Simplified Personalized Nutritional Planner MVP - Requirements Document

1. Project Overview

1.1 Project Name

Personalized Nutritional Planner - MVP

1.2 Project Purpose

Develop a simplified web application that generates AI-powered weekly meal plans based on dietary restrictions and personal metrics, featuring voice logging capabilities for food tracking.

1.3 Target Audience

- Health-conscious individuals seeking personalized nutrition guidance
- People with specific dietary restrictions (Keto, Mediterranean, Vegan, Diabetic)
- Busy professionals wanting automated meal planning solutions

1.4 Project Goals

- Deliver core meal planning functionality with 4 dietary options
- Implement voice-based food logging for user convenience
- Create a scalable foundation for future feature expansion
- Validate market demand with minimal viable product

2. Technology Stack & Architecture

2.1 Frontend

- **Framework**: React JS (Latest stable version)
- **Hosting**: Netlify (Free tier)
- **Styling**: CSS Modules or Styled Components
- **State Management**: React Context API or Redux Toolkit

2.2 Backend

- **Framework**: Laravel (Latest LTS version)
- **Hosting**: Railway (Free tier - 512MB RAM, $5 monthly credit)
- **API**: RESTful API architecture
- **Authentication**: Laravel Sanctum for API token management

2.3 Database

- **Database**: PostgreSQL (Railway's included database)
- **ORM**: Laravel Eloquent
- **Migration Strategy**: Laravel migrations for version control

2.4 External Integrations

- **Voice Processing**: Web Speech API (browser-native) + Google Speech-to-Text API fallback
- **Nutrition Data**: USDA FoodData Central API or Edamam Nutrition API
- **Deployment**: Netlify for frontend, Railway for backend

3. Functional Requirements

3.1 User Management

3.1.1 User Registration

- Email-based registration with email verification
- Basic profile setup: name, age, gender, weight, height
- Goal selection: weight loss, maintenance, muscle gain, health management

3.1.2 User Authentication

- Secure login/logout functionality
- Password reset capability
- Session management with token-based authentication

3.2 Body Metrics & Goal Setting

3.2.1 BMI/BMR Calculator

- Automatic BMI calculation based on height/weight
- BMR calculation using Mifflin-St Jeor equation
- Activity level selection (4 options: sedentary, lightly active, moderately active, very active)

3.2.2 Goal Configuration

- Target weight setting

- Timeline specification (weeks/months)
- Dietary preference selection from 4 options:
    - Keto (high fat, low carb)
    - Mediterranean (balanced, heart-healthy)
    - Vegan (plant-based)
    - Diabetic-friendly (low glycemic index)

3.3 AI Meal Plan Generator

3.3.1 Weekly Meal Planning

- Generate 7-day meal plans with breakfast, lunch, dinner
- Nutritionally balanced meals based on user's dietary preference
- Calorie targets aligned with BMR calculations and goals
- Recipe suggestions with ingredient lists

3.3.2 Smart Scheduling

- Lifestyle pattern recognition (busy weekdays vs. relaxed weekends)
- Meal prep suggestions for time-constrained users
- Alternative meal options for dietary flexibility

3.3.3 Grocery List Generation

- Automatic shopping list creation from selected meal plans
- Ingredient quantity calculations
- Optional categorization by grocery store sections

3.4 Voice Food Logging

3.4.1 Voice Recognition

- Voice-to-text conversion for meal logging
- Natural language processing for food identification
- Support for common phrases: "Log my breakfast: scrambled eggs with toast"

3.4.2 Food Database Integration

- Comprehensive food database with nutritional information
- Macro tracking (calories, protein, carbs, fats)
- Portion size estimation and conversion

3.4.3 Daily Tracking

- Daily calorie and macro consumption tracking
- Progress visualization with simple charts

- Weekly/monthly progress summaries

3.5 Dashboard & Analytics

3.5.1 User Dashboard

- Current meal plan display
- Daily nutrition progress
- Goal progress tracking
- Quick access to voice logging

3.5.2 Progress Analytics

- Weight progress charts
- Nutritional adherence metrics
- Meal plan completion rates
- Achievement badges and milestones

4. Non-Functional Requirements

4.1 Performance Requirements

- Page load time: < 3 seconds on 3G connection
- Voice recognition response time: < 2 seconds
- Meal plan generation: < 5 seconds
- API response time: < 1 second for standard requests

4.2 Scalability Requirements

- Support for 1,000+ concurrent users in MVP phase
- Database designed for horizontal scaling
- API rate limiting to prevent abuse
- Caching strategies for frequently accessed data

4.3 Security Requirements

- HTTPS encryption for all data transmission
- Password hashing with bcrypt
- API token-based authentication
- Input validation and sanitization
- CORS configuration for secure cross-origin requests

4.4 Usability Requirements

- Mobile-responsive design (mobile-first approach)
- Intuitive navigation with < 3 clicks to key features
- Voice logging accessible within 1 click from any page
- Progressive Web App (PWA) capabilities for mobile users

4.5 Reliability Requirements

- 99.5% uptime target
- Graceful error handling with user-friendly messages
- Data backup and recovery procedures
- Offline functionality for basic meal plan viewing

5. User Stories & Acceptance Criteria

5.1 User Registration & Setup

As a new user, I want to create an account and set up my profile so that I can receive personalized meal plans.

Acceptance Criteria:

- User can register with email and password
- User receives email verification
- User completes profile setup with basic metrics
- System calculates BMI/BMR automatically
- User selects dietary preference and goals

5.2 Meal Plan Generation

As a registered user, I want to generate a weekly meal plan based on my dietary preferences so that I can follow structured nutrition guidance.

Acceptance Criteria:

- User can select from 4 dietary options
- System generates 7-day meal plan with 3 meals per day
- Meals meet caloric and macro requirements
- User can regenerate plans with different options
- Grocery list is automatically created

5.3 Voice Food Logging

As a user, I want to log my meals using voice commands so that I can quickly track my nutrition without manual typing.

Acceptance Criteria:

- Voice logging button is prominently displayed
- System accurately converts speech to text
- Food items are recognized and added to daily log
- Nutritional information is automatically calculated
- User can edit voice-logged entries

5.4 Progress Tracking

As a user, I want to view my nutritional progress so that I can stay motivated and adjust my eating habits.

Acceptance Criteria:

- Dashboard shows daily calorie/macro progress
- Weekly and monthly progress charts are available
- Goal achievement is clearly indicated
- Historical data is accessible
- Progress can be shared or exported

6. Technical Specifications

6.1 Database Schema

Users Table:

```
- id (Primary Key)
- email (Unique, Required)
- password_hash (Required)
- name (Required)
- age, gender, weight, height
- activity_level, dietary_preference
- target_weight, target_timeline
- created_at, updated_at
```

Meal Plans Table:

```
- id (Primary Key)
- user_id (Foreign Key)
- week_start_date
- dietary_type
- meals_json (JSON field with meal data)
- created_at, updated_at
```

Food Logs Table:

```
- id (Primary Key)
- user_id (Foreign Key)
- food_name, quantity, unit
- calories, protein, carbs, fats
- logged_at, created_at
```

6.2 API Endpoints

Authentication:

- POST /api/register
- POST /api/login
- POST /api/logout
- POST /api/password/reset

User Management:

- GET /api/user/profile
- PUT /api/user/profile
- GET /api/user/metrics

Meal Planning:

- POST /api/meal-plans/generate
- GET /api/meal-plans/{id}
- GET /api/meal-plans/current
- GET /api/grocery-list/{meal_plan_id}

Food Logging:

- POST /api/food-logs
- GET /api/food-logs/daily
- PUT /api/food-logs/{id}
- DELETE /api/food-logs/{id}

6.3 Frontend Components Structure

```
src/
├── components/
│   ├── Auth/
│   ├── Dashboard/
│   ├── MealPlanning/
│   ├── VoiceLogging/
│   ├── Progress/
│   └── Common/
├── pages/
├── hooks/
├── services/
├── utils/
└── styles/
```

7. Development Phases & Timeline

Phase 1: Core Foundation (Weeks 1-4)

- Project setup and environment configuration
- User authentication system
- Basic user profile management
- Database schema implementation

Phase 2: Meal Planning Engine (Weeks 5-8)

- BMI/BMR calculation system
- Meal plan generation algorithm
- Recipe database integration
- Grocery list functionality

Phase 3: Voice Logging (Weeks 9-10)

- Voice recognition implementation
- Food database integration
- Daily tracking system
- Voice-to-nutrition conversion

Phase 4: UI/UX & Analytics (Weeks 11-12)

- Dashboard development
- Progress tracking charts
- Mobile responsiveness
- Performance optimization

Phase 5: Testing & Deployment (Weeks 13-14)

- Unit and integration testing
- User acceptance testing
- Deployment to production
- Performance monitoring setup

8. Success Metrics & KPIs

8.1 Technical Metrics

- System uptime: > 99.5%
- Average API response time: < 1 second
- Voice recognition accuracy: > 90%
- Page load time: < 3 seconds

8.2 User Engagement Metrics

- User registration conversion: > 15%
- Daily active users retention: > 30% after 7 days
- Meal plan generation frequency: > 2 per week per active user
- Voice logging usage: > 60% of active users

8.3 Business Metrics

- Monthly active users: 500+ within 3 months
- User acquisition cost: < $10
- Freemium to premium conversion: > 5%
- Average session duration: > 3 minutes

9. Risk Assessment & Mitigation

9.1 Technical Risks

Risk: Voice recognition accuracy issues
Mitigation: Implement fallback manual entry, continuous model training

Risk: Third-party API limitations (nutrition data)
Mitigation: Multiple API providers, local database caching

Risk: Performance issues with meal plan generation
Mitigation: Pre-computed meal templates, background processing

9.2 Business Risks

Risk: Low user adoption
Mitigation: Comprehensive user testing, iterative improvements

Risk: Competition from established apps
Mitigation: Focus on unique voice logging USP, superior UX

10. Post-MVP Roadmap

Phase 2 Features

- Wearable device integration
- Advanced AI coaching
- Social features and meal sharing
- Premium analytics dashboard

Phase 3 Features

- Nutritionist consultation integration
- Meal delivery service partnerships
- Advanced dietary restriction support
- Mobile app development

This requirements document provides a comprehensive foundation for developing the Simplified Personalized Nutritional Planner MVP with the specified tech stack (React JS + Laravel + PostgreSQL) hosted on Netlify and Railway.