# VIBE CODER ROADMAP 2026

## Легенда

- Знать обязательно
- Знать крайне рекомендуется
- Было бы неплохо / для общего развития

Автор: Станислав Быстрицкий (Vibecoder.School)

Самый **полный курс** по вайбкодингу сайтов и веб-приложений

УЗНАТЬ ПОДРОБНЕЕ

## START

## Фундамент AI

- Что такое LLM
- Как LLM генерирует код. Что такое токены
- Галлюцинации, контекстное окно
- Проектные правила
- Декомпозиция задач для AI. Роадмап
- Выбор LLM для планирования и разработки

## Инструменты разработки

- IDE (Cursor, Windsurf, TRAE, VS Code с расширениями, и др. на выбор)
- CLI инструменты (Claude Code, Gemini, Qwen Code и др. на выбор)
- Режимы работы (ask, plan, agent)
- Чекпоинты
- Работа с контекстом
- Проектные правила
- Пользовательские команды
- Индексация документации
- MCP
- Базовая работа с терминалом (навигация, создание и редактирование папок, файлов)
- Индексация кодовой базы, как она работает и зачем нужна
- Воспоминания
- Голосовой ввод

## Figma

- Что это такое? Роль в разработке
- Навигация, горячие клавиши
- MCP для переноса дизайна в код
- Генерация дизайна (Figma Make или сторонние инструменты)

## База верстки

- HTML (что это такое, строение страницы, теги, классы, ид, атрибуты, формы)
- CSS (что это такое, строение CSS правила, CSS-селекторы, самые популярные CSS свойства, flexbox, Media-запросы и адаптивность, подключение css к html)
- JS (что это такое, основные типы данных, переменные, условия if else, циклы, функции, подключение js к html)

## GIT

- Что это такое. Зачем нужен
- Какие сервисы существуют GitHub, GitLab, BitBucket
- Создание репозитория
- Основные команды: git add, git commit, gut push, git pull
- Ветвление
- .gitignore
- Менеджмент секретов (что НИКОГДА не нужно коммитить)
- Git Flow

## Базовый веб и браузер

- Базовый HTTP (запросы, ответы, коды, headers, body)
- Инструменты разработчика в браузере
- Хранение данных в браузере (cookie, localStorage, sessionStorage)
- Клиент-серверное взаимодействие
- Что такое API

## Вводный React

- Что такое React, зачем он нужен
- Как создать проект
- Структура React приложения
- JSX разметка
- Компоненты. Что это, зачем применяются
- Полезные библиотеки
- Локальный запуск фронтенд-приложения
- Typescript (если выучил JS): Что это и зачем применяется
- Роутинг
- Пропсы
- Маппинг
- Работа с состоянием (Redux/RTK или Zustand)
- Фронтенд архитектуры. Какие существуют. Промпт для FSD
- Альтернативные фреймворки, Next JS (server-side rendering)

## Контекст-инжиниринг

- Написание точных запросов к LLM
- Написание требований к продукту
- Предоставление нужного контекста (стек технологий, требования)
- Итеративный подход
- Методологии разработки (спецификаций, BMAD-метод)

## Стилизация

- Tailwind
- SAAS Modules
- Библиотека компонентов shadcn

## Базовые концепции БД

- Что такое БД, зачем они нужны
- Таблицы, колонки, данные
- Supabase
- SQLite. Работа с локальной БД
- MCP для работы с БД
- Отношения (one-to-many, many-to-many)
- PostgreSQL. Зачем нужен. Подключение, просмотр и управление с помощью pgAdmin

## Деплоймент приложений

- Деплой статитечских сайтов (Vercel, GitHub Pages)
- Переменные окружения (.env файлы, secrets)
- Подключение домена
- Деплой на VDS, Railway
- CI/CD (GitLab, GitHub Actions)
- Мониторинг логов, ошибок

## Бэкенд

- Что это такое. Роль в разработке
- CRUD-операции. Их роль
- Правила REST API
- Проектирование REST API
- Авторизация (в т.ч. JWT)
- Написание требований для ИИ
- Выбор языка и фреймворка (NodeJS + Express или Python + Django REST Framework / FastAPI / Flask)
- Локальный запуск бэкенда
- Кэширование (redis)

## JUNIOR