internStudio

ANDROID DEVELOPMENT INTERNSHIP

TASK GUIDE

Build apps people carry with them.
Native Android, done properly.

SIX ADVANCED PROJECTS — CHOOSE ANY TWO

01 **Offline-First Expense & Budget Tracker**

02 **Real-Time Location-Based Service Finder**

03 **Habit Tracking & Personal Analytics App**

04 **Peer-to-Peer Chat Application with Media Sharing**

05 **Multi-User Task & Project Collaboration App**

06 **E-Commerce Shopping App with Order Tracking**

INTERNSHIP	REQUIRED TASKS	FOCUS
Android Development	Complete any 2 projects	Kotlin/Java · Local Storage · APIs · Background Work · UI

INTERNSHIP INSTRUCTIONS

Your Internship Tasks

Select and complete **any 2 projects** from the task list. Each should be a real, installable Android app — not a single-screen demo of one widget.

Design	Build	Test	Polish	Document	Publish

EVERY PROJECT SHOULD INCLUDE

- [ ] A properly structured app (not one giant Activity)
- [ ] Local persistence (Room or equivalent)
- [ ] Sensible handling of the app lifecycle
- [ ] Network calls handled off the main thread
- [ ] Loading, empty and error states in the UI
- [ ] Permissions requested only when needed, explained clearly
- [ ] Works correctly after rotation / process death
- [ ] A real app icon and reasonable UI polish

BEFORE YOU SUBMIT

- [ ] Complete any 2 projects from the list
- [ ] Push the project to GitHub
- [ ] Provide a debug APK or install instructions
- [ ] Add a README with setup steps
- [ ] Add screenshots or a short demo video
- [ ] Submit details via the submission form
- [ ] Share your work on LinkedIn as required

> **Suggested technologies:** Kotlin · Jetpack Compose or XML layouts · Room · Retrofit · Coroutines/WorkManager. Optional: Firebase, Google Maps SDK.

Submit your completed work here

Once both projects are working end to end and pushed to GitHub, submit your project details through the official submission form.

https://forms.gle/vxL5JE8cvRXXfDBv8

> **A note on quality.** Test what happens when the network drops, the app is backgrounded, or the screen rotates mid-task. Those moments are where real apps are judged.

01 Offline-First Expense & Budget Tracker

Build an expense tracker that works fully offline and syncs when connectivity returns.

REQUIREMENTS

- Local database (Room) as the source of truth
- Background sync to a remote API when online
- Category-based spending charts
- Monthly summary notifications
- Full CRUD on transactions while offline
- Conflict handling for edits made while offline
- Budget limits with visual progress
- Works correctly after app restart

Focus: Offline-first architecture, local persistence and sync conflict handling.

02 Real-Time Location-Based Service Finder

Build an app that finds and displays nearby services (e.g. clinics, ATMs, restaurants) on a live map.

REQUIREMENTS

- Request and handle location permissions properly
- Display results on an interactive map
- Show distance and basic details per result
- Handle location unavailable / denied gracefully
- Query nearby places from a real API
- Support search and category filtering
- Geofencing or proximity alerts for saved places
- Cache recent results for offline viewing

Focus: Location services, maps integration and graceful permission handling.

03 Habit Tracking & Personal Analytics App

Build an app that tracks daily habits and surfaces meaningful personal analytics over time.

REQUIREMENTS

- Create and track multiple habits
- Local notifications / reminders
- Handle missed days without breaking the data model
- Data export (CSV or similar)
- Daily check-in with streak tracking
- Charts showing trends over weeks and months
- Widget or quick-add shortcut
- Works correctly across app restarts and reboots

Focus: Scheduled notifications, local analytics and a genuinely useful daily-use app.

04 Peer-to-Peer Chat Application with Media Sharing

Build a real-time chat app supporting text and media messages between users.

REQUIREMENTS

- User authentication
- Media (image) sharing in chat
- Offline message queueing
- Multiple conversations / contacts
- Real-time message delivery
- Message delivery / read status
- Push notifications for new messages
- Efficient handling of large chat histories (pagination)

Focus: Real-time messaging, media handling and push notification integration.

05 Multi-User Task & Project Collaboration App

Build an app where teams manage shared tasks and projects, synced across devices.

REQUIREMENTS

- User authentication and team membership
- Real-time or near-real-time sync across devices
- Push notifications for assignment/updates
- Search and filtering across tasks
- Shared projects with tasks assigned to members
- Offline support with sync on reconnect
- Task status and due-date tracking
- Role-based permissions (owner vs. member)

Focus: Multi-user sync, offline support and role-based collaboration.

06 E-Commerce Shopping App with Order Tracking

Build a complete shopping app from product browsing through checkout and order tracking.

REQUIREMENTS

- Product catalog with search and filters
- Checkout flow with address and payment (test mode)
- Push notifications on order status changes
- Image loading and caching done properly
- Cart management with persistent state
- Order history and status tracking
- Wishlist / saved items
- Graceful handling of network failures during checkout

Focus: A complete mobile commerce flow, state management and reliable network handling.

FINAL STEP

Ready to Submit?

Verify everything below before you send your work in.

SUBMISSION

- [ ] GitHub repository
- [ ] README with setup steps
- [ ] Debug APK or install instructions
- [ ] Screenshots or demo video
- [ ] Submission form completed
- [ ] LinkedIn post where required

POLISH

- [ ] Permissions requested sensibly
- [ ] Reasonable UI polish and a real icon
- [ ] No obvious crashes in normal use

APP

- [ ] Properly structured, not one giant Activity
- [ ] Survives rotation and process death
- [ ] Loading / empty / error states handled

DATA

- [ ] Local persistence works correctly
- [ ] Network calls off the main thread
- [ ] Data syncs correctly when connectivity returns

Important. Complete any 2 projects. Each should be a real, installable app you can demo on a device — not just code that compiles.

Submission Form

Submit your repository, live link and screenshots here.

https://forms.gle/vxL5JE8cvRXXfDBv8

internStudio

LEARN · BUILD · GROW

www.internstudio.in

team.internstudio@gmail.com