Calories365: An Innovative Approach to Calorie Tracking with Voice Input

Larysa Katerynych[1,†], Maksym Kubichka[1,*,†], Kostiantyn Zhereb[1]

[1] *Taras Shevchenko National University of Kyiv, 60 Volodymyrska St., Kyiv, 01033, Ukraine*

Abstract

In this article, an innovative system called Calories365 is presented – a web application for automated calorie tracking using voice input in multiple languages. The developed system significantly reduces the time required for registering meals compared to traditional methods (manual entry and photo analysis) and ensures high data input accuracy. A comparative analysis with popular services such as MyFitnessPal, FoodDiary, and Lifesum is conducted, highlighting the competitive advantages of the proposed solution. The paper also provides a detailed description of the application's architecture based on modern technologies (Laravel, Vue.js, Docker, Redis, Meilisearch) and its integration with the OpenAI API for automating voice command recognition and analysis. Prospects for further system development are considered, including the possibility of implementing custom machine learning models and personalized recommendations for optimizing dietary intake.

Keywords

Calories365, voice input, calorie tracking, food automation, OpenAI API, web application

1. Introduction

In today's world, the automation of everyday tasks is becoming increasingly important, and the development of intelligent information systems holds a priority in software engineering.

One such task is dietary control [1], specifically the automation of calorie counting, which requires high accuracy and user convenience. Traditional calorie counting services mostly rely on manual data input, which is labor-intensive and time-consuming. Some modern applications attempt to solve this issue by analyzing photos of food; however, this method often lacks sufficient accuracy.

This article proposes an innovative solution—the Calories365 project, which combines a web application with voice input functionality [2] for maintaining a food diary. The task is addressed using voice input, which significantly simplifies and accelerates the registration of food intake compared to manual input, and provides better accuracy compared to photo analysis. A distinctive feature of Calories365 is its implementation of voice input in Ukrainian —a functionality that is currently rarely encountered among similar solutions [3, 4]. Thanks to the efficient organization of language files in Vue and Laravel, and the use of the OpenAI API with extended language support, the system can be easily scaled to other languages.

At the outset of the project's development, the possibility of voice input was absent in other calorie counting services; however, similar solutions have recently started appearing among competitors, confirming the relevance of this approach. While existing market alternatives, such as MyFitnessPal, predominantly support voice input in English, the novelty of Calories365 lies in its implementation of voice input in Ukrainian—a feature that is practically non-existent in similar solutions.

Voice input has proven its high efficiency compared to traditional methods of recording food consumption by significantly reducing data entry time.

Workshop "Software engineering and semantic technologies" SEST, co-located with 15th International Scientific and Practical Programming Conference UkrPROG'2025, May 13-14, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

katerynych@gmail.com (L. Katerynych); maxim.kubichka@gmail.com (M. Kubichka); zhereb@gmail.com (K. Zhereb)

0000-0001-7837-764X (L. Katerynych); 0009-0003-6372-5869 (M. Kubichka); 0000-0003-0881-2284 (K. Zhereb)

The Calories365 system is built using modern technologies such as Laravel, Vue.js, Docker, and the OpenAI API, employing a flexible architecture. The integrated mechanism—"voice → artificial intelligence → structured data → storage in a database"—is universal and can be applied in other scenarios where quick and convenient information entry is required. This allows the system to rapidly adapt to new requirements and demonstrates the successful application of modern methods in creating intelligent information systems, expert systems, and decision support systems. Such applications combining web technologies and voice input are currently becoming popular [5].

2. Description of the Features of the Calories365 Application

Calories365 offers users a comprehensive set of features designed for the most convenient and effective food diary management. The application consists of several main sections, each developed for simplicity and ease of use.

2.1. Voice Input of Products.

The primary innovation of Calories365 is the voice input page, where users can quickly and conveniently add information about consumed products using their voice. The system automatically recognizes spoken language, analyzes what is said, and converts the information into structured data about products and calories. Users can manually edit the obtained data or use the "Generate" button to regenerate the information via the OpenAI API for maximum accuracy. Finally, users can save the products in their diary. The interface of the voice input page is demonstrated in Figure 1.

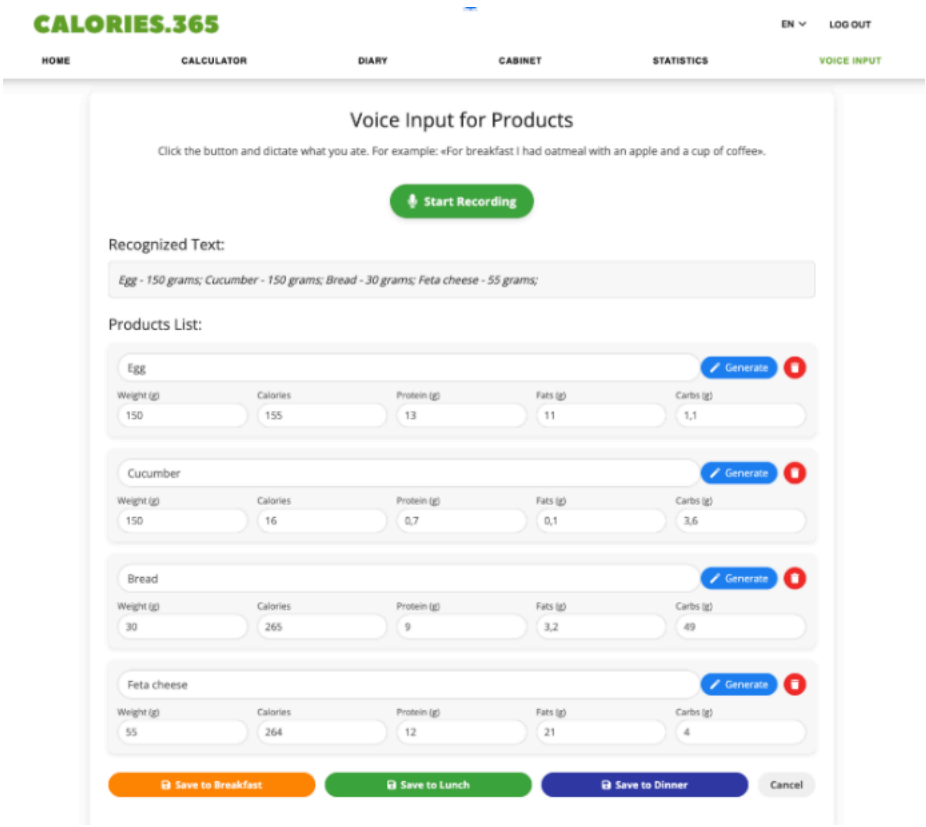

Figure 1: The interface of the voice input page

2.2. Voice Input of Products.

The "Diary" section allows users to manually add products, view all entries made throughout the day, and edit or delete information. The diary is organized by meals (breakfast, lunch, dinner) and automatically calculates the total number of calories, thereby simplifying dietary control. Figure 2 shows the diary page.

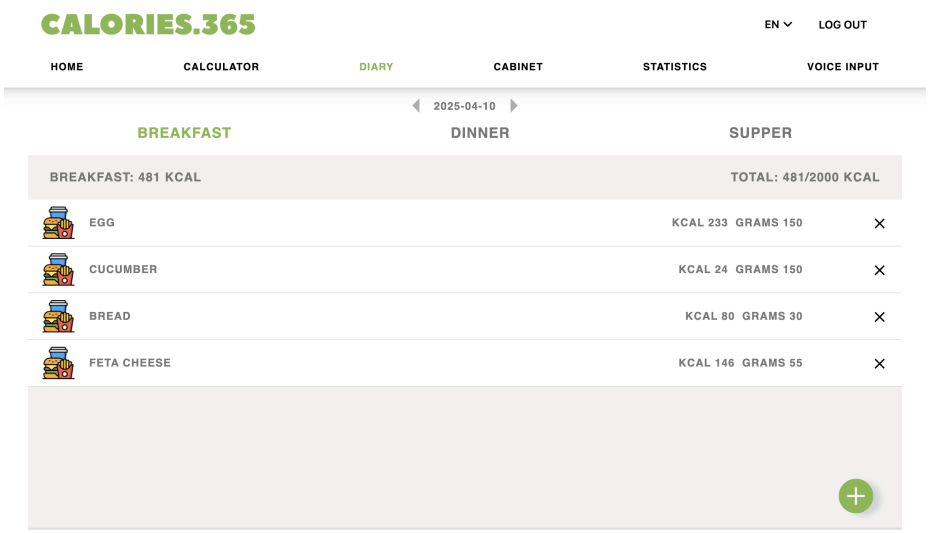

Figure 2: The interface of the diary page

2.3. Nutrition Statistics.

The "Statistics" section offers the user a convenient calendar that displays the number of calories consumed each day compared to a set norm. Days are automatically highlighted in different colors depending on the degree of conformity to the norm, which allows for a quick evaluation of nutritional efficiency over the month. The statistics component is shown in Figure 3.

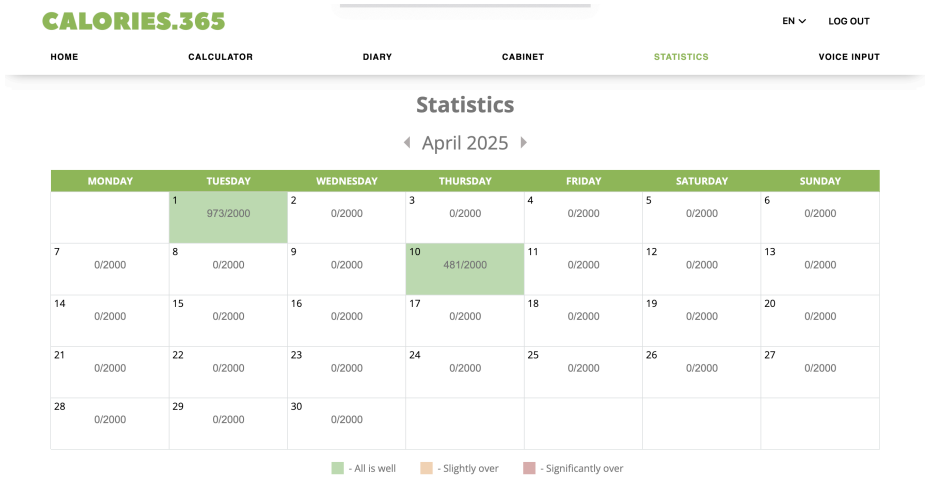

Figure 3: The interface of the statistics page

2.4. Calorie Calculator.

The Calories365 Calorie Calculator uses well-known nutritional models [1, 6]. It computes the basal metabolic rate (BMR) using the Harris-Benedict formula (BMR = 88.362 + 13.397 × weight + 4.799 × height − 5.677 × age for men, and BMR = 447.593 + 9.247 × weight + 3.098 × height − 4.330 × age for women) or, if the user has provided a body fat percentage, using the Katch-McArdle formula (BMR = 370 + 21.6 × LBM, where LBM = weight × (100 − %fat)/100). The body mass index (BMI) is calculated as BMI = weight / (height/100)^2, after which the system classifies the user's condition according to the standard scale. Maintenance calories are determined by the equation Maintenance = BMR × activity coefficient, and the daily target is calculated as Daily = Maintenance × 0.8 (for weight loss) | × 1 (for weight maintenance) | × 1.2 (for weight gain). The optimal macronutrient ratio is computed as: carbohydrates = 100 − proteins − fats, with proteins and fats adjusted according to the goal and activity level. The estimated time to reach the target weight is derived from the equation Days = |current weight − target weight| × 7700 / (maintenance − daily), which gives the user a clear idea of the time required to achieve the result.

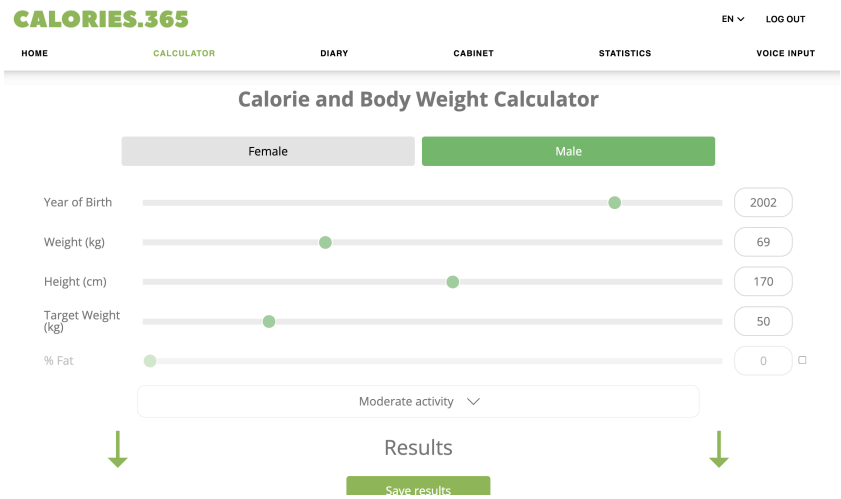

Figure 4: The interface of the calculation page

The user interface of Calories365 shown in Figure 4 allows users to manage their nutrition as effectively and conveniently as possible while achieving their set goals.

3. Comparative Analysis

In today's market, various calorie-counting services offer a range of solutions, each with its own advantages and drawbacks. To objectively evaluate the effectiveness of Calories365, a comparative study was conducted with the most popular applications, including MyFitnessPal, which is one of the market leaders and recipient of the "Editor's Choice" award, as well as FoodDiary and Lifesum. During the experiment, manual, voice, and photo-based data entry methods for recording consumed food were compared. The study was conducted by recording the screen while entering the same meal and subsequently analyzing the timing. The meal included the following: 30 grams of bread, 150 grams of cucumber, 3 eggs, 55 grams of feta cheese, half of a medium-sized onion, 125 grams of tomato, 15 ml of olive oil, 4 slices of chorizo, 30 grams of hard cheese, 10 grams of mustard, and 7 grams of butter. The complete set of experimental data is available in the "Appendices" section.

The comparative results of the data entry timing are presented in Table 1.

Table 1: Comparative Results of Data Entry Timing

Data Entry Method	Application	Data Entry Time
Manual Entry	MyFitnessPal	2 min 54 sec
Voice Entry	MyFitnessPal	53 sec
Manual Entry	Calories365	2 min 05 sec
Voice Entry	Calories365	40 sec

As seen from the table, voice input significantly reduces the time compared to manual entry. A key feature of Calories365 is its support for multiple languages and the ability to edit the obtained product data and regenerate information about a specific product using the "Generate" button after voice input. This function is not available in MyFitnessPal, making Calories365 more flexible and user-friendly. It should also be noted that during voice input, MyFitnessPal demonstrated inaccuracies—for example, a cucumber was recognized as 75 grams instead of the actual 150 grams, and a tomato as 65 grams instead of 125 grams. Our results show improvements in input speed that are similar to those reported in [7, 8].

An analysis of the accuracy of data entry using photographs was also conducted. MyFitnessPal recognized only a portion of the ingredients, making significant errors in both quantity and the presence of products. For instance, a red pepper was mistakenly identified instead of a cucumber, and the number of eggs and feta cheese was inaccurately determined. FoodDiary also inaccurately identified the meal, committing significant mistakes and generalizations.

The overall comparative analysis is presented in Table 2.

Table 2: Comparative Analysis of Calorie Tracking Applications

MyFitnessPal	English only	Absent	Yes, low accuracy
Lifesum	Absent	Absent	Absent
Food Diary	Absent	Partially	Yes, low accuracy
Calories365	Multilingual	Full	Absent

The main competitive advantages of Calories365 are its convenient voice input with multilingual support and the subsequent refinement of data through integration with the OpenAI API, as well as its flexibility in use. Additionally, Calories365 provides comprehensive functionality (calorie norm calculation, user-friendly statistics). Compared to solution reported in [9], the main benefits of our solution are voice input and web interface, improving user experience and convenience.

Thus, the conducted studies confirm that the voice method of data entry, as implemented in Calories365, improves both efficiency (faster input) and effectiveness (more accurate input) [9]. The ability to easily edit and regenerate information via the OpenAI API integration significantly improves the user experience. In contrast, manual data entry is more labor-intensive and inconvenient, while photo-based input demonstrates low accuracy and significant errors compared to other methods. Consequently, Calories365 effectively addresses the problem of convenient and accurate calorie tracking.

4. Architecture and Technologies Used

When developing Calories365, modern technologies were used to ensure high performance, scalability, and system security. The server side is implemented using Laravel—a popular open-source PHP framework built on the MVC pattern. The backend utilizes the Sanctum and Fortify modules for reliable user authentication and data protection. The core application data is stored in MySQL, while Redis is employed for effective caching and session storage. Fast product search is achieved through Meilisearch [10], which allows for instantaneous retrieval of required products from a large dataset.

The frontend part of Calories365 is built on the Vue.js framework using a Single Page Application (SPA) approach, ensuring high interactivity and a fast interface response. To isolate and simplify the deployment process, Docker and docker-compose are used, allowing the application to run in a unified containerized environment. For secure traffic and application protection, Cloudflare is applied with a configured SSL certificate, which guarantees the confidentiality of user data.

4.1. Frontend Architecture for Dynamic Components in Calories365.

Calories365 employs a unified architecture to create flexible and scalable form and table components. At the core of this system are configuration objects—declarative structures that fully define the interface and behavior of the components.

Example of a table configuration used for the administrative panel of the application:

```
export const config_table = [
  { label: 'ID', key: 'id', type: 'default', action: null, limit: 40 },
  { label: 'Name', key: 'name', type: 'link', action: 'show', limit: 40 },
  { label: 'Calories', key: 'calories', type: 'default', action: null,
limit: 40 },
  { label: 'Delete', key: 'delete', type: 'button', action: 'delete',
limit: 40 }
];
```

The configuration data is passed to a container component responsible for generating the interface.

Below is an example of using the container for the table:

```
<div class="row">
    <table-main-part
        :columns="config_table"
```

```
        :data="products"
        @handle="handleEvent"/>
</div>
```

The container component dynamically selects the type of components to render based on the configuration. This approach allows for quick and convenient addition of new fields and interaction types without needing to change the main component code. In addition, internal component events are processed by adapters that pass them to the higher level where the business logic is implemented. This separation of concerns helps to maintain code clarity and simplicity, significantly easing the process of scaling the interface for new tasks.

The dynamic components architecture is actively used for product editing forms, food intake history tables, user settings, and administrative panels. Thanks to this architecture, Calories365 can easily adapt to new requirements without major changes to the codebase.

4.2. Deployment Infrastructure (CI/CD)

Calories365 utilizes modern Continuous Integration and Continuous Delivery (CI/CD) practices, implemented through Docker and GitHub Actions. The Docker configuration creates a clear and isolated environment for both development and production:

- PHP-FPM (container: calories365_php);
- Nginx (container: calories365_nginx);
- MySQL (container: calories365_mysql);
- Redis (container: calories365_redis);
- Meilisearch (container: calories365_meili);
- A separate container for background Laravel tasks (queue worker).

For production deployment, a multi-stage Docker build is used to optimize the size of the final images and enhance performance. The containers are configured via docker-compose, which allows for precise management of dependencies between the services. GitHub Actions automatically triggers the update process on the server whenever a push is made to the main branch. This automation process enables rapid delivery of updates to the production server, minimizes the risk of human error, and ensures the stability of the application.

Demonstration of docker-compose.yml code for a container:

```
services:
  calculator_php:
    build:
      context: .
      dockerfile: Dockerfile
      target: php-final
    container_name: calories365_php
    working_dir: /var/www
    env_file:
      - ./.env
    depends_on:
      - meilisearch
      - calories_mysql
      - calories_redis
    networks:
      - internal_net
```

In summary, the technologies and architectural solutions used in Calories365 ensure high performance, security, and ease of maintenance—key characteristics of modern intelligent information systems.

5. Integration of the Frontend with the OpenAI API

The innovative aspect of Calories365 is largely ensured by the integration of the web application with the OpenAI API, which enables the automation of voice message recognition and the structural analysis of food intake directly within the user interface.

5.1. Technological Architecture for Integration with OpenAI

Calories365 employs two key OpenAI models:

- Whisper API (model whisper-1) — for highly accurate conversion of voice recordings into text.
- GPT-4o — for analyzing the user's text, extracting information about products, and automatically generating data regarding the nutritional composition (calories, proteins, fats, carbohydrates)

Voice recording on the frontend is implemented using modern Web Audio API technologies: access to the user's microphone is obtained via MediaDevices.getUserMedia(), the audio data stream is recorded using the MediaRecorder API, after which a Blob object in webm format is created and sent to the server.

5.2. Complete Cycle of Voice Message Processing in Calories365

- Voice Recording and Sending (Frontend):
  The user clicks the record button in the interface, after which audio recording begins via MediaRecorder. When the recording is finished, the Blob object containing the audio file is automatically sent to the server for further processing.
- Audio Processing on the Server (Backend):
  On the server side, the VoiceController receives the audio file and passes it to the SpeechToTextService. The audio file is then processed by the Whisper API, which returns a text transcription.
- Analysis of the Food Intake Text:
  The obtained transcription is sent to the method analyzeFoodIntake(), which uses GPT-4o. This method applies a specially designed prompt that allows it to extract from the text a list of products, their quantities, and approximate caloric values in a structured format.
- Product Search in the Database Using Meilisearch:
  The resulting data is compared with the existing product database. If the relevance of the found product (rankingScore) is equal to or greater than 0.9, the information is taken directly from the database. In cases of low relevance or if the product is absent in the database, an additional information generation step is triggered.
- Generation of New Data via GPT-4o:
  When the product is not found in the database or is found insufficiently accurately, the system automatically requests GPT-4o to generate detailed information about the product's nutritional composition. Consequently, the user receives structured data such as calories, proteins, fats, and carbohydrates.
- Database Update:
  After user confirmation, the newly generated data is automatically added to the database, allowing Calories365 to gradually expand and refine its own product database.
- Error Handling and Fallback Strategies:
  In case of errors, such as the unavailability of the OpenAI API or issues with the received data, the system displays clear messages to the user and offers alternatives: retrying the generation, manual editing, or manual data entry.

5.3. Working with Results on the Frontend

After the server processes the data, the Calories365 interface displays to the user the transcription text, the list of recognized products, and their nutritional composition. The interface allows users to easily change product names (with the possibility of re-searching in the database), edit the number of calories and nutrients manually, generate additional information via OpenAI, and delete any redundant items from the list.

Thanks to such deep integration of the Calories365 frontend with the OpenAI API, the application ensures the convenience and speed of managing a food diary, allowing users to literally "talk" their diet while the system automatically processes and structures the obtained data. This significantly reduces the user's time investment, increases the accuracy of the records, and creates a unique, positive user experience.

Schematic Process of Integrating Calories365 with the OpenAI API:

- The user records a voice message using the Web Audio API.
- The audio file is transmitted to the server.
- The Whisper API transcribes the audio into text.
- GPT-4o analyzes the text, identifies the products, their weight, and caloric content.
- The obtained data is matched against the product database via Meilisearch.
- If the product is not found or the relevance is insufficient, GPT-4o generates new data.
- The processed results are sent to the Calories365 frontend.
- The user reviews, edits, or confirms the data.
- After confirmation, the new or updated data is stored in the database.

6. Prospects and Directions for Further Development

The Calories365 project has significant potential for both technical and business development. Below are the key prospective directions that will help the project remain innovative and competitive in the market of intelligent information systems.

6.1. Developing an In-House Recognition Model or Fine-Tuning Existing Models.

An important step is to create a customized self-hosted machine learning model for voice recognition or fine-tune existing models such as Whisper and GPT-4o. This will allow for:

- A significant reduction in the costs associated with using external APIs.
- Much faster processing of voice messages due to reduced latency in an in-house infrastructure.
- Improved recognition accuracy for Ukrainian and other languages by optimizing the models for the specific use cases of Calories365.

6.2. In-Depth Diet Analysis and Personalized Recommendations

Integrating machine learning algorithms will enable a deeper analysis of users' eating habits and behavior. This includes:

- Automatic analysis of a user's diet to identify patterns of consumption, including both healthy and unhealthy habits.
- Generating personalized recommendations based on individual parameters (weight, height, age, physical activity), as well as allergies, preferences, and medical indications.
- Developing adaptive meal plans with dynamic adjustments based on the user's behavior and changes in their physical condition.

6.3. Using a Vector Database for Efficient Search and Recommendations.

Integrating a vector database (e.g., Pinecone, Qdrant, or pgvector) will significantly improve the process of product search and recommendation generation for Calories365:

- Vector search will quickly identify the most relevant products even when queries are imprecise or incomplete.
- The use of semantic search will greatly enhance the accuracy of search queries and the personalization of recommendations for individual users.
- Automation of nutrient data generation for new products with higher accuracy through vector representations.

6.4. Implementing Machine Learning for Increased Accuracy and Quality.

Deploying machine learning models that learn from historical user data will significantly improve the accuracy of predictions and the overall quality of Calories365:

- Automatically improving the accuracy of voice query recognition through regular retraining on collected user data.
- Using predictive models to forecast users' eating habits and needs, which will allow for more effective personalized meal plans and recommendations.
- Optimizing the performance of neural network models (such as GPT-4o) for the specifics of the Ukrainian language and typical dietary habits of Ukrainian users.

6.5. Improving Interfaces and Expanding Functionality.

Continued work on the Calories365 interface and chatbot will make them even more intuitive and user-friendly:

- Enhancing the UX/UI design by updating interfaces with an emphasis on convenience, accessibility, and ease of use.
- Expanding analytical features by adding more detailed statistics and data visualization (for example, weight change graphs, nutrient balance, dietary trends).
- Integrating with external services, such as popular health platforms, fitness trackers, and medical services, to provide a comprehensive approach to a healthy lifestyle.

6.6. Introducing Social Interaction Functionality.

Adding social features to Calories365 could significantly enhance user engagement and retention:

- Enabling users to share their achievements and meal plans with friends or within communities.
- Organizing competitions, challenges, and rankings to further motivate users.
- Facilitating the exchange of recipes, dietary tips, and advice on healthy living among users.

In summary, the proposed future directions will allow Calories365 to continuously improve its technological foundation, ensure the highest quality user experience, and remain a relevant, modern, and competitive solution in the market of intelligent information systems.

7. Conclusions

In summary, it can be stated that the Calories365 project is an innovative solution in the field of dietary control, successfully integrating modern technologies to automate the calorie counting process. The use of voice input significantly accelerates food registration and provides higher accuracy compared to traditional methods (manual entry and photo analysis). The main

competitive advantage of Calories365 is its support for voice input in Ukrainian and its intuitive interface.

The project architecture, based on Laravel, Vue.js, Docker, Redis, Meilisearch, and other modern solutions, ensures flexibility, modularity, and high scalability of the system. Integration with the OpenAI API enables the automation of voice message recognition and analysis, which should result in continuous growth and improvement of the database.

Future prospects for Calories365 include developing customized self-hosted machine learning models, implementing personalized recommendations based on dietary analysis, integrating with popular messaging platforms, and enhancing the user interface.

Thus, Calories365 demonstrates the effective application of modern technologies to address real problems faced by today's users and has significant potential for further technical and business development in the field of intelligent information systems, while the versatility of the proposed approach allows it to be easily adapted to other application areas where there is a need for convenient and prompt data registration through voice messages.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI tools (ChatGPT, models o1 and o3-mini) in order to: Grammar and spelling check, Citation management. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S.S. Gropper, J.L. Smith, T.P. Carr, Advanced Nutrition and Human Metabolism, 8th ed. Cengage Learning, 2021.

[2] D.Yu, L. Deng, Automatic Speech Recognition: A Deep Learning Approach. Springer, London, 2015.

[3] L. Kobylyukh, Z. Rybchak, O. Basystiuk, Analyzing the Accuracy of Speech-to-Text APIs in Transcribing the Ukrainian Language, in: Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems, CEUR-WS, 2023, pp. 217-227.

[4] A. Dumyn, S. Fedushko, Y. Syerov, Review of Automatic Speech Recognition Systems for Ukrainian and English Language, in: P. Štarchoň, S. Fedushko, K. Gubíniová (Eds), Data-Centric Business and Applications, volume 212 of Lecture Notes on Data Engineering and Communications Technologies, Springer, Cham, 2024, pp. 319-334.

[5] A. Alateeq, M. Roantree, C. Gurrin, Voxento 2.0: a prototype voice-controlled interactive search engine for lifelogs, in: Proceedings of the 4th Annual on Lifelog Search Challenge, ACM, New York, NY, 2021, pp. 65-70.

[6] Bendavid, Itai, et al. "The centenary of the Harris–Benedict equations: How to assess energy requirements best? Recommendations from the ESPEN expert group." Clinical nutrition 40.3 (2021): 690-701.

[7] Norda, Marvin, et al. "Evaluating the efficiency of voice control as human machine interface in production." IEEE Transactions on Automation Science and Engineering 21.3 (2023): 4817-4828.

[8] Dutsinma, Faruk Lawal Ibrahim, et al. "A Systematic Review of Voice Assistant Usability: An ISO 9241–11 Approach." SN computer science 3:267 (2022): 1-23.

[9] V. Pleskach, Ye. Vakulenko, A. Serdyuk, Information System for Calorie Calculation, in: Proceedings of the 14th International Scientific and Practical Programming Conference (UkrPROG 2024), CEUR-WS, 2024, pp. 168-179.

[10] Meilisearch Documentation. (2025). Retrieved from https://www.meilisearch.com/docs .

A. Online Resources

The experimental data for this study can be accessed at:
https://drive.google.com/drive/folders/1wovLaCTOQaGBDN7K9w5vAzHA6Y1rXpXQ?usp=sharing
.