**Real Estate Document specific**

Contents

[**1.** **Structure of project** 1](#_Toc161070626)

[A. **WebAdmin** 1](#_Toc161070627)

[**B.** **Microservices/Microservices\_Net5** 1](#_Toc161070628)

[**C.** **Microservices/RealEstateServices** 1](#_Toc161070629)

[D. **WebClient**: 2](#_Toc161070630)

[**2.** **Technical solution collection** 2](#_Toc161070631)

[A. Microservices architecture 2](#_Toc161070632)

[B. API authentication with JWT – **Done (Microservices\_Net5)** 2](#_Toc161070633)

[C. Entity Framework - **Done (Microservices\_Net5)** 2](#_Toc161070634)

[D. MVC web – Done (Admin Web) 2](#_Toc161070635)

[E. Git action CICD 2](#_Toc161070636)

[F. Azure 2](#_Toc161070637)

[G. Docker, Docker compose 2](#_Toc161070638)

[H. Mongo DB 2](#_Toc161070639)

[I. Angular 2](#_Toc161070640)

[J. Kubernetes 2](#_Toc161070641)

Real Estate full project was developed for technical testing and demo with implement some functions as Redfin website. That structure designed including Admin, Client, APIs. The technical will collaborating with Azure as cloud server, Git for versioning, Docker as deploying process management.

1. **Structure of project**
   1. **WebAdmin**

Developing admin site, collecting data from microservices provided

Technical: MVC net 6, Azure Cosmos DB, Connecting to APIs, Azure

* 1. **Microservices/Microservices\_Net5**

Technical: Api net 5, Entity Frameworks code first, SQL, Docker, Azure, Docker Compose

* 1. **Microservices/RealEstateServices**

Technical: Api net 6, Cosmos DB, Azure, MongoDB

* 1. **WebClient**:

Developing client side, integrating with Apis

Technical: Angular, Docker

1. **Technical solution collection**
2. Microservices architecture
3. API authentication with JWT – **Done (Microservices\_Net5)**
   * Validate token online: <https://jwt.io/>
   * Compare token with secret key in similar process to generate and validating token.
4. Entity Framework - **Done (Microservices\_Net5)**
   * Configurating for Entity framework net 5,6 with code first
   * Example code as query database
5. MVC web – Done (Admin Web)
   * MVC web with CRUD
6. Git action CICD
   * Auto deploying to Azure when code change as configuration
   * Auto testing when build, push, deploy
7. Azure
   * Azure app services: create, deploying with Docker, git action, Azure pipelines
   * Azure container, Azure register: to manage and deploy container with Azure
   * Virtual machine (Win Server and Linux)
     + Install, setup docker, port and deploy with Docker, Docker compose
     + Server configuration with inbound, out bound port
8. Docker, Docker compose
   * Deploy application with Docker CLI
     + Note: A Net core API deploying with Docker successful, can testing with Postman, but the home page (Swagger/index page) not show as expected
   * Create docker file, understanding structure of Docker
   * Manage Docker compose file with Network, Volume
   * Example Docker compose with an SQL DB, Admin web, Api
9. Mongo DB
10. Angular
11. Kubernetes