

DEV MOBILE

Looking at your fitness app development plan, I can provide you with the specific technologies and resources for each phase of your project. Let me break this down by week and feature:

Week 1: Project Setup and User Authentication

Technologies:

- **Android Studio:** Latest stable version for Android development
- **Kotlin:** Primary programming language
- **MVVM Architecture:** Using ViewModel, LiveData components
- **Firebase Authentication:** For user login/registration
- **Firebase Firestore:** For user profile storage
- **GitHub:** For version control
- **GitHub Actions:** For continuous integration

Resources:

- [Android Studio Download][<https://developer.android.com/studio>]
- [Firebase Console][<https://console.firebase.google.com/>] - Set up a new project
- [Firebase Authentication Guide][<https://firebase.google.com/docs/auth/android/start>]
- [GitHub Actions Documentation][<https://docs.github.com/en/actions>]
- [MVVM Architecture Components Guide][<https://developer.android.com/topic/libraries/architecture>]
- [Android Jetpack Navigation Component][<https://developer.android.com/guide/navigation>]
- For login/registration flow

Week 2: Exercise Library and Video Playback

Technologies:

- **ExoPlayer:** For video playback
- **Firebase Storage:** For storing exercise videos
- **RecyclerView:** For displaying exercise lists
- **Glide or Coil:** For image loading and caching
- **Room Database:** Optional local caching of exercise data
- **Material Design Components:** For UI elements

Resources:

- [ExoPlayer Documentation][<https://exoplayer.dev/hello-world.html>]
- [Firebase Storage Guide][<https://firebase.google.com/docs/storage/android/start>]
- [RecyclerView Implementation Guide][<https://developer.android.com/guide/topics/ui/layout/recyclerview>]
- [Coil Image Loading Library][<https://coil-kt.github.io/coil/>]
- [Material Design Components][<https://material.io/develop/android>]
- [Android Paging Library][<https://developer.android.com/topic/libraries/architecture/paging/v3-overview>] - For efficient loading of exercise lists

Week 3: Workout Planning and Progress Tracking

Technologies:

- **Room Database:** For local storage of workout plans
- **Firebase Firestore:** For cloud syncing of workout data
- **MPAndroidChart:** For progress visualization
- **WorkManager:** For scheduling workouts
- **SharedPreferences:** For user settings
- **Coroutines:** For asynchronous operations

Resources:

- [Room Database Implementation][<https://developer.android.com/training/data-storage/room>]
- [MPAndroidChart Library][<https://github.com/PhilJay/MPAndroidChart>]
- [WorkManager Documentation][<https://developer.android.com/topic/libraries/architecture/workmanager>]
- [Kotlin Coroutines Guide][<https://kotlinlang.org/docs/coroutines-guide.html>]
- [Firestore Data Modeling Guide][<https://firebase.google.com/docs/firestore/data-model>]
- [Two-way Data Binding][<https://developer.android.com/topic/libraries/data-binding>] - For workout entry forms

Week 4: Notifications and Final Polish

Technologies:

- **Android Notifications API:** For creating user notifications
- **WorkManager:** For scheduling notifications
- **Firebase Cloud Messaging:** For potential server-triggered notifications
- **Espresso:** For UI testing
- **JUnit:** For unit testing
- **Firebase App Distribution:** For beta testing
- **Material Design Theming:** For consistent UI polish

Resources:

- [Notifications API Guide][<https://developer.android.com/develop/ui/views/notifications>]
- [Firebase Cloud Messaging][<https://firebase.google.com/docs/cloud-messaging>]
- [Espresso Testing Framework][<https://developer.android.com/training/testing/espresso>]
- [JUnit Testing in Android][<https://developer.android.com/training/testing/local-tests>]
- [Firebase App Distribution][<https://firebase.google.com/docs/app-distribution>]
- [Material Design Theming Guide][<https://material.io/design/material-theming>]
- [Android Debug Bridge (ADB)][<https://developer.android.com/studio/command-line/adb>] - For advanced debugging

DevOps Implementation Resources

GitHub Setup:

- [GitHub Branch Protection Rules][<https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/defining-the-mergeability-of-pull-requests/about-protected-branches>]
- [GitHub Projects][<https://docs.github.com/en/issues/planning-and-tracking-with-projects/learning-about-projects>]
- [GitHub Pull Request Templates][<https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/creating-a-pull-request-template-for-your-repository>]

CI/CD Pipeline:

- [Android CI/CD with GitHub Actions][<https://proandroiddev.com/android-ci-cd-pipeline-with-github-actions-demystifying-building-with-gradle-build-customization/>]
- [Gradle Build Customization][<https://developer.android.com/studio/build/gradle-tips>]
- [Firebase App Distribution via GitHub Actions][<https://firebase.google.com/docs/app-distribution/android/distribute-gradle>]

Testing Resources:

- [Android Testing Fundamentals][<https://developer.android.com/training/testing/fundamentals>]
- [Mockito for Mocking in Tests][<https://site.mockito.org/>]
- [Android UI Testing Best Practices][<https://developer.android.com/training/testing/ui-testing/espresso-testing>]

Agile Implementation Tools

- [Trello][<https://trello.com/>] - Alternative to GitHub Projects for sprint tracking
- [Discord][<https://discord.com/>] or [Slack][<https://slack.com/>] - For team communication
- [Google Drive][<https://drive.google.com/>] - For shared documents
- [Burndown Chart Templates][<https://www.atlassian.com/agile/tutorials/burndown-charts>] - Simple Excel/Google Sheets templates

Additional Helpful Resources

- **UI/UX Design:** [Figma][<https://www.figma.com/>] - For collaborative design before implementation
- **Performance Monitoring:** [Firebase Performance Monitoring][<https://firebase.google.com/docs/perf-mon>]
- **Crash Reporting:** [Firebase Crashlytics][<https://firebase.google.com/docs/crashlytics>]
- **Network Calls:** [Retrofit][<https://square.github.io/retrofit/>] with [OkHttp][<https://square.github.io/okhttp/>] - If you need external API integration
- **Dependency Injection:** [Hilt][<https://developer.android.com/training/dependency-injection/hilt-android>] - For managing dependencies
- **Offline Support:** [Room + Network Bound Resource Pattern][<https://developer.android.com/topic/libraries/architecture/guide.html#addendum>]