A MULTIFACETED SWIM TRAINING APP ON ENHANCING SKILLS AND PERFORMANCE

Xiuhan (Daniel) Fu

Santa Margarita Catholic High School, 22062 Antonio Pkwy, Rancho Santa Margarita, CA

ABSTRACT

This research develops a swimmer's fitness journal along with a nutritional guidance application. The method employs journaling, expert suggestions, and detailed nutrition tips aimed at tackling precision issues using strictly-controlled research trials. The application exhibits remarkable accuracy in suggesting proteins and carbohydrates. It promises of more refinement according to personal goals and level of activity. The paper begins by presenting a brief overview of the issue in the swimming community. This highlights the significance of individualized dietary guidelines and coaching services tailored for swimmers.

Integration of BMI and BMR calculations for accurate nutritional recommendations is explained in this section of the methodology [13]. The author emphasizes on the need for accuracy in this work. It is also validated to previous standards and methodologies. The section of experiments and results reveals that several experiments were carried out in different situations, which prove the accuracy of the application's recommendations on nutrition. More nutrients can be incorporated as well. Another part of this article explores possible scenarios where this application applies, including swimmer focused design, more generalized fitness advice and nutrition recommendations targeting a more extensive audience.

Existing solutions are compared within a framework of a methodology comparison where one looks at features, effectiveness, and drawbacks. Limitations and avenue of future improvements are outlined in the summary section in addition to extending nutrient recommendations. This has bearing on application's importance in health of the population. Finally, this detailed fitness diary helps improve swimmer's training and performance. It might even benefit others who are looking for customized fitness and dietary guidelines.

KEYWORDS

Swimming, Fitness, Journaling, Advice

1. INTRODUCTION

There are a range of challenges that swimmers encounter in their training and competition journey. One significant issue is often the overlooked aspect of nutrition. Many swimmers struggle to grasp and implement effective nutritional strategies that can optimize their performance, increase recovery, and mitigate risks of injury. According to the National Library of Medicine, neglecting proper nutrition poses significant risks to athletes, potentially impairing their performance, delaying recovery, and increasing susceptibility to injuries and illnesses. Inadequate intake of essential nutrients can compromise energy levels, muscle function, and overall health, undermining an athlete's ability to excel in their sport [1]. Additionally, countless people have trouble with planning, and swimmers encounter the difficulty of keeping track of training progress, techniques, and goals. Manual tracking on paper can be prone to oversight and forgotten, potentially hindering improvement. For decades, the sport of swimming has grappled with these issues. While professional athletes have access to specialized coaching and nutritionists, half of the total number of swimmers rely on schools to host swimming lessons, not to mention private coaching [2].

The importance of proper nutrition, consistent tracking of performance, expert advice, and knowledge of safe training environments has long been recognized in the swim community. In the long run, lack of proper nutrition may result in impaired performance, reduced muscle mass and strength, delayed recovery, and long-term growth impairments, especially in young athletes. Furthermore, with many affected by the cost of lessons, a lot of swimmers are swimming with poor techniques without a coach to correct them. Common causes of injuries are wrong technique and poorly planned training. The most prevalent overuse injury is the shoulder, which could result in a long-term permanent injury. Additionally, knowing safe training locations around you can reduce the risk of accidents or encounters with hazardous conditions. A survey conducted by USA Swimming found that 75% of swimmers who consistently tracked their progress in a log journal and constantly improved their techniques reported seeing significant improvements in their performance over time [3].

1.1. Method Proposal

The proposed solution is a comprehensive swimming app that integrates a nutrition calculator, a training journal, and a map feature to locate nearby pools, addressing the challenges swimmers face in nutrition management, progress tracking, and safe training environments. The app will have three main features:

Nutrition calculator: It will provide personalized nutrition recommendations based on the user’s gender, weight, BMI, activity level, and more. Users can input their statistics and receive tailored advice on nutrient intake for carbohydrates, protein, and calories to ensure they meet their energy and nutrient needs for optimal performance.

Log Journal: Swimmers can digitally record their training progress, techniques, and goals. The app will offer reminders for consistent tracking after every practice and/or meet, based on the user’s preference. Users only need to input a date and location to fully organize each journal entry they create, which will eliminate the potential for oversight and allow for more effective and consistent training planning.

AI Advice: This section revolutionizes the way users access expert guidance. By integrating cutting-edge artificial intelligence, swimmers can now receive personalized and instant responses to any questions they have about training or technique. With 24/7 availability and the ability to understand natural languages, the AI Advice Section provides a level of support and empowerment that sets our app apart from others. It’s like having a team of experts on call, always ready to offer expert insights whenever swimmers need them, boosting their motivation, confidence, and ultimately their performance in the pool.

By combining nutrition guidance, progress tracking, and pool location services in a single app, swimmers have a centralized tool that addresses multiple facets of their training journey. The nutrition calculator provides tailored recommendations, accounting for individual preferences and training objectives. The app also offers a user-friendly platform that can be accessed anytime, anywhere, eliminating the need for manual tracking methods and providing easy access to important information.

Compared to traditional paper tracking, the app provides a digital platform that is less prone to oversight and offers reminders for consistent recording. Additionally, it integrates nutrition guidance and pretty much every type of advice one can think of, creating a one-stop solution for swimmers' needs. This app addresses the challenges comprehensively, making it a more efficient and effective solution for swimmers looking to optimize their training and performance

2. CHALLENGES

2.1. Challenge A

There were several problems with the nutrition page, which is supposed to have the user input their "weight," "height," "gender," and other information, and return their body mass index as well as suggested intake of various nutrients. Firstly, it lacks comprehensive error handling, particularly in the calculateNutrition function, where it doesn't handle null or zero values for carbohydrates and protein. Input validation is also missing, making it vulnerable to invalid user inputs. Furthermore, hardcoded string literals should be replaced with constants or localized strings for maintainability. Unused code blocks like _showNutrientAdviceDialog should be removed for clarity. Most of those problems were rooted in the long and complex code, so it is planned to include more comments to clarify complex logic and enhance accessibility through labels, which would also improve the code's quality.

2.2. Challenge B

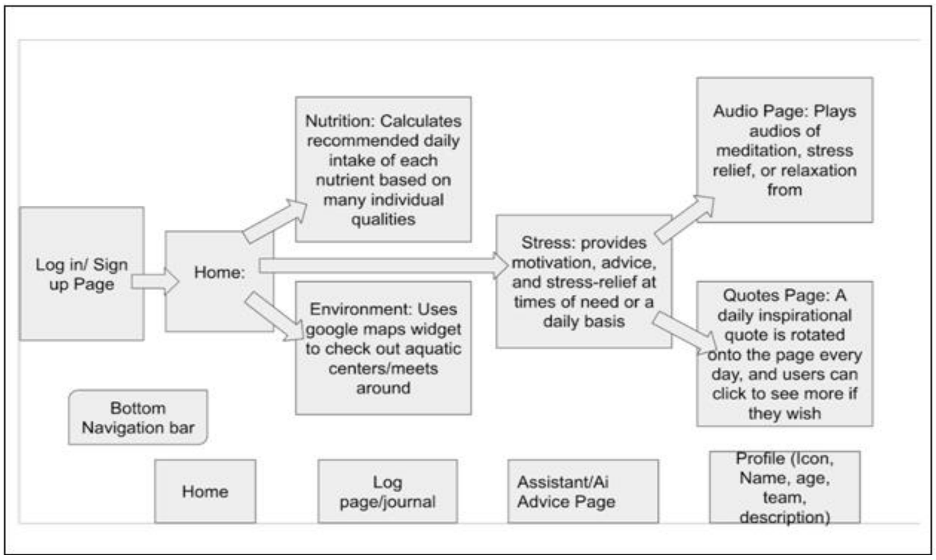
One key problem in designing the AI Advice Page for the swimmer's app is to provide accurate and dependable solutions to user queries. To begin, the model's training data must be large and relevant to swimming to minimize misrepresentation. Second, potential biases or skewed information generated by AI must be addressed. It would be necessary to use a strong pre-processing method to screen out any unsuitable or untrustworthy content. Furthermore, regular monitoring and updating of the training data, as well as user feedback loops, could assist refine the AI's responses over time. Finally, integrating a reporting system for consumers to indicate any wrong advice would add a high degree of safety and accuracy assurance.

2.3. Challenge C

A potential issue with the current implementation of the audio page could arise if new audio files are added to the Firebase Storage after the initial app launch. The audioFiles list is populated during the initState function and does not dynamically update if new files are added. This means that users might not see the latest audio files without restarting the app. To address this, the project could implement a mechanism to refresh the list of audio files at runtime. One approach could be to include a "Refresh" button on the UI that triggers a call to fetchAudioFiles when pressed. This way, users can manually update the list of available audio files. Furthermore, if immediate updates are critical, for this project, it is planned to explore Firebase Database or Firestore to track changes in the storage and automatically push updates to the app. This would ensure that the list of audio files is always up-to-date without requiring manual intervention from the user.

3. METHOD ANALYSIS

3.1. A – Diagram


Log in/ Sign up Page
Home:
Nutrition: Calculates recommended daily intake of each nutrient based on many individual qualities
Environment: Uses google maps widget to check out aquatic centers/meets around
Stress: provides motivation, advice, and stress-relief at times of need or a daily basis
Audio Page: Plays audios of meditation, stress relief, or relaxation from
Quotes Page: A daily inspirational quote is rotated onto the page every day, and users can click to see more if they wish
Bottom Navigation bar
Home
Log page/journal
Assistant/Ai Advice Page
Profile (Icon, Name, age, team, description)

3.1. System Overview

The Log Page serves as a platform for users to record their swimming activities. It encompasses various fields like journal title, date, location, strokes, distance, results, and goals. The entered data is then saved to a Firebase Firestore database, providing a structured and accessible record of the user's swim sessions. This component is crucial for users to keep track of their progress and performance over time.

The Nutrition Page is another central feature, where users input key details like weight, height, age, gender, carbohydrates, and protein intake. This information is managed through specific controllers for easy retrieval. The BMI is calculated using the calculateBMI() function, updating the display accordingly. When the user clicks "Calculate Nutrition," calculateNutrition() verifies that all necessary information is provided. If so, it computes BMI and offers tailored advice on carbohydrate and protein intake based on user specifics. The calculateBMR() function calculates Basal Metabolic Rate using weight, height, age, and gender. This is vital for generating precise nutrition advice.

Get Car bohydrates Advice() and get Protein Advice() then provide personalized recommendations based on the calculated metrics and user inputs. _show Validation Error Dialog() ensures users are prompted if all necessary information isn't provided. This page streamlines the user experience, making it user-friendly and efficient.

The assistance page comprises a Chat gpt model 3.5 widget in a Flutter application. The widget facilitates user interaction with the AI. Messages are managed through controllers, and chat history is stored. Communication with the Open AI API is moderated to generate responses. The user can ask any question or advice related to swimming or sports in general and they will receive a professional answer right away. The program's structure allows for seamless interaction between the user and AI.

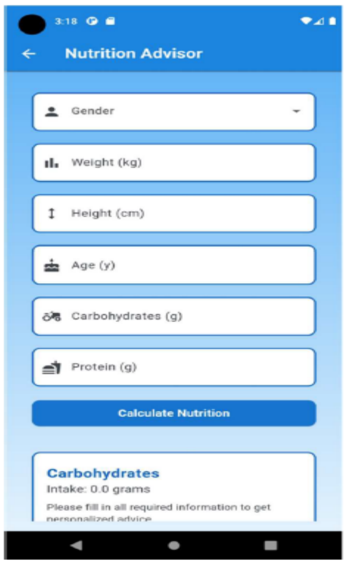
After logging in, the user is taken to the Nutrition Page, one of your application's three key components. They provide their personal information such as weight, height, age, gender, and nutrition intake here. The program validates this information and calculates the BMI, offering personalized advice on carbohydrates and protein consumption based on the user's particulars. In the home screen, the user can also navigate to the AI Advice Page, another major component, where they can seek advice on various health and fitness topics. This feature leverages AI-powered recommendations, providing users with valuable insights and tips. The Log Page, Nutrition Page, and AI Advice Page are interlinked through navigation buttons or tabs, allowing seamless movement between them. The program ensures data integrity by utilizing controllers and functions for calculations. The user can log out from any page, returning them to the Log Page, effectively concluding their session. Overall, this flow ensures a smooth user experience, guiding them through the application's features in a logical sequence. It balances user input, data processing, and output of personalized advice effectively, providing a comprehensive health and fitness tool.

Flutter was utilized to create this program.

3.2. Component Analysis A

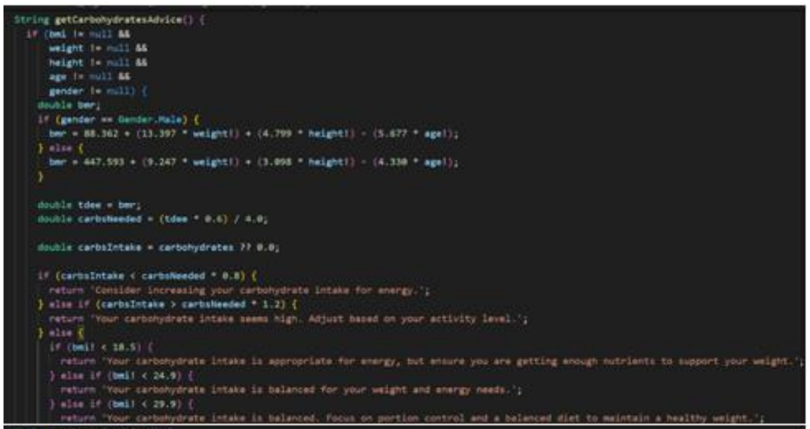
The "Nutrition Page" is a vital component of the application. It allows users to input personal details like weight, height, age, gender, and nutrition intake. Its primary purpose is to calculate the user's BMI and provide tailored advice on carbohydrate and protein consumption. This component relies on Flutter for the user interface and Dart for logic and calculations. It doesn't use specialized concepts like NLP or neural networks. Instead, it employs basic nutrition principles. In summary, it processes user input to offer personalized nutrition advice, enhancing the user's health journey.

3.3. Ui Screenshot

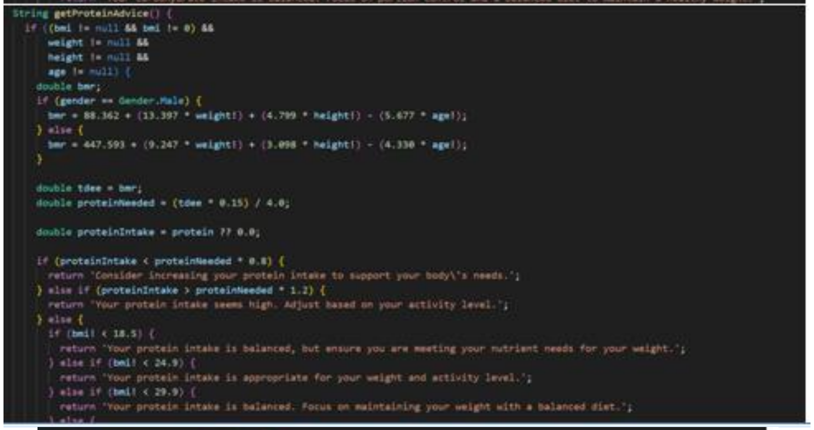


Carbohydrates
Intake: 0.0 grams
Please fill in all required information to get personalized advice

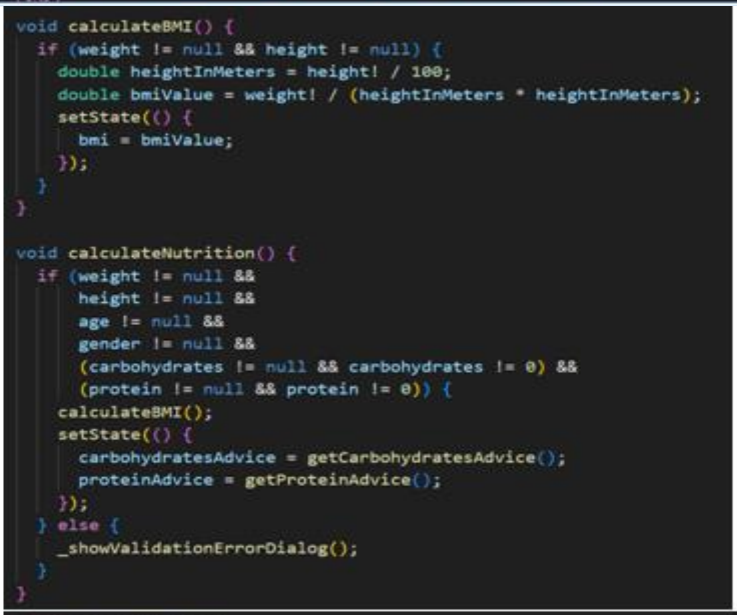
3.4. Code Sample



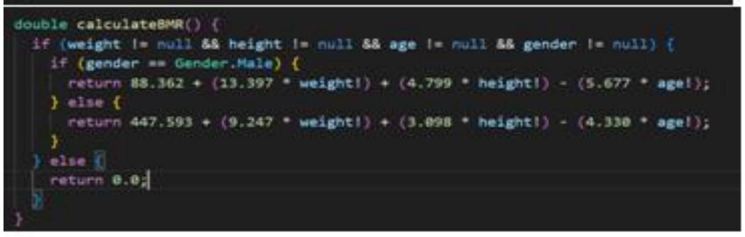
```
String getCarbohydratesAdvice() {
  if (bmi != null &&
      weight != null &&
      height != null &&
      age != null &&
      gender != null) {
    double bmr;
    if (gender == Gender.Male) {
      bmr = 88.362 + (13.397 * weight!) + (4.799 * height!) - (5.677 * age!);
    } else {
      bmr = 447.593 + (9.247 * weight!) + (3.098 * height!) - (4.330 * age!);
    }

    double tdee = bmr;
    double carbsNeeded = (tdee * 0.6) / 4.0;

    double carbsIntake = carbohydrates ?? 0.0;

    if (carbsIntake < carbsNeeded * 0.8) {
      return 'Consider increasing your carbohydrate intake for energy.';
    } else if (carbsIntake > carbsNeeded * 1.2) {
      return 'Your carbohydrate intake seems high. Adjust based on your activity level.';
    } else {
      if (bmi! < 18.5) {
        return 'Your carbohydrate intake is appropriate for energy, but ensure you are getting enough nutrients to support your weight.';
      } else if (bmi! < 24.9) {
        return 'Your carbohydrate intake is balanced for your weight and energy needs.';
      } else if (bmi! < 29.9) {
        return 'Your carbohydrate intake is balanced. Focus on portion control and a balanced diet to maintain a healthy weight.';
```



```
String getProteinAdvice() {
  if ((bmi != null && bmi != 0) &&
      weight != null &&
      height != null &&
      age != null) {
    double bmr;
    if (gender == Gender.Male) {
      bmr = 88.362 + (13.397 * weight!) + (4.799 * height!) - (5.677 * age!);
    } else {
      bmr = 447.593 + (9.247 * weight!) + (3.098 * height!) - (4.330 * age!);
    }

    double tdee = bmr;
    double proteinNeeded = (tdee * 0.15) / 4.0;

    double proteinIntake = protein ?? 0.0;

    if (proteinIntake < proteinNeeded * 0.8) {
      return 'Consider increasing your protein intake to support your body\'s needs.';
    } else if (proteinIntake > proteinNeeded * 1.2) {
      return 'Your protein intake seems high. Adjust based on your activity level.';
    } else {
      if (bmi! < 18.5) {
        return 'Your protein intake is balanced, but ensure you are meeting your nutrient needs for your weight.';
      } else if (bmi! < 24.9) {
        return 'Your protein intake is appropriate for your weight and activity level.';
      } else if (bmi! < 29.9) {
        return 'Your protein intake is balanced. Focus on maintaining your weight with a balanced diet.';
      } else {
```



```
void calculateBMI() {
  if (weight != null && height != null) {
    double heightInMeters = height! / 100;
    double bmiValue = weight! / (heightInMeters * heightInMeters);
    setState(() {
      bmi = bmiValue;
    });
  }
}

void calculateNutrition() {
  if (weight != null &&
      height != null &&
      age != null &&
      gender != null &&
      (carbohydrates != null && carbohydrates != 0) &&
      (protein != null && protein != 0)) {
    calculateBMI();
    setState(() {
      carbohydratesAdvice = getCarbohydratesAdvice();
      proteinAdvice = getProteinAdvice();
    });
  } else {
    _showValidationErrorDialog();
  }
}
```



```
double calculateBMR() {
  if (weight != null && height != null && age != null && gender != null) {
    if (gender == Gender.Male) {
      return 88.362 + (13.397 * weight!) + (4.799 * height!) - (5.677 * age!);
    } else {
      return 447.593 + (9.247 * weight!) + (3.098 * height!) - (4.330 * age!);
    }
  } else {
    return 0.0;
  }
}
```

3.5. Code Explanation

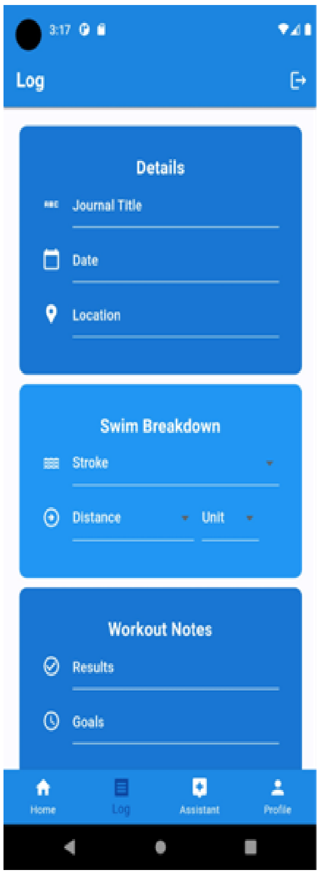
The code includes a number of essential procedures for computing and giving the user-specific nutritional advice. First, the calculate BMI() function uses the user's provided height and weight to calculate the Body Mass Index (BMI). The state of the program is then updated with this value. The complete process of providing nutritional advice is coordinated by the calculate Nutrition() method. It confirms the accessibility of crucial user data, such as body mass index, height, age, gender, and protein and carbohydrate intake. If all necessary information is provided, calculate BMI() starts the BMI computation, ensuring current BMI values. It then uses the functions get Carbohydrates Advice() and get Protein Advice() to update the advice for both car bs and protein. It prompts the user with an error message using _show Validation Error Dialog() if any important information is missing. The Basal Metabolic Rate (BMR), which is based on variables including weight, height, age, and gender, is calculated using the calculate BMR() function. Taking into account the user's BMI, weight, height, age, and gender, the functions get Carbohydrates Advice() and get Protein Advice() provide tailored advice on carbohydrate and protein intake, respectively.

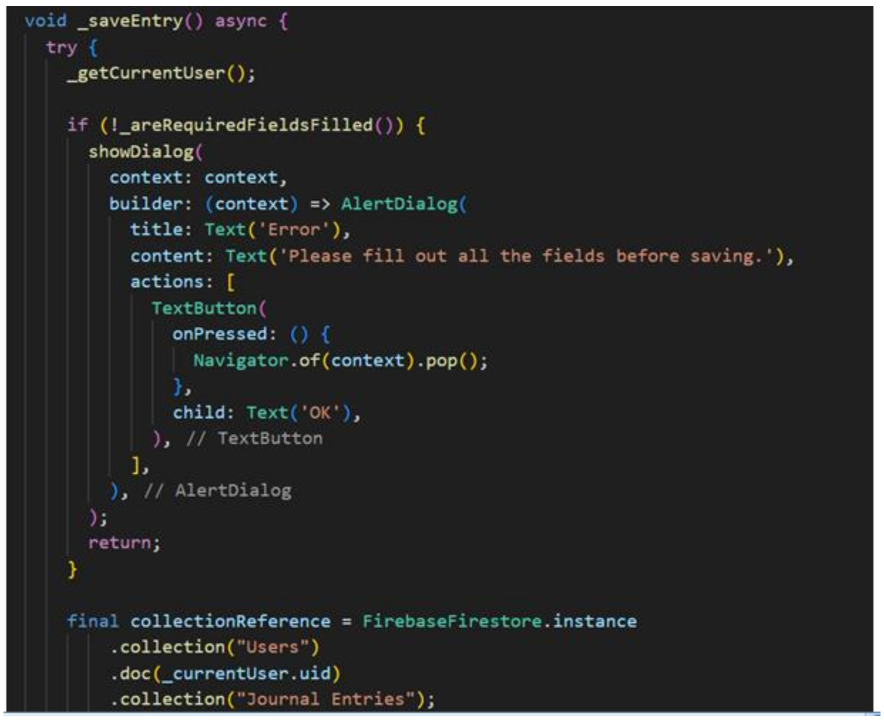
Based on determined variables, these approaches compare the user's actual intake to the suggested quantities. In the end, this collection of techniques successfully directs the user toward selecting a diet that is in line with their unique health objectives. It starts working when the user interacts with the nutrition page, entering their information and starting the computation. User privacy and data security are ensured because the program runs locally on the device and does not communicate with a backend server.

3.6. Component B

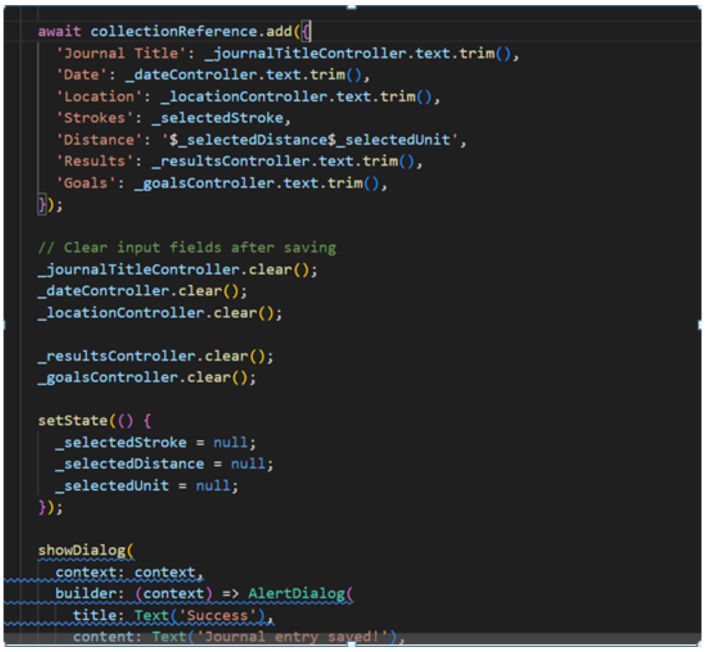
The Log Page component's goal is to provide a user interface for logging swim-related data. It allows users to enter information such as the journal title, date, location, stroke type, distance, unit, results, and swim goals. This data is then saved to a Firestore database linked to the current user.

The code relies on several packages to build this system, including cloud_firestore for communication with the Firestore database and firebase_auth for user authentication. Furthermore, it makes use of standard Flutter libraries such as material.dart for UI components and intl.dart for date formatting. In general, this component serves by providing a structured interface via which users can input and save swim-related data. Before attempting to save the data to the Firestore database, it ensures that all required fields are completed. Overall, this component is critical in supporting the logging and tracking of swim-related data throughout the program.

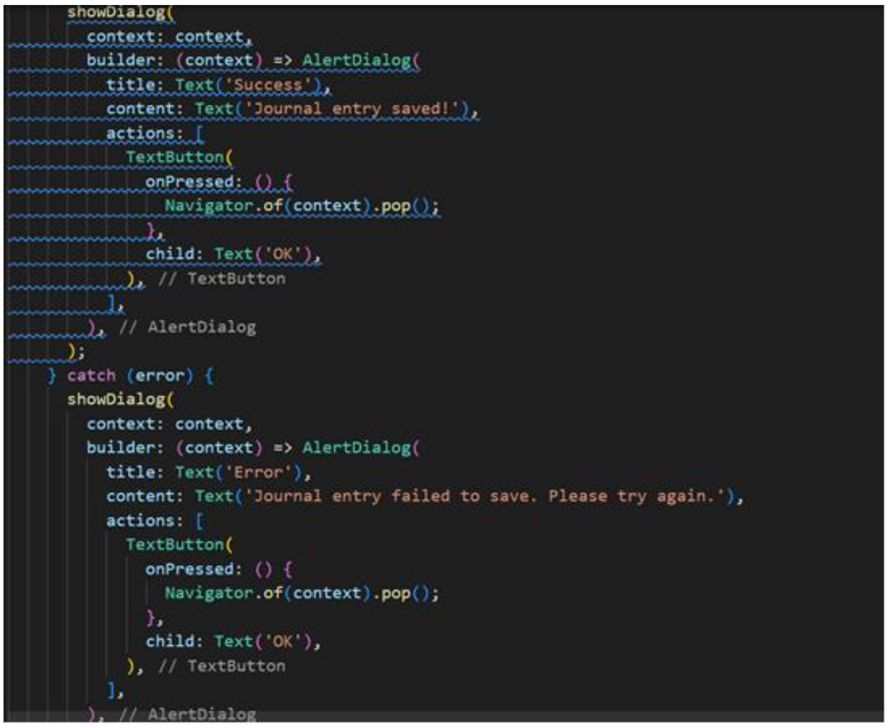




```
void _saveEntry() async {
  try {
    _getCurrentUser();

    if (!_areRequiredFieldsFilled()) {
      showDialog(
        context: context,
        builder: (context) => AlertDialog(
          title: Text('Error'),
          content: Text('Please fill out all the fields before saving.'),
          actions: [
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text('OK'),
            ), // TextButton
          ],
        ), // AlertDialog
      );
      return;
    }

    final collectionReference = FirebaseFirestore.instance
        .collection("Users")
        .doc(_currentUser.uid)
        .collection("Journal Entries");
```



```
await collectionReference.add({
  'Journal Title': _journalTitleController.text.trim(),
  'Date': _dateController.text.trim(),
  'Location': _locationController.text.trim(),
  'Strokes': _selectedStroke,
  'Distance': '$_selectedDistance$_selectedUnit',
  'Results': _resultsController.text.trim(),
  'Goals': _goalsController.text.trim(),
});

// Clear input fields after saving
_journalTitleController.clear();
_dateController.clear();
_locationController.clear();

_resultsController.clear();
_goalsController.clear();

setState(() {
  _selectedStroke = null;
  _selectedDistance = null;
  _selectedUnit = null;
});

showDialog(
  context: context,
  builder: (context) => AlertDialog(
    title: Text('Success'),
    content: Text('Journal entry saved!'),
```



```
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: Text('Success'),
        content: Text('Journal entry saved!'),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: Text('OK'),
          ), // TextButton
        ],
      ), // AlertDialog
    );
  } catch (error) {
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: Text('Error'),
        content: Text('Journal entry failed to save. Please try again.'),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: Text('OK'),
          ), // TextButton
        ],
      ), // AlertDialog
```

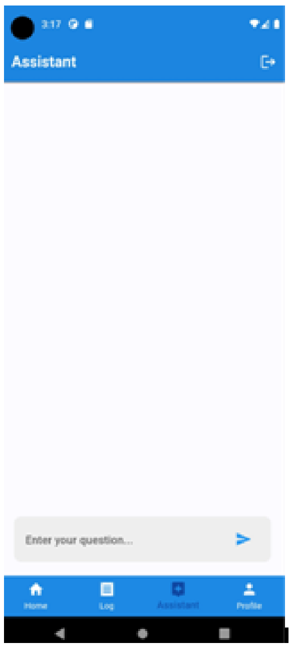
When a user interacts with the "Save" button in the user interface, the _saveEntry() method in the given code is called. This function is in charge of handling the logic when a user wants to save a swim log entry. To begin, it checks to see if all of the required fields, such as journal title, date, location, stroke, distance, unit, outcomes, and goals, are filled out. Following successful input validation, the procedure retrieves the currently authenticated user using Firebase Authentication. Following that, it connects to the Firestore database, creating a link to the user's journal entry collection. It conducts housekeeping duties after saving by resetting input fields, clearing controllers, and unselecting options.

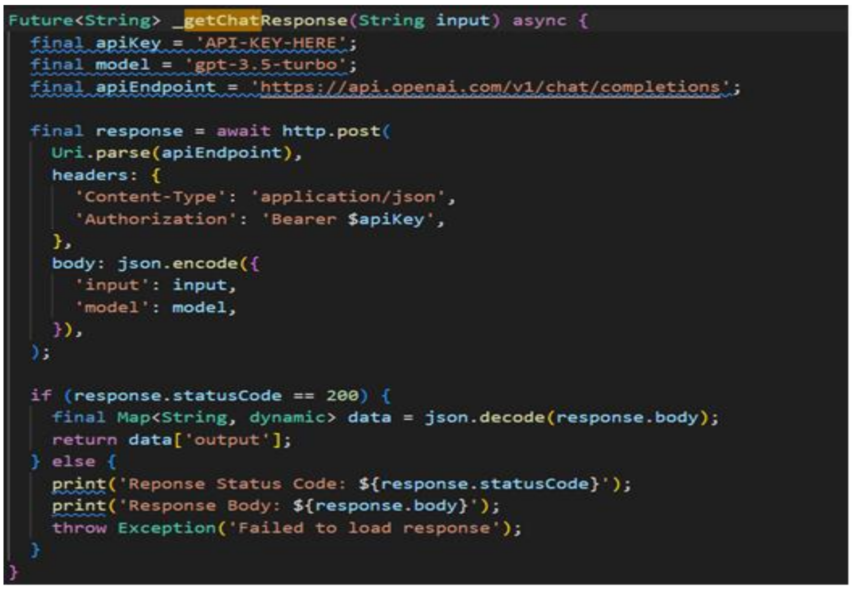
Several controllers for managing text input are initialized by the _Log Page State class, including _journal Title Controller, _date Controller, _location Controller, _results Controller, and _goals Controller. There is also a variable called _current User that contains information about the currently authorized user. The code also creates lists of options for stroke, distance, and unit selection, which are used in the user interface.

This code interfaces with a backend server, particularly Firestore, which is a Firebase NoSQL cloud database. When _save Entry() is called, the server receives a request to add a new document to the user's Firestore collection. The server responds to this request by producing and saving a new document with the user's swim log details, essentially persisting the user's data in the cloud database.

3.7. Component C

The Assistant Page component allows users to communicate with an Open AI-driven chat bot that is powered by the GPT-3.5 Turbo model. It uses the HTTP package for API communication and is written in Flutter. Users enter questions, which causes an OpenAI API request for a produced answer, imitating a discussion. For authentication and selection, the API key and model information are used. The _sendMessage() function handles updating the chat history by adding the user's message, formatted as "You: [message]", followed by processing the message through _getChatResponse(). The generated response from the chatbot is appended to the chat history as "ChatGPT: [response]". If any errors occur during the process, an error message is displayed in the chat history.





```
Future<String> _getChatResponse(String input) async {
  final apiKey = 'API-KEY-HERE';
  final model = 'gpt-3.5-turbo';
  final apiEndpoint = 'https://api.openai.com/v1/chat/completions';

  final response = await http.post(
    Uri.parse(apiEndpoint),
    headers: {
      'Content-Type': 'application/json',
      'Authorization': 'Bearer $apiKey',
    },
    body: json.encode({
      'input': input,
      'model': model,
    }),
  );

  if (response.statusCode == 200) {
    final Map<String, dynamic> data = json.decode(response.body);
    return data['output'];
  } else {
    print('Reponse Status Code: ${response.statusCode}');
    print('Response Body: ${response.body}');
    throw Exception('Failed to load response');
  }
}
```

The code segment is a Flutter application that works with OpenAI's GPT-3.5 Turbo model to build a conversation interface. The AssistantPage class is the primary component in charge of handling the chat UI. It includes methods such as _getChatResponse and _sendMessage.

The contact with OpenAI's server is handled by the _getChatResponse method. This function is called when a user submits a message. It creates an API request using the user's input, the model selected, and the API key. The request is forwarded to OpenAI's server, which extracts and returns the chat bot's output if it receives a successful response (status code 200). If an error occurs, it logs the status code and response body before throwing an exception.

When the user sends a message, the _send Message method is called. It adds the user's message to the conversation history before calling _get Chat Response to receive a response from the chat bot. When an answer is received, it is added to the chat history.

The variables _input Controller and _chat History govern the user's input in the chat, while _chat History records the conversation history. The user interface is created with Flutter widgets such as Scaffold, Column, Expanded, List View. builder, and Text Field.

The application sends an API request to Open AI's server when a user delivers a message. This request is processed by the server, and a response is generated. This response is then returned to the program and shown in the chat UI.

4. Experiments

4.1. Experiment A

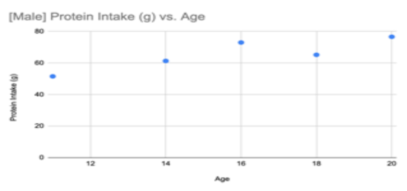
For the nutrition page, one blind spot is the user not providing valid input for weight, height, age, or nutrition intake values, which could lead to incorrect advice or errors in calculation.

4.2. Design

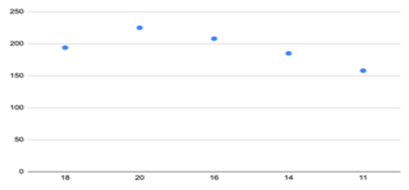
First, the program would simulate invalid input for one or more of the fields with blindspot. In the application, deliberately input incorrect or incomplete data. The control data are sourced from primarily two formulas used to calculate Body Mass Index and Basal Metabolic Rate (BMR). The formula for BMR is sourced by Garnet Health. It will also consist of valid inputs that are within the expected range [4]. It will serve as a baseline to compare against the results obtained from the experiment. We can record the behavior of the program when faced with invalid input. The program should detect and notify the user of the invalid input, providing clear error messages. Additionally, it should not crash or produce unexpected behavior. Finally, the app should guide the user to correct the input or prevent them from proceeding until valid data is provided.

4.3. Data and Visualization

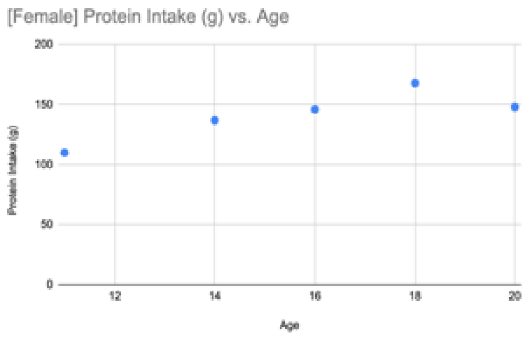
My nutrition calculator



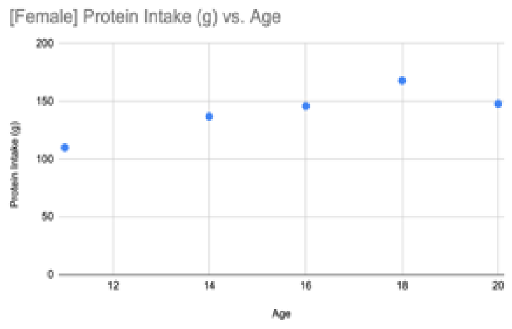
https://www.bodybuilding.com/fun/calpro.htm's calculator:



My calculator:



Theirs:



4.4. Analysis

This code calculates daily protein intake according to age, height, and body weight using BMR as a basis. Determining the correct BMR is crucial and entails attributes such as the age, sex, height and body weight that are drawn using established calculations. This gap between the estimated protein requirement and actual values suggests that there may be a mismatch of BMR and protein consumption. This precision is essential, usually expressed in terms of grams per kilogram of bodyweight (g/kg).

It is recommended to debug systematically, involving thorough examination of the conversion procedure, unit checking and controlled tests with sample data. Diagnostics, such as cross-referencing with authoritative sources and sensitivity analysis, are also useful. Specialized opinions could be sought from expert consultation if it is necessary for resolution. A critical assessment of the conversion process should be carried out together with specific tests towards locating where the discrepancy in protein calculation was generated.

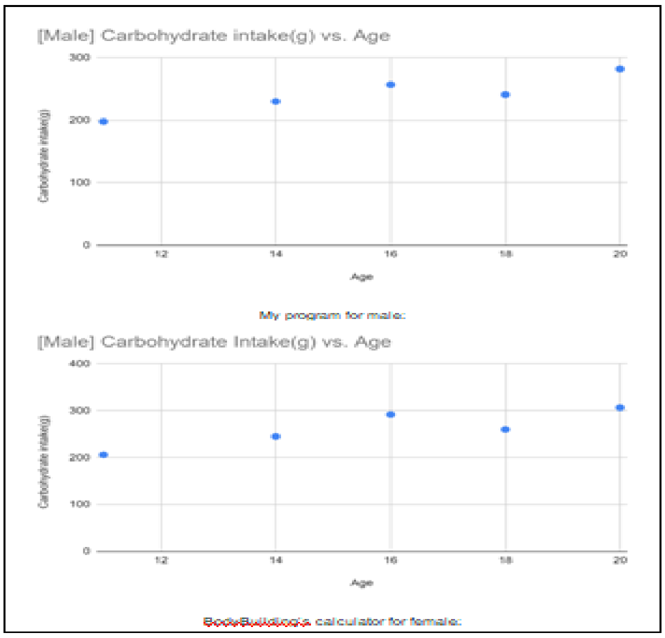
It is also important to consider sensitivity analysis to find out how much error in BMR calculation could cause significant differences in protein need. If these measures do not resolve the issue, consulting a nutritionist or any expert is the next step.

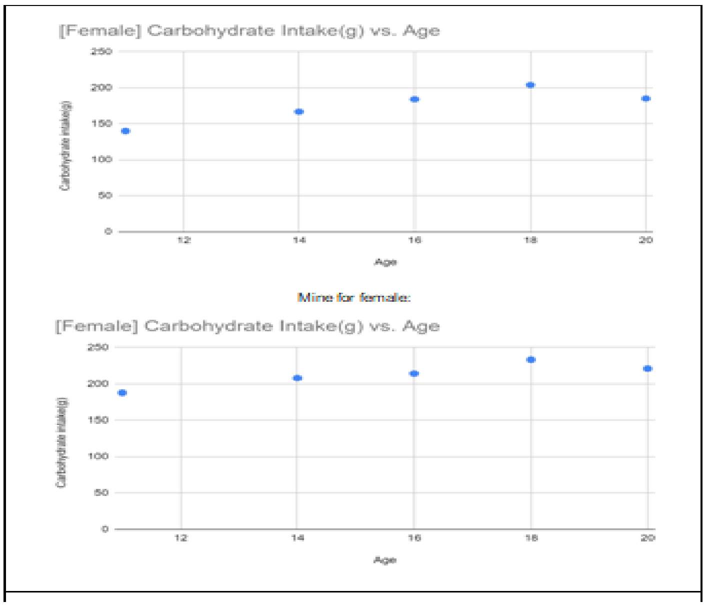
4.5. Experiment 2

Another possible blind spot is the calculation for carbohydrates. Even though the information collected had different sources for the formula of protein and carbohydrate, uncertainty remains on whether the calculations are correct for the carbohydrates data possessed.

In order to evaluate the application's performance, deliberately erroneous or incomplete data was introduced, creating controlled scenarios akin to blind spots. This involves intentionally inputting inaccurate or partial information. The foundational data is sourced from two critical formulas used for computing Body Mass Index (BMI) and Basal Metabolic Rate (BMR), with BMR sourced from Garnet Health. Alongside, valid inputs falling within the expected range will be included to serve as a reference for comparison. Following this, I will systematically document the program's response when confronted with instances of invalid input. The application's functionality must encompass the capacity to swiftly detect and communicate the presence of invalid data, providing lucid error notifications. Equally important is the program's stability, ensuring it refrains from crashing or displaying unforeseen behavior under these circumstances.

Body Building’s calculator for male





The code experiment was designed to establish daily amounts of carbohydrates based on submitted information such as age, height, and weight. Interesting insights in the comparison with a professional carbohydrate calculator. The predicted carbohydrate requirements for the five tests for men were off by less than 20 grams on average from those of professional calculator calculations. This shows an acceptable rate of precision, implying that the underlying programming is sound and fit for the male population. Nonetheless, the situation is dramatically different for female users. Women’s carbohydrate needs displayed much more variability than the pro calculator, with an average difference amounting to nearly 40 grams. Although this difference appears more noticeable in the case of females, it is still acceptable due to the natural variability of metabolism and biology in people.

However, several reasons might account for these variations. However, one should not forget that nutrition requirements may be affected by many individual-specific factors other than age, height, and bodyweight, such as changes in hormone level and physical activity. On top of that, minor differences in the base calculations or premises used in the specialized calculator and your code may explain some of these differences. The accuracy of the carbohydrates can be improved by checking the algorithms and formulas against current nutrition practices. Also, more variables or making adjustments on the existing ones would yield a higher level of accuracy, especially with the females.

5. METHODOLOGY COMPARISON

5.1. Methodology A

In an article regarding the nutritions for swimmers by Swimming World Magazine offers practical advice for swimmers on optimizing their nutrition. It highlights the significance of carbohydrates and proteins in their diet, providing specific intake recommendations based on body weight. The emphasis on hydration and electrolyte balance, particularly before competitions, is valuable. However, it could benefit from addressing individual dietary needs, micronutrient considerations, and the psychological impact of nutrition on an athlete's performance. Additionally, long-term nutritional planning and the potential use of supplements are aspects that could be explored further [5].

This project allows users to input their specific details, such as gender, weight, height, and age, enabling it to provide highly personalized nutrition advice. This level of customization is not possible with a generic article. Also, the app offers personalized advice on carbohydrate and protein intake, as well as BMI information. In contrast, the first article primarily focuses on general recommendations without considering individual factors.

5.2. Methodology B

An article published by Runners World offers a set of swimming techniques for triathletes, drawing from the expertise of Michael Phelps, Terry Laughlin, and David Marsh. It emphasizes streamlining the body, effective hand placement, body rotation, and head positioning to reduce resistance and optimize propulsion. This approach is effective in improving triathletes' swimming proficiency, particularly given the demands of their multi-discipline training. However, it's important to note that individual variability and the need for ongoing practice may influence progress. Additionally, the guide does not address open water specifics, race strategies, or equipment considerations, which are important for a well-rounded training regimen [6]. The project code creates a journal with search functionality, which is much more efficient and easier than journaling on paper. Helper methods such as _buildErrorMessage and _BuildLoadingIndicator for displaying different types of message and formatted text to ensure readability and have the entries easy to understand. Lastly, ‘StreamBuilder’ listens to changes in the user’s journal entries and updates the UI accordingly.

5.3. Methodology C

Finally, Your Swim Log talks about how crucial journaling daily entries and logging individual times are for athletics. The journaling solution works by having swimmers write down their ideal workouts and races, helping them visualize success and focus their mental approach. It's effective in reducing pre-race nerves, as studies show it improves performance under pressure. However, it's not a replacement for physical training and may not cover all aspects of performance improvement, such as external factors or specific technical skills [7].

This digital platform enhances the journaling process by providing a convenient and organized way for swimmers to document their practices, races, and reflections. Additionally, it integrates Firebase services for authentication and Firestore for easy data management. Compared to the earlier passage which discussed the benefits of journaling, this code provides a practical tool that enables swimmers to directly apply the described techniques in a digital format.

6. CONCLUSIONS

6.1. Limitations And Improvements

Under the “Nutrition Advisor” page, users are able to fill in the most important data concerning their condition (gender, weight, height and age) as well as diet analysis. Afterward, the app provides personalized advice on carbs, protein, and weight based on such information input. It provides the opportunity for addition of more specific useful products such as fats, vitamins, minerals, and fiber. It is expected that this growth will enable consumers to understand their own diet in detail. Additionally, for enhancing the reliability and relevance of recommended guidelines that are specific to particular nutrients needs should be embraced.

This can be done through quoting from respected sources such as the DRIs and RDAs recommended by recognized health bodies. Since our studies used a calculator that incorporates weight goals and activity level, we may discuss their possible inclusion into this process. This would increase the accuracy of the advice, for body weight goals and physical activity greatly influence nutrient requirements. With the inclusion of these details, the tool is set to provide more individualized and persuasive recommendations, thus raising the app’s relevance to people who wish to follow customized eating plans.

6.2. Concluding Remarks

This project is a key step toward developing a comprehensive fitness journal and nutritional guidance app. The Journal Page allows for easy access to entries, whereas the Nutrition Page gives personalized help via BMI calculation along with customized counsel. On the advice/assistant page, the ChatGPT widget improves user interaction, while the quotations, stress, and audio features promote well-being. In essence, the software not only meets but exceeds its purpose by providing a user-centric platform that tackles all aspects of health and wellness. It blends fitness tracking with mindful health habits to provide a well-rounded approach to fitness.

7. SUMMARIES

7.1. Experiment Recap

Experiment 1 aimed to identify potential blind spots related to invalid user input in the nutrition calculator. Invalid data for weight, height, age, or nutrition values could lead to errors. The experiment involved deliberately inputting incorrect and comparing it with valid inputs within expected ranges. Control data were sourced from established formulas for Body Mass Index (BMI) and Basal Metabolic Rate (BMR), with the BMR formula from Garnet Health. The program's behavior in handling invalid input was observed, emphasizing the need for clear error messages and program stability.

Experiment 2 focused on assessing the accuracy of carbohydrate calculations. Data was sourced from critical BMI and BMR formulas, with BMR from Garnet Health. The results revealed acceptable precision for men, with minor discrepancies. However, variability was more pronounced for female users, indicating the influence of individual-specific factors beyond basic demographics. Suggestions included refining algorithms, incorporating additional variables, and aligning calculations with current nutrition practices to enhance accuracy, particularly for females.

7.2. Methodology Comparison

Methodology A focuses on optimizing nutrition for swimmers, as outlined in an article from Swimming World Magazines. It places a strong emphasis on the significance of carbohydrates and proteins in a swimmer's diet, providing specific intake recommendations based on body weight. It also highlights the importance of maintaining proper hydration and electrolyte balance, particularly before competitions. While this approach offers valuable foundational advice for swimmers, it doesn’t really address individual dietary needs and micronutrient considerations [8]. Methodology B, outlined in an article from Runners World, talks about swimming techniques tailored for triathletes. The approach draws advice from accomplished swimmers like Michael Phelps, Terry Laughlin, and David Marsh. It places a strong emphasis on refining techniques such as streamlining the body, ensuring effective hand placement, mastering body rotation, and optimizing head positioning. While this is highly effective in enhancing triathletes' swimming proficiency, it lacks coverage on crucial aspects like open water specifics, race strategies, and equipment considerations.

Methodology C, advocated by Your Swim Log, explains the significance of journaling daily entries and logging individual times for athletes, particularly swimmers. The approach encourages swimmers to write down their ideal workouts and races, which aids in visualizing success and improving their mental approach. This practice proves highly effective in reducing pre-race nerves and enhancing performance under pressure, according to studies. However, it's important to note that while journaling is a valuable tool, it cannot serve as a complete replacement for physical training.

REFERENCES

[1] Beck, K. L., Thomson, J. S., Swift, R. J., & von Hurst, P. R. (2015, August 11). Role of nutrition in performance enhancement and postexercise recovery. U.S. National Library of Medicine. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4540168/

[2] Moore, C. (2021, August 31). Millions of children can’t swim because lessons are costly, survey suggests. Fox Business. https://www.foxbusiness.com/lifestyle/millions-children-cant-swim-lessons-costly-survey-says

[3] Department of Health & Human Services. (2007, August 7). Swimming - preventing injury. Better Health Channel. https://www.betterhealth.vic.gov.au/health/healthyliving/swimming-preventing-injury

[4] Garnet Health. (2016, July 1). Basal Metabolic Rate Calculator. Garnet Health. https://www.garnethealth.org/news/basal-metabolic-rate-calculator#:~:text=Calculate%20Basal%20Metabolic%20Rate&text=Your%20basal%20metabolism%20rate%20is

[5] Duran, B. (2018, July 10). The Big Deal About a Swimmer’s Nutrition. Swimming World News. https://www.swimmingworldmagazine.com/news/the-big-deal-about-a-swimmers-nutrition/

[6] Bean, M. (2009, November 23). Seven Steps To Better Swimming Technique. Runner’s World. https://www.runnersworld.com/uk/training/triathlon/a764520/seven-steps-to-better-swimming-technique/

[7] The Power of Journaling for Swimmers. (2017, July 7). YourSwimLog.com. https://www.yourswimlog.com/journaling-for-swimmers/

[8] Duran, B., & Duran, B. (2023, October 18). Wellness Wednesday: The Big Deal About a Swimmer's Nutrition: What to Know. Swimming World News. https://www.swimmingworldmagazine.com/news/the-big-deal-about-a-swimmers-nutrition/

[9] Sabounchi, N. S., Rahmandad, H., & Ammerman, A. S. (2013). Best-fitting prediction equations for basal metabolic rate: informing obesity interventions in diverse populations. International Journal of Obesity, 37(10), 1364–1370. https://doi.org/10.1038/ijo.2012.218

[10] Poirier-Leroy, O. (2020, June 4). The power of journaling for swimmers. SwimSwam. https://swimswam.com/power-journaling-swimmers/