**Nour Score Widget - Wireframe**

**Widget Layout**

**Overall Structure:**

* **Position:** Positioned below the header, occupying the top half of the screen.
* **Shape:** Circular graph or pie chart.
* **Dimensions:** Approximately 300-400px in diameter, depending on screen size.

**Sections:**

1. **Calories:**
   * **Representation:** A segment of the circle, color-coded in shades of gray.
   * **Progress Indication:** The segment's fill level indicates the user’s progress towards their daily caloric goal.
   * **Size:** Proportional to the total daily goal, with clear demarcation for different progress levels.
2. **Macros:**
   * **Segments:**
     + **Proteins:** One segment with a distinct shade of gray.
     + **Fats:** Another segment with a different shade of gray.
     + **Carbohydrates:** The third segment with yet another shade of gray.
   * **Progress Indication:** Each segment’s fill level shows the intake progress towards the daily goals for each macronutrient.
   * **Size:** Each segment is proportional to the respective macronutrient target within the total daily caloric intake.
3. **Hydration:**
   * **Representation:** A separate ring within or outside the main circle.
   * **Progress Indication:** The ring's fill level shows progress towards the daily hydration goal.
   * **Size:** Smaller than the main circle but visually distinct to differentiate from the caloric and macro segments.

**Interactive Elements:**

1. **Hover Effects:**
   * **Desktop:**
     + **Tooltips:** Appear on hover, showing detailed information such as percentages and numerical values for each segment.
     + **Design:** Simple boxes with a slight drop shadow, matching the monochrome color scheme.
     + **Content:**
       - **Calories:** “Calories: X of Y (Z%)”
       - **Proteins:** “Proteins: Xg of Yg (Z%)”
       - **Fats:** “Fats: Xg of Yg (Z%)”
       - **Carbohydrates:** “Carbs: Xg of Yg (Z%)”
       - **Hydration:** “Hydration: X of Y (Z%)”
2. **Tap/Press Interactions:**
   * **Mobile:**
     + **Detailed Stats Modal:** Tapping a segment brings up detailed stats in a modal or overlay.
     + **Design:** Full-screen or partial overlay, consistent with the app’s design.
     + **Content:** Expanded information including daily goals, current intake, and recommendations for improvement.
     + **Interaction:** Swipe to close or a close button in the corner.

**Design:**

* **Style:** Minimalist with clean lines, maintaining the monochrome color scheme.
* **Color Scheme:** Different shades of gray for various segments to maintain clarity and differentiation.
* **Font:** Sans-serif, consistent with the rest of the app.
* **Visual Indicators:** Clear borders between segments, subtle animations for progress updates to provide a dynamic feel.

**Accessibility:**

* **ARIA Labels:**
  + Each segment of the circle has ARIA labels to describe the content for screen readers.
  + Examples:
    - aria-label="Calories: X of Y (Z%)"
    - aria-label="Proteins: Xg of Yg (Z%)"
    - aria-label="Fats: Xg of Yg (Z%)"
    - aria-label="Carbohydrates: Xg of Yg (Z%)"
    - aria-label="Hydration: X of Y (Z%)"
* **Keyboard Navigation:**
  + Each segment is keyboard-navigable, with focus states that highlight the selected segment.
  + Pressing Enter or Space on a focused segment will display detailed information similar to the tap/press interaction on mobile.

**Visual Summary**

* **Widget (Circular graph or pie chart, 300-400px diameter)**
  + **Top Half of Screen:** Below the header
  + **Sections:**
    - **Calories:** Shade of gray segment, proportional fill
    - **Macros:**
      * Proteins: Distinct gray segment
      * Fats: Another distinct gray segment
      * Carbohydrates: Another distinct gray segment
    - **Hydration:** Separate ring, proportional fill
* **Interactive Elements:**
  + **Hover Effects (Desktop):** Tooltips with detailed info
  + **Tap/Press (Mobile):** Detailed stats modal
* **Design:**
  + Minimalist, monochrome, clean lines
  + Different shades of gray for segments
  + Subtle progress animations
* **Accessibility:**
  + ARIA labels for screen readers
  + Keyboard navigable with focus states

**Backend Components for Nour Score Widget**

To support the functionality and features of the Nour Score Widget, several backend components are necessary. These components handle data storage, processing, and retrieval, ensuring that the frontend can display accurate and up-to-date information.

**1. Data Storage**

**Database Design:**

* **User Profiles:** Stores user information, including nutritional goals, preferences, and history.
  + **Schema:**
    - user\_id: Unique identifier for the user
    - name: User's name
    - age, gender, weight, height: Basic demographic information
    - daily\_caloric\_goal: User's daily caloric target
    - macro\_goals: Object containing protein, fat, and carbohydrate targets
    - hydration\_goal: User's daily water intake goal
* **Nutritional Data:** Stores daily intake data for calories, macros, and hydration.
  + **Schema:**
    - user\_id: Reference to the user
    - date: Date of the recorded data
    - calories: Total calories consumed
    - proteins: Total grams of protein consumed
    - fats: Total grams of fat consumed
    - carbohydrates: Total grams of carbohydrates consumed
    - hydration: Total water intake

**2. API Endpoints**

**Endpoints:**

* **Get User Nutritional Goals:**
  + **Endpoint:** GET /api/nutrition/goals
  + **Description:** Fetches the user's nutritional goals.
  + **Response:** JSON object with caloric and macronutrient goals.
* **Get Daily Intake:**
  + **Endpoint:** GET /api/nutrition/daily-intake
  + **Description:** Fetches the user's daily intake for the specified date.
  + **Parameters:**
    - user\_id: ID of the user
    - date: Date for which data is requested (optional, defaults to current date)
  + **Response:** JSON object with daily intake data for calories, proteins, fats, carbohydrates, and hydration.
* **Update Daily Intake:**
  + **Endpoint:** POST /api/nutrition/daily-intake
  + **Description:** Updates the user's daily intake data.
  + **Request Body:**
    - user\_id: ID of the user
    - date: Date of the intake data
    - calories: Calories consumed
    - proteins: Grams of protein consumed
    - fats: Grams of fat consumed
    - carbohydrates: Grams of carbohydrates consumed
    - hydration: Water intake
  + **Response:** Status message indicating success or failure.

**3. Real-time Data Fetching**

**WebSockets:**

* Implement WebSockets for real-time updates of the Nour Score Widget.
  + **Connection:** Establish a WebSocket connection for each user session.
  + **Events:**
    - nutrition-update: Triggered when new nutritional data is available. Sends updated data to the client.
  + **Backend Handling:** On receiving new intake data, the server broadcasts an update to all connected clients for the user.

**4. Data Processing and Analysis**

**Nutritional Analysis Service:**

* **Function:** Analyzes user intake data and compares it against goals.
  + **Components:**
    - **Daily Summary Calculation:**
      * Calculates the total intake for calories, proteins, fats, carbohydrates, and hydration for the day.
    - **Goal Comparison:**
      * Compares the user's intake against their goals and determines the percentage of goals met.
    - **Recommendations:**
      * Generates recommendations or alerts if intake is below or above the target ranges.

**5. Security and Authentication**

**JWT Authentication:**

* Ensure all API endpoints are protected using JWT (JSON Web Tokens) for secure access.
  + **Token Generation:** Generate tokens on user login or registration.
  + **Token Verification:** Middleware to verify tokens on each API request.

**HTTPS:**

* Use HTTPS for all data transmission to ensure data security and privacy.

**6. Logging and Monitoring**

**Logging:**

* Implement logging for all API requests and WebSocket connections.
  + **Components:**
    - **Request Logs:** Log details of each API request, including user ID, endpoint accessed, and timestamp.
    - **Error Logs:** Log any errors or exceptions that occur during API processing.

**Monitoring:**

* Use monitoring tools to track the performance of API endpoints and WebSocket connections.
  + **Metrics:**
    - Response times
    - Error rates
    - Active WebSocket connections

**Summary of Backend Components**

1. **Data Storage:**
   * User Profiles schema
   * Nutritional Data schema
2. **API Endpoints:**
   * GET /api/nutrition/goals
   * GET /api/nutrition/daily-intake
   * POST /api/nutrition/daily-intake
3. **Real-time Data Fetching:**
   * WebSocket implementation for real-time updates
4. **Data Processing and Analysis:**
   * Daily Summary Calculation
   * Goal Comparison
   * Recommendations generation
5. **Security and Authentication:**
   * JWT Authentication
   * HTTPS for data transmission
6. **Logging and Monitoring:**
   * Request and error logging
   * Performance monitoring