Project Overview:

Contos, Ltd. is into Energy Sector and wants to build a software maintaining the equipments used in plant production. They want to build the software which lists all the equipments installed in the plant and check their maintenance status. Employees should be able to look into the due date of the next maintenance and raise a notification for the same. If the parameters of the equipment like temperature,rpm etc or not upto the mark should raise the Notification for correcting the mal functioning

Functional Requirement

- Technicians should be able register and login to the EMS
- Once User Login he should be able to see all the Equipments that are available in the given plant
- User should be able see the details of the individual equipments like brief description , installed date, next service due date and its parameters like temperature and rpm
- Technicians should be able to filter the equipments by functional location and equipment
- Technicians should be able to sort the equipments by next service due date
- Plant administrators should be able to Add Equipment
- Plant administrators should be able to Edit Equipment
- Plant administrators should change status to Order or Out of Order
- Technicians should be able to see the details of Equipment and see if there is a concern should raise a notification against the given equipment
- Technicians should be able to see the notifications against the given equipment

Security Requirements

- There will be two kinds of Technicians and Plant Administrator
- When we register the users we should have field differentiate the Technicians and Plant Administrator
- The password should be encrypted when sending to the server
- The traffic between the browser and server should be secured and use SSL.
- Application shoul be secured and accessible via https
- Once the user is authenticated he should get a token which user needs to present for sub sequent api request
- The password should not be stored in plain text in the server database

Technical Requirements

- Use Reactjs for the front end development
- Use ASP.Net Core with GraphQL for the Back End Development
- Use Entity Framework core to get and update the data from the database
- Use the SQL Server as the DataStore
- Use Appolo Client for querying the data from the server in ReactJS

- Use Bootstrap for application styles
- Use React-Router for navigating between the page in the application
- Use JWT token for authorization and claims
- In ASP.Net Core use the IOC pattern/ Dependency Injection for all the services queries/mutations
- Use Repository pattern to get the data from the Entity frame work

Scalability and Cloud Requirements

- Application Development should have Perfomance and Scalability in consideration
- Web API should be able to containerized on need basis and can be deployed to azure container instances
- Web API should be able to publish in Azure App Services and should be platform agnostic i.e it should run on Windows/Linux. Should use .net core 3.1 as runtime of the application