**Frontend**

Создать приложение на React, содержащее авторизацию пользователя и 3 вкладки (или страницы):

- Роли

- Пользователи

- Пакеты

Вкладка "Роли":

1. Должна включать в себя отображение списка с ролями и разрешениями для них.
2. Пользователь с ролью "admin" должен иметь возможность добавлять/изменять/редактировать запись о любой роли (в соответствии с разрешениями в БД).
3. Пользователь с ролью "user" должен иметь возможность просматривать список ролей без их редактирования.
4. Пользователи с прочими ролями могут действовать в зависимости от разрешений (add\_roles, edit\_roles, delete\_roles) и имеют возможность просматривать список ролей.

Вкладка "Пользователи":

1. Должна включать в себя отображение списка всех пользователей с их ролями.
2. Пользователь с ролью "admin" должен иметь возможность добавлять/изменять/редактировать запись о любом пользователе (в соответствии с разрешениями в БД).
3. Пользователь с ролью "user" должен иметь возможность изменять только свою запись (в соответствии с разрешениями в БД).
4. Пользователи с прочими ролями могут действовать в зависимости от разрешений (add\_users, edit\_users, delete\_users) и имеют возможность просматривать список ролей.
5. Любой пользователь всегда может редактировать свою запись, за исключением изменения роли.

Вкладка "Пакеты":

1. Должна содержать список пакетов и пользователей, которые имеют к ним доступ.
2. Пользователь с ролью "admin" должен видеть список всех пакетов и иметь возможность давать к ней доступ любому пользователю.
3. Пользователь с ролью "user" должен иметь возможность просматривать только список доступных ему пакетов
4. По умолчанию доступны 2 роли "admin" и "user" (которые не могут быть удалены или изменены)

**API (Предпочтительно FastAPI)**

Реализовать API c возможностью авторизации пользователей, которое будет реализовывать функционал для frontend указанный выше через взаимодействие с Базой данных.

**База данных (Предпочтительно PostgreSQL)**

Содержит таблицы:

1. users(user\_id:serial, username:varchar(80), role\_id:integer) - список пользователей (произвольные имена пользователей)
2. roles(role\_id:serial, rolename:varchar(80), add\_roles:boolean, edit\_roles:boolean, delete\_roles:boolean, add\_users:boolean, edit\_users:boolean, delete\_users:boolean) - список ролей (две роли по умолчанию admin, user) и разрешения для них
3. softs(soft\_id:serial, softname:varchar(140)) - список пакетов (произвольные названия)
4. softs\_access(soft\_id:integer, user\_id:integer) - доступ к пакетам, если есть пара пакет-пользователей, то имеется и соответствующий доступ

О разрешениях:

add\_roles - разрешение на добавление ролей

edit\_roles - разрешение на редактирование ролей

delete\_roles - разрешение на удаление ролей

add\_users - разрешение на добавление пользователей

edit\_users - разрешение на редактирование пользователей

delete\_users - разрешение на удаление пользователей

При выполнении задания может быть использовано ORM