📘 Mudra Tube: Comprehensive Technical Documentation & Developer Guide

1. प्रोजेक्ट परिचय (Project Overview)

Mudra Tube एक आधुनिक, मोबाइल-फर्स्ट **Telegram WebApp** प्लेटफ़ॉर्म है। यह उपयोगकर्ताओं (Users) को बिना किसी पारंपरिक वीडियो विज्ञापन (No Video Ads) के, सीधे **Telegram Channel Join Tasks** और **CPA Offerwalls (App Installs/Surveys)** के माध्यम से पुरस्कार (Coins) अर्जित करने की अनुमति देता है। अर्जित कॉइन्स को एक पारदर्शी विड्रॉल सिस्टम के जरिए **UPI** के माध्यम से सीधे बैंक खाते में ट्रांसफर किया जा सकता है।

मुख्य तकनीकी विशेषताएं (Key Technical Features):

- **Zero Backend Server (Serverless):** पूरी तरह से फ्रंटएंड-फर्स्ट आर्किटेक्चर, जो सीधे **Firebase Firestore** डेटाबेस से संचार करता है।
- **Live Telegram Verification:** Telegram Bot API के getChatMember मेथड का उपयोग करके असली मेंबरशिप चेक करना।
- **Anti-Cheat Mechanics:** completed_tasks एरे लॉजिक जो एक यूजर को एक टास्क से बार-बार कॉइन्स लेने से रोकता है।
- **Dynamic CPA Tracking:** ऑफरवॉल लिंक के साथ यूजर आईडी ऑटो-टैग होना ताकि पोस्टबैक सही यूजर तक पहुँचे।

2. डेटाबेस स्कीमा डिजाइन (Firebase Firestore Schema)

प्रोजेक्ट में **Firebase Firestore (NoSQL)** का उपयोग किया गया है। इसमें मुख्य रूप से दो Collections (कलेक्शन) हैं:

A. users Collection

हर टेलीग्राम यूजर का डेटा उसकी user_id (स्ट्रिंग) के रूप में स्टोर होता है।

- user_id (String): टेलीग्राम से प्राप्त यूनिक यूजर आईडी (उदा. "123456789")।
- username (String): यूजर का टेलीग्राम यूजरनेम या फर्स्ट नेम।
- balance (Number): यूजर के पास वर्तमान में उपलब्ध कुल कॉइन्स (डिफ़ॉल्ट: 0)।
- completed_tasks (Array of Strings): पूरा किए गए टास्क की लिस्ट (उदा. ["channel_join"])।
- created_at (Timestamp): अकाउंट बनने का समय (Firebase Server Timestamp)।

B. withdrawals Collection

जब भी कोई यूजर पैसे निकालने की रिक्वेस्ट करता है, इस कलेक्शन में एक नया डॉक्यूमेंट जोड़ा जाता है।

- user_id (String): विड्रॉल करने वाले यूजर की आईडी।
- username (String): यूजर का नाम।

- upi_id (String): यूजर द्वारा दर्ज की गई यूपीआई आईडी (उदा. user@paytm)।
- coins (Number): काटे गए कुल कॉइन्स (न्यूनतम 300)।
- amount_inr (Number): रुपये में वैल्यू (coins / 300)।
- status (String): रिक्वेस्ट की स्थिति ("pending" / "completed" / "rejected")।
- requested_at (Timestamp): रिक्वेस्ट सबमिट करने का समय।

3. कोड मॉड्यूल की विस्तृत कार्यप्रणाली (Code Modules & Logic)

1. Telegram WebApp Initialization

यह ब्लॉक सुनिश्चित करता है कि ऐप टेलीग्राम के अंदर खुलने पर पूरी स्क्रीन पर फैल जाए और सही यूजर की पहचान (Context) मिल सके:

```
const tg = window.Telegram?.WebApp;
if (tg) { tg.expand(); }
const tgUser = tg?.initDataUnsafe?.user;
const userId = tgUser ? String(tgUser.id) : "test_user_01";
const userName = tgUser ? (tgUser.username || tgUser.first_name) :
"Guest";
```

2. रियल-टाइम डेटा सिंक (loadUserData)

यह फ़ंक्शन Firebase Firestore से रियल-टाइम डेटा सुनता है (onSnapshot)। अगर यूजर नया है, तो डेटाबेस में उसका डॉक्यमेंट ऑटोमैटिकली क्रिएट हो जाता है:

- यूज़र का बैलेंस स्क्रीन पर अपडेट होता है।
- अगर completed_tasks में "channel_join" मौजूद है, तो वेरीफाई बटन लॉक होकर Done (Gray) हो जाता है।

3. चैनल वेरिफिकेशन लॉजिक (verifyChannel)

चीटिंग रोकने के लिए यह सबसे महत्वपूर्ण मॉड्यूल है:

1. यूजर लिंक पर क्लिक करके चैनल पर जाता है।
2. चैनल जॉइन करने के बाद जब वह "Verify" दबाता है, तो ऐप टेलीग्राम बॉट एपीआई को कॉल करता है: https://api.telegram.org/bot<BOT_TOKEN>/getChatMember?chat_id=<CHANNEL_ID>&user_id=<USER_ID>
3. यदि रिस्पॉन्स में स्टेटस member, administrator, या creator आता है, तो Firebase में balance में +50 जोड़े जाते हैं और completed_tasks में टास्क जोड़ दिया जाता है।

4. UPI विड्रॉल सिस्टम (requestWithdrawal)

- नियम: न्यूनतम विड्रॉल सीमा **300 Coins (₹1.00)** है।
- फॉर्मूला: INR = Coins / 300
- रिक्वेस्ट सबमिट होते ही डेटाबेस में एंट्री बनती है और यूजर का बैलेंस रीसेट (0) हो जाता है।

4. स्टेप-बाय-स्टेप डिप्लॉयमेंट गाइड (Step-by-Step

Deployment)

चरण 1: Firebase सेटअप

1. Firebase Console पर नया प्रोजेक्ट बनाएं।
2. **Firestore Database** इनेबल करें।
3. प्रोजेक्ट सेटिंग्स से Web App की firebaseConfig डिटेल्स कॉपी करें।

चरण 2: Telegram Bot सेटअप

1. Telegram पर @BotFather से नया बॉट बनाएं और **Bot Token** प्राप्त करें।
2. अपना चैनल बनाएं और बॉट को उस चैनल का **Administrator** (एडमिन) बनाएं।

चरण 3: कोड कॉन्फ़िगरेशन

index.html फ़ाइल के कॉन्फ़िगरेशन ब्लॉक में अपनी डिटेल्स भरें:

```
const BOT_TOKEN = "YOUR_BOT_TOKEN_HERE";
const CHANNEL_ID = "@your_channel_username";
const CHANNEL_LINK = "https://t.me/your_channel";
const OFFERWALL_LINK = "https://your-cpa-link.com?subid=";
```

चरण 4: होस्टिंग और बॉट मेनू सेटअप

1. इस कोड को **Vercel** या **Netlify** पर डिप्लॉय करें और लाइव URL प्राप्त करें।
2. @BotFather पर जाएं -> /mybots -> अपना बॉट चुनें -> **Bot Settings** -> **Menu Button** पर क्लिक करके अपनी वेब ऐप का लाइव URL सेट करें।

5. सुरक्षा और एंटी-चीट उपाय (Security & Best Practices)

- बॉट एडमिनिस्ट्रेशन: बॉट का चैनल में एडमिन होना अनिवार्य है, अन्यथा getChatMember API एरर देगा।
- क्लाइंट-साइड सुरक्षा: completed_tasks एरे यह सुनिश्चित करता है कि एक ही यूजर एक टास्क को बार-बार क्लिक करके कॉइन्स न लूट सके।
- डेटा प्राइवेसी: कोई भी संवेदनशील पासवर्ड या सीक्रेट डेटा फ्रंटएंड कोड में हार्डकोडेड नहीं होना चाहिए (सिवाय पब्लिक बॉट टोकन के, हालांकि उच्च स्तर पर इसे बैकएंड सर्वर के जरिए सुरक्षित किया जाना चाहिए)।