

Фаза 1: Запуск MVP – создание ядра и ручная проверка

Цель: Быстро запустить рабочую версию продукта, чтобы проверить основную бизнес-идею: готовы ли пользователи проходить обучение и выполнять целевое действие (регистрация на бирже) для получения доступа к полному курсу. На этом этапе мы минимизируем техническую сложность, используя ручные процессы там, где автоматизация требует времени.

Шаг 1.1: Создание образовательных ботов на платформе Gemini («Gems»)

Ваше обучение будет проходить не в Telegram, а в двух кастомизированных чат-ботах, созданных прямо на платформе Google Gemini. Это значительно упрощает разработку, так как вам не нужно программировать логику диалога с нуля.

- **Действие:**

1. **Перейдите в Google AI Studio или приложение Gemini.** Вам понадобится аккаунт Google.
2. **Создайте нового «Gem» бота.** Найдите опцию для создания кастомного бота (например, «Create a new Gem»).
3. **Настройте Gem Bot 1 (первые 3 урока):**
 - **Имя:** Дайте ему понятное имя, например, «SferaTC: Вводный курс».
 - **Инструкции (Промпт):** Это самый важный шаг. Вы должны текстом описать, кем является бот и как он должен себя вести. Используйте четкую структуру, чтобы задать четырехчастную модель урока.
 - **Роль:** «Ты — опытный и дружелюбный преподаватель по криптотрейдингу. Твоя задача — провести пользователя через три вводных урока».
 - **Структура урока:** «Каждый урок состоит из четырех частей. Начинай с теории. После теории переходи к ответам на вопросы по пройденному материалу. Затем проведи интерактивную игру для закрепления знаний. В конце дай домашнее задание. Переход к следующему уроку возможен только после того, как пользователь его выполнит».
 - **Интерактивная игра:** Опишите механику игры. Для начала это может быть простой текстовый квест или викторина. [1, 2] Пример инструкции: «Для интерактивной игры предложи пользователю сценарий рыночной ситуации и 3 варианта действий. Оцени его выбор и объясни, почему он был верным или неверным».
 - **Завершение:** «После успешного выполнения домашнего задания к третьему уроку, поздравь пользователя и сообщи ему: „Для получения доступа к остальным 27 урокам, перейди в наш Telegram-бот SferaTC Bot для дальнейших инструкций“. Затем предоставь ему эту ссылку: [здесь будет ваша глубокая ссылка, которую мы создадим позже]».
4. **Создайте Gem Bot 2 (остальные 27 уроков):** Повторите процесс, но с другими инструкциями. Этот бот будет содержать основной курс. Ссылка на него будет приватной и будет выдаваться только в Telegram-боте после верификации.

5. Получите публичную ссылку на **Gem Bot 1**. Сохраните ее, она понадобится для Telegram-бота.

Шаг 1.2: Создание ИИ-психолога на платформе ChatGPT

Аналогично обучающим ботам, эта функция будет реализована на внешней платформе, что снимает с вас нагрузку по ее разработке.

- **Действие:**
 1. Войдите в свой аккаунт **ChatGPT** (требуется платная подписка для создания кастомных GPT).
 2. Перейдите в раздел «**Explore GPTs**» и нажмите «**Create a GPT**».
 3. **Настройте бота:** В поле для инструкций опишите его роль и ограничения, как мы это делали в предыдущем отчете. [3, 4]
 - **Пример инструкции:** «Ты — эмпатичный ИИ-психолог, специализирующийся на стрессе, связанном с трейдингом. Твоя задача — оказывать эмоциональную поддержку. Не давай финансовых советов и медицинских диагнозов. Всегда напоминай, что ты — ИИ, а не лицензированный специалист».
 4. **Сохраните GPT и получите на него публичную ссылку.** Она понадобится для кнопки в Telegram-боте.

Шаг 1.3: Разработка Telegram-бота SferaTC (Центральный хаб)

Это ядро вашей экосистемы. Он будет управлять пользователями, монетизировать их и направлять на другие сервисы.

- **Технологии:** Мы будем использовать язык программирования **Python** и библиотеку **python-telegram-bot**. Это популярный, надежный и хорошо документированный выбор. [5, 6]
- **Действия по разработке:**
 1. **Регистрация бота и получение токена:**
 - Найдите в Telegram пользователя **@BotFather**.
 - Отправьте ему команду `/newbot`.
 - Следуйте инструкциям, чтобы задать имя и юзернейм для вашего бота.
 - BotFather выдаст вам **API токен**. Сохраните его в надежном месте — это ключ к управлению вашим ботом. [7]
 2. **Настройка окружения:**
 - Установите Python на ваш компьютер.
 - Создайте папку для проекта и установите библиотеку командой: `pip install python-telegram-bot`. [6]
 3. **Создание главного меню:**
 - В коде вашего бота создайте главное меню с кнопками: «Пройти бесплатное обучение», «ИИ-психолог», «Полезные инструменты», «Бесплатный ChatGPT», «Поддержка». Для этого используется объект `ReplyKeyboardMarkup`. [8]
 - Напишите обработчики для каждой кнопки. Например, при нажатии на «Пройти бесплатное обучение» бот должен отправить сообщение со ссылкой на вашего **Gem Bot 1**. При нажатии на «ИИ-психолог» — ссылку на вашего **кастомного GPT**.

4. Реализация воронки и ручной верификации:

- **Глубокая ссылка (Deep Link):** Сначала нужно сгенерировать ссылку, которую будет выдавать Gem Bot 1. Она будет иметь вид https://t.me/YourBotUsername?start=trial_completed. [9, 10] Вставьте эту ссылку в инструкции для Gem Bot 1.
- **Обработка ссылки:** В коде бота напишите специальный обработчик для команды /start, который будет проверять наличие параметра trial_completed. [11]
- **Сценарий после пробного периода:** Если пользователь пришел по этой ссылке, бот должен отправить ему поздравительное сообщение и инструкции по монетизации: «Чтобы получить доступ к полному курсу, зарегистрируйтесь на бирже X по нашей реферальной ссылке и пополните баланс. После этого пришлите сюда свой ID пользователя на бирже для проверки».
- **Процесс ручной проверки:**
 - Когда пользователь присыпает свой ID, бот должен переслать это сообщение вам (или в специальный чат для администраторов).
 - Вы вручную заходите в свой партнерский кабинет на бирже и проверяете, действительно ли этот пользователь зарегистрировался по вашей ссылке.
 - Если все верно, вы используете специальную администраторскую команду в боте, например, /approve ID_пользователя.
 - Бот, получив эту команду, меняет статус пользователя в своей базе данных на «approved».
 - Теперь, когда этот пользователь обращается к боту, бот видит его новый статус и предоставляет ему ссылку на **Gem Bot 2** с полным курсом.

5. Раздел «Полезные инструменты»:

- Для этого раздела используйте InlineKeyboardMarkup, чтобы создавать аккуратные «карточки» с кнопками под каждым описанием инструмента. [8]
- Кнопка «Вideoобзор» будет вести на ваше видео в YouTube.
- Кнопка «Перейти на сайт» будет содержать вашу реферальную ссылку на инструмент. [12, 13]

Фаза 2: Автоматизация и партнерство

Цель: Убрать ручной труд из процесса верификации, что позволит масштабировать проект. Эта фаза начинается после того, как MVP доказал свою жизнеспособность.

- **Действие:**

1. **Исследование API бирж:** Начните изучать документацию для разработчиков на сайтах криптобирж, с которыми вы сотрудничаете. Ищите разделы, посвященные партнерской программе (Affiliate/Partner API). [14, 15]
2. **Прямой контакт:** Скорее всего, публичного API для отслеживания рефералов не будет. Напишите напрямую в отдел по работе с партнерами биржи. Объясните вашу бизнес-модель и запросите техническую возможность (API-доступ) для автоматической проверки регистраций и депозитов по вашему реферальному коду.

3. **Рассмотрение сторонних платформ:** Если биржи не предоставляют прямого API, изучите сторонние платформы для отслеживания аффилиатов, такие как **Volumum** или **PartnerMatrix**. [16, 17] Некоторые из них начинают предлагать решения для отслеживания в Telegram. [17]
4. **Техническая реализация:** Когда вы получите доступ к API (напрямую или через платформу), ваш разработчик должен будет заменить ручной процесс на автоматический. Бот будет периодически (например, раз в час) отправлять запросы к API, чтобы проверить статус пользователей, ожидающих подтверждения. При получении успешного ответа, бот автоматически обновит статус пользователя в базе данных и отправит ему доступ к полному курсу.

Фаза 3: Расширение – AI Analyst в формате Telegram Mini App (TMA)

Цель: Создать уникальный, высокоценный продукт, который повысит удержание пользователей и станет вашим ключевым конкурентным преимуществом.

- **Технологии:** ТМА — это, по сути, веб-сайт, который открывается внутри Telegram. Для его создания используются стандартные веб-технологии: **HTML**, **CSS**, **JavaScript** и фреймворки вроде **React** или **Vue**. [18, 19]
- **Действия по разработке:**
 1. **Создание веб-приложения (Frontend):**
 - Разработайте интерфейс вашего аналитического инструмента. Он будет включать графики, таблицы и поля для ввода данных.
 - Для отображения интерактивных финансовых графиков используйте специализированные библиотеки, например, **Highcharts** (платная, но очень мощная) или **Chart.js** (бесплатная, для более простых задач). [20, 21]
 - Интегрируйте в код вашего веб-приложения официальный SDK от Telegram (`telegram-web-app.js`), чтобы приложение могло взаимодействовать с интерфейсом мессенджера. [22]
 2. **Создание серверной части (Backend):**
 - Вам понадобится сервер, который будет поставлять данные для вашего ТМА.
 - **Безопасность — превыше всего.** Когда пользователь открывает ваше ТМА, Telegram передает в него специальную строку `initData`. Ваше веб-приложение должно отправлять эту строку на ваш бэкенд с каждым запросом. Бэкенд **обязан** проверять криптографическую подпись этой строки, используя ваш токен бота. Это единственный способ убедиться, что запрос пришел от реального пользователя из Telegram, а не от злоумышленника. [23, 24, 25]
 3. **Интеграция с Telegram:**
 - Загрузите ваше готовое веб-приложение на любой хостинг с поддержкой HTTPS.
 - Снова обратитесь к **@BotFather**, используйте команду `/myapps` (или `/setmenubutton`), чтобы привязать URL вашего веб-приложения к кнопке меню в SferaTC Bot. [26]
 4. **Реализация данных в реальном времени:**

- Для отображения рыночных данных в реальном времени используйте технологию **WebSockets**. Это позволит вашему серверу мгновенно отправлять обновления на клиент (в ТМА), как только они появляются, без необходимости постоянно перезагружать страницу. [27, 28]

Этот пошаговый план поможет вам двигаться от простой, но работающей идеи к сложной и многофункциональной платформе, проверяя гипотезы и минимизируя риски на каждом этапе.