

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная Работа №1

“Вёрстка сайта средствами HTML, CSS и Bootstrap”

Выполнил:
Стукалов Артем

Группа
К33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Выполнить вёрстку сайта средствами HTML, CSS и Bootstrap. Продумать и реализовать моменты, в которых необходим JS. Тема сайта: социальная сеть.

Ход работы

Выполнение работы будет продемонстрировано на примере сайта web.vk.me. А именно на тех элементах приложения, в разработке которых я принимал непосредственное участие(разработка с нуля/рефакторинг)

Рассмотрена будет верстка списка истории сообщений, так как там применяется достаточно большое кол-во flex-box а также есть достаточно простой, но интересный трюк с использованием “position: sticky” элементов. В качестве части с использованием JS будет рассмотрен один из вариантов реализации системы модальных окон.

Верстка

Упрощенная схема верстки приведена ниже. Во всех дальнейших объяснениях можно ориентироваться на нее.

Начнем с элементов wrapper и content. Wrapper - главный истории сообщений, который непосредственно и скроллится. Content - обертка над всем внутренним контентом, которая растягивается под всю его высоту. Нужна для оптимизации рендеринга списка на этапе композиции за счет вынесения всего контента на отдельный слой.

Далее внутри списка сообщения группируются сначала по дате, а потом по отправителю. Каждый из контейнеров content, dateGroup и userStack имеют свойство “display: flex” для выравнивания элементов внутри них. Отступы между отдельными сообщениями, элементами userStack и dateGroup задаются с использованием “padding” из-за особенностей реализации виртуального списка и необходимости отрисовывать честные размеры сколбара.

Элементы userStack и dateGroup обладают свойством “position: relative”, а элементы date и A(avatar) имеют “position: sticky; top:0” для реализации красивого прилипания к верху родительского блока.

content

wrapper

dateGroup

date

UserStack

message

message

4

dateGroup

date

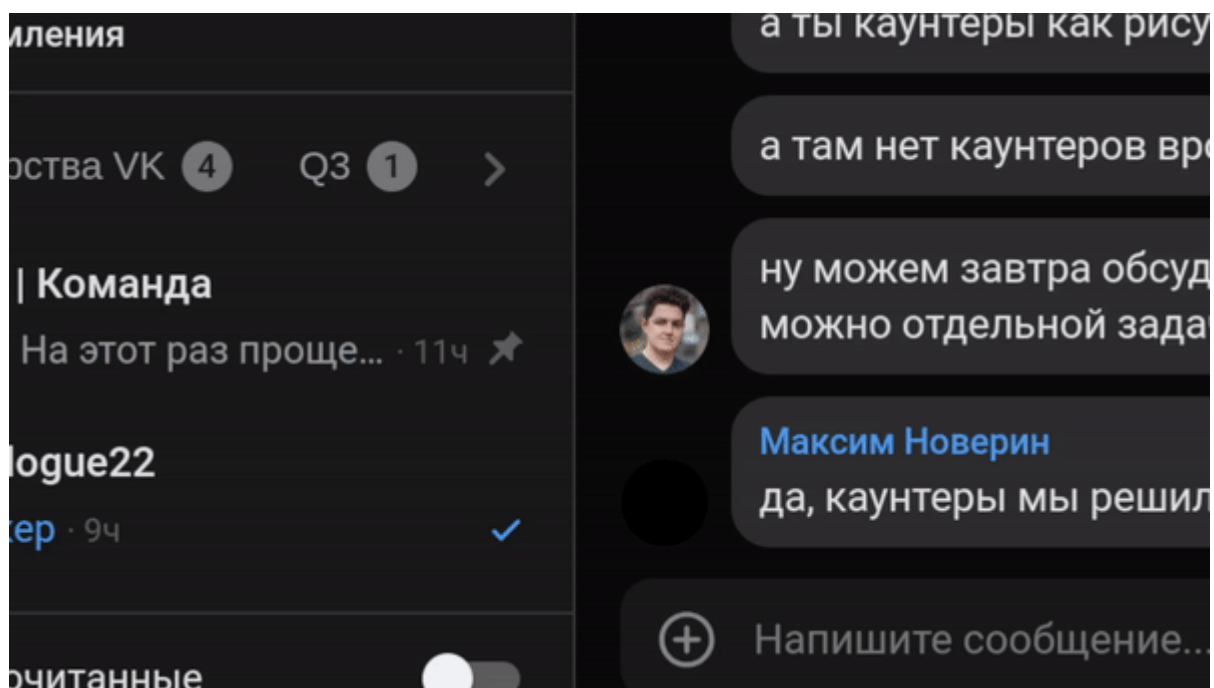
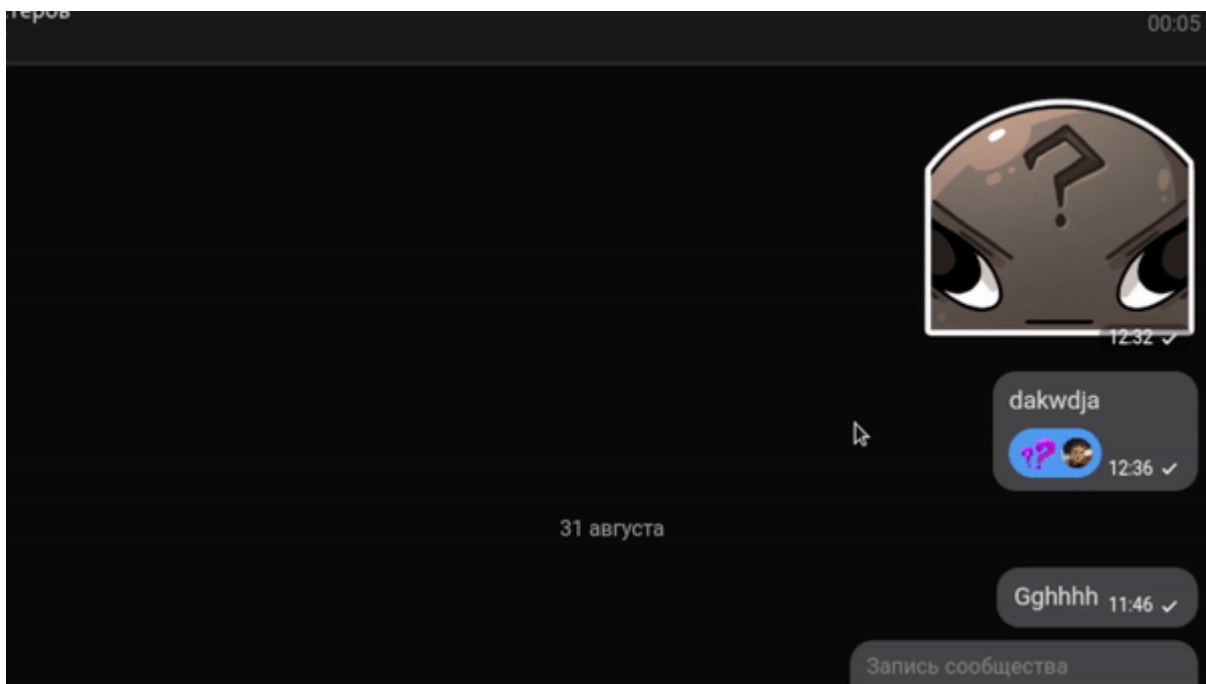
UserStack

message

message

message

4



Модальные окна

Главное проблемой при реализации модальных окон является наложение контента из-за неверно выставленных z-index элементов. Можно конечно очень тщательно выбирать эти значения, но при интеграции приложения на другой сайт, никто не гарантирует работоспособность такого решения. Также существует проблема относительного позиционирования модального окна в случае вставки в DOM дерево на уровне использования. Именно поэтому гораздо проще

устранить эти проблемы еще на этапе верстки. Будем вставлять слой для модальных окон в конец тега body. Таким образом мы решим проблему с позиционированием и z-index. Такой слой для модальных будет иметь “position: relative”, что позволит растягивать модальные окна по всей ширине экрана за счет использования “position: fixed”.

```
▼ <body lang="ru" dir="ltr" class="vkui--sizeX-none vkui--vkBase--dark" scheme="space_gray">
  <div id="captcha-root"></div>
  ▶ <div id="reforged-root" class="vkui__root">...</div>
.. ▶ <div id="reforged-modal-root" class="MEConfig MEAppConfig vkui__portal-root LayerModal">...</div> == $0
```

Само приложение написано на реакт, так что такой вариант реализации возможен с использованием “React.Portal”. Таким образом мы сможем использовать модальные окна на любом уровне дерева компонентов, а также упростим взаимодействие до простого вызова компонента, инкапсулирует всю сложную логику внутри.

```
1 const errorModal = useErrorModal()
2
3 return (
4   <ErrorModal errorText={errorModal.errorText} onClose={errorModal.close} />
5 )
```

Вывод

В процессе выполнения работы освоил базовые навыки реализации сложной верстки списков и построения системы модальных окон, позволяющей избежать базовые проблемы.