# Техническое задание для курса Angular.

В качестве закрепления и подтверждение навыков, полученных в результате прохождения курса по Angular, в рамках проекта GRSE TalentLab, должен быть предоставлен проект, содержащий следующие элементы:

* Проект должен быть выполнен на основе архитектуры SPA (Single Page Application) и использовать технологии фреймворка Angular (17 версии) как свою основу.
* Приложение должно быть написано с использованием языка Typescript
* Приложение должно предоставлять возможность аутентификации. Т.е. должно быть возможно создать юзера и залогиниться этим юзером в проект (админка и другой юзер менеджмент, как и авторизации не обязательны, но будут большим плюсом).
* Приложение должно содержать 2 ключевые «фичи» (feature): работы с рецептами (Recipe Book) и работа с списком для закупок (Shopping list).

1. Recipe Book

В пределах данной фичи должна быть имплементирована возможность работы с рецептами, а именно:

1. Возможность составлять рецепты из комбинации продукт + количество продукта (учет типа продукта для задания его количества – литры/кг будет плюсом)
2. Возможность задавать имя, описание и ссылку на картинку для каждого рецепта (возможность прямой загрузки картинки для отображения с компьютера клиента будет плюсом)
3. Должна иметься возможность редактировать все вышеперечисленные параметры после создания и сохранения рецепта.
4. Должна иметься возможность по одному клику добавить все продукты и их количество, указанные в рецепте в Shopping List (аналогично добавлению в корзину)
5. Должна иметься возможность сохранять созданные рецепты и уметь отображать их в формате удобном для чтения (read-only format).
6. Должен существовать лист (из которого можно выбирать рецепты для дальнейшего редактирования или просмотра), который будет отображать все созданные рецепты в пределах одной сессии (сохранение списка рецептов между сессиями для каждого юзера будет плюсом)
7. Shopping list

В пределах данной фичи должна быть имплементирована возможность работы с набором продуктов для закупки, а именно:

* + - * Возможность добавить новый продукт (наименование и количество) в список закупок
      * Отображение уже добавленных (к примеру, из Recipe Book-а) продуктов и их количества в пределах одной сессии (сохранение списка всех продуктов между сессиями для каждого юзера будет плюсом)
      * Возможность изменения добавленных продуктов и их количества в пределах списка
      * Возможность поиска и фильтрации по продуктам в существующем списке.
* Внутри приложения для навигации между вышеуказанными фичами должен осуществляться корректный роутинг построенный на основе @angular/router. Навигация должна происходить на основе выбора пунктов в меню (topside fixed menu, hamburger menu, или другие аналогичные типы меню). В случае если в приложении предусмотрена админка или юзер менеджмент, то роуты должны быть защищены конкретными гвардами для авторизированного доступа к разным частям приложения.
* Приложение должно использовать возможности библиотеки RxJs (<https://rxjs.dev/>) для корректной работы с запросами. (Даже если запросы будут представлять возврат mock-данных из файлов/localstorage-а)
* Работа с формами, к примеру, формами редактирования, должна быть основана на reactive или template driven формах (при работе с UI фреймворками такими как Angular Material допускается изменение подхода для корректного функционирования UI компонентов)
* Приложение должно быть стилизовано, т.е. должна быть использована либо одна из UI-библиотек (Angular Material, Bootstrap etc) либо ручная стилизация с использованием каскадных таблиц стилей (использование препроцессоров таких как SCSS будут плюсом)
* Плюсом будет если приложение будет «сдеплоено» (deployed), т.е. будет возможность зайти и посмотреть приложение онлайн где-то в интернете.
* Плюсом будет если ваше приложение будет корректно «бандлиризировано» и будет использовать технологии lazy-loading-а модулей.