

Bootcamp Technical Task – Back-End PHP

"API для зборів з функціональністю"

Завдання: Розробити API, яке дозволить створювати, керувати та отримувати інформацію про збори з додатковою функціональністю.

Вимоги:

- База даних:
 - Створити базу даних для зборів з наступними таблицями:
 - **collections:** Збори, які користувачі створюють.
 - **id:** Унікальний ідентифікатор.
 - **title:** Заголовок зборів.
 - **description:** Опис зборів.
 - **target_amount:** Кінцева сума збору.
 - **link:** Посилання на збори.
 - **created_at:** Дата створення.
 - **contributors:** Внески користувачів до зборів.
 - **id:** Унікальний ідентифікатор.
 - **collection_id:** Зовнішній ключ до таблиці зборів.
 - **user_name:** Ім'я користувача, який зробив внесок.
 - **amount:** Сума внеску.
- Створення зборів:
 - Реалізувати можливість для створення нових зборів за допомогою POST-запиту з вказанням заголовку, опису, кінцевої суми та посилання.
- Отримання списку зборів:
 - Реалізувати можливість отримання списку всіх зборів за допомогою GET-запиту.
 - Кожен збір повинен містити ідентифікатор, заголовок, опис, кінцеву суму та посилання.
- Додавання внесків:
 - Реалізувати можливість долучення до зборів внесків за допомогою POST-запиту з вказанням імені користувача та суми.
- Отримання деталей збору:
 - Реалізувати можливість отримання деталей конкретного збору за його ідентифікатором за допомогою GET-запиту.
 - Деталі повинні містити заголовок, опис, кінцеву суму, посилання та список внесків з ім'ями користувачів та сумами.

- Фільтрація зборів:
 - Реалізувати можливість фільтрування зборів за залишеною сумою до досягнення кінцевої суми.
 - Отримати список зборів, які мають суму внесків менше за цільову суму.
- Додаткові можливості (необов'язково):
 - Реалізувати можливість редагування та видалення зборів та внесків.
 - Додати авторизацію через токени для доступу до API.

Технічні деталі:

- Використовуйте PHP для розробки серверної частини.
- Використовуйте SQL для взаємодії з базою даних.
- Рекомендується використовувати фреймворк для створення API, наприклад, Symfony або Laravel.
- Врахуйте архітектурні принципи RESTful API та забезпечте обробку помилок та валідацію даних.
- Додайте коментарі та документацію до коду, де це необхідно.