Android Mini Project Ideas PDF

An Android mini project should take one to four weeks and teach one core skill: layouts and click handling, lists, local storage with Room, network calls with Retrofit, or a device feature such as the camera. A notes app, quiz app, weather app and QR scanner are good choices in Kotlin.

Key takeaways

- Use Kotlin and Android Studio; start with Jetpack Compose or XML layouts as your course teaches.
- Learn lists, navigation and Room before network calls.
- Use free public APIs for weather, news or quotes.
- Test on a real phone and record a short demo video.
- Each idea fits in one to four weeks.

Who this PDF is for

This PDF is for 2nd and 3rd year students who want Android apps for labs or semester submissions. Every idea runs in Android Studio with Kotlin and builds skills step by step: UI, lists, storage, networking and device features.

Finish one beginner app before you try an intermediate one, and keep each app in its own GitHub repository with a short README and screenshots.

How to choose your project

- **Pick one main skill** for each app.
- **Keep screens few**: two to four is enough.
- **Use a real phone** with USB debugging.
- **Sketch screens on paper** before you code.

Beginner mini projects (1 to 2 weeks)

1. Calculator App

- **Problem:** Basic and scientific calculations.
- **Data and services:** No external data.
- **Tech stack:** Kotlin, Jetpack Compose or XML layouts.
- **Main modules:** Button grid, expression handling, history.
- **Steps:** Build the layout, handle taps, evaluate expressions safely, keep history.

- **Show in the viva:** How screen rotation affects state.
- **Extensions:** Add unit conversion.

2. Unit Converter

- **Problem:** Convert length, weight and temperature.
- **Data and services:** No external data.
- **Tech stack:** Kotlin, Compose.
- **Main modules:** Category picker, input, conversion logic, swap units.
- **Steps:** Write conversion functions, connect to dropdowns, validate input.
- **Show in the viva:** Input validation.
- **Extensions:** Add currency conversion with a free API.

3. Quiz App

- **Problem:** Timed multiple-choice quiz with scores.
- **Data and services:** Questions in a local JSON file.
- **Tech stack:** Kotlin, Compose, ViewModel.
- **Main modules:** Question screen, timer, score, result screen.
- **Steps:** Load JSON, show questions, keep state in a ViewModel, show results.
- **Show in the viva:** Why the ViewModel keeps state across rotation.
- **Extensions:** Add categories.

4. BMI and Water Intake Tracker

- **Problem:** Calculate BMI and log daily water.
- **Data and services:** No external data.
- **Tech stack:** Kotlin, Compose, DataStore.
- **Main modules:** BMI form, water counter, daily reset, reminder.
- **Steps:** Build the calculator, store daily count in DataStore, reset at midnight.
- **Show in the viva:** DataStore versus SharedPreferences.
- **Extensions:** Add a weekly chart.

Intermediate mini projects (2 to 3 weeks)

5. Notes App with Room

- **Problem:** Create, edit, search and delete notes.
- **Data and services:** No external data.
- **Tech stack:** Kotlin, Compose, Room, ViewModel.
- **Main modules:** Notes list, editor, search, categories.
- **Steps:** Define the Room entity and DAO, build list and editor screens, add search.
- **Show in the viva:** How Room maps objects to tables.
- **Extensions:** Add reminders with notifications.

6. Weather App

- **Problem:** Show weather for current location or a searched city.

- **Data and services:** A free weather API such as OpenWeatherMap.
- **Tech stack:** Kotlin, Retrofit, Coroutines, Location services.
- **Main modules:** Location permission, API call, result screen, error states.
- **Steps:** Request location, call the API with Retrofit, show results, handle no-internet.
- **Show in the viva:** Coroutines in simple words.
- **Extensions:** Cache the last result.

7. News Reader

- **Problem:** Browse headlines by category.
- **Data and services:** A news API with a free developer tier.
- **Tech stack:** Kotlin, Retrofit, Compose, Coil for images.
- **Main modules:** Category tabs, list, detail view, bookmarks.
- **Steps:** Fetch headlines, show lists with images, open articles, save bookmarks in Room.
- **Show in the viva:** How images load efficiently.
- **Extensions:** Add offline reading.

8. Attendance Tracker for Students

- **Problem:** Track your attendance per subject.
- **Data and services:** No external data.
- **Tech stack:** Kotlin, Compose, Room.
- **Main modules:** Subjects, mark present or absent, percentage, warnings.
- **Steps:** Store records per subject, compute percentage, warn under your college's limit.
- **Show in the viva:** How the percentage is calculated.
- **Extensions:** Predict classes needed to reach the limit.

Stretch mini projects (3 to 4 weeks)

9. QR Code Scanner and Generator

- **Problem:** Scan and create QR codes.
- **Data and services:** No external data.
- **Tech stack:** Kotlin, CameraX, ML Kit Barcode Scanning, a QR generator library.
- **Main modules:** Camera preview, scan handling, generator, history.
- **Steps:** Set up CameraX, run barcode scanning, show results, generate codes from text.
- **Show in the viva:** Camera permission flow.
- **Extensions:** Add Wi-Fi QR sharing.

10. Music Player

- **Problem:** Play songs stored on the phone.
- **Data and services:** Audio files on the device.
- **Tech stack:** Kotlin, Media3 ExoPlayer, MediaSession.
- **Main modules:** Song list, player controls, background playback, notification controls.
- **Steps:** Read media files, play with ExoPlayer, keep playing in background.
- **Show in the viva:** Why background playback needs a service.

- **Extensions:** Add playlists.

11. Chat App with Firebase

- **Problem:** One-to-one chat with login.
- **Data and services:** Firebase project (free tier).
- **Tech stack:** Kotlin, Firebase Auth, Firestore, Compose.
- **Main modules:** Sign-in, user list, chat screen, real-time updates.
- **Steps:** Set up auth, store messages in Firestore, listen for changes.
- **Show in the viva:** Firestore security rules.
- **Extensions:** Add image messages.

12. Habit Tracker with Reminders

- **Problem:** Build daily habits with streaks.
- **Data and services:** No external data.
- **Tech stack:** Kotlin, Room, WorkManager, notifications.
- **Main modules:** Habit list, check-ins, streaks, daily reminders.
- **Steps:** Store check-ins, compute streaks, schedule reminders with WorkManager.
- **Show in the viva:** How WorkManager schedules work.
- **Extensions:** Add home screen widgets.

Mini project report structure

A mini project report is usually 10 to 20 pages. Keep it short and clear:

1. **Title page and certificate** in your department's format.
2. **Introduction:** the problem in one paragraph and your objectives as a list.
3. **Tools used:** Android Studio version, Kotlin, libraries, minimum Android version and test phone.
4. **Design:** one block diagram or flowchart, plus screen sketches and the navigation flow.
5. **Implementation:** what each module does, with 3 to 6 screenshots.
6. **Results:** screenshots of each screen and a short demo video link.
7. **Conclusion and future scope,** then references.

Ask your guide for the exact format before you start writing.

Common mistakes to avoid

- App crashes on rotation because state is not saved.
- Running network calls on the main thread.
- Missing permission handling, so the app fails silently.
- No screenshots in the report.

Frequently asked questions

What is a good Android mini project for beginners?

A calculator, unit converter or quiz app. They teach layouts, clicks and state without a back end.

Compose or XML for a mini project?

Use what your course teaches. Jetpack Compose is Google's modern toolkit and is good to learn.

Can I build Android apps on a low-end laptop?

Android Studio needs a fair amount of memory. Testing on a real phone instead of the emulator helps a lot.

Which free APIs can I use?

Weather, news and quote APIs with free tiers work well. Read each API's usage limits.

Does this PDF include full source code?

No. It gives the problem, data, stack, modules and steps for each project. You write the code yourself, which is what examiners test in the viva.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- Android Basics with Compose
- Android Developers: Room
- Firebase documentation
- Google ML Kit barcode scanning

About this PDF

Prepared by the Uncodemy training team as practice project ideas. Datasets and tools named are public; check each dataset's licence before use. We update it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550