

LogBible - Technical Documentation

This documentation provides a technical overview, setup instructions, and deployment guide for the **LogBible** application. It is intended for developers and system administrators maintaining or deploying the software.

1. Architecture Overview

LogBible is a monolithic web application built on the **Laravel** framework. It combines a robust backend for managing Bible data, user study logs, and AI generations with a modern frontend utilizing Blade templates and Tailwind CSS.

Key Components

- **MVC Pattern:** Follows standard Model-View-Controller architecture.
- **AI Integration:** Integrates with **Google Gemini AI** (`GeminiService.php`) to generate explanations and images based on scripture.
- **Data Strategy:**
 - **Structured Data:** MySQL is used for user data, logs, AI records, and metadata.
 - **Static Data:** The Bible text itself is structured in JSON files within `storage/bible-data/` for efficient retrieval and seeding, alongside database tables (`books`, `chapters`, `verses`).
- **Web Installer:** Includes a built-in installation wizard (`InstallController`) to simplify the initial setup on production servers.

2. Tech Stack

Backend

- **Framework:** Laravel 11.x (PHP 8.2+)
- **Database:** MySQL / MariaDB
- **AI Engine:** Google Gemini API
- **Image Handling:** GD Library / ImageMagick (server-side)

Frontend

- **Templating:** Laravel Blade
- **Styling:** Tailwind CSS (configured via `tailwind.config.js`)
- **Build Tool:** Vite (`vite.config.js`)
- **Scripting:** Vanilla JS / Alpine.js (implied for interactivity)

System Requirements

- PHP $\geq$ 8.2
- Composer
- Node.js & NPM
- BCMath, CType, JSON, Mbstring, OpenSSL, PDO, Tokenizer, XML PHP Extensions.

3. Project Structure

The codebase follows the standard Laravel directory structure with specific additions for the Bible logic:

Directory	Description
<code>app/Http/Controllers</code>	<code>Api/</code> <code>Admin/</code> <code>BibleController.php</code>
<code>app/Models</code>	<code>Book</code> <code>Chapter</code> <code>Verse</code> <code>AiExplanation</code> <code>StudyLog</code>
<code>app/Services</code>	<code>GeminiService.php</code>

`database/migrations`			
`database/seeders`			
`resources/views`		`bible`	`admin` `install`
`storage/bible-data`		Critical:	
`public/assets`			