Проект. Платформа-визитница VISITKI

Стас Звягинцев Jan 26
тут наверное проще будет целиком перезалить, чем отмечать изменения

Add a comment...

На софт-скиловом треке студенты курса «Веб-разработчик плюс» практикуются в самопрезентации. Мы хотим создать внутренний сервис, который сделает эту практику удобной и интерактивной: студенты будут публиковать свои визитки, давать обратную связь и изучать визитки друг друга.

В проектном месяце вы сверстаете фронтенд и бэкенд прототипа, который поможет команде разработать этот сервис.

Фронтенд

Рекомендуемый стек:

- React
- SCSS
- библиотеки React-Router, XLSX

Использование Redux в данном проекте избыточно.

Для инициализации проекта воспользуйтесь Create React App, опционально можете использовать шаблон typescript.

Авторизацию вам предстоит реализовать на клиенте через Яндекс ID (OAuth), а для работы с картами и подсказками городов в форме воспользуйтесь API Яндекс Карт (`SuggestView` , `geocode`).

Авторизация

Для авторизации пользователя вам нужно перенаправить его согласно документации на форму входа в Yandex. Обратно пользователь вернётся на указанный при регистрации приложения callback-адрес с переданным в query-параметрах кодом подтверждения.

В данном проекте мы реализуем двухшаговую авторизацию: первый шаг — это получение кода подтверждения на клиенте, второй шаг — получение токена доступа на сервере и возврат JWT-токена для дальнейшей работы с нашим API.

Сейчас на фронтенде мы сделаем только первый шаг и оставим для разработчиков бекенда точки интеграции в виде двух функций-заглушек:

1. `getToken(code)` — получает код подтверждения, возвращает произвольную строчку в качестве JWT-токена.
2. `getUser(token)` — получает JWT-токен (из первой функции) и возвращает данные об авторизованном сейчас пользователе в виде объекта, содержащего `_id`, `name`, `email`, `cohort`, `photo`, `role`. Поле `role` может принимать значения “student” или “curator”.

Полученный токен нужно сохранить в `localStorage` браузера и восстанавливать его при загрузке страницы. При выполнении запросов к API бекенда он должен быть передан в заголовке `Authorization`.

Тестирование

На этом этапе проекта вы реализуете фронтенд-часть. У вас есть только заглушка бэкенда, которая не умеет выдавать токены и определять роль авторизованного пользователя. Поэтому для тестирования вам потребуется предусмотреть способ установки сведений о пользователе локально в браузере. Фактически вам нужно возвращать из функции `getUser` три различных набора сведений:

- для студента №1;
- для студента №2;
- для куратора.

Данные о студентах можно взять из заглушки, а для куратора возвращается только email, так как у него нет профиля. Роль проставьте соответственно.

Вам нужно самостоятельно выбрать способ смены роли в приложении, но обязательно задокументировать в README, какие действия нужно произвести для смены пользователя, чтобы ревьювер смог посмотреть всю функциональность.

В качестве идеи для переключения ролей предлагаем создать дополнительную временную страницу `/test` с кнопками, по нажатию на которые подменяются данные пользователя. Также будет удобно, если будет кнопка очистки `localStorage` для выхода пользователя из приложения.

Основная функциональность

1. Заполнение собственной визитки.
2. Просмотр всех визиток когорты.
3. Просмотр детальной версии визитки.
4. Комментирование (или реакция эмоджи) визитки целиком или отдельных её блоков.
5. Просмотр комментариев (реакций) на свою визитку (недоступно другим студентам).
6. Пост-модерация реакций (для кураторов).
7. Управление составом когорт (для кураторов)

Макеты и описания страниц

Макеты всех страниц (версии для ПК, планшета и мобильного телефона) можно изучить по ссылке.

Схема и заглушка для API

Для работы с бекендом в package.json добавьте параметр proxy с корневым адресом нашего бекенда: `https://visitki.practicum-team.ru/api`.

Для работы с бекендом в package.json добавьте параметр proxy с корневым адресом нашего бекенда: `https://visitki.practicum-team.ru/api`.

OpenAPI файл описания бекенда доступен по ссылке, его можно импортировать в Postman, чтобы создать преднастроенную коллекцию. После импорта файла вы сразу сможете тестировать запросы.

Описание страниц

Неавторизованный пользователь видит только страницу авторизации.

После авторизации через Яндекс ID:

- Студент попадает на главную страницу с визитками студентов своей когорты. Роут — `/`.
- Куратора переадресует на главную страницу админки для модерирования комментариев. Роут — `/admin/`.

Авторизованный пользователь на всех страницах видит:

- стандартный хедер с лого и профилем авторизованного пользователя;
- стандартный футер.

Главная страница

На главной странице студент видит краткие версии визиток всей когорты. Все фотографии отображаются в черно-белом режиме.

1. С главной страницы студент может перейти:
   - на страницу выбранной визитки с детальной информацией;
   - на карту, где отмечено расположение других студентов.
2. На главной странице есть выпадающее меню, которое позволяет отфильтровать студентов по выбранному городу.
3. Если навести мышь на визитку, в правом верхнем углу появится значок комментирования, по нажатию на который появляется плашка с полем для ввода

4. Для владельца визитки и куратора поверх значка комментирования отображается количество реакций на визитку, а нажатие на значок открывает плашку с самими комментариями.

Информацию о том, визитки какой когорты показывать на главной страницы, нужно брать из данных сессии авторизованного студента.

Для куратора когорта не задана, поэтому его нужно переадресовывать на главную админки (модерация комментариев). Но куратор может заходить на страницу любой когорты по адресу `/cohort/{name}` по ссылкам из админки. Для студента страницы когорт, кроме собственной, недоступны.

Страница с детальной информацией

У каждого пользователя в своем профиле можно выбрать, в каком стиле будет отображаться его визитка: «серьёзный», «дерзкий», «романтичный». В зависимости от этого выбора, вёрстка страницы с детальной информацией будет немного отличаться (разницу можно увидеть в макетах). Для изменения темы оформления просто добавляйте соответствующий CSS-класс на страницу. Рекомедуем стилизацию тем построить на css-переменных, так будет удобнее и универсальнее.

1. На странице с детальной информацией отображается
    a. имя;
    b. фото;
    c. девиз/цитата;
    d. город;
    e. контакты (Телеграм, Github опционально);
    f. блок «увлечения, досуг, интересы»;
    g. блок «семья, статус, домашние животные»;
    h. блок «сфера работы»;
    i. блок «учёба»;

    В блоках про увлечения и семью могут быть загружены дополнительные фото.

В блоках про увлечения и семью могут быть загружены дополнительные фото.

2. На странице с детальной информацией возможности комментирования или реагирования аналогичны тому, как реализована работа с обратной связью к визитке целиком на главной странице (пункты 3 и 4 описания главной страницы), только в применении к каждому тематическому блоку, фотографии профиля и цитате.

Страница редактирования профиля

На странице профиля студент может заполнить или изменить информацию о себе. Набор полей и обязательность заполнения смотрите в макете.

Страница карты

На странице карты используется встроенная Яндекс Карта с плашками, на которых видно превью фото, имя и населенный пункт студента.

Блок профиля авторизованного пользователя

Отображается в хедере на каждой странице, содержит аватар и имя пользователя.

- клик по профилю студента в хедере ведёт на страницу с детальной информацией;
- для студента при наведении на профиль видна выпадающая панель со ссылкой на страницу редактирования профиля;
- куратор на месте профиля видит серую заглушку аватарки и текст «админка» вместо имени, клик по профилю ведёт на главную админки.

Блок реакций

Доступен на главной странице (странице когорты) и странице конкретного студента. Максимальная ширина как в макете, высота изменяется в зависимости от количества комментариев, но ограничена 400px. Если комментарии не влезают в блок, то появляется скролл. Максимальная длина отдельного комментария 200 символов.

Нажатие на неактивную эмоцию создает новую реакцию через API, увеличивает счетчик соответствующей эмоции и выделяет ее. Проставленные ранее активным пользователем

Нажатие на неактивную эмоцию создает новую реакцию через API, увеличивает счетчик соответствующей эмоции и выделяет ее. Проставленные ранее активным пользователем эмоции должны быть выделены, повторное нажатие на выделенную эмоцию снимает её путём отправки аналогичного запроса на бекенд.

Административная страница: модерирование комментариев

Главная страница админки для кураторов по адресу `/admin`, на которой они могут:

- видеть список всех комментариев в порядке от самых свежих к самым старым,
- фильтровать их по имени отправителя, получателя и номеру когорты,
- удалять те, что нарушают правила.

Административная страница: управление когортами

Дополнительная страница админки для кураторов по адресу `/admin/users`, на которой отображается список всех студентов в порядке времени изменения от нового к старому.

Можно фильтровать список и изменять на месте отдельные профили (только email и номер когорты, который заполняется как произвольная строка). Если студент уже авторизовывался и заполнял свой профиль, отображается имя, по нажатию на которое куратор может перейти на страницу детальной информации студента. Студент может быть «мягко» удалён (читайте — перемещён) в когорту `deleted`, по которой можно будет найти всех удалённых студентов.

Данные когорт могут быть загружены массово в виде CSV или XLSX файлов, которые вам нужно разобрать прямо в браузере и через API создать или изменить в базе соответствующие профили. Идентифицировать профили можно по уникальному email студента. Файл когорты содержит таблицу с двумя колонками email и номер когорты.

Разбор файла можно реализовать с помощью библиотеки XLSX. Файл обязательно должен быть выбран пользователем через поле загрузки файлов, и уже полученный объект вы сможете прочитать, используя API FileReader.

После выбора файла и разбора данных сведения о студентах предварительно отображаются в верхней части страницы с выделением цветом. А справа появляются кнопки для подтверждения сохранения и отмены загрузки. При обновлении страницы данные также сбрасываются.

До подтверждения сохранения или отмены куратор может редактировать и удалять отдельные строчки загруженного списка студентов прямо на странице, без обращения к серверу.

После подтверждения сохранения данные отправляются построчно на сервер и после выполнения всех запросов обновляется таблица. Если отдельных студентов не удалось сохранить из-за возникшей ошибки, их нужно оставить в верхней части таблицы. Оставьте выделение в таблице, кнопки подтверждения и отмены при этом тоже остаются видимыми.

Что нужно сделать?

Ваша задача — сверстать все страницы для ПК, планшета и мобильного телефона:

1. Страница авторизации
2. Главная страница (страница когорты) с вариантами всплывающих плашек для владельца/куратора и других студентов. Страница доступа по адресу `/` для студента и `/cohort/{name}` для куратора.
3. Страница детальной информации: три версии ("серьезный"/"дерзкий"/"романтичный") с вариантами поведения для владельца/куратора и другого студента.
4. Страница заполнения профиля
5. Страница с картой
6. Страница модерирования комментариев
7. Страница управления когортами

Реализовать необходимую функциональность:

1. Запрашивать состояния авторизации с бекенда.
2. Разделение ролей студента и куратора. Для студента необходимо проверять принадлежность к когорте и владение визиткой при помощи данных о студенте из токена после авторизации.
3. Отправлять данные профиля на бэкенд.
4. Получать комментарии и реакции с бэкенда для авторизованного пользователя.
5. Для куратора получать с бекенда список профилей и комментариев, загружать и разбирать файл когорты.

Когда первая часть работы будет готова, отправьте её на ревью, вставив ссылку на мёрдж-реквест в форму ниже.

На выполнение первой части вам даётся месяц. Постарайтесь выполнить основной объем за 2 недели, на доработки заложите ещё неделю, а последнюю неделю отведите на ревью.