Bitties FAQ

Top of Form

1. What is the "Bitties" app and its core purpose?

The "Bitties" app is a mobile and web application designed to help a group of friends who have been jointly investing in Bitcoin since 2022. Its primary purpose is to provide live updates on their shared Bitcoin fund, display real-time charts, and offer individual member tabs. The app aims to track their collective progress towards a specific savings goal, which is a trip to the Masters Tournament in Augusta in 2036. It will facilitate tracking contributions, managing members, and providing insights into their Bitcoin portfolio.

2. What are the key visual design and accessibility principles for the "Bitties" app?

The "Bitties" app will adopt a premium and elegant aesthetic, echoing the Masters Tournament's visual style. All text will exclusively use the "Poppins" font. The primary brand colour will be a specific shade of blue (#0656A3), with accent colours including green (#00722D) for success, yellow (#FFB81C) for alerts, and red (#D62718) for errors. Backgrounds will primarily be white (#FFFFFF) with very light grey (#F6F7FB) for surfaces.

Accessibility is a paramount concern, ensuring the app can be fully used by individuals with only one hand and limited typing ability. All interactions, including navigation, data entry, and toggles, will be achievable with single-tap/click controls, dropdowns, and accessible buttons. The user interface will incorporate modern design touches like rounded corners (12-20px), subtle drop-shadows, micro-interactions, and generous spacing for a clean, uncluttered look.

3. How will the "Bitties" app handle data and ensure live updates?

The "Bitties" app will pull current and historical Bitcoin price data (in both ZAR and USD) from public APIs such as CoinGecko and CryptoCompare. It will also retrieve current and projected trip costs for the Augusta 2036 trip (flights, accommodation, tickets, spending) via live web scraping and APIs where possible. All live data will update regularly and automatically to ensure users always have the most current information. The app is designed to be robust against missing, incomplete, or inconsistent data.

4. What are the core features and functionalities of the "Bitties" app?

The "Bitties" app will include several core features:

* **Dual-Currency Dashboard:** Users can instantly switch all charts and values between ZAR and USD.
* **Personal & Group Dashboards:** Provides individual member views and collective group totals, including BTC held, portfolio value, and progress.
* **Trader Analytics:** Offers live and historic BTC charts (candlestick, line, area), contribution/balance graphs, P&L, volatility, drawdown, and various timeframes.
* **Speculation Tool:** A "What-if" simulator for contribution amounts, BTC price, inflation, trip costs, and member count, allowing for instant forecast updates and scenario sharing.
* **Manual BTC Entry:** Authorised persons can add BTC purchases via a single accessible form, which then gets split amongst the pool members.
* **Notifications:** Email and potentially WhatsApp reminders for contributions, milestones, and BTC price alerts.
* **Membership Management:** Tools to easily manage join/leave events, adjust shares fairly, support user roles (admin/member/viewer), and maintain an audit log.
* **Trip Goal Logic:** Auto-tracks and dynamically updates the savings goal based on the latest trip cost data.

5. What is the proposed development approach and technology stack for "Bitties"?

The "Bitties" app can be built using either Python or PowerShell, with the specific choice to be determined. The development will utilise GitHub for version control and Cloudflare if necessary. The build will be modular, with a project file created to manage components. Errors will trigger a version change, and emphasis will be placed on token efficiency for GPT chats and handover briefs. All styling will be organised in a separate module for global implementation and easy modification without altering core app code. The final output of any script will be a single block that can be directly copied and pasted, eliminating manual folder creation or file pasting.

6. How will data for contributions, members, and BTC purchases be formatted for the app?

The project brief states that relevant data regarding contributions, members (new and old), and Bitcoin purchases is available. While the exact formatting is not specified in the provided text, the app's design anticipates that this data will be structured in a way that allows for automated creation, migration, and completion of data structures. The development process will include guidance on how to format this data to ensure seamless integration with the app's modules and functionalities. The app is designed to be robust against missing, incomplete, or inconsistent data.

7. What considerations have been made for the app's extendability and future features?

The "Bitties" app is designed with modularity in mind to facilitate future expansion. The core foundation of the app is built to be extended with additional features such as more interactive analytics, advanced charts, leaderboards, and a robust notification system beyond the initial scope. The modular architecture, particularly the separation of styling and core logic, ensures that new features can be integrated without requiring significant overhauls of existing code. This allows for future additions like more detailed speculation tools or expanded data sources as needed.

8. How will the app ensure a "zero manual setup" and ease of deployment?

A key requirement for "Bitties" is "zero manual setup." This means all data structure creation, migration, and completion will be 100% automated, eliminating the need for users to manually set up or configure anything. The app's design dictates that all actions and logic must be automated and documented, with no prompts for user input during code execution. The goal is for the entire app setup, data migration, and UI deployment to be as close to a "single upload, everything works" process as possible, with the final code provided as a single, copy-pastable block.

Bottom of Form

.