1. Real-Time Audio Calls:-

- Integration: use Twilio Voice SDK for frontend, and generate access tokens on backend.

- Backend: Track call logs, start/end time, and coin deduction per minute (cron- socket)

- Frontend: show user status (online / Busy) using websocket & call duration Timer.

2. Wallet & KYC system:-

- Wallet DB Schema:

Users (id, name, email, wallet_balance)
Transactions (id, user_id, type, amount, timestamp)
Referral (id, referrer_id, referred_user_id, reward)
KYC (id, user_id, status, doc_type, doc_id)

- Referral logic:

→ On new signup via referal link, credit coins

→ on user sign-up, generate unique referal code and store the user table. (referral Code)

- when a new user signs up with a referral Code, validate code, credit referrer and refree with coins. via transaction table.

- Use a queue (ex: Rabbit MQ) to handle reward processing asynchronously Prevent abuse by limiting referrals per use & verifying via email/phone.

⇒ KYC Verification (Karza/signzy)

To integerate this API for PAN/Adhar verification. Create end point to collect user documents (via secure file upload) send to API and store KycId, status in KYC table.

→ Implement webhook to update KYC status, Restrict withdrawals until status = verified.

3. Performance + Scalability

Top 3 priorities for 1 Lakh + Daily Users

1) Database Optimization : Use connections pooling indexing and caching (Redis) for frequent (eg) wallet balance, game, status). Partition transactions table by date for faster reads.

2. Load Balancing : Deploy app on kubernetes with horizontal pod autoscaling. Use a load Balancer to distribute traffic accross nodes.

3. API Rate Limitting : Use rate limitting to prevent Abuse. optimize endpoints with pagination and batch processing for high-traffic features like game lobbies.

Fast load Times for Tier 2/3 cities.

1. Optimization : Minify JS/CSS use lazy loading for images and implement code spilitting in React JS. Web P images smaller sizes.

2. Network Optimization : Serve assets via CDN with Indian edge servers. Compress API responses. with GZIP. Use HTTP/2 for faster multiplexing.

3. Low-End Device support :- Avoid heavy animations use lightweight libraries & test on low-end devices.

4. Team Collaboration

Key Areas to Audit

→ Code Quality
→ Infrastructure.

→ Doumentation

Handling Tech Dept / Documentation Gaps / Refactoring

→ Tech Debt : prioritize critical issues.

→ Doumentation Gaps : Conduct knowledge transfer sessions with agency.

→ Refactoring : Use linters (ES Lint, prettier) to enforce code style. Write Unit Test critical modules before refactoring.