understand cloud computing basics
    b. AWS Pricing
    c. Regions & Availability zones
    d. Use EC2 features
    e. Apply load balancing & auto scaling
    f. Use S3 bucket
    g. Use SQS
    h. Use DynamoDB
    i. Use AWS CLI
    j. Use AWS SDK
25. Project (5 Classes)
    a. Project requirement planning
    b. Project management process
    c. Code review
26. Exams (3 Classes)
    a. Mid term exam
    b. Final exam
    c. Project Demo