

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РЕАЛІЗАЦІЯ WEB-ДОДАТКУ ДЛЯ РЕКОМЕНДАЦІЙ ЗІ
ЗДОРОВОГО ХАРЧУВАННЯ»

Виконав: студент 4 курсу, групи 6.1220
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)
освітньої програми Комп'ютерні науки
(назва освітньої програми)

О.І. Пономаренко

(ініціали та прізвище)

Керівник доцент кафедри комп'ютерних наук, д.е.н., доцент
Полукетова Н.Р.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент д.т.н, професор, завідувач кафедри фундаментальної
та прикладної математики ЗНУ Гребенюк С. М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний
Кафедра комп'ютерних наук
Рівень вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(шифр і назва)
Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук,
д.т.н., доцент

Шило Г.М.

(підпис)

“ 25 ” грудня 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Пономаренку Олександрю Ігоровичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Реалізація web-додатку для рекомендацій зі здорового харчування
- керівник роботи Полуектова Н.Р., доцент кафедри комп'ютерних наук, д.е.н.
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)
- затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с
2. Строк подання студентом роботи 05.06.2024
3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Розробка телеграм-бота
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 25.12.2023

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	21.01.2024	
2.	Збір вихідних даних.	13.02.2024	
3.	Обробка методичних та теоретичних джерел.	17.03.2024	
4.	Розробка першого та другого розділу.	13.04.2024	
5.	Розробка третього розділу.	29.04.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	08.06.2024	
7.	Захист кваліфікаційної роботи.	23.06.2024	

Студент _____
(підпис)

О.І. Пономаренко
(ініціали та прізвище)

Керівник роботи _____
(підпис)

Н.Р. Полуектова
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.Г. Спиця
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Реалізація web-додатку для рекомендацій зі здорового харчування»: 48 с., 24 рис., 10 джерел, 1 додаток.

АЛГОРИТМИ РОЗРАХУНКУ, БАЗОВИЙ МЕТАБОЛІЗМ, ЗДОРОВИЙ СПОСІБ ЖИТТЯ, ІНТЕРАКТИВНІ РЕКОМЕНДАЦІЇ, КАЛОРІЙНІСТЬ, ОПТИМІЗАЦІЯ ХАРЧУВАННЯ, ПЕРСОНАЛІЗОВАНІ ПЛАНИ, ПІДТРИМКА КОРИСТУВАЧІВ, ТЕЛЕГРАМ БОТ, ФІЗИЧНА АКТИВНІСТЬ, PYTHON.

Об'єкт дослідження – телеграм бот для здоров'я та фітнесу, який надає персоналізовані рекомендації щодо харчування та фізичної активності, базуючись на індивідуальних параметрах користувачів.

Предмет дослідження – методи та алгоритми, що використовуються для розробки телеграм бота, який забезпечує персоналізовані рекомендації щодо харчування та фізичної активності.

Мета роботи: Розробити телеграм бота, який надає користувачам персоналізовані рекомендації щодо харчування та фізичної активності, базуючись на їх індивідуальних параметрах та цілях. Для досягнення цієї мети необхідно було проаналізувати існуючі підходи до створення подібних ботів, дослідити ефективні алгоритми і формули для розрахунку потреб користувачів, а також розробити і протестувати власний бот.

У роботі розроблено телеграм бот для здоров'я та фітнесу, який надає персоналізовані рекомендації щодо харчування та фізичної активності.

SUMMARY

Bachelor's Qualifying Theses «Implementation of a web application for healthy nutrition recommendations»: 48 pages, 24 figures, 10 references, 1 supplement.

CALCULATION ALGORITHMS, BASAL METABOLISM, HEALTHY LIFESTYLE, INTERACTIVE RECOMMENDATIONS, CALORIES, NUTRITION OPTIMIZATION, PERSONALIZED PLANS, USER SUPPORT, TELEGRAM BOT, PHYSICAL ACTIVITY, PYTHON.

Object of study – telegram bot for health and fitness, which provides personalized recommendations on nutrition and physical activity based on individual user parameters.

The subject of the study – methods and algorithms used to develop a telegram bot that provides personalized nutrition and physical activity recommendations.

Objective: to develop a Telegram bot that provides users with personalized nutrition and physical activity recommendations based on their individual parameters and goals. To achieve this goal, it was necessary to analyze existing approaches to creating such bots, research effective algorithms and formulas for calculating user needs, and develop and test our own bot.

This paper develops a telegram bot for health and fitness that provides personalized recommendations on nutrition and physical activity.

ЗМІСТ

Завдання на кваліфікаційну роботу	2
Реферат	4
Summary.....	5
Вступ.....	7
1 Вступ до фітнес-телеграм-ботів	9
1.1 Актуальність розробки фітнес-телеграм-ботів	9
1.2 Виявлення основних можливостей та обмежень існуючих оптимізаційних інструментів	10
1.3 Мета та завдання роботи.....	14
2 Теоретичні основи розрахунку енергетичних потреб.....	16
2.1 Формула харріса-бенедикта.....	16
2.2 Рівні фізичної активності.....	17
2.3 Персоналізація енергетичних потреб відповідно до цілей користувача	19
3 Реалізація фітнес-телеграм-бота	21
3.1 Вибір технологій та інструментів розробки	21
3.2 Вибір технологій та інструментів розробки	23
3.3 Опис функціональності бота	25
3.3.1 Обробка вхідних даних користувача	25
3.3.2 Розрахунок енергетичних потреб та надання рекомендацій.....	26
3.3.3 Інтерактивний інтерфейс та навігація	28
3.4 Перспективи розвитку фітнес-телеграм-бота	28
3.4.1 Інтеграція бази рецептів.....	29
3.4.2 Відстеження прогресу	29
3.4.3 Персональні нагадування.....	29
3.4.4 Тренувальні плани.....	30

3.4.5 Інформаційна база про спортивне харчування та рекомендації	30
Висновки.....	32
Перелік посилань	34
Додаток А Код телеграм бота.....	36

ВСТУП

У сучасному світі технології займають важливе місце у повсякденному житті, зокрема в сфері здоров'я та фітнесу. Одним з таких інноваційних рішень є використання телеграм ботів, які забезпечують персоналізований підхід до здоров'я і фізичної активності. Враховуючи зростаючу потребу в оптимізації здорового способу життя, мій проект "Телеграм Бот" є актуальним та своєчасним рішенням.

Мета роботи полягає у розробці телеграм бота, який надає користувачам персоналізовані рекомендації щодо харчування та фізичної активності, базуючись на їх індивідуальних параметрах та цілях. Для досягнення цієї мети необхідно було проаналізувати існуючі підходи до створення подібних ботів, дослідити ефективні алгоритми і формули для розрахунку потреб користувачів, а також розробити і протестувати власний бот.

Актуальність роботи полягає у зростаючій кількості людей, які прагнуть вести здоровий спосіб життя, але стикаються з браком знань та ресурсів для досягнення своїх цілей. Існуючі програми та сервіси для здоров'я та фітнесу мають свої обмеження та недоліки. Створення нового інструменту, який враховує індивідуальні потреби користувачів і використовує науково обґрунтовані алгоритми, є важливим кроком у вирішенні цієї проблеми.

Практична важливість роботи полягає в тому, що розроблений телеграм бот пропонує зручний і доступний спосіб отримання персоналізованих рекомендацій щодо харчування та фізичної активності. Бот доступний у будь-який момент і в будь-якому місці, де є доступ до Інтернету, що робить його корисним для людей з насиченим графіком. Використання перевірених формул та наукових методів гарантує точність і надійність рекомендацій.

Цей проект є результатом моєї пристрасті до програмування та бажання допомогти людям досягати своїх цілей у сфері здоров'я та фітнесу, використовуючи сучасні технології та науковий підхід. У найближчому

майбутньому я планую розширити функціонал бота, додавши базу рецептів, функції відстеження прогресу, персональні нагадування та тренувальні плани, а також розпочати комерційне використання бота з можливістю купівлі передплат.

1 ВСТУП ДО ФІТНЕС-ТЕЛЕГРАМ-БОТІВ

1.1 Актуальність розробки фітнес-телеграм-ботів

У сучасному світі, де стрес, брак часу та недостатня фізична активність стали нормою, ведення здорового способу життя перетворилося на справжній виклик. Численні дослідження свідчать про зростання рівня ожиріння, цукрового діабету та інших захворювань, пов'язаних з неправильним харчуванням і браком руху. Водночас, люди все більше усвідомлюють важливість піклування про своє здоров'я та фізичну форму, шукаючи доступні та зручні способи отримання персоналізованих порад і підтримки.

Стрімкий розвиток технологій відкриває нові можливості у сфері фітнесу та здорового способу життя. Мобільні додатки, веб-сайти та месенджери стають платформами, які дозволяють користувачам отримувати корисну інформацію, відстежувати свій прогрес і взаємодіяти з експертами у зручний для них спосіб. Серед цих інструментів особливо виділяються телеграм-боти, які забезпечують швидкий і зручний доступ до персоналізованих рекомендацій прямо у месенджері.

Популярність Телеграму невпинно зростає, і станом на 2023 рік він налічує понад 700 мільйонів активних користувачів у всьому світі (див. рис.1.1). Люди цінують Телеграм за його швидкість, безпеку та широкий спектр функцій, включаючи можливість створювати та використовувати ботів. Фітнес-телеграм-боти стають все більш затребуваними, адже вони дозволяють користувачам отримувати персоналізовані рекомендації щодо харчування, фізичних вправ і способу життя в режимі реального часу, без необхідності завантажувати додаткове програмне забезпечення або відвідувати спеціалізовані веб-сайти.

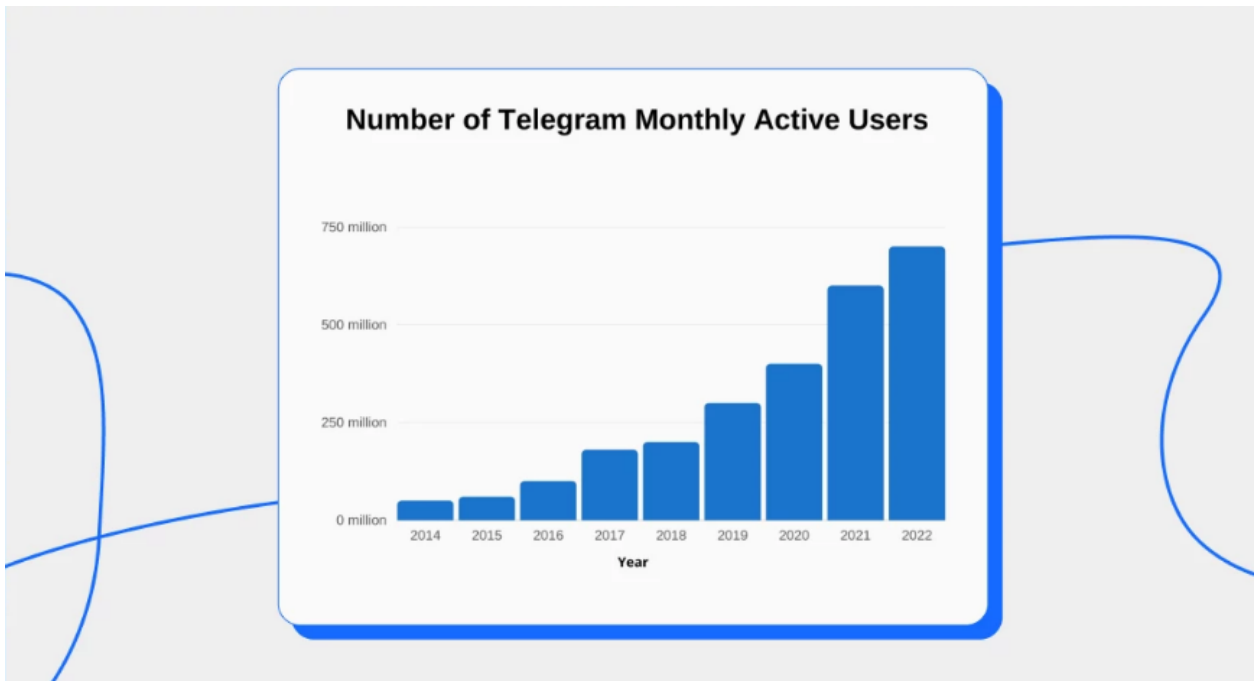


Рисунок 1.1 – Статистика активних користувачів Телеграму

Розробка фітнес-телеграм-бота, здатного розраховувати індивідуальні енергетичні потреби на основі достовірних формул та алгоритмів, надавати персоналізовані рекомендації та мотивувати користувачів дотримуватися здорового способу життя, є актуальним завданням, яке може допомогти людям досягти їхніх фітнес-цілей більш ефективно та зручно.

1.2 Виявлення основних можливостей та обмежень існуючих оптимізаційних інструментів

На ринку присутня низка додатків, веб-сайтів і телеграм-ботів, розроблених для підтримки здорового способу життя та досягнення фітнес-цілей. Однак більшість з них мають певні обмеження або недоліки, що знижують їхню ефективність та зручність використання.

Мобільні додатки для фітнесу та підрахунку калорій, такі як MyFitnessPal, Lose It! та Lifesum (див. рис. 1.2 – 1.4), безумовно, є популярними рішеннями. Вони пропонують користувачам інструменти для відстеження

харчування, фізичної активності та прогресу. Однак ці додатки часто є досить складними для використання, вимагають ручного введення великої кількості даних і не завжди надають персоналізовані рекомендації на основі індивідуальних потреб користувача.



Рисунок 1.2 – Мобільний додаток MyFitnessPal

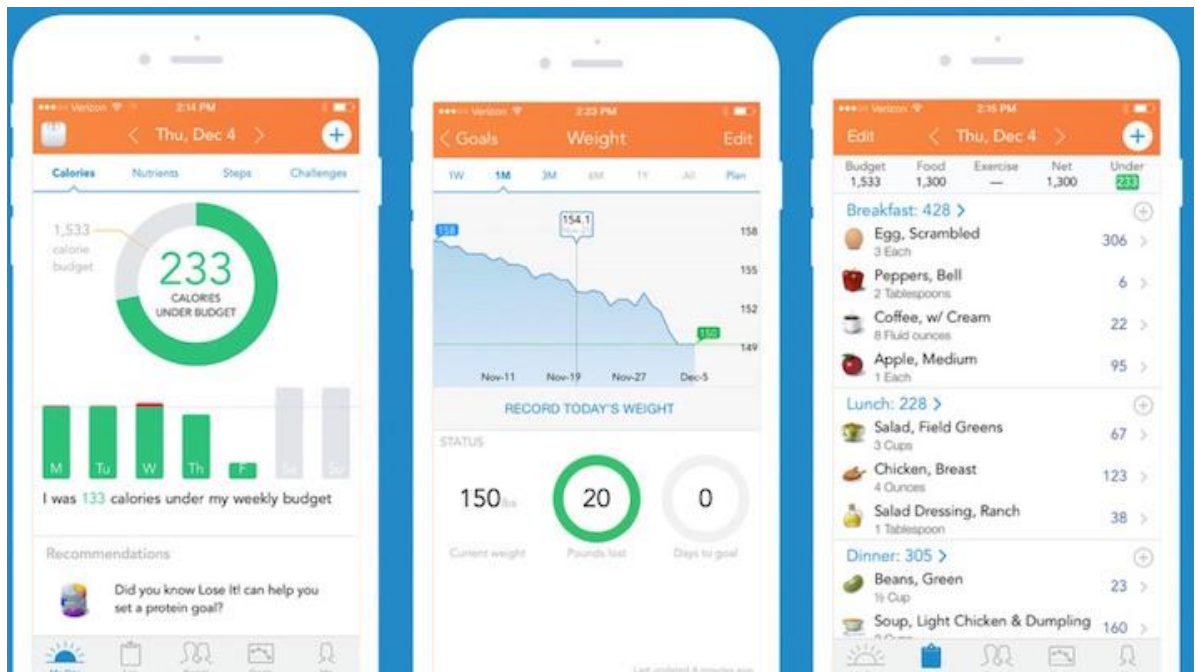


Рисунок 1.3 – Мобільний додаток Lose It!

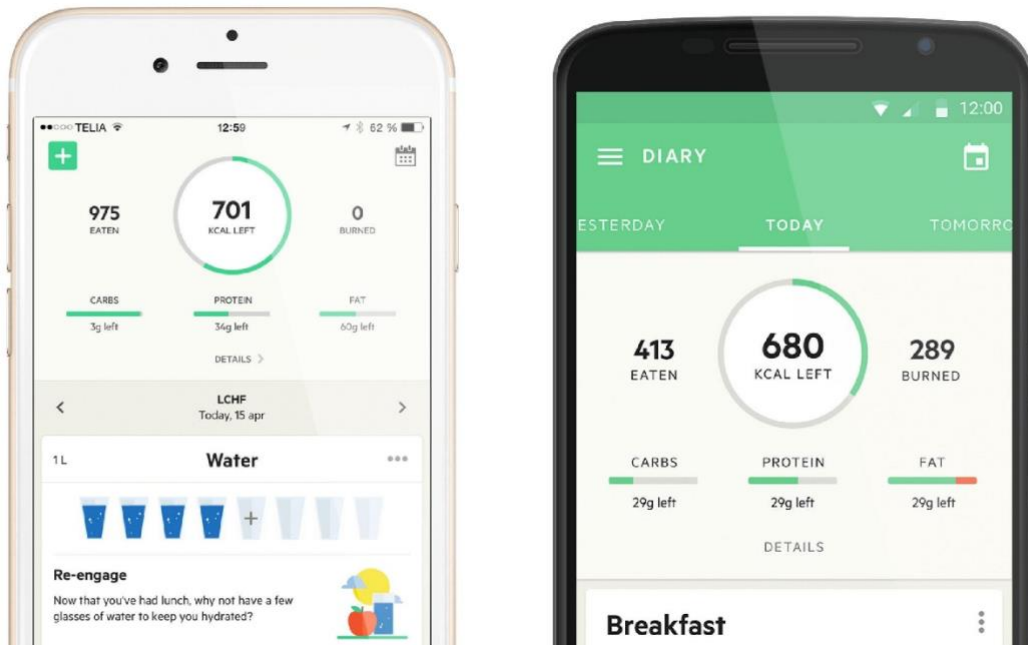


Рисунок 1.4 – Мобільний додаток Lifesum

Існують також численні веб-сайти, присвячені здоровому харчуванню та фітнесу, такі як Eat This Much, FitDay та SparkPeople (див. рис.1.5 – 1.7). Вони пропонують користувачам різноманітні функції, включаючи калькулятори калорій, плани дієт та тренувань. Проте більшість цих веб-сайтів мають громіздкі інтерфейси, вимагають створення облікових записів та часто містять надлишкову інформацію, яка може бути складною для сприйняття.

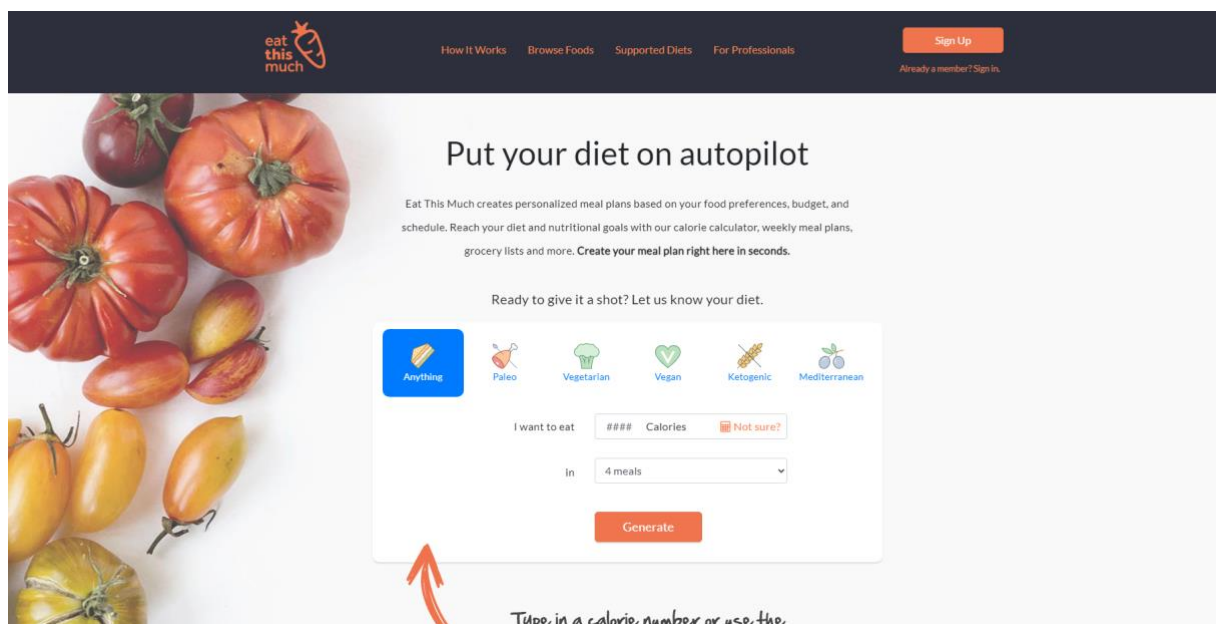


Рисунок 1.5 – Веб-сайт Eat This Much

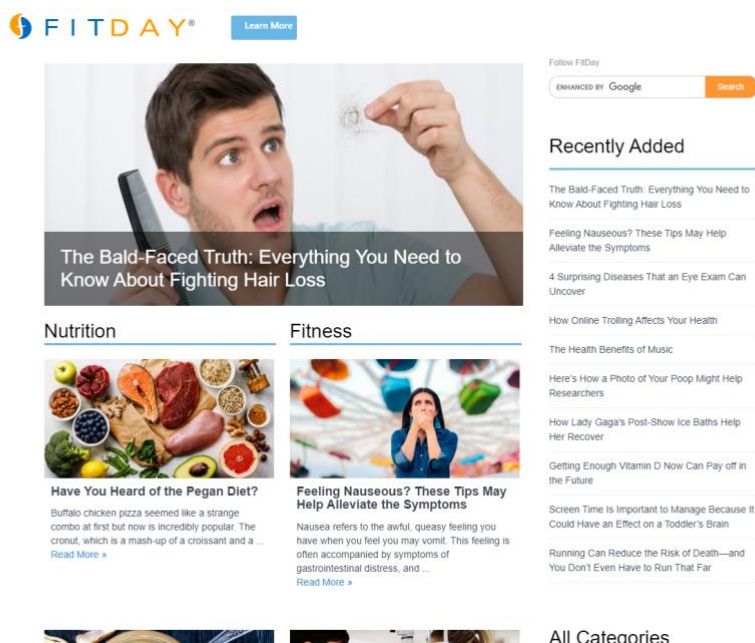


Рисунок 1.6 – Веб-сайт FitDay

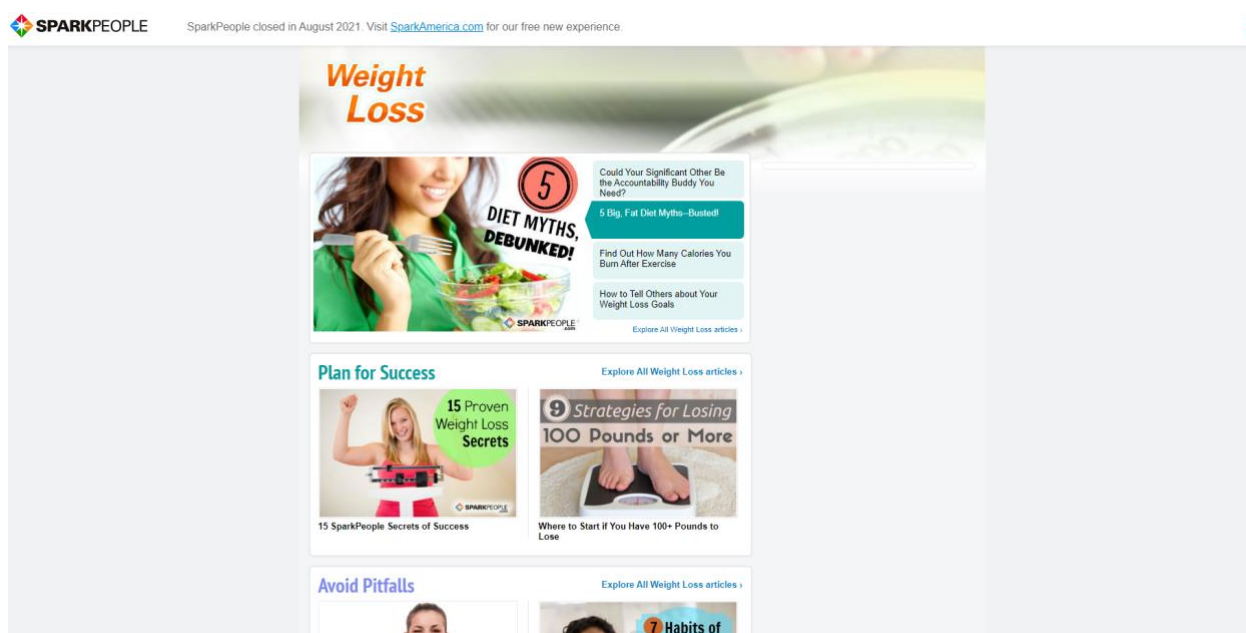


Рисунок 1.7 – Веб-сайт SparkPeople

Що стосується телеграм-ботів для фітнесу, то їх кількість на ринку досить обмежена. Наявні боти зазвичай пропонують лише базові функції, такі як підрахунок калорій або надання загальних порад щодо харчування та вправ. Вони рідко забезпечують персоналізовані рекомендації, засновані на індивідуальних параметрах користувача, і часто не є інтуїтивними у використанні.

Одним з основних недоліків існуючих рішень є відсутність комплексного та зручного підходу до надання персоналізованих рекомендацій щодо харчування та фізичної активності. Більшість додатків і веб-сайтів вимагають від користувачів самотійно вводити велику кількість даних та інформації, що робить процес громіздким і незручним. Крім того, часто рекомендації не враховують повною мірою індивідуальні параметри користувача, такі як вік, стать, рівень фізичної активності тощо.

Телеграм-боти, які пропонують персоналізовані рекомендації на основі перевірених наукових формул та алгоритмів, здатні усунути ці недоліки. Вони забезпечують зручний та інтуїтивно зрозумілий спосіб взаємодії з користувачем, не вимагаючи завантаження додаткового програмного забезпечення або створення облікових записів. Такі боти можуть надавати персоналізовані поради щодо харчування та фізичної активності, враховуючи індивідуальні потреби користувача, і таким чином, допомагати досягати бажаних фітнес-цілей більш ефективно та зручно.

1.3 Мета та завдання роботи

Метою даної кваліфікаційної роботи є розробка інтелектуального фітнес-телеграм-бота, здатного надавати персоналізовані рекомендації щодо харчування та фізичної активності на основі наукових формул та алгоритмів. Цей бот покликаний стати зручним і доступним помічником для користувачів, які прагнуть вести здоровий спосіб життя та досягати своїх фітнес-цілей.

Для досягнення поставленої мети необхідно виконати ряд важливих завдань:

- ґрунтовно вивчити теоретичні основи розрахунку енергетичних потреб організму, зокрема формули Харріса-Бенедикта та Міффіна-Джеора, а також методики визначення рівня фізичної активності;

- обґрунтувати вибір найбільш точної та надійної формули для розрахунку денної норми калорій, враховуючи її популярність, наукову обґрунтованість та перевірену ефективність;
- розробити алгоритм та програмний код для реалізації фітнес-телеграм-бота, забезпечивши його здатність збирати необхідні дані від користувача, обчислювати індивідуальні енергетичні потреби та надавати персоналізовані рекомендації щодо харчування та фізичної активності;
- створити зручний та інтуїтивно зрозумілий інтерфейс взаємодії з користувачем, який дозволить легко вводити необхідні дані та отримувати рекомендації в привабливому та доступному форматі;
- провести ретельне тестування фітнес-телеграм-бота, забезпечивши його коректну роботу в різних сценаріях використання та стабільність при обробці різноманітних вхідних даних;
- визначити перспективи подальшого розвитку та вдосконалення функціональності фітнес-телеграм-бота, такі як інтеграція бази рецептів, відстеження прогресу користувачів, персональні нагадування та надання тренувальних планів;
- дослідити можливості комерціалізації фітнес-телеграм-бота, розглянувши варіанти монетизації, такі як платні передплати або додаткові платні функції.

Реалізація поставлених завдань дозволить створити унікальний і корисний продукт, який поєднає переваги месенджера Телеграм, науково обґрунтовані алгоритми та інтуїтивний інтерфейс. Фітнес-телеграм-бот стане цінним інструментом для користувачів, які прагнуть досягти своїх фітнес-цілей зручним та ефективним способом, отримуючи персоналізовані рекомендації безпосередньо у своєму улюбленому месенджері.

2 ТЕОРЕТИЧНІ ОСНОВИ РОЗРАХУНКУ ЕНЕРГЕТИЧНИХ ПОТРЕБ

2.1 Формула Харріса-Бенедикта

Формула Харріса-Бенедикта є однією з найбільш відомих і широко використовуваних формул для розрахунку базової добової витрати енергії (базового метаболізму) людини. Вона була розроблена в 1919 році вченими Джеймсом Артуром Харрісом і Френсісом Гано Бенедиктом і стала золотим стандартом у сфері дієтології та фітнесу завдяки своїй точності та універсальності.

Формула Харріса-Бенедикта враховує такі ключові фактори, як вага, зріст і вік людини, для обчислення базового метаболізму - кількості калорій, яку організм витрачає у стані повного спокою для підтримки життєво важливих функцій. Ця інформація є критично важливою для визначення загальної добової потреби в калоріях та складання збалансованого раціону харчування.

Формула Харріса-Бенедикта:

$$BMR = (447,593 + (9,247 \times \text{вага [кг]}) + (3,098 \times \text{зріст [см]}) - (4,33 \times \text{вік [кількість років]})) \times \text{рівень фізичної активності}$$

Після обчислення базового метаболізму, отримане значення множиться на коефіцієнт фізичної активності, який враховує рівень рухової активності людини протягом дня. Це дозволяє розрахувати загальну добову потребу в калоріях для підтримки ваги або досягнення цілей по схудненню чи набору ваги.

Популярність формули Харріса-Бенедикта пояснюється її високою точністю та надійністю. Вона була перевірена численними дослідженнями на різних вікових групах і етнічних групах, демонструючи стабільно високу точність у порівнянні з іншими формулами та методами розрахунку базового метаболізму.

Крім того, формула Харріса-Бенедикта є зручною у використанні, оскільки вимагає лише кількох базових параметрів, які легко отримати. Це робить її доступною для широкого кола користувачів, включаючи фітнес-ентузіастів, дієтологів та фахівців зі здоров'я.

Завдяки своїй точності, простоті та науковій обґрунтованості, формула Харріса-Бенедикта стала золотим стандартом у галузі дієтології та фітнесу, широко використовуючись у додатках, онлайн-калькуляторах та програмах для розрахунку енергетичних потреб людини.

2.2 Рівні фізичної активності

Рівень фізичної активності є одним з ключових факторів, який необхідно враховувати при розрахунку загальної добової потреби в калоріях. Він відображає, скільки енергії людина витрачає на рухову активність протягом дня, включаючи заняття спортом, роботу, домашні справи та інші види активності.

Зазвичай виділяють п'ять основних рівнів фізичної активності, які використовуються в різних формулах для розрахунку енергетичних потреб, таких як формула Харріса-Бенедикта:

Сидячий спосіб життя (мало або зовсім не фізичної активності)

Цей рівень характерний для людей, які проводять більшу частину дня сидячи, наприклад, працюючи в офісі або займаючись іншими малорухомими видами діяльності. Коефіцієнт фізичної активності для цього рівня зазвичай становить від 1,2 до 1,3.

Невелика активність (легкі вправи/спорт 1-3 дні на тиждень)

Цей рівень підходить для людей, які ведуть в цілому малорухомий спосіб життя, але займаються легкими вправами або помірною фізичною активністю 1-3 рази на тиждень. Коефіцієнт активності в цьому випадку зазвичай коливається від 1,375 до 1,45.

Помірна активність (помірні вправи/спорт 3-5 днів на тиждень)

Цей рівень характерний для осіб, які регулярно займаються помірними вправами або спортом 3-5 разів на тиждень. Коефіцієнт фізичної активності для цього рівня зазвичай становить від 1,55 до 1,64.

Висока активність (інтенсивні вправи/спорт 6-7 днів на тиждень)

Цей рівень відповідає людям, які ведуть дуже активний спосіб життя, займаючись інтенсивними видами спорту або фізичними вправами 6-7 днів на тиждень. Коефіцієнт активності в цьому випадку може коливатися від 1,725 до 1,8.

Дуже висока активність (професійні спортсмени або фізична робота)

Цей рівень характерний для професійних спортсменів або людей, чия робота пов'язана з дуже інтенсивною фізичною активністю. Коефіцієнт фізичної активності для цього рівня може становити від 1,9 до 2 або навіть вище.

Правильне визначення рівня фізичної активності є критично важливим для точного розрахунку енергетичних потреб організму. Недооцінка або переоцінка рівня активності може призвести до неточних результатів і, як наслідок, неефективних рекомендацій щодо раціону харчування та фізичних навантажень.

У фітнес-телеграм-боті важливо забезпечити зручний і інтуїтивний спосіб визначення рівня фізичної активності користувача, наприклад, за допомогою опису типових сценаріїв або інтерактивних запитань. Це дозволить отримати максимально точні результати розрахунків і надати користувачам відповідні персоналізовані рекомендації.

2.3 Персоналізація енергетичних потреб відповідно до цілей користувача

Одним з ключових аспектів успішного досягнення фітнес-цілей є персоналізація енергетичних потреб відповідно до індивідуальних цілей користувача. Чи то схуднення, набір м'язової маси чи підтримка поточної ваги, кожна мета вимагає відповідного коригування добової норми калорій.

Схуднення

Для ефективного і безпечного зниження ваги рекомендується створити дефіцит калорій, коли організм витрачає більше енергії, ніж споживає з їжею. Загальноприйнятим підходом є зменшення добової норми калорій на 15-20% від базового метаболізму. Це забезпечує поступове і стійке зниження ваги без шкоди для здоров'я.

Наприклад, якщо розрахункова добова потреба в калоріях для підтримки поточної ваги становить 2000 ккал, при створенні дефіциту в 20% добова норма скоротиться до 1600 ккал. Такий підхід дозволяє організму поступово використовувати запаси жиру як джерело енергії, що призводить до схуднення.

Набір м'язової маси

У цьому випадку потрібно створити профіцит калорій, коли організм споживає більше енергії, ніж витрачає. Додатковий надлишок калорій необхідний для забезпечення процесу синтезу нових м'язових волокон. Зазвичай рекомендують збільшувати добову норму калорій на 10-20% від базового метаболізму.

Наприклад, якщо розрахункова добова потреба в калоріях становить 2500 ккал, при створенні профіциту в 15% добова норма зросте до 2875 ккал. Цей додатковий калоражний надлишок в поєднанні з належним тренувальним режимом призведе до нарощування м'язової маси.

Підтримка ваги

Для підтримки поточної ваги необхідно дотримуватися розрахованої добової норми калорій, яка відповідає базовому метаболізму з урахуванням рівня фізичної активності. У цьому випадку немає потреби створювати дефіцит або профіцит калорій.

Важливо пам'ятати, що ці рекомендації є загальними орієнтирами, і для кожного окремого користувача необхідно враховувати його індивідуальні особливості, такі як вік, стать, склад тіла, стан здоров'я та інші фактори. Фітнес-телеграм-бот повинен забезпечувати персоналізацію енергетичних потреб на основі введених користувачем даних та обраної ним мети.

Крім того, рекомендації можуть включати не лише коригування добової норми калорій, але й поради щодо оптимального співвідношення макронутрієнтів (білків, жирів і вуглеводів) відповідно до мети користувача. Це дозволить забезпечити комплексний підхід до досягнення бажаних результатів у фітнесі та здоров'ї.

3 РЕАЛІЗАЦІЯ ФІТНЕС-ТЕЛЕГРАМ-БОТА

3.1 Вибір технологій та інструментів розробки

Розробка фітнес-телеграм-бота вимагає ретельного вибору відповідних технологій та інструментів, які забезпечать ефективність, безпеку та масштабованість майбутнього продукту. У цьому підрозділі ми розглянемо ключові технології та інструменти, які були обрані для реалізації проекту, а також обґрунтування цього вибору.

Мова програмування Python

Для написання коду фітнес-телеграм-бота була обрана мова програмування Python. Цей вибір зумовлений кількома важливими перевагами Python:

- легкість у вивченні та використанні;
- потужні бібліотеки для роботи з даними та обробки природної мови;
- широка підтримка в спільноті розробників;
- дозволяє швидко створювати прототипи;
- забезпечує високу продуктивність при розробці додатків різного масштабу та складності.

Бібліотека pyTelegramBotAPI

Для інтеграції з Telegram Messenger і створення телеграм-бота була використана бібліотека pyTelegramBotAPI. Ця бібліотека забезпечує зручний і добре задокументований інтерфейс для взаємодії з Telegram Bot API, що значно спрощує процес розробки та керування ботом. pyTelegramBotAPI дозволяє легко обробляти вхідні повідомлення від користувачів, надсилати відповіді, обробляти команди та взаємодіяти з різними типами вмісту, такими як текст, зображення та кнопки.

Середовище розробки

Для створення фітнес-телеграм-бота було обрано середовище розробки Thonny. Це легковагове середовище, орієнтоване насамперед на навчання програмуванню мовою Python. Незважаючи на свою простоту, Thonny пропонує корисні функції, такі як редактор коду з підсвічуванням синтаксису, інтегрований інтерпретатор Python та зручні інструменти для налагодження.

Вибір Thonny був зумовлений кількома причинами:

- має інтуїтивний та мінімалістичний інтерфейс, що робить його ідеальним для початківців програмістів або тих, хто вперше знайомиться з Python;
- не вимагає складного налаштування середовища та встановлення додаткових залежностей, що полегшує процес розробки;
- був спеціально розроблений для полегшення навчання програмуванню на Python;
- його функції, такі як покрокове виконання коду та відстеження змінних, допомагають зрозуміти поведінку програми;
- дозволяє встановлювати та використовувати сторонні пакети Python, такі як pyTelegramBotAPI та googletrans, необхідні для розробки фітнес-бота.

Бібліотека telebot

Ця бібліотека використовується для створення та керування Telegram ботом. Вона надає класи та методи для обробки повідомлень від користувачів, відправлення відповідей, обробки команд тощо. У коді імпортується за допомогою рядка `import telebot`.

Модуль types з telebot

Містить різні класи та типи даних, які використовуються для взаємодії з Telegram Bot API. У коді він імпортується за допомогою рядка `from telebot import types`. Зокрема, у коді використовується клас `types.ReplyKeyboardMarkup` для створення клавіатури з кнопками для взаємодії з користувачем.

Модулі random та randrange

Стандартним модулем Python, який забезпечує функції для генерації псевдовипадкових чисел. У коді він імпортується за допомогою рядка `import random`. Окрема функція `randrange` імпортується з цього модуля за допомогою рядка `from random import randrange`. Ці імпорти не використовуються безпосередньо в наданому коді, але можуть бути використані в подальшому для генерації випадкових чисел або випадкового вибору елементів.

Модуль time

Стандартний модуль Python, який надає різні функції для роботи з часом. У коді він імпортується за допомогою рядка `import time`. У наданому коді він використовується в функції `main()` для затримки перед перезапуском бота у випадку виникнення помилки.

Вибір цих технологій та інструментів зумовлений їх популярністю, надійністю, зручністю використання та відповідністю вимогам проекту. Вони забезпечують ефективну розробку, легку інтеграцію з Telegram Messenger, підтримку багатомовності, зручне середовище для написання коду та можливості для масштабування та розгортання фітнес-телеграм-бота в майбутньому.

3.2 Вибір технологій та інструментів розробки

Архітектура фітнес-телеграм-бота була розроблена з урахуванням принципів модульності, масштабованості та зручності підтримки. Вона базується на архітектурному патерні "Обробник подій" (Event-Driven Architecture), який відмінно підходить для розробки чат-ботів та інших додатків, що реагують на вхідні події та повідомлення від користувачів.

Центральним компонентом архітектури є головний цикл обробки подій (Event Loop), який керує потоком взаємодії між користувачем і ботом. Цей цикл постійно перевіряє наявність нових повідомлень від користувачів

Telegram і передає їх відповідним обробникам повідомлень (Message Handlers).

Обробники повідомлень є окремими модулями, які відповідають за обробку певних типів повідомлень або команд від користувачів. Кожен обробник повідомлень містить логіку для аналізу вхідних даних, виконання необхідних обчислень або запитів до зовнішніх сервісів (наприклад, для перекладу тексту або отримання інформації про харчові продукти) та формування відповіді для надсилання користувачеві.

Одним з ключових обробників повідомлень є обробник команди `"/start"`, який ініціалізує взаємодію з користувачем, представляє основні функції бота та пропонує користувачеві вибрати подальші дії за допомогою спеціальної клавіатури з кнопками.

Іншим важливим обробником є обробник для розрахунку енергетичних потреб користувача. Цей обробник використовує вхідні дані від користувача (вага, зріст, вік та рівень фізичної активності) для розрахунку денної норми калорій за формулою Харріса-Бенедикта або Міффіна-Джеора. Отримані результати потім представляються користувачеві разом з рекомендаціями щодо збільшення або зменшення калорійності залежно від його цілей (схуднення або набір маси).

Для зберігання тимчасових даних користувача під час розрахунку енергетичних потреб використовується спеціальний модуль зберігання даних (Data Storage). Цей модуль забезпечує безпечне зберігання та доступ до проміжних даних, таких як вага, зріст та вік користувача, протягом усього процесу розрахунку.

Архітектура фітнес-телеграм-бота також включає модуль керування помилками (Error Handling), який відповідає за обробку та повідомлення про будь-які виняткові ситуації або помилки, що виникають під час роботи бота. Цей модуль забезпечує відправлення повідомлень про помилки розробникам або адміністраторам бота для подальшого аналізу та вирішення проблем.

Для забезпечення масштабованості та можливості розширення функціоналу в майбутньому, архітектура фітнес-телеграм-бота була спроектована з використанням принципів модульності та слабого зв'язку між компонентами. Це дозволяє легко додавати нові обробники повідомлень, інтегрувати додаткові сервіси або модифікувати існуючу функціональність без необхідності змінювати весь код бота.

Така архітектура забезпечує ефективну обробку вхідних повідомлень від користувачів, надійний розрахунок енергетичних потреб, зручний інтерфейс взаємодії з ботом, а також можливість легкого масштабування та розширення функціональності в майбутньому.

Повний код Телеграм боту представлений у додатку А.

3.3 Опис функціональності бота

Фітнес-телеграм-бот пропонує користувачам зручний та інтерактивний спосіб обчислення їхніх денних енергетичних потреб, а також отримання персоналізованих рекомендацій щодо корекції калорійності залежно від їхніх фітнес-цілей. Основна функціональність бота розділена на кілька ключових модулів, які забезпечують плавну взаємодію з користувачем та точні розрахунки.

3.3.1 Обробка вхідних даних користувача

Після запуску бота користувачеві показано дві формули, для розрахунку енергетичних потреб: формулу Харріса-Бенедикта або формулу Міффіна-Джеора (див. рис.3.1). Після перегляду формули бот розпочинає інтерактивний діалог, під час якого запитує у користувача необхідні дані, такі як вага, зріст, вік та рівень фізичної активності.

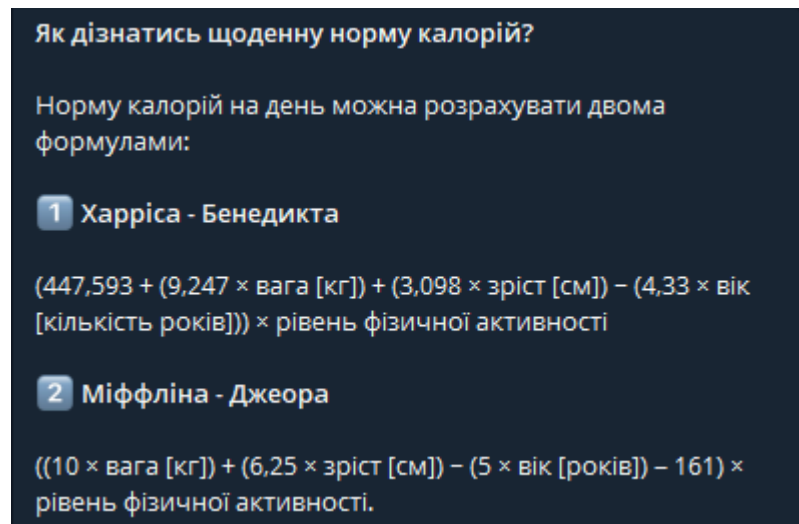


Рисунок 3.1 – Ознайомлення з формулами

Для забезпечення зручності взаємодії та запобігання помилкам, бот використовує спеціальну клавіатуру з кнопками для отримання певних даних від користувача. Це дозволяє уникнути необхідності вводити дані вручну та зменшує ризик некоректного введення даних.

Під час процесу збору даних бот зберігає проміжні значення у тимчасовому сховищі, щоб не втратити їх при переході між різними етапами розрахунку. Це забезпечує безперебійну взаємодію та дозволяє користувачеві легко відновлювати процес у разі будь-яких перерв або перезапусків бота.

3.3.2 Розрахунок енергетичних потреб та надання рекомендацій

Після отримання всіх необхідних даних від користувача, бот виконує розрахунок денної норми калорій, використовуючи обрану формулу (Харріса-Бенедикта або Міффіна-Джеора). Ці формули враховують такі параметри, як вага, зріст, вік та рівень фізичної активності користувача, для точного визначення його енергетичних потреб.

Після завершення розрахунків бот надає користувачеві результат у форматі, зручному для сприйняття. Крім самої денної норми калорій, бот також пропонує рекомендації щодо корекції калорійності залежно від цілей користувача (див. рис.3.2). Наприклад, якщо користувач хоче схуднути, бот

рекомендує зменшити калорійність на певний відсоток, а якщо метою є набір м'язової маси, бот пропонує збільшити калорійність.

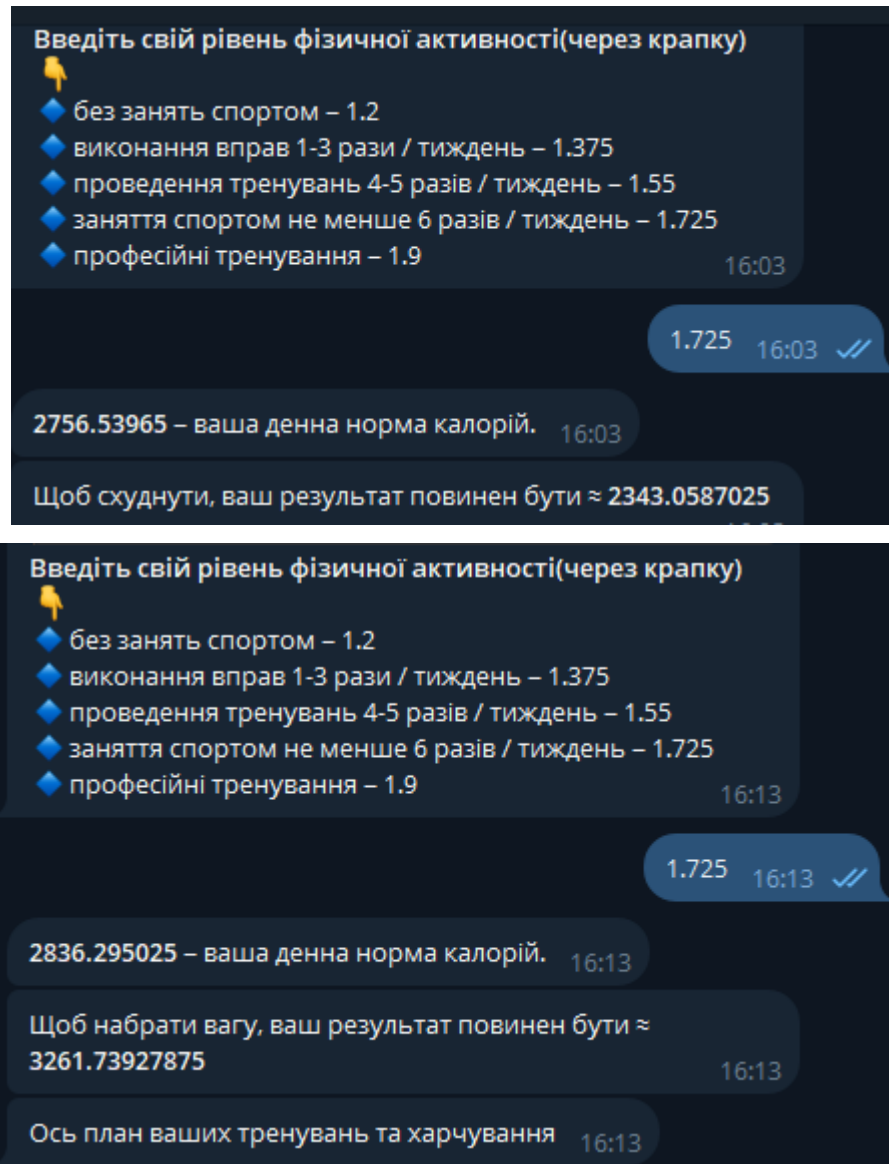


Рисунок 3.2 – Рекомендації

Ці рекомендації базуються на загальноприйнятих принципах фітнесу та дієтології, а також враховують індивідуальні характеристики користувача, такі як вага, зріст та рівень фізичної активності. Це дозволяє надавати персоналізовані та ефективні поради для досягнення бажаних результатів.

3.3.3 Інтерактивний інтерфейс та навігація

Протягом усієї взаємодії з ботом користувачеві надається зручний та інтуїтивний інтерфейс для навігації та вибору дій. Бот використовує спеціальні клавіатури з кнопками, які дозволяють користувачеві легко переходити між різними етапами розрахунку, повертатися до попередніх кроків або змінювати вибір формули для розрахунку.

Крім того, бот забезпечує чітку та зрозумілу комунікацію з користувачем, надаючи докладні інструкції та пояснення на кожному етапі процесу. Це допомагає уникнути плутанини та забезпечує безпроблемну взаємодію навіть для тих користувачів, які вперше використовують фітнес-бота.

Функціональність фітнес-телеграм-бота спрямована на надання користувачам зручного та точного способу обчислення їхніх енергетичних потреб, отримання персоналізованих рекомендацій та досягнення бажаних фітнес-цілей.

3.4 Перспективи розвитку фітнес-телеграм-бота

Фітнес-телеграм-бот, розроблений у рамках цього проекту, є потужним інструментом для розрахунку енергетичних потреб та надання персоналізованих рекомендацій користувачам. Однак, існує широкий спектр можливостей для подальшого розвитку та удосконалення функціональності бота, що дозволить зробити його ще більш корисним та привабливим для користувачів.

3.4.1 Інтеграція бази рецептів

Однією з перспективних можливостей для розширення функціональності фітнес-телеграм-бота є інтеграція бази рецептів здорових страв. Ця база даних може містити детальну інформацію про калорійність, поживні речовини та інгредієнти різноманітних рецептів, що дозволить користувачам легко планувати та контролювати своє харчування відповідно до їхніх енергетичних потреб та фітнес-цілей.

Користувачі зможуть отримувати рекомендовані рецепти на основі своїх персональних даних, таких як денна норма калорій, харчові переваги та дієтичні обмеження.

3.4.2 Відстеження прогресу

Ще одним важливим напрямком розвитку фітнес-телеграм-бота є впровадження функції відстеження прогресу користувачів. Ця функція дозволить користувачам вводити свої поточні дані, такі як вага, обхвати тіла та показники фізичної форми, та відстежувати зміни з часом.

Бот зможе візуалізувати прогрес користувача за допомогою графіків та діаграм, що полегшить аналіз та мотивацію для продовження фітнес-практик. Крім того, бот може надавати персоналізовані рекомендації щодо коригування дієти, тренувань або способу життя на основі аналізу прогресу користувача.

3.4.3 Персональні нагадування

Для підвищення залученості користувачів та покращення дотримання фітнес-режиму, фітнес-телеграм-бот може бути доповнений функцією персональних нагадувань. Ця функція дозволить користувачам встановлювати нагадування про прийом їжі, тренування, вимірювання показників та інші важливі події, пов'язані з їхніми фітнес-цілями.

Нагадування можуть бути налаштовані з урахуванням індивідуального розкладу користувача, а також можуть містити корисні підказки та мотиваційні повідомлення для підтримки користувачів на шляху до досягнення їхніх цілей.

3.4.4 Тренувальні плани

Для забезпечення комплексного підходу до фітнесу, фітнес-телеграм-бот може бути розширений шляхом інтеграції тренувальних планів. Користувачі зможуть отримувати персоналізовані тренувальні програми, засновані на їхніх фізичних даних, цілях та перевагах щодо типів тренувань.

Бот може надавати детальні інструкції з виконання вправ, включаючи відео або анімації для демонстрації правильної техніки. Крім того, бот може відстежувати прогрес користувача в тренуваннях та автоматично коригувати плани залежно від досягнутих результатів.

Інтеграція цих нових функцій не лише зробить фітнес-телеграм-бота більш комплексним та корисним інструментом для користувачів, але й дозволить створити повноцінну екосистему для підтримки здорового способу життя та досягнення фітнес-цілей. Регулярні оновлення та вдосконалення бота забезпечать його актуальність і привабливість для широкої аудиторії користувачів, які прагнуть покращити своє фізичне здоров'я та самопочуття.

3.4.5 Інформаційна база про спортивне харчування та рекомендації

Для надання користувачам більш повної підтримки у досягненні їхніх фітнес-цілей, фітнес-телеграм-бот може бути доповнений розділом з інформацією про спортивне харчування та рекомендаціями щодо здоров'я і безпеки під час тренувань.

Цей розділ може містити статті та корисні відомості про такі популярні спортивні добавки, як протеїн, гейнер, креатин та L-аргінін. Користувачі зможуть отримувати детальну інформацію про склад, дозування, переваги та можливі побічні ефекти цих добавок, що допоможе їм зробити обґрунтований вибір при плануванні свого раціону.

Крім того, бот може надавати рекомендації щодо адекватного споживання рідини під час тренувань та в повсякденному житті. Ця функція враховуватиме індивідуальні фактори, такі як вага тіла, інтенсивність тренувань та кліматичні умови, для забезпечення оптимального рівня гідратації та запобігання ризикам дегідратації.

Нарешті, у цьому розділі можуть бути розміщені поради щодо профілактики травм та належної техніки виконання вправ. Бот може надавати відеоінструкції та анімації, що демонструють правильну форму під час різних типів тренувань, а також рекомендації щодо розминки, розтягування та відновлення після інтенсивних навантажень.

Інтеграція такого інформаційного розділу в фітнес-телеграм-бота забезпечить користувачів цінними знаннями та рекомендаціями, що сприятиме безпечному та ефективному досягненню їхніх фітнес-цілей.

ВИСНОВКИ

Розробка фітнес-телеграм-бота, представлена в цій дипломній роботі, є важливим кроком у напрямку створення зручних та ефективних інструментів для підтримки здорового способу життя та досягнення фітнес-цілей. Цей проект продемонстрував можливість поєднання сучасних технологій, наукових підходів та інтуїтивного інтерфейсу для надання користувачам персоналізованих рекомендацій щодо енергетичних потреб.

Завдяки ретельному вивченню теоретичних основ розрахунку енергетичних потреб, таких як формули Харріса-Бенедикта та Міффіна-Джеора, а також урахуванню рівнів фізичної активності, фітнес-телеграм-бот здатний надавати точні та науково обґрунтовані результати. Ця функціональність забезпечує користувачам надійну основу для планування та коригування свого харчування відповідно до їхніх індивідуальних потреб та цілей.

Вибір сучасних технологій та інструментів розробки, таких як мова програмування Python, бібліотеки pyTelegramBotAPI та googletrans, а також зручне середовище розробки Thonny, забезпечив ефективну реалізацію фітнес-телеграм-бота та можливість його подальшого вдосконалення та масштабування.

Архітектура фітнес-телеграм-бота, побудована на принципах модульності та слабкого зв'язку між компонентами, дозволяє легко додавати нові функції та інтегрувати додаткові сервіси в майбутньому. Наприклад, інтеграція бази рецептів здорових страв, функції відстеження прогресу, персональних нагадувань та тренувальних планів може значно розширити можливості бота та зробити його ще більш корисним інструментом для користувачів.

Крім того, інтерактивний інтерфейс бота, що включає клавіатури з кнопками та чітку навігацію, забезпечує зручну та інтуїтивну взаємодію з

користувачами, спрощуючи процес розрахунку енергетичних потреб та отримання рекомендацій.

Загалом, розробка фітнес-телеграм-бота є успішним прикладом поєднання сучасних технологій та наукових підходів для вирішення актуальної проблеми – надання користувачам персоналізованих рекомендацій щодо енергетичних потреб та підтримки здорового способу життя. Цей проект відкриває широкі перспективи для подальшого вдосконалення та розширення функціональності, що дозволить зробити фітнес-телеграм-бот потужним та універсальним інструментом для досягнення фітнес-цілей.

ПЕРЕЛІК ПОСИЛАНЬ

1. Thonny: The Beginner-Friendly Python Editor. Realpython. URL : <https://realpython.com/python-thonny/> (дата звернення: 14.02.2024).
2. How to Create a Telegram Bot using Python. Freecodecamp. URL : <https://www.freecodecamp.org/news/how-to-create-a-telegram-bot-using-python/> (дата звернення: 04.01.2024).
3. Як розрахувати свій базовий метаболізм: формула харріса-бенедикта. Apollo next. URL : <https://apollo.online/blog-post/yak-rozrahuvaty-svij-bazovuj-metabolizm-formula-harrisa-benedykta/> (дата звернення: 16.03.2024).
4. Детальне пояснення розрахунку страв. Калькулятор Калорій. URL : <https://calc.tablyscjakalorijnosti.com.ua/explanation> (дата звернення: 16.03.2024).
5. Найкращий час для постингу в Telegram у 2023 році. Webprom. URL : <https://web-promo.ua/ua/blog/najkrashij-chas-dlya-postingu-v-telegram-u-2023-roci/> (дата звернення: 06.01.2024).
6. Article About (Thonny) Python IDE. Linkedin. URL : <https://www.linkedin.com/pulse/article-thonny-python-ide-dr-tharanitharan-gurusamy-ilrwc> (дата звернення: 14.02.2024).
7. Thonny IDE User Guide. QuecPython. URL : https://python.quectel.com/doc/Application_guide/en/dev-tools/Thonny/index.html (дата звернення: 15.04.2024).
8. Nicolas Modrzyk. Building Telegram Bots: Develop Bots in 12 Programming Languages using the Telegram Bot API: learn about bot programming. Токуо: 2018. 293р. (дата звернення: 05.01.2024).
9. Fabrizio Romano, Gastón C. Hillar, Arun Ravindran. Learn Web Development with Python: Learning Path. UK: 2018. 779р. (дата звернення: 07.01.2024).

10. Telegram Bot API. Telegram Bots. URL:
<https://core.telegram.org/bots/api> (дата звернення: 08.02.2024).

ДОДАТОК А

Код Телеграм бота

```

# pip install pyTelegramBotAPI
# pip install googletrans
import telebot
from telebot import types
import random
from random import randrange
import time

bot = telebot.TeleBot('7090938274:AAGTQ4wSXo9XBN-Ia984x0LOhB6RFfq9nME')
storage = {}

def init_storage(user_id):
    storage[user_id] = dict(first_number=None, second_number=None)

def store_number(user_id, key, value):
    storage[user_id][key] = dict(value=value)

def get_number(user_id, key):
    return storage[user_id][key].get('value')

@bot.message_handler(commands=['start', 'help'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("🍷 Набрати")
    btn10 = types.KeyboardButton("🍏 Схуднути")

    markup.add(btn1, btn10)
    user = message.from_user
    user_id = user.id
    first_name = user.first_name
    last_name = user.last_name
    username = user.username
    bot.send_message(message.chat.id, text="<b>Як дізнатись щоденну норму калорій?</b>\n\nНорму калорій на день можна розрахувати двома формулами:\n\n□<b>Харріса - Бенедикта</b>\n\n(447,593 + (9,247 × вага [кг]) + (3,098 × зріст [см]) - (4,33 × вік [кількість років])) × рівень фізичної активності\n\n□<b>Міффіна - Джеора</b>\n\n((10 × вага [кг]) + (6,25 × зріст [см]) - (5 × вік [років]) - 161) × рівень фізичної активності.".format(message.from_user), reply_markup=markup, parse_mode="html")
    bot.send_message('540517419', f'Підключився новий користувач!!!\nID: {user_id}\nName: {first_name}\nLastname: {last_name}\nnik: {username}')

@bot.message_handler(content_types=['text'])
def func(message):
    init_storage(message.from_user.id)

```

```

if message.text == "👉 Набрати":
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b15 = types.KeyboardButton("👉 Далі ➡")
    markup.add(b15)
    bot.send_message(message.chat.id, text='Щоб я зміг Вам допомогти, розкажіть мені трохи про себе.👉\nНатисніть на кнопку "Далі ➡", щоб продовжити.👉'.format(message.from_user), reply_markup=markup)
    bot.register_next_step_handler(message, plus_one)

elif message.text == "🍲 Схуднути":
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b15 = types.KeyboardButton("🍲 Далі ➡")
    markup.add(b15)
    bot.send_message(message.chat.id, text='Щоб я зміг Вам допомогти, розкажіть мені трохи про себе.👉\nНатисніть на кнопку "Далі ➡", щоб продовжити.👉'.format(message.from_user), reply_markup=markup)
    bot.register_next_step_handler(message, plus_six)

elif message.text == "🍽 Харчування👉":
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("👉 🍳")
    b17 = types.KeyboardButton('👉 🍷')
    b18 = types.KeyboardButton("👉 🍷")
    b19 = types.KeyboardButton('👉 🍷')
    back1 = types.KeyboardButton('👉 Назад ⬅️')
    markup.add(b16, b17, b18, b19, back1)
    bot.send_message(message.chat.id, text='Оберіть👉\n👉 🍳 - Сніданок\n👉 🍷 - Обід\n👉 🍷 - Полудень\n👉 🍷 - Вечеря'.format(message.from_user), reply_markup=markup)

elif message.text == "👉 🍳":
    bot.send_message(message.chat.id, text='🍳 <b>Сніданок</b>\n2 яйця цілих\n10г волоських горіхів\n70г вівсянки + 5г кориці + ківі', parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("👉 🍳")
    b17 = types.KeyboardButton('👉 🍷')
    b18 = types.KeyboardButton("👉 🍷")
    b19 = types.KeyboardButton('👉 🍷')
    back1 = types.KeyboardButton('👉 Назад ⬅️')
    markup.add(b16, b17, b18, b19, back1)
    bot.send_message(message.chat.id, text='Оберіть👉\n👉 🍳 - Сніданок\n👉 🍷 - Обід\n👉 🍷 - Полудень\n👉 🍷 - Вечеря'.format(message.from_user), reply_markup=markup)

elif message.text == "👉 🍷":
    bot.send_message(message.chat.id, text='🍷 <b>Обід</b>\n150г курячого філе\n60г гречки зелені овочі яблуко', parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)

```

```

b16 = types.KeyboardButton("🍌")
b17 = types.KeyboardButton('🍌')
b18 = types.KeyboardButton("🍌")
b19 = types.KeyboardButton('🍌')
back1 = types.KeyboardButton('👉 Назад 🏠')
markup.add(b16,b17,b18,b19,back1)
bot.send_message(message.chat.id,text='Оберіть 👉\n🍌 - Сніданок\n\n🍌 - Обід\n\n🍌 - Полудень\n\n🍌 - Вечеря'.format(message.from_user),
reply_markup=markup)

```

```

elif message.text == "🍌":
    bot.send_message(message.chat.id,text='🍌 <b> Полудень</b>\n200г курячого
філе\n70г гречки\nзелені овочі + 1 чайна ложка оливкової олії',parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("🍌")
    b17 = types.KeyboardButton('🍌')
    b18 = types.KeyboardButton("🍌")
    b19 = types.KeyboardButton('🍌')
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.add(b16,b17,b18,b19,back1)
    bot.send_message(message.chat.id,text='Оберіть 👉\n🍌 - Сніданок\n\n🍌 - Обід\n\n🍌 - Полудень\n\n🍌 - Вечеря'.format(message.from_user),
reply_markup=markup)

```

```

elif message.text == "🍌":
    bot.send_message(message.chat.id,text='🍌 <b>Вечеря</b> \n100г курячого
філе\n50г авокадо або 15г волоських горіхів \nзелені овочі + 1 чайна ложка оливкової
олії',parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("🍌")
    b17 = types.KeyboardButton('🍌')
    b18 = types.KeyboardButton("🍌")
    b19 = types.KeyboardButton('🍌')
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.add(b16,b17,b18,b19,back1)
    bot.send_message(message.chat.id,text='Оберіть 👉\n🍌 - Сніданок\n\n🍌 - Обід\n\n🍌 - Полудень\n\n🍌 - Вечеря'.format(message.from_user),
reply_markup=markup)

```

```

elif message.text == "👉 Тренування 🍌":
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("👉 Фулбаді для чоловіків")
    b17 = types.KeyboardButton('👉 Жіночі тренування (з упором на сідниці)')
    b18 = types.KeyboardButton("👉 Чотириденний спліт (для чоловіків)")
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.row(b16,b17)
    markup.row(b18,back1)
    bot.send_message(message.chat.id,text='👉 Оберіть тренування, яке вам
потрібно 👉'.format(message.from_user), reply_markup=markup)

```

```

elif message.text == "👤 Фулбаді для чоловіків":
    markup = types.InlineKeyboardMarkup()
    button1 = types.InlineKeyboardButton("👤 Тренування💪",
url='https://telegra.ph/Fulbad%D1%96-dlya-cholov%D1%96k%D1%96v-06-20')
    markup.add(button1)
    bot.send_message(message.chat.id, text="👤 Фулбаді для чоловіків
👉".format(message.from_user), reply_markup=markup)
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("👤 Фулбаді для чоловіків")
    b17 = types.KeyboardButton('👤 Жіночі тренування (з упором на сідниці)')
    b18 = types.KeyboardButton("👤 Чотириденний спліт (для чоловіків)")
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.row(b16, b17)
    markup.row(b18, back1)
    bot.send_message(message.chat.id, text='💪 Оберіть тренування, яке вам
потрібно👉'.format(message.from_user), reply_markup=markup)

```

```

elif message.text == "👤 Жіночі тренування (з упором на сідниці)":
    markup = types.InlineKeyboardMarkup()
    button1 = types.InlineKeyboardButton("👤 Тренування💪",
url='https://telegra.ph/ZH%D1%96noch%D1%96-trenuvannya-z-uporom-na-
s%D1%96dnic%D1%96-06-20')
    markup.add(button1)
    bot.send_message(message.chat.id, text="👤 Жіночі тренування (з упором на
сідниці) 👉".format(message.from_user), reply_markup=markup)
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("👤 Фулбаді для чоловіків")
    b17 = types.KeyboardButton('👤 Жіночі тренування (з упором на сідниці)')
    b18 = types.KeyboardButton("👤 Чотириденний спліт (для чоловіків)")
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.row(b16, b17)
    markup.row(b18, back1)
    bot.send_message(message.chat.id, text='💪 Оберіть тренування, яке вам
потрібно👉'.format(message.from_user), reply_markup=markup)

```

```

elif message.text == "👤 Чотириденний спліт (для чоловіків)":
    markup = types.InlineKeyboardMarkup()
    button1 = types.InlineKeyboardButton("👤 Тренування💪",
url='https://telegra.ph/CHotiridennij-spl%D1%96t-dlya-cholov%D1%96k%D1%96v-06-20')
    markup.add(button1)
    bot.send_message(message.chat.id, text="👤 Чотириденний спліт (для чоловіків)
👉".format(message.from_user), reply_markup=markup)
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("👤 Фулбаді для чоловіків")
    b17 = types.KeyboardButton('👤 Жіночі тренування (з упором на сідниці)')
    b18 = types.KeyboardButton("👤 Чотириденний спліт (для чоловіків)")
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.row(b16, b17)
    markup.row(b18, back1)
    bot.send_message(message.chat.id, text='💪 Оберіть тренування, яке вам
потрібно👉'.format(message.from_user), reply_markup=markup)

```



```

elif message.text == "👉 Назад 🏠":
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b30 = types.KeyboardButton("🍔 Харчування 🍔")
    b31 = types.KeyboardButton("🏋️ Тренування 🏋️")
    markup.add(b30,b31)
    bot.send_message(message.chat.id, text="Ось план ваших тренувань та харчування".format(message.from_user), reply_markup=markup,parse_mode="html")

```

```

elif message.text == "🍔 Харчування":
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("🍌")
    b17 = types.KeyboardButton('🥑')
    b18 = types.KeyboardButton("🍷")
    b19 = types.KeyboardButton('🥑')
    back1 = types.KeyboardButton('🍔 Назад 🏠')
    markup.add(b16,b17,b18,b19,back1)
    bot.send_message(message.chat.id,text='Оберіть 👉\n🍌 - Сніданок\n\n🥑 - Обід\n\n🍷 - Полудень\n\n🥑 - Вечеря'.format(message.from_user), reply_markup=markup)

```

```

elif message.text == "🍌":
    bot.send_message(message.chat.id,text='🍌 <b>Сніданок</b>\n2 яйця цілих\n10г волоських горіхів\n70г вівсянки + 5г кориці + ківі',parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("🍌")
    b17 = types.KeyboardButton('🥑')
    b18 = types.KeyboardButton("🍷")
    b19 = types.KeyboardButton('🥑')
    back1 = types.KeyboardButton('🍔 Назад 🏠')
    markup.add(b16,b17,b18,b19,back1)
    bot.send_message(message.chat.id,text='Оберіть 👉\n🍌 - Сніданок\n\n🥑 - Обід\n\n🍷 - Полудень\n\n🥑 - Вечеря'.format(message.from_user), reply_markup=markup)

```

```

elif message.text == "🥑":
    bot.send_message(message.chat.id,text='🥑 <b>Обід</b>\n150г курячого філе\n60г гречки зелені овочі яблуко',parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("🍌")
    b17 = types.KeyboardButton('🥑')
    b18 = types.KeyboardButton("🍷")
    b19 = types.KeyboardButton('🥑')
    back1 = types.KeyboardButton('🍔 Назад 🏠')
    markup.add(b16,b17,b18,b19,back1)
    bot.send_message(message.chat.id,text='Оберіть 👉\n🍌 - Сніданок\n\n🥑 - Обід\n\n🍷 - Полудень\n\n🥑 - Вечеря'.format(message.from_user), reply_markup=markup)

```

```

elif message.text == "🍷":
    bot.send_message(message.chat.id,text='🍷 Полудень\n200г курячого філе\n70г гречки\nзелені овочі + 1 чайна ложка оливкової олії',parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("🍌")

```

```

b17 = types.KeyboardButton('🥑')
b18 = types.KeyboardButton("🍷")
b19 = types.KeyboardButton('🥑')
back1 = types.KeyboardButton('👉 Назад 🏠')
markup.add(b16,b17,b18,b19,back1)
bot.send_message(message.chat.id,text='Оберіть 👉\n🍷 - Сніданок\n\n🥑 - Обід\n\n🍷 - Полудень\n\n🥑 - Вечеря'.format(message.from_user), reply_markup=markup)

elif message.text == "🥑":
    bot.send_message(message.chat.id,text='🥑 Вечеря \n100г курячого філе\n50г авокадо або 15г волоських горіхів\nзелені овочі + 1 чайна ложка оливкової олії',parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("🍷")
    b17 = types.KeyboardButton('🥑')
    b18 = types.KeyboardButton("🍷")
    b19 = types.KeyboardButton('🥑')
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.add(b16,b17,b18,b19,back1)
    bot.send_message(message.chat.id,text='Оберіть 👉\n🍷 - Сніданок\n\n🥑 - Обід\n\n🍷 - Полудень\n\n🥑 - Вечеря'.format(message.from_user), reply_markup=markup)

elif message.text == "👉 Тренування":
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("Фулбаді для чоловіків👉")
    b17 = types.KeyboardButton('Жіночі тренування (з упором на сідниці)👉')
    b18 = types.KeyboardButton("Чотириденний спліт (для чоловіків)👉")
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.row(b16,b17)
    markup.row(b18,back1)
    bot.send_message(message.chat.id,text='👉 Оберіть тренування, яке вам потрібно👉'.format(message.from_user), reply_markup=markup)

elif message.text == "Фулбаді для чоловіків👉":
    markup = types.InlineKeyboardMarkup()
    button1 = types.InlineKeyboardButton("👉 Тренування👉",
url='https://telegra.ph/Fulbad%D1%96-dlya-cholov%D1%96k%D1%96v-06-20')
    markup.add(button1)
    bot.send_message(message.chat.id,text="👉 Фулбаді для чоловіків👉".format(message.from_user), reply_markup=markup)
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b16 = types.KeyboardButton("Фулбаді для чоловіків👉")
    b17 = types.KeyboardButton('Жіночі тренування (з упором на сідниці)👉')
    b18 = types.KeyboardButton("Чотириденний спліт (для чоловіків)👉")
    back1 = types.KeyboardButton('👉 Назад 🏠')
    markup.row(b16,b17)
    markup.row(b18,back1)
    bot.send_message(message.chat.id,text='👉 Оберіть тренування, яке вам потрібно👉'.format(message.from_user), reply_markup=markup)

elif message.text == "Жіночі тренування (з упором на сідниці)👉":
    markup = types.InlineKeyboardMarkup()

```

```

        button1 = types.InlineKeyboardButton("👤 Тренування💪",
url='https://telegra.ph/ZH%D1%96noch%D1%96-trenuvannya-z-uporom-na-
s%D1%96dnic%D1%96-06-20')
        markup.add(button1)
        bot.send_message(message.chat.id,text="👤 Жіночі тренування (з упором на
сідниці) 👉".format(message.from_user), reply_markup=markup)
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        b16 = types.KeyboardButton("Фулбаді для чоловіків😊")
        b17 = types.KeyboardButton('Жіночі тренування (з упором на сідниці)👤')
        b18 = types.KeyboardButton("Чотириденний спліт (для чоловіків)😊")
        back1 = types.KeyboardButton('👉 Назад 🏠')
        markup.row(b16,b17)
        markup.row(b18,back1)
        bot.send_message(message.chat.id,text='💪 Оберіть тренування, яке вам
потрібно👉'.format(message.from_user), reply_markup=markup)

    elif message.text == "Чотириденний спліт (для чоловіків)😊":
        markup = types.InlineKeyboardMarkup()
        button1 = types.InlineKeyboardButton("👤 Тренування💪",
url='https://telegra.ph/CHotiridennij-spl%D1%96t-dlya-cholov%D1%96k%D1%96v-06-20')
        markup.add(button1)
        bot.send_message(message.chat.id,text="😊 Чотириденний спліт (для чоловіків)
👉".format(message.from_user), reply_markup=markup)
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        b16 = types.KeyboardButton("Фулбаді для чоловіків😊")
        b17 = types.KeyboardButton('Жіночі тренування (з упором на сідниці)👤')
        b18 = types.KeyboardButton("Чотириденний спліт (для чоловіків)😊")
        back1 = types.KeyboardButton('👉 Назад 🏠')
        markup.row(b16,b17)
        markup.row(b18,back1)
        bot.send_message(message.chat.id,text='💪 Оберіть тренування, яке вам
потрібно👉'.format(message.from_user), reply_markup=markup)

    elif message.text == "👉 Назад 🏠":
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        b30 = types.KeyboardButton("👉 Харчування")
        b31 = types.KeyboardButton("💪 Тренування")
        markup.add(b30,b31)
        bot.send_message(message.chat.id, text="Ось план ваших тренувань та
харчування".format(message.from_user), reply_markup=markup,parse_mode="html")

def plus_one(message):
    if message.text == "👉 Далі ➡️":
        bot.reply_to(message,"<b>Скільки Ви приблизно важите у кілограмах?
👤:</b> ",parse_mode="html")
        bot.register_next_step_handler(message, plus_two)

def plus_two(message):
    second_number = message.text
    store_number(message.from_user.id, "second_number",second_number)
    bot.reply_to(message, "<b>Який у Вас зріст в сантиметрах? 📏:
</b> ",parse_mode="html")

```

```

bot.register_next_step_handler(message, plus_three)

def plus_three(message):
    third_number = message.text

    if not third_number.isdigit():
        msg = bot.reply_to(message, '⚠ <b>Вводити можна тільки цілі
цифри!</b>', parse_mode="html")
        bot.register_next_step_handler(message, plus_three)
        return

    store_number(message.from_user.id, "third_number", third_number)
    bot.reply_to(message, "<b>Скільки вам років?</b>", parse_mode="html")
    bot.register_next_step_handler(message, plus_four)

def plus_four(message):
    fourth_number = message.text

    if not fourth_number.isdigit():
        msg = bot.reply_to(message, '⚠ <b>Вводити можна тільки цілі
цифри!</b>', parse_mode="html")
        bot.register_next_step_handler(message, plus_four)
        return

    store_number(message.from_user.id, "fourth_number", fourth_number)
    bot.reply_to(message, "<b>Введіть свій рівень фізичної активності(через
крапку)</b>👉\n♦ без занять спортом – 1.2\n♦ виконання вправ 1-3 рази / тиждень –
1.375\n♦ проведення тренувань 4-5 разів / тиждень – 1.55\n♦ заняття спортом не менше 6
разів / тиждень – 1.725\n♦ професійні тренування – 1.9 ", parse_mode="html")
    bot.register_next_step_handler(message, plus_five)

def plus_five(message):
    fiveth_number = message.text

    if not ( "." in fiveth_number) or ( ',' in fiveth_number):
        msg = bot.reply_to(message, '⚠ <b>Вводити можна тільки числа з десятковим
дробом!(через крапку)</b>', parse_mode="html")
        bot.register_next_step_handler(message, plus_five)
        return

    store_number(message.from_user.id, "fiveth_number", fiveth_number)

    number_2 = get_number(message.from_user.id, "second_number")
    number_3 = get_number(message.from_user.id, "third_number")
    number_4 = get_number(message.from_user.id, "fourth_number")
    number_5 = get_number(message.from_user.id, "fiveth_number")

    result_plus = 9.247*float(number_2)

```

```

result_plus_2 = float(result_plus)+447.593
result_plus_3 = 3.098*float(number_3)
result_plus_4 = 4.33*float(number_4)
result_plus_5 = float(result_plus_2) + float(result_plus_3) - float(result_plus_4)
result_plus_6 = float(result_plus_5)*float(number_5)
result_plus_7= 15*float(result_plus_6)/100
result_plus_8=result_plus_6-result_plus_7
result_plus_9=result_plus_6+result_plus_7

```

```

bot.send_message(message.chat.id,text=f"<b>{result_plus_6}</b> – ваша денна
норма калорій.".format(message.from_user,parse_mode="html")
bot.send_message(message.chat.id,text=f"Щоб набрати вагу, ваш результат
повинен бути ≈ <b>{result_plus_9}</b>".format(message.from_user,parse_mode="html")
markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
b30 = types.KeyboardButton("🍌 Харчування 🗓️")
b31 = types.KeyboardButton("🏋️ Тренування 🗓️")
markup.add(b30,b31)
bot.send_message(message.chat.id, text="<b>Ось план ваших тренувань та
харчування</b>👉".format(message.from_user), reply_markup=markup,parse_mode="html")

```

```

def plus_six(message):
    if message.text == "🌳 Далі ➡️":
        bot.reply_to(message,"<b>Скільки Ви приблизно важите у кілограмах?
🏋️:</b> ",parse_mode="html")
        bot.register_next_step_handler(message, plus_seven)

```

```

def plus_seven(message):
    second_number = message.text
    store_number(message.from_user.id, "second_number",second_number)
    bot.reply_to(message, "<b>Який у Вас зріст в сантиметрах? 📏:
</b>",parse_mode="html")
    bot.register_next_step_handler(message, plus_eight)

```

```

def plus_eight(message):
    third_number = message.text

    if not third_number.isdigit():
        msg = bot.reply_to(message, ‘ ⚠️ <b>Вводити можна тільки цілі
цифри!</b>’,parse_mode="html")
        bot.register_next_step_handler(message, plus_seven)
        return

```

```

store_number(message.from_user.id, "third_number",third_number)
bot.reply_to(message, "<b>Скільки вам років?</b>",parse_mode="html")
bot.register_next_step_handler(message, plus_nine)

```

```

def plus_nine(message):
    fourth_number = message.text

    if not fourth_number.isdigit():
        msg = bot.reply_to(message, ‘ ⚠️ <b>Вводити можна тільки цілі
цифри!</b>’,parse_mode="html")

```

```

bot.register_next_step_handler(message, plus_eight)
return

store_number(message.from_user.id, "fourth_number",fourth_number)
bot.reply_to(message, "<b>Введіть свій рівень фізичної активності(через
крапку)</b>👉\n◆ без занять спортом – 1.2\n◆ виконання вправ 1-3 рази / тиждень –
1.375\n◆ проведення тренувань 4-5 разів / тиждень – 1.55\n◆ заняття спортом не менше 6
разів / тиждень – 1.725\n◆ професійні тренування – 1.9 ",parse_mode="html")
bot.register_next_step_handler(message, plus_ten)

def plus_ten(message):
    fiveth_number = message.text

    if not ( "." in fiveth_number) or ( ',' in fiveth_number):
        msg = bot.reply_to(message, '⚠️ <b>Вводити можна тільки числа з десятковим
дробом!(через крапку)</b>',parse_mode="html")
        bot.register_next_step_handler(message, plus_nine)
        return

    store_number(message.from_user.id, "fiveth_number",fiveth_number)

    number_2 = get_number(message.from_user.id, "second_number")
    number_3 = get_number(message.from_user.id, "third_number")
    number_4 = get_number(message.from_user.id, "fourth_number")
    number_5 = get_number(message.from_user.id, "fiveth_number")

    result_plus = 9.247*float(number_2)
    result_plus_2 = float(result_plus)+447.593
    result_plus_3 = 3.098*float(number_3)
    result_plus_4 = 4.33*float(number_4)
    result_plus_5 = float(result_plus_2) + float(result_plus_3) - float(result_plus_4)
    result_plus_6 = float(result_plus_5)*float(number_5)
    result_plus_7= 15*float(result_plus_6)/100
    result_plus_8=result_plus_6-result_plus_7
    result_plus_9=result_plus_6+result_plus_7

    bot.send_message(message.chat.id,text=f"<b>{result_plus_6}</b> – ваша денна
норма калорій.".format(message.from_user),parse_mode="html")
    bot.send_message(message.chat.id,text=f"Щоб схуднути, ваш результат повинен
бути ≈ <b>{result_plus_8}</b>".format(message.from_user),parse_mode="html")
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    b30 = types.KeyboardButton("🍔 Харчування")
    b31 = types.KeyboardButton("🏋️ Тренування")
    markup.add(b30,b31)
    bot.send_message(message.chat.id, text="<b>Ось план ваших тренувань та
харчування</b>👉".format(message.from_user), reply_markup=markup,parse_mode="html")

```

```

def main():
    while True:
        try:
            # Запускаємо бота
            bot.polling(none_stop=True)
        except Exception as e:
            user_id = '540517419' # Встановіть ідентифікатор користувача для
повідомлень про помилки
            error_message = f"Помилка при запуску бота: {e}\nПерезапуск бота через 5
секунд..."
            bot.send_message(user_id, error_message)
            time.sleep(5)
if __name__ == "__main__":
    try:
        main()
    except Exception as e:
        user_id = '540517419' # Встановіть ідентифікатор користувача для повідомлень
про помилки
        error_message = f"Помилка при запуску бота: {e}\nПерезапуск бота через 5
секунд..."
        bot.send_message(user_id, error_message)
        time.sleep(5)

```

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, _____

студент __4__ курсу, __денної__ форми навчання, математичного факультету, спеціальності __122 комп'ютерні науки_____,
адреса електронної пошти _____, –

підтверджую, що написана мною кваліфікаційна робота бакалавра на тему «__Подання телеграм бота для досягнення цілей у схудненні та наборі ваги_____» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

– заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

– згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Студент

05.06.2024

(дата)

Пономаренко О.І.

(прізвище, ініціали)

Науковий керівник

(дата)

(підпис)

(прізвище, ініціали)