

# Inclusion – Milestone 4

Andrew Rodriguez

28 Sep 2025

CST-391

## **Project Overview**

- Purpose: Personal Bible study web app to manage verse-specific comments across translations.
- Backend: REST API with CRUD operations.
- Database: MySQL with multiple tables to support functionality.
- Frontend Milestone: Angular app successfully connected to backend API
- Tools: Node.js, Express, TypeScript, Postman, MySQL Workbench

```
modifier_ob.
  mirror object to mirror
mirror_mod.mirror_object
 peration == "MIRROR_X":
elror_mod.use_x = True
mirror_mod.use_y = False
lrror_mod.use_z = False
 _operation == "MIRROR_Y"
____ror_mod.use_x = False
lrror_mod.use_y = True
 lrror_mod.use_z = False
  _operation == "MIRROR_Z"
  _rror_mod.use_x = False
  lrror_mod.use_y = False
  rror_mod.use_z = True
  melection at the end -add
   ob.select= 1
   er ob.select=1
   ntext.scene.objects.action
  "Selected" + str(modification
   irror ob.select = 0
  bpy.context.selected_obj
  lata.objects[one.name].sel
  int("please select exactin
  - OPERATOR CLASSES ----
      mirror to the selected
    ect.mirror_mirror_x"
  oxt.active_object is not
```

# Implementation and Key Learnings

Implementation: Angular services integrated with REST API

Routing established for Books, Verses, and Comments pages

Verses displayed dynamically by translation

Comments created and retrieved by verse

Key Learnings: Define API contracts early for smoother integration

Postman testing catches backend issues before frontend work

Modular Angular services aid debugging and reuse

Keeping schema diagrams updated reduces relation errors

## **Challenges and Pending Issues**

- Challenges: Aligning Angular routes with backend endpoints
- Debugging async calls and data binding
- CORS setup and environment configuration
- **Pending Issues:** Comment confirmation page incomplete
- Performance concerns with large verse datasets
- Input validation for comments still pending

#### **Next Steps**



Finalize comment confirmation workflow in Angular



Standardize error handling & validation across API



Optimize database queries for larger translations



Conduct broader user testing across translations