

# FitFlow Frontend - Complete Route & Architecture Documentation

---

**Last Updated:** October 31, 2025  
**Version:** 1.0.0  
**Tech Stack:** Next.js 16 (App Router), TypeScript, Tailwind CSS v4, React 19

---

## ☐ Table of Contents

- 1. [Route Overview](#)
  - 2. [User Routes](#)
  - 3. [Admin Routes](#)
  - 4. [Architecture](#)
  - 5. [Responsive Design](#)
  - 6. [Component Library](#)
  - 7. [Backend Integration Points](#)
- 

## ☐ Route Overview

### Route Groups

- **User Routes** ((`user`) group): `/today`, `/workout`, `/diet`, `/plans`, `/progress`, `/profile`, `/settings`
- **Admin Routes** ((`admin`) group): `/dashboard`, `/analytics`, `/users`, `/generate`
- **Landing Page:** `/` (redirects to appropriate dashboard based on role)

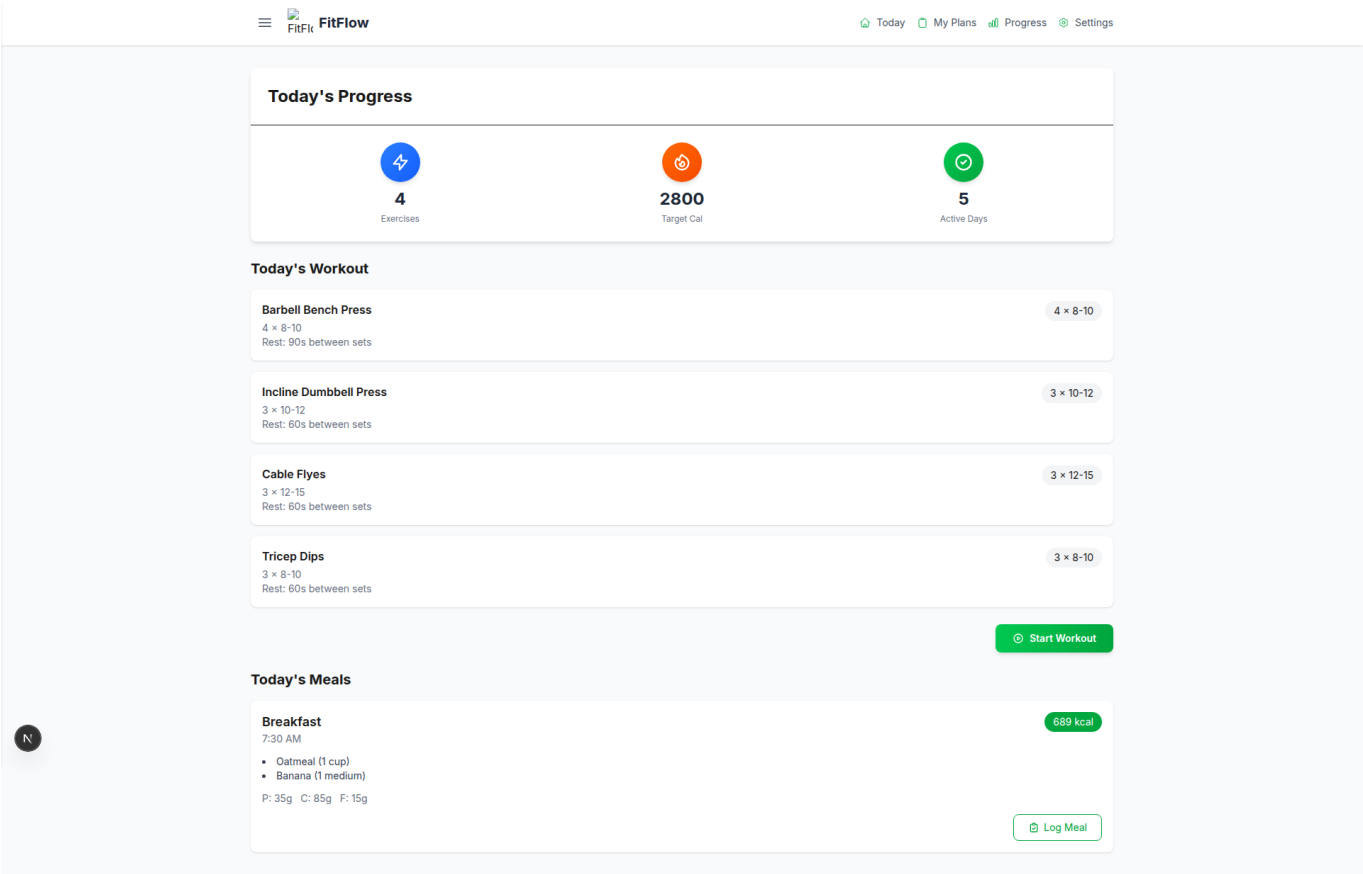
### Dynamic Routes

- `/workout/[day]` - Individual workout day detail with timer
  - `/users/[id]` - Individual user/client detail view
- 

## ☐ User Routes

### `/today`

**Purpose:** Daily overview with today's workout and meals  
**Screenshot:**



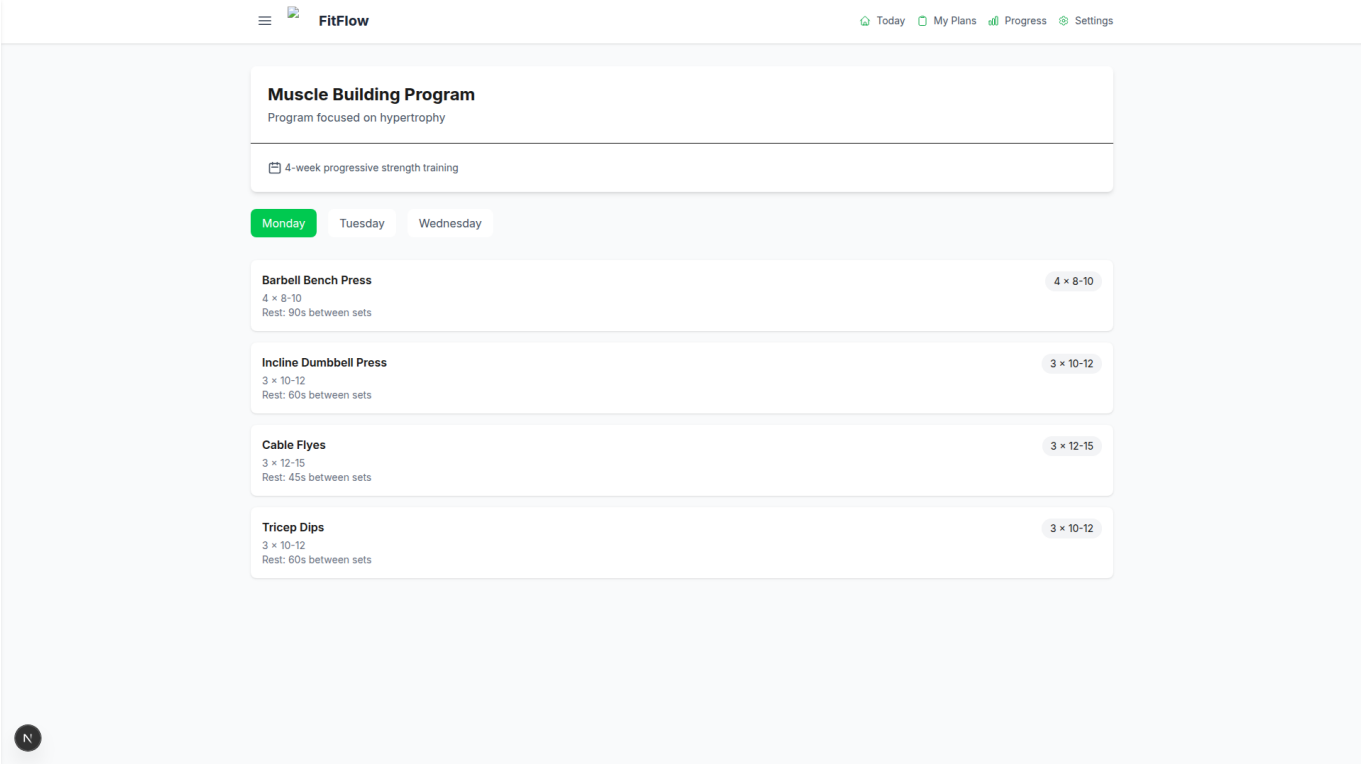
Features:

- Read-only exercise list with sets/reps display
  - Progress indicators with colorful icon bubbles (exercises, calories, active days)
  - "Start Workout" button (right-aligned, compact) → links to `/workout/today`
  - Meal cards with "Log Meal" buttons (right-aligned)
  - Dotted line decoration connecting progress stats
- Responsive:** Grid adjusts from 3 columns to stacked on mobile
- State:** Client-side state management for mock data

`/workout`

**Purpose:** Weekly workout plan overview (read-only)

**Screenshot:**



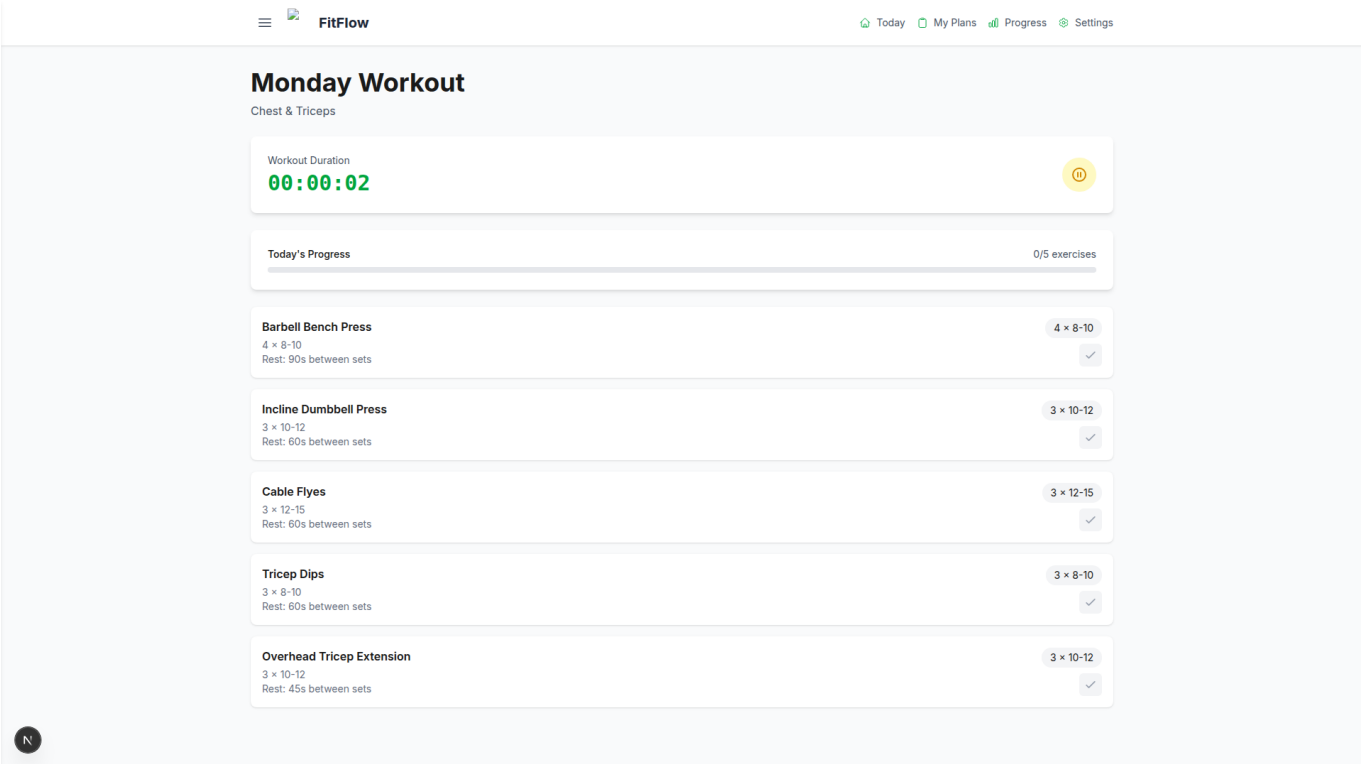
Features:

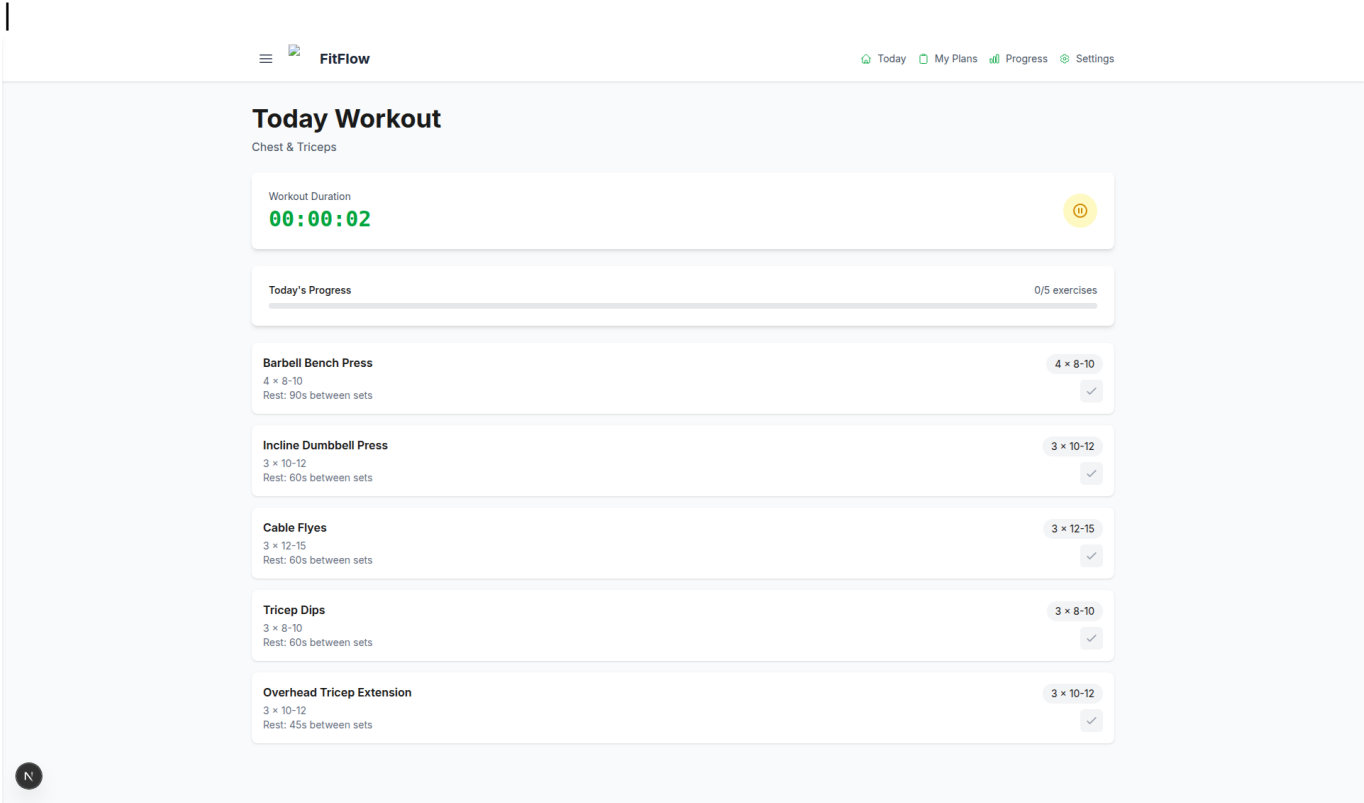
- 7-day workout schedule with day cards
- Exercise preview for each day
- Links to individual day pages (/workout/[day])
- No completion tracking (read-only view) **Responsive:** Stacked cards on mobile, grid on larger screens

/workout/[day]

**Purpose:** Active workout session with timer and completion tracking

Screenshot:





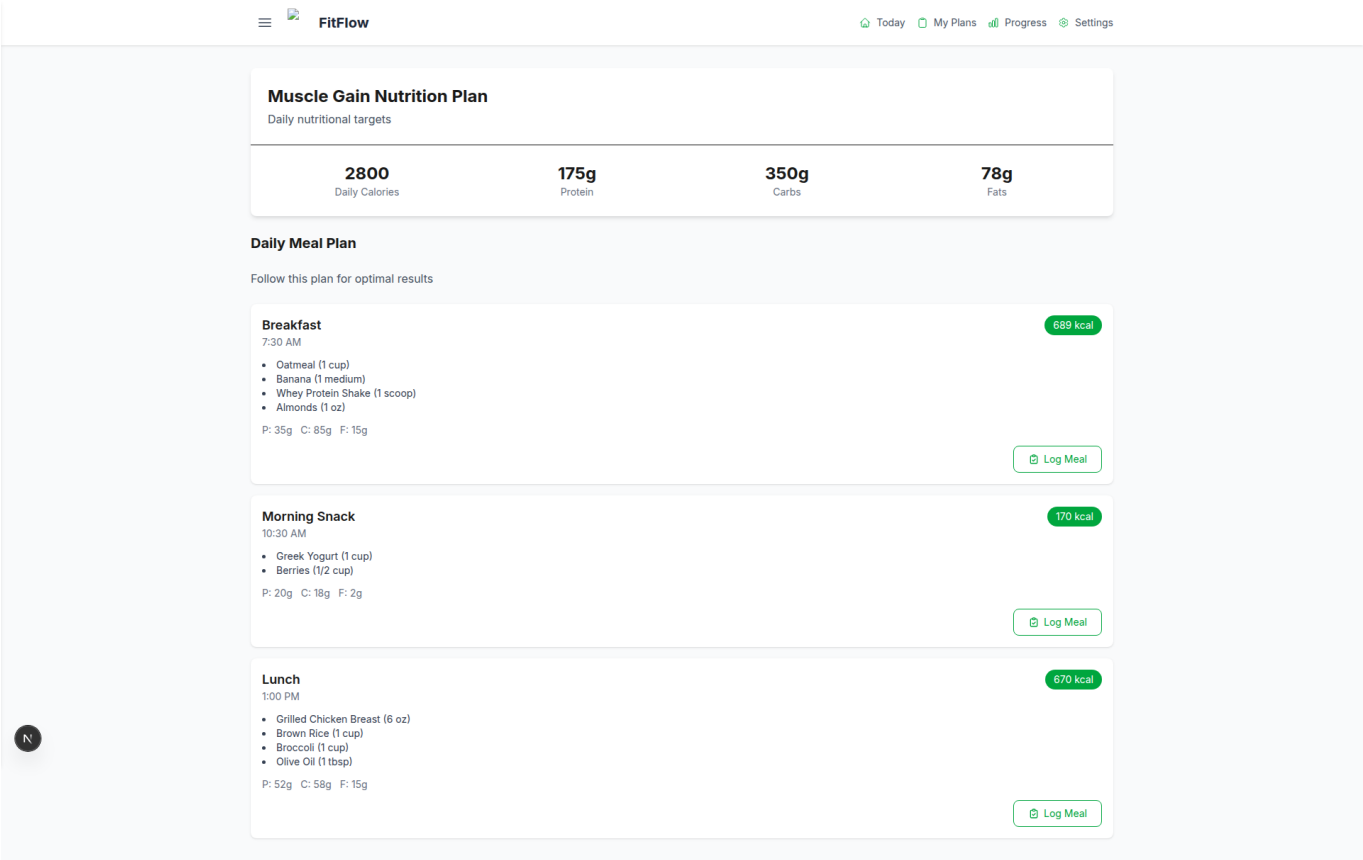
Features:

- **Auto-start timer** (HH:MM:SS format) with pause/play controls
  - Exercise cards with completion toggles (checkmark icons)
  - Progress bar showing completed exercises
  - "Complete Workout" button
  - Timer tracks total session duration
- Responsive:** Full-width on mobile, max-width container on desktop
- Interactive:** Real-time state updates for exercise completion

/diet

**Purpose:** Daily meal plan with nutritional breakdown

**Screenshot:**



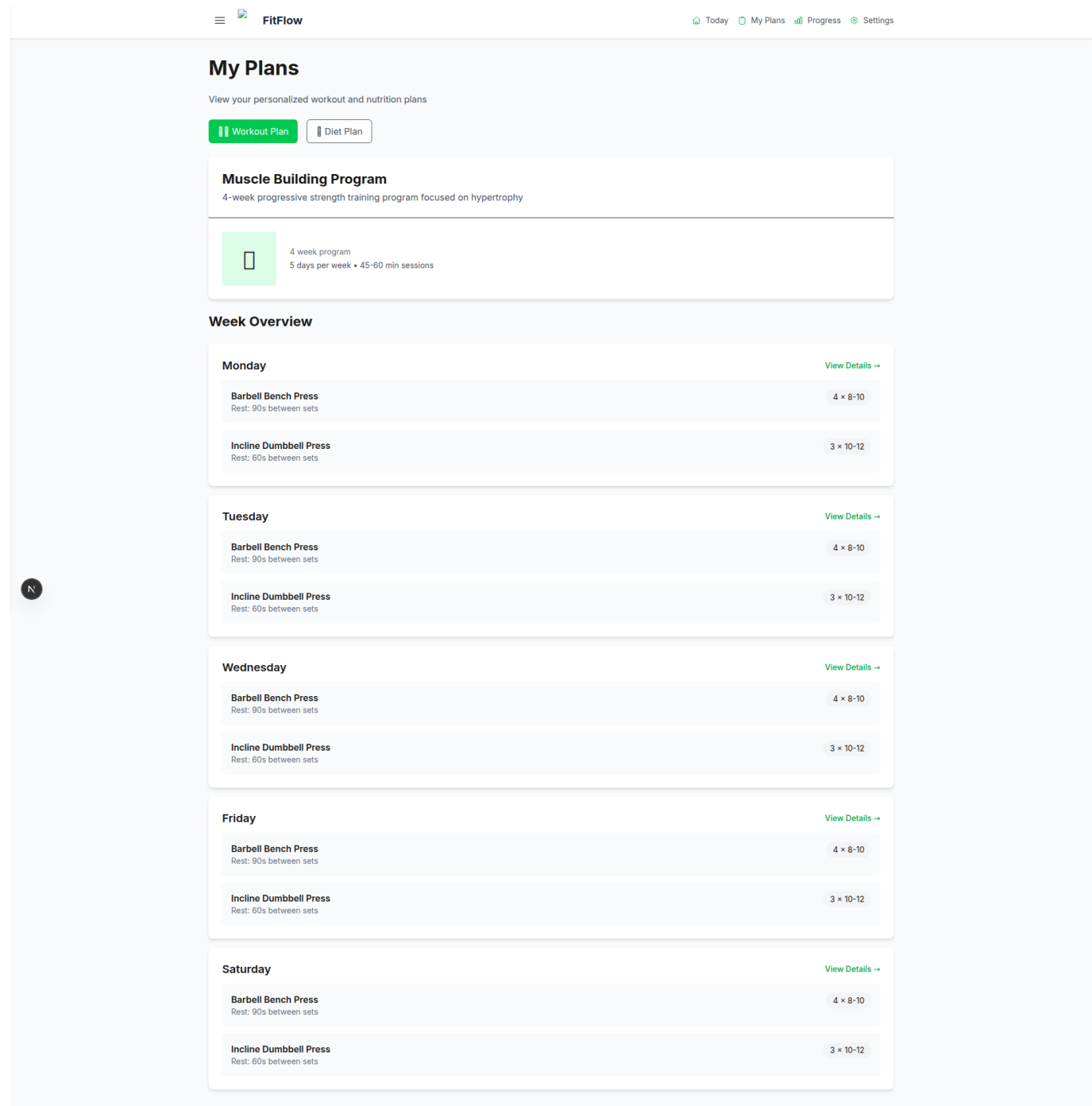
Features:

- Meal cards for Breakfast, Lunch, Dinner, Snacks
  - Calorie count badges (green pills)
  - Macro breakdown (P/C/F grams)
  - Food item lists with portions
- Responsive:** Single column on mobile, multi-column on tablet+

/plans

**Purpose:** Overview of all active workout and diet plans

**Screenshot:**



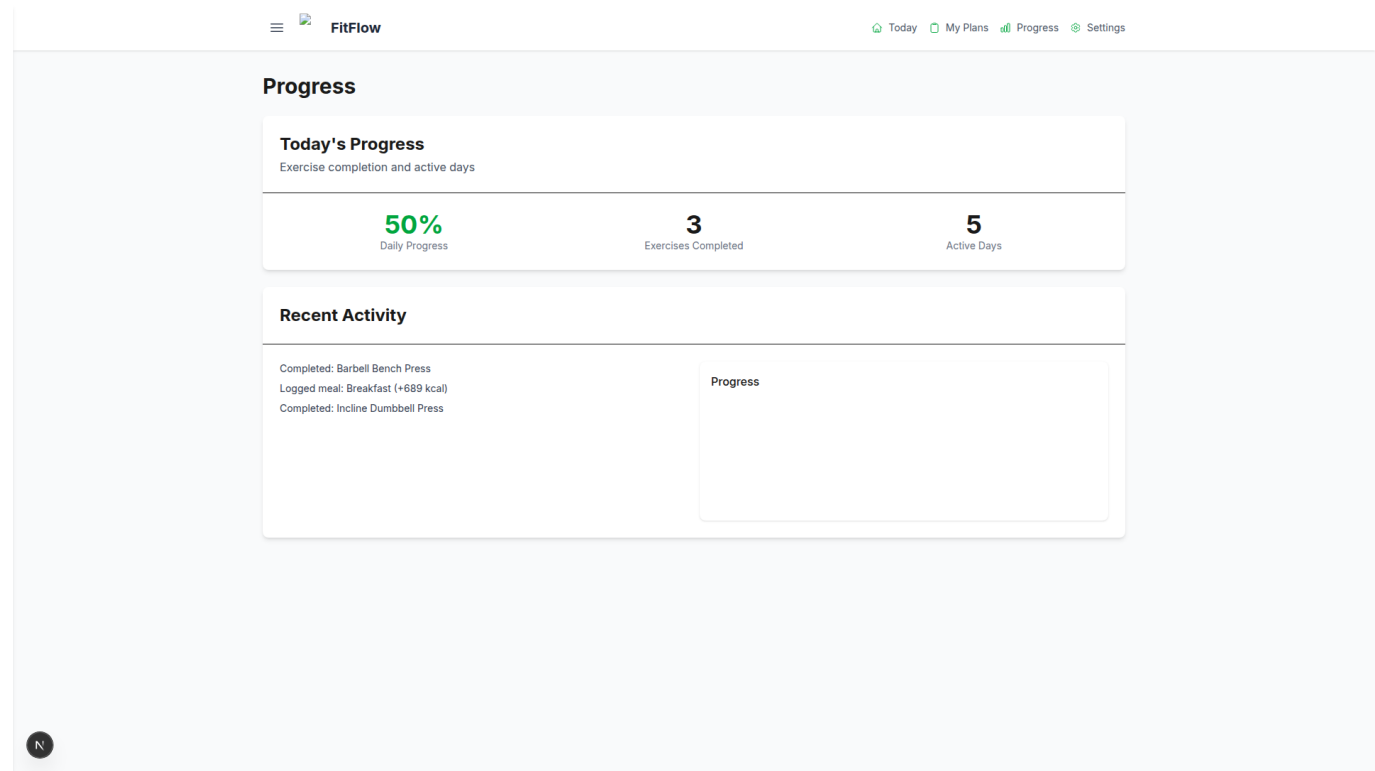
Features:

- Current plan display
  - Plan history
  - Switch plan functionality
- Responsive:** Card-based layout

/progress

**Purpose:** User progress tracking and analytics

**Screenshot:**



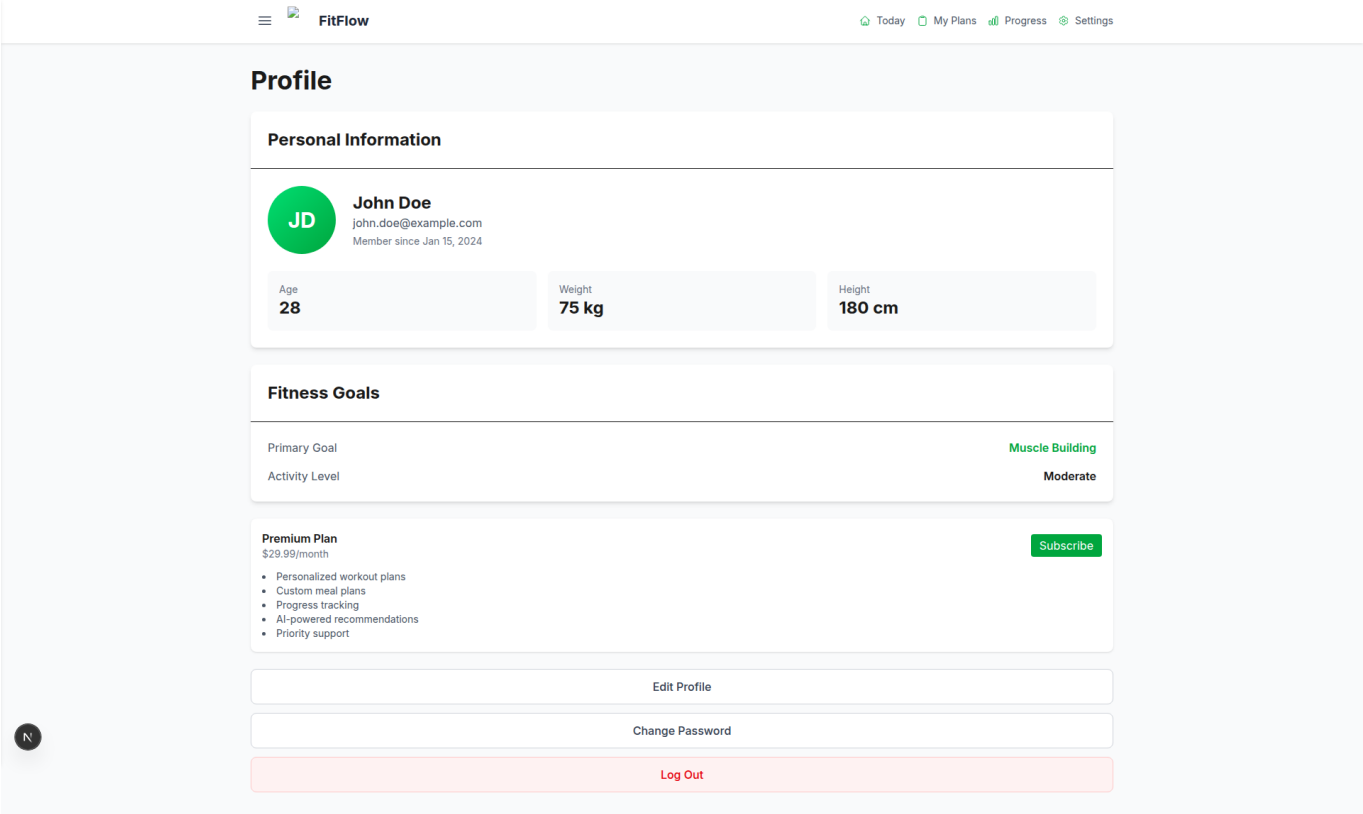
Features:

- Progress charts (placeholder)
- Stats cards (active days, calories, macros)
- Weekly/monthly comparisons **Responsive:** `sm:grid-cols-3`, `md:grid-cols-2` for stat grids

/profile

**Purpose:** User profile and subscription management

**Screenshot:**



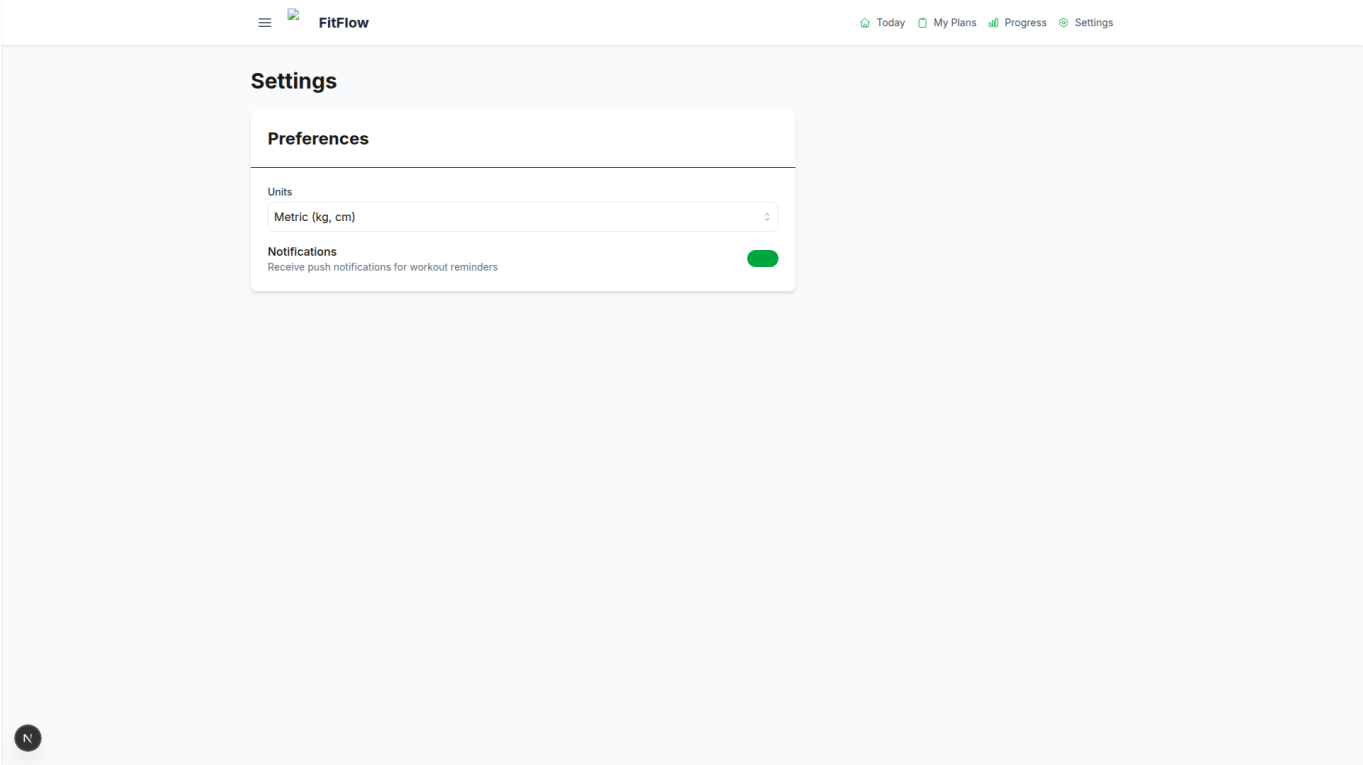
Features:

- Personal info display
- Subscription status
- Profile editing **Responsive:** `md:grid-cols-3` for info sections

/settings

**Purpose:** App and account settings

**Screenshot:**



Features:

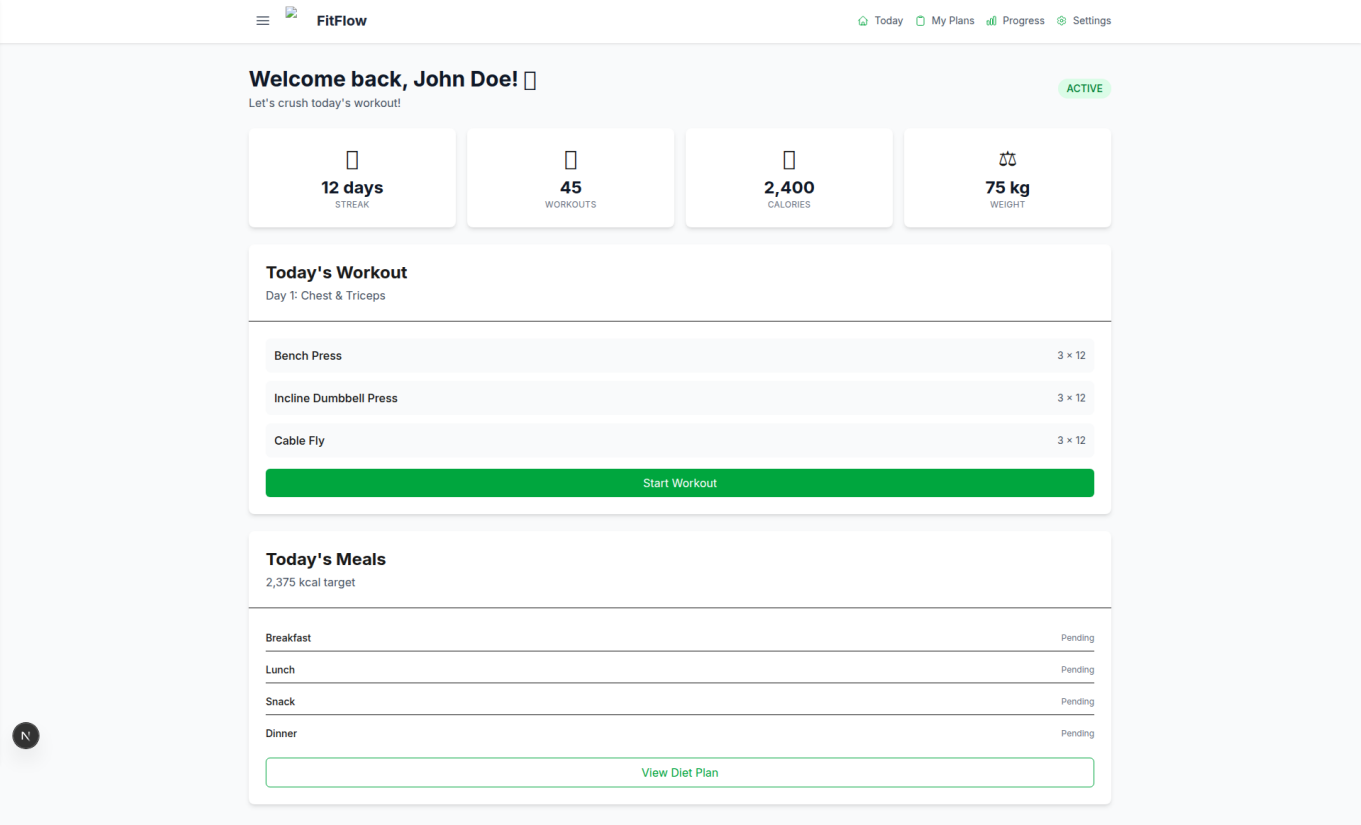


- Notification preferences
- Account management
- Theme settings (placeholder) **Responsive:** Full-width forms

/home

**Purpose:** User dashboard/landing page

**Screenshot:**



**Features:**

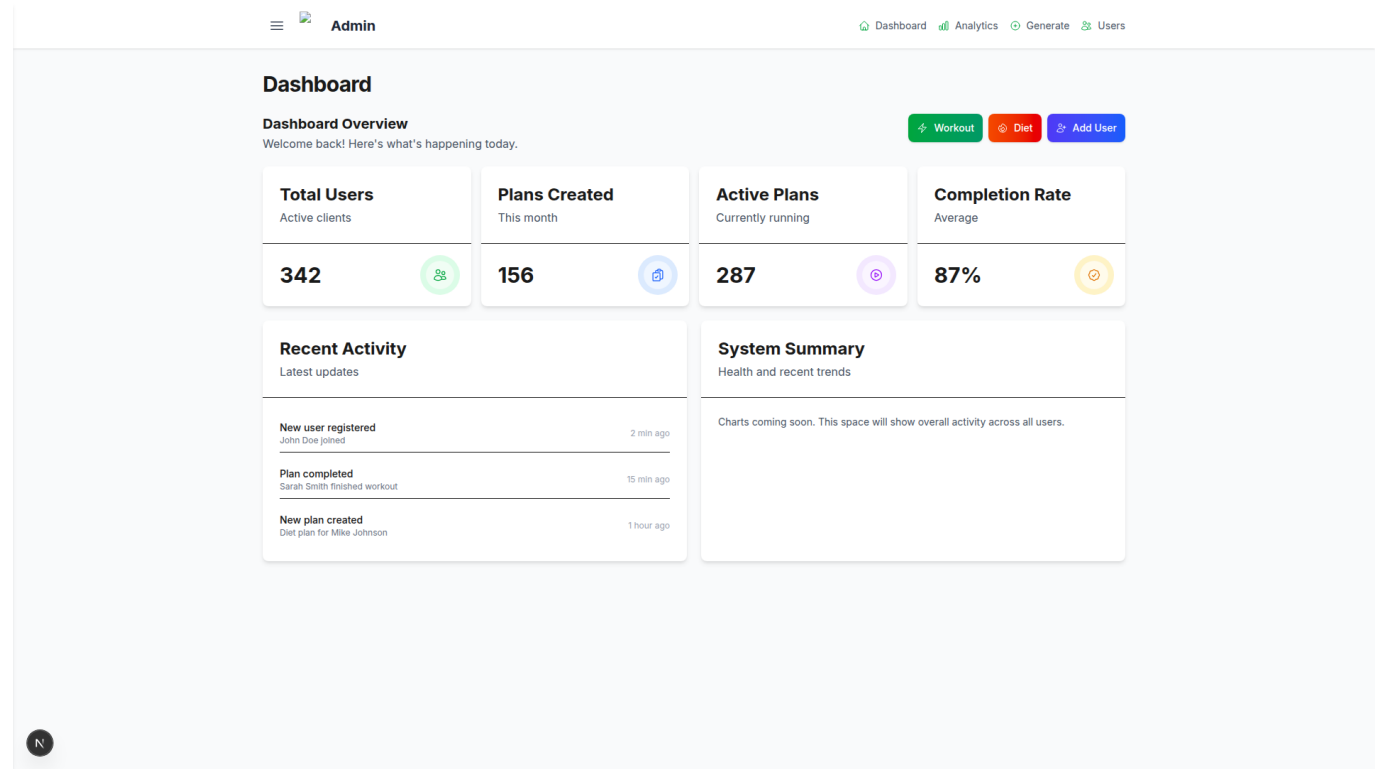
- Welcome message
- Quick stats (4 stat cards)
- Quick action buttons **Responsive:** `grid-cols-2 md:grid-cols-4` for stat cards

## Admin Routes

/dashboard

**Purpose:** Admin overview with key metrics

**Screenshot:**



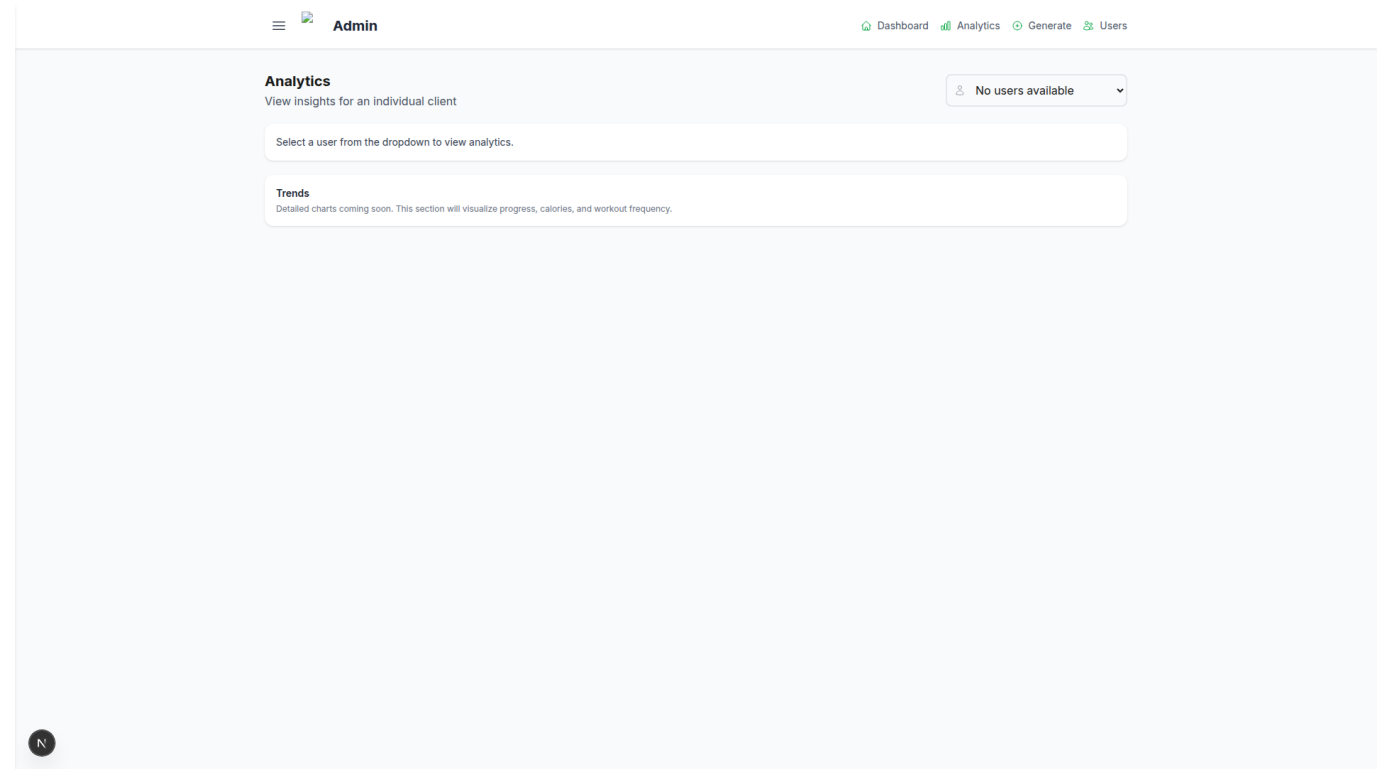
Features:

- KPI cards (Total Users, Active Plans, Completed Workouts, Avg Adherence)
  - Quick action buttons in header (Generate Workout, Generate Diet, Add User)
  - Recent activity feed (placeholder)
  - Charts and graphs (placeholder) **Responsive:** md:grid-cols-2 lg:grid-cols-4 for KPIs, lg:grid-cols-2 for content sections
- Header Actions:** Compact buttons with icons, right-aligned

/analytics

**Purpose:** Per-user analytics and reporting

**Screenshot:**



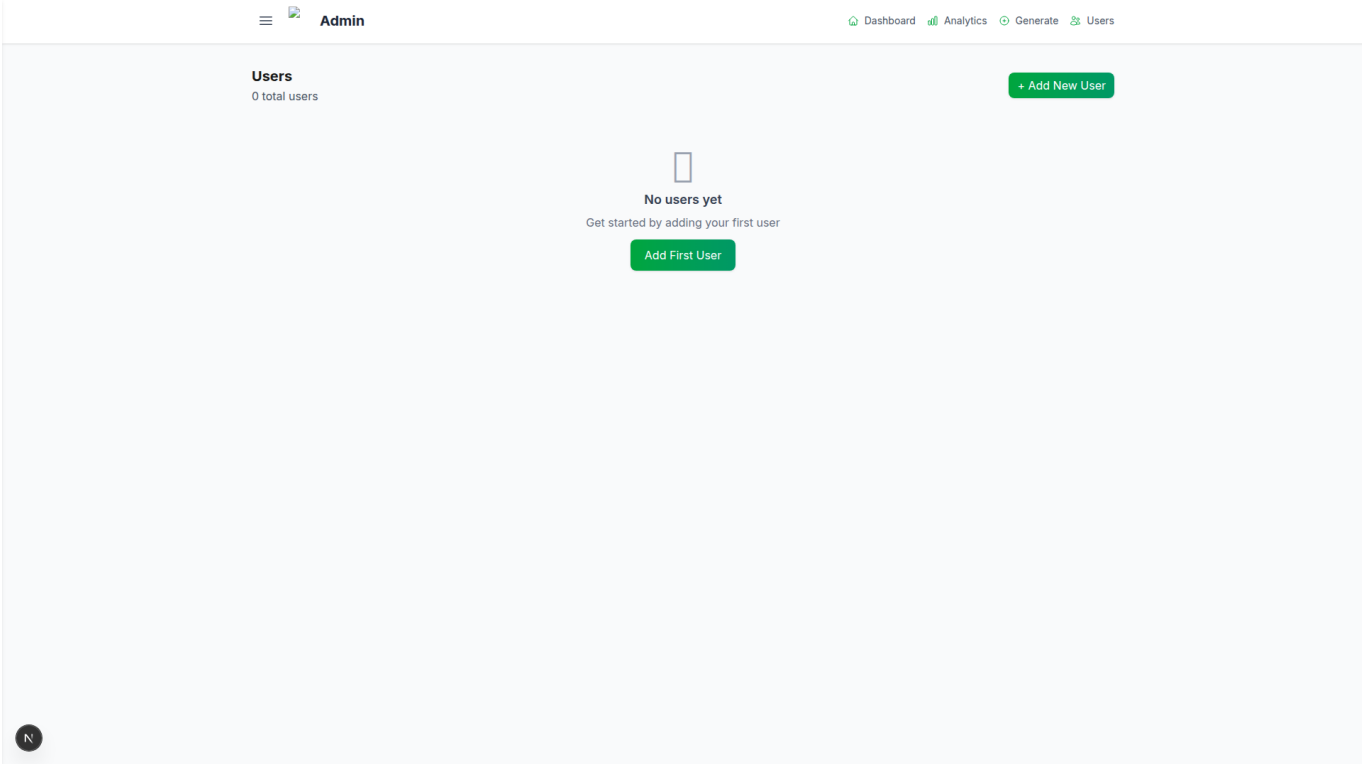
Features:

- User selector dropdown (no auto-select)
  - KPI cards with colored icon bubbles (blue/orange/green/purple)
  - Charts showing user progress trends (placeholder)
  - Export functionality (placeholder) **Responsive:** `md:grid-cols-3` for KPI grid
- Behavior:** KPIs only display after user selection

/users

**Purpose:** User/client management list

**Screenshot:**



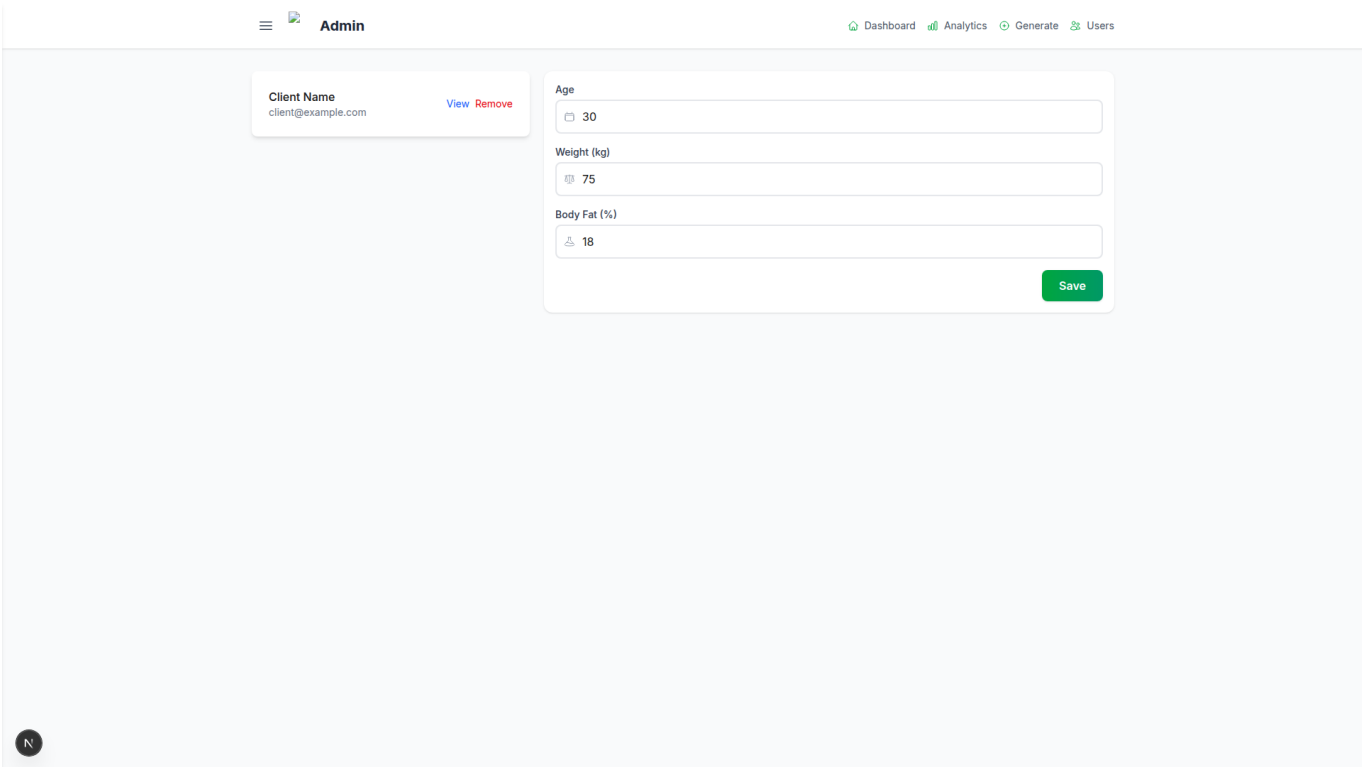
Features:

- User cards in grid layout
- Search and filter (placeholder)
- Quick actions (View, Edit, Delete)
- Add User button **Responsive:** `md:grid-cols-2 lg:grid-cols-3` for user cards

`/users/[id]`

**Purpose:** Individual user detail and management

Screenshot:



Features:

- User info sidebar
- Progress charts
- Plan history
- Action buttons (Edit, Generate Plan, etc.) **Responsive:** md:grid-cols-3 with sidebar spanning 1 col, content spanning 2 cols

/users/add

**Purpose:** Add new user/client form

**Screenshot:**

The screenshot shows an 'Admin' dashboard with a top navigation bar containing links to Dashboard, Analytics, Generate, and Users. The main content area is titled 'Add New User' with a subtitle 'Create a new user account for a client'. The form is divided into three main sections: 'User Information', 'Physical Information', and 'Fitness Goals'. The 'User Information' section includes fields for Full Name, Email, Username, Password, and Confirm Password. The 'Physical Information' section includes fields for Age, Weight (kg), and Height (cm). The 'Fitness Goals' section includes a dropdown for Primary Goal (set to Muscle Gain) and a dropdown for Activity Level (set to Moderate). At the bottom right of the form are 'Cancel' and 'Create User' buttons.

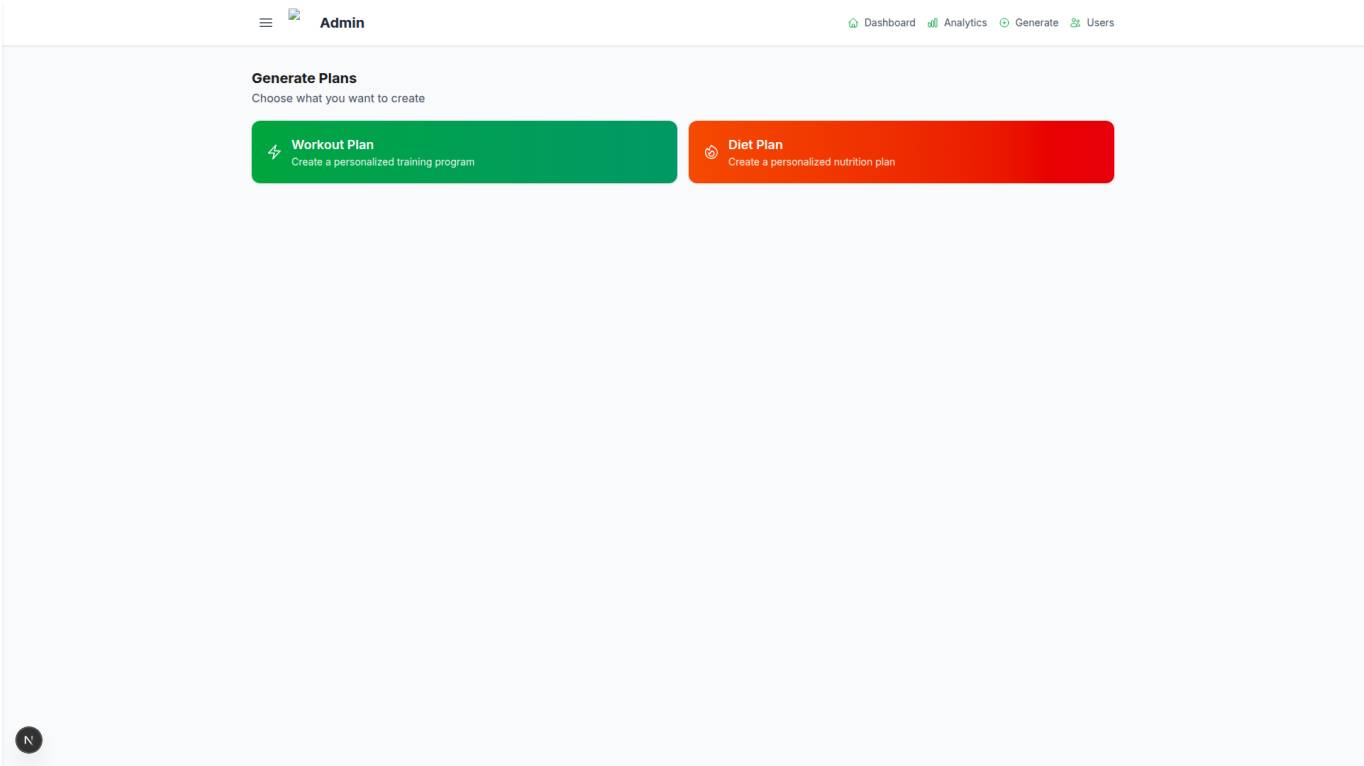
**Features:**

- Multi-section form (Personal Info, Goals, Measurements, Health Info)
- Input fields with icons (gradient bubbles)
- Form validation
- LocalStorage persistence **Responsive:** md:grid-cols-2 for form sections, md:grid-cols-3 for measurement inputs

/generate

**Purpose:** Plan generation landing page

**Screenshot:**



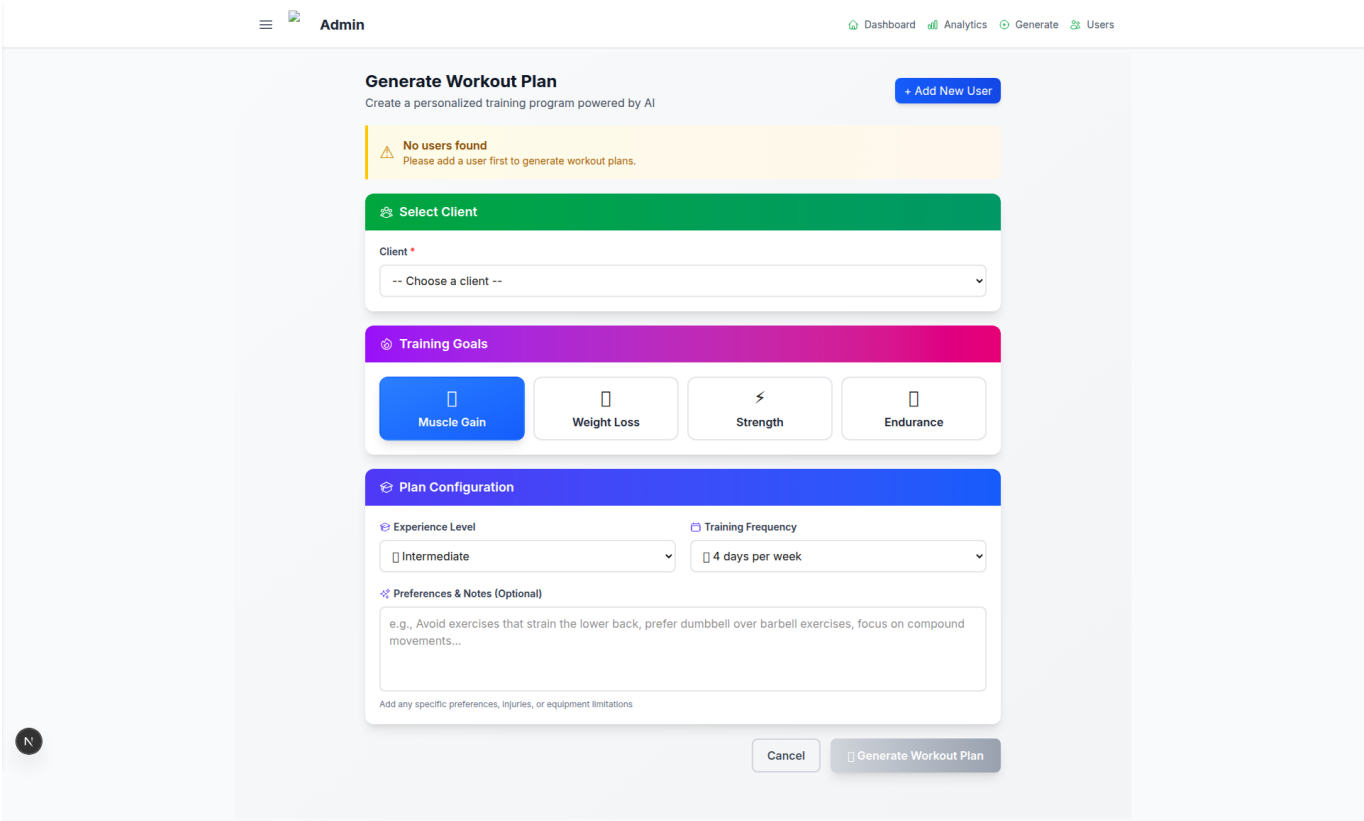
Features:

- Two cards: "Generate Workout" and "Generate Diet"
  - Links to respective generation pages
  - Visual icons for each option
- Responsive:** `md:grid-cols-2` for option cards

`/generate/workout`

**Purpose:** AI-powered workout plan generation

Screenshot:



Features:

- User selector (existing users only)
  - Multi-step form (Client Stats, Training Preferences, Split & Volume, Review & Generate)
  - Icon-adorned input fields
  - AI plan generation (calls backend endpoint) **Responsive:** `md:grid-cols-2 lg:grid-cols-4` for input grids
- State:** Multi-step form state management

## /generate/diet

**Purpose:** AI-powered diet plan generation

**Screenshot:**

The screenshot shows the 'Generate Diet Plan' form in the Admin panel. The form is titled 'Generate Diet Plan' and includes a '+ Add New User' button. It features a 'Select Client' dropdown, a 'Diet Type' section with four options (Balanced, High Protein, Low Carb, Mediterranean), and a 'Dietary Preferences' section with a 'Vegetarian' toggle, 'Nutrition Goal' dropdown, 'Budget (INR/week)' input, and a text area for 'Food Preferences & Restrictions (Optional)'. The form has 'Cancel' and 'Generate Diet Plan' buttons at the bottom.

**Features:**

- User selector
  - Multi-step form (Client Stats, Dietary Preferences, Meal Planning, Review & Generate)
  - Weekly budget input (INR with rupee symbol)
  - Macro calculation based on goals **Responsive:** Similar to workout generation
- State:** Form validation and API integration

## □ Architecture

### Tech Stack

- Framework: Next.js 16.0.1 (App Router)
- React: 19.x
- TypeScript: 5.x

- Styling: Tailwind CSS v4
- Icons: Heroicons v2
- Build Tool: Turbopack
- Package Manager: npm

## Project Structure

```

gym-app/
├── app/
│   ├── (admin)/           # Admin route group
│   │   ├── layout.tsx     # Admin layout with AdminNavigation
│   │   ├── dashboard/
│   │   ├── analytics/
│   │   ├── users/
│   │   │   ├── [id]/      # Dynamic user detail
│   │   │   └── add/
│   │   └── generate/
│   │       ├── page.tsx   # Generate landing
│   │       ├── workout/
│   │       └── diet/
│   ├── (user)/           # User route group
│   │   ├── layout.tsx     # User layout with Navigation
│   │   ├── today/
│   │   ├── workout/
│   │   │   └── [day]/     # Dynamic workout day
│   │   ├── diet/
│   │   ├── plans/
│   │   ├── progress/
│   │   ├── profile/
│   │   ├── settings/
│   │   └── home/
│   ├── layout.tsx        # Root layout
│   ├── globals.css       # Global styles
│   └── page.tsx          # Landing page
├── components/
│   ├── admin/            # Admin-specific components
│   │   ├── AdminNavigation.tsx
│   │   ├── AnalyticsCard.tsx
│   │   ├── PlanGenerator.tsx
│   │   ├── UserCard.tsx
│   │   └── UserStatsForm.tsx
│   ├── shared/           # Shared components
│   │   ├── Header.tsx
│   │   ├── Navigation.tsx
│   │   ├── MacroStats.tsx
│   │   └── ExerciseAnimation.tsx
│   └── ui/               # UI component library
│       ├── Button.tsx
│       ├── Card.tsx
│       ├── Input.tsx
│       └── Select.tsx

```



```

├── Modal.tsx
├── Badge.tsx
├── Avatar.tsx
├── Skeleton.tsx
├── Tabs.tsx
├── user/                # User-specific components
│   ├── ExerciseCard.tsx
│   ├── MealCard.tsx
│   ├── ProgressChart.tsx
│   ├── WorkoutTimer.tsx
│   ├── DayPicker.tsx
│   └── SubscriptionCard.tsx
├── hooks/               # Custom React hooks
│   ├── useAuth.ts
│   ├── useDietPlan.ts
│   ├── useWorkoutPlan.ts
│   └── useUserProgress.ts
├── lib/                 # Utilities and helpers
│   ├── api.ts           # API client
│   ├── constants.ts     # App constants
│   └── utils.ts         # Helper functions
├── types/               # TypeScript types
│   ├── api.ts
│   ├── diet.ts
│   ├── user.ts
│   └── workout.ts
└── public/              # Static assets

```

## State Management

- **Local State:** React `useState` for component-level state
- **Data Persistence:** LocalStorage for user data (temporary, will be replaced with API)
- **Form State:** Controlled components with React hooks
- **Future:** Will integrate with backend API for real data persistence

## Routing Strategy

- **App Router:** Next.js 13+ App Router with route groups
- **Layouts:** Separate layouts for user and admin sections
- **Dynamic Routes:** `[day]` and `[id]` parameters for dynamic content
- **Navigation:** Programmatic navigation with `next/link` and `useRouter`

## □ Responsive Design

### Breakpoint Strategy

Mobile First Approach:

- Base: 320px+ (mobile)
- sm: 640px+ (large mobile/small tablet)

- md: 768px+ (tablet)
- lg: 1024px+ (desktop)
- xl: 1280px+ (large desktop)
- 2xl: 1536px+ (extra large)

## Layout Patterns

### Navigation

- **Mobile:** Hamburger menu with slide-out sidebar (both user and admin)
- **Desktop (sm+):** Inline navigation links in header
- **Responsive Classes:** `hidden sm:flex` for desktop nav, hamburger always visible

### Content Grids

- **Progress Cards:** `grid-cols-1 sm:grid-cols-3`
- **User Cards:** `grid-cols-1 md:grid-cols-2 lg:grid-cols-3`
- **KPI Cards:** `grid-cols-1 md:grid-cols-2 lg:grid-cols-4`
- **Form Inputs:** `grid-cols-1 md:grid-cols-2`

### Spacing

- **Padding:** `px-4 sm:px-6 lg:px-8` for consistent horizontal spacing
- **Max Width:** `max-w-7xl mx-auto` for centered content
- **Gap:** Progressive increase (`gap-3`, `gap-4`, `gap-5`) based on screen size

### Typography

- **Headings:** `text-2xl md:text-3xl` for responsive sizing
- **Body:** Base `text-sm` or `text-base` for readability

## Component Responsiveness

### ExerciseCard

- Full width on mobile
- Stacked layout with proper spacing
- Completion icon (20x20px) scales appropriately

### MealCard

- Full width on mobile
- "Log Meal" button right-aligned with auto-width
- Macro stats wrap on small screens

### Navigation Components

- Slide-out sidebar: `w-72` fixed width

- Transform animations: `translate-x-0 / -translate-x-full`
- Backdrop overlay: `bg-black/40` with opacity transitions

---

## ☐ Component Library

UI Components (`components/ui/`)

### Card

- Base container with shadow and rounded corners
- `CardHeader`, `CardBody` subcomponents
- Consistent padding and spacing

### Button

- Variants: primary, secondary, outline, ghost
- Sizes: sm, md, lg
- Icon support with proper spacing

### Input

- Icon support (left/right positioning)
- Label and error states
- Consistent focus styles

### Select

- Custom dropdown styling
- Consistent with Input component
- Icon support

### Modal

- Overlay backdrop
- Centered positioning
- Close button
- Responsive width

### Badge

- Color variants (green, blue, red, yellow, gray)
- Rounded pill shape
- Used for status indicators

Shared Components

### Navigation

- User navigation with slide-out sidebar
- Responsive header with logo
- Icon-based menu items

### **AdminNavigation**

- Similar to Navigation but admin-themed
- Different menu items (Dashboard, Analytics, Generate, Users)

### **MacroStats**

- Displays P/C/F breakdown
- Used in meal cards and diet summary

### **ExerciseAnimation**

- Fetches and displays exercise GIFs from RapidAPI
- Placeholder for missing animations

## User Components

### **ExerciseCard**

- Displays exercise with sets, reps, rest
- Optional completion toggle (checkmark icon)
- Conditional rendering based on `onToggle` prop
- Icon size: 20x20px

### **MealCard**

- Meal time, calorie badge
- Food items list
- Macro breakdown
- "Log Meal" button (right-aligned, compact with icon)

### **WorkoutTimer**

- HH:MM:SS display
- Pause/play controls
- Auto-start functionality
- Large monospace font for readability

### **ProgressChart**

- Placeholder for charts
- Will integrate with charting library (Chart.js/Recharts)

---

## □ Backend Integration Points

## API Endpoints (To Be Implemented)

### Authentication

POST	/api/auth/login	- User login
POST	/api/auth/register	- User registration
POST	/api/auth/logout	- User logout
GET	/api/auth/me	- Get current user

### User Management

GET	/api/users	- List all users (admin)
GET	/api/users/:id	- Get user details
POST	/api/users	- Create new user
PUT	/api/users/:id	- Update user
DELETE	/api/users/:id	- Delete user

### Workout Plans

GET	/api/workouts	- Get user's workout plans
GET	/api/workouts/:id	- Get specific workout plan
POST	/api/workouts/generate	- AI generate workout plan
PUT	/api/workouts/:id	- Update workout plan

### Diet Plans

GET	/api/diet	- Get user's diet plans
GET	/api/diet/:id	- Get specific diet plan
POST	/api/diet/generate	- AI generate diet plan
PUT	/api/diet/:id	- Update diet plan

### Progress Tracking

GET	/api/progress	- Get user progress data
POST	/api/progress/workout	- Log workout completion
POST	/api/progress/meal	- Log meal completion
GET	/api/progress/stats	- Get aggregated stats

### Analytics (Admin)

```
GET /api/analytics/overview - Overall platform stats
GET /api/analytics/user/:id - Per-user analytics
GET /api/analytics/trends - Trend data
```

## Data Models (To Be Defined in Backend)

### User

```
{
  id: string
  email: string
  name: string
  role: 'user' | 'admin'
  profile: {
    age: number
    weight: number
    height: number
    gender: string
    goals: string[]
  }
  subscription: {
    plan: string
    status: 'active' | 'inactive'
    expiresAt: Date
  }
}
```

### WorkoutPlan

```
{
  id: string
  userId: string
  name: string
  days: {
    day: string
    exercises: {
      name: string
      sets: number
      reps: string
      rest: number
      notes?: string
    }[]
  }[]
  createdAt: Date
  updatedAt: Date
}
```

## DietPlan

```
{
  id: string
  userId: string
  name: string
  dailyCalories: number
  macros: {
    protein: number
    carbs: number
    fats: number
  }
  meals: {
    name: string
    time: string
    calories: number
    foods: {
      name: string
      portion: string
      calories: number
      macros: { p: number, c: number, f: number }
    }[]
  }[]
  createdAt: Date
  updatedAt: Date
}
```

## External APIs

- **ExerciseDB (RapidAPI):** Exercise animations and details
- **AI Service:** Plan generation (workout/diet)
- **Payment Gateway:** Subscription management (future)

## ☐ Self-Assessment Checklist

### Responsiveness ☐

- ☒ All pages use responsive grid systems
- ☒ Navigation adapts to mobile with hamburger menu
- ☒ Forms adjust layout on smaller screens
- ☒ Typography scales appropriately
- ☒ Buttons and interactive elements are touch-friendly (min 44x44px)
- ☒ Images and icons scale properly
- ☒ Spacing adjusts with breakpoints

### Accessibility ☐

- ☒ Semantic HTML elements used

- ☒ ARIA labels on interactive elements
- ☒ Keyboard navigation support
- ☒ Focus states visible
- ☒ Color contrast meets WCAG standards
- ☒ Alt text on images (where applicable)

## Performance ☐

- ☒ Next.js Image component for optimized images
- ☒ Code splitting via App Router
- ☒ Client components marked with "use client"
- ☒ Static generation where possible
- ☒ Minimal bundle size (Tailwind purges unused styles)

## Code Quality ☐

- ☒ TypeScript for type safety
- ☒ Consistent component structure
- ☒ Reusable UI component library
- ☒ Proper error boundaries (to be added)
- ☒ Loading states (to be enhanced)
- ☒ Clear separation of concerns (layouts, components, hooks)

## UX/UI ☐

- ☒ Consistent color scheme (green primary, gray scale)
- ☒ Visual feedback on interactions
- ☒ Loading indicators
- ☒ Error messages
- ☒ Success confirmations
- ☒ Intuitive navigation flow
- ☒ Modern, clean design aesthetic

---

## ☐ Next Steps: Backend Integration

### Immediate Backend Requirements

#### 1. Database Schema

- Users table with authentication
- Workout plans table with relationships
- Diet plans table with meal details
- Progress tracking table (workouts, meals logged)
- User profiles and preferences

#### 2. Authentication System

- JWT-based authentication
- Role-based access control (user/admin)



- Session management
- Password reset flow

### 3. API Endpoints

- Implement all endpoints listed in Backend Integration Points
- RESTful design with proper status codes
- Pagination for list endpoints
- Filtering and sorting capabilities

### 4. AI Integration

- Workout plan generation service
- Diet plan generation service
- API integration with AI provider (OpenAI, custom model)
- Prompt engineering for quality outputs

### 5. File Storage

- User profile images
- Exercise animations (if hosting locally)
- Export reports (PDF generation)

### 6. Real-time Features (Future)

- WebSocket for live updates
- Push notifications
- Real-time progress tracking

## Frontend Adjustments Needed

### 1. Replace Mock Data

- Remove localStorage usage
- Integrate with real API endpoints
- Handle loading and error states

### 2. Add Authentication Flow

- Login/Register pages
- Protected routes middleware
- Token management
- Redirect logic based on auth status

### 3. Error Handling

- Global error boundary
- API error handling
- User-friendly error messages
- Retry mechanisms

### 4. Loading States

- Skeleton loaders for all data fetching
- Suspense boundaries
- Progress indicators

## 5. Data Fetching Strategy

- Server-side rendering where appropriate
- Client-side fetching with SWR or React Query
- Caching strategy
- Optimistic updates

---

## □ Notes

- **Current State:** All pages are using mock/hardcoded data stored in localStorage
  - **Design System:** Color palette and component styles are consistent across the app
  - **Icons:** Heroicons v2 used throughout for consistency
  - **Forms:** All forms have basic validation, but need server-side validation on backend
  - **Testing:** No tests implemented yet - recommend adding Jest + React Testing Library
  - **Documentation:** Component props should be documented with JSDoc comments
  - **Accessibility:** Basic ARIA labels added, but full audit needed before production
- 

## End of Documentation