

Програма набуття досвіду в CyberBionic Systematics

Дякуємо за цікавість до програми набуття досвіду у CyberBionic Systematics.

Перед проведенням співбесіди ми хочемо, щоб Ви виконали завдання, яке описане в цьому документі.

Для розв'язку вам необхідно використовувати JavaScript (або TypeScript).

З вашим кодом Ви можете надати короткий опис свого дизайну та структури застосунку.

Будь ласка, надішліть свій готовий розв'язок на пошту: **hr@cbsystematics.com**.

Оскільки ми використовуватимемо Ваш код як інструмент для інтерв'ю, CyberBionic Systematics залишає за собою право відкласти ваше технічне інтерв'ю, якщо ваш код не отримано або не прийнято через допущені помилки чи невідповідність вимогам завдання. Конкретний зворотний зв'язок про конкретний код Ви зможете отримати протягом п'яти робочих днів після надсилання результату на пошту, але терміни можуть змінюватись. CyberBionic Systematics не дає гарантій щодо термінів перевірки роботи та зворотного зв'язку за конкретною роботою.

CyberBionic Systematics рада можливості запропонувати Вам кар'єру в нашій команді. Ми бажаємо Вам удачі та з нетерпінням чекаємо на Вашу відповідь.

З повагою –

Менеджер програми набуття досвіду

Завдання на програму набуття досвіду за фахом Frontend-розробник

Необхідно написати клієнтський односторінковий застосунок, який надаватиме можливість маніпулювати даними, зв'язаними з авторами та їхніми книгами.

Автор має такі атрибути:

- Прізвище (текст, обов'язкове).
- Ім'я (текст, обов'язкове).
- По батькові (текст, необов'язкове).
- Дата народження (дата, обов'язкова).
- Список книг.

Книга має такі атрибути:

- Назва (текст, обов'язкова).
- Кількість сторінок (число, обов'язкове).
- Жанр (текст, обов'язковий, зі списку допустимих значень).

Потрібно реалізувати можливість додавати, видаляти та редагувати авторів. Необхідно реалізувати можливість додавати, видаляти та редагувати книги.

На всіх формах має бути реалізована валідація.

На всіх формах, де є табличне представлення даних, має бути реалізовано сортування даних за стовпцями.

Реалізуйте можливість пошуку книг за назвою.

Жанри, які можуть використовуватися під час роботи з книгами, мають редагуватися окремо. Під час створення книги можна вказати лише той жанр, який попередньо було додано до списку жанрів. Користувач має мати можливість додати свій жанр.

Користувацький інтерфейс має бути інтуїтивно зрозумілим.

Приклад інтерфейсу списку авторів:

Авторы

Жанры

[Добавить автора](#)

Авторы

Книг в базе

Пушкин А.С.

5

[Редактировать](#) [Удалить](#) [Детали](#)

Тургенев И.С.

1

[Редактировать](#) [Удалить](#) [Детали](#)

Шевченко Т.Г.

2

[Редактировать](#) [Удалить](#) [Детали](#)

Приклад інтерфейсу сторінки даних одного автора:

Имя

Фамилия

Отчество

Дата рождения

Книги:

Кобзарь

[редактировать](#)

Поэзия 200 ст.

[удалить](#)

[Добавить книгу](#)

- Дані застосунку необхідно зберегти в LocalStorage або реалізувати REST API для збереження інформації на сервері. Зверніть увагу на необхідність зміни джерела даних застосунку в майбутньому.
-
- Мови та технології (виберіть один з варіантів):
 - HTML, CSS, JavaScript.
 - HTML, CSS, jQuery.
 - HTML, CSS, JavaScript, AngularJS.
 - HTML, CSS, TypeScript, Angular.
-
- Вимоги до результату:
 - Результатом має бути робочий застосунок. Вихідні файли застосунку розмістіть на github або bitbucket і надішліть посилання на свій репозиторій.
 - Великою перевагою буде адаптивна верстка й облік кросбраузерності. Плюсом буде використання Bootstrap чи іншої бібліотеки для стилізації інтерфейсу. Зробіть акцент на зручності використання застосунку. Зверніть увагу на розташування елементів управління.