Full Stack Asp.net MVC - Syllabus

1. Tool Installation
   1. Installing Visual Studio
   2. Installing SQL Server
   3. Installing Git & Tortoise Git
   4. Installing Docker
   5. Installing VS code
   6. Installing Tracker Tool
2. Version Control (2 Classes)
   1. Why we need version control
   2. One step vs Two step version control
   3. Github overview
   4. Using git through SSH
   5. Understanding Gitflow
   6. Use case of tortoise git
   7. Using git commands
3. C# Overview (2 Classes)
   1. Linq
   2. Delegates & Events
   3. Important C# new features
   4. Reflection
4. Asp.net Core MVC Nuts & Bolts (2 Classes)
   1. Understand project types and structure
   2. Model - View - Controller in depth
   3. Using NuGet
   4. Using configuration & routing
5. Understanding Razor (1 Class)
   1. Concept of layout and child page
   2. Tag helpers & Html helpers
   3. Partial view
   4. Sections
6. Dependency Injection & Logger Configuration (2 Classes)
   1. Using service collection
   2. Using Autofac
   3. Using serilog
7. Working with SQL Server (2 Class)
   1. Creating and managing database
   2. Working with tables
   3. Working with stored procedures
   4. Permission control
   5. Backup and restore database
8. Frontend development (2 Classes)
   1. Understand bootstrap grid
   2. Implementing bootstrap theme
   3. Understanding SASS
9. Data access with Ado.net (1 Class)
   1. Ado.net configuration
   2. Good practices
   3. Using Ado.net to read & write in database
10. Object Oriented Principles & Patterns (3 Classes)
    1. Understanding SOLID principles
    2. Understanding other 5 important principles
    3. Understanding creational patterns
    4. Use case, class & sequence diagrams
11. Setting up project architecture (2 Classes)
    1. Understanding Clean Architecture
    2. Understanding Repository pattern
    3. Understanding Unit of Work pattern
    4. Understanding CQRS & Mediator
12. CRUD example with Ado.net (2 Classes)
    1. Using datatables
    2. Understanding & Implementing Dynamic SQL
    3. Implement Ado.net Data Access features
    4. Application of service layer
    5. Integration of Business Objects
    6. Using View Model to work with controller and service
13. Working with Entity Framework (4 Classes)
    1. Understand ORM
    2. Code first overview
    3. Create Entity
    4. Apply Migrations
    5. Entity Relationship
    6. Fluent API
    7. DbContext
    8. Data Seeding
    9. Using good practices
    10. Implementation of CRUD with Entity Framework
14. Using AutoMapper for object cloning (1 Class)
    1. Installation and configuration
    2. Use of Automapper to map model and entity Objects
15. Understanding Security Issues (1 Class)
    1. SQL Injection
    2. CSRF
    3. Script Injection
    4. DDoS
    5. Importance of SSL Certificate & Https
    6. Using captcha
    7. Database Backup, Error Logging, Encrypting to minimize damage
16. Implementing Asp.net Core Identity (3 Classes)
    1. Configure Identity
    2. Convert Page based code to MVC
    3. Configure Authentication
    4. Customize Service classes
    5. Move Identity Code in separate project
    6. Configure Authorization
    7. Role based authorization
    8. Policy based authorization
    9. Claim based authorization
17. Working with Web API (3 Classes)
    1. Why we need Web API
    2. Understanding Web API structure
    3. Working with Postman
    4. Implement JWT Authorization
    5. Apply JWT to restrict access
    6. Implementing CORS
    7. Integrating with CRUD operation
18. Working with Worker Service (1 Class)
    1. Create and configuring worker service
    2. Deploying worker service
19. Testing Asp.net Project (3 Classes)
    1. Understand Unit Tests
    2. Unit Test good practices
    3. Using Automoq
    4. Understanding code coverage
    5. Assertion using Shouldly
20. Dockering Asp.net Core project (2 Classes)
    1. Why we need docker
    2. Creating docker image
    3. Creating docker container
    4. Common docker commands
    5. Working with docker hub
    6. Deploy Asp.net Core MVC project in docker
    7. Using docker-compose
21. Working in AWS (5 Classes)
    1. Understand cloud computing basics
    2. AWS Pricing
    3. Regions & Availability zones
    4. Use EC2 features
    5. Apply load balancing & auto scaling
    6. Use S3 bucket
    7. Use SQS
    8. Use DynamoDB
    9. Use AWS CLI
    10. Use AWS SDK
    11. Concept of Microservice
22. Working with TypeScript (2 Classes)
    1. Installation & configuration
    2. Important TypeScript language features
23. Create Web App with Angular (2 Classes)
    1. Creating Angular app using Visual studio & npm
    2. Angular project structure
    3. Connecting Angular App with Web API
    4. Create new component
    5. Apply good practices