

Mobile Application Developer Assignment

Build a Full-Featured Quote App Using AI Tools

Time estimate: 2-4 hours

Deadline: 2 days from receiving this assignment

The Task

Build **QuoteVault** — a complete quote discovery and collection app with user accounts, cloud sync, and personalization features.

What We're Really Testing

More than coding, this is an AI-tool proficiency test.

We want to see how effectively you use AI tools (Claude Code, Cursor, Copilot, ChatGPT, etc.) to build software. We expect you to finish faster and produce cleaner output by leveraging AI effectively. We would love to see your workflow.

In addition to the feature marks above, we evaluate your AI workflow:

Feature Requirements

1. Authentication & User Accounts (15 marks)

- [] Sign up with email/password
- [] Login/logout functionality
- [] Password reset flow
- [] User profile screen (name, avatar)
- [] Session persistence (stay logged in)

Tech: Supabase Auth

2. Quote Browsing & Discovery (20 marks)

- [] Home feed displaying quotes (paginated or infinite scroll)
- [] Browse quotes by category (minimum 5: Motivation, Love, Success, Wisdom, Humor)
- [] Search quotes by keyword
- [] Search/filter by author
- [] Pull-to-refresh functionality
- [] Loading states and empty states handled gracefully

Tech: Supabase Database (seed with 100+ quotes across categories)

3. Favorites & Collections (15 marks)

- [] Save quotes to favorites (heart/bookmark)
- [] View all favorited quotes in a dedicated screen
- [] Create custom collections (e.g., "Morning Motivation", "Work Quotes")
- [] Add/remove quotes from collections
- [] Cloud sync — favorites persist across devices when logged in

Tech: Supabase Database (userFavorites, collections tables)

4. Daily Quote & Notifications (10 marks)

- [] "Quote of the Day" prominently displayed on home screen
- [] Quote of the day changes daily (server-side or local logic)
- [] Local push notification for daily quote
- [] User can set preferred notification time in settings

Tech: Local notifications (iOS/Android native)

5. Sharing & Export (10 marks)

- [] Share quote as text via system share sheet
- [] Generate shareable quote card (quote + author on styled background)
- [] Save quote card as image to device
- [] At least 3 different card styles/templates to choose from

Tech: Image generation (canvas/view snapshot)

6. Personalization & Settings (10 marks)

- [] Dark mode / Light mode toggle
- [] At least 2 additional themes or accent colors
- [] Font size adjustment for quotes
- [] Settings persist locally and sync to user profile

7. Widget (10 marks)

- [] Home screen widget displaying current quote of the day
- [] Widget updates daily
- [] Tapping widget opens the app to that quote

Tech: iOS WidgetKit or Android App Widget

8. Code Quality & Architecture (10 marks)

- [] Clean project structure (separation of concerns)
 - [] Consistent naming conventions
 - [] No hardcoded strings (use constants/localization)
 - [] Error handling throughout
 - [] README with clear setup instructions
-

Total Marks: 100

Category	Marks
AI tools usage	35
Working app with feature completeness	35
Design tools usage	10
Loom Video	10
Documentation	10
Total	100

Design Requirements

- Use **Figma Make** or **Stitch** (stitch.withgoogle.com) to generate your UI designs
- App should look polished and professional — not default/stock UI
- Consistent design language throughout
- Include link to your designs or screenshots in submission

Technical Requirements

- **Framework:** SwiftUI, React Native, Flutter, or Kotlin — your choice
 - **Backend:** Supabase (Auth + Database required)
 - **Quotes data:** Seed your Supabase database with 100+ quotes across 5 categories. You can use any public quote API or dataset to source these.
 - App should run without crashes
 - Handle offline states gracefully
-

Submission Requirements

1. GitHub Repository (Public)

Your repo must include:

- Complete source code
- README with:
 - Setup instructions (including Supabase config)
 - Your AI coding approach and workflow
 - List of all AI tools used
 - Link to Stitch/Figma Make designs
 - Any known limitations or incomplete features

2. Loom Video (8-10 minutes)

This is critical. Record yourself covering:

1. App Demo (3-4 min)

- Walk through all features
- Show auth flow, browsing, favorites, collections
- Demo the widget and notifications
- Show share card generation

2. Design Process (2 min)

- Show your Stitch/Figma Make workflow
- How you exported and implemented designs

3. AI Workflow Showcase (3-4 min)

- Your development setup (Claude Code, Cursor, etc.)
- Specific prompts that worked well
- How you debugged issues with AI
- Any clever techniques or workflows

3. TestFlight Build (iOS — Nice to Have)

If building with SwiftUI, submit a TestFlight link. Send invites to: lazyme2305@gmail.com

Marking Guidelines

Excellent (90-100%)

- All features implemented and working
- Polished UI with attention to detail
- Smooth animations and transitions
- Comprehensive error handling
- Excellent Loom video demonstrating strong AI workflow

Good (70-89%)

- Most features implemented (may be missing widget or some polish)
- Clean UI, minor rough edges
- Good error handling
- Clear Loom video with decent AI workflow demonstration

Acceptable (50-69%)

- Core features working (auth, browse, favorites, share)
- Basic UI, some UX issues
- Some error handling
- Loom video present but lacking detail

Below Expectations (<50%)

- Missing core features
- Crashes or major bugs
- Poor UI/UX
- Incomplete or missing Loom video

Tips for Success

- **Prioritize core features first:** Auth → Browse → Favorites → Share. Then add polish.
- **Don't hide AI usage:** We want to see your workflow. Screen record if possible.
- **Quality over quantity:** A polished app with 80% features beats a buggy app with 100%.
- **Test on real device:** Especially notifications and widgets.
- **Seed good data:** 100+ quality quotes across categories makes the app feel complete.

FAQs

Q: Can I use any framework? A: Yes — SwiftUI, React Native, Flutter, Kotlin. Your choice.

Q: What if I can't complete all the features? A: Submit what you have. Document incomplete features in README. We value quality over quantity.

Q: Can I use UI libraries or templates? A: Yes, but mention them in your README. We evaluate how you build on top of starting points.

Q: What if I've never used Supabase? A: Their docs are excellent, and AI tools can help. This is part of what we're evaluating — your ability to learn quickly.

Q: Is the widget required? A: It's worth 10 marks. If you're short on time, prioritize core features first.

Questions?

Reply to this email and we'll get back to you within 24 hours.

Good luck! 