

Архитектурный и стратегический план для образовательной экосистемы SferaTC

Краткое резюме

Данный документ представляет собой всесторонний технический и стратегический анализ предлагаемой образовательной экосистемы, ориентированной на обучение криптовалютному трейдингу. Архитектура платформы состоит из трех ключевых компонентов: двух кастомизированных чат-ботов («Gems»), созданных непосредственно на платформе Google Gemini («Gem Bot 1» для пробного периода и «Gem Bot 2» для полного курса), центрального Telegram-бота («SferaTC Bot») для управления пользовательским путем и монетизации, а также перспективного аналитического инструмента в формате Telegram Mini App («AI Analyst») для расширения функциональности.

Основная стратегическая оценка подтверждает инновационность бизнес-модели, основанной на «freemium» образовательной воронке, где обучение вынесено на внешнюю платформу Gemini. Центральной технической и коммерческой задачей для Telegram-бота SferaTC является надежное отслеживание регистраций и пополнений счетов на партнерских биржах по реферальным ссылкам, что составляет ядро механизма монетизации.

Ключевые архитектурные рекомендации включают использование библиотеки python-telegram-bot для разработки SferaTC Bot, применение микросервисного подхода для управления интеграциями с различными API искусственного интеллекта (ИИ) и реализацию безопасной, валидируемой на стороне бэкенда архитектуры для будущего Mini App.

Предлагается поэтапный план развития: последовательное создание SferaTC Bot и двух «Gems» на платформе Gemini, внедрение временной системы ручной верификации рефералов в Telegram-боте и, наконец, разработка продвинутого инструмента AI Analyst TMA. Такой подход позволяет минимизировать риски и валидировать бизнес-модель на ранних этапах.

Часть I: Воронка привлечения и конверсии пользователей

Раздел 1: SferaTC Bot и Gem Bot 1 – Начало пути и квалификация

Этот раздел детализирует архитектуру двухэтапной воронки привлечения, которая начинается в Telegram, переходит на платформу Gemini для обучения и возвращается в Telegram для монетизации.

1.1. Этап 1: SferaTC Bot как точка входа

Пользовательский путь начинается в Telegram-боте SferaTC. В главном меню будет кнопка «Пройти бесплатное обучение». При нажатии на эту кнопку SferaTC Bot не начинает обучение, а выполняет единственную функцию: предоставляет пользователю прямую ссылку на **Gem Bot 1**, который является кастомизированным чат-ботом, созданным и работающим на ресурсах Gemini. Таким образом, Telegram-бот выступает в роли шлюза, направляя трафик на первую ступень образовательной воронки.

1.2. Этап 2: Gem Bot 1 (пробный курс) – механизм вовлечения

Gem Bot 1 — это не отдельное веб-приложение, а кастомизированная версия Gemini, известная как «Gem». Такие боты создаются непосредственно в интерфейсе Google (например, в Gemini App или Google AI Studio), что позволяет настроить их поведение без написания кода.

- **Техническая основа:** Создание «Gem» заключается в определении его роли и инструкций через текстовые промпты. Вы задаете его персону (например, «Ты — опытный преподаватель по трейдингу»), определяете его задачи и ограничения. Платформа Gemini сама обрабатывает эти инструкции, создавая специализированного чат-бота. Это значительно упрощает разработку, так как не требует создания и хостинга собственного веб-приложения.
- **Педагогическая структура:** Сохраняется четырехэтапная структура урока (Теория, Вопросы и ответы, Игра, Домашнее задание), как и было предложено. Эта структура будет реализована через промпт-инжиниринг. Например, для интерактивной игры вы дадите «Gem» инструкцию: «Проведи с пользователем текстовую игру-симуляцию для закрепления теории. Предложи сценарий и три варианта ответа. Оцени выбор пользователя и дай обратную связь».
- **Завершение и переход:** После успешной сдачи домашнего задания к третьему уроку, Gem Bot 1, следуя заложенным в него инструкциям, информирует пользователя о необходимости вернуться в Telegram-бот SferaTC для получения дальнейших указаний и предоставляет ему уникальную глубокую ссылку (deep link).

Раздел 2: Переход и монетизация – от Gemini к Telegram и обратно

Этот раздел посвящен критически важным механизмам передачи пользователя между платформами для обеспечения бесшовного пути и монетизации.

2.1. Механизм 1: Глубокие ссылки (Deep Linking) для возврата в Telegram

После завершения пробного курса в Gem Bot 1, пользователь должен быть без проблем перенаправлен обратно в SferaTC Bot. Для этого используется функциональность глубоких ссылок Telegram.

- **Реализация:** Gem Bot 1 генерирует URL вида `https://t.me/SferaTC_Bot?start=payload`. payload будет уникальным, криптографически защищенным токеном (например, JWT), содержащим идентификатор пользователя и подтверждение завершения пробного периода. Это гарантирует, что только пользователи, прошедшие первые три урока, смогут получить доступ к следующему этапу воронки.

- **Пользовательский опыт:** При переходе по ссылке, SferaTC Bot немедленно обработает payload, идентифицирует пользователя и вместо стандартного приветствия выведет специальное сообщение с поздравлениями и мотивационным текстом. Сразу после этого бот представит инструкции для следующего шага: регистрации на бирже.

2.2. Механизм 2: Реферальный шлюз в SferaTC Bot

Это ядро бизнес-модели. SferaTC Bot должен проверить, что пользователь зарегистрировался и пополнил счет на бирже по реферальной ссылке. Как и в первоначальном анализе, **рекомендуется начать с ручной верификации (Метод D)**, чтобы быстро запустить проект и проверить гипотезу, параллельно работая над автоматизацией через API бирж (Метод A) или сторонние платформы (Метод B).

2.3. Механизм 3: Предоставление доступа к полному курсу

После успешной верификации SferaTC Bot выполняет свою последнюю функцию в образовательной воронке: предоставляет пользователю ссылку на **Gem Bot 2**. Этот второй «Gem», также созданный на платформе Gemini, будет содержать остальные 27 уроков полного курса. Ссылка на Gem Bot 2 будет доступна только верифицированным пользователям через тот же скрытый раздел в SferaTC Bot, в который они попали после пробных уроков.

Часть II: Ядро экосистемы – Telegram-бот SferaTC

В этой части подробно рассматривается архитектура SferaTC Bot как центрального хаба для всех сервисов, за исключением самого процесса обучения.

Раздел 3: Онбординг и монетизация – реферальный шлюз

Этот раздел посвящен самому сложному и жизненно важному компоненту бизнес-модели: проверке того, что пользователь успешно зарегистрировался и пополнил счет на бирже по реферальной ссылке.

3.1. Анализ методологий отслеживания рефералов

- **Метод А: Прямые API криптовалютных бирж:**
 - **Концепция:** Программно запрашивать API биржи для подтверждения регистрации нового пользователя и депозита, связанных с реферальным кодом проекта.
 - **Осуществимость:** Крайне сложная. Хотя биржи предоставляют обширные API для торговли, рыночных данных и управления счетами, они **редко предоставляют публичные эндпоинты для отслеживания реферальных регистраций или статуса пополнения**. Эта функциональность обычно зарезервирована для внутреннего использования или для крупных официальных партнеров. Доступ, скорее всего, потребует переговоров о специальной API-интеграции напрямую с отделом развития бизнеса биржи.

- **Метод В: Сторонние платформы для отслеживания аффилиатов:**
 - **Концепция:** Использовать профессиональную платформу для аффилированного маркетинга (например, Voluum, Scaleo, PartnerMatrix), которая интегрируется с криптовалютными биржами. Эти платформы предоставляют надежные API для отслеживания конверсий.
 - **Осуществимость:** Более реалистично, чем Метод А. Платформы, такие как PartnerMatrix, активно разрабатывают функции отслеживания, специфичные для Telegram. Рабочий процесс будет включать: 1) Бот предоставляет пользователю уникальную ссылку для отслеживания, сгенерированную аффилиатной платформой. 2) Платформа отслеживает клик и последующую конверсию (регистрацию/депозит) на стороне биржи. 3) Бэкенд бота опрашивает API аффилиатной платформы для проверки статуса конверсии для этого пользователя. Это влечет за собой дополнительные расходы на сервис, но предлагает более стандартизированное и потенциально надежное решение.
- **Метод С: Встроенная партнерская программа Telegram API:**
 - **Концепция:** Использовать встроенную функциональность партнерской программы Telegram.
 - **Осуществимость:** **Неприменимо к данному случаю.** API партнерской программы Telegram предназначен для разработчиков Mini Apps для начисления комиссий за покупки, совершенные с помощью **Telegram Stars** *внутри* экосистемы Telegram. Он не может отслеживать внешние действия, такие как регистрация и пополнение счета фиатной валютой/криптовалютой на стороннем веб-сайте биржи.
- **Метод D: Ручная или полуавтоматическая верификация (прагматичный первый шаг):**
 - **Концепция:** В качестве временного решения потребовать от пользователя предоставить доказательство выполнения. Это может быть его новый ID пользователя на бирже или скриншот пополненного счета. Сотрудник поддержки будет вручную проверять это и предоставлять доступ.
 - **Осуществимость:** 100% осуществимо и является рекомендуемым подходом для запуска. Это отделяет разработку платформы от длительного и неопределенного процесса заключения технических партнерств. Это создает операционную нагрузку, но позволяет проверить бизнес-модель перед тем, как вкладывать значительные средства в сложные и, возможно, недоступные API-интеграции.

3.2. Предлагаемое решение и реализация

Рекомендация: Начать с **Метода D (Ручная верификация)** для первоначального запуска, чтобы доказать жизнеспособность бизнес-модели. Параллельно следует работать над **Методом А (Прямое партнерство)** с целевыми биржами в качестве долгосрочной цели. Если прямые партнерства окажутся недостижимыми, следует перейти к оценке **Метода В (Сторонние платформы)**.

Рабочий процесс (ручной):

1. SferaTC Bot предоставляет пользователю реферальную ссылку.
2. Затем он просит пользователя ввести свой уникальный ID с биржи после регистрации и пополнения счета.

3. Эта заявка помечается для администратора.
4. Администратор проверяет реферала в партнерской панели биржи.
5. Администратор выполняет команду в боте (например, /approve_user <user_id>), которая обновляет статус пользователя в базе данных, предоставляя ему доступ к разделу со ссылкой на Gem Bot 2.

3.3. Ключевой результат: Таблица 1 – Сравнение методологий отслеживания рефералов

Характеристика	Метод А: Прямой API биржи	Метод В: Сторонняя платформа	Метод С: Встроенный API Telegram	Метод D: Ручная верификация
Сложность реализации	Очень высокая (требуется партнерство)	Высокая (интеграция API с платформой)	Н/П (неприменимо)	Низкая (рабочий процесс администратора)
Надежность отслеживания	Очень высокая (первоисточник)	Высокая (зависит от платформы)	Н/П	Средняя (возможна человеческая ошибка)
Структура затрат	Потенциально нулевая (бонус партнерства)	Ежемесячная подписка + % от дохода	Н/П	Время персонала
Детализация данных	Высокая (полные данные пользователя)	Средняя (данные о конверсии)	Н/П	Низкая (бинарное одобрение)
Масштабируемость	Высокая	Высокая	Н/П	Очень низкая
Время выхода на рынок	Очень долгое / Неопределенное	Среднее	Н/П	Немедленное
Рекомендация	Долгосрочная цель	Жизнеспособная альтернатива	Неприменимо	Стратегия запуска

Раздел 4: SferaTC Bot – архитектура функций и стратегия меню

Этот раздел описывает структуру и реализацию дополнительных функций SferaTC Bot.

4.1. Разработка ядра бота

Технология: Библиотека python-telegram-bot является зрелым и хорошо документированным выбором, предлагающим надежные функции для обработки команд, колбэков и обновлений. Проект будет структурирован вокруг объекта Application с различными обработчиками (CommandHandler, CallbackQueryHandler, MessageHandler).

Архитектура меню:

- Главное меню будет содержать кнопки «Пройти бесплатное обучение», «ИИ-психолог», «Полезные инструменты», «Бесплатный ChatGPT» и «Поддержка». Оно будет реализовано с использованием ReplyKeyboardMarkup.
- Раздел «Полный интерактивный курс» будет скрыт и доступен только по глубокой ссылке из Gem Bot 1, а затем, после верификации, будет содержать ссылку на Gem

Bot 2.

- Раздел «Полезные инструменты» будет использовать InlineKeyboardMarkup с InlineKeyboardButton для каждого инструмента, что позволит создать чистое, карточное представление с кнопками «Видеообзор» и «Перейти на сайт».

4.2. Функция: ИИ-психолог

Реализация: Данная функция будет реализована не внутри SferaTC Bot, а путем перенаправления пользователя на внешний ресурс. В главном меню бота будет кнопка «Бесплатный ИИ психолог». При нажатии на эту кнопку бот предоставит пользователю прямую ссылку на уже существующего кастомного бота, который разработан и размещен в ChatGPT Store.

Архитектурное преимущество: Такой подход полностью выносит сложную логику диалога и управления персоной ИИ на стороннюю платформу (OpenAI), что значительно упрощает архитектуру SferaTC Bot. Ответственность за инжиниринг промптов, этические ограничения и поддержание диалога ложится на кастомного GPT-бота, в то время как SferaTC Bot выполняет лишь функцию шлюза.

4.3. Функция: Полезные инструменты для трейдинга

Структура контента: Этот раздел будет представлять собой курируемый список партнерских продуктов. Каждый инструмент будет представлен в виде «карточки», состоящей из изображения, краткого описания и двух инлайн-кнопок.

Оптимизация конверсии: Презентация будет соответствовать лучшим практикам аффилированного маркетинга в приложениях. Описания будут сосредоточены на решении проблемы пользователя (например, «Устали вручную вести журнал сделок? Этот журнал автоматизирует это.»), а не просто на перечислении функций. Построение доверия имеет первостепенное значение; описания должны быть честными и прозрачными, а видеообзоры — предоставлять реальную ценность.

Обработка ссылок: Кнопки «Перейти на сайт» будут содержать ваши реферальные ссылки. Кнопки «Видеообзор» будут ссылаться на ваш YouTube-канал, создавая синергию контента, которая способствует как партнерскому доходу, так и росту канала.

4.4. Функции: Бесплатный ChatGPT и Поддержка

Реализация: Эти функции могут быть реализованы на той же базовой большой языковой модели, что и ИИ-психолог, но с другим, более общим системным промптом. Промпт для бота поддержки будет спроектирован так, чтобы быть полезным для вопросов, связанных с платформой, и предоставлять четкий путь к эскалации на человека, если он не сможет решить проблему.

Часть III: Перспективное видение – Telegram Mini App (TMA) «AI Analyst»

Эта часть представляет собой стратегический и технический план разработки высокоценного инструмента «AI Analyst», использующего передовые возможности Telegram Mini Apps. Она остается неизменной, так как не затронута уточнениями

архитектуры.

Раздел 5: От бота к Mini App – стратегический и технический план

5.1. Обоснование использования ТМА

Стандартный интерфейс чат-бота ограничен текстом и простыми кнопками. Инструмент «AI Analyst» требует богатого, интерактивного пользовательского интерфейса для визуализации данных (диаграммы, графики), ввода сложных данных и опыта, подобного нативному приложению. ТМА, которые по сути являются веб-приложениями, работающими в WebView внутри Telegram, являются идеальной технологией для этого. ТМА создаются с использованием стандартных веб-технологий (HTML, CSS, JavaScript/TypeScript) и могут быть развернуты на любом сервере с поддержкой HTTPS. Для создания сложного пользовательского интерфейса настоятельно рекомендуется использовать современный фронтенд-фреймворк, такой как React или Vue.

5.2. Процесс настройки и развертывания

ТМА регистрируется и связывается с SferaTC Bot через BotFather с помощью команд `/newapp` или `/myapps` для установки его URL-адреса. Его можно запустить через кнопку меню (`/setmenubutton`) или по прямой ссылке (`t.me/SferaTC_Bot/analyst_app`). Фронтенд-приложение должно включать Telegram Web App SDK (`telegram-web-app.js` или npm-пакет `@twa-dev/sdk`). Этот SDK является мостом, который позволяет веб-приложению взаимодействовать с клиентом Telegram (например, для получения цветов темы, данных пользователя или закрытия приложения).

5.3. Критический компонент: Безопасная аутентификация пользователя

Когда пользователь запускает ТМА, клиент Telegram передает специальный набор параметров запуска, включая строку под названием `initData` (или `tgWebAppData`). Эта строка содержит информацию о пользователе и, что особенно важно, криптографическую подпись. Фронтенд ТМА должен отправлять эту строку `initData` на свой собственный бэкенд-сервер с каждым запросом API (например, в заголовке `Authorization`). Бэкенд-сервер **обязан** проверять подпись в этой строке `initData`, используя секретный токен бота. Этот процесс включает в себя специфический алгоритм хеширования HMAC-SHA256. Существуют библиотеки для Node.js (`@telegram-apps/init-data-node`) и Go (`init-data-golang`), которые упрощают этот процесс. Эта проверка является **единственным** способом для бэкенда убедиться, что запрос исходит от легитимного пользователя, запустившего приложение через Telegram. Без этого любой мог бы отправлять поддельные запросы на ваши эндпоинты API. Это краеугольный камень безопасности ТМА.

5.4. Двухнаправленная коммуникация

ТМА может отправлять данные обратно боту, который его запустил, используя метод `Telegram.WebApp.sendData()`. Это полезно для простых взаимодействий, когда приложению нужно инициировать действие в чате бота. Для обновлений в реальном времени от бота/бэкенда к приложению (например, для отправки нового анализа)

требуется прямой канал связи, такой как WebSockets, что подробно описано в следующем разделе.

Раздел 6: Создание «AI Analyst» – расширенные возможности и UI/UX

6.1. Поток данных в реальном времени с помощью WebSockets

Аналитический инструмент требует данных с рынка в реальном времени. Постоянный опрос сервера через HTTP неэффективен и медленен. WebSockets предоставляют постоянный, двунаправленный канал связи между клиентом TMA и бэкенд-сервером. JavaScript-клиент TMA будет создавать WebSocket-соединение с бэкендом (`const websocket = new WebSocket("wss://your-backend-url/")`). Затем бэкенд сможет отправлять данные (например, новые тики цен, торговые сигналы) клиенту в реальном времени. Клиент будет использовать обработчики событий (`onopen`, `onmessage`, `onerror`) для управления соединением и входящими данными. Использование защищенных WebSocket-соединений (`wss://`) является обязательным. Реализация должна корректно обрабатывать обрывы соединения и переподключения для обеспечения надежного пользовательского опыта.

6.2. Визуализация данных: Выбор библиотеки для построения графиков

Отрисовка сложных, интерактивных финансовых графиков в мобильном WebView требует библиотеки, которая является одновременно мощной и легковесной для обеспечения плавной работы.

- **Анализ вариантов:**

- **Chart.js:** Открытый исходный код, очень популярен. Хорошая производительность и простота в начале работы. Может потребовать дополнительной настройки для сложных финансовых графиков.
- **Highcharts:** Коммерческая библиотека, известная отличной производительностью, обширной документацией и специальными функциями для биржевых графиков (Highcharts Stock). Очень сильный кандидат для этого случая, несмотря на плату за лицензию.
- **D3.js:** Мощная, низкоуровневая библиотека, предлагающая непревзойденную гибкость для создания уникальных визуализаций. Однако у нее очень крутая кривая обучения, и это скорее графический инструмент, чем библиотека для построения графиков.
- **AnyChart:** Еще один надежный коммерческий вариант с конкретными продуктами для биржевых (AnyStock) и временных данных, что делает его прямым конкурентом Highcharts.

6.3. Ключевой результат: Таблица 2 – Анализ JavaScript-библиотек для построения графиков в AI Analyst TMA

Библиотека	Производительность / Легковесность	Простота интеграции	Функции для финансовых графиков	Мобильная адаптивность	Лицензия	Рекомендации
Chart.js	Отличная	Очень высокая	Базовые (расширяемые)	Высокая	Open-Source (MIT)	Подходит для простых, стандартных графиков.
Highcharts	Отличная	Высокая	Отличные (встроенные)	Отличная	Коммерческая	Настоятельно рекомендуется для серьезной финансовой визуализации.
AnyChart	Очень хорошая	Высокая	Отличные (встроенные)	Отличная	Коммерческая	Сильная альтернатива Highcharts.
D3.js	Зависит от реализации	Низкая	Н/П (создание с нуля)	Зависит от реализации	Open-Source (ISC)	Избыточно, если не требуются уникальные визуализации.

Заключение и итоговые стратегические рекомендации

- **Стратегия поэтапного внедрения:**
 1. **Фаза 1 (MVP):** Разработать SferaTC Bot и два отдельных «Gem» бота (для пробного и полного курсов) на платформе Gemini. В SferaTC Bot реализовать систему ручной верификации рефералов. Запустить и проверить всю воронку: от первого контакта в Telegram до монетизации и предоставления доступа к полному курсу.
 2. **Фаза 2 (Автоматизация и партнерство):** Пока Фаза 1 активна, активно работать над техническими партнерствами с биржами для автоматизации отслеживания рефералов. В случае неудачи интегрировать стороннюю аффилиатную платформу.
 3. **Фаза 3 (Расширение):** Начать разработку AI Analyst TMA. Запустить его как премиум-функцию для полностью конвертированных пользователей, добавляя значительную ценность экосистеме и оправдывая долгосрочное удержание пользователей.
- **Снижение критических рисков:** Основной риск заключается в зависимости от внешних бирж для механизма монетизации. Поэтапный подход с начальной ручной верификацией является ключевой стратегией для снижения этого риска, позволяя

бизнесу генерировать доход и доказывать свою модель, работая над долгосрочным автоматизированным решением.

- **Политика безопасности:** Необходимо придерживаться принципа «безопасность прежде всего». Это включает безопасное обращение с ключами API (использование переменных окружения или менеджеров секретов), надежную проверку всех вводимых пользователем данных и критически важную реализацию проверки подписи initData для бэкенда ТМА.
- **Итоговое архитектурное видение:** Конечным состоянием является сложная, многокомпонентная цифровая экосистема. Модульный, микросервисный бэкенд является оптимальной архитектурой для поддержки этого видения, обеспечивая масштабируемость, поддерживаемость и гибкость для развития платформы с течением времени.

Источники

1. Build custom experts with Gems - Google Gemini, <https://gemini.google/overview/gems/> 2. How to Create Your Own Custom AI Bots in Google Gemini | Lifehacker, <https://lifehacker.com/tech/how-to-create-your-own-custom-ai-bots-in-google-gemini> 3. Google AI Studio quickstart - Gemini API, <https://ai.google.dev/gemini-api/docs/ai-studio-quickstart> 4. Your own customized chatbot with Google Gemini API - Folder IT, <https://folderit.net/your-own-customized-chatbot-with-google-gemini-api/> 5. Telegram Bot Features, <https://core.telegram.org/bots/features> 6. Deep links - Telegram APIs, <https://core.telegram.org/api/links> 7. Deep Linking - aiogram 3.16.0 documentation, https://docs.aiogram.dev/en/v3.16.0/utils/deep_linking.html 8. Deep linking — aiogram 2.25.1 documentation, https://docs.aiogram.dev/en/v2.25.1/utils/deep_linking.html 9. Passing parameters to a Telegram bot via URL - Latenode community, <https://community.latenode.com/t/passing-parameters-to-a-telegram-bot-via-url/8817> 10. Best Top 10 Crypto APIs for Developers in 2025 - Apidog, <https://apidog.com/blog/best-top-10-crypto-apis/> 11. Can I Use a Crypto API to Track My Portfolio?, https://www.tokenmetrics.com/blog/crypto-apis-portfolio-tracking?0fad35da_page=31&74e29fd5_page=49 12. API Integration for Crypto Exchanges: What You Need to Know - Debut Infotech, <https://www.debutinfotech.com/blog/api-integration-for-crypto-exchanges> 13. Affiliate Tracking Software - Voluum | Voluum – Ad Tracker for Media Buyers, <https://voluum.com/affiliate-marketing-software/> 14. 25 Free & Affordable Affiliate Tracking Software Worth Using (2025 Update) - Scaleo, <https://www.scaleo.io/blog/best-25-free-affiliate-tracking-software/> 15. PartnerMatrix Breaks New Ground with Telegram Affiliate Tracking That Reveals Hidden Revenue Streams - Affiverse, <https://www.affiversemedia.com/partnermatrix-breaks-new-ground-with-telegram-affiliate-tracking-that-reveals-hidden-revenue-streams/> 16. Affiliate programs - Telegram APIs, <https://core.telegram.org/api/bots/referrals> 17. Building Telegram Bot with Python-Telegram-Bot: A Comprehensive Guide - Medium, <https://medium.com/@moraneus/building-telegram-bot-with-python-telegram-bot-a-comprehensive-guide-7e33f014dc79> 18. python-telegram-bot Documentation, <https://media.readthedocs.org/pdf/python-telegram-bot/latest/python-telegram-bot.pdf> 19. python-telegram-bot v22.4, <https://docs.python-telegram-bot.org/> 20. python-telegram-bot/python-telegram-bot: We have made you a wrapper you can't refuse - GitHub, <https://github.com/python-telegram-bot/python-telegram-bot> 21. deeplinking.py -

python-telegram-bot v22.1,
<https://docs.python-telegram-bot.org/en/v22.1/examples.deeplinking.html> 22. From BotFather to 'Hello World' - Telegram APIs, <https://core.telegram.org/bots/tutorial> 23. Creating a custom menu in a Telegram bot: How to implement buttons like this?, <https://community.latenode.com/t/creating-a-custom-menu-in-a-telegram-bot-how-to-implement-buttons-like-this/8430> 24. How to Add a Menu to a Telegram Bot Using Telethon Library - Stack Overflow,
<https://stackoverflow.com/questions/75084490/how-to-add-a-menu-to-a-telegram-bot-using-telethon-library> 25. How to increase affiliate conversions: 5 research-backed strategies for content creators (2025) - Impact, <https://impact.com/influencer/how-to-increase-affiliate-conversions/> 26. In-App Affiliate Marketing: A Beginner's Guide - wecantrack,
<https://wecantrack.com/insights/in-app-affiliate-marketing/> 27. How Affiliate Marketing Can Help Increase Conversion Rates,
<https://www.refersion.com/blog/increase-conversion-with-affiliate-marketing/> 28. 8 Chatbot Best Practices and Tips - Talkdesk, <https://www.talkdesk.com/blog/chatbot-best-practices/> 29. Telegram Mini Apps Creation Handbook - DEV Community,
https://dev.to/simplr_sh/telegram-mini-apps-creation-handbook-58em 30. Telegram Mini App Development — Ultimate Guide 2025 - EJAW,
<https://ejaw.net/telegram-mini-app-development-2025/> 31. About the Platform | Telegram Mini Apps, <https://docs.telegram-mini-apps.com/platform/about> 32. Telegram Mini Apps: How They Work and Their Benefits - Umnico, <https://umnico.com/blog/telegram-mini-apps-guide/> 33. Step-by-step guide | The Open Network - TON Docs,
<https://docs.ton.org/v3/guidelines/dapps/tma/tutorials/step-by-step-guide> 34. Creating New App | Telegram Mini Apps, <https://docs.telegram-mini-apps.com/platform/creating-new-app> 35. Init Data | Telegram Mini Apps, <https://docs.telegram-mini-apps.com/platform/init-data> 36. Telegram Mini Apps, <https://core.telegram.org/bots/webapps> 37. Authorizing User | Telegram Mini Apps, <https://docs.telegram-mini-apps.com/platform/authorizing-user> 38. Telegram-Mini-Apps/init-data-golang - GitHub,
<https://github.com/Telegram-Mini-Apps/init-data-golang> 39. Server-Side Setup for Telegram Mini App with Web3Auth, <https://web3auth.io/docs/guides/telegram-miniapp-server> 40. Build a Telegram Mini App Using Pinata's Files API - No Database!,
<https://pinata.cloud/blog/build-a-telegram-mini-app-using-pinatas-files-api-no-database/> 41. Publishing Mini Apps | The Open Network - TON Docs,
<https://docs.ton.org/v3/guidelines/dapps/tma/guidelines/publishing> 42. Telegram Mini App: can't transfer data from mini app to bot - Stack Overflow,
<https://stackoverflow.com/questions/79399811/telegram-mini-app-cant-transfer-data-from-mini-app-to-bot> 43. MiniApp - Telegram Mini Apps,
<https://docs.telegram-mini-apps.com/packages/telegram-apps-sdk/1-x/components/mini-app> 44. The WebSocket API (WebSockets) - Web APIs - MDN,
https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API 45. WebSocket - The Modern JavaScript Tutorial, <https://javascript.info/websocket> 46. Writing WebSocket client applications - Web APIs - MDN,
https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_client_applications 47. 6 Best JavaScript Charting Libraries for Dashboards in 2025 - Embeddable,
<https://embeddable.com/blog/javascript-charting-libraries> 48. Chart.js | Open source HTML5 Charts for your website, <https://www.chartjs.org/> 49. Highcharts - Interactive Charting Library for Developers, <https://www.highcharts.com/> 50. D3 by Observable | The JavaScript library for bespoke data visualization, <https://d3js.org/> 51. AnyChart is a lightweight and robust JavaScript

charting library, <https://www.anychart.com/>