

Vibe Coding + SvelteKit Challenge

Описание проекта

Создайте веб-приложение на SvelteKit с использованием TypeScript и IndexedDB для локального хранения данных. Выполненное тестовое задание должно демонстрировать навыки vibe coding — работу с ИИ-ассистентами для генерации кода.

Выбор типа приложения

Выберите **ОДИН** из вариантов (рекомендуем "Список задач" для начинающих):

- **Список задач** (название, описание, статус: "активная"/"завершена") ☆ РЕКОМЕНДУЕТСЯ
- **Книжная библиотека** (название, автор, жанр, год издания)
- **Список контактов** (имя, телефон, email, компания)
- **Заметки** (заголовок, текст, дата создания, теги)

Технические требования

Обязательные технологии:

- **SvelteKit** — основной фреймворк
- **TypeScript** — типизация кода
- **IndexedDB/dexie** — локальное хранение данных
- **Tailwind CSS** или **Svelte UI** — стилизация

Выбор UI библиотеки:

- **Tailwind CSS** — утилитарный CSS-фреймворк, больше контроля над дизайном
- **Svelte UI** — готовые компоненты для Svelte, быстрее в разработке
- Выбор за кандидатом, но нужно обосновать в документации

Обязательные функциональные требования:

1. **CRUD операции** — создание, чтение записей
2. **Локальное хранение** в IndexedDB с простой схемой (1-2 таблицы)
3. **Поиск** по основному полю (название/имя/заголовок)
4. **Базовая типизация** основных интерфейсов

Бонусные функциональные требования:

- Обновление и удаление записей
- Адаптивный интерфейс (мобильный < 768px, десктоп > 768px)
- Полная типизация всех компонентов

Vibe Coding критерии оценки

1. Эффективность работы с ИИ (35%)

Почему это важно: В нашей команде мы активно используем ИИ-инструменты для ускорения разработки. Нам нужен человек, который умеет "говорить" с ИИ на одном языке и получать от него качественный код.

Примеры хороших промптов:

- ❌ "Сделай красивую форму"
- ✅ "Создай форму с полями name (обязательное, строка), email (обязательное, валидация email), phone (опциональное, маска +7-XXX-XXX-XX-XX)"

2. Быстрое прототипирование (25%)

Измеримые критерии:

- **Время до первого работающего прототипа** — не более 1.5 часа
- **Количество функций в прототипе** — минимум 2 из 4 обязательных функций
- **Готовность к демонстрации** — приложение можно показать и объяснить

3. Качество финального кода (20%)

Измеримые критерии:

- **Отсутствие console.log** в финальном коде
- **Базовая типизация** — основные интерфейсы имеют типы
- **Структура проекта** — логичное разделение на папки
- **Именование** — понятные имена переменных и функций

4. Пользовательский опыт (15%)

Измеримые критерии:

- **Время до первого действия** — пользователь может добавить запись за 30 секунд
- **Обратная связь** — есть уведомления об успехе/ошибке операций
- **Адаптивность** — интерфейс корректно отображается на десктопе (1920px)

5. Документация процесса (5%)

Измеримые критерии:

- **README.md** — содержит инструкции по запуску и скриншот
- **Логи промптов** — минимум 5 ключевых промптов с ответами ИИ

- **Временные метки** — указано время на каждый этап

Конкретные задания с временными рамками

Задание 1: "Быстрая настройка" (1 час) 🕒

Цель: Получить работающий проект за 1 час

- Настройте проект SvelteKit с TypeScript (15 минут)
- Создайте простую структуру через ИИ (25 минут)
- Настройте базовую IndexedDB/dexie схему (20 минут)

Критерий успеха: Проект запускается командой `npm run dev`

Задание 2: "Основная функциональность" (2 часа) 🕒

Цель: Получить работающее приложение с базовым функционалом

- Создайте форму добавления данных (45 минут)
- Реализуйте отображение списка данных (45 минут)
- Добавьте простой поиск (30 минут)

Критерий успеха: Можно добавить запись, увидеть её в списке, найти по поиску

Задание 3: "Финальная доработка" (1 час) 🕒

Цель: Довести до презентабельного состояния

- Завершите базовую типизацию (25 минут)
- Документируйте процесс работы с ИИ (20 минут)
- Подготовьте README и протестируйте (15 минут)

Критерий успеха: Проект готов к демонстрации и имеет документацию

Задание 4: "Отладка и тестирование" (30 минут) 🕒

Цель: Убедиться в стабильности работы

- Исправление ошибок (15 минут)
- Тестирование в разных браузерах (10 минут)
- Финальная проверка функциональности (5 минут)

Критерий успеха: Приложение работает стабильно без ошибок

Задание 5: "Бонусные функции" (если останется время) 🕒

Цель: Реализовать дополнительные возможности

- Удаление записей
- Редактирование записей
- Адаптивность интерфейса

Критерий успеха: Дополнительные функции работают корректно

Чек-лист обязательных требований

Технические (обязательно):

- ☐ Проект запускается командой `npm run dev`
- ☐ Нет критических ошибок в консоли браузера
- ☐ Работает в Chrome (обязательно), Firefox/Safari (желательно)
- ☐ Размер проекта < 2MB
- ☐ Все зависимости в package.json

Обязательные функциональные требования:

- ☐ Можно добавить новую запись
- ☐ Можно просмотреть список записей
- ☐ Можно найти запись по поиску
- ☐ Данные сохраняются в IndexedDB

Бонусные функциональные требования:

- ☐ Можно удалить запись
- ☐ Можно редактировать запись
- ☐ Интерфейс адаптивный (мобильный + десктоп)

Документация (обязательно):

- ☐ README.md с инструкциями по запуску
- ☐ Скриншот интерфейса в README
- ☐ Минимум 5 ключевых промптов к ИИ
- ☐ Указано время на каждый этап

Что мы будем оценивать

1. Навыки работы с ИИ:

- Количество и качество промптов
- Конкретность требований в промптах
- Итеративность улучшений (минимум 3 итерации)

2. Процесс разработки:

- Время до первого прототипа (максимум 1.5 часа)
- Количество функций в прототипе (минимум 2 из 4 обязательных)
- Готовность к демонстрации

3. **Финальный результат:**

- Выполнение всех обязательных пунктов чек-листа
- Качество кода (базовая типизация, структура, именование)
- Пользовательский опыт (время до первого действия < 30 сек)

Дополнительные требования

Время выполнения: 4.5 часа (включая отладку)

Размер проекта: Не более 2MB

Браузеры: Chrome 120+ (обязательно), Firefox 120+, Safari 17+ (желательно)

Что отправить

1. **Исходный код** (архив или ссылка на GitHub)
2. **README.md** с описанием:
 - Краткого процесса работы с ИИ
 - Основных функций приложения
 - Инструкций по запуску
 - Скриншота интерфейса
3. **Краткая документация:**
 - Основные промпты к ИИ (от 5 штук)
 - Время, затраченное на каждый этап
 - Основные решения и их обоснование

Критерии успеха

Ты продемонстрируешь навыки в vibe coding, если:

- Выполнишь все обязательные пункты чек-листа
- Уложишься в 4.5 часа
- Создашь работающий прототип за первые 1.5 часа
- Документируешь процесс взаимодействия с ИИ

Бонусные баллы (если останется время)

Бонусные задания (выполнять по порядку):

1. **Удаление записей** — кнопка удаления для каждой записи
2. **Редактирование** — возможность изменить существующие записи
3. **Адаптивность** — корректное отображение на мобильных устройствах
4. **Полная типизация** — все функции и компоненты имеют типы
5. **Простая анимация** — fade-in эффекты при добавлении/удалении

6. Сортировка — по дате или алфавиту

Важно: Бонусы выполняются только после выполнения всех обязательных требований!

Рекомендуемые инструменты

- **Cursor** — основной IDE для работы с ИИ-ассистентом
- **SvelteKit** — для создания приложения
- **TypeScript** — для типизации
- **IndexedDB/dexie** — для локального хранения данных
- **Tailwind CSS** или **Svelte UI** — для стилизации

Советы по выбору UI библиотеки:

- **Tailwind CSS:** если кандидат хочет показать навыки CSS и дизайна
- **Svelte UI:** если кандидат хочет сосредоточиться на логике и быстро получить результат

Примеры ожидаемого результата

Минимальный результат (проходной балл):

- Проект запускается
- Можно добавить и просмотреть записи
- Есть поиск
- Данные сохраняются в IndexedDB
- Выполнены все обязательные требования

Отличный результат:

- Все обязательные функции работают плавно
- Документированный процесс работы с ИИ
- Выполнены бонусные задания (удаление, редактирование, адаптивность)

Советы по оптимизации времени

1. **Начните с самого простого варианта** — "Список задач"
2. **Используйте готовые шаблоны SvelteKit** для быстрого старта
3. **Сначала сделайте базовый функционал, потом улучшайте**
4. **Не тратьте время на идеальный дизайн** — сосредоточьтесь на функциональности
5. **Документируйте процесс по ходу работы** — не оставляйте на конец