**Questions**

1. What would be the main API endpoints you would create when building this solution?

| **Http Method** | **URL** | **Sample Json** | **Description** |
| --- | --- | --- | --- |
| Post | /login | {  "username":"owner",  "password":"123"  } | Provide username and password  Then api return JWT token when successful login |
| Get | /customer/all | [  {  "id": 2,  "firstName": "firstName1",  "lastName": "lastName",  "contactNumber": "",  "active": true,  "addedBy": 2  },  ] | Please provide a JWT token . then return all customer list |
| Post | /customer/create | {  "firstName":"Amal",  "lastName":"Samantha",  "contactNumber":"9922929"  } | This end point helps to insert new customers to the system . |
| Put | /customer/innactive/1 |  | This end point provide soft delete for customer records |
| Delete | /customer/delete/1 |  | This end point provide permanent delete for customer records |
| Put | /customer/update/3 | {  "firstName":"bandara ",  "lastName":"ddd",  "contactNumber":"888888"  } | This endpoint update existing customer data |
| Get | /user/all | {  "id": 1,  "firstName": "owner",  "lastName": "owner",  "userName": "owner",  "password": "123",  "RoleId": 1  }, | Get all system users |
| Get | /medication/all | [  {  "id": 37,  "name": "test1cccMMM",  "description": "dhhdhdd",  "quantity": 10,  "active": false,  "addedBy": 2  },  ] | Get all medication records |
| Post | /medication/create | {  "name":"test",  "description": "ddddd",  "quantity": 0  } | Create new medication record |
| Put | /medication/innactive/37 |  | Soft delete medication record |
| Post | /medication/update/37 | {  "name":"test1cccMMM",  "description": "test",  "quantity": 10  } | Update detail of medication record |
| Delete | /medication/delete/36 |  | Delete medication record permanently |

2. What are the data models you would create to implement this solution?

1. User

{

firstName: DataTypes.STRING,

lastName: DataTypes.STRING,

userName: DataTypes.STRING,

password: DataTypes.STRING,

RoleId: DataTypes.INTEGER,

},

1. Role

{

role: DataTypes.STRING,

},

1. Medication

{

name: DataTypes.STRING,

description: DataTypes.STRING,

quantity: DataTypes.INTEGER,

active: DataTypes.BOOLEAN,

addedBy: DataTypes.INTEGER,

},

1. Customer

{

firstName: DataTypes.STRING,

lastName: DataTypes.STRING,

contactNumber: DataTypes.STRING,

active: DataTypes.BOOLEAN,

addedBy: DataTypes.INTEGER,

},

3. What are the frameworks / libraries you would use to build this solution?

|  | **Name of the package What would you accomplish using that?** |
| --- | --- |
| ExpressJS | Web Application Framework |
| joi | Validate request Body |
| passport | Generate user authentication |
| passport-jwt | Generate JWT token |
| rbac | All user role based access control |
| sequelize | Mysql orm |
| sequelize-cli | Run sequelize migration and seeders |
| jest | Setup test environment |
| chai | Mock sever |

4. What are the main programs and tools you would use to build this solution?

| **Name of the program** | **What would you accomplish using that?** |
| --- | --- |
| Visual Studio Code | IDE |
|  |  |

5. What is the approach you would use to implement User Permissions based on User

Roles?

In user role based access control I used the RBAC node package . When user login with jwt token we insert user role to request parameter



Then we write a middle where in the route file . inside of middleware we check user role and whether user role have right access to path



