

ИИ-РАЗРАБОТКА БЕЗ ПРОГРАММИСТА

КАК СОЗДАВАТЬ САЙТЫ, ЛЕНДИНГИ
И ЦИФРОВЫЕ СИСТЕМЫ С ПОМОЩЬЮ
ChatGPT, Claude И ДРУГИХ ИИ-ИНСТРУМЕНТОВ



— МЕЛЬНИКОВА Ю.С. —

Мельникова Юлия

**Электронная книга «ИИ-
разработка без программиста»**

«Автор»

2026

Юлия М. С.

Электронная книга «ИИ-разработка без программиста» /
М. С. Юлия — «Автор», 2026

Хотите создавать сайты, приложения и онлайн-сервисы, но не умеете программировать? Начните с искусственного интеллекта! «ИИ-разработка без программиста» — практическое руководство по созданию цифровых продуктов с помощью ChatGPT, Claude и других нейросетей. Уровень подготовки: начинающий, с нуля. Знание языков программирования не требуется. Достаточно базовых навыков работы с компьютером и интернетом. Для выполнения практических заданий рекомендуется ноутбук или ПК. Чему вы научитесь: - Создавать сайты, лендинги и первые версии приложений. - Составлять технические задания для ИИ. - Генерировать, изменять и проверять код. - Работать с GitHub и запускать проекты на сервере. - Автоматизировать бизнес-процессы. - Сокращать время и затраты на разработку. Ваш результат — понимание полного процесса создания цифрового продукта: от идеи до запуска и развития с помощью ИИ. Начните создавать собственные проекты, даже если раньше никогда не писали код!

© Юлия М. С., 2026

© Автор, 2026

Содержание

Предисловие и содержание	5
Введение	8
От чужой очереди — к своим решениям	10
Введение. Практические шаги	12
Глава 1. Как превратить идею в продукт	13
Конец ознакомительного фрагмента.	17

Мельникова Юлия

Электронная книга «ИИ-разработка без программиста»

Предисловие и содержание

ИИ-разработка без программиста

Как создавать сайты, лендинги и цифровые системы с помощью ChatGPT, Claude и других ИИ-инструментов

Содержание

Введение

1. Как превратить идею в продукт
2. Домен и VPS: покупаем адрес и готовим площадку
3. PowerShell и SSH: первый безопасный вход на сервер
4. ChatGPT и Remote Desktop Commander: разрешённый доступ к серверу
5. GitHub и ChatGPT: сохраняем код, а не переписки
6. Первый деплой: Nginx, HTTPS и проверка результата
7. Как определить MVP и не строить лишнего
8. Как заставить ИИ правильно понять задачу
9. Создание технического задания
10. Архитектура будущего проекта
11. Проектирование интерфейсов
12. База данных, пользователи и роли
13. Что такое frontend и backend
14. Как читать код, не зная языка программирования
15. Как давать ИИ задачу на изменение проекта
16. Почему нельзя просить «просто всё исправить»
17. Работа небольшими изменениями
18. Поиск ошибок
19. Логи: как понять, что произошло
20. Code review при помощи ИИ
21. Зачем нужен Git
22. Репозиторий проекта
23. Ветки
24. Commit
25. Pull Request
26. Merge
27. Почему нельзя экспериментировать сразу в main
28. Как откатить неудачное изменение

29. Лендинг с нуля
30. Структура страницы, которая приводит к действию
31. Генерация текстов и офферов
32. Адаптивная версия
33. Формы, заявки и интеграции
34. SEO при помощи ИИ
35. Аналитика и события
36. Публикация сайта
37. Личный кабинет
38. Регистрация и авторизация
39. Административная панель
40. Пользователи и права доступа
41. Работа с файлами
42. Уведомления
43. Поиск и фильтрация
44. Отчёты и аналитика
45. Платежи и подписки
46. Интеграции через API
47. Почему «страница открывается» ещё ничего не значит
48. Автоматические тесты
49. Проверка интерфейса
50. Проверка API
51. Проверка авторизации
52. Тестирование платежей
53. Регрессионные ошибки
54. Чек-лист перед публикацией
55. Что такое сервер
56. Домен и DNS
57. HTTPS
58. Docker
59. Nginx
60. Переменные окружения и секреты
61. Как происходит deployment
62. Логи production
63. Резервные копии
64. Мониторинг
65. ИИ-продуктолог
66. ИИ-аналитик
67. ИИ-дизайнер
68. ИИ-разработчик
69. ИИ-тестирующий
70. ИИ-DevOps
71. ИИ-код-ревьюер
72. Как разделять задачи между моделями
73. Лендинг услуги
74. Интернет-магазин
75. CRM для отдела продаж
76. SaaS-сервис
77. Система аналитики

- 78. AI-сервис на базе API
 - 79. Внутренняя корпоративная система
 - 80. Проект с личным кабинетом и оплатой
 - 81. ИИ придумал существование функции
 - 82. ИИ переписал слишком много кода
 - 83. После исправления сломалось другое
 - 84. Данные удалились
 - 85. Секретный ключ попал в код
 - 86. Тестов нет, но «всё работает»
 - 87. Разработка превратилась в хаос
 - 88. Когда всё-таки нужен профессиональный разработчик
 - 89. Как поддерживать roadmap в актуальном состоянии
 - 90. Backlog
 - 91. Приоритеты
 - 92. Статусы задач
 - 93. Definition of Done
 - 94. Как выпускать изменения регулярно
 - 95. Как развивать продукт месяцами вместе с ИИ
 - 96. Заключение. От идеи к работающему продукту
- Справочник терминов**

Введение

Введение



Я хорошо знаю, как легко потерять важную идею в чужой очереди. Не раз в моей работе возникали небольшие задачи: исправить неудобную форму, добавить простую функцию, убрать лишние действия для клиента. Для команды разработки это могла быть мелочь. Для меня — возможность сэкономить сотни часов, улучшить продукт или увеличить выручку. Я понимала, что именно нужно сделать и почему это важно. Но между идеей и результатом вставали программисты, дизайнеры, проектные менеджеры, согласования, сметы и расписания. Моя задача снова оказывалась где-то в конце списка: у каждого находилось дело с более высоким приоритетом.

От чужой очереди — к своим решениям

От чужой очереди — к своим решениям



Проходили недели. Когда до задачи наконец доходили руки, мне приходилось заново объяснять детали, восстанавливать переписку и напоминать, зачем всё затевалось. Иногда к этому моменту терялись и время, и первоначальный смысл идеи. Но кто лучше меня знал, как должен работать мой продукт?

Почему реализация моих идей должна зависеть от чужого расписания, чужих решений и чужих расценок? Я решила проверить, можно ли разорвать эту цепочку ожидания. Не просить месяцами, не передавать каждую мелочь через нескольких исполнителей, а самой управлять разработкой с помощью искусственного интеллекта.

Оказалось, многие задачи действительно можно решать за часы или дни вместо недель и месяцев: самой ставить цель, собирать прототип, проверять результат и вносить улучшения. Так из практических экспериментов выросли работающие цифровые продукты и появился новый для меня способ разработки — без постоянного ожидания свободного программиста. **Эта книга родилась из моего опыта.** Я покажу, как пройти тот же путь самостоятельно, не забывая о тестах, безопасности и ответственности за результат.

Введение. Практические шаги

У этой книги есть конкретный результат: читатель сможет пройти путь от регистрации домена до публикации собственного сайта, а затем понять, как развивать его в полноценный цифровой продукт с помощью ИИ.

Поэтому начнём с плана, а не с хаотичных действий. В первой главе мы создадим дорожную карту проекта в Google Таблицах и зафиксируем последовательность работ — от покупки домена и настройки VPS до подключения GitHub к ChatGPT, настройки Remote Desktop Commander и публикации сайта. Затем, в главах 2–6, пройдем этот маршрут по шагам: зарегистрируем домен, подготовим сервер, настроим безопасный доступ, подключим рабочие инструменты и опубликуем первую страницу с HTTPS. Каждую операцию можно проверить самостоятельно.

Как пользоваться практическими главами. Не нужно заучивать команды из книги. Они помогают понять, что происходит. На каждом этапе можно спрашивать ChatGPT или Claude: «Что мне сделать сейчас, почему именно так и как проверить результат?». Главное — понимать цель и безопасность действий, а не помнить синтаксис PowerShell, SSH или Git наизусть.

Это учебный проект, а не рекомендация вслепую повторять команды на действующем бизнес-сервере. Примеры доменов, IP-адресов и названий учётных записей условные. Не передавайте ИИ пароли, токены или приватные ключи и не выполняйте потенциально разрушительные команды без резервной копии и понимания их последствий.

После практического старта мы перейдем к проектированию продукта, интерфейсам, серверной логике, тестированию, выпуску изменений и управлению разработкой. Книга не обещает заменить профессионального инженера в задачах повышенного риска. Она учит строить управляемый процесс, задавать верные вопросы ИИ и проверять, что действительно работает.

Глава 1. Как превратить идею в продукт



Сначала — дорожная карта проекта в Google Таблицах

Прежде чем писать код, мы заведём **roadmap** (дорожную карту)— понятный список шагов от идеи до запуска. В обычном разговоре легко забыть, что уже сделано: сегодня обсуждали форму, завтра базу данных, послезавтра вспомнили о домене. В дорожной карте у каждой задачи есть номер, приоритет, ответственный, статус, зависимость и измеримый результат.

На первых этапах нам нужен **MVP (Minimum Viable Product, минимально жизнеспособный продукт)**— самая простая версия, которая уже решает главную задачу пользователя и позволяет проверить, нужна ли она людям. MVP не означает плохое качество или отсутствие защиты данных. Он означает, что в первую версию входит только необходимое: например, для сервиса записи клиентов — форма, сохранение заявки и уведомление, но пока не система бонусов, сложная аналитика и мобильное приложение.

Цель этой главы:создать в Google Таблицах roadmap реальной идеи, выделить обязательные задачи MVP и затем двигаться по нему вместе с ChatGPT или Claude — по одной проверяемой задаче за раз.

Шаг 1. Подключаем Google Таблицы к ChatGPT

В ChatGPT откройте **Настройки → Плагины (или Apps)**и найдите **Google Drive**. Пройдите подключение Google-аккаунта и внимательно проверьте запрашиваемые разрешения. Доступ к таблицам предоставляется через подключённый Google Drive; отдельное приложение Google Sheets для работы из самого чата может не требоваться. Возможность создавать или изменять таблицы зависит от тарифа, настроек рабочей области и выданных разрешений.

Другой вариант — открыть Google Таблицы и установить **ChatGPT for Google Sheets** через Google Workspace Marketplace, затем запустить его из меню расширений. Это отдельный способ работать с ИИ непосредственно внутри таблицы.

Проверка подключения:попросите ChatGPT перечислить доступные действия Google Drive и проверить, может ли он создать *новую*таблицу. Если доступно только чтение, не считайте её созданной: либо разрешите нужное действие в подключении, либо попросите ИИ подготовить структуру таблицы для ручной вставки. Не отправляйте в чат пароль от Google.

Подсказка ИИ:«Помоги подключить мой Google Drive к ChatGPT, чтобы мы могли создать roadmap в Google Таблицах. Я начинающий пользователь. Проведи по одному шагу через настройки, объясняя, какие разрешения нужны, и в конце проверь реальную возможность создать файл. Не утверждай, что таблица создана, пока не получишь ссылку на неё».

Шаг 2. Даём ChatGPT конкретное техническое задание

Не пишите просто «сделай план сайта». Такая команда даёт список хороших пожеланий, но не рабочий инструмент. Используйте запрос, в котором указано, **какой именно файл должен появиться и как вы будете им управлять**.

Я хочу реализовать проект с помощью ИИ, не умею программировать и планирую двигаться по задачам поэтапно.

Идея:[опишите проект своими словами]. **Для кого:**[основные пользователи]. **Главная проблема:**[что не работает без продукта]. **Ограничения:**[бюджет, срок, доступные сервисы].

Сначала задай до семи уточняющих вопросов, без которых нельзя составить рабочий план. После моих ответов **создай в моём Google Drive новую Google Таблицу** «Roadmap — [название проекта]». Если создание недоступно, сообщи это и подготовь аналогичную таблицу для ручного переноса; не выдумывай ссылку.

Сделай лист «**Задачи**». Колонки: ID (например, PRJ-001), Этап, Задача, Короткое описание, Ожидаемый результат, Для MVP (Да/Нет), Приоритет (P0/P1/P2), Статус, Зависит от ID, Ответственный, Оценка трудоёмкости, Плановая дата, Фактическая дата, Проверка готовности, Ссылка на результат, Комментарий, Выполнено (чекбокс).

Статусы для выпадающего списка: **Не начато → В работе → На проверке → Готово**, а также **Заблокировано**. Приоритет P0 — критично для MVP, P1 — важно после запуска, P2 — дополнительные улучшения. Сделай фильтры, закрепи строку заголовков, выдели статусы цветом. Добавь чекбоксы выполнения. Если формула или выпадающий список не поддерживаются доступным действием, объясни, как настроить их вручную.

Добавь лист «**MVP**», который показывает только обязательные задачи первой версии, и лист «**Прогресс**» с количеством задач, числом выполненных, числом заблокированных и долей готовности MVP. Добавь лист «**Риски и решения**»: ID, риск или вопрос, влияние, принятое решение, дата, ссылка на задачу.

Разбей проект на этапы: проверка идеи → структура MVP → прототип → домен и инфраструктура → разработка → тестирование → публикация → первые отзывы. Сформируй 20–35 конкретных задач, укажи зависимости по ID и логическую последовательность, **не ставь выдуманные даты и трудозатраты**. В MVP включи только действительно необходимые задачи.

Не начинай разработку и не отмечай задачи как выполненные без моей проверки. Покажи ссылку на созданную таблицу, объясни, с какой задачи начать и по каким признакам её можно считать завершённой.

Возможно, ChatGPT сначала попросит подтвердить действие с Google Drive. Подтверждайте только создание нужной таблицы. Факт успешного создания — **реальная ссылка на файл, который открывается в вашем аккаунте**, а не текст «Готово!».

Шаг 3. Читаем roadmap без технического образования

Откройте вкладку «Задачи». Например, проект — небольшой сайт для приёма заявок.

ID | Задача | MVP | Приоритет | Статус

ID: PRJ-001; **Задача:** Описать клиента и главную проблему; **MVP:** Да; **Приоритет:** P0; **Статус:** Не начато

ID: PRJ-002; **Задача:** Утвердить структуру первой страницы; **MVP:** Да; **Приоритет:** P0; **Статус:** Не начато

ID: PRJ-003; **Задача:** Подобрать и зарегистрировать домен; **MVP:** Да; **Приоритет:** P0; **Статус:** Не начато

ID: PRJ-004; **Задача:** Создать форму заявки и проверить отправку; **MVP:** Да; **Приоритет:** P0; **Статус:** Не начато

ID: PRJ-005; **Задача:** Подключить HTTPS и опубликовать сайт; **MVP:** Да; **Приоритет:** P0; **Статус:** Не начато

ID: PRJ-006; **Задача:** Добавить расширенную аналитику; **MVP:** Нет; **Приоритет:** P1; **Статус:** Не начато

ID— уникальный номер задачи; его не меняют при сортировке. **Зависимость**— номер задачи, без завершения которой нельзя приступить к следующей. **Статус** показывает, что происходит сейчас; **чекбокс** подтверждает завершение. Наличие красивого списка не означает, что сайт построен: отметка «Готово» появляется только после проверки результата.

Пример приёма задачи PRJ-004: «Отправил форму с тестовыми данными, получил сообщение об успехе, заявка действительно сохранилась в указанном месте, повторная отправка не создала ошибку». Это критерий готовности, который можно проверить, а не абстрактное «форма работает».

Шаг 4. Двигаемся по задачам вместе с ChatGPT

Каждое рабочее занятие можно начинать так:

Открой roadmap проекта в Google Таблицах. Покажи первую задачу со статусом «Не начато», у которой завершены зависимости и которая имеет приоритет P0. Объясни человеческим языком, что нужно сделать. Составь короткий план, дай команды по одной и предложи проверку результата. **Не меняй статус на «Готово» без моего подтверждения.** После проверки обнови именно эту строку по ID, сохрани ссылку на подтверждение результата и предложи следующую задачу.

Если таблица не подключена, можно скопировать нужную строку в чат, выполнить задачу, а статус поменять вручную. Смысл системы не в постоянной интеграции, а в том, что **план не исчезает между переписками.**

Раз в несколько задач просите ИИ проверять: какие P0 ещё открыты, какие задачи заблокированы, не появилось ли новых обязательных требований, не смешались ли MVP и улучшения. Если ИИ предложит сразу переписать всю архитектуру, вернитесь к конкретному ID и его критерию готовности.

Как понять, что план действительно помогает

В хорошем roadmap нет строк вроде «сделать крутой продукт» или «заняться маркетингом». Каждая задача отвечает на вопросы: *что получим, как проверим, что должно быть готово до неё, нужна ли она для первого запуска.* Вы можете на день отвлечься, вернуться и продолжить не с памяти, а с конкретной следующей строки.

Далее в главе мы разберём, как уточнить саму идею и потребность пользователя. После каждого уточнения обновляйте roadmap: он должен отражать реальное решение, а

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.