

Nuxt 3 + Vue 3

30-дневный практический челлендж

Конкретные задачи для самостоятельного выполнения

Цель: карьерный рост | Уровень: начинающий | Время: 30-60 мин/день

Z.ai Skill Coach

Обзор челленджа

Как работать с заданиями:

1. Прочтайте задачу и требования полностью
2. Попробуйте решить самостоятельно — это ключ к обучению
3. Если застряли — используйте подсказку (но не раньше 10 минут попыток)
4. Проверьте результат по критериям успеха
5. Используйте официальную документацию: vuejs.org и nuxt.com

Что вы создадите за 30 дней:

- Неделя 1: 5 мини-компонентов Vue 3
- Неделя 2: 5 переиспользуемых компонентов с props/emits/slots
- Неделя 3: Блог на Nuxt 3 с 4+ страницами
- Неделя 4: Каталог товаров с корзиной и поиском

Неделя 1: Vue 3 Основы

День 1: Создание проекта и компонент Greeting

Задача: Создать проект Vue 3 через Vite и компонент приветствия.

Требования к компоненту Greeting.vue:

- Поле ввода для имени пользователя
- Текст приветствия появляется только когда имя введено
- Если поле пустое — показать сообщение 'Введите ваше имя'
- Имя должно отображаться в приветствии

Подсказка: Используйте v-model для привязки input и v-if/v-else для условного отображения.

Критерий успеха: Вводите имя — видите 'Привет, [имя]!'. Стираете — видите приглашение.

Время: 30-45 минут

День 2: Компонент Counter

Задача: Создать счётчик с кнопками увеличения и уменьшения.

Требования к компоненту Counter.vue:

- Отображение текущего числа (начинается с 0)
- Кнопка '+' увеличивает на 1
- Кнопка '-' уменьшает на 1
- Кнопка 'Сброс' возвращает к 0
- Число не может быть меньше 0 (блокировать кнопку '-')

Подсказка: Используйте ref() для реактивной переменной и @click для обработчиков.

Критерий успеха: Счётчик работает, не уходит в минус, сбрасывается.

Время: 30-40 минут

День 3: Компонент TodoList

Задача: Создать список задач с функциями CRUD.

Требования к компоненту TodoList.vue:

- Поле ввода для новой задачи
- Кнопка 'Добавить' (или Enter)
- Каждая задача имеет: текст, чекбокс выполнено, кнопка удаления
- Выполненные задачи зачёркнуты
- Счётчик: 'Всего: X | Выполнено: Y'

Подсказка: Используйте массив объектов {id, text, done} и v-for с :key.

Критерий успеха: Можно добавлять, отмечать, удалять задачи. Счётчик работает.

Время: 45-55 минут

День 4: Computed — фильтрация в TodoList

Задача: Добавить фильтрацию задач: все / активные / выполненные.

Требования:

- Три кнопки для переключения фильтра
- 'Все' — показывает все задачи
- 'Активные' — только невыполненные
- 'Выполненные' — только выполненные
- Активный фильтр визуально выделен

Подсказка: Используйте computed для отфильтрованного массива.

Критерий успеха: Переключение фильтров корректно фильтрует список.

Время: 30-40 минут

День 5: Компонент ColorPicker

Задача: Создать палитру цветов с предпросмотром.

Требования к компоненту ColorPicker.vue:

- Массив из 6-8 цветов на выбор
- Цвета отображаются как цветные кнопки/круги
- При клике — выбранный цвет выделяется рамкой
- Квадрат 100x100px залит выбранным цветом
- Текст: 'Выбран: [название цвета]'

Подсказка: Используйте @click и :style для динамических стилей.

Критерий успеха: Клик меняет цвет квадрата и название.

Время: 35-45 минут

День 6: Watch — Таймер с автосохранением

Задача: Создать секундомер, который сохраняется в localStorage.

Требования к компоненту Timer.vue:

- Отображение времени в формате HH:MM:SS
- Кнопки: Старт / Стоп / Сброс
- При обновлении страницы время сохраняется
- Кнопка Старт неактивна во время работы, Стоп — неактивна при паузе

Подсказка: Используйте watch для сохранения в localStorage при изменении секунд.

Критерий успеха: Таймер работает, сохраняется после перезагрузки.

Время: 45-55 минут

День 7: Dashboard — объединение компонентов

Задача: Создать страницу с виджетами всех компонентов недели.

Требования:

- Сетка 2x2 с виджетами: Counter, TodoList, ColorPicker, Timer
- Каждый виджет в карточке с заголовком и рамкой
- Импортируйте все созданные компоненты в App.vue
- Стилизуйте с помощью CSS Grid или Flexbox

Критерий успеха: 4 виджета отображаются в сетке, каждый работает.

Время: 30-40 минут

Рефлексия недели: Какой компонент был самым сложным? Почему?

Неделя 2: Компоненты и Props/Emits

День 8: Props — компонент UserCard

Задача: Создать карточку пользователя с props для данных.

Требования к компоненту UserCard.vue:

- Props: name (обязательный), email (по умолчанию 'email@example.com'), role (по умолчанию 'user')
- Если avatar не передан — показать круг с первой буквой имени
- Роль отображается в цветном бейдже
- Карточка имеет рамку и отступы

Требования к App.vue:

- Отобразите 3 карточки с разными данными
- Одна карточка только с name (проверка дефолтных значений)

Подсказка: Используйте defineProps с валидацией типов.

Критерий успеха: Карточки отображают разные данные, дефолтные значения работают.

Время: 35-45 минут

День 9: Emits — компонент Rating

Задача: Создать компонент рейтинга со звёздами (1-5).

Требования к компоненту Rating.vue:

- 5 звёзд (можно использовать символы \square и \blacksquare)
- При наведении — звёзды подсвечиваются до курсора
- При клике — выбирается оценка и отправляется наверх через emit
- Prop: modelValue (текущая оценка)
- Emit: update:modelValue (для v-model)

Требования к App.vue:

- Используйте v-model для привязки оценки
- Отображайте текст: 'Ваша оценка: X из 5'

Подсказка: Используйте defineEmits и @mouseenter/@mouseleave для hover.

Критерий успеха: Звёзды кликабельны, оценка отображается в родителе.

Время: 40-50 минут

День 10: Slots — компонент Card

Задача: Создать универсальную карточку со слотами.

Требования к компоненту Card.vue:

- Slot 'header' для заголовка (с дефолтным значением через prop title)
- Default slot для основного контента (с дефолтным текстом)
- Slot 'footer' (показывается только если используется)
- Стилизация: рамка, тень, скруглённые углы

Требования к App.vue:

- Создайте 3 разных варианта использования Card:
 1. С prop title и default slot
 2. С кастомным #header slot
 3. С всеми тремя slots (header, default, footer)

Подсказка: Используйте \$slots.footer для проверки наличия слота.

Критерий успеха: Карточка гибко настраивается через слоты.

Время: 35-45 минут

День 11: Компонент Modal

Задача: Создать модальное окно с Teleport.

Требования к компоненту Modal.vue:

- Props: isOpen (boolean), title (string)
- Emit: close (при закрытии)
- Затемнённый overlay на весь экран
- Модальное окно по центру экрана
- Кнопка 'X' для закрытия
- Клик по overlay закрывает модал
- Используйте Teleport to='body'

Требования к App.vue:

- Кнопка 'Открыть модальное окно'
- В модале: текст и кнопка 'Закрыть'

Критерий успеха: Модал открывается/закрывается, клик вне окна закрывает его.

Время: 40-50 минут

День 12: Composable — useCounter

Задача: Создать composable для логики счётчика.

Требования к composables/useCounter.js:

- Возвращает: count (ref), increment(), decrement(), reset()
- Опциональный параметр: initialValue (по умолчанию 0)
- Опциональный параметр: min и max (границы значений)

Требования к App.vue:

- Создайте два счётчика с разными настройками
- Первый: обычный (0-100)
- Второй: с начальным значением 10 и границами 5-15

Критерий успеха: Два независимых счётчика с разными настройками.

Время: 35-45 минут

День 13: Composable — useFetch

Задача: Создать composable для загрузки данных с API.

Требования к composables/useFetch.js:

- Принимает URL как параметр
- Возвращает: data, loading, error, refetch()
- Автоматически загружает данные при вызове
- Обрабатывает ошибки сети

Требования к компоненту UserList.vue:

- Используйте useFetch для загрузки с <https://jsonplaceholder.typicode.com/users>
- Показывайте 'Загрузка...' пока loading=true
- При ошибке — сообщение об ошибке
- При успехе — список имён пользователей

Критерий успеха: Данные загружаются, отображаются состояния loading/error.

Время: 45-55 минут

День 14: Компонент FormValidator

Задача: Создать форму регистрации с валидацией.

Требования к компоненту RegisterForm.vue:

- Поля: email, password, confirmPassword
- Валидация email: обязательен, содержит @
- Валидация password: обязательен, минимум 6 символов
- Валидация confirmPassword: должен совпадать с password
- Ошибки показываются ПОСЛЕ первой отправки формы
- Кнопка отправки активна только если форма валидна
- При успешной отправке — alert('Регистрация успешна!')

Подсказка: Используйте computed для объекта errors и isValid.

Критерий успеха: Валидация работает, ошибки показываются, форма отправляется.

Время: 50-60 минут

Неделя 3: Nuxt 3 Фундамент

День 15: Создание проекта Nuxt 3

Задача: Создать Nuxt 3 проект и изучить структуру.

Шаги:

1. Создайте проект: `npx nuxi@latest init my-nuxt-blog`
2. Установите зависимости: `npm install`
3. Запустите: `npm run dev`
4. Изучите файлы: `app.vue, nuxt.config.ts`

Требования к `pages/index.vue`:

- Создайте папку `pages`
- Создайте `index.vue` с заголовком 'Мой блог' и приветствием
- Добавьте ссылку `NuxtLink to='/about'` — 'О блоге'

Критерий успеха: Страница открывается, ссылка есть (пока ведёт в никуда).

Время: 30-40 минут

День 16: Маршрутизация

Задача: Создать страницы: `about, blog, blog/[id]`.

Требования к `pages/about.vue`:

- Заголовок 'О блоге'
- Текст описания блога
- Ссылка 'На главную' (`NuxtLink to='/'`)

Требования к `pages/blog/index.vue`:

- Заголовок 'Блог'
- Список из 3 постов (пока локальный массив)
- Каждый пост: заголовок (ссылка на `/blog/[id]`), краткое описание

Требования к `pages/blog/[id].vue`:

- Получить `id` из `route.params.id`
- Показать 'Пост №[id]'
- Ссылка 'Назад к списку'

Критерий успеха: Навигация между всеми страницами работает.

Время: 45-55 минут

День 17: Layouts

Задача: Создать layout с шапкой и футером.

Требования к `layouts/default.vue`:

- Header с навигацией: Главная, Блог, О блоге
- Slot для контента страницы

- Footer с текстом '2024 Мой блог'
- Header тёмный фон, footer — серый

Требования к app.vue:

- Убедитесь, что используется NuxtLayout и NuxtPage

Критерий успеха: Шапка и футер отображаются на всех страницах.

Время: 35-45 минут

День 18: useFetch — загрузка постов

Задача: Загрузить посты из JSONPlaceholder API.

Требования к pages/blog/index.vue:

- Используйте useFetch для загрузки с https://jsonplaceholder.typicode.com/posts?_limit=10
- Показывайте 'Загрузка...' пока данные грусятся
- Отобразите 10 постов: title и body (обрезанный до 100 символов)

Требования к pages/blog/[id].vue:

- Загрузите конкретный пост по id из API
- Отобразите title и полный body

Критерий успеха: Посты загружаются из API, детальная страница работает.

Время: 40-50 минут

День 19: Компоненты в Nuxt

Задача: Вынести пост в компонент PostCard.

Требования к components/PostCard.vue:

- Prop: post (объект с id, title, body)
- Стилизованная карточка с рамкой и отступами
- Заголовок — ссылка на страницу поста
- Краткое описание (обрезанное)

Требования к pages/blog/index.vue:

- Используйте PostCard вместо прямого вывода
- Nuxt автоматически импортирует компоненты из components/

Критерий успеха: Список постов через компонент PostCard.

Время: 35-45 минут

День 20: Страница автора

Задача: Создать страницу пользователя с его постами.

Требования к pages/users/[id].vue:

- Загрузите пользователя: [https://jsonplaceholder.typicode.com/users/\[id\]](https://jsonplaceholder.typicode.com/users/[id])
- Загрузите посты пользователя: [https://jsonplaceholder.typicode.com/users/\[id\]/posts](https://jsonplaceholder.typicode.com/users/[id]/posts)
- Покажите: name, email, company.name

- Список постов пользователя (используйте PostCard)

Требования к PostCard.vue:

- Добавьте ссылку на автора: 'Автор: Пользователь #userId'
- Ссылка ведёт на /users/[userId]

Критерий успеха: Можно перейти к странице автора, видны его посты.

Время: 45-55 минут

День 21: useHead и SEO

Задача: Добавить уникальные title и meta для каждой страницы.

Требования к pages/blog/[id].vue:

- Используйте useHead для установки title равным title поста
- Добавьте meta description с содержимым body (обрезанным)

Требования к pages/users/[id].vue:

- title = имя пользователя
- meta description = информация о пользователе

Требования к nuxt.config.ts:

- Установите дефолтный title: 'Мой блог на Nuxt 3'

Критерий успеха: Проверьте title в браузере — он разный на разных страницах.

Время: 30-40 минут

Неделя 4: Каталог товаров

День 22: Новый проект — главная страница

Задача: Создать проект каталога товаров с главной страницей.

Шаги:

1. Создайте новый проект: `npx nuxi@latest init product-catalog`
2. Создайте `layouts/default.vue` с шапкой и футером

Требования к `pages/index.vue`:

- Заголовок 'Каталог товаров'
- 4 категории в сетке: Электроника, Одежда, Книги, Спорт
- Каждая категория: иконка/эмодзи, название, количество товаров
- Категория — ссылка на `/category/[id]`

Критерий успеха: 4 категории отображаются, кликабельны.

Время: 45-55 минут

День 23: Страница категории

Задача: Загрузить товары категории из FakeStore API.

Требования к `pages/category/[id].vue`:

- Определите категорию по id (1=electronics, 2=jewelery, 3=men's clothing, 4=women's clothing)
- Загрузите товары: [https://fakestoreapi.com/products/category/\[category\]](https://fakestoreapi.com/products/category/[category])
- Отобразите товары в сетке (3-4 колонки)

Требования к `components/ProductCard.vue`:

- Prop: `product` (объект товара)
- Изображение, название, цена
- Ссылка 'Подробнее' на `/product/[id]`

Критерий успеха: Товары загружаются, отображаются в сетке.

Время: 50-60 минут

День 24: Страница товара

Задача: Создать детальную страницу товара.

Требования к `pages/product/[id].vue`:

- Загрузите товар: [https://fakestoreapi.com/products/\[id\]](https://fakestoreapi.com/products/[id])
- Расположение: изображение слева, информация справа
- Изображение: максимальная высота 400px, contain
- Информация: title, category, price, description, rating
- Кнопка 'В корзину' (пока просто заглушка)

Критерий успеха: Страница товара отображает всю информацию.

Время: 40-50 минут

День 25: Корзина (composable)

Задача: Создать composable для корзины и счётчик в шапке.

Требования к composables/useCart.js:

- items — массив товаров в корзине
- addItem(product) — добавить товар (если есть — увеличить quantity)
- removeItem(productId) — удалить товар
- total — вычисляемая сумма всех товаров
- count — общее количество товаров

Требования к layouts/default.vue:

- Добавьте иконку корзины в шапку (ссылка на /cart)
- Покажите badge с количеством товаров (count)

Требования к pages/product/[id].vue:

- Кнопка 'В корзину' вызывает addItem(product)
- При добавлении — краткое уведомление 'Добавлено!'

Критерий успеха: Товары добавляются, счётчик в шапке обновляется.

Время: 50-60 минут

День 26: Страница корзины

Задача: Создать страницу корзины с возможностью удаления.

Требования к pages/cart.vue:

- Если корзина пуста: сообщение 'Корзина пуста' + ссылка на главную
- Список товаров: изображение, название, количество, цена за шт, сумма
- Кнопка 'Удалить' для каждого товара
- Итоговая сумма внизу
- Кнопка 'Оформить заказ' (пока alert: 'Заказ оформлен!')

Критерий успеха: Можно посмотреть корзину, удалить товары, увидеть сумму.

Время: 40-50 минут

День 27: Поиск товаров

Задача: Добавить поиск по товарам.

Требования к layouts/default.vue:

- Добавьте поле поиска в шапку
- При отправке формы — переход на /search?q=[запрос]

Требования к pages/search.vue:

- Получите query из route.query.q
- Загрузите все товары: <https://fakestoreapi.com/products>
- Отфильтруйте по query в title или description
- Покажите результаты через ProductCard

- Если результатов нет: 'Ничего не найдено'

Критерий успеха: Поиск работает, результаты отображаются.

Время: 45-55 минут

День 28: Обработка ошибок и переходы

Задача: Добавить страницу ошибки и переходы между страницами.

Требования к error.vue:

- Создайте в корне pages/ файл error.vue
- Покажите: statusCode, message
- Кнопка 'На главную' с clearError()

Требования к app.vue:

- Добавьте transition для NuxtPage: fade-переход 0.3s

Требования к ProductCard.vue:

- Добавьте скелетон-загрузку при ожидании данных

Критерий успеха: Страница ошибки работает, переходы плавные.

Время: 40-50 минут

День 29: README и рефакторинг

Задача: Описать проект и почистить код.

Требования к README.md:

- Название и описание проекта
- Список функций (категории, корзина, поиск)
- Используемые технологии
- Инструкция по запуску
- Структура проекта

Рефакторинг:

- Проверьте все компоненты на повторяющийся код
- Вынесите общие стили
- Добавьте комментарии к сложным функциям

Критерий успеха: README понятный, код чистый.

Время: 40-50 минут

День 30: Ретроспектива

Задача: Оценить прогресс и спланировать развитие.

Чек-лист навыков:

Навык	Освоено?
ref / reactive	

computed / watch	
v-for / v-if / v-show	
props / emits	
slots	
composables	
Nuxt pages / layouts	
useFetch / useAsyncData	
useHead (SEO)	

Вопросы для рефлексии:

- Что получилось лучше всего?
- Что вызвало наибольшие сложности?
- Какой проект хотите добавить в портфолио?

Следующие шаги:

- Pinia для глобального state-менеджмента
- TypeScript для типобезопасности
- Vitest для тестирования
- Деплой на Vercel или Netlify

Время: 30-40 минут

Поздравляем с завершением 30-дневного челленджа!

Полезные ресурсы

Ресурс	URL
Vue 3 документация	vuejs.org/guide/introduction.html
Nuxt 3 документация	nuxt.com/docs
FakeStore API	fakestoreapi.com
JSONPlaceholder API	jsonplaceholder.typicode.com

Удачи в изучении Vue 3 и Nuxt 3!