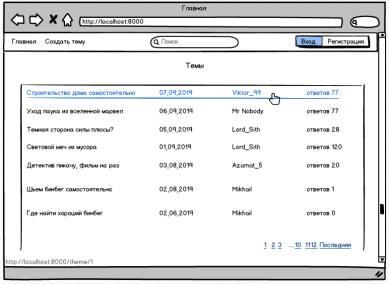
| Создайте git-репозиторий с именем **exam-09**. Делайте  регулярно коммиты и правильно их оформляйте. Регулярно делайте push в удаленный репозиторий для сохранения результатов вашей работы. |
| --- |

**Контрольная работа: Форум.**

В этой контрольной работе вам нужно будет реализовать простой форум с темами и ответами на них. В вашем проекте должна быть реализована регистрация и авторизация пользователей, добавление новых тем, добавление ответов на темы. Добавьте демо данные в ваше приложение. Можете использовать любую базу данных. Фронтэнд часть должна быть аккуратной и функциональной.

**Этап 1 - Создание темы**

После аутентификации пользователя, у него есть возможность создать новую тему на форуме. При создании темы он определяет ее название и содержимое. На главной странице нужно отобразить список тем, отсортированных по дате создания (вверху новые). В списке отображать название темы, дату создания, автора и количество ответов в теме. Реализуйте постраничный вывод сообщений в теме. Число сообщений на одну страницу определите самостоятельно. Внизу страницы должны быть кнопки по переходу на предыдущую и на следующую страницу.



Примечания:

* Вся строка информации о теме является ссылкой на детали этой темы
* Создавать тему может только аутентифицированный пользователь
* Для аутентифицированного пользователя кнопка Вход и Регистрация не должна отображаться - Выводится кнопка Выход и кнопка ведущая в Личный кабинет

**Этап 2 - Ответ в теме**

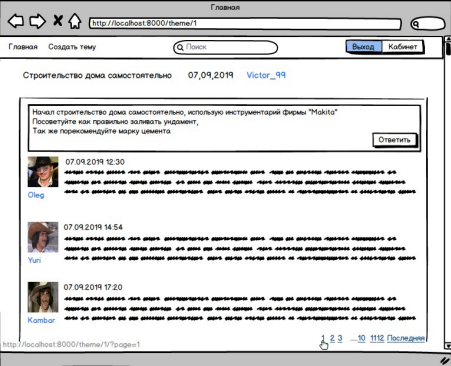
Авторизованные пользователи должны иметь возможность писать сообщения в теме. Данная функция должна быть реализована с помощью асинхронных javascript запросов (fetch, axios, или что-то другое). То есть, пользователь, просматривая тему, видит:

1. Список ответов в теме. Ответы отсортированы по дате в порядке от самого первого. Каждое сообщение отображает своего автора, содержимое и дату создания сообщения.
2. Кнопка "Ответить" для создания ответа:

a) По нажатию на кнопку происходит прокручивание страницы вниз до формы отправки ответа. Пользователь заполняет ее текстом, нажимает кнопку “отправить” и сообщение добавляется внизу списка сообщений. С помощью Javascript вы можете вызвать функцию, которая прокрутит полосу прокрутки вниз и вы увидите это сообщение.

b) По нажатию на кнопку происходит открытие модального окна для ответа на тему (В модальном окне будет форма с текстовым полем и кнопкой отправки)

После отправки сообщения пользователя, оно должно отобразиться в списке ответов. Пример страницы темы:



**Этап 3 - Постраничный вывод ответов**

Реализуйте постраничный вывод сообщений в теме. Число сообщений на одну страницу определите самостоятельно. Внизу страницы должны быть кнопки по переходу на предыдущую и на следующую страницу. Заголовок темы при этом должен всегда отображаться вверху на любой странице.

**Бонусы**

* С помощью асинхронных запросов javascript сделайте функцию голосования за сообщение. Каждый пользователь может поставить лайк или дизлайк на любое сообщение (кроме своего). При этом должно меняться отображение количества лайков и дизлайков у сообщения. Проголосовать 1 пользователь может только 1 раз за одно сообщение.
* При регистрации сделайте возможность пользователю загружать аватар. Аватар должен отображаться рядом с каждым сообщением пользователя в темах
* Введите счетчик сообщений пользователя на форуме. Отображайте количество сообщений пользователя рядом с каждым его сообщением в темах