

Сделайте `npm install` в директории frontend, чтобы выкачать все модули. Запуск приложения frontend надо осуществлять так же, как и в предыдущем домашнем задании.

В директории backend настройте базу данных в backend/profile.properties. Примените скрипт data.sql для добавления данных в БД (у обоих пользователей пароль 12345). Запустите backend, убедитесь в работоспособности приложения: можно зайти, выйти, отображаются посты в сайдбаре.

1.
Сделайте страницу со списком всех пользователей в системе. Для этого реализуйте endpoint (в бэкенде) GET /api/1/users. Можно воспользоваться кодом из предыдущего ДЗ для реализации на стороне frontend.
2.
Отобразить на главной (Index) в основной части все посты (желательно переиспользовать код предыдущего дз). Выводите посты в типичном виде (как article, используйте разметку из предыдущих заданий).
3.
Перенесите во frontend форму регистрации из предыдущего задания и сделайте, чтобы она работала через REST. Надо добавить endpoint: POST /api/1/users. Допустимо, чтобы ошибка валидации от бэкенда прилетала в одно место формы (а не по полям), то есть как и при аутентификации. После успешной регистрации надо автоматически аутентифицировать и перекидывать на Index.
4.
Поддержите через REST форму создания поста (endpoint: POST /api/1/posts). После успешного создания поста иницируйте перезагрузку (обновление) поля данных posts в App. Это приведет к тому, что новый пост появится в сайдбаре и на главной.
5. Добавьте в базу данных и модель поддержку комментариев. Реализуйте отдельную страницу для поста (прямо как в предыдущем дз, можете переиспользовать код), она должна отображать один пост и его комментарии (с автором), а также возможность добавлять комментарии. Переходить на эту страницу надо по ссылке View All в сайдбаре на соответствующем посте и по клику в заголовок поста с Index. В постах на главной поддержать вывод правильного количества комментариев.