Approach Document

Project: Oneverse

1. Introduction

This project aims to build a modern Bible study application that supports multiple translations and languages. The application will allow users to browse, search, and compare Bible verses, with additional features like bookmarking, highlighting, and personal notes.

Goals:

- Provide access to multiple Bible versions and languages.
- Enable powerful search and cross-reference features.
- Offer a clean, user-friendly interface across web and mobile devices.

2. Requirements Overview

Functional Requirements

- Browse by book → chapter → verse.
- Switch between Bible versions and languages.
- Full-text search (keyword, phrase).
- Parallel comparison of different versions.
- User features: bookmarks, highlights, notes (optional for MVP).

3. Architecture & Technology Stack

Frontend: React + TailwindCSS
Backend: Node.js (Express)
Database: MongoDB (NoSQL for flexible verse storage)

4. Development Approach

- **Methodology:** Agile Scrum (2-week sprints).
- **Version Control:** GitHub (feature-branch workflow).
- **Milestones:**
    - Sprint 1: Setup backend API with sample Bible data.
    - Sprint 2: Build frontend for browsing verses.
    - Sprint 3: Implement search and comparison features.
    - Sprint 4: Add user features (bookmarks, highlights).

5. Data Management

- Store verses in a structured relational format using Supabase (PostgreSQL).
- Support multiple versions and languages by tagging records with `version_id` and `language`.
- Enable full-text search using PostgreSQL.
- User data (bookmarks, highlights, notes) stored in separate tables with Row-Level Security.
- Backup and recovery plan: use Supabase automated backups and point-in-time recovery.

6. Deployment & DevOps

CI/CD Pipeline: GitHub Actions for automated builds and tests

Hosting:

- Frontend on **Vercel**
- Backend on **Supabase Edge Functions**
- Database on **Supabase PostgreSQl**
- **Monitoring:** Supabase dashboard, Sentry for frontend error tracking