

Inclusion – Milestone 6

Andrew Rodriguez

26 Oct 2025

CST-391

Combined Project Overview

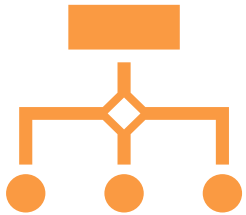
- **Purpose:** Personal Bible study web app to manage verse-specific comments across multiple Bible translations.
- **Backend:** REST API with full CRUD operations for user comments and verse data.
- **Database:** MySQL with multiple relational tables supporting key functionality.
- **Frontend Progress:**
 - Milestone 3–4: Angular app successfully connected to the backend API.
 - Milestone 5: Transitioned to a React frontend integrated with the same API.
- **Tools & Technologies:** Node.js, Express, JavaScript/TypeScript, Postman, MySQL Workbench.



Combined Implementation and Key Learnings

- **Implementation:**
 - Successfully implemented full CRUD endpoints with Express and TypeScript.
 - Frontend frameworks (Angular → React) integrated with the REST API.
 - Routing established for **Books**, **Verses**, and **Comments** pages using Angular Router and later React Router.
 - Verses displayed dynamically by translation; comments created and retrieved per verse.
 - API functionality verified with Postman; database changes confirmed in MySQL Workbench.
- **Key Learnings:**
 - Define API contracts early for smoother frontend-backend integration.
 - Postman testing quickly exposes backend issues before UI work.
 - Modular design (services/components/hooks) improves reusability and debugging.
 - Keeping schema diagrams updated prevents relational mapping errors.

Combined Challenges and Pending Issues

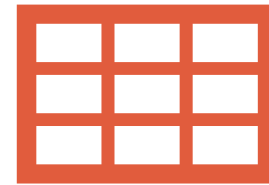


Challenges:

Aligning frontend routes (Angular/React Router) with backend API endpoints.

Debugging asynchronous API calls, data binding, and component state management.

Configuring CORS policies and environment variables for smooth communication.



Pending Issues:

Comment confirmation and book selection dropdowns remain incomplete.

Performance concerns when handling large verse datasets.

Input validation for comments still pending implementation.



Combined Next Steps / Future Work

- **Next Steps:**
 - Continue improving and expanding API documentation.
 - Refine database queries and strengthen error handling for better scalability.
 - Optimize performance to handle larger verse datasets efficiently.
 - Complete frontend components such as comment confirmation and book selection dropdowns.
 - Implement robust input validation for user comments.
 - Maintain seamless integration between frontend routes (React) and backend API endpoints.