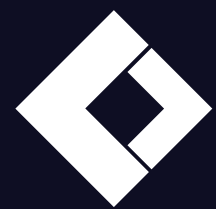


ЗАДАНИЕ НА ЭКЗАМЕН

2021

FULL STACK
JS DEVELOPER



training
center

by FRESHCODE

ИСПРАВЛЕНИЕ БАГОВ

254

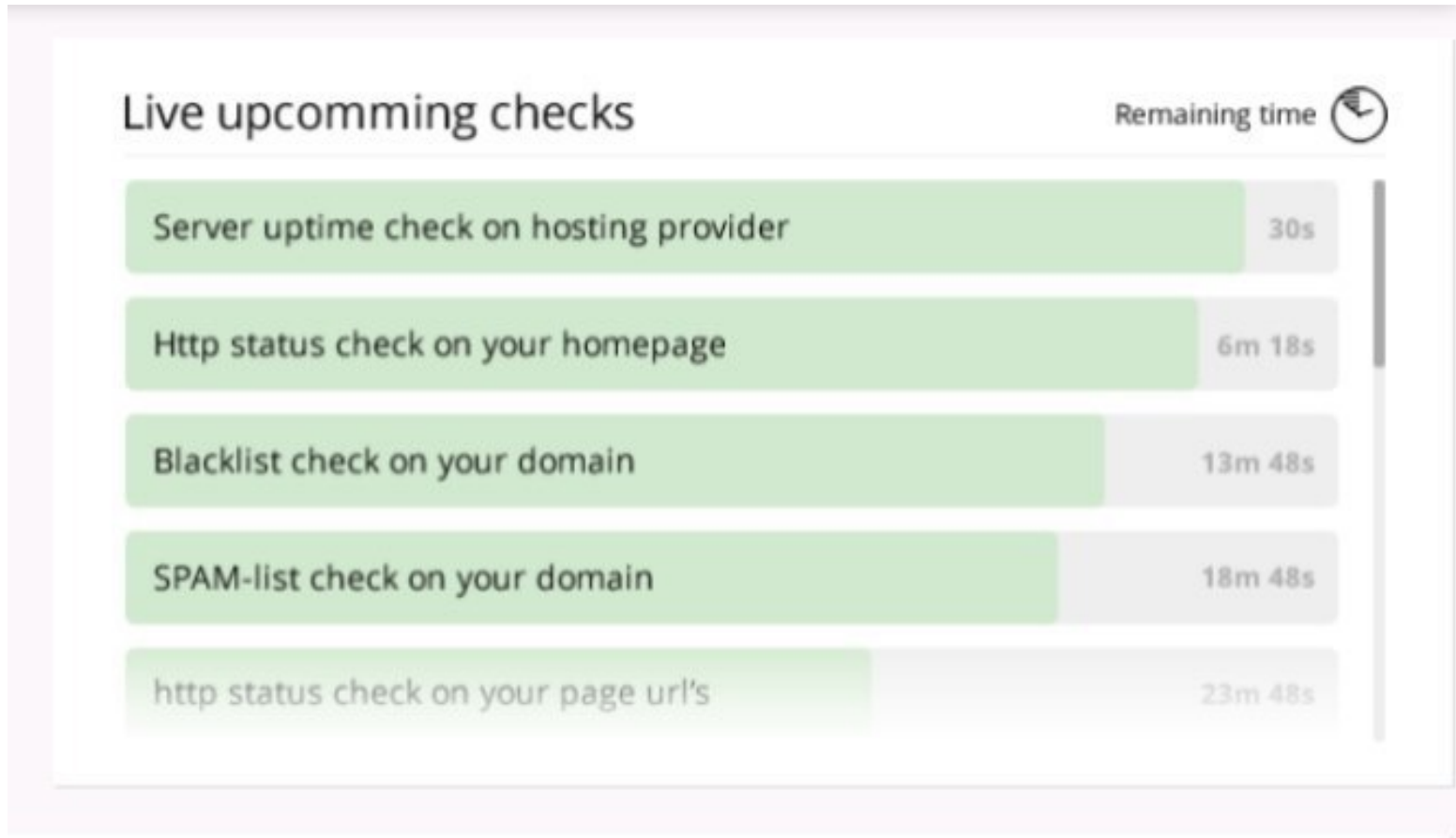
1. Проверить работоспособность всего проекта, в том числе различные способы создания contest, отображение картинок.
2. Проверить правильность работы: отправка сообщений, снятие денег/оплата за контест/контесты.
3. Общий стиль кода всего проекта(однотипность запросов к базе) и функций.

Основное — сделать однотипным свой код, в едином стиле(структуру компонентов, контроллеров, доступа к базе и т.д.)

Что касается рефакторинга — это не самоцель, а демонстрация навыков, поэтому можно ограничиваться переписыванием только явно плохих мест.

ВЁРСТКА

REACT



1. Необходимо сверстать страницу, используя **гриды** или **flexbox** <https://www.squadhelp.com/howitworks.php> + адаптив

64

1. Динамический брендинг. Пользователь хочет быть всегда в тренде и соответствовать сезону или, допустим, к Новому году, хочет обновить слоган, лого или имя. Для этого необходимо пользователю дать возможность создавать таймер и следить, когда же придет время для создания нового контента. Создать новую страницу **Events** и ссылку на нее. Сделать возможным ввод данных(имя события, дата, время, за сколько времени уведомить пользователя о предстоящем таймере) и подтверждение кнопкой. Каждый новый таймер обратного отсчета до того времени, которое указал пользователь, необходимо помещать в список. Список таймеров должен быть отсортирован(ближайшее событие выше). Когда таймер дойдет до указанного пользователем времени - отобразить красный бейджик с кол-вом событий. Пример таймеров - см. скрин. Не забывать о компонентном подходе и адаптиве. Не забывать об остановке таймеров, когда компонент размонтирован. Будет плюсом, если компонент таймера будет написан на хуках. Данные хранятся на фронте, local storage.

204

2. Используя эту страницу <https://www.squadhelp.com/start-contest?step=2&type=1> необходимо сделать **ButtonGroup** компонент.

54

DB NO-SQL

1. Необходимо найти и посчитать, сколько записей в коллекции **Messages** содержат слово «паровоз». Использовать агрегацию для этих целей. Подсчет выполнять на стороне БД. Всё это — за 1 запрос.

14

DB SQL

1. К имеющейся **no-sql DB** необходимо разработать структуру базы данных с использованием **SQL(PostgreSQL)** для чатов, используя существующую БД в качестве эталона. Необходимо минимизировать последствия после миграции сервера с **no-sql** на **sql**(например, имена столбцов должны быть одинаковыми по возможности). Изменения, которые будут влиять на таблицу юзеров, необходимо модифицировать строго посредством **SQL(ALTER TABLE users..., etc...)**. Результаты работы необходимо залить на гит в виде отдельного **SQL** файла. Предоставить схемы **UML** в виде скриншота со всеми возможными отношениями(не забудьте привязать таблицу юзеров).

64

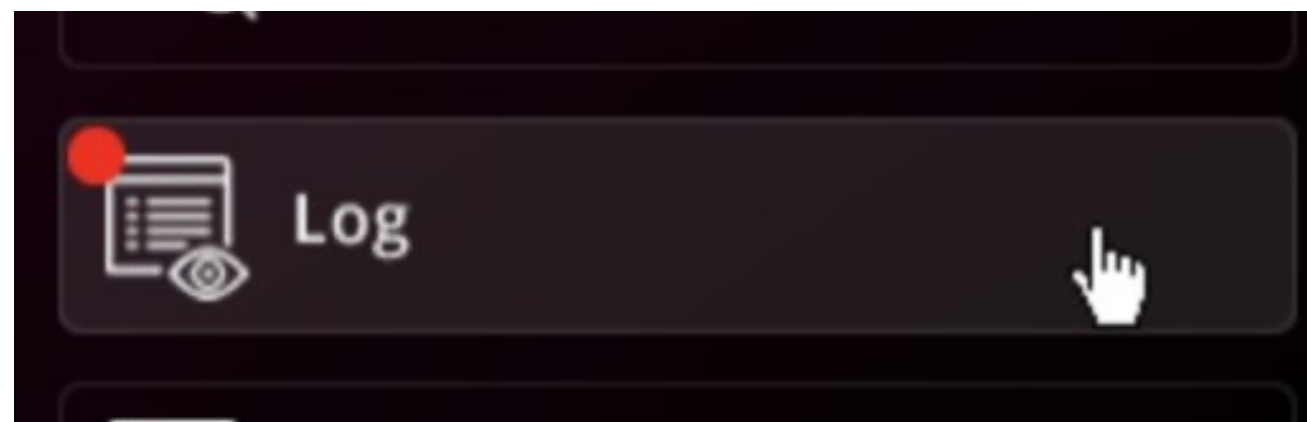
2. Вывести количество юзеров по ролям **{admin: 40, customer: 22, ...}**

14

DB SQL

- | | | |
|----|---|----|
| 3. | Всем юзерам с ролью customer , которые осуществляли заказы в новогодние праздники в период с 25.12 по 14.01 , необходимо зачислить по 10% кэшбэка со всех заказов в этот период. | 14 |
| 4. | Для роли creative необходимо выплатить 3-м юзерам с самым высоким рейтингом по 10\$ на их счета. | 14 |

NODEJS



- | | | |
|----|--|----|
| 1. | Создать логгер ошибок без использования библиотек. Логировать необходимо в JSON файл с заданной обобщенной структурой {message: "", time: 1587410256097, code: 404, stackTrace: {}} | 64 |
| 2. | Задача относится к задаче No 1. Каждый день в определённое время необходимо выполнять копирование содержимого файла и помещать его в новый файл с новым именем(например, таймштамп текущего дня). При этом, содержимое файла перед записью необходимо трансформировать в другой вид {message: "some message", code: 500, time: 1587410256097} . После чего очистить содержимое файла с которого копировали так, чтобы можно было продолжать писать в него ошибки. | 84 |

1. **Login(User should be able to recover lost/forgotten password by email).** Предусмотреть создание новой страницы для восстановления пароля с двумя полями для ввода email и нового пароля, и кнопкой, чтобы отправить. Сервер, принимая email и новый пароль, генерирует токен, в который кодирует захешированный пароль — отправляет на почту ссылку с токеном. Пользователь переходит по этой ссылке, попадает на фронт, с фронта вычленяет токен из ссылки и отправляет его на сервер, а сервер в свою очередь декодирует токен, вычленяет хеш пароля и обновляет поле пароля юзера.
2. Модерация предложений.
Добавить новую роль **Moderator**. Модератор может: посмотреть все оферы, подтвердить офер, отклонить офер. **Customer** не должен видеть каждый новый офер, пока его не подтвердит модератор. Важно! Только модератор может подтвердить офер и больше никто. Модератор не может видеть ничего, кроме оферов! **Creative** может видеть только свои оферы и статусы офера. Предусмотреть отдельную страницу, которая соответствует стилям всего приложения для просмотра таблицы с оферами, на которую может попасть только модератор. Оферов будет много, поэтому пагинация будет плюсом. Добавить рассылку решения модератора на почту **Creative**.

204

254

3. Взяв за основу структуру БД, разработанную в первом задании секции **DB SQL**, описать **Sequelize** модели и миграции. Изменить сопутствующую логику на клиенте и сервере. То есть, приложение работало с чатами на **no-sql db**, теперь должно работать точно так же на **sql db**.

ПРАВИЛА

1. Задания сформулированы для проекта, который прикреплен в архиве.
2. Экзаменационное задание рассчитано на **145 часов** рабочего времени, то есть один месяц. Эстимейты выполнены с большим запасом.
3. Компания относится с пониманием, что не каждый студент может выделить столько времени каждый день, поэтому мы даем Вам до **2,5 месяцев** времени на выполнение всех заданий. При этом каждый студент раз в **2 недели** пишет в тг менеджеру учебного центра **@freshcode_timofey** свой прогресс(что сделано за 2 недели). Обязательно выполнение абсолютно всех заданий(как будто это ТЗ от заказчика).
4. Результаты выполнения заданий нужно заливать на гит в отдельные ветки. Также нужно делать скриншоты результатов работы. В конце нужно собрать все ветки в главную.
5. Когда студент выполнит все задания он пишет менеджеру учебного центра в тг и согласовывает дату и время защиты.
6. На защите необходимо будет ответить на вопросы по коду и сопутствующей теории в рамках программы обучения, обосновать принятые в ходе выполнения заданий решения.
7. Плагиат кода наказывается аннулированием результатов работы студентов, у которых выявлен одинаковый или очень схожий код.
8. После защит студенты получают сертификаты от учебного центра **FRESHCODE** с указанием программы обучения и результатами экзамена. Студенты, которые не выполняют экзаменационные задания, получают сертификаты с указанием программы обучения и с пометкой “Экзамен не сдавал”.

ЖЕЛАЕМ УДАЧИ!