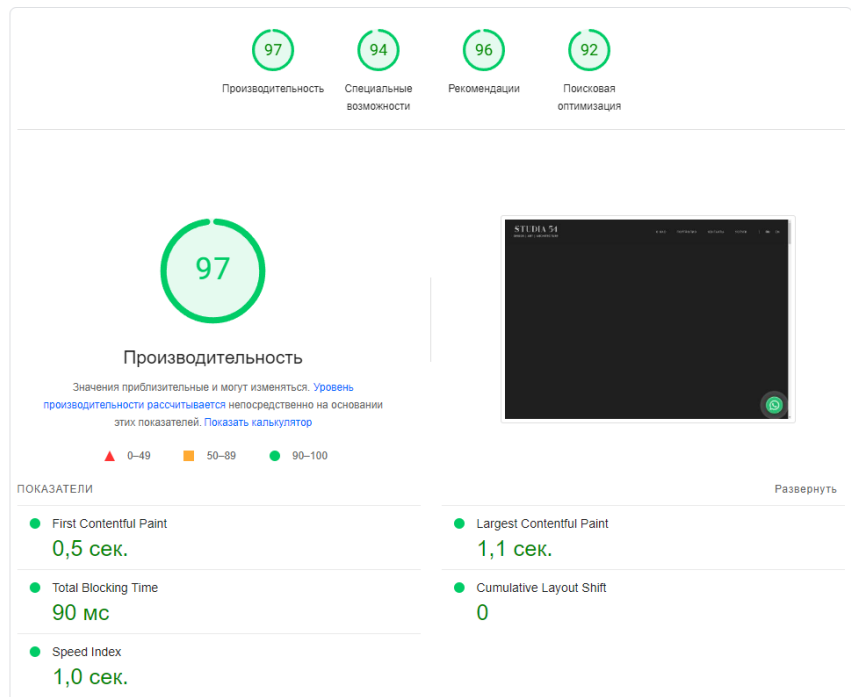


1. Оценка сайта **Studia 54**

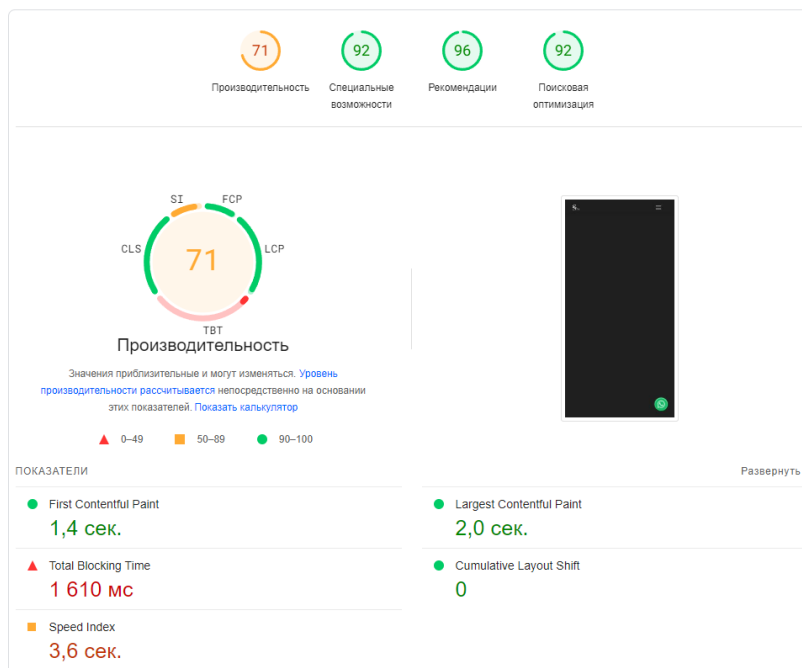
На первый взгляд, сайт выглядит профессионально и технологично, с хорошим дизайном и реализацией. Однако анализ внутреннего построения выявил некоторые проблемы. PageSpeed Insights показал, что сайт хорошо оптимизирован для десктопа, но с мобильными устройствами есть небольшие недостатки. Время загрузки составляет 3,6 секунды, что может вызывать неудобства для пользователей.



На скриншоте видно, что по производиелтности сайт набрал 97 баллов, но есть ошибки на которые аналитическая система ругается, такие как :

ДИАГНОСТИКА	
▲ Удалите неиспользуемый код JavaScript	— Потенциальная экономия – 279 КиБ
■ Для изображений не заданы явным образом атрибуты width и height.	
■ Уменьшите размер кода CSS	— Потенциальная экономия – 7 КиБ
■ Включите сжатие текста	— Потенциальная экономия – 110 КиБ
■ Задайте правила эффективного использования кеша для статических объектов	— Найдено 14 ресурсов
■ Настройте подходящий размер изображений	— Потенциальная экономия – 8 КиБ
■ Отложите загрузку скрытых изображений	— Потенциальная экономия – 31 КиБ
■ Удалите неиспользуемый код CSS	— Потенциальная экономия – 89 КиБ

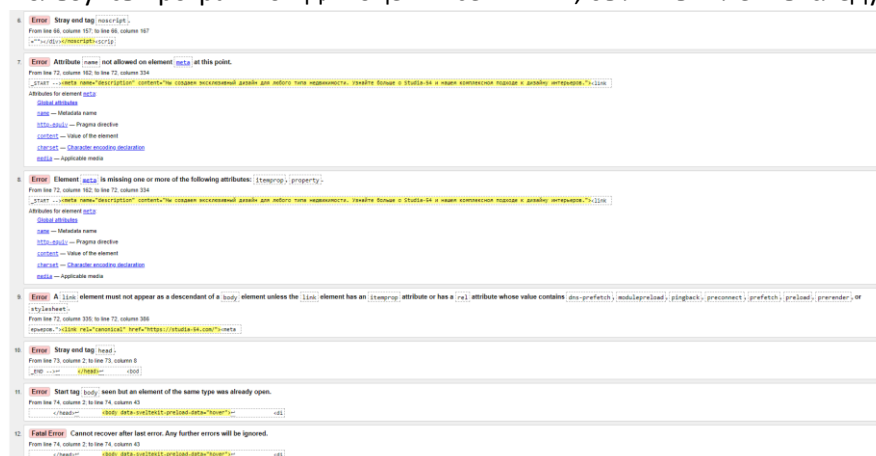
но картина с мобильными устройствами немного хуже:



Мы видим что сайт блокируется на 1,5 секунды, а скорость загрузки составляет 3,6 секунды, что может сильно напрягать пользователей, ошибок здесь немного больше такие как:

▲	Минимизируйте работу в основном потоке	— 6,2 сек.	▼
▲	Сократите время выполнения кода JavaScript	— 3,8 сек.	▼
▲	Уменьшите влияние стороннего кода	— Сторонний код заблокировал основной поток на 1 040 мс	▼
■	Для изображений не заданы явным образом атрибуты width и height.		▼
■	Уменьшите размер кода CSS	— Потенциальная экономия – 7 КиБ	▼
■	Включите сжатие текста	— Потенциальная экономия – 110 КиБ	▼
■	Задайте правила эффективного использования кеша для статических объектов	— Найдено 17 ресурсов	▼
■	Отложите загрузку скрытых изображений	— Потенциальная экономия – 39 КиБ	▼
■	Удалите неиспользуемый код CSS	— Потенциальная экономия – 89 КиБ	▼
■	Удалите неиспользуемый код JavaScript	— Потенциальная экономия – 283 КиБ	▼

Пользуясь программой для оценки семантики, были выявлены следующие проблемы:



Ошибки могут быть вызваны спецификой использования тех или иных библиотек/фреймворков, но стоит уделить внимание данным предупреждениям.

2. Оценка времени создания сайта

Проектирование и построение архитектуры: 1-3 недели

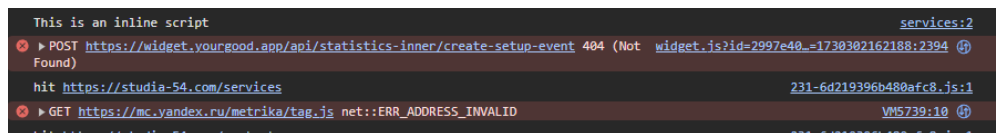
Верстка и адаптивная верстка: 2-4 недели

Разработка функционала (формы, фильтры и т. д.): 3–5 недель

Тестирование и отладка: 1–2 недели

3. Проверка сайта на ошибки

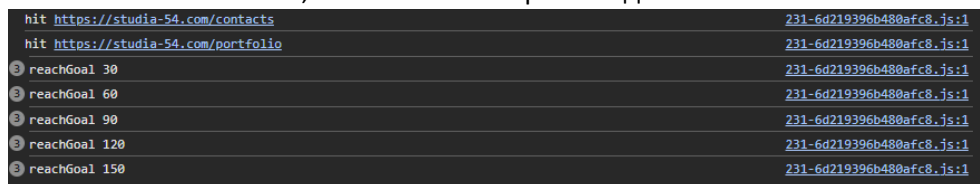
Ошибки консоли



Ошибка GET <https://mc.yandex.ru/metrika/tag.js> net::ERR_ADDRESS_INVALID: Это ошибка загрузки скрипта Яндекс.Метрики. Возможные причины: Неправильный или некорректный URL. Проблемы с DNS (например, неправильное разрешение домена). Возможен блок со стороны сетевого фильтра или брандмауэра.

Ошибка POST <https://widget.yourgood.app/api/statistics-inner/create-setup-event> 404 (Not Found): Ошибка 404 означает, что запрашиваемый ресурс не найден. Возможные причины: Неверный URL API (ошибка в конфигурации). Сервер или API временно недоступен. Эндпоинт больше не существует или был изменён.

Это конечно не ошибка, но вызывает вопросы надобности этих логов в консоли



4. SSL-сертификат нужен для защиты данных, передаваемых между сервером и пользователем (шифрование). Он повышает доверие к сайту и влияет на SEO.

Получить его можно через сертификационные центры (например, Comodo, Let's Encrypt). Let's Encrypt предлагает бесплатные SSL сертификаты. Интеграция обычно проводится через хостинг-провайдеров или вручную.

5. Плюсы и минусы готовых платформ:

Плюсы: Быстрота запуска. Меньше затрат на разработку. Поддержка и обновления от платформы.

Минусы: Ограниченная кастомизация. Зависимость от платформы. Могут возникнуть проблемы с переносом данных и функционала на другую платформу.

6. Инструменты в работе

Программы: VScode, Git, Insomnia, MD compass

Фреймворки: Next.js, express.js

Библиотеки: React, Axios, React Router, Redux или Zustand, Jest и React Testing Library, ESLint и Prettier, Tailwind CSS, различные ui решения.

7. Active Directory (AD) — это служба управления пользователями, компьютерами и группами в Windows-сетях.

Чтобы создать пользователя в AD: Открыть «Active Directory Users and Computers».

Найти OU (organizational unit) или создать новую.

ПКМ по OU → New → User.

Ввести данные пользователя.

Назначить пользователю группы и права через свойства пользователя.

Чтобы создать группу:

ПКМ по OU → New → Group.

Ввести имя группы и тип (например, Security или Distribution).

Назначить права и добавить пользователей.

8. Виртуализация позволяет создавать виртуальные версии серверов, операционных систем, хранилищ данных и сетевых ресурсов. Она используется для консолидации серверов, повышения эффективности использования ресурсов и упрощения управления.

Преимущества:

Экономия на оборудовании.

Гибкость в управлении.

Изоляция приложений.

Быстрое развертывание новых виртуальных машин.

9. Шаги решения проблемы:

Проверить подключение к сети.

Убедиться, что сетевой ресурс доступен (ping или проверка подключения к серверу).

Проверить права пользователя на доступ к папке.

Попробовать переподключить сетевой диск.

Если не помогает, перезагрузить компьютер и сеть.

Проверить настройки брандмауэра или антивируса.

10. Последний код

```
1  const NewBudgetSheet = ({
2    isSheetOpen,
3    budgets,
4    setBudgets,
5    setIsSheetOpen,
6  }) => {
7    const dispatch = useDispatch()
8    const [form] = Form.useForm()
9    const [user] = useUser()
10
11    const [newName, setNewName] = useState<string>('')
12    const [maxExpense, setMaxExpense] = useState<number>()
13    const [icon, setIcon] = useState<string | null>(null)
14    const [color, setColor] = useState<string>('#1677ff')
15
16    const [isIconPeckerOpen, setIsIconPeckerOpen] = useState<boolean>({false})
17
18    const budgetsName = budgets?.budgets.map((budgets) =>
19      budgets.name.toLowerCase()
20    )
21
22    const onDismiss = () => {
23      form.resetFields()
24
25      setNewName('')
26      setMaxExpense(0)
27      setColor('#1677ff')
28      setIsSheetOpen(false)
29      setIcon(null)
30    }
31
32    const submit = async () => {
33      try {
34        const budgetId = user?.budgets
35        const res = await $api.post<UserBudgets>('/budgets', {
36          budgetId,
37          name: newName,
38          maxExpenses: maxExpense,
39          color,
40          icon,
41        })
42        onDismiss()
43        setBudgets((budgets: res.data.budgets))
44      } catch (e) {
45        onDismiss()
46        dispatch(setError('Пожалуйста, попробуйте ещё раз, что-то пошло не так'))
47      }
48    }
49
50    const handleIconSelect = (selectedIcon: string) => {
51      setIcon(selectedIcon)
52      form.setFieldsValue({ icon: selectedIcon })
53    }
54
55    const openIconPecker = () => {
56      if (isSheetOpen) {
57        setIsSheetOpen(false)
58        setIsIconPeckerOpen(true)
59      }
60    }
61
62    const closeIconPecker = () => {
63      if (isIconPeckerOpen) {
64        setIsIconPeckerOpen(false)
65        setIsSheetOpen(true)
66      }
67    }
68    return (
69      <>
70        <BottomSheet
71          open={isSheetOpen}
72          onDismiss={() => onDismiss()}
73          snapPoints={() => 570}
74          header={<SheetHeader />}
75        >
76          <NewBudgetsForm
77            openIconPecker={openIconPecker}
78            setColor={setColor}
79            setNewName={setNewName}
80            setMaxExpense={setMaxExpense}
81            setIcon={setIcon}
82            form={form}
83            setIsSheetOpen={setIsSheetOpen}
84            onDismiss={onDismiss}
85            submit={submit}
86            icon={icon}
87            color={color}
88            budgetsName={budgetsName ?? []}
89          />
90        </BottomSheet>
91        <IconPicker
92          setIcon={handleIconSelect}
93          onClose={closeIconPecker}
94          open={isIconPeckerOpen}
95        />
96      </>
97    )
98  }
99  export { NewBudgetSheet }
```

Этот компонент отвечает за создание новых бюджетов в приложении. Он разделён на основе архитектуры FSD, поэтому у нас есть два BottomSheet, которые позволяют реализовать драйвер, реагирующий на нажатия пользователя. В нём находится форма, куда мы передаём всю необходимую бизнес-логику, таким образом в NewBudgetsForm

реализована концепция чистой (тупой) компоненты, которая ничего не знает о преобразовании данных и просто служит для взаимодействия. Также имеется компонент `IconPicker`, который является общим UI-компонентом, возвращающим выбранную пользователем иконку. Компонент хорошо структурирован и оптимизирован. Однако предела совершенству нет, и даже здесь его можно было бы облегчить