

НУЛЕВОЙ СОТРУДНИК

Как я создал и продал за миллион рублей
проект, созданный с помощью ИИ

Пошаговое руководство по созданию
и запуску своего продукта с помощью
искусственного интеллекта

Закрия Арон

Основатель Swois.com и TrueSpace

Первое издание · 2025

О чём это пособие

Это пособие — практический путь от идеи до работающего продукта. Здесь нет абстрактных советов. Только конкретные шаги, промпты и инструменты, которые помогли мне создать платформу, набрать почти 9 тысяч пользователей за неделю и продать её за миллион рублей.

Прочитав это пособие, ты:

- ✓ Узнаешь реальную историю создания BookVision от идеи до продажи
- ✓ Научишься раскапывать свою экспертность с помощью мастер-промпта
- ✓ Поймёшь, как генерировать сильные идеи в синтезе с ИИ
- ✓ Получишь пошаговую инструкцию по разработке продукта через Claude Code
- ✓ Освоишь вывод проекта в интернет и его продвижение

✂ КЛЮЧЕВАЯ МЫСЛЬ

Окно возможностей открыто прямо сейчас. Вам не нужен опыт в программировании, не нужна команда и не нужен бюджет. Нужна только идея, нейросеть и готовность действовать.

Пролог

Единственный способ делать великие дела — любить то, что
делаешь
Стив Джобс

Привет! Меня зовут Арон Закрия.

Думаю, ты уже догадался, и я постараюсь быть кратким в этом пособии, рассказав о своей очень интересной истории, связанной с работой с искусственным интеллектом.

Сегодня, в этом пособии, я бы хотел рассказать именно общую историю о том, как у меня получилось создать платформу, набрать там почти 9 тысяч пользователей за неделю и продать её за миллион рублей.

Вы знаете, такие заголовки, такие истории, конечно, звучат как что-то не внушающее доверия, но история реальная. И я не виноват в том, что моя история совпала с реально какими-то фантастическими вещами.

Не знаю, читали вы или нет, но в моём детстве достаточно популярными для меня были сказки братьев Гримм и Андерсена.

Они были в такой дорогой вариации с прорисованными иллюстрациями, со всеми дьяволами, мифическими существами и, так сказать, обкуренными историями, которые особенно любили братья Гримм. У Андерсена всё было намного цивилизованнее, но за что я любил братьев Гримм, так это за то, что они под детскими сказками продавали то, что сейчас можно читать с удовольствием тем, кто любит такие странные истории.

Но, знаете, в этой книге мне нравился не столько текст, сколько наличие именно этих прорисованных иллюстраций. Я довольно долго, наверное, до класса шестого, особо и не читал обычные книги, то есть классику и так далее, то, что советовали в школе, я тоже не читал.

Я долго выяснял, почему у меня была такая история, почему этот переход оказался сложным. Он был связан с тем, что у отца была библиотека, и в этой библиотеке была отдельная полка, где стояли именно книги этого издательства и этого выпуска, где всегда были иллюстрации.

Это было связано с тем, что я читал их, и мне всегда было необходимо видеть иллюстрацию в конце главы или какого-то важного момента в книге, где появляется какое-то существо, организовывается диалог или что-то такое. Мне важно было это как-то визуализировать для себя, понять, как это происходило.

Так вот, из-за того что в других, более серьёзных книгах не было этих иллюстраций, мне не хотелось их читать. Я всегда искал картинки, где-то пытался найти какой-то фильм, учитывая, насколько это плохо было тогда доступно, чтобы посмотреть на героев, на то, как они выглядят, где они находятся и так далее.

Потому что, сами понимаете, детский ум, тем более в селе, не особо может вообще визуализировать, например, роскошь из «Войны и мира» — вот эти мундиры, вот это всё. Это сложно достаточно для подростка.

Собственно, на этом, наверное, очень долго проходил мой переход. Сначала я читал очень очень медленно, потому что всегда визуализировал для себя. Таким образом, я решил эту

проблему для себя и очень заинтересовался книгами.

Я начал читать очень много книг, разных абсолютно, начиная от Драйзера, его я полностью перечитал, заканчивая Диккенсом, самые психологические его книги. Не говоря уже про русскую классику, которую я до сих пор перечитываю, и мне очень нравится.

Вот этот переломный момент, который у меня состоялся, — почему мне начало нравиться, — заключался в том, что я научился это визуализировать про себя.

И здесь я хочу просто выделить, что мы, люди, нуждаемся в визуализации. Нам хочется увидеть это своими глазами, потому что история, которая пишется, — это то, что хорошо знают авторы. Они изначально понимают, как это выглядит.

Поэтому мы, как читатели, не можем увидеть эту книгу полностью, потому что не видели этих героев наяву. Нам хочется визуализировать, и в этом и была вся идея BookVision.

Но я расскажу это — это не такая длинная история. Я вам расскажу, почему и как вы можете получать такие идеи, которые могут закрывать потребности людей. В этом и была моя потребность — в том, что я хотел визуализации.

Я писал другое пособие, и, возможно, я это пособие тоже туда включу в будущем.

Это пособие рассказывает про синтез и как раз про то, как синхронизироваться с искусственным интеллектом.

Я вам крайне рекомендую почитать это пособие прежде, чем вы серьёзно погрузитесь в процесс создания идеи и генерации идей и так далее.

Потому что это пособие полностью посвящено тому, чтобы синхронизироваться и рассказать ему о себе: рассказать, кто ты, чем ты занимаешься, какая у тебя история и так далее, чтобы он мог в тебя погружаться непосредственно.

Ещё раз, я крайне рекомендую именно это пособие посмотреть, потому что здесь полностью повторить процесс в том качестве, в котором оно там описано, я не смогу.

✧ КЛЮЧЕВАЯ МЫСЛЬ

Идея BookVision родилась из моей собственной детской потребности — потребности в визуализации. Самые сильные идеи рождаются из личного опыта, который ты прожил.

Раскопай себя

Кто хочет сдвинуть мир, пусть сдвинет себя

Сократ

Но если вам очень лениво смотреть предыдущее пособие, если вы хотите дальше двигаться, давайте я здесь коротко перескажу, что вам нужно делать.

Итак, сейчас ваша задача в том, чтобы раскопать себя. Звучит странно, конечно, но мы, люди, умеем мыслить моментом. Мы можем уходить в прошлое, но даже так мы мыслим чаще всего моментами, вот так вот последовательно.

У нейросетей система работает по-другому. Они умеют оперировать большим количеством информации за раз. Для того чтобы они приняли информацию, выкачали из нас эту информацию, нам нужно поработать с ними и поработать достаточно внимательно. То есть нам нужно, грубо говоря, показать нейросети нашу карту, структурировать карту нашей жизни, чтобы она смогла доставать из нас идеи, для того чтобы она нас направляла, чтобы мы двигались с ней.

Возможно, вам не нужно будет даже до конца доходить до этого процесса, и по ходу уже будут возникать какие-то идеи. Это раз.

Второе. Эта история не является однозначной. Тут нет такого, что вам нигде нельзя ошибиться. Как раз таки в этом и заключается суть. То есть вы можете где-то вести не так, где-то не ответить, где-то затупить и так далее. Это тоже идеи, понимаете?

Но давайте вернёмся к промпту. У вас сейчас должна быть любая модель под рукой. Любая. Та модель, которая для вас удобна.

Возьмите, если есть DeepSeek, отлично. Откройте DeepSeek. Если есть ChatGPT, замечательно. Откройте Claude. Главное, это не должны быть очень отсталые модели, типа 4o просто или 4o Mini, тем более. Но и при этом фанатеть, пытаться включить самую последнюю модель за подписку за 100 долларов тоже не нужно.

То есть откройте ту модель, которая вам удобна. Лучшую модель, которая у вас сейчас доступна. Если у вас нет никакого ChatGPT или ещё чего-то, просто зайдите в App Store, скачайте себе DeepSeek и установите там думающую модель. И как только вы откроете, просто отправьте туда вот этот промпт и следуйте дальше по инструкции, отвечайте на вопросы, пусть она вас погрузит.

ПРОМПТ

Я хочу сформировать системную картину о самом себе. Пусть это будет мастер-промпт моей жизни. Моя цель — не просто описать себя, а раскопать в своём опыте, наблюдениях, желаниях и раздражениях то, из чего могут родиться сильные идеи: продукты, проекты, бизнесы, контент. Причём идеи не ограниченные только моей профессией или текущей деятельностью, а основанные на всём, что я замечал, хотел, представлял на протяжении жизни.

Ты должен задавать мне вопросы по очереди, не больше двух-трёх за раз, и двигаться по следующим направлениям:

Первое – кто я сейчас. Чем занимаюсь, как зарабатываю, в какой среде нахожусь, что умею делать руками и головой.

Второе – моя история. Чем я занимался раньше, какие были этапы, что я пробовал, что бросал, что получалось само собой, а что давалось с трудом.

Третье – детские и юношеские наклонности. Что меня увлекало в детстве, о чём я мечтал, что я хотел видеть в мире, но этого не существовало. Какие у меня были фантазии, идеи, даже если они казались глупыми. Что я хотел, чтобы кто-то сделал, но никто не делал.

Четвёртое – наблюдения и раздражения. Что меня бесит в существующих продуктах, сервисах, подходах. Что я замечаю такого, чего другие не замечают. Какие моменты в жизни вызывали у меня мысль «почему это до сих пор так?» или «это же можно сделать лучше».

Пятое – скрытые желания и незакрытые потребности. Что я хотел купить, использовать, попробовать, но этого не существовало – ни онлайн, ни офлайн. Какие задачи я решал вручную или костылями, хотя хотел бы иметь для этого нормальный инструмент.

Шестое – что я потребляю и чем вдохновляюсь. Какие книги, фильмы, подкасты, каналы, люди на меня повлияли. Что из этого не просто понравилось, а запало, зацепило, изменило мышление.

Седьмое – мои сильные стороны и суперспособности. В чём я разбираюсь лучше большинства. Что мне даётся легко. За чем ко мне обращаются другие люди. Что я делаю интуитивно хорошо, даже если не считаю это навыком.

Восьмое – паттерны и пересечения. На основе всего, что я расскажу, ищи повторяющиеся темы, пересечения между разными этапами моей жизни, между увлечениями и навыками. Именно на этих пересечениях чаще всего рождаются уникальные идеи.

Задавай вопросы так, чтобы я не просто перечислял факты, а вспоминал конкретные ситуации, моменты, ощущения. Направляй меня, если я отвечаю слишком общо. Копай глубже. Если я говорю «мне нравилось рисовать», спроси – что именно рисовал, почему, что при этом чувствовал, что хотел этим добиться.

В конце, когда картина будет собрана, сформируй структурированный профиль обо мне и предложи направления для идей – продуктов, проектов, бизнесов, контента – которые находятся на пересечении моего опыта, моих наблюдений и незакрытых потребностей, которые я замечал в мире.

Давай начнём.

Смотрите, этот процесс не нужно проходить сразу. Вы можете ответить на парочку вопросов, отойти, вернуться, снова ответить и так далее. Это точно будет для вас полезно.

Это очень важно для того, чтобы вы имели цифровую картину о себе, чтобы вы понимали примерно, кто вы и чем вы занимаетесь, куда двигаетесь и так далее.

И сейчас давайте мы представим, что вы ответили на вопросы и уже погрузились во всю эту историю. ИИ начнёт вам предлагать крутые идеи; он реально начнёт вам предлагать идеи, но вы можете его направлять так, как считаете правильным.

Не бойтесь этого ни в коем случае, не бойтесь ходить по шаблонам, не бойтесь менять шаблон. Вы можете рассказывать какие-то другие истории, которые у вас были в жизни, что вы хотели бы создать.

По-любому, у вас когда-то были какие-то идеи, а вот классно было бы, если бы было вот это. Используйте это, используйте своё мышление, не пытайтесь полностью всё перекинуть на ИИ.

Ваша задача сейчас — просто с помощью ИИ генерировать какие-то идеи на основе гибкости вашего естественного интеллекта.

Давайте представим, что у вас уже появились идеи. Знаете, я вам совру, если скажу, что идеи формируются исходя из какого-то конкретного математического шаблона и математической модели.

И вот вам больше ничего делать не нужно, но при этом то, что я вам рассказал, поможет вам генерировать очень сильные идеи, если вы хотя бы немного будете прикладывать из себя ту необходимую часть работы, которая нужна именно в отсутствие этой шаблонности.

Смелость — вот что нужно ко всему. Это реально поможет вам погрузиться в это и понять правильный синтез с искусственным интеллектом.

Очень многие люди неправильно это представляют. Если вы в интернете слышите: «Я запустил своего агента, он для меня сделает всё», — это неправда. Он не может такое делать, по крайней мере, пока что.

Мощности нашего мозга в десятки раз больше, чем у искусственного интеллекта, не говоря уже о тех вещах, которые в принципе у него отсутствуют. Именно поэтому этот метод, о котором я вам говорю, — о том, что нужно правильно взаимодействовать с искусственным интеллектом, — делает вычислительную часть и рутинную часть.

Вы являетесь вдохновителем — это самая лучшая история.

✂ КЛЮЧЕВАЯ МЫСЛЬ

Не пытайтесь полностью всё перекинуть на ИИ. Ваша задача — быть вдохновителем, а ИИ берёт на себя вычислительную и рутинную часть. В этом и есть суть синтеза.

От идеи к реализации

Начало — половина всякого дела

Пифагор

Наконец-то мы перейдём непосредственно к реализации нашего проекта.

Вообще, вариантов реализации очень много: от очень лёгкого к сложному. В этом пособии я не буду погружаться в то, какие есть варианты. Вы можете об этом просто написать ChatGPT или вашему инструменту сказать:

🔗 ПРОМПТ

Я хочу вот это реализовать. С помощью ИИ пришли мне самые готовые платформы, типа Cursor и Lovable, благодаря которым я смогу реализовать полностью весь этот функционал.

То есть, если вам нужно что-то очень лёгкое, вы можете написать этот промпт и добавить там, что: «Я хочу что-то очень простое и понятное».

Теперь мой путь. Я скачал Visual Studio Code и установил там Claude Code. В целом, ничего сложного в этом нет: Visual Studio бесплатный, а на Claude Code нужно купить подписку.

Давайте я сейчас вам помогу непосредственно весь этот процесс пройти. Если я сейчас здесь буду прописывать непосредственно команды, как вам это сделать, это будет намного сложнее, чем если мы с вами сможем использовать те инструменты и те возможности, которые на данный момент у нас есть.

И поэтому просто напишите вашей нейросети вот следующий запрос.

🔗 ПРОМПТ

Я полный новичок в программировании и не разбираюсь в технических вещах. Мне нужно, чтобы ты провёл меня пошагово, как инструктор, через весь процесс установки Visual Studio Code и интеграции туда Claude Code. Веди меня по одному шагу за раз. Не давай мне сразу весь список действий, а именно по очереди: один шаг — я делаю — отчитываюсь тебе — ты даёшь следующий.

На каждом шаге ты должен: сказать, на какой сайт зайти и дать точную ссылку; описать, на какую именно кнопку нажать, где она находится на странице; если нужно что-то скачать — уточнить мою операционную систему и сказать, какой файл выбрать; если нужно открыть терминал или командную строку — объяснить, где это находится и как открыть; если нужно ввести команду — дать точную команду, которую я просто скопирую и вставлю.

Вот что мне нужно пройти от начала до конца: скачать и установить Visual Studio Code; разобраться с базовым интерфейсом, чтобы я понимал, где что находится; зарегистрироваться на сайте Anthropic и получить API-ключ; установить Claude Code; подключить API-ключ; запустить Claude Code и убедиться, что всё работает.

Перед началом спроси у меня: какая у меня операционная система — Windows, Mac или Linux; устанавливал ли я когда-нибудь что-то через командную строку или терминал;

есть ли у меня уже аккаунт на сайте Anthropic.

Если я чего-то не понимаю, объясняй простым языком без технических терминов. Если используешь термин, сразу расшифруй его. Если мне нужно что-то найти на экране, опиши, где именно это находится: сверху, снизу, слева, справа, какого цвета кнопка, что на ней написано.

Если я застрял или у меня ошибка, попроси прислать скриншот или скопировать текст ошибки, и помоги разобраться.

Давай начнём.

Итак, тот способ, о котором я вам сейчас говорю, — он самый лёгкий, он самый современный.

Я мог, конечно, написать для вас инструкции на несколько страниц, но эти инструкции плохи тем, что, во-первых, они не адаптируются под ваш уровень сейчас, под ваши знания; во-вторых, под ваши технические возможности. У вас могут быть разные операционные системы, где может возникнуть один этап, а в другой операционке — другой, и так далее.

То есть лучше всего, конечно, использовать для этого нейросеть — потому что в неё можно прямо загружать скриншоты ваших проблем, которые у вас возникают, и она сможет вам помочь полностью во всём этом деле разобраться.

✦ ВАЖНО

Нейросеть адаптируется под вашу конкретную ситуацию: операционную систему, уровень знаний, возникающие ошибки. Никакая статичная инструкция этого не даст.

Разработка проекта

Гений — это один процент вдохновения и девяносто девять процентов пота
Томас Эдисон

Итак, вы установили Visual Studio Code, установили Claude Code, всё работает. Что дальше?

Не кодить. Вообще не кодить. Я сам поначалу открывал терминал и писал что-то вроде «сделай мне приложение». И получал кашу. Так это не работает. И если бы это так работало, представляете, какая была бы конкуренция? Все бы просто писали «сделай мне приложение» и получали готовый продукт. Поэтому для нас с вами хорошая новость в том, что, научившись правильно взаимодействовать с ИИ, мы, не разбираясь в коде, тем не менее обладаем экспертизой. Потому что мы умеем управлять этим процессом и правильно с ним взаимодействовать.

Разработка с ИИ — это процесс. Сначала объясняете, что строите, потом составляете план, потом строите. Если перепрыгнете первые два шага, потеряете кучу времени.

Мы будем стараться минимизировать все наши ручные действия. Наша задача — сделать Claude Code нашим администратором, нашей правой рукой. Мы ему пишем, он делает. Залезает в терминал, делает git, запускает проект, выкладывает проект. Всё в рамках нашего проекта. Возможно, это рискованно, потому что вы даёте ему много свободы, но дальше вы сами решаете, насколько хотите рисковать. Можете контролировать каждый шаг, можете дать больше автономии. Суть в том, что мы работаем с ним максимально, чтобы самим не лезть в технические дебри.

Первое, что нужно сделать после запуска Claude Code, — это проверить, какая модель у вас стоит. Прямо в терминале, где у вас открыт Claude Code, набираете `/model` и нажимаете Enter. Он покажет вам, какая модель сейчас используется, и предложит выбрать другую. Там будет Opus 4.6, Sonnet 4.5, Haiku 4.5. Если у вас стоит Haiku или Sonnet, можно на них работать, но это может усложнять процесс. Я рекомендую переключиться на Opus 4.6. Для этого просто набираете `/model opus` и нажимаете Enter. Всё, модель переключилась. Opus 4.6 — это последняя и самая сильная модель, с ней проще всего работать.

И ещё: Claude Code умеет залезать в интернет. Он может сам найти документацию, изучить библиотеку, посмотреть, как что-то правильно делается. Вам не нужно ему всё объяснять, он сам разберётся в технических вещах.

Шаг 1. Документация проекта

У вас есть идея. Пишете Claude Code:

📌 ПРОМПТ

Я хочу создать проект. Вот моя идея: [подробно описываете, что за продукт, для кого он, что делает, какие у него основные функции, как вы это видите, что должен делать пользователь когда заходит, какие страницы будут, что на них происходит]. Создай файл PRD.md с полной документацией проекта. В документации должно быть: описание продукта

и его цель, целевая аудитория, полный список функций, структура всех страниц или экранов с описанием что на каждой происходит, какие данные хранятся в базе данных и как они связаны между собой, роли пользователей если они есть. Технический стек проекта: Node.js для серверной части, Supabase для базы данных, фронтенд на React или HTML/CSS/JavaScript. Хостинг планируется на beget.com. Не пиши код. Только документация.

Читаете, что он написал. Если что-то не так, говорите — он исправит.

Шаг 2. Память проекта

В Claude Code есть файл CLAUDE.md — это память проекта. Каждый раз, когда вы открываете Claude Code, он его читает и сразу понимает контекст. Без него он каждый раз как новый сотрудник. Пишете:

ПРОМПТ

На основе PRD.md создай файл CLAUDE.md в корне проекта. В нём должно быть: краткое описание проекта в два-три предложения, полный технический стек, описание структуры папок проекта, все основные команды для запуска и сборки, правила разработки, например комментарии в коде на русском языке, названия переменных на английском. Этот файл — это твоя память. Каждую новую сессию ты должен начинать с чтения этого файла и понимать, с каким проектом работаешь, что уже сделано и что ещё предстоит.

Шаг 3. План разработки

Без плана Claude Code будет делать всё подряд, как ему удобно.

ПРОМПТ

На основе PRD.md составь пошаговый план разработки и сохрани в PLAN.md. Разбей всё на маленькие конкретные задачи, которые выполняются по одной. Каждая задача должна содержать: что именно создаётся, какие файлы затрагиваются, что должно работать после выполнения этой задачи, как проверить что всё работает. Порядок: сначала базовая структура проекта и настройка окружения, потом схема базы данных, потом серверная часть, потом фронтенд, потом интеграция всего вместе.

Шаг 4. Пошаговая разработка

Берёте первый пункт и пишете:

ПРОМПТ

Выполни первый пункт из PLAN.md. После выполнения покажи что было создано, какие файлы появились, и объясни что каждый файл делает. Дождись моего подтверждения прежде чем переходить к следующему пункту.

Шаг за шагом. Поработали час-два, сохранили, отошли. Вернулись — Claude Code подхватил CLAUDE.md и сразу в контексте.

Когда нужно посмотреть, как проект выглядит, пишете Claude Code: «Запусти проект». Он

запустит, и вы открываете в браузере адрес localhost:3000. Это ваш проект, работающий на вашем компьютере. Вводите в адресную строку localhost:3000 и видите свой сайт. Работает только у вас, но можно всё потрогать, понажимать, проверить.

Шаг 5. Бэкапы и сохранение

Когда у вас что-то работает и вы довольны результатом, сохраните копию. Самый простой способ — сказать Claude Code: «Сделай копию моего проекта на рабочий стол». Он скопирует папку, и у вас будет резервная версия. Если потом что-то сломается, можно к ней вернуться.

Есть и более правильный способ — Git. Это система, которая сохраняет историю всех изменений. Можно откатиться к любой предыдущей версии проекта. Не нужно разбираться в Git самому, просто пишете Claude Code: «Инициализируй git в моём проекте и сделай первый коммит с сообщением "начальная версия проекта"». Он всё сделает. И дальше, после каждого удачного этапа, говорите ему: «Сделай коммит с описанием что было сделано». Это как точка сохранения в игре — можно откатиться, если что.

Если что-то сломалось, у вас есть несколько способов показать проблему Claude Code. Можете скопировать текст ошибки и отправить ему: «Вот ошибка, помоги разобраться. [вставляете текст ошибки]». А можете вообще не копировать ничего, а просто сделать скриншот экрана с ошибкой и прикрепить его прямо в терминал Claude Code. Он увидит скриншот, прочтает что там написано и сам разберётся, в чём проблема. Это вообще самый удобный способ, особенно если вы не понимаете, что именно нужно копировать.

✂ КЛЮЧЕВАЯ МЫСЛЬ

Хостинг — это потом. Сначала всё должно работать у вас на компьютере через localhost. Не торопитесь выходить в интернет, пока проект не готов.

Выход в интернет

Будущее принадлежит тем, кто верит в красоту своей мечты

Элеонора Рузвельт

Когда всё работает у вас локально, когда вы открываете localhost:3000 и видите свой проект, следующий шаг — это вывести его в интернет, чтобы другие люди тоже могли им пользоваться.

Для этого нам нужен сервер. Сервер — это по сути компьютер, который работает 24/7 где-то в дата-центре и на котором крутится ваш проект. Мы будем использовать beget.com. Это российский хостинг с облачными серверами, там всё на русском, всё понятно, и самое главное, у них есть готовые решения, то есть вам не нужно ничего настраивать вручную.

Заходите на beget.com, регистрируетесь, создаёте аккаунт. После регистрации вы попадёте в панель управления. Наверху будет вкладка «Облако», нажимаете на неё. Дальше нажимаете кнопку «Создать» и выбираете «Виртуальный сервер».

Вам нужно создать два сервера. Первый для самого проекта, второй для базы данных.

Начнём с первого. Когда вы создаёте виртуальный сервер, вам предложат выбрать конфигурацию. Для начала хватит самой минимальной — это 1 ядро, 1 ГБ памяти, 10 ГБ диска, стоит 7 рублей в день. Потом, если проект вырастет, мощности можно увеличить в любой момент, это делается в пару кликов прямо в панели.

Дальше вам предложат выбрать готовое решение. Там будет список, нажимаете «Показать все 57 приложений» и в поиске вводите «Node». Появится Node.js, выбираете его. Он вам также предложит указать доменное имя. Если у вас нет домена, можете использовать бесплатный технический домен в зоне .beget.app. На него автоматически установится SSL-сертификат, это значит, что сайт будет работать через https.

Создаёте сервер. После создания вам на почту придут все данные для доступа: IP-адрес сервера, логин (обычно root) и пароль. Это ваши ключи от сервера, сохраните их.

Теперь второй сервер, для базы данных. Всё то же самое: «Облако», «Создать», «Виртуальный сервер». Конфигурацию тоже можно взять минимальную для начала. Но в готовых решениях вместо Node.js ищите «Supabase» и выбираете его. При создании вам предложат указать домен, имя пользователя и пароль. Запомните их или запишите — они понадобятся для входа в панель Supabase. После создания сервера заходите по адресу вашего домена (тот, что вы указали), вводите логин и пароль, и попадаете в панель управления Supabase. Там нажимаете «Connect» в верхнем левом углу и получаете данные для подключения — URL и ключи.

Итого у вас теперь есть: данные от сервера с Node.js (IP, логин, пароль) и данные от Supabase (URL, ключи, логин, пароль). Эти данные нужно передать вашему проекту, чтобы он знал, куда подключаться.

И вот тут момент с безопасностью. Эти пароли и ключи нельзя просто вписывать прямо в код. Потому что если вы когда-нибудь выложите код куда-то, например на GitHub, все эти пароли будут видны. Но при этом нам нужно, чтобы Claude Code имел к ним доступ и мог с ними работать. Решение простое. Пишете Claude Code:

🔗 ПРОМПТ

Создай в корне проекта файл `.env` для хранения секретных данных. Вот мои данные:
Сервер (Node.js): IP: [вставляете] Логин: [вставляете] Пароль: [вставляете]
Supabase: URL: [вставляете] Ключ: [вставляете]

Сохрани их в файл `.env`, добавь `.env` в `.gitignore` чтобы этот файл никогда не попадал в `git` и никуда не утёк, и настрой проект так, чтобы он читал эти данные из `.env` автоматически. Также создай файл `.env.example` с такой же структурой, но без реальных значений, чтобы было понятно какие переменные нужны.

Что здесь происходит. Файл `.env` хранится только у вас на компьютере. Он в `.gitignore`, значит `git` его игнорирует, он никуда не попадает, никуда не загружается. `Claude Code` при этом спокойно с ним работает, читает оттуда данные, подключается к серверам. А `.env.example` — это просто шаблон без паролей, чтобы вы помнили, какие переменные нужны, если будете настраивать проект на другом компьютере.

Это абсолютно нормальная практика, так делают все разработчики. Данные лежат локально на вашем компьютере и никуда не уходят.

Когда проект готов к выкладке на сервер, говорите `Claude Code`:

🔗 ПРОМПТ

Мне нужно выложить проект на мой сервер. Вот данные сервера: [IP, логин, пароль, всё из `.env`]. Подключись к серверу, загрузи туда проект, установи все зависимости и запусти. Проект должен работать по адресу моего домена.

Он сам подключится к серверу, загрузит файлы, всё настроит и запустит. Вы открываете браузер, вводите адрес вашего домена и видите свой проект, уже доступный для всех в интернете.

Проект мы создали, можем его дорабатывать. У нас создаётся копия проекта, мы безопасно его выкладываем. Можно потом присоединить к нему свой домен и так далее. По факту, поздравляю — у вас готовый проект, который вы разработали.

🔗 КЛЮЧЕВАЯ МЫСЛЬ

Вы только что прошли путь от идеи до работающего продукта в интернете. Без единой строчки кода, написанной вручную.

Продвижение

Не бойтесь совершенства — вам его не достичь

Сальвадор Дали

Теперь его нужно как-то продвигать. Инструментов много, это обычный бизнес-проект. Есть платные инструменты, есть бесплатные. Я считаю, что нужно пользоваться бесплатными. И вот почему.

Может, вы не знали, но я вам скажу как человек, который набрал 450 тысяч подписчиков за 2 года. Инстаграму и другим платформам, где нормально работают рекомендательные алгоритмы — типа Инстаграма, TikTok, YouTube — им всё равно, сколько у вас подписчиков. Для того чтобы ваше видео стало вирусным, оно просто должно быть интересным. Не важно, куда вы его залили. Подписчики дают тебе гарантию базовых просмотров. Если ты стабильно снимаешь более-менее качественный контент, то от подписчиков у тебя просмотров в районе 20 процентов на Reels — это базовые. А всё что выше — это уже вирусность. И каждый на своём аккаунте может создавать вирусные видео.

Вам нужно создать сценарии и снять Reels, где вы будете обозревать максимально любительским образом ваш продукт. Для этого не нужны крутые камеры, настройки, освещение. Нет. Нужно просто показать проект так, чтобы он заинтересовал человека.

Для создания сценариев можете использовать вот этот промпт. Отправьте его в любую нейросеть:

ПРОМПТ

Мой сервис: [описание вашего проекта, что он делает, для кого, какую проблему решает]. Помоги придумать 10 сценариев для коротких видео. Сначала задай мне вопросы по очереди: кто мои пользователи, почему они приходят, какая у них реакция когда видят результат, что они говорят друзьям когда скидывают ссылку, что в моём продукте вызывает вау-эффект. После этого предложи 10 сценариев до минуты. Сценарии должны быть разных форматов: просто показать экран с результатом, разыграть сценку между людьми, снять реакцию человека который впервые пробует, задать вопрос в камеру и показать ответ через сервис, сравнение до и после. Не ограничивай меня форматом «говорящая голова зачитывает текст». Предлагай такие идеи, которые человек увидит в ленте и остановится. Каждый сценарий должен с первых секунд попадать в точку, где зрителю это актуально. Он должен захотеть это использовать.

Берёте эти сценарии, создаёте аккаунт и начинаете выкладывать. Это мой способ. У вас, разумеется, могут быть свои. Моя задача просто рассказать, как я это делал. У меня с BookVision одно видео зашло на 600 тысяч просмотров и привело 10 тысяч пользователей. Те самые 10 тысяч пришли чисто с одного Reels, который залетел. Возможно, без этого залёта ко мне пришлось бы тысячи полторы, потому что у меня есть подписчики. Но все остальные пришли от того, что видео и сама идея залетела. Поэтому если вы думаете, что вам нужны подписчики для того, чтобы продвигать свой проект, то нет.

Обязательно оставьте где-нибудь контакты и возможность связаться с вами. Для инвесторов, для тех, кто хочет вступить с вами в долю, проинвестировать, или просто купить ваш проект. У

огромного количества компаний есть потребность в таких проектах. И эта потребность не обязательно в том, чтобы зарабатывать непосредственно с вашего продукта. Многие проекты являются буферной частью большого бизнеса и помогают людям лучше генерировать трафик.

Объясню на примере. Представьте себе Литрес. Литрес может привлекать пользователей через платную таргетированную рекламу, например «самые дешёвые книги», и цена за переход будет условно 150 рублей. А теперь представьте вторую рекламу, где написано «бесплатно сгенерируй своего персонажа из книги». И для того чтобы сгенерировать больше пяти персонажей, нужно перейти на Литрес и что-то там сделать. В этом случае мы собираем ещё более заинтересованную аудиторию, но при этом намного дешевле, потому что «бесплатно сгенерируй» звучит намного лучше. Вот именно поэтому ваш проект может продаться. Даже если сам по себе он не будет генерировать деньги, он может стать хорошим дополнением другого большого проекта.

Вариантов может быть много. Вы можете подключить эквайринг и принимать оплаты, например за дополнительные генерации или за что-то ещё в рамках вашей идеи. Вариантов монетизации очень много, это обычный бизнес.

✂ КЛЮЧЕВАЯ МЫСЛЬ

Вам не нужны подписчики, чтобы продвигать проект. Нужно одно вирусное видео. Алгоритмам всё равно, сколько у вас подписчиков — им важно, насколько интересен ваш контент.

Заключение

В любом случае, я описал честно весь свой путь. Что я делал, как я делал, насколько это было возможно. Надеюсь, это пособие было для вас полезно. Если было — пишите своим родственникам, друзьям, пусть они тоже погружаются. Мы все уникальные, у каждого есть свои идеи. Я думаю, что мы со своими крутыми идеями можем сделать этот мир интереснее, живее. Это шанс для тех, у кого были сильные идеи, у кого есть опыт, у кого есть свои желания, реализовать эти желания и двигаться дальше.

*Это пособие для вас писал Арон Закрия,
с большим уважением, любовью и заботой.*

2025