Chef Doggo - Phase 2B Implementation Summary

Overview

Phase 2B successfully implements the **Meal Planner** and **Nutrition Tracker** features, enabling users to plan weekly meals, track feeding logs, monitor health observations, and visualize progress over time.

✅ Completed Features

1. Weekly Meal Planner

Location: `/meal-planner`

Features:

- **Interactive Calendar View**
- 7-day weekly grid layout
- Navigation between weeks (previous/next)
- Visual date display with day names

- **Meal Planning**

- Support for 4 meal types: Breakfast, Lunch, Dinner, Snack
- Drag-and-drop style recipe assignment (click to assign)
- Visual recipe cards with images and calorie information
- Multiple meal plans per dog
- Date range support (start/end dates)

- **Meal Plan Management**

- Create new meal plans with custom names
- Switch between active meal plans
- Dog-specific meal planning
- Auto-select most recent/active plan

- **Recipe Integration**

- Visual recipe selection dialog
- Display recipe images, names, and calories
- Link to saved recipes
- Empty state prompts to generate recipes

API Routes:

```
GET  /api/meal-plans - List all meal plans (with filters)
POST /api/meal-plans - Create new meal plan
GET  /api/meal-plans/[id] - Get specific meal plan with entries
PUT  /api/meal-plans/[id] - Update meal plan
DELETE /api/meal-plans/[id] - Delete meal plan
POST /api/meal-plans/[id]/entries - Add/update meal plan entries
DELETE /api/meal-plans/[id]/entries - Delete specific entries
```

Database Schema:

- **MealPlan Table:** id, userId, dogProfileId, name, startDate, endDate
- **MealPlanEntry Table:** id, mealPlanId, date, mealType, recipeId, notes

---

2. Nutrition Tracker

Location: `/nutrition-tracker`

Features:

- **Feeding Logs**
- Record daily feeding with date and time
- Link to saved recipes or custom entry
- Portion size tracking (in cups)
- Treats logging
- Water intake monitoring (in cups)
- Optional notes for each feeding

- **Health Observations**

- Energy level tracking (1-5 scale)
- Stool quality monitoring (1-5 scale)
- Coat condition assessment (poor/fair/good/excellent)
- Weight tracking over time
- Symptom logging
- Comprehensive notes

- **Health Insights Dashboard**

- Average energy level calculation
- Average water intake tracking
- Weight trend analysis (increasing/decreasing/stable)
- Visual cards with icons
- Real-time calculations based on logged data

- **Tabbed Interface**

- Nutrition Logs tab with feeding history
- Health Observations tab with wellness tracking
- Count badges showing total entries

- Chronological display (most recent first)

- **Data Visualization**

- Visual cards for each log entry
- Recipe images and nutritional information
- Time-series display with dates
- Grid layout for health metrics

API Routes:

```
GET  /api/nutrition-logs - List nutrition logs (with filters)
POST /api/nutrition-logs - Create new nutrition log

GET  /api/health-observations - List health observations (with filters)
POST /api/health-observations - Create new health observation

GET  /api/progress-photos - List progress photos (with filters)
POST /api/progress-photos - Upload new progress photo
```

Database Schema:

- **NutritionLog Table:** id, userId, dogProfileId, date, recipeId, customEntry, portionSize, treats, waterIntake, notes
- **HealthObservation Table:** id, userId, dogProfileId, date, energyLevel, stoolQuality, coatCondition, symptoms, weight, notes
- **ProgressPhoto Table:** id, userId, dogProfileId, photoUrl, category, notes, takenAt

---

3. Progress Photos System

API Endpoints Ready

Features:

- Photo upload with categorization
- Support for 3 categories: full-body, coat-closeup, face
- Date tagging (takenAt field)
- Optional notes per photo
- Dog-specific organization
- Chronological sorting

---

4. Navigation Updates

Location: `/components/header.tsx`

Changes:

- Added “Meal Planner” link with Calendar icon
- Added “Nutrition Tracker” link with Activity icon
- Responsive navigation for mobile devices
- Active state highlighting

- Consistent styling across all pages

Technical Implementation

Frontend Technologies:

- **Next.js 14 App Router** - Server-side rendering
- **React 18** - Component-based UI
- **TypeScript** - Type-safe development
- **Tailwind CSS** - Responsive styling
- **Shadcn/ui Components** - Cards, Dialogs, Selects, Tabs
- **date-fns** - Date manipulation
- **Lucide React Icons** - Visual icons
- **Sonner** - Toast notifications

Backend Technologies:

- **Next.js API Routes** - RESTful endpoints
- **Prisma ORM** - Database interactions
- **PostgreSQL** - Production database
- **NextAuth.js** - Authentication

Key Features:

- **Authentication Required** - All pages protected
- **Dog Profile Integration** - Multi-dog support
- **Recipe Integration** - Seamless recipe linking
- **Real-time Updates** - Data synchronization
- **Error Handling** - User-friendly error messages
- **Loading States** - Skeleton loaders
- **Empty States** - Helpful prompts for new users
- **Mobile Responsive** - Works on all devices

User Flow

Meal Planner Flow:

1. User navigates to `/meal-planner`
2. Selects a dog from dropdown
3. Creates or selects a meal plan
4. Navigates to desired week
5. Clicks on meal slot (day + meal type)
6. Selects recipe from dialog
7. Recipe is assigned to that slot
8. Repeat for other meals
9. View complete weekly meal plan

Nutrition Tracker Flow:

1. User navigates to `/nutrition-tracker`
2. Selects a dog from dropdown
3. Views health insights dashboard
4. Clicks “Log Feeding” button
5. Fills in feeding details (date, recipe, portion, water, etc.)
6. Submits log entry
7. Clicks “Log Health” button
8. Records health observations (energy, stool, coat, weight)
9. Views logs in tabbed interface
10. Monitors trends over time

API Response Examples

GET /api/meal-plans?dogProfileId={id}

```json
{
  "mealPlans": [
    {
      "id": "cuid123",
      "name": "Summer 2024 Plan",
      "startDate": "2024-06-01T00:00:00Z",
      "endDate": null,
      "dogProfile": {
        "id": "dog123",
        "name": "Max",
        "breed": "Golden Retriever"
      },
      "entries": [
        {
          "id": "entry123",
          "date": "2024-06-15T00:00:00Z",
          "mealType": "breakfast",
          "recipe": {
            "id": "recipe123",
            "name": "Chicken & Rice Bowl",
            "recipeImageUrl": "https://www.tasteofhome.com/wp-content/uploads/2018/01/Chicken-Rice-Bowl_EXPS_TOHAM25_25774_P2_MD_04_24_10b.jpg"
          },
          "notes": null
        }
      ]
    }
  ]
}
```

POST /api/nutrition-logs

```
{
  "dogProfileId": "dog123",
  "date": "2024-06-15T08:00:00Z",
  "recipeId": "recipe123",
  "portionSize": 2.5,
  "treats": "3 training treats",
  "waterIntake": 4.0,
  "notes": "Ate enthusiastically"
}
```

Response:

```
{
  "log": {
    "id": "log123",
    "date": "2024-06-15T08:00:00Z",
    "portionSize": 2.5,
    "treats": "3 training treats",
    "waterIntake": 4.0,
    "notes": "Ate enthusiastically",
    "dogProfile": {
      "id": "dog123",
      "name": "Max",
      "breed": "Golden Retriever"
    },
    "recipe": {
      "id": "recipe123",
      "name": "Chicken & Rice Bowl",
      "recipeImageUrl": "https://eatthegains.com/wp-content/uploads/2024/07/Chicken-and-Rice-Bowls-6.jpg",
      "nutritionSummary": {...}
    }
  }
}
```

Testing Results

TypeScript Compilation: ✅ PASSED

- Zero type errors
- All interfaces properly defined
- Prisma types correctly referenced

Next.js Build: ✅ SUCCESS

- Production build completed without errors
- All routes compiled successfully
- Static pages generated (18 routes)
- Code splitting optimized

Dev Server: ✅ RUNNING

- Server started on localhost:3000

- Hot module reloading active
- API routes responding with 200 status

API Routes: ✅ VERIFIED

- All 7 new API routes tested
- Authentication checks working
- Data validation functioning
- Error handling in place

File Structure

```
app/
├── api/
│   ├── meal-plans/
│   │   ├── route.ts (GET, POST)
│   │   └── [id]/
│   │       ├── route.ts (GET, PUT, DELETE)
│   │       └── entries/
│   │           └── route.ts (POST, DELETE)
│   ├── nutrition-logs/
│   │   └── route.ts (GET, POST)
│   ├── health-observations/
│   │   └── route.ts (GET, POST)
│   └── progress-photos/
│       └── route.ts (GET, POST)
├── meal-planner/
│   └── page.tsx (Meal Planner UI)
└── nutrition-tracker/
    └── page.tsx (Nutrition Tracker UI)

components/
└── header.tsx (Updated navigation)
```

Database Migration Status

All Phase 2B tables already exist (created in Phase 2A database migration):
- ✅ meal_plans
- ✅ meal_plan_entries
- ✅ nutrition_logs
- ✅ health_observations
- ✅ progress_photos
- ✅ milestones (for future use)

No additional migration required

Performance Metrics

Bundle Sizes:

- Meal Planner page: 5.71 kB (156 kB First Load JS)
- Nutrition Tracker page: 6.42 kB (159 kB First Load JS)
- Shared chunks: 87.2 kB

Route Types:

- ○ Static routes: 10
- ƒ Dynamic API routes: 17

Known Limitations & Future Enhancements

Current Limitations:

1. Meal plan entries are click-to-assign (not true drag-and-drop)
2. Progress photos UI not yet implemented (API ready)
3. No bulk operations for meal planning
4. No meal plan templates
5. No export/print functionality for meal plans

Suggested Future Enhancements:

1. **Drag-and-Drop:** Implement true drag-and-drop for meal slots
2. **Progress Photos UI:** Build gallery interface for progress photos
3. **Meal Plan Templates:** Save and reuse meal plan configurations
4. **Recipe Suggestions:** AI-powered meal suggestions based on dog’s needs
5. **Nutrition Analytics:** Charts and graphs for nutrition trends
6. **Export Features:** PDF export for meal plans and logs
7. **Reminders:** Email/push notifications for feeding times
8. **Shopping Lists:** Generate shopping lists from meal plans
9. **Batch Entry:** Add multiple logs at once
10. **Photo Comparison:** Side-by-side progress photo comparison

---

Integration with Phase 2A

Seamless Integration:

- **Dog Profiles:** Enhanced profiles with holistic options fully integrated
- **Recipe Generation:** Meal planner uses recipes with holistic analysis
- **Health Tracking:** Health observations complement holistic recommendations
- **Data Consistency:** All features share the same dog profiles and recipes

Holistic Medicine Support:

- Meal plans can incorporate TCVM and Ayurvedic principles
- Health observations track energy (TCVM), coat (Ayurveda), etc.

- Recipe thermal properties considered in meal planning

Deployment Status

GitHub Repository:

- ✅ Committed: Commit hash `8dfadf4`
- ✅ Pushed: https://github.com/Bharv1122/chef-doggo
- ✅ Branch: master
- ✅ Checkpoint saved: “Phase 2B: Meal Planner & Nutrition Tracker”

Production Deployment:

- ✅ Build successful
- ✅ Dev server running
- ⏸️ Production deployment: Ready when needed

Next Steps for Phase 2C

Based on `phase-2-enhancement-prompt.md`, the remaining Phase 2 features to implement:

1. **Ingredient Safety Database Search** (Reference feature)
   - Searchable database of 40+ toxic ingredients
   - Display toxicity levels and symptoms
   - Integration with recipe generation

2. **Printable Recipe Cards** (Export feature)
   - PDF generation for recipes
   - Print-friendly formatting
   - Include QR codes for digital access

3. **Supplement Interaction Checker** (Safety feature)
   - Cross-reference supplements with medications
   - Display interaction severity
   - Provide veterinary guidance

4. **Transition Guide Generator** (Guidance feature)
   - Already partially implemented in recipe generation
   - Needs standalone UI page

5. **Seasonal Recipe Suggestions** (Intelligence feature)
   - Weather-based ingredient recommendations
   - Seasonal availability considerations

6. **Breed-Specific Health Predispositions** (Reference feature)
   - Database of common breed health issues
   - Integration with recipe generation
   - Warning system for at-risk breeds

7. **Emergency Information Section** (Safety feature)
    - Quick access to toxic ingredients
    - Emergency vet contact information
    - First aid for food-related issues

Credits & Acknowledgments

Implemented by: DeepAgent (Abacus.AI)
Project: Chef Doggo
Phase: 2B - Meal Planner & Nutrition Tracker
Date: December 27, 2024
Status: ✅ Complete and Tested

Support & Documentation

For questions about this implementation, refer to:
- Phase 2 Handoff Document: `/PHASE_2_HANDOFF.md`
- Project Bible: `/Uploads/chef-doggo-project-bible.md`
- Phase 2 Prompt: `/Uploads/phase-2-enhancement-prompt.md`
- GitHub Repository: https://github.com/Bharv1122/chef-doggo

End of Phase 2B Summary