# **🛠 Sprint Task Breakdown – Version 1.0**

This document outlines the **task-level execution plan** for **version v1.0**, focusing on foundational setup, core AI integration, and minimum viable user experience. Each sprint covers frontend, backend, and DevOps activities, and is aligned with internal dependencies, DoD criteria, and known risks.

### **✅ Version 1.0 – Foundational Launch (Aug 1 – Sept 12)**

**Goal:** Deliver intelligent AI-powered study guide generation with core app infrastructure.

## **🌀 Sprint 1: Aug 1–Aug 14**

**Sprint Goal:** Set up project foundation and core LLM integration.

### **✅ Frontend Tasks:**

* Scaffold new Flutter project with null safety and Clean Architecture folder structure
* Implement UI for verse/topic input with validation and loading state
* Build initial navigation stack (Onboarding > Home > Result > Error)
* Implement basic Onboarding screen (language select, app intro)
* Prepare accessibility: font scaling, color contrast compliance

### **✅ Backend Tasks:**

* Set up Firebase Auth (email, anonymous) with Supabase fallback
* Integrate GPT-3.5 or Claude API using secure token call from Supabase Edge Function
* Create prompt template for structured Bible study guide response
* Implement local mock mode for LLM (offline testing via JSON samples)

### **✅ DevOps Tasks:**

* Set up GitHub Actions for CI/CD (Flutter build, lint, test)
* Configure Firebase & Supabase projects (Auth + DB)
* Create GitHub Secrets for LLM keys (OPENAI/CLAUDE)

### **✅ Deliverables:**

* Working Flutter app scaffold (mobile + web)
* Functional auth system (Firebase + Supabase fallback)
* Prompt templates ready with mock mode fallback
* Onboarding and input screen functional

### **✅ DoD (Definition of Done):**

* Code pushed and compiles on CI
* Auth and LLM endpoints testable (mock + live mode)
* Schema baseline defined for study queries
* Accessibility checklist passed (font, contrast, scaling)

### **⚠️ Dependencies / Risks:**

* Access to valid OpenAI or Claude keys
* Prompt template refinement required for theological correctness

## **🌀 Sprint 2: Aug 15–Aug 28**

**Sprint Goal:** Generate and display structured study guide

### **✅ Frontend Tasks:**

* Build result screen with 5 core sections: Summary, Explanation, Related Verses, Reflection, Prayer
* Implement collapsible cards for each section
* Add retry button, error widget, and empty state fallback
* Handle empty history UI (no local cache)

### **✅ Backend Tasks:**

* Implement input → prompt → structured parser flow
* Save response metadata to Supabase (guide ID, timestamp, verse/topic)
* Cache recent guides locally using Hive (5–10 max)
* Implement retry & timeout logic for failed LLM responses

### **✅ DevOps Tasks:**

* Enable GitHub CI for pull requests (analyze, test)
* Deploy edge function to Supabase project
* Write integration tests for LLM-to-structured output

### **✅ Deliverables:**

* Frontend display of AI-generated guide (with fallback)
* Guide caching system (local + cloud metadata)
* Retry/error UX finalized
* Screen reader test passed

### **✅ DoD:**

* Input → Output validation working for 5+ test prompts
* Local + cloud cache working and synced
* Accessibility verified for result page

### **⚠️ Dependencies / Risks:**

* LLM token limits and latency
* Structural deviation in LLM response format

## **🌀 Sprint 3: Aug 29–Sept 12**

**Sprint Goal:** Final polish and prepare MVP release

### **✅ Frontend Tasks:**

* Add skeleton loaders, shimmer for slow responses
* Light/dark theme switcher with saved preference
* Optimize layout for tablet + web breakpoints
* Introduce intl-based localization structure (EN only)

### **✅ Backend Tasks:**

* Finalize system prompt with theological tone tuning
* Add version tagging to saved guides (metadata)
* Ensure fallback content logic supports mock mode in offline/test environments

### **✅ DevOps Tasks:**

* Generate Android/iOS builds for internal testing
* Configure Google Play & TestFlight internal testing tracks
* Set up Supabase log monitoring for LLM usage + error spikes

### **✅ Deliverables:**

* First deployable release (internal only)
* Guide prompt 90% aligned with sermon transcript tone
* Basic telemetry tracking enabled (e.g. prompt count, guide latency)

### **✅ DoD:**

* Build deployable on both Android and iOS internal tracks
* Guide version metadata persisted
* All critical paths tested (input → guide → error → cache)

### **⚠️ Dependencies / Risks:**

* Store review approval timing
* Prompt theological alignment drift across languages (future)

✅ **General Notes:**

* Supabase replaces Firestore across tasks for consistency
* Added onboarding and accessibility tasks where previously implied
* All fallback, retry, and offline logic made explicit in both frontend and backend
* Localization groundwork (intl) started in Sprint 3 for later support
* QA/test matrix made visible with accessibility and integration goals