



Tasks Front End 5.0 (Eng version)

FE #1 - Init Application

We will create our application on `React` and `Typescript` using Best Practices.

You must use `functional components` and `hooks` during developing.

We use `GitHub` and only `privat repositories` in our work.

Your mentors will do the `merge` themself, you only create `pull requests` and mentors checks it. Mentors should be added as `collaborators` to your repository.

- ☐ Read information about `React` and `Typescript`
- ☐ Create a `React` application using `TypeScript`
- ☐ You should be used standard `Github Workflow` during development. Create branches:
 - `main` ,
 - `develop` ,
 - and a branch for the current task: `FE-1-init-application`
- ☐ Create a file structure that will include `directories` :
 - `Components` ,
 - `Pages` ,
 - `Store` ,
 - `Api` ,
 - `Types` ,
 - `Utils` ,
 - etc.
- ☐ Describe how your application starts in `README.md`
- ☐ Create a `start page` with project name and greeting
- ☐ Set up the `.gitignore` file. There should't be nothing useless in your project

- ☐ Set up the `.env` file. All secret keys and other variables, including `hosts` and `ports`, must be in this file. Set up import of variables from `.env`
- ☐ Create `.env.sample` file. This file stores a list of variables that are used in the project, but with empty fields. This is necessary for other developers for quickly understand what variables are required for your project.

!!! the `.env` file must NOT be on git
`.env.sample` file must be on git

- ☐ * Use `semantic tags` during developing a project
- ☐ Create PR (Pull Request) from `FE-1-init-application` into `develop`

FE #2 - Add Dockerfile

We use `containers` for developing flexible as well and deploying our application. We will use `Docker-compose` for local development and `Dockerfile` for deploying our application

- ☐ Read about `Docker`
- ☐ View examples of launching `React` applications within `Docker`
- ☐ Set up the `Dockerfile` file
- ☐ Set up the `.dockerignore` file. There shouldn't be anything useless get into `Docker`. e.g. directory with your dependencies
- ☐ Launch your application within `Docker`
- ☐ Describe in `README.md` how your application launches within `Docker`
- ☐ * Add a volumes to `Docker` to rebuild `image` and restart the `container` when changes are made in the project
- ☐ Create PR (Pull Request)

FE #3 - Add common pages

Should be added a `list of pages` with mock data.

- ☐ `About` page
- ☐ `User registration` page
- ☐ `User authorization` page
- ☐ `List of users` page
- ☐ `User Profile` page
- ☐ `List of companies` page
- ☐ `Company Profile` page

- ☐ Create the **universal modal window** that can be used anywhere in our application
- ☐ Set up **routing**
- ☐ * **Maximum reuse of components** , Creation of templates
- ☐ Create PR (Pull Request)

FE #3.1 - ADD GH-PAGES

You need to add GH-PAGES and deploy your application - **Deprecated**

FE #4 - Add Redux

For **state management** we will use **Redux** or **Redux-Toolkit**

- ☐ Read about **Redux** and **Redux-Toolkit** and how working with these libraries
- ☐ Install **Redux** or **Redux-Toolkit** and implement it to your application.
- ☐ Create the **state** that will store the test string value
- ☐ Create a simple **reducer** or **slice** , **action creator** (for changing a test string on **Redux** or **Redux-Toolkit**)
- ☐ Use **useDispatch** and **useSelector** to work with data from **Redux** or **Redux-Toolkit**
- ☐ Show how **state management** works: changing test string from **state**
- ☐ Create PR (Pull Request)

FE #5 - Add Axios

We will use **Axios** to send requests and receive responses.

- ☐ Read about **Axios**
- ☐ Set up **Axios** and create **instance**
- ☐ Create an **endpoint** that will check the **Health Check** of the Backend
- ☐ Create PR (Pull Request)

FE #6 - Authorization

Необходимо реализовать **авторизацию** в нашем проекте при помощи **логина** **пароля** и с помощью **Auth0** . Наше приложение должно иметь возможность регистрации/авторизации с двух источников - **логин пароль** и **Auth0** .

- ☐ Реализовать функционал на **странице регистрации** . Должна присутствовать **валидация** поля **email** и поля **подтверждение пароля** . **Email** не должен валидироваться только по знаку @
- ☐ Ознакомиться с **регулярными выражениями** . Применить их для **валидации**

- ☐ Реализовать функционал на **странице авторизации**. **Валидация** email обязательна
- ☐ Зарегистрироваться в **Auth0** и создать свое **API** для авторизации-
~~<https://manage.auth0.com/dashboard> - Applications - APIs~~
- ☐ Создать свое **Application** в **Auth0** -
~~<https://manage.auth0.com/dashboard> - Applications - Applications~~
Создать **single-page-application**
В настройках **Allowed Callback URLs** - **Allowed Logout URLs** и **Allowed Web-Origins** добавить эти url:
~~<http://localhost:3000>, <https://romanxeo.github.io>,
<https://romanxeo.github.io/internship-token>~~
- ☐ Установить и подключить библиотеку **Auth0** для всего приложения
- ☐ На **странице авторизации** и регистрации добавить кнопку авторизации при помощи **Auth0**
- ☐ Добавить функционал получения **токена** и хранения его в **localStorage**
- ☐ В случае успешной **авторизации** В шапке приложения отображать **email** и **имя** пользователя. Информация о текущем пользователе хранить в **Redux**
- ☐ Настроить **роутинг** с учетом **приватности**. Наше приложение должно иметь список страниц доступных;
 - **только авторизованным** пользователям:
 - список пользователей,
 - профайл пользователя,
 - список компаний,
 - профайл компании;
 - **только не авторизованным** пользователям:
 - авторизация,
 - регистрация;
 - **общие** страницы:
 - страница приветствия,
 - страница о проекте.
- ☐ Создать PR (Pull Request)

FE #7 - Add User CRUD

Реализовать **CRUD** для работы с **User**.

- ☐ Реализовать функционал **получения всех пользователей** при помощи **Axios**
- ☐ Записывать информацию о списке пользователей в **Redux**

- ☐ Реализовать функционал получения информации о пользователе по его `id` и так же хранить информацию в `Redux`
- ☐ Реализовать функционал `редактирования` информации о пользователе (дополнительная информация, пароль, аватар)
- ☐ Реализовать функционал `удаления` пользователя по его `id`
- ☐ Добавить обработку ошибок
- ☐ * для получения всех пользователей реализовать компонент `пагинации` . Сделать его универсальным для любых сущностей
- ☐ Создать PR (Pull Request)

FE #8 - Add Validation

Настроить `Валидацию` в проекте и в дальнейшем придерживаться её.

- ☐ Пользователь `должен` иметь возможность `редактировать` только своё `имя` и свой `пароль`
- ☐ Пользователь `не должен` иметь возможности `редактировать` свой `email`
- ☐ Пользователь `должен` иметь возможность `удалять` только `свой профиль`
- ☐ Создать PR (Pull Request)

FE #9 - Add Company CRUD

Реализовать `CRUD` для работы с Компаниями `Company` . Придерживайтесь Валидаций и Разрешений.

- ☐ Реализовать возможность `создания компании` при помощи заполнения формы в `модальном окне` . Обязательная информация к заполнению:
 - названии компании
 - описании компании
 - ~~режим-видимости компании~~
- ☐ На странице компании `Владелец` должен иметь возможность `редактировать` основную информацию о компании и ~~режим-ее-видимости~~, а так же иметь возможность `удаления` компании
- ☐ `Владелец` должен иметь возможность `удалять` свою компанию
- ☐ Так же реализовать функционал для `вывода списка компаний` и `вывод компании по id`

FE #10 - Add Company Actions

Реализовать взаимодействия Компаний `Company` с Пользователя `User` . Придерживайтесь Валидаций и Разрешений.

- ☐ `Владелец` должен иметь возможность `отправить приглашение` в свою компанию неограниченному количеству других `пользователей`

- ☐ Владелец должен иметь возможность отменить свое приглашение
- ☐ Пользователь должен иметь возможность принять приглашение в Компанию - после чего последует автоматическое вступление пользователя в участники группы
- ☐ Пользователь должен иметь возможность отклонить приглашение в Компанию
- ☐ Пользователь должен иметь возможность отправить запрос на вступление в компанию. Список компаний неограничен
- ☐ Пользователь должен иметь возможность отменить свой отправленный запрос на вступление в компанию
- ☐ Владелец должен иметь возможность принять запрос на вступление в компанию
- ☐ Владелец должен иметь возможность отклонить запрос на вступление в компанию
- ☐ Владелец должен иметь возможность исключать пользователей из компании
- ☐ Пользователь должен иметь возможность выйти из компании
- ☐ Реализовать функционал с помощью которого каждый User должен иметь возможность посмотреть список своих запросов в компании
- ☐ Реализовать функционал с помощью которого каждый User должен иметь возможность посмотреть список приглашений его в компании
- ☐ Реализовать функционал с помощью которого Владелец компании может увидеть список приглашенных пользователей
- ☐ Реализовать функционал с помощью которого Владелец компании может увидеть список запросов на вступление в компанию
- ☐ Реализовать функционал с помощью которого можно увидеть список пользователей в компании
- ☐ * все действия связанные с
 - отклонением и удалением заявок,
 - исключением пользователя,
 - назначением администратора,
 - снятием роли администратора,
 - выходом из компании,
 - удалением компании;должны происходить через подтверждающее модальное окно
- ☐ Создать PR (Pull Request)

FE #11 - Add Admin Role

Добавить роль Администратора **Admin** в Компании. Реализовать следующие возможности. Придерживайтесь Валидаций и Разрешений.

- ☐ **Владелец** должен иметь возможность **назначать** и **убирать администраторов** в своей компании из списка участников своей компании
- ☐ Реализовать функционал с помощью которого можно увидеть **список администраторов** в компании

FE #12 - Add Quizzes

Реализовать Викторины (Квизы) **Quizzes**. Придерживайтесь Валидаций и Разрешений.

- ☐ **Владелец** и **Администраторы** должны иметь возможность создавать неограниченное количество **Квизов** в своих компаниях
- ☐ Для создания **Квизов** необходимо использовать **модальное окно**
- ☐ **Квиз** должен содержать в себе:
 - Название
 - Описание
 - Число, указывающее частоту прохождения квиза в днях
 - Список вопросов:
 - Каждый элемент должен включать в себе:
 - вопрос
 - варианты ответа
 - правильный ответ (только один)
- ☐ **Квиз** должен иметь возможность включать в себя неограниченное количество вопросов и неограниченное количество ответов на каждый вопрос
- ☐ **Квиз** не должен иметь в себе менее двух вопросов и менее двух вариантов ответа на каждый вопрос
- ☐ **Владелец** и **Администраторы** должны иметь возможность редактировать и удалять **Квизы** в своих компаниях
- ☐ На **странице компании** должен выводиться список всех доступных **Квизов** (доступно только для участников компании)
- ☐ ** Добавить возможность выбора нескольких ответов
- ☐ Создать PR (Pull Request)

FE #13 - Quiz workflow

Пользователь должен иметь возможность проходить **квизы**. Для этого необходимо перенаправлять его на **отдельную страницу** со списком всех вопросов и вариантов ответов (у каждого **квиза** свой роут по id)

- ☐ Реализовать возможность прохождения **квизов**
- ☐ После прохождения **квиза**, данные необходимо отправить на сервер. Должен приходить ответ о **результате** прохождения данного **квиза**.
- ☐ после прохождения **квиза** **пользователь** должен возвращаться на **страницу компании**
- ☐ Создать PR (Pull Request)

FE #14 - Add Analytics

Получение **Аналитики**. Для построения графиков будем использовать библиотеку **Chart.JS**

- ☐ Ознакомиться с **Chart.JS**. Установить библиотеку

Владелец и **Администраторы** компаний на странице компании должны иметь кнопку Аналитика, после нажатия на которую выводится следующая статистика:

- ☐ График **Средних баллов** всех пользователей с динамикой во времени
- ☐ Должна быть возможность выбрать определенного **пользователя** из своей компании и отобразить график **Средних баллов по всем квизам** выбранного пользователя с динамикой во времени
- ☐ В списке пользователей компании Владелец и Администратор должны видеть время последнего прохождения теста у каждого из пользователей.
- ☐ Любой **пользователь**, на странице профайла, должен видеть информацию о среднем бале по всем **квизам** компаний в целом (совокупный рейтинг - звездочки)
- ☐ **Пользователь**, только на своем профайле, должен видеть график **Средних баллов** по каждому из квизов, с динамикой во времени
- ☐ **Пользователь**, только на своем профайле, должен видеть список **квизов и времени его последнего прохождения**
- ☐ Создать PR (Pull Request)

FE #15 - User Profile Page

Реализовать страницу **Пользователя**. Придерживайтесь валидаций и разрешений. Страница **пользователя** должна содержать:

- ☐ **Основную информации** (имя, о себе и email)
- ☐ **Список компаний** в которых он состоит

- ☐ **Значение совокупного рейтинга** - отображать при помощи графических элементов (звездочки, например)
- ☐ Пользователь должен иметь возможность **редактировать информацию о себе** на странице профайла
- ☐ Пользователь должен иметь возможность **покинуть компанию**
- ☐ * удаление профайла и выход из компании должно быть реализовано через **подтверждающее модальное окно**
- ☐ Создать PR (Pull Request)

FE #16 - Add export data

Экспортирование данных . Пользователь должен иметь возможность экспорта в файл формата **CSV**

- ☐ **Пользователь** на странице своего профайла должен иметь возможность выгружать файл со **своими** результатами/ответами на **квизы** , которые он проходил
- ☐ **Владелец** и **Администраторы** компании на странице своей компании должны иметь возможность выгрузить результаты/ответы **одного** определенного пользователя или всех пользователей сразу из своей компании.
- ☐ * **Владелец** и **Администраторы** компании на странице своей компании должны иметь возможность выгрузить результаты/ответы по одному определенному **квизу**
- ☐ Создать PR (Pull Request)

FE #17 - Other Actions

Различные действия

- ☐ **Пользователь** должен видеть список заявок на вступление в компанию.
- ☐ **Пользователь** должен иметь возможность отклонить полученную заявку
- ☐ **Пользователь** должен видеть список заявок, поданные в различные компании. **Пользователь** должен иметь возможность отменить поданную заявку
- ☐ Создать PR (Pull Request)

FE #18 - Add Notifications

Необходимо реализовать **систему уведомлений** . Придерживайтесь Валидаций и Разрешений.

- ☐ Каждый пользователь должен видеть список системных уведомлений, которые ему приходят

- ☐ Пользователь должен иметь возможность пометить уведомление как прочитанное
- ☐ Создать PR (Pull Request)

FE #19 - Tests

Весь код в нашем проекте должен быть покрыт **тестами**

- ☐ Прочитать про **тестирование** в React
- ☐ Ознакомиться с **JEST**, **React Testing Library**
- ☐ Протестировать **компоненты** на отзывчивость по нажатию различных кнопок в зависимости от страниц

Покроем тестами наш **Redux**

- ☐ Ознакомиться с **Mock** данных
- ☐ Протестировать логику **Redux**
- ☐ Создать PR (Pull Request)

FE #20 - Add Github Actions

Нам необходимо **автоматизировать деплой** новых версий нашего приложения. Для этого нам необходимо создать свой **github action** и написать **workflow** скрипт для **депоя** приложения

- ☐ Прочитать что такое **github actions**
- ☐ Разобрать синтаксис для **workflow** файлов
- ☐ Реализовать деплой на **AWS** с помощью **github actions**
Можно взять реализацию:
 - AppRunner
 - ECS
 - EC2
- ☐ Создать PR (Pull Request)

FE #21 - Add Import Data

** Необходимо добавить возможность импортирования новых **КВИЗОВ** с помощью **Excel**, которые обновят все вопросы и ответы. Должна быть возможность, как добавлять, так и создавать новые **КВИЗЫ**

FE #22

FE #23

<http://3.75.186.163/docs#/>