Contents

Scenario: Design Global Feed for Emerging Markets

Describe your approach to building a global feed optimized for low-bandwidth, high-latency environments.

Purpose

- Tests your ability to design for real-world constraints (low bandwidth, high latency, unreliable connectivity)
- Evaluates how you prioritize user experience and core functionality under technical/resource limitations
- Assesses your skill in making pragmatic trade-offs between quality, performance, and cost
- Checks your awareness of global infrastructure, localization, and offline-first patterns
- Demonstrates your ability to communicate architectural decisions and justify choices clearly

---

Summary

Design a global feed that is resilient to low bandwidth, high latency, and intermittent connectivity. Prioritize lightweight content, aggressive caching, and offline support to ensure a smooth user experience in emerging markets.

Content Strategy

- Prioritize text and low-resolution images over video or rich media
- Use adaptive image compression and lazy loading
- Pre-fetch and cache trending or local content for offline access
- Personalize feed with local language and regionally relevant topics

---

Network Optimization

- Use delta updates and compact payloads (e.g., Protocol Buffers, gzip)
- Support partial feed refresh (only fetch new/changed items)
- Implement background sync and retry logic for unreliable networks
- Use CDN edge nodes close to users for static assets

Data Storage & Caching

- Cache feed data on device (IndexedDB, SQLite, localStorage)
- Use TTL and LRU eviction for local cache
- Store user actions locally and sync when online

Backend & API Design

- Provide paginated, stateless APIs for feed retrieval
- Support batch requests and server-side filtering
- Use region-aware backend infrastructure (multi-region deployment)
- Degrade gracefully: serve cached or static content if backend is unreachable

---

Security & Privacy

- Encrypt data in transit (TLS)
- Respect local privacy laws and user consent for data collection

---

Metrics for Success

- Feed load time and data usage per session
- Offline engagement rates
- Error rates and retry success
- User retention in low-connectivity regions

Trade-offs & Challenges

- Lower media quality for bandwidth savings
- Potential staleness of cached/offline content
- Increased complexity in sync and conflict resolution

Client-Side Considerations

- Use lightweight frontend frameworks and defer non-critical scripts
- Enable low-data mode toggle in app settings
- Implement skeleton loaders for perceived performance
- Optimize for older/low-end Android devices with limited RAM/CPU

Sync & Conflict Resolution

- Use operation-based CRDTs or timestamp-based last-write-wins policies
- Resolve sync conflicts gracefully with user feedback when needed
- Sync changes in the background to reduce disruption

Testing & Validation

- Use network throttling to simulate 2G/3G and airplane mode scenarios
- Test on entry-level devices common in target regions
- Collect metrics via analytics SDKs with minimal overhead

Accessibility & Inclusion

- Ensure text content supports screen readers and proper font scaling
- Design UI with high contrast and offline-friendly fonts
- Provide content in local languages with proper fallback logic