**Технічне завдання для розробки веб-застосунку на React.js, Node.js, Express, та MongoDB**:

**Загальний опис**

Розробити веб-застосунок для ідентифікації користувачів за допомогою сканування QR-кодів. Система повинна зберігати дані в базі даних (MongoDB) та відображати їх на веб-сторінці.

**Функціональні вимоги**

Сторінка для сканування QR-коду:

На цій сторінці повинен бути відображений QR-код, який користувач може сканувати.

Після сканування QR-коду система повинна верифікувати девайс користувача.

Якщо користувач успішно ідентифікований, інформація про його девайс повинна бути записана до бази даних.

**Обмеження на сканування**:

Кожен користувач може сканувати QR-код лише один раз на добу (від 00:00 до 23:59).

Система повинна перевіряти, чи не було вже сканувань з цього девайсу протягом поточного дня.

Якщо користувач вже сканував код, система повинна вивести повідомлення про те, що він може сканувати лише після 00:00 наступного дня.

**Таблиця всіх користувачів:**

Створити таблицю, в якій буде відображатися список всіх користувачів, які сканували QR-код.

Таблиця повинна містити інформацію про ID девайсу, кількість відсканованих записів, та дату останнього сканування.

**Технічні вимоги**

* Frontend React.js (опціонально TypeScript)

Створити компонент для відображення сторінки сканування QR-коду.

Додати обробку відповідей від сервера та виведення повідомлень користувачу.

Реалізувати компонент для відображення таблиці з інформацією про користувачів.

* Backend (Node.js, Express):

Створити сервер, який буде обробляти запити від фронтенду.

Написати логіку для ідентифікації користувачів за допомогою QR-коду та збереження даних в базі даних.

Реалізувати перевірку обмеження на сканування та повідомлення користувачам.

* База даних (MongoDB):

Створити колекцію для збереження інформації про користувачів та їх сканування.

Зберігати дані у вигляді об'єктів, що містять ID девайсу, кількість відсканованих записів та дату останнього сканування.

**Додаткові вимоги:**

Додати адаптивний дизайн для зручного використання на різних пристроях.

Забезпечити безпеку за допомогою HTTPS та інших необхідних заходів.

**Додаткові зауваження**

Для взаємодії між фронтендом та бекендом використовуйте REST API.

Рекомендується використовувати пакети та бібліотеки, такі як axios для HTTP-запитів та mongoose для роботи з MongoDB.

**Деплой**

Зберегти проект у Github та завантажити на render.com для перевірки надіслати production link у приватні повідомлення.