**Landing Page:**

Header/Hero Section CTA(Call to Action):

"Get Started" leads to a quiz to tailor recipe recommendations and diet plans.

"Create Your Diet Plan" opens the recipe page consisting a questionnaire about goals, preferences, and allergies.

Register/Login Options:

"Register" and "Login" buttons in the header section.

Registration Page:

Collect essential information like name, email, password, age and dietary preferences.

Welcome Upon Success:

Upon successful registration, redirect users to the family member details page.

Login Page:

Forgot Password Option:

Includes a link for password recovery to assist returning users.

Redirect to Family Details:

Upon successful login, also redirects users to the family member details page.

Family Member Details Page:

Clear Instructions:

Provides clear guidance on how to add family members and their details.

Intuitive Form:

A user-friendly form for entering information (name, gender, age, preferences).

Allow for multiple family member additions with options to remove or edit.

Save and Continue:

Offer a prominent "Save" or "Continue" button to store information and progress.

**Recipes Page:**

Category filters: Clicking filters like "vegetarian", "easy", or "quick-to-make" refreshes recipe lists dynamically.

Recipe cards:

Images open a larger, zoomable view with recipe details.

Titles link to individual recipe pages with full instructions, nutritional information, and reviews.

"Add to My Plan" button lets users save recipes to their personalized diets.

Search bar: Allows users to find specific ingredients or recipe names.

**Diet Plans Page:**

Plan options: Clicking different plan types (weight loss, muscle gain, etc.) reveals detailed descriptions and benefits.

Sample weeks: Clicking "See Sample Week" opens a calendar with meal suggestions and grocery lists.

Customization options: Clicking "Customize this Plan" leads users to adjust meal preferences, allergies, and cooking frequency.

"Start My Plan" button: Initiates the subscription process or offers a free trial option.

**Additional Click Streams:**

Blog posts: Titles and featured images entice users to click and read full articles.

Email campaigns: CTAs within emails, like "Shop Now" or "Download Recipe", lead to relevant landing pages.

Social media posts: Links in captions direct users to specific website sections or recipes.

**Core technologies:**

HTML: For website structure and semantics.

CSS: Sass or LESS for modularity and maintainability in complex styles.

JavaScript: jQuery for common DOM manipulation tasks.

**Front-end frameworks:**

React: A powerful and popular choice for building complex and dynamic UIs with reusable components. Redux for state management and React Router for navigation.

**Interactive elements and animations:**

JavaScript libraries: GSAP, Anime.js, or Velocity.js for animations.

CSS frameworks: Bulma or Tailwind CSS for pre-built UI components and styling utilities.

Motion libraries: Consider Framer Motion for declarative animation workflows and state management.

**Multi-platform support:**

Responsive design: Bootstrap or Foundation to ensure that the website adapts to different screen sizes and devices.

Progressive Web Apps (PWAs): PWA techniques like service workers and offline caching for a native-like app experience.

Hybrid mobile app development: If you're considering future mobile apps, frameworks like React Native or Ionic can help build cross-platform mobile apps with your existing web knowledge.

**User authentication:**

OAuth and OpenID Connect: Popular protocols for secure user authentication with third-party providers like Google or Facebook.

JWT (JSON Web Token): A lightweight and stateless approach for user authorization and data exchange.

**Real-time updates (future):**

WebSockets: Enable bi-directional communication between client and server for real-time data updates.

GraphQL: A query language for fetching data from an API efficiently, often used with real-time applications.

**Additional resources:**

Version control systems (Git): For code collaboration and tracking changes.

Package managers (npm, yarn): To manage dependencies and libraries.

