

The Experiment - Complete Development Roadmap

Project Overview

App Name: The Experiment
Tagline: "What if I don't give up?"
Purpose: Personal workout planning app using saved Instagram workout videos
Tech Stack: Flutter, Firebase, Riverpod, Cloud AI
Learning Goal: Master Flutter/Firebase/Cloud AI for career advancement & ACE certification

Current Status (Completed)

Week 1: Flutter Architecture Bootcamp

Completed Components:

- Project setup with modular architecture (data/logic/ui layers)
- Foundation models: `MuscleGroup`, `WorkoutVideo`, `WorkoutPlan`
- Repository Pattern: `VideoRepository` interface + `LocalVideoRepository`
- Riverpod providers for state management
- UI widgets: `MuscleGroupSelector`, `DurationSlider`
- Home screen with workout generation
- Mock data: 15 workout videos across 5 muscle groups

Git Status: Committed and pushed to GitHub

Week 2: Advanced Flutter Patterns (Async Programming)

Completed Components:

- Async/await understanding (Future vs Stream)
- `WorkoutTimerService` with Stream-based timer
- `WorkoutExecutionScreen` with `StreamBuilder`
- Real-time countdown timer, progress tracking
- Pause/Resume/Skip functionality
- Completion screen

Key Learnings:

- Sequential vs parallel async operations
 - Error handling with `try/catch`
 - Streams for continuous data flow
 - `StreamBuilder` for reactive UI
-

Development Roadmap by Phase

Phase 1: Solid Foundations (Weeks 3-4)

Week 3: Firebase Core Mastery

Study Focus: Firestore structure, Authentication, Cloud Storage basics

Implementation Tasks:

1. **Firebase Project Setup**

- Create Firebase project
- Add Firebase to Flutter app (iOS + Android + Web)
- Configure Firebase Authentication

2. **Data Migration**

- Design Firestore schema:
 - users/{userId} - User profiles
 - users/{userId}/workouts/{workoutId} - Workout history
 - users/{userId}/videos/{videoId} - User's saved videos
- Create `FirebaseVideoRepository` implements `VideoRepository`
- Migrate from local mock data to Firestore

3. **Authentication**

- Implement email/password sign-up and login
- Google Sign-In integration
- User profile creation

4. **Cloud Storage**

- Set up storage buckets for video thumbnails
- Implement thumbnail upload/download

Integration Points:

- Swap `LocalVideoRepository` → `FirebaseVideoRepository` in providers
- Add authentication wrapper to app
- Save completed workouts to Firestore

Learning Exercises (generate on the day):

- Firestore security rules practice
- Query optimization challenges
- Authentication flow debugging

Week 4: Local Testing & Debugging

Study Focus: Flutter DevTools, Firebase Emulator Suite, logging, error tracking

Implementation Tasks:

1. **Firebase Emulator Setup**

- Install and configure Firebase Emulator Suite
- Set up local Firestore, Auth, and Storage emulators
- Create seed data scripts

2. **Testing Infrastructure**

- Write unit tests for models and repositories
- Widget tests for custom widgets
- Integration tests for workout flow

3. **Error Tracking**

- Implement Firebase Crashlytics

- Add custom logging throughout app
- Create error boundary widgets

4. Debug Improvements

- Add development-only debug screens
- Implement network request logging
- Create mock data switcher (local/Firebase/mock)

Integration Points:

- Test all existing features with emulator
 - Fix any bugs discovered during testing
 - Document common issues and solutions
-

Phase 2: Full-Stack Depth (Weeks 5-10)

Week 5: Firestore Optimization

Study Focus: Query optimization, indexing, data fetching strategies

Implementation Tasks:

1. Query Optimization

- Implement pagination for video lists
- Add compound indexes for complex queries
- Optimize workout history queries

2. Caching Strategy

- Implement VideoCache service (from async exercises)
- Add offline persistence
- Create data refresh strategies

3. Performance Monitoring

- Add query performance tracking
- Identify and fix slow queries
- Implement lazy loading for large datasets

Feature Addition:

- Workout history screen showing past workouts
 - Filter/sort functionality for video library
 - Search with debouncing (from async exercises)
-

Week 6: Firebase Functions

Study Focus: Serverless logic, Cloud Functions triggers

Implementation Tasks:

1. Cloud Functions Setup

- Initialize Cloud Functions in project
- Deploy first "Hello World" function

2. Workout Completion Triggers

- Function triggered on workout completion
- Calculate and update user statistics
- Generate achievement badges

3. Scheduled Functions

- Daily workout reminder notifications
- Weekly progress summary emails
- Clean up old temporary data

4. API Integration Prep

- Create function to validate Instagram API tokens
- Set up secure credential storage

New Features:

- Push notifications for workout reminders
 - Achievement system (3-day streak, 10 workouts, etc.)
 - Weekly progress emails
-

Week 7: User Roles & Security Deep Dive

Study Focus: RBAC, security rules testing

Implementation Tasks:

1. Security Rules Refinement

- Write comprehensive Firestore security rules
- Implement user data isolation
- Add validation rules for data integrity

2. Admin Features (optional, for portfolio)

- Admin dashboard for app monitoring
- User management interface
- Content moderation tools

3. Security Testing

- Test security rules with multiple user accounts
- Attempt unauthorized access scenarios
- Document security model

Portfolio Enhancement:

- Create detailed security documentation
 - Add "Security Features" section to README
 - Prepare security discussion points for interviews
-

Week 8: Data Pipelines & Analytics

Study Focus: BigQuery basics, analytics integration

Implementation Tasks:

1. Firebase Analytics Setup

- Implement event tracking throughout app
- Track key metrics: workout starts, completions, user retention

2. BigQuery Integration

- Export Firebase Analytics to BigQuery
- Create basic analytics queries
- Build data visualization dashboards

3. User Insights

- Most popular exercises
- Average workout duration
- User engagement patterns
- Muscle group preferences

New Features:

- Personal statistics dashboard
- Progress graphs (workouts per week, total time)

- Exercise frequency heatmap
-

Week 9: CI/CD & Deployment

Study Focus: Automated pipelines, deployment strategies

Implementation Tasks:

- 1. GitHub Actions Setup**
 - Automated testing on pull requests
 - Automated builds for iOS/Android
 - Firebase deployment automation
- 2. Multi-Environment Setup**
 - Development environment
 - Staging environment
 - Production environment
- 3. Release Management**
 - Version numbering strategy
 - Release notes automation
 - Beta testing workflow (TestFlight, Play Console)
- 4. App Store Preparation**
 - Create app icons and screenshots
 - Write app store descriptions
 - Set up app store listings

Deployment:

- Deploy to Firebase Hosting (web version)
 - Prepare for app store submission
-

Week 10: Lab Consolidation & Mini Capstone

Study Focus: Combining learned modules into cohesive features

Implementation Tasks:

- 1. Feature Polish**
 - UI/UX improvements based on self-testing
 - Animation and transition refinements
 - Accessibility improvements
- 2. Integration Testing**
 - End-to-end testing of complete user flows
 - Performance testing under load
 - Cross-platform testing (iOS/Android/Web)
- 3. Documentation**
 - Complete technical documentation
 - Create user guide
 - Record demo video for portfolio

Portfolio Milestone:

- Polished, demo-ready version
 - Professional README with architecture diagrams
 - Demo video showcasing key features
-

Phase 3: Applied Cloud & AI (Weeks 11-16)

Week 11: AI Insights Integration

Study Focus: ML pipelines, Vertex AI basics

Implementation Tasks:

- 1. **Instagram API Integration**
 - Implement InstagramVideoRepository
 - OAuth flow for Instagram authentication
 - Fetch user's saved Instagram videos
 - Download and cache video metadata
- 2. **AI-Powered Features** (Basic)
 - Automatic exercise categorization (if Instagram doesn't provide)
 - Smart workout recommendations based on history
 - Difficulty level estimation
- 3. **Cloud Functions + AI**
 - Function to process new videos
 - Extract metadata (duration, title parsing)
 - Generate workout suggestions

Major Feature:

- Replace mock data with real Instagram videos
 - Automatic workout plan generation based on goals
-

Week 12: Advanced BigQuery

Study Focus: Complex queries, aggregations, visualizations

Implementation Tasks:

- 1. **Advanced Analytics**
 - Complex queries with joins and aggregations
 - User cohort analysis
 - Workout pattern detection
- 2. **Personalized Insights**
 - "You worked out 3x more this month than last"
 - "Your most improved muscle group is..."
 - "You're most consistent on Tuesdays"
- 3. **Data Visualization**
 - Interactive charts using recharts
 - Progress over time graphs
 - Comparison with personal bests

New Screen:

- Insights dashboard with personalized statistics
-

Week 13: Pattern Detection & Recommendations

Study Focus: AI-based recommendations, trend analysis

Implementation Tasks:

- 1. **Workout Pattern Analysis**

- Detect favorite exercises
 - Identify optimal workout times
 - Find muscle group neglect patterns
2. **Smart Recommendations**
 - "You haven't worked legs in a week"
 - "Your best performance is in the morning"
 - "Try adding core exercises to your routine"
 3. **Adaptive Planning**
 - Workout plans that adapt to history
 - Progressive overload suggestions
 - Rest day recommendations

AI Feature:

- Smart workout suggestions based on patterns
-

Week 14: PDF Export & Reporting

Study Focus: Document generation, automated reporting

Implementation Tasks:

1. **Report Generation**
 - Weekly workout summary PDFs
 - Monthly progress reports
 - Exercise library PDF export
2. **Cloud Function for PDFs**
 - Server-side PDF generation
 - Email delivery of reports
 - Cloud Storage for generated reports
3. **Sharing Features**
 - Share workout plans with friends
 - Export data for personal records
 - Social media integration (optional)

New Features:

- Export workout history as PDF
 - Email weekly summaries
 - Shareable workout achievements
-

Week 15: Geographic & Visual Enhancements

Study Focus: Mapping, geospatial data, visualization

Implementation Tasks:

1. **Workout Location Tracking** (optional)
 - Track where workouts are performed
 - Map view of workout locations
 - Location-based workout suggestions
2. **Visual Enhancements**
 - Better video preview thumbnails
 - Animated progress indicators
 - Custom illustrations for muscle groups
3. **AR/VR Exploration** (stretch goal)
 - AR form checker (very advanced)

- 3D muscle group visualizations

Polish:

- Visual redesign with modern UI/UX trends
 - Micro-animations and delightful interactions
-

Week 16: Applied AI Capstone

Study Focus: Integration of all AI features

Implementation Tasks:

- 1. **AI Dashboard**
 - Comprehensive AI-powered insights
 - Predictive analytics (workout adherence)
 - Goal achievement probability
- 2. **Complete AI Features**
 - Full recommendation engine
 - Automated workout generation
 - Performance prediction
- 3. **Testing & Refinement**
 - AI accuracy testing
 - User feedback integration
 - Model performance optimization

Capstone Deliverable:

- Fully functional AI-powered workout app
 - Complete feature set
 - Portfolio-ready presentation
-

Phase 4: Cloud Engineer Prep (Weeks 17-20)

Week 17: Deployment Deep Dive

Implementation: Production-ready deployment

Week 18: Monitoring & Logging

Implementation: Comprehensive observability

Week 19: Performance Optimization

Implementation: App performance tuning

Week 20: Final Portfolio Polish

Implementation: Documentation, demo video, interview prep

Key Milestones for Portfolio

Milestone 1: MVP (End of Week 6)

- Working workout app with Firebase backend
- Authentication and data persistence
- Basic workout tracking
- **Demo-able to recruiters**

Milestone 2: Feature Complete (End of Week 12)

- Instagram API integration
- Analytics and insights
- Push notifications
- **Strong portfolio piece**

Milestone 3: AI-Powered (End of Week 16)

- Full AI recommendation system
- Advanced analytics
- Predictive features
- **Standout portfolio project**

Milestone 4: Production Ready (End of Week 20)

- Published to app stores (optional)
- Complete documentation
- Professional demo video
- **Interview-ready showcase**

Technical Architecture Evolution

Current Architecture (Weeks 1-2)



UI Layer (Widgets/Screens)



Logic Layer (Providers/Services)



Data Layer (Repositories/Models)



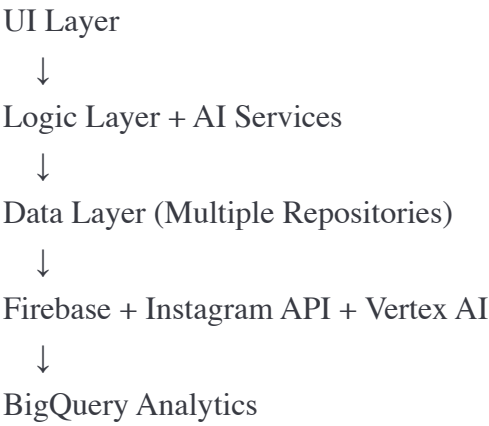
Mock Data (Local)

Phase 2 Architecture (Weeks 3-10)

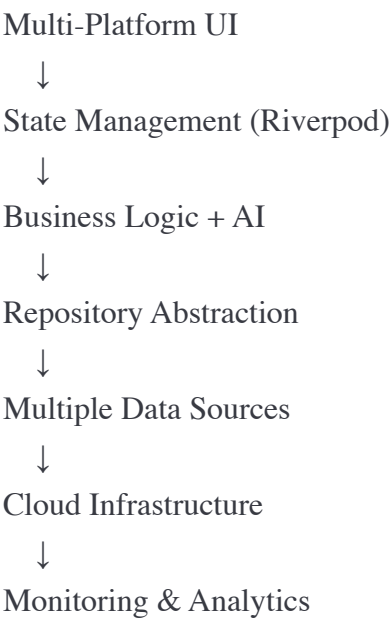




Phase 3 Architecture (Weeks 11-16)



Phase 4 Architecture (Weeks 17-20)



ACE Certification Alignment

ACE Skills Developed

- **Cloud Infrastructure:** Firebase, Cloud Functions, Cloud Storage
 - **Data Management:** Firestore, BigQuery, data pipelines
 - **Security:** IAM, authentication, security rules
 - **Deployment:** CI/CD, multi-environment setup
 - **Monitoring:** Analytics, Crashlytics, logging
 - **AI/ML Integration:** Vertex AI, recommendation systems
 - **Serverless Computing:** Cloud Functions
 - **Database Design:** NoSQL data modeling
 - **API Integration:** Instagram API, OAuth flows
-

Daily Learning Pattern

Suggested Daily Structure

1. **Review yesterday's work** (10 min)
 2. **Study theory** from study plan (30-45 min)
 3. **Work session with Claude** (2-3 hours)
 - Implement features
 - Ask questions for deep understanding
 - Complete exercises generated on the spot
 4. **Git commit & documentation** (15 min)
 5. **Reflection** (10 min)
 - What did I learn?
 - What would I explain differently in an interview?
 - What's next?
-

Questions to Guide Each Session

When starting a new feature:

1. "What problem does this solve for users?"
2. "How does this fit into the overall architecture?"
3. "What are alternative approaches and why is this one best?"
4. "What could go wrong and how do we handle it?"
5. "How will this help in my ACE certification?"

When completing a feature:

1. "Can I explain why every line of code exists?"
 2. "What would I say about this in an interview?"
 3. "What did I learn that I didn't know before?"
 4. "Where else could this pattern be applied?"
 5. "What would I do differently next time?"
-

Portfolio Documentation Checklist

README Requirements

- ☐ Project overview and motivation
- ☐ Architecture diagram
- ☐ Tech stack with justification
- ☐ Key features with screenshots
- ☐ Setup instructions
- ☐ Design decisions explained
- ☐ Challenges faced and solutions
- ☐ Future enhancements
- ☐ What I learned

Code Quality

- ☐ Consistent formatting
- ☐ Meaningful variable names
- ☐ Comments explaining "why" not "what"
- ☐ No commented-out code
- ☐ Clear file organization
- ☐ Separation of concerns

Demo Materials

- ☐ Screenshots of key screens
- ☐ Screen recording of user flow
- ☐ Architecture diagram
- ☐ Database schema diagram
- ☐ Demo video (2-3 minutes)



Success Criteria

Technical Success







- ☒ App works reliably across platforms
- ☒ Clean, modular, maintainable code
- ☒ Proper error handling throughout
- ☒ Responsive UI with good UX
- ☒ Secure data handling
- ☒ Scalable architecture

Learning Success

- ☒ Can explain every architectural decision
- ☒ Understand async programming deeply
- ☒ Comfortable with Firebase ecosystem
- ☒ Can discuss trade-offs confidently

-  Ready for technical interviews
-  ACE-relevant skills demonstrated

Portfolio Success

-  Professional-looking project
 -  Clear documentation
 -  Demonstrates problem-solving
 -  Shows growth and learning
 -  Interview talking points prepared
 -  Stands out to recruiters
-

Context for Future Sessions

Paste this when starting a new session:

"I'm working on 'The Experiment' - a Flutter workout app using saved Instagram videos. Currently on Week [X] of my learning plan. Today I want to work on [specific feature/concept]. Here's where I left off: [brief summary]. I learn best by doing and asking questions to deepen understanding. Please provide exercises that challenge me to think, not just copy code."

Emergency Troubleshooting Guide

Common Issues & Solutions

1. **Firestore not connecting:** Check google-services.json placement
2. **Build errors after dependency changes:** Run `flutter pub get` and `flutter clean`
3. **Emulator issues:** Restart emulator, check Firebase CLI version
4. **Git conflicts:** Use `git status` to see conflicts, resolve manually
5. **Stream not updating:** Check if stream is being listened to correctly

When Stuck

1. Check error message carefully
 2. Google the exact error
 3. Check Firebase/Flutter documentation
 4. Ask Claude with full error context
 5. Take a break and come back fresh
-

Motivation Reminders

- "What if I don't give up?" - You're finding out!
 - Every day you code is a day closer to your goal
 - Small progress daily = massive progress over months
 - It's okay to not understand everything immediately
 - Questions show intelligence, not lack of knowledge
 - Your unique path (mom → IT) is your strength, not weakness
 - This app represents your commitment to growth
-

End of Roadmap Document

Last Updated: Week 2 Complete

Next Session: Week 3 - Firebase Core Mastery