# pROJECT REQUIREMENTS

## Basic requirements

|  |  |
| --- | --- |
| Requirement | Implementation |
| React + Typescript web app | Yes |
| GraphQL saving data to relational database with Entity framework | * GraphQL implemented * Using Java Spring Framework (Checked with MSP, confirmed okay) * NOSQL was used instead of an relational database. (This is for an advanced requirement) |
| Apollo client & React hooks | Yes |
| UI Library/Styled components | Materials UI |
| Utilise source control via github | <https://github.com/fsit869/MSA2021_Notebook> |
| Responsive UI | Yes |
| Deployment frontend and backend using Github Actions | Database server: MongoDB Atlas Front end: Back end: |

## advanced requirements implmented

* All components integrated with Storybook
* Unit testing with Jest
* React Router for handling navigation
* Creation and usage of custom branding that is integrated into product
* NOSQL database used instead of relational database
* OAuth2 [Github or Google]

# producT

A website that allows a place to ask questions and for people to answer.

To be able to answer or comment, user must be logged in with OAuth2. If the user Is not logged in, the user is only available to view questions and comments.

# fRONT END:

* Created with Typescript + React
* Styled with Material UI
* GraphQL data fetched using Apollo client
* Webpage routing using React Router
* Components integrated with storybook JS.
* Testing using JEST

## UI Wireframe:

# bACK END

* Programmed using Java using Spring Framework
* Data stored with NOSQL, using MongoDB Atlas
* Security using Github Oauth2
* GraphQL used for querying data

## SECURITY:

Certain queries will be available without OAuth. Such as viewComments. However to add comments, requires the user to login using Oauth2.OAuth2 security is handling using Spring Security. Once the authorisation is complete, Oauth redirects to main page. Which then redirects to the front end. This allows flexibility after authorisation. All other subdomains of the website are locked.

To declutter the workspace of the POJO [Plain old java objects], LOMBOK was used. This library uses annotations to create getters and setters for objects.