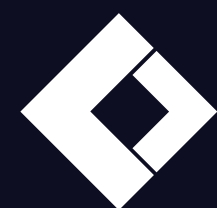


ЗАВДАННЯ НА ІСПИТ

FRESHCODE.TRAINING

2023

FULL STACK
JS DEVELOPER






training
center

by FRESHCODE

FULL STACK JS DEVELOPER by FRESHCODE

ВИПРАВЛЕННЯ БАГІВ

25Г

1.  Перевірити працездатність усього проєкту, зокрема різні способи створення contest, відображення картинок.
2.  Перевірити правильність роботи: надсилання повідомлень, зняття грошей / оплата за контест / контести.
3.  Загальний стиль коду усього проєкту (однотипність запитів до бази) і функцій.

Основне: зробити однотипним свій код, в єдиному стилі (структуру компонентів, контролерів, доступу до бази тощо).

Що стосується рефакторингу - це не самоціль, а демонстрація навичок, тому можна обмежуватися переписуванням тільки явно поганих місць."

ВЕРСТКА



4. Необхідно зверстати сторінку <https://www.squadhelp.com/how-it-works> використовуючи **ґріди** або **flexbox + адаптив**

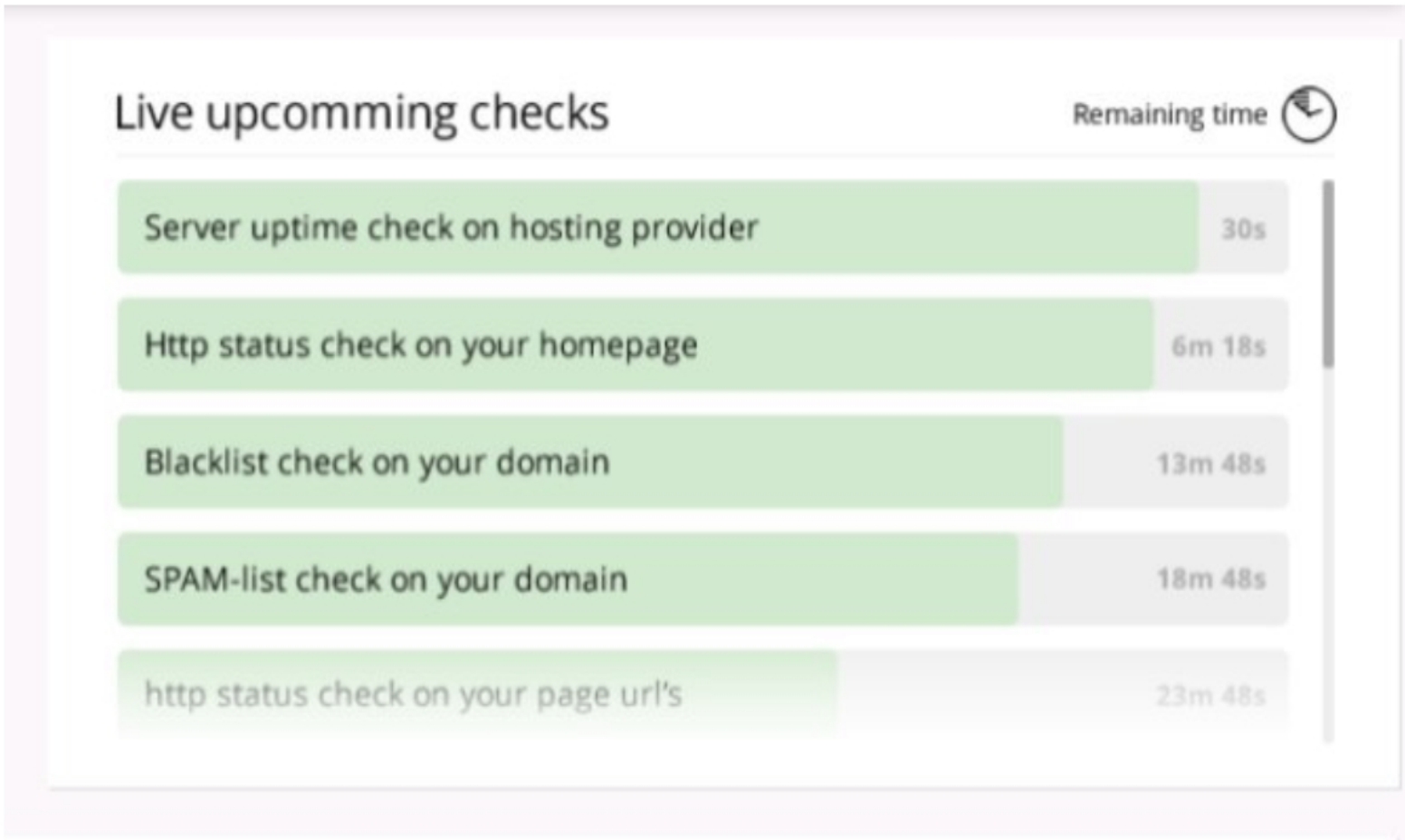
6Г

REACT



5. Динамічний брендинг. Користувач хоче бути завжди в тренді і відповідати сезону або, припустимо, до Нового року, хоче оновити слоган, лого або ім'я. Для цього необхідно користувачеві дати можливість створювати таймер і стежити, коли ж прийде час для створення нового контесту. Створити нову сторінку **Events** і посилання на неї. Зробити можливим введення даних (ім'я події, дата, час, за скільки часу повідомити користувача про майбутній таймер) і підтвердження кнопкою. Кожен новий таймер зворотного відліку до того часу, який вказав користувач, необхідно поміщати в список. Список таймерів має бути відсортований (найближча подія вище). Коли таймер дійде до зазначеного користувачем часу - відобразити червоний бейджик із кількістю подій. Приклад таймерів - див. скрін. Не забувати про компонентний підхід і адаптив.

20Г



6. Використовуючи наступну сторінку необхідно зробити ButtonGroup компонент: <https://www.squadhelp.com/start-contest?step=2&type=1>

5Г

DB NO-SQL



7. Необхідно знайти і порахувати, скільки записів у колекції **Messages** містять слово "паровоз". Використовувати агрегацію для цих цілей. Підрахунок виконувати на стороні БД. Усе це - за 1 запит.

1Г

DB SQL



8. "8. До наявної **no-sql DB** необхідно розробити структуру бази даних з використанням **SQL (PostgreSQL)** для чатів, використовуючи наявну БД як еталон. Необхідно мінімізувати наслідки після міграції сервера з **no-sql** на **sql** (наприклад, імена стовпців мають бути однаковими за можливості). Зміни, які впливатимуть на таблицю користувачів, необхідно модифікувати строго за допомогою **SQL (ALTER TABLE users..., etc...)**. Результати роботи необхідно залити на GIT у вигляді окремого **SQL** файлу. Надати схеми **UML** у вигляді скріншота з усіма можливими



6Г



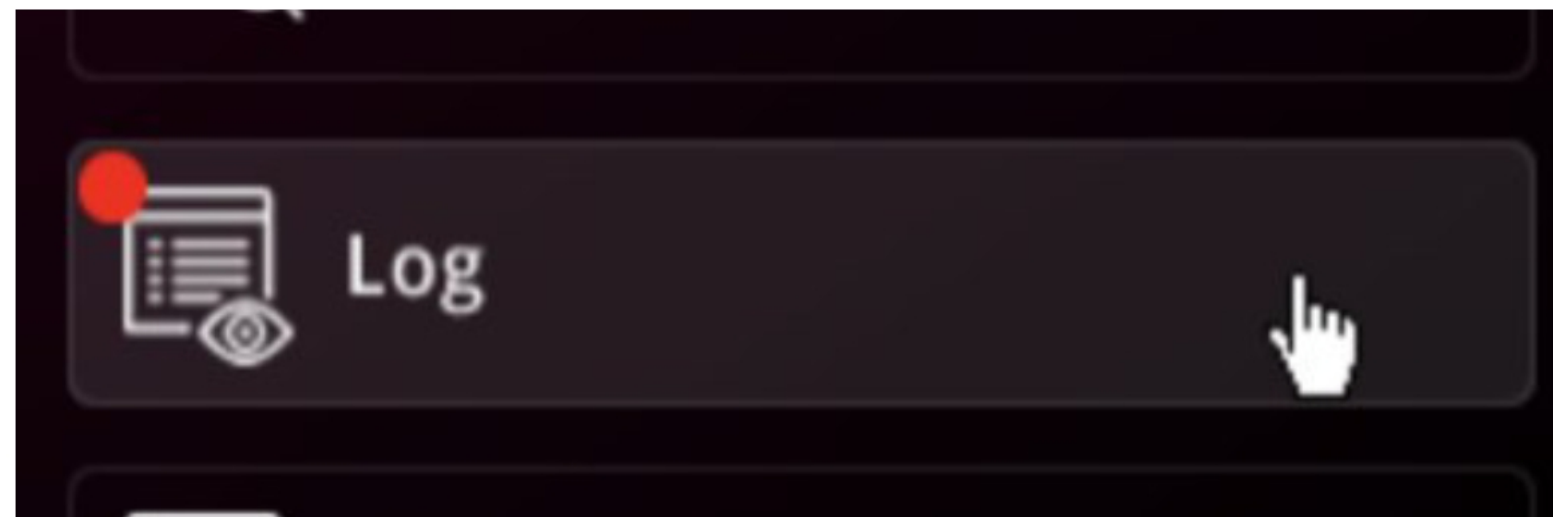
9. Вивести кількість юзерів за ролями {admin: 40, customer: 22, ...}



1Г

DB SQL

-  10. Усім юзерам з роллю **customer**, які здійснювали замовлення в новорічні свята в період з **25.12 по 14.01**, необхідно зарахувати по **10% кешбеку** з усіх замовлень у цей період. 1Г
-  11. Для ролі **creative** необхідно виплатити **3-м** юзерам з найвищим рейтингом по **10\$** на їхні рахунки. 1Г

NODEJS



-  12. Створити логгер помилок. Логувати необхідно у файл із заданою узагальненою структурою **{message: "", time: 1587410256097, code: 404, stackTrace: {}}** 4Г
-  13. Завдання відноситься до попереднього завдання № 12. Щодня в певний час необхідно виконувати копіювання вмісту файлу і поміщати його в новий файл з новим ім'ям (наприклад, таймштамп поточного дня). При цьому вміст файлу перед записом необхідно трансформувати в інший вигляд **{message: "some message", code: 500, time: 1587410256097}**. Після чого очистити вміст файлу з якого копіювали так, щоб можна було продовжувати писати в нього помилки. 8Г



14. Модерація пропозицій (оферів).
Додати нову роль **Moderator**. Модератор може: подивитися всі офери, підтвердити офер, відхилити офер. **Customer** не повинен бачити кожен новий офер, поки його не підтвердить модератор. Важливо! Тільки модератор може підтвердити офер і більше ніхто. Модератор не може бачити нічого, крім оферів! **Creative** може бачити тільки свої офери і статуси своїх оферів. Передбачити окрему сторінку, яка відповідає стилям усього застосунку, для перегляду таблиці з оферами, на яку може потрапити тільки модератор. Оферів буде багато, тому пагінація буде плюсом. Додати розсилку рішення модератора на пошту **Creative**.

25Г

15. Взявши за основу структуру БД, розроблену в першому завданні секції **DB SQL** (завдання № 8), описати **Sequelize** моделі та міграції. Змінити супутню логіку на клієнті та сервері. Тобто, застосунок працював із чатами на **no-sql db**, тепер має працювати так само на **sql db**.

20Г

ПРАВИЛА

1. Завдання сформульовано для проєкту, який прикріплено в архіві `squadhelp`.
2. Екзаменаційне завдання розраховане на **123 години** робочого часу, тобто три тижні. Естимейти виконані з великим запасом.
3. Компанія ставить з розумінням, що не кожен студент може виділити стільки часу щодня, тому ми даємо Вам до **2,5 місяців** часу на виконання всіх завдань. При цьому кожен студент раз на **2 тижні** пише менеджеру навчального центру свій прогрес (що зроблено за 2 тижні). Обов'язкове виконання абсолютно всіх завдань (наче це ТЗ від замовника).
4. Результати виконання завдань потрібно заливати на GIT в окремі гілки. Наприкінці потрібно зібрати всі гілки в головну `main`.
5. Коли студент виконає всі завдання він пише менеджеру навчального центру для узгодження дати і часу захисту.
6. На захисті необхідно буде відповісти на запитання щодо коду і супутньої теорії в рамках програми навчання, обґрунтувати прийняті під час виконання завдань рішення.
7. Плагіат коду карається анулюванням результатів роботи студентів, у яких виявлено однаковий або дуже схожий код.
8. Після захистів студенти отримують сертифікати від навчального центру **FRESHCODE** із результатами іспиту. Студенти, які не виконають екзаменаційні завдання, отримують сертифікати із зазначенням програми навчання та з позначкою "іспит не складав" ("without exam").

БАЖАЄМО УДАЧІ!