

# Capstone Project

## Week 5 Progress Report

2/9/2026 - 2/15/2026

Commit range:

- 52c90c810d78b6bb2a90d51da8701d78c1c4b51c →  
f543c57962c212b3283e17db2e07f0ef0eedd01f

See `project-statistics-detailed.md` for LoC and other project statistics.

# Summary

This week focused on:

- Stabilizing server route error handling
- Expanding controller coverage for business & menu resources
- Implementing async route handling ( `asyncHandler` )
- Improving client UX for menu items, allergens, and addresses
- Introducing new middleware, services, and shared components

# Key Outcomes

- Centralized error handling with new `errorHandler` middleware
- Added controllers for:
  - `address`
  - `menu` , `menuItem`
  - `business` , `businessUser`
- Improved client UI:
  - Extracted shared `AddressFields`
  - Added `item_type` support
  - Integrated `allergenCache` into item flows
- Renamed `userRoutes.js` → `businessUserRoutes.js` for domain clarity

# New Files (Server)

- server/src/controllers/menuController.js
- server/src/controllers/menuItemController.js
- server/src/controllers/addressController.js
- server/src/controllers/businessController.js
- server/src/controllers/businessUserController.js
- server/src/middleware/errorHandler.js
- server/src/services/authService.js
- server/src/services/businessMembershipService.js
- server/src/utils/getIdToken.js

## New Files (Client)

- `client/src/components/common/AddressFields.jsx`  
→ Shared address component used across setup & edit flows

# Major File Modifications (Server)

- Applied `asyncHandler` across all routes
- Renamed `userRoutes.js` → `businessUserRoutes.js`
- Updated services:
  - `authService`
  - `businessMembershipService`
  - `addressService`
  - `menuService`
  - `menuItemService`
- Relaxed overly strict ID validation in schemas
- Updated `server.js` to use new middleware & route imports

# Major File Modifications (Client)

- Extracted shared `AddressFields`
- Updated:
  - `EditBusinessInfo`
  - `Setup Step 1`
  - `AddMenuItem`
  - `AllergenList`
- Added `item_type` support
- Integrated `allergenCache` into menu item flows
- Made menu item description optional

# Tests

## Updated / Impacted

- Routes now async
- Schema relaxations change expected validation messages

## Needed Next

- Unit tests for new controllers
- Integration tests for `errorHandler`
- Client tests for:
  - `AddressFields`
  - `allergenCache` integration

# Rationale

## Why these changes?

- **Centralized error handling**
  - Consistent formatting, cleaner controllers
- **Controller/service separation**
  - Clearer architecture, easier testing
- **UX improvements**
  - Shared components reduce duplication and improve maintainability
- **Pragmatic validation**
  - Relaxed ID formats reduce friction with Firestore documents

# Files Touched (Concise)

## Added

- Controllers: `menu` , `menuItem` , `address` , `business` , `businessUser`
- Middleware: `errorHandler`
- Services: `authService` , `businessMembershipService`
- Utils: `getIdToken`
- Client: `AddressFields.jsx`

## Removed / Archived

- Non-MVP multi-menu features in `MenuItemPanel`
- Small admin var in `ProfileIcon`

# Files Touched (Concise)

## Modified

- server/src/routes/\*
- server/src/services/\*
- server/src/schemas/index.js
- server/server.js
- client/src/components/\*
- docs/file-structure.txt

## How to Run / Verify Locally

### 1. Install dependencies

```
npm install
```

### 2. Run

```
npm run dev
```

## Manual Verification Checklist

- Menu creation/edit flows show correct item\_type
- Business setup screens use shared AddressFields
- Server endpoints (/api/menus, /api/menuitems, /api/addresses)
  - Return expected shapes
  - Errors formatted by errorHandler

## Notes & Next Steps

- Update tests for new controllers & error formatting
- Add Firestore emulator integration tests
- Consider memoizing allergenCache for heavy render paths

## Completion Summary

This week improved backend robustness, expanded controller coverage, unified error handling, and enhanced client UX. The system is now more consistent, maintainable, and aligned with the domain model.

## Metrics

### 🔥 Week 5 Burndown Rates

- 6/6 Week 5 requirements completed
  - 100% total
  - ~14% per day

## Metrics (continued)

### 🔥 Sprint 1 Burndown Rates

- 1.5/4 Sprint 1 features completed
  - 37.5% total
  - ~19% per week
  - ~2.7% per day
- 14/41 Sprint 1 requirements completed
  - ~34% total
  - ~17% per week
  - ~2.4% per day