**Odoo X CHARUSAT - IDEA DOCUMENTATION**

**Team Name : Syntax Sorcerers**

1. **Member Details:**

1.Mayank Tumbadiya - (Backend Developer, Project Manager)

2.Patel Shail – (Frontend Developer)

3.Sheth Tirth – (Graphic Designer)

4.Bhadaliya Manthan – (Backend Developer)

1. **Problem Statement:**

* **Chosen Problem:**
* Many people struggle with maintaining a healthy diet due to a lack of personalized nutrition guidance and real-time dietary monitoring.
* **Problem Analysis:**
* Factors such as age, weight, activity level, and fitness goals influence individual calorie needs.
* Many users either lack the knowledge to plan meals accordingly or find it time-consuming to search for appropriate recipes.
* This leads to unhealthy eating habits and an increased risk of nutrition-related diseases.
* A platform that automatically suggests meals based on daily calorie requirements can simplify meal planning and promote healthier eating habits.

* **Target Audience**
  + - **Health-conscious individuals**
    - **Fitness enthusiasts**
    - **Patients managing dietary restrictions**
    - **Corporate wellness programs**

1. **Solution Overview:**

* Brief Explanation:

Our website provides personalized meal suggestions and recipes based on users' daily calorie goals, helping them stay within their calorie intake for weight loss, maintenance, or muscle gain.

* Approach:

Users input their calorie goal, and the platform automatically suggests meals with detailed recipes, addressing the challenges of time-consuming meal planning and lack of personalization.

* Uniqueness:

Unlike other solutions, our platform combines calorie-based meal recommendations with tailored recipes, offering a simple, one-stop solution for users to achieve their dietary goals.

1. **Frameworks/Technologies**

* Frontend: HTML, CSS, JavaScript (React for dynamic interfaces).
* **Reason**: Provides responsive design and dynamic user interfaces for a smooth experience.
* Backend: Node.js
* **Reason:** Efficient for handling real-time requests and building scalable applications.
* Database: MongoDB for storing user data, recipes, and meal plans.
* **Reason**: For it’s Flexible-oriented data model which allows developers to store complex and data-structure easily
* APIs: Integrate third-party APIs for nutrition data (e.g., USDA FoodData Central, Spoonacular, EDAMAN).
* **Reason:** Flexible, scalable storage for user data, recipes and meal plans.
* Hosting: Use platforms like AWS, Google Cloud, or Vercel for hosting.
* **Reason:** Reliable and scalable platforms for hosting the app with easy deployment.

1. **Feasibility and Implementation**

* **Implementation Ease:**
* Development is straightforward using common technologies (HTML, CSS, JavaScript, Node.js).
* Easy deployment on platforms like Vercel or Netlify with cloud databases like MongoDB for scalability.
* **Effectiveness:**
* Provides personalized meal suggestions based on calorie goals, solving the problem of manual meal planning.
* Ensures users stay within their calorie intake for weight loss, muscle gain, or maintenance, offering an efficient, user-friendly solution.

1. **UI/UX MOCKUP**

* **Screen Overview:**
* **User Flow:**
* **Landing & Attraction –** Users are welcomed with an engaging tagline and a high-quality food image.
* **Navigation & Exploration –** The top menu provides access to "Menu," "Services," and "Contact" sections.
* **Call-to-Action (CTA) –** Users are encouraged to click "View Menu" or "Booking Now" to explore further.
* **Conversion & Engagement** – Users are guided towards booking or ordering food.
* **Accessibility Considerations:**
* **High Contrast –** White and green text on a dark background improves readability.
* **Large, Readable Fonts** – Bold typography ensures easy reading for all users
* **Clear CTA Buttons** – Prominent green buttons help users take action quickly.
* **Minimalist Layout** – Simple and clutter-free design enhances focus and usability.
* **Alt Text & Semantic HTML** – Ensuring screen reader compatibility.

1. **Business Scope and Use Case**

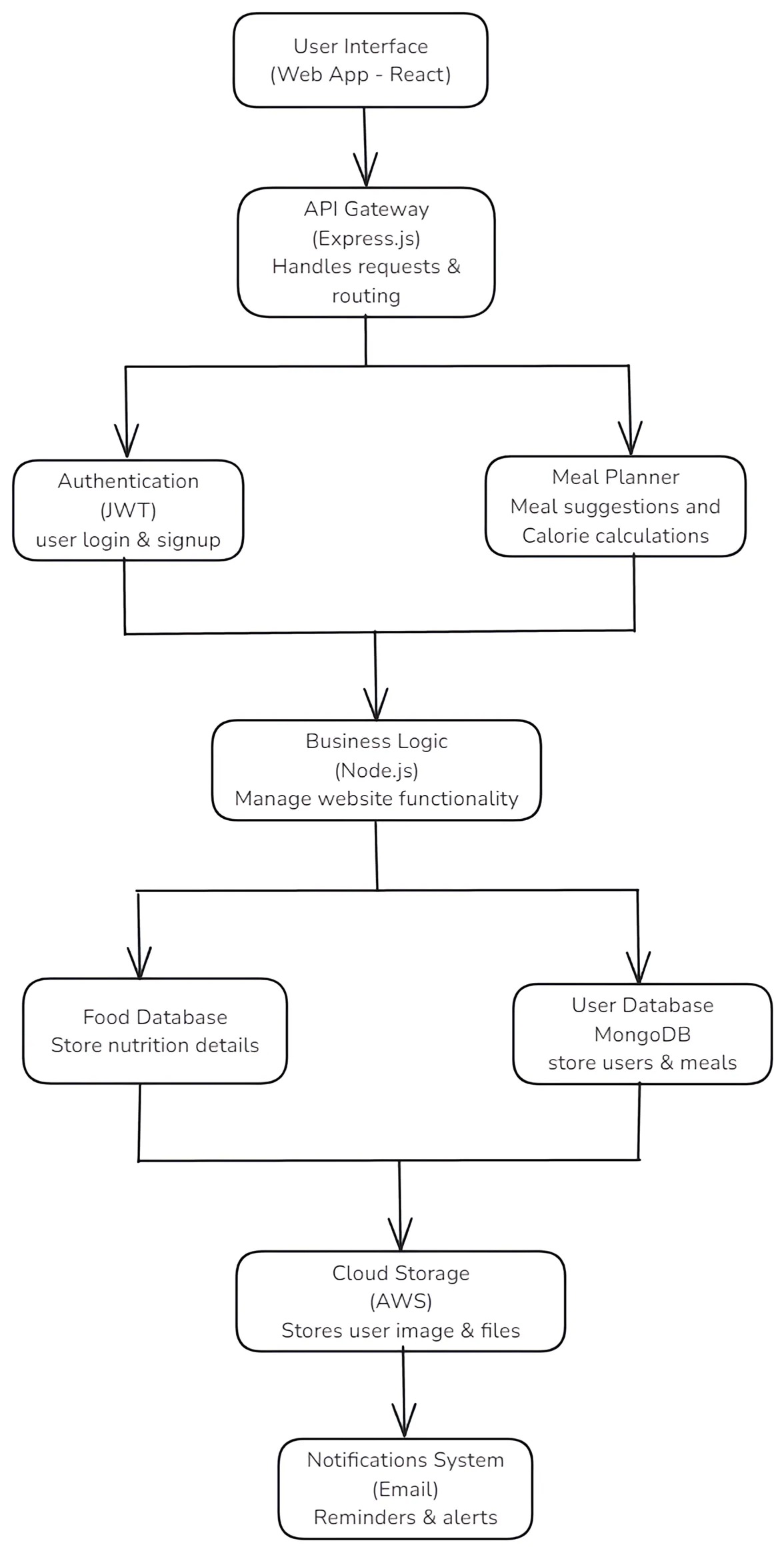
* **Use Case Scenarios:**
* Weight Loss: A user receives low-calorie meal suggestions based on their calorie target.
* Muscle Gain: A fitness enthusiast gets high-protein meals to match their calorie intake.

Busy Professionals: Quick, easy-to-make meals that fit within their calorie goal for the day.

* **Market Need:**
* Rising health and fitness awareness is driving demand for personalized meal planning tools.
* Users struggle with calorie counting and meal planning, creating a need for a simple, customized solution.

1. **System Design and Architecture:**

* **Tech Stack:**
* Frontend: HTML, CSS, JavaScript (React for dynamic interfaces).
* Backend: Node.js
* Database: MongoDB for storing user data, recipes, and meal plans.
* APIs: Integrate third-party APIs for nutrition data (e.g., USDA FoodData Central, Spoonacular , EDAMAN).
* Hosting: Use platforms like AWS, Google Cloud, or Heroku for hosting.
* **DESIGN PATTERNS:**
  + - **MVC (Model-View-Controller): Separates data, UI, and business logic.**
    - **RESTful API Design: For communication between frontend and backend.**
    - **Singleton Pattern (optional): Manages a single instance of the meal planner service.**
* **Functional Flow:**

****

1. **Coding Approach:**

* **Development Strategy:**
  + - **Frontend (React.js): Uses reusable components, API services, Redux/Context API for state management, and utility functions for calculations.**
    - **Backend (Node.js + Express.js): Implements controllers for handling HTTP requests, models for MongoDB schemas, API routes, business logic services, and utility functions.**
    - **Database (MongoDB): Stores user data, meal plans, and nutrition details with seed data for testing.**
    - **Testing: Includes unit and integration tests to ensure system reliability.**