Bible API Research Tool

Robert McCoy

robertemccoy.tech

March 2025

Project Overview

The objective of this project was to design and deploy a web-based Bible research tool that allows users to look up scripture passages in real time through a public API. An additional goal was to explore integration of a concordance system, using Strong's numbers, for word-based lookups and study.

Having previously used no-code and visual automation platforms such as Mind Studio and Zapier, I chose to approach this project with a different tool: Postman. This allowed me to expand my technical skill set, work directly with web APIs, and better understand request/response workflows.

The final deliverable includes a live web interface embedded in a WordPress site, backed by real-time API data, and tested extensively within Postman.

Tools and Platforms Used

ChatGPT (Postman Assistant) – Provided technical assistance, code generation, debugging help, and real-time support throughout the project. See the Lessons Learned section for additional context on its use and limitations.

Human–AI Collaboration

As a non-programmer, I relied heavily on collaborative interaction with ChatGPT's Postman Assistant to accomplish technical tasks. This interaction extended well beyond static Q&A, forming a dynamic and iterative working relationship. The assistant provided:

- Instruction in Postman fundamentals and API structure.
- Troubleshooting support for request failures and incorrect responses.
- Scripting assistance for writing JavaScript tests in Postman.
- Help in designing mock servers and simulating API behavior.
- Guidance in integrating front-end HTML and JavaScript with live API calls.

This collaboration allowed me to learn in real time, while making forward progress on a technically complex project.

Development Methods

Tool Selection: Chat GPT/Postman Assistant

Postman was selected as a platform to learn a new workflow centered around direct API interaction, as opposed to previous experiences with no-code platforms.

Testing and API Integration: Bible-API.com was selected for its free, public access and its ability to return plain JSON scripture data. Additional providers like Biblia.com and API.Bible were explored, but limitations in access and licensing prevented their use in this phase.

Mock Server Attempt: A mock server was created in Postman using a custom JSON file containing a sample Strong's Concordance dataset. While technically successful, the mock was ultimately limited in dynamic lookup functionality. I decided to remove this from the final public version, but plan to reintroduce concordance support in the future through an authorized API service such as Biblia or API.Bible.

Front-End Interface: A custom HTML and JavaScript interface was built with the assistant's guidance. This front-end uses standard fetch calls to retrieve Bible verses and displays them in a clean, responsive layout. The file was uploaded to WordPress and embedded using Elementor's HTML tools.

Documentation: A fully documented version of the Postman collection was published at: https://documenter.getpostman.com/view/43565881/2sB2cPjkEg

Final Deployment: The completed tool was published at the following live URL: https://robertemccoy.tech/bible-research/

Lessons Learned

This project provided several important learning opportunities:
- A deeper understanding of HTTP requests, status codes, and REST API architecture.
- Fluency in the Postman environment, including tests, variables, and environments.
- Increased confidence using browser tools for real-time debugging.
- The ability to create, test, and deploy a JavaScript-based interface for public use
- An appreciation for how AI collaboration can serve not only as a helper but also as a technical mentor.

-Throughout the project, extended sessions with the AI assistant occasionally led to slower performance and less reliable responses. This appeared to be related to session length and repeated file uploads. As a result, I restarted the session twice using a clear prompt to

reestablish context. This strategy helped restore responsiveness and maintain project clarity.

Conclusion and Future Work

This project represents a strong example of practical development, design, and deployment skills. I was able to leverage the combined power of modern tools, public APIs, and AI support to create a usable, well-tested system.

As a result of this experience, I have chosen to adopt Postman as my primary tool for future API-based projects. The clarity and control it offers has made it my preferred workspace for exploring, testing, and documenting web services.

Looking ahead, I plan to reintroduce the concordance functionality with full integration using either a licensed version of Biblia's API or a similar structured service. I also intend to improve the front-end experience, introduce search by keyword or topic, and expand the system to support additional translations or commentary.

The Bible Research Tool stands as a complete, functional product developed through active learning and experimentation, and is ready to evolve further with future enhancements.

Throughout the project, extended sessions with the AI assistant occasionally led to slower performance and less reliable responses. This appeared to be related to session length and repeated file uploads. As a result, I restarted the session twice using a clear prompt to reestablish context. This strategy helped restore responsiveness and maintain project clarity.