|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»    [*Факультет социально-экономических и компьютерных наук*](https://perm.hse.ru/scs/) |
| Молчанов Захар Валерьевич  **РАЗРАБОТКА САЙТА ДЛЯ СОСТАВЛЕНИЯ ПРОГРАММЫ ТРЕНИРОВОК И ПЛАНА ПИТАНИЯ**  *Курсовая работа*  студента образовательной программы «Программная инженерия» по направлению подготовки *09.03.04 Программная инженерия*  Руководитель  преподаватель кафедры информационных технологий в бизнесе  М.В. Иванова |

Пермь, 2023

**Аннотация**

Курсовая работа студента второго курса ПИ-21-2 Молчанова Захара Валерьевича по теме «Разработка сайта для составления программы тренировок и плана питания».

Данная работа описывает процесс разработки сайта по подбору и составлению программы тренировок и плана питания, а также подсчет базового обмена веществ человека. Работа включает: анализ сайтов аналогов в сфере здоровья и постановку требований к системе; описание процесса проектирования системы, в т. ч. диаграммы прецедентов активности в нотации UML, ER-диаграмму модели данных, а также запросы к модели данных, реализуемой в СУБД MS SQL; процесс разработки сайта на языке Python c помощью фреймворка Flask для web-приложений и HTML, CSS и JavaScript ­– для пользовательского интерфейса.

Работа содержит 38 страницы основного текста, включает в себя 5 таблиц и 4 приложения.

Оглавление

[Введение 4](#_Toc130914616)

[Глава 1 Анализ предметной области и постановка требований к системе 6](#_Toc130914617)

[1.1 Введение в предметную область 6](#_Toc130914618)

[1.2 Анализ систем-аналогов 7](#_Toc130914619)

[1.3 Постановка требований к системе 11](#_Toc130914620)

[1.4 Выводы к главе 15](#_Toc130914621)

[Глава 2 Проектирование приложения Healthy People 16](#_Toc130914622)

[2.1 Моделирование процессов подбора тренировок и плана питания 16](#_Toc130914623)

[2.2 Выбор архитектуры приложения, средств и технологий разработки 18](#_Toc130914624)

[2.3 Проектирование пользовательского интерфейса 25](#_Toc130914625)

[2.4 Проектирование базы данных 35](#_Toc130914626)

[2.5 Выводы к главе 37](#_Toc130914627)

[Глава 3 Разработка приложения Healthy People 39](#_Toc130914628)

[3.1Разработка основного функционала сайта 39](#_Toc130914629)

[3.2Тестирование сайта 44](#_Toc130914630)

[3.3 Выводы к главе 45](#_Toc130914631)

[Заключение 46](#_Toc130914632)

[Библиографический список 47](#_Toc130914633)

[Приложение А Описание данных 49](#_Toc130914634)

[Приложение Б Описание API 54](#_Toc130914635)

[Приложение В Сценарии тестирования 57](#_Toc130914636)

[Приложение Г Техническое задание 59](#_Toc130914637)

[Приложение Д Руководство оператора 5](#_Toc130914638)

# Введение

В современном мире все больше людей стремятся вести здоровый образ жизни и следить за своим физическим состоянием. Для этого необходимо правильно питаться и регулярно тренироваться. У новичка при составлении программы тренировок может возникнуть большое количество сложностей, например: какие упражнения лучше всего сочетаются; какую методологию тренировок выбрать Split или FullBody; как правильно тренироваться. Также зачастую возникают трудности с подбором рациона питания, ведь если человек не знает свой базовый обмен веществ, то он не сможет контролировать свой вес.

Подбор эффективной программы тренировок и индивидуального рациона питания трудоемкий процесс. Чтобы выбрать правильную программу тренировок или рацион питания, нужно обратиться к специалисту, но можно начать заниматься по безопасной и универсальной для человека программе тренировкой с сбалансированным планом питания, чем ждать и думать о том, что нужно сходить к диетологу и тренеру.

На данный момент существует не так много сайтов для подбора и составления программ тренировок и плана питания, a также зачастую эти функции не объединены на сайтах.

Таким образом, целью курсовой работы является разработка сайта по подбору программы тренировок и плана питания с использованием базы данных.

Для достижения цели необходимо решить следующие задачи:

1. Проанализировать существующие решения и сформулировать требования к системе.
2. Составить техническое задание.
3. Описать и детализировать модель данных.
4. Выполнить проектирование информационной системы с помощью диаграмм нотации UML.
5. Разработать запросы к базе, позволяющие выполнять добавление/ изменение/удаление данных и просмотр выборки данных на основе указанных пользователей параметров.
6. Выбрать средства и инструменты реализации сайта.
7. Провести пробное тестирование и отладку сайта, при необходимости внести корректировки и исправить выявленные ошибки.

Объектом данного исследования являются сервисы для персонализированного подбора программы тренировок и/или плана питания, а предметом исследования – разработка wеб-приложение для подбора оптимальной программы тренировок и плана питания.

В процессе используются такие методы исследования, как анализ, синтез, дедукция и индукция, формализация, объектно-ориентированное моделирование, эксперимент (создание веб-приложения).

# Анализ предметной области и постановка требований к системе

Эта глава посвящена анализу предметной области составления программ тренировок и плана питания, который включает в себя:

1. введение в предметную область ЗОЖ;
2. обзор существующих на рынке решений для формирования видения концепции разрабатываемой информационной системы;
3. постановка требований, предъявляемых к реализуемой информационной системе.

## Введение в предметную область

План тренировок – это структурированный набор упражнений, которые разибты на группы в зависимости от дня, когда они должны быть выполнены в рамках тренировочной программы.

План питания представляет собой составленный перечень продуктов, содержание калорий, белков, жиров и углеводов, в которых рассчитывается в соответствии с базовым обменом веществ.

Новичок — это человек, который только начинает заниматься фитнесом или спортом и не имеет опыта в тренировках или имеет очень малый опыт. Он может быть любого возраста, пола. Основные характеристики новичка: низкий уровень физической подготовки, ограниченный опыт в тренировках.

Выбор программы тренировок и плана питания – важный шаг на пути к достижению желаемых результатов. При выборе программы тренировок необходимо учитывать свои цели, уровень физической подготовки, а также индивидуальные особенности. При выборе плана питания необходимо учитывать свои потребности в питательных веществах и калориях, основанных на базовом обмене веществ.

Сайт, реализующий подбор программы тренировок, расчет базового обмена веществ и подбор рациона питания по метаболизму на рынке не был найден. Есть сайты, с помощью которых можно или программу тренировок, или план питания с подсчетом базового обмена веществ.

## Анализ систем-аналогов

Анализ систем аналогов позволяет выявить преимущества и недостатки аналогов, для формализации требований к разрабатываемому приложению. В рамках данного исследования проводится сравнительный анализ различных аспектов сайтов, включая функциональность и контент

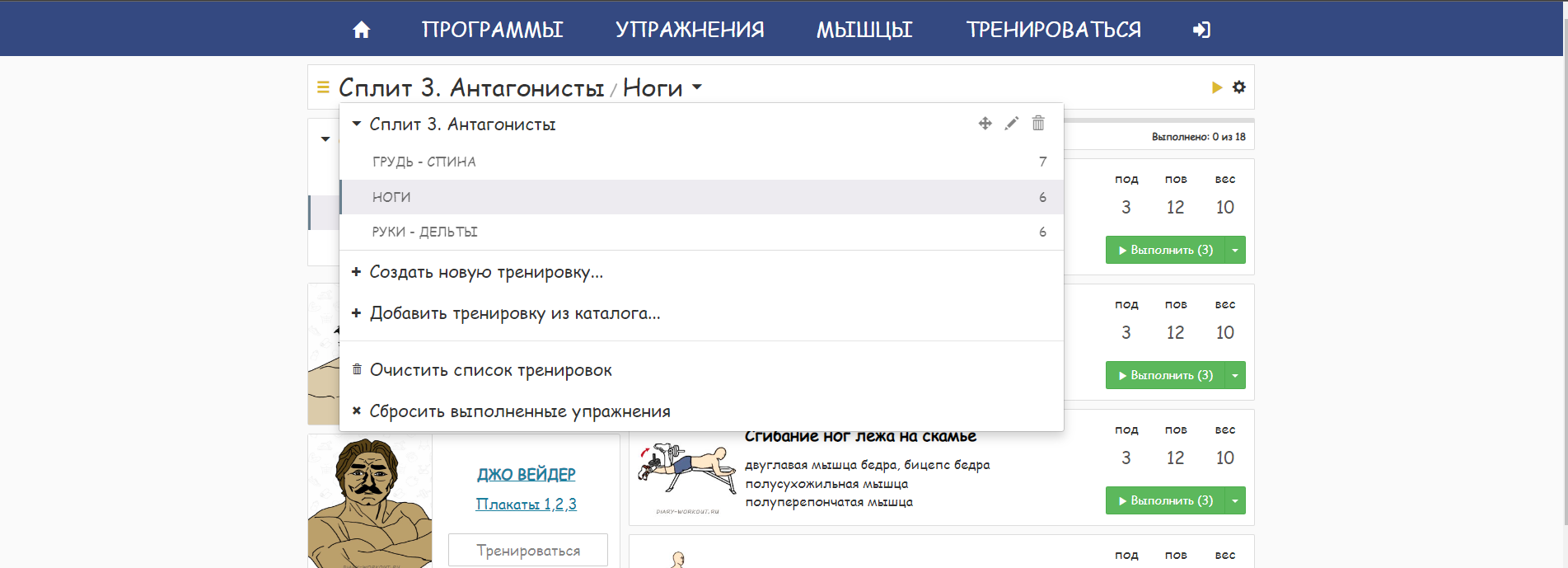
### 1.2.1 Сайт Diary-workout.ru

Первый сайт это [Дневник тренировок (diary-workout.ru)](https://diary-workout.ru/?ysclid=l8j63o15mo477124697) (см. Рисунок 1). В нём можно выбрать программу тренировок из предложенного списка, отсортированного по собственным предпочтениям. Сайт является бесплатным, а также сохраняется выбранная программа тренировок. Из минусов, с моей точки зрения это перегруженность интерфейса, ведь пользователь посещает сайт, с целью подобрать программу тренировок, а сложность интерфейса может его отпугнуть. Эту перегруженность (см. Рисунок 2). Так же, при недолгом использовании сайта, были замечены неточности. Например, при выборе программы для домашних условий, предлагается программа тренировок с использование специального инвентаря, который имеется не у каждого дома. Можно сделать вывод о том, что лучше упростить подбор программы тренировок для пользователя.

Изображение выглядит как текст

Автоматически созданное описание

**Рисунок 1 – Сайт дневника тренировок (каталог программ)**

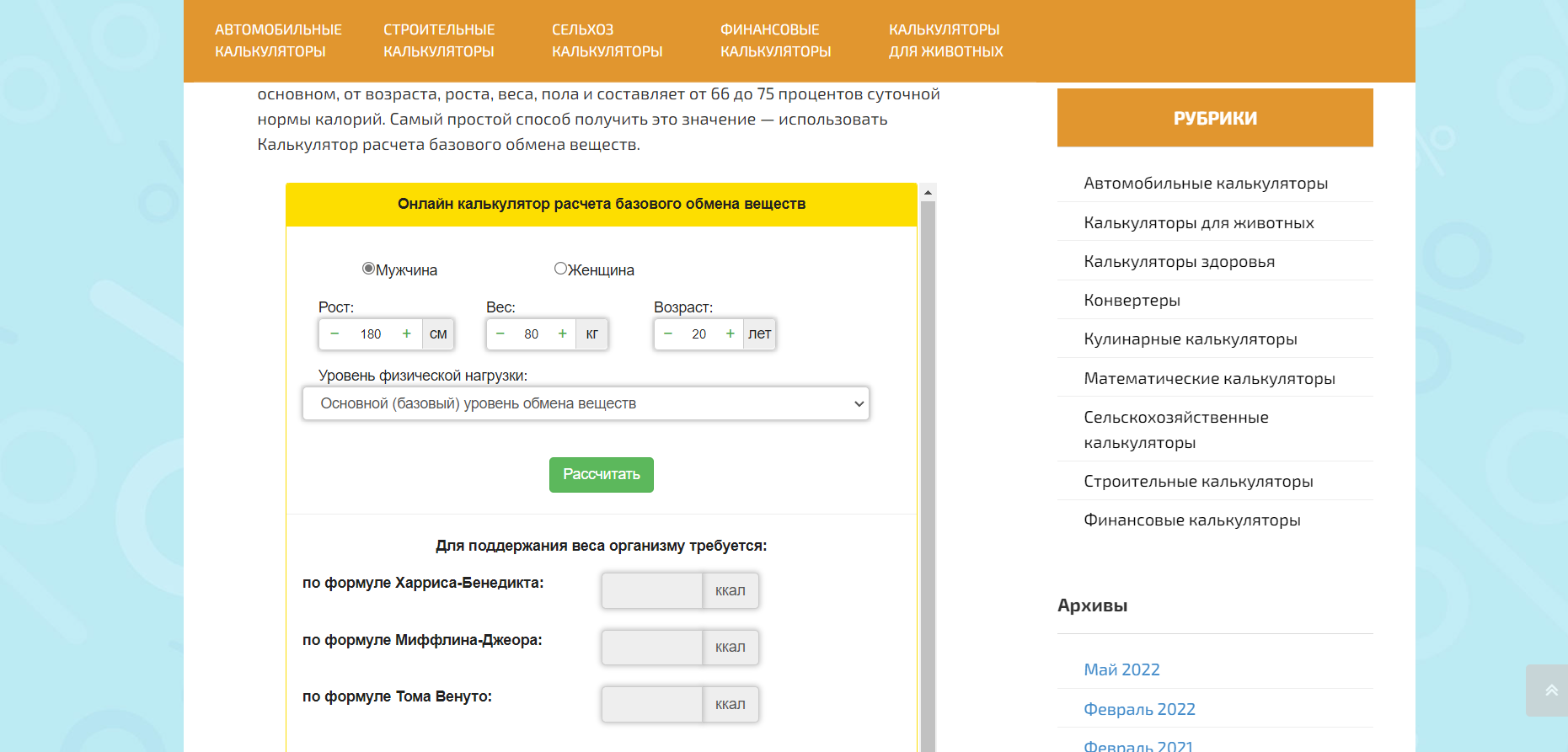


**Рисунок 2 – Каталог упражнений на сайте дневника тренировок**

### 1.2.2 Сайт калькулятор расчета базового обмена веществ

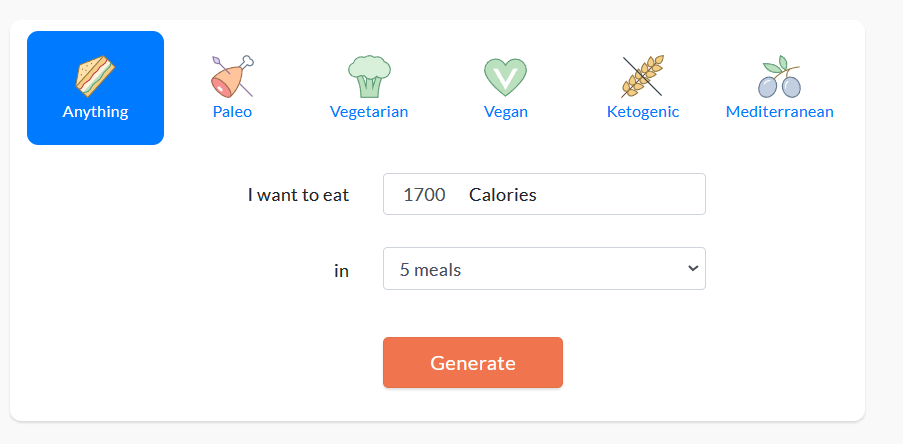
Сайт, который имеет сразу 2 функции расчёта базового обмена веществ и составления плана питания, найти не удалось. Существуют калькуляторы расчёта базового метаболизма организма человека (см. рисунок 3). Основной проблемой сайта является то, что на этом сайте реализованы и другие калькуляторы, не относящиеся к спортивной сфере. Интерфейс является не перегруженным, пользователь может сразу (без дополнительной инструкции) воспользоваться для подсчёта базового обмена веществ.

Однако, для составления плана питания пользователю придется искать другие ресурсы или использовать другие калькуляторы. Это может быть неудобно для пользователей, которые ищут интегрированный подход для оценки своего обмена веществ и планирования рациона питания.



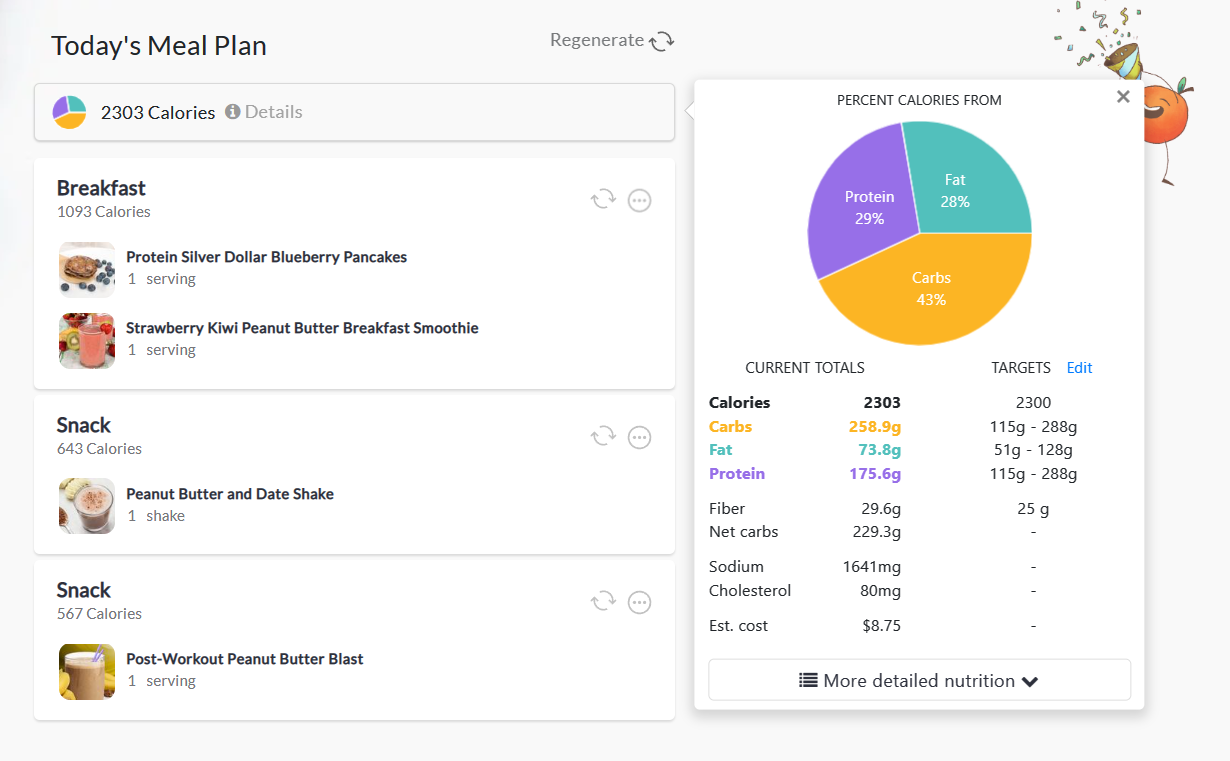
**Рисунок 3 – Калькулятор обмена веществ**

### 1.2.3 Сайт подбора рациона питания eatthismuch.com

Eatthismuch – это один из самый популярных англоязычных сайтов, который помогает подобрать план питания. Сайт достаточно удобен в использовании, но не обладает функцией автоматического расчета базового обмена веществ, которую нужно ввести вручную для подбора рациона питания (см. рисунок 4). 

**Рисунок 4 – Сайт eatthismuch.com**

Одно из преимуществ сайта – это график, который показывает количество калорий, белков, жиров и углеводов в составе рациона питания (см. Рисунок 5). Однако следует учитывать, что некоторые продукты, которые предлагает использовать сайт, могут быть опасны для здоровья.



**Рисунок 5 – График на сайте eatthismuch.com**

## Постановка требований к системе

Для выявления требований к разрабатываемому приложению, была составлена (см. Таблица 1), представляющая анализ представленных аналогов по выбранным критериям.

**Таблица 1 – Сводка анализ систем-аналогов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Сайты | | | Решения |
| Diary-workaut.ru | Калькулятор расчета БОВ | Eatthismuch.com | Принятие данной функции |
| Функционал | Русский язык | + | + | - | Да |
| Расчет БОВ | - | + | - | Да |
| Подбор программы тренировок | + | - | - | Да |
| Подбор рациона питания | - | - | + | Да |
| Вывод инфографики о диете | - | - | + | Да |
| Разбиение подбора диет на категории | - | - | + | Нет |
| Экспорт и импорт данных в xlsx формате | - | - | - | Да |

Таблица решений помогла выявить целевые разделы и функции, которые необходимо представить на разрабатываемом сайте по подбору программы тренировок и плана питания, а именно: расчет базового обмена веществ, подбор программы тренировок по предпочтениям, подбор рациона питания, вывод инфографики о диете и экспорт данных в xlsx формате.

В результате построения таблицы решений было опущено потенциальное требование по разбиению типов диет на категории. т. к., если у человека есть проблемы с питанием или аллергия на продукты, то человеку необходимо обращаться к специалисту.

В результате анализа аналогов, можно определить следующие функциональные требования к приложению:

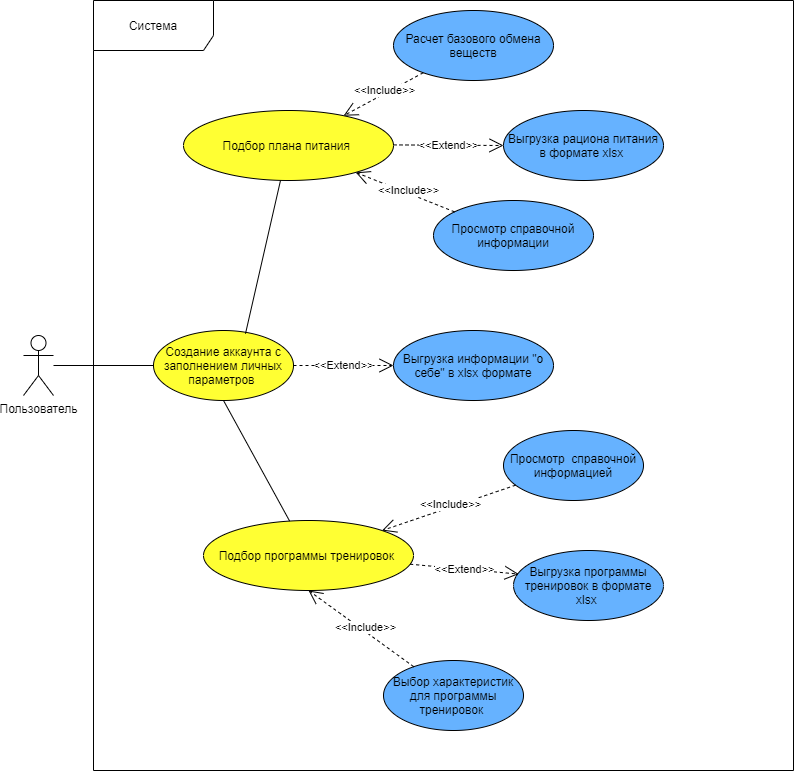
Функциональные требования:

1. Возможность регистрации нового пользователя на сайте.
2. Возможность авторизации зарегистрированных пользователей.
3. Подбор программы тренировок по выбранным характеристикам.
4. Скачивание программы тренировок, плана питания и всей информации о пользователе в xlsx формате.
5. Подбор плана питания по базовому обмену веществ.
6. Вычисление базового обмена веществ.

Нефункциональные требования:

1. Сайт должен быть полностью на русском языке.
2. Сайт должен быть удобным и интуитивно понятным для пользователей.

На основе полученных требований была построена диаграмма прецедентов (см. рисунок 6).



**Рисунок 6 – Диаграмма прецедентов**

Описание прецедентов:

Прецедент: Создание аккаунта с заполнением личных параметров

Описание: данный прецедент позволяет пользователям создать аккаунт на сайте, заполнив свои личные параметры. Пользователь должен ввести свое имя, фамилию, адрес электронной почты, пароль. Эти данные могут использоваться для авторизации на сайте и доступа к персональным настройкам.

Прецедент: Выгрузка информации "о себе" в xlsx формате

Описание: данный прецедент расширяет функционал создания аккаунта, позволяя пользователям выгрузить информацию о себе в формате xlsx. Это может быть полезно для пользователей, которые хотят сохранить свою информацию на своем компьютере.

Прецедент: Подбор программы тренировок на сайте

Описание: данный прецедент позволяет пользователям выбрать характеристики, в соответствии с которыми будет сформирована программа тренировок. Пользователь может выбрать различные параметры, такие методология тренировки, количество дней в программе, место тренировок. На основе выбранных пользователем характеристик, система формирует программу тренировок. Программа может содержать различные упражнения. Пользователь может просмотреть программу, сохранить ее на своем профиле. Этот прецедент включает в себя подпрецеденты "Просмотр справочной информации" и "Выбор характеристик для программы тренировок", а также расширяется прецедентом "Выгрузка программы тренировок в формате xlsx".

Прецедент: Просмотр справочной информации

Описание: В рамках этого подпрецедента пользователь может ознакомиться со справочной информацией, содержащейся на сайте о тренировках. Возможна навигация по разным разделам сайта с целью получения информации о правильном выборе программы тренировок.

Прецедент: Выбор характеристик для программы тренировок

Описание: в рамках этого подпрецедента пользователи выбирает характеристики для программы тренировок. Пользователь может выбрать различные параметры, такие методология тренировки, количество дней в программе, место тренировок. Эти параметры позволяют пользователям настроить программу под свои потребности и физическую форму.

Прецедент: Выгрузка программы тренировок в формате xlsx

Описание: данный прецедент расширяет функционал подбора программы тренировок, позволяя пользователям выгрузить программу в формате xlsx.

Прецедент: Подбор плана питания на сайте

Описание: данный прецедент позволяет пользователям подобрать план питания в соответствии с его базовым обменов веществ. Данный план питания включает в себя продукты. Пользователь может просмотреть план, сохранить его на своем профиле. Этот прецедент включает в себя подпрецеденты "Просмотр справочной информации" и "Расчет базового обмена веществ", а также расширяется прецедентом "Выгрузка плана питания в xlsx формате".

Прецедент: Расчет базового обмена веществ

Описание: в рамках этого подпрецедента система проводит расчет базового обмена веществ пользователя на основе его личных параметров (рост, вес, возраст и т.д.). Эта информация необходима для того, чтобы определить оптимальное количество калорий в день для достижения желаемых целей. Пользователь может использовать результаты расчета для выбора соответствующего плана питания.

Прецедент: Выгрузка плана питания в xlsx формате

Описание: данный прецедент расширяет функционал подбора плана питания, позволяя пользователям выгрузить персональный план питания в формате xlsx. Это может быть полезно для пользователей, которые хотят сохранить свою информацию на своем компьютере.

## Выводы к главе

В ходе анализа предметной области и постановки требований к веб-приложению были сделаны значительные усилия для выявления функциональных и нефункциональных требований. Название сайта было выбрано “Healthy People”. В процессе анализа было разработано техническое задания (см. ПРИЛОЖЕНИЕ Г) для приложения, которое определения определяет функциональные и нефункциональные требования к приложению, а также устанавливает основные принципы разработки.

# Глава 2 Проектирование приложения Healthy People

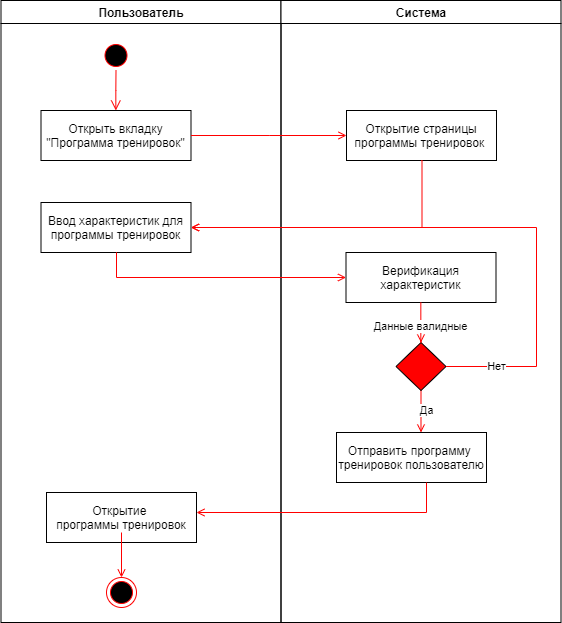
В данной главе рассматриваются вопросы, связанные с моделированием поведения системы, выбором инструментов и технологий для ее разработки, определением модели базы данных и пользовательского интерфейса.

## 2.1 Моделирование процессов подбора тренировок и плана питания

В этом разделе представлены диаграммы активности для 2 прецедентов: подбор плана питания и подбор программы тренировок. Диаграмма активности необходима для понимания последовательности действий, которые должны выполнить пользователь и система для достижения определенной цели.

### 2.1.1 Процесс подбора плана питания

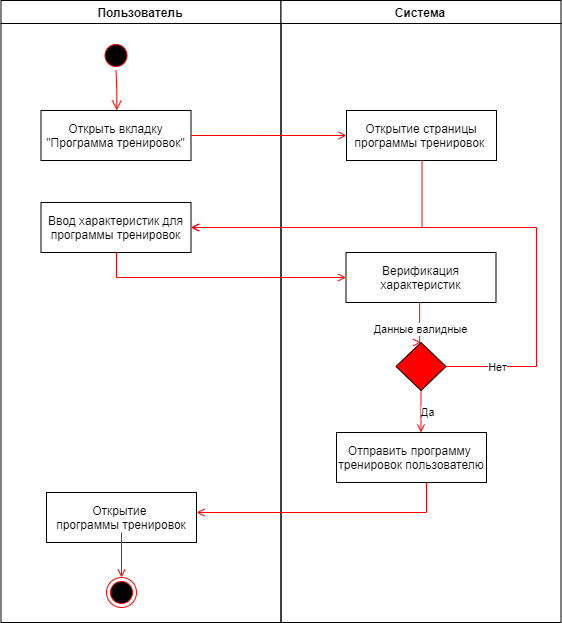
На рисунке 7 представлена диаграмма активности для прецедента «Подбор плана питания».



**Рисунок 7 – Диаграмма активности для прецедента "Подбор плана питания"**

### 2.1.2 Процесс подбор программы тренировок

На рисунке 8 представлена диаграмма активности для прецедента “Подбор программы тренировок”.



**Рисунок 8 – Диаграмма активности для прецедента "Подбор программы тренировок"**

## 2.2 Выбор архитектуры приложения, средств и технологий разработки

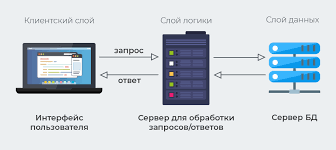
Клиент серверная архитектура является одной из наиболее распространенных архитектур в разработке программного обеспечения, Эта модель включает в себя две основные компоненты: клиент и сервер.

Клиент представляет собой программу или устройств о, которое запрашивает услуги у сервера. Клиенты могут быть как человеком, использующим интерфейс пользователя, так и другой программой, которая запрашивает данные или услуги у сервера. Клиент может быть расположен на разных устройствах и иметь различные операционные системы.

Сервер, в свою очередь, является центральной компьютерной системой, которая обрабатывает запросы клиентов и предоставляет им необходимые услуги и данные. Сервер может выполнять множество функций, от обработки запросов клиентов до хранения и управления данными.

Клиент-серверная архитектура предоставляет ряд преимуществ, таких как распределение нагрузки между сервером и клиентами, возможность централизованного управления и контроля, повышенную защиту данных и возможность удаленного доступа к информации.

Пример клиент-серверной архитектуры (см. Рисунок 9)



**Рисунок 9 – Клиент-серверная архитектура**

Как происходит общение между клиентом и сервером (REST API)? Сервером и БД (драйвер)?

Общение между сервером и клиентом. Когда клиент отправляет запрос на сервер, он обычно использует протокол HTTP (Hypertext Transfer Protocol) для связи. Запрос HTTP состоит из нескольких элементов, включая метод запроса (GET, POST, PUT, DELETE), URL-адрес сервера, к которому нужно обратиться, и заголовки, содержащие дополнительную информацию о запросе. Сервер, в свою очередь, обрабатывает запрос и отправляет обратно клиенту HTTP-ответ, который содержит статус-код (например, 200 OK или 404 Not Found), заголовки и тело ответа. Тело ответа может быть любым текстом или структурированным форматом, таким как XML или JSON.

Для общения между сервером и БД используется драйвер БД — это программное обеспечение, которое позволяет установить соединение между сервером и конкретной БД. Драйверы обычно разработаны специально для определенной БД и взаимодействуют с ней по определенному протоколу, например, через SQL или ODBC. Для того чтобы использовать драйвер, необходимо установить его на сервере или на компьютере, где запущен клиентское приложение. Для того чтобы установить соединение, драйвер передает данные об учетной записи, адресе сервера, порте и других параметрах, необходимых для подключения к БД. После того как соединение установлено, сервер может отправлять запросы на языке SQL, используя драйвер, и получать ответы от БД.

Для разработки клиент-серверного приложения необходимо определить язык программирования и фреймворк для написания бизнес-логики сервера, средства реализации клиента, а также СУБД.

### 2.2.1 Выбор СУБД

Для разрабатываемого веб-приложения требуется база данных, в которой будет храниться информация о зарегистрированном пользователе, о его программе тренировок, плане питания и личных данных. Для базы данных необходимо определить СУБД, в которой будут созданы необходимые таблицы, и они будут связаны между собой.

#### **2.2.1.1 MS SQL**

Microsoft SQL Server (MS SQL) [см. https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16] – это одна из наиболее распространенных реляционных СУБД используемых для хранения и управления данными в корпоративной среде. Рассмотрим некоторые его плюсы и минусы, а также объясним почему его лучше не выбирать для моего веб-приложения.

Плюсы MS SQL:

1. MS SQL Server очень надежен и обеспечивает высокую степень защиты данных. Это СУБД имеет встроенные функции безопасности, такие как аутентификация Windows и шифрование данных, что делает его идеальным для бизнес-приложений.
2. Следующим плюсом является масштабируемость, MS SQL Server может масштабироваться для удовлетворения постоянно меняющихся потребностей корпоративных предприятий, что позволяет ему обрабатывать огромные объемы данных и работать с большим количеством пользователей.

Недостатки MS SQL:

1. Цена: MS SQL Server является коммерческим продуктом, поэтому его стоимость может быть высокой для небольших организаций или проектов.
2. Высокие требования к системным ресурсам: MS SQL Server требует высокой производительности системных ресурсов, что может привести к проблемам с производительностью и скоростью работы в случае недостаточного оборудования.

#### **2.2.1.2 MySQL**

MySQL — это одна из наиболее распространенных реляционных СУБД с открытым исходным кодом. Рассмотрим кратко некоторые его плюсы и минусы.

Плюсы MySQL. MySQL доступен бесплатно и имеет открытый исходный код, что делает его доступным для широкого круга пользователей. Также можно отметить высокую производительность.

### 2.3.2 Выбор языка и фреймворка для написания Backend

Для разработки веб-приложения необходимо определиться с выбором языка программирования для Backend – части приложения. Backend – часть отвечает за обработку запросов от клиента и работу с базой данных. Для реализации backend используются различные языки программирования, рассмотрим C# с фреймворком ASP.NET и Python с фреймворком Flask.

#### **2.3.2.1 ASP.NET и C#**

ASP.NET [см. https://docs.microsoft.com/en-us/aspnet/] — это фреймворк для создания веб-приложений, который может быть использован для разработки как небольших, так и больших проектов. Рассмотрим некоторые плюсы и минусы использования ASP.NET для создания небольшого веб-приложения.

Плюсы:

1. Высокая производительность: ASP.NET обеспечивает высокую производительность веб-приложений благодаря своей архитектуре и интеграции с IIS.
2. Удобство разработки: Веб-приложения, созданные на ASP.NET, могут быть быстро и легко разработаны благодаря большому количеству инструментов и библиотек, доступных разработчикам широкие возможности: ASP.NET предоставляет широкий спектр инструментов и возможностей для создания различных типов веб-приложений, таких как веб-сайты, веб-службы, приложения для обработки данных и многое другое.
3. Безопасность: ASP.NET имеет встроенные механизмы безопасности, такие как проверка подлинности и авторизация, которые могут быть использованы для защиты веб-приложений от атак и несанкционированного доступа к данным. Масштабируемость: ASP.NET позволяет создавать масштабируемые веб-приложения, которые могут обслуживать множество пользователей одновременно без снижения производительности.

Минусы:

1. Сложность: В отличие от некоторых других фреймворков, создание веб-приложений на ASP.NET может быть сложным и требовать определенного уровня знаний и опыта.
2. Зависимость от Windows: ASP.NET является фреймворком для разработки веб-приложений, которые работают на платформе Windows. Это означает, что разработка веб-приложений на ASP.NET может ограничиваться использованием только Windows-серверов.

#### **2.3.2.2 Flask и Python**

FLASK [см. https://flask.palletsprojects.com/en/2.1.x/] – это микрофреймворк для создания веб-приложений на языке программирования Python [см. https://www.python.org/doc/]. Рассмотрим плюсы и минусы использования FLASK и Python для создания веб-приложений.

Плюсы:

1. Простота использования: FLASK предоставляет минимальный набор инструментов и библиотек, что делает его очень простым в использовании и позволяет быстро создавать веб-приложения на Python.
2. Гибкость: FLASK имеет очень гибкую архитектуру, что позволяет разработчикам легко настраивать и расширять функциональность веб-приложения в зависимости от потребностей проекта.
3. Обширная документация: Python и FLASK имеют обширную и подробную документацию, которая облегчает процесс изучения и использования фреймворка.

Минус – производительность. FLASK и Python в целом могут иметь небольшие проблемы с производительностью при работе с большими объемами данных или приложений с высокой нагрузкой. В таких случаях могут потребоваться дополнительные усилия по оптимизации и настройке приложения для повышения производительности.

### 2.3.3 Технологии реализации клиента

HTML, CSS и JavaScript являются основными технологиями, используемыми при разработке клиентской части веб-приложений. HTML [см. <https://developer.mozilla.org/en-US/docs/Web/HTML>], также известный как язык разметки гипертекста, используется для создания структуры веб-страницы. Он позволяет определять различные элементы, такие как заголовки, параграфы, списки, изображения и ссылки, с помощью тегов, которые указывают браузеру, как отображать содержимое.

CSS [см. https://developer.mozilla.org/en-US/docs/Web/CSS], или каскадные таблицы стилей, является языком, который используется для оформления веб-страницы. Он определяет, как элементы HTML должны отображаться на странице, включая цвет, размер, шрифт и расположение. CSS обеспечивает возможность создавать различные эффекты, такие как анимация и переходы.

JavaScript [см. https://developer.mozilla.org/en-US/docs/Web/JavaScript] — это язык программирования, который используется для создания интерактивных элементов на веб-странице. Он позволяет создавать динамические эффекты, обрабатывать события, отправлять запросы на сервер и изменять содержимое страницы без необходимости перезагрузки. JavaScript используется для создания сложных приложений, таких как игры и интерактивные формы.

### 2.3.4 Вывод к разделу

В данном разделе была рассмотрена проблема выбора архитектуры приложения, средств технологий разработки и СУБД для реализации проекта. В результате анализа было принято решение использовать клиент-серверную архитектуру, поскольку она является распространенным подходом к организации взаимодействия между компьютерными системами и обеспечивает более эффективное использование ресурсов и возможность управления доступом к данным. Было решено использовать СУБД MySQL, поскольку она является бесплатным программным обеспечением с открытым исходным кодом и обладает высокой производительностью.

Для реализации серверной части были выбраны Flask и Python вместо ASP.NET и C# по ряду причин, таких как простота использования и возможность быстрого создания MVP веб-приложений. Flask также является микрофреймворком, что обеспечивает минимальный набор инструментов и легкость настройки для конкретных нужд проекта.

В качестве клиентских технологий были выбраны HTML, CSS и JavaScript. HTML используется для описания структуры веб-страницы, CSS - для оформления и стилизации, а JavaScript - для реализации динамического поведения страницы и взаимодействия с сервером. Выбор данных технологий обусловлен их широкой поддержкой и возможностью создания динамических и удобных интерфейсов.

## 2.3 Проектирование пользовательского интерфейса

Разрабатываемый сайт – Healthy People. Сайт предоставляет пользователю возможность составлять рацион питания по базовому обмену веществ и выбор программы тренировок по предпочтениям. Цель сайта – стать помощником для новичков в ЗОЖ. Для входа на сайт пользователь должен пройти регистрацию или аутентификацию.

После входа пользователь может рассчитать свой базовый обмен веществ, для этого он должен внести следующую информацию: возраст, рост, вес, уровень физической активности. (Изначально эти поля после регистрации пустые). Далее пользователь сможет составить рацион питания, выбирая из каталога продукты. Дневной рацион состоит из 4 приёмов пищи.

Перейдя на другую страницу сайта, пользователь может выбрать программу тренировок из каталога по собственным предпочтениям. Каталог программ, не хранится в базе данных, он будет создаваться из различных программ тренировок по запросу.

Данная предметная область предназначена для одной категории пользователей.

При создании интерфейса веб-приложения в первую очередь стоит учесть, сколько страниц будет на сайте. При определении требований к сайту были определены следующие функциональные страницы: страница подбора программы тренировок, страница подбора рациона питания и страницу личного кабинета. А из нефункциональных требований необходимо добавить страницу авторизации и регистрации. Ссылка на макет сайта: [healthy people – Figma](https://www.figma.com/file/RR90aXECK7xntvKLiXqs6V/healthy-people?node-id=0-1&t=y8L9szDNDRNy67iM-0) [см. https://www.figma.com/file/RR90aXECK7xntvKLiXqs6V/healthy-people?node-id=0-1&t=q9beFL7oUPbP5hss-0].

### 2.3.1 Создание ключевого персонажа

Создание ключевого персонажа — это важный шаг в проектировании интерфейса, который позволяет ориентироваться на потребности конечного пользователя. При определении ключевого персонажа, необходимо учитывать основные характеристики и потребности целевой аудитории, такие как возраст, пол, уровень образования, интересы и т.д.

Выбор пола ключевого персонажа не играет роли, поэтому далее будет описан персонаж мужского пола.

1. Идентификатор имя: Андрей
2. Демографические показатели: Мужчина, возраст 25 лет, семейное положение неженат, детей не имеет. Вид деятельности: работает в офисе в IT сфере. Уровень достатка ниже среднего. Образование­ – IT.
3. География: житель регионов, где нет большого выбора тренеров и диетологов.
4. Психография: стиль жизни–работа над индивидуальными привычками, чтение книг и программирование является основными интересами Андрея, езда на велосипеде вечером является любимым хобби Андрея.
5. Ресурсы – Андрей не обладает большими финансами, ему не хочется тратить значительное количество денег на тренера в фитнесс зале и на диетолога. Андрея достаточно занятой человек, но пару часов найдёт для занятия спортом.
6. Опыт использования интернета: Андрей пользуется интернетом на постоянной основе, у него не возникнет проблем при использовании сайта, но он бы не хотел долго разбираться с интерфейсом сайта, ведь пользователь имеет малое количество времени на спорт.
7. Целью посещения сайта является составить план питания, зная свой базовый обмен веществ и выбрать предложенную из каталога программу тренировок, по своим предпочтениям.
8. Поведенческие характеристики: Андрей хотел бы узнать достоверность вычислений базового обмена веществ, а также по каким пропорциям строиться план питания. Он бы хотел убедиться полезные ли программы тренировок именно для него, и кто по этим программам тренируется.

В целом из характеристик ключевого персонажа можно выделить основные моменты.

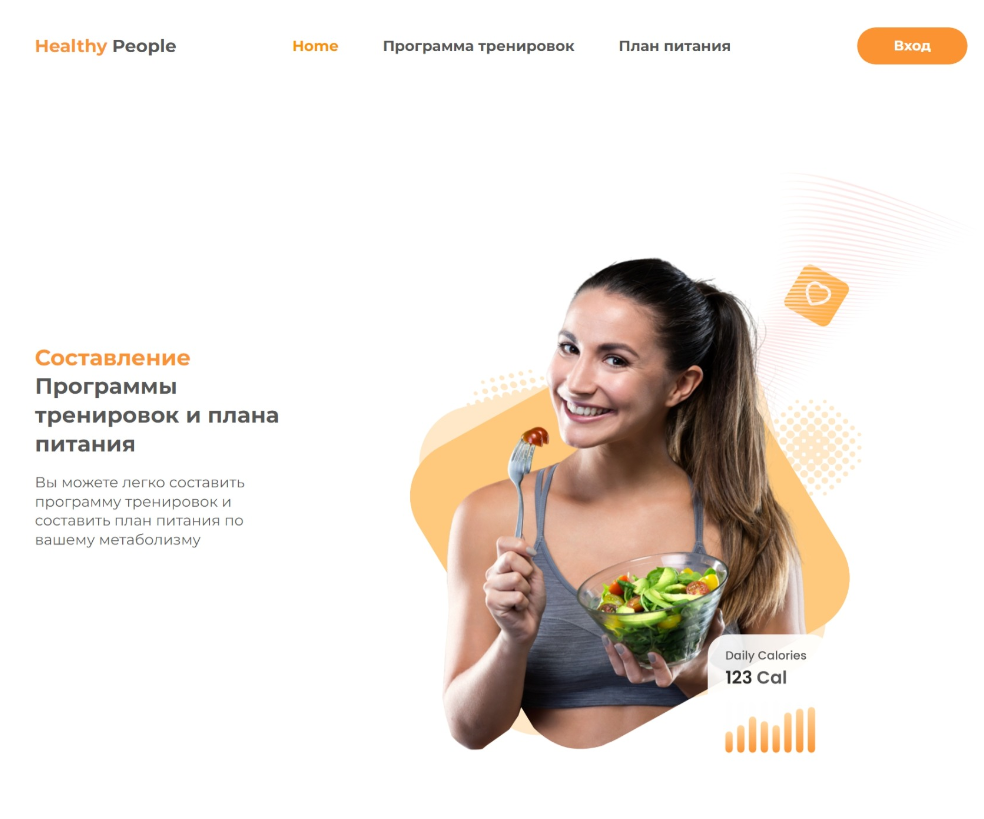
Барьеры Андрея:

* Сомневается в достоверности вычислений базового обмена веществ, а также как сайт подбирает план питания исходя из базового метаболизма;
* Хочет знать, что подобранные программы тренировок эффективны и кто по этим программам тренируется;
* Андрей не хочет тратить много времени на изучение интерфейса сайта и сразу же приступить к работе с ним;
* Андрей хочет, чтобы программа тренировок была подобранна именно под его организм.

Можно добавить следующее требование к сайту: В разделе для подсчёта базового обмена веществ будет справка, в которой написана следующая информация: по какой формуле происходит вычисление базового обмена веществ, при составлении плана питания, какие пропорции соблюдены для каждого приёма пиши, а также рекомендации какое количество калорий добавить или сократить от базового обмена веществ для похудения или увеличении мышечной массы, и рекомендации по изменению пищевой ценности( белки, жиры, углеводы) плана питания для этих двух целей.

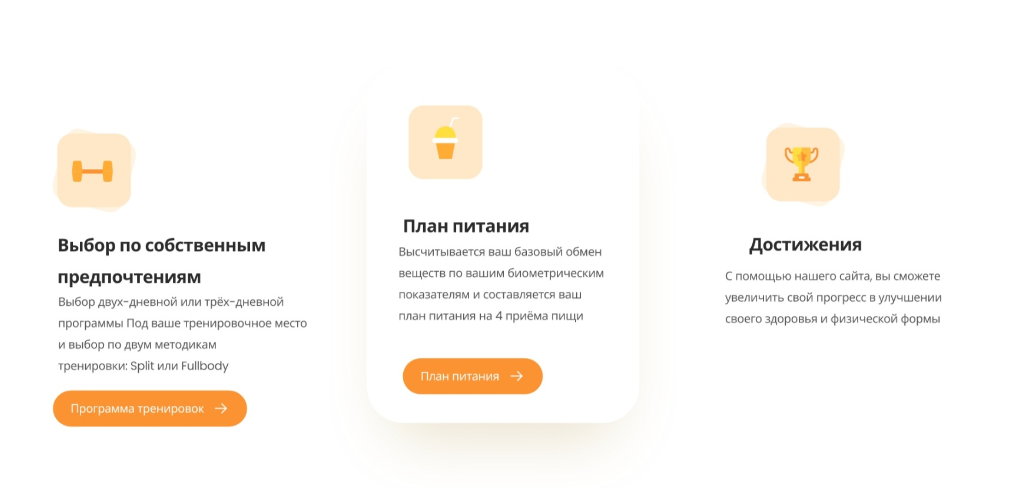
### 2.3.2 Интерфейс главной страницы

Проектирование главной страницы сайта — это важный процесс, который включает в себя не только создание панели навигации, но и размещение важной информации на странице. Панель навигации (см. Рисунок 10) содержит основные разделы сайта, которые пользователи чаще всего посещают, такие как домашняя страница, программа питания, программа тренировок, вход и кабинет. Наличие панели навигации облегчает навигацию по сайту и помогает пользователям быстро и легко находить нужны.



**Рисунок 10 – Главная страница сайта**

Однако навигационная панель не является единственным элементом главной страницы, который необходимо учитывать при ее проектировании. Важно также уделять внимание размещению важной информации о сайте (см. Рисунок 11), такой как описание функциональности сайта. Эта информация должна быть легко доступна и удобно располагаться на странице.



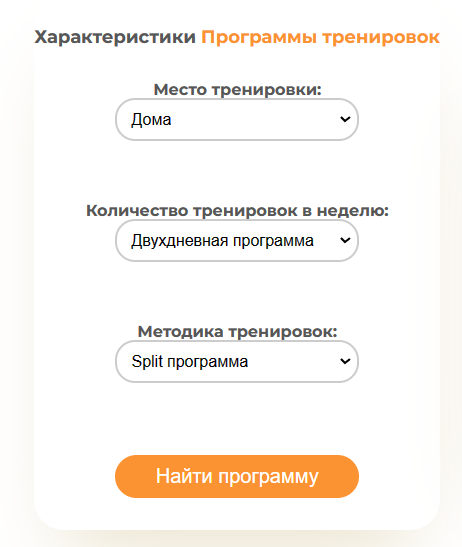
**Рисунок 11 – Общая информация о сайте**

Также на главной странице сайта можно размещать информацию о создателе, что поможет создать доверие и убедить пользователей в качестве сайта и его продуктов или услуг. Важно учитывать дизайн главной страницы, чтобы он был привлекательным и не отвлекал пользователей от главной цели посещения сайта.

В целом, проектирование главной страницы сайта с навигационной панелью и общей информацией является важным этапом создания веб-ресурса.

### 2.3.3 Интерфейс страницы “Программа тренировок”

Проектирование страницы "Программа тренировок" на сайте является важным этапом создания веб-ресурса, который предназначен для людей, занимающихся спортом или желающих начать тренировки. На странице "Программа тренировок" должна присутствовать форма (см. Рисунок 12) для выбора характеристик программы тренировок, которая поможет пользователям подобрать наиболее подходящие упражнения и режимы тренировок.



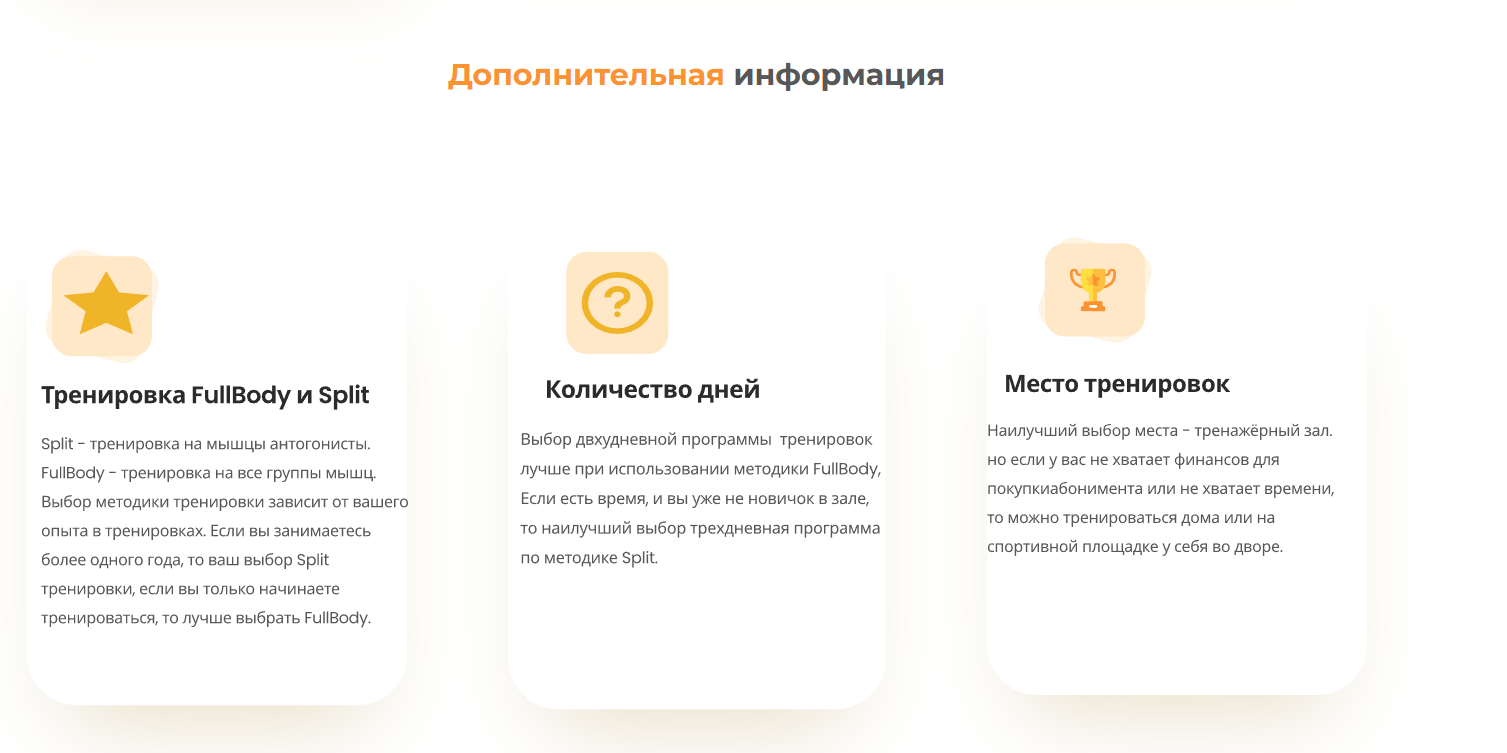
**Рисунок 12 – форма для ввода характеристик**

Таблица для представления программы тренировок (см. Рисунок 13) должна быть четкой и легко читаемой, содержать количество подходов и повторений. Кнопка для скачивания программы тренировок должна быть удобной и заметной, чтобы пользователи могли быстро скачать программу на свои устройства.



**Рисунок 13 – Таблица программа тренировок**

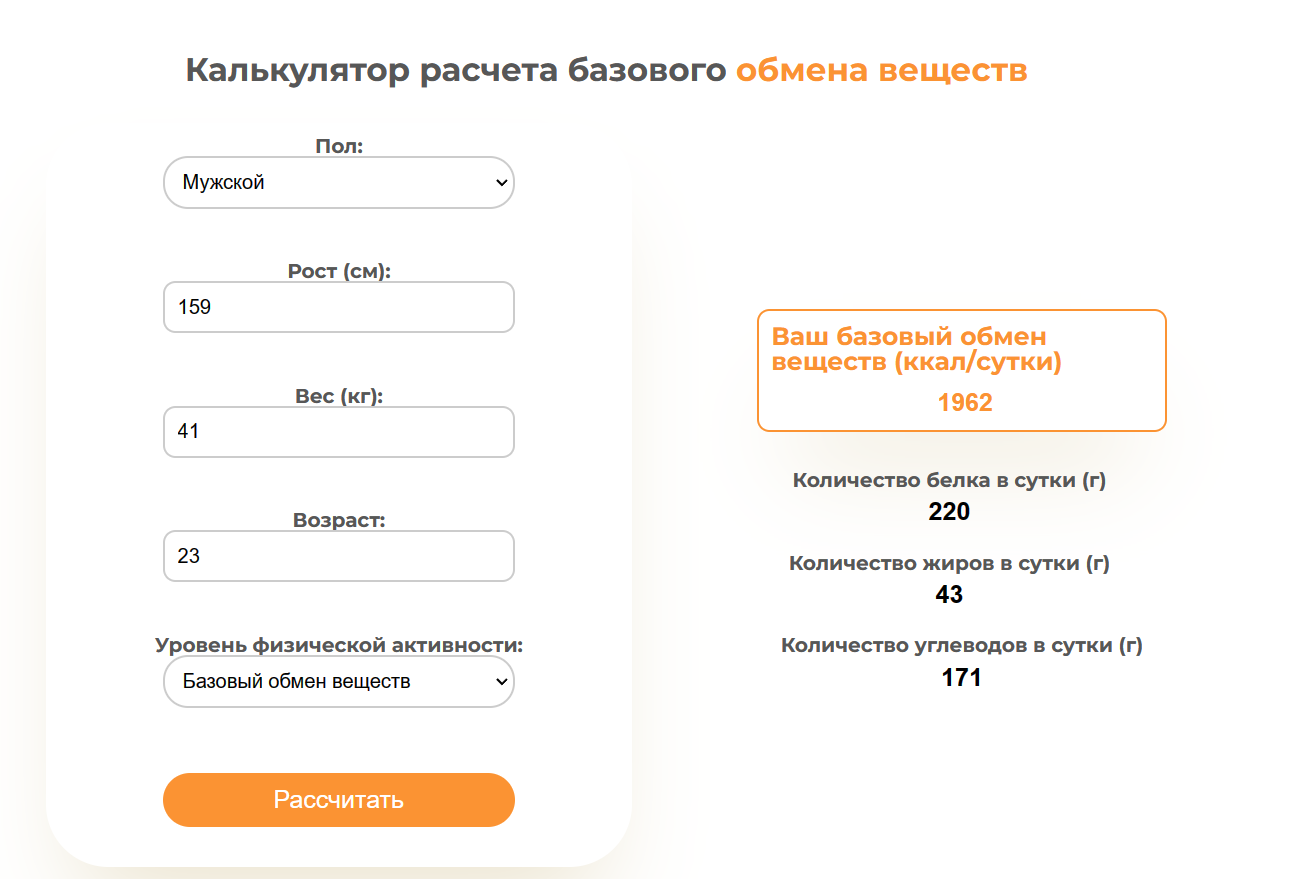
Полезная информация о тренировках (см. Рисунок 14) на странице "Программа тренировок" также является важной, поскольку она помогает правильно выбрать характеристика для программы тренировок. Дизайн страницы "Программа тренировок" должен быть привлекательным и современным, с учетом того, что пользователи часто предпочитают использовать мобильные устройства для доступа к информации. На странице также должны присутствовать удобные навигационные элементы для перехода на другие страницы сайта, такие как "Программа питания" и “Личный кабинет”.



**Рисунок 14 – Справочная информация**

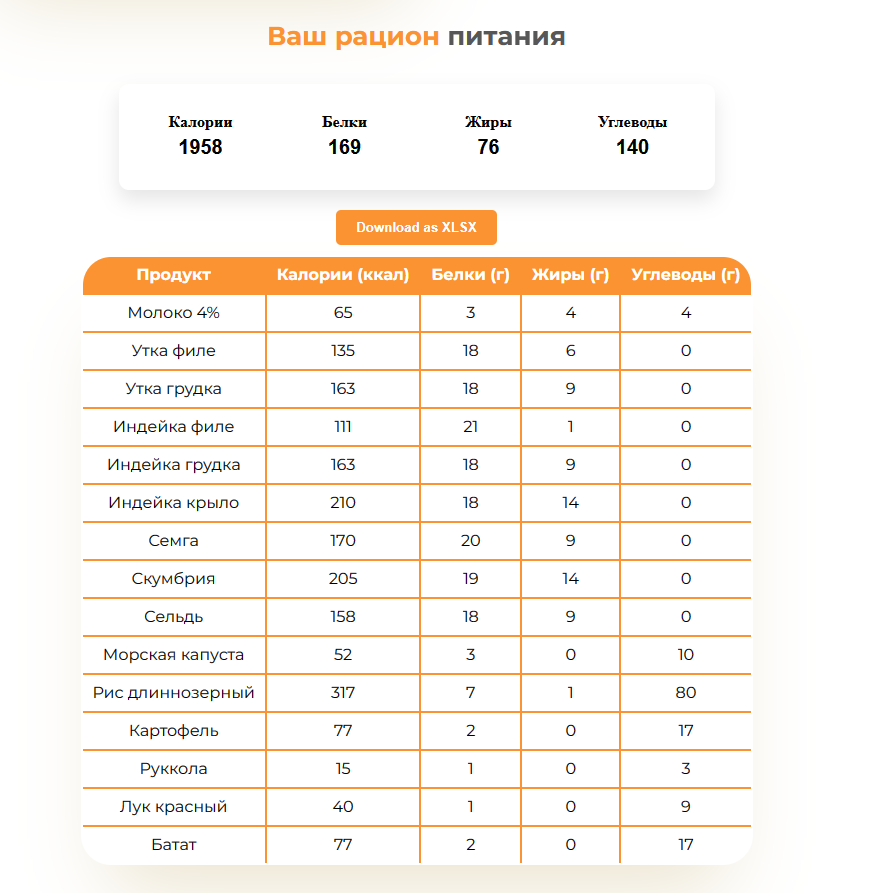
### 2.3.4 Проектирование страницы “План питания”

Страница План питания на сайте является важным компонентом для пользователей, которые хотят улучшить свое здоровье и физическую форму. Она содержит несколько элементов, которые помогают пользователям создать эффективный план питания и диету. Калькулятор расчета базового обмена веществ (см. Рисунок 15) позволяет пользователям ввести свои данные, такие как возраст, пол, рост и вес, возраст, уровень физической активности и вычислить количество калорий, необходимых им для поддержания базового обмена веществ. Эта функция помогает пользователям определить количество калорий, которое они должны потреблять в день, чтобы достичь своих целей.



**Рисунок 15 – калькулятор базового обмена веществ**

Кроме калькулятора, страница содержит информацию о рационе питания, которая помогает пользователям выбрать правильные продукты и составить правильное питание. Таблица представления рациона питания (см. Рисунок 16) помогает пользователям увидеть, какие продукты они должны включать в свой рацион и в каком количестве. Кнопка для скачивания рациона питания позволяет пользователям скачать рацион в удобном формате и сохранить его на своем устройстве.



**Рисунок 16 – Рацион питания**

При проектировании страницы План питания, важно учитывать удобство и простоту использования. Калькулятор должен быть легко доступен и прост в использовании, таблица должна быть понятной и содержать информацию, которую пользователь может быстро просмотреть. Также необходимо обеспечить хорошую читаемость текста и удобную навигацию по странице, чтобы пользователи могли легко найти нужную информацию и скачать рацион питания.

## 2.4 Проектирование базы данных

Проектирование базы данных для сайта, где пользователь выбирает программу тренировок и план питания, является важным этапом разработки. Для создания удобного и функционального интерфейса необходимо тщательно продумать структуру базы данных. Это позволит не только обеспечить правильное хранение и быстрый доступ к данным, но и улучшить пользовательский опыт. Одним из ключевых элементов при проектировании базы данных является определение соответствующих таблиц, связей между ними и выбор подходящих типов данных. Благодаря правильному подходу к проектированию базы данных сайт сможет обеспечить удобный и быстрый доступ к тренировкам и плану питания, что позволит пользователям достигать своих целей эффективно и без лишних проблем.

Информация, которой оперирует сайт (см. Приложение A. Для обеспечения эффективного и надежного хранения этой информации, необходимо выполнить процесс нормализации данных до третьей нормальной формы (3 НФ). Это позволит избежать избыточности данных и дублирования информации в таблицах, что упростит процесс обновления и модификации данных.

При проектировании необходимо учесть следующее:

* Данные по количеству КБЖУ в продуктах рассчитываются на 100 граммов продукта. Данные заполняются из справочников по питанию.
* Пароль не имеет определённой маски, шифруется хэш-функцией bcrypt.
* Уровень физической активности вносится в базу данных, как число с коэффициентом для вычисления базового обмена веществ. Так же есть текстовое пояснение, какой коэффициент выбрать.

При установлении функциональных зависимостей при проектировании БД необходимо учесть следующие связи между атрибутами:

**1ФЗ:** По коду продукта можно однозначно определить его название, калорийность, количество белков, количество углеводов, количество, жиров;

**2ФЗ:** По коду программы можно однозначно определить название программы, код дня тренировок, код места, код методики тренировок;

**3ФЗ:** По коду дня тренировок можно однозначно определить количество дней.

**4ФЗ:** По коду места можно однозначно определить название места;

**5ФЗ:** По коду методики можно однозначно определить название методики;

**6ФЗ:** По коду пользователя можно однозначно определить имя, фамилию, хэш пароля, код программы, код рациона, Email, дату рождения, рост, вес, уровень код уровня физической активности, базовый обмен веществ.

**7ФЗ:** По коду уровня физической активности можно однозначно определить название уровня физической активности и коэффициент.

**8ФЗ:** По коду рациона можно однозначно определить общее количество калорий, общее количество белков, общее количество жиров, общее количество углеводов.

**9ФЗ:** По коду упражнения можно однозначно определить название упражнения, количество подходов, а также количество повторений.

**Приведение к 3НФ:**

Отношение находится в третьей нормальной форме, если оно находится во второй нормальной форме и каждый неключевой атрибут не транзитивно зависит от первичного ключа.

В таблице «Пользователи», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В таблице «Продукты», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В таблице «Упражнения», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В таблице «Рационы питания», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В таблице «Программы тренировок», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

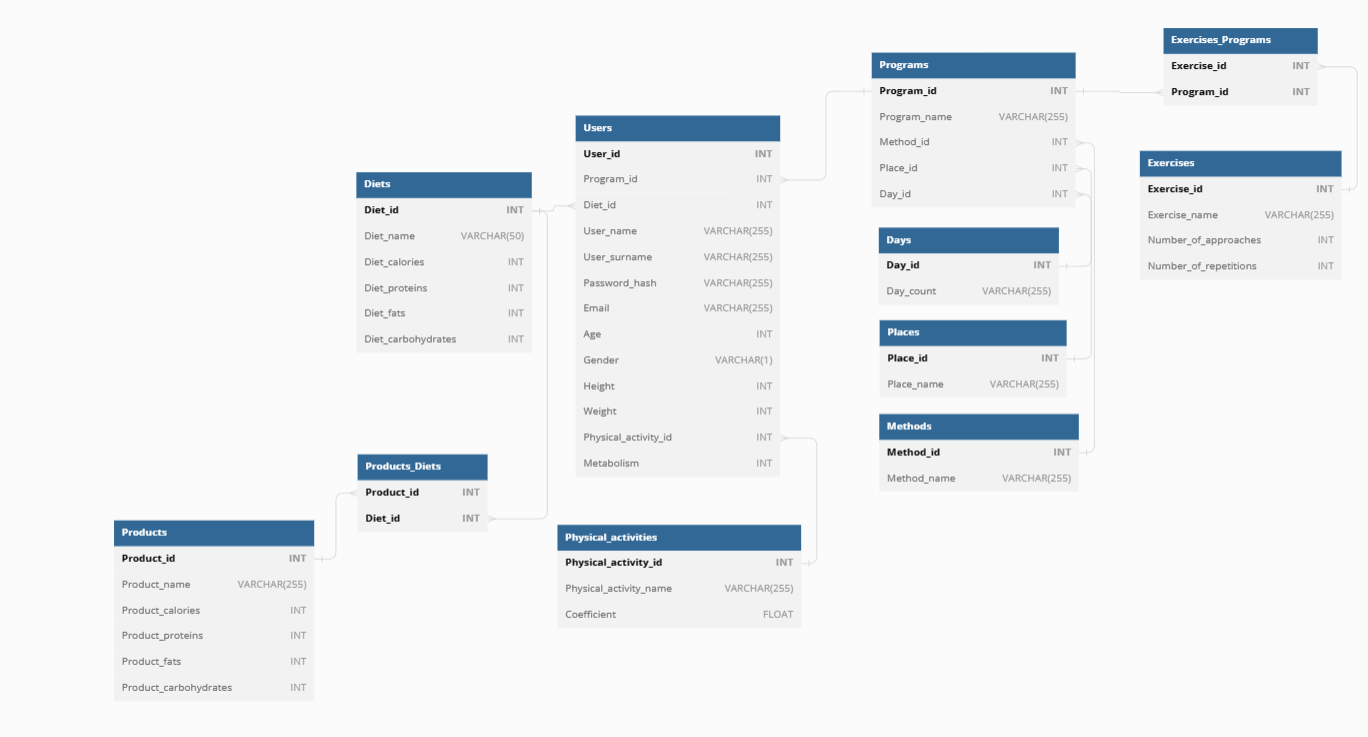
В таблице «Уровни физической активности», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В таблице «Методики тренировок», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В таблице «Дни тренировок», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В таблице «Места тренировок», отсутствуют зависимости между неключевыми атрибутами, следовательно она находиться в 3НФ.

В результате нормализации получена ER-схема базы данных, где каждое отношение находиться в 3НФ (см. рисунок 17).



**Рисунок 17– ER-диаграмма базы данных**

## 2.5 Выводы к главе

Проектирование веб-приложения является важным этапом создания сайта. В главе рассмотрена разработка веб-сайта Healthy People, цель которого заключается в предоставлении пользователю персонализированной программы тренировок и питания. В ходе проектирования были определены процессы подбора программы и плана питания, а также выбрана архитектура приложения, технологии разработки и СУБД для реализации проекта.

Одним из ключевых решений, принятых в ходе проектирования, является выбор клиент-серверной архитектуры и СУБД MySQL. Это решение обосновано широкой популярностью такой архитектуры и бесплатным программным обеспечением MySQL с высокой производительностью. Кроме того, для реализации серверной части были выбраны Flask и Python вместо ASP.NET и C# из-за простоты использования и возможности быстрого создания MVP веб-приложений.

Для создания пользовательского интерфейса был использован метод ключевого персонажа, который позволил определить характеристики и потребности пользователей. В результате был разработан макет сайта на Figma. Для реализации клиентской части были выбраны HTML, CSS и JavaScript, обеспечивающие широкую поддержку и возможность создания динамических и удобных интерфейсов.

Также была спроектирована база данных, учитывающая требования проекта. Она включает таблицы для пользователей, программ тренировок и питания. В целом, проектирование веб-сайта Healthy People было выполнено с учетом потребностей пользователей и определенных технических решений, обеспечивающих эффективное взаимодействие компонентов приложения и удобство использования пользовательского интерфейса.

# Глава 3 Разработка приложения Healthy People

После проектирования веб-приложения следует приступить к реализации самого сайта. В этой главе описана разработка функционала, тестирование, а также показан результат работы.

## 3.1Разработка основного функционала сайта

В этой главе рассматривается реализация основного функционала сайта: авторизация, регистрация, подбор программы тренировок, подбор программы питания и скачивания xlsx формате. А также реализованные запросы.

Таблица описывающая все endpoint’ы приложения (см. Приложение Б), является важным инструментом для понимания и разработки API. Она содержит информацию о доступных конечных точках, необходимых параметрах и телах запросов, а также возможных кодов ответов.

Для взаимодействия с базой данных используется SQLAlchemy. Это библиотека для работы с базами данных в Python, которая позволяет создавать таблицы, вставлять, изменять и удалять данные.

### 3.1.1 Регистрация

Для регистрации был написан декоратор на /sign.html и функцию add\_user () (см. Рисунок 18).

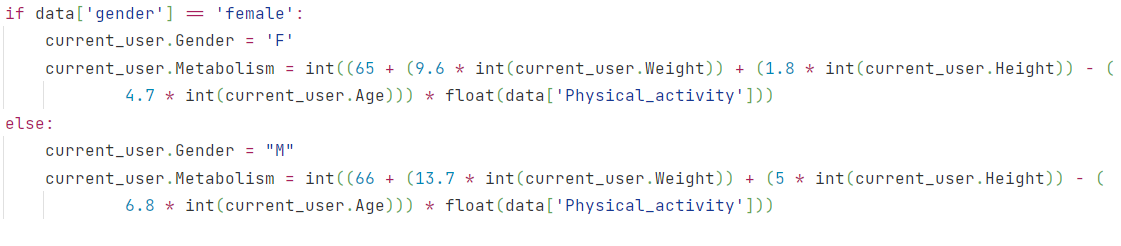


**Рисунок 18 – Декоратор и функция для регистрации**

Данный код представляет собой обработчик запросов на регистрацию пользователя на сайте. Если метод запроса POST, то из полученных данных формы формируется новый пользователь с зашифрованным паролем. Затем происходит добавление нового пользователя в базу данных и логин пользователя. В случае успешной регистрации возвращается сообщение об успешной регистрации и информация о новом пользователе. Если же метод запроса GET, то возвращается шаблон страницы для регистрации пользователя. Для хэширования пароля используется библиотека BCrypt. Эта библиотека является безопасной, по сравнению с md5. Для отслеживания сессий была использована библиотеку flask\_login и функции login\_user, current\_user.

### 3.1.2 Вычисление базового обмена веществ и подбор программы питания

Для вычисления базового обмена веществ (БОВ) в этом коде используется формула, которая зависит от нескольких факторов, таких как вес, рост, возраст и пол. В зависимости от пола, формула может отличаться. Вот код для вычисления БОВ в зависимости от пола (см. Рисунок 19).

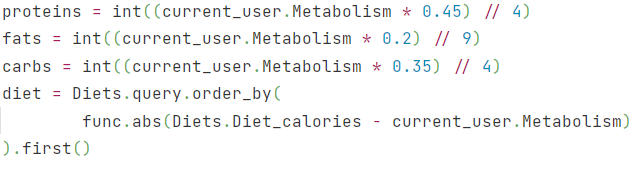


**Рисунок 19 – Вычисление БОВ**

Здесь сначала проверяется, является ли пол, указанный пользователем, женским (female) или мужским (male), и устанавливаем соответствующий флаг. Затем используется формула для вычисления БОВ в зависимости от пола. Для женщин используется формула Mifflin-St Jeor, а для мужчин – формула Harris-Benedict.

Также в коде используется физическая активность пользователя для уточнения БОВ, что позволяет более точно подобрать программу питания. Коэффициенты физической активности задаются в запросе пользователя и передаются в код.

После вычисления БОВ в коде используется следующий блок для подбора программы питания: (см. Рисунок 20).



**Рисунок 20 – Подбор программы питания**

Здесь сначала вычисляется количество белков, жиров и углеводов, необходимых для пользователя в соответствии с его БОВ. Затем выбирается диету из базы данных, которая наиболее близка по количеству калорий к БОВ пользователя. Это делается с помощью запроса к базе данных, который сортирует диеты по разнице в калориях и выбирает первую диету. Затем устанавливается идентификатор выбранной диеты для пользователя и сохраняем его в базу данных. Затем выбираются продукты, соответствующие выбранной диете, и передаем их в качестве аргумента в шаблон для отображения пользователю. Если у пользователя уже есть выбранная диета, то просто выбираются продукты, соответствующие этой диете, и передаем их в шаблон. Если у пользователя еще нет выбранной диеты, то просто передаются значение None в качестве аргумента в шаблон.

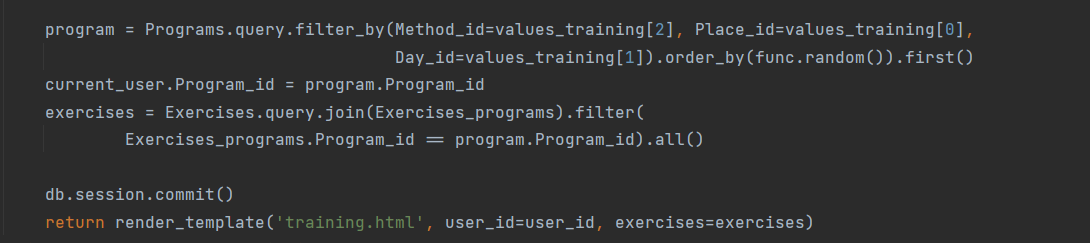
### 3.1.3 Подбор программы тренировок

Для реализации подбора программы тренировки, был использован запрос к базе данных, который фильтрует программы тренировок по следующим параметрам: метод тренировки (FullBody или split), место тренировки (дом или спортзал) и частота тренировок (два или три дня в неделю). Затем выбирается случайную программу из отфильтрованного списка и устанавливаю ее идентификатор в профиле пользователя.

Если пользователь уже имеет выбранную программу тренировки, используется ее идентификатор, чтобы выбрать упражнения, входящие в эту программу, и передать их в шаблон для отображения на странице тренировки.

Если у пользователя нет выбранной программы тренировки, отображается страница тренировки без упражнений и дополнительных инструкций, чтобы пользователь мог выбрать программу тренировок.

Часть кода для реализации подбора программы тренировки приведен ниже:



**Рисунок 21 – Подбор программы тренировок**

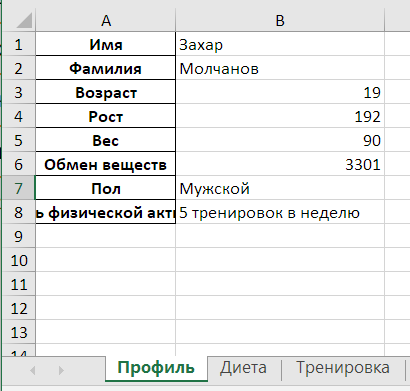
### 3.1.4 Выгрузка программы тренировок/питания

Для реализации скачивания программы тренировки нужно отправить запрос на сервер для загрузки. Для этого необходимо определить маршрут '/download/exercises' и указать метод GET. Затем нужно получить список упражнений из базы данных с помощью query -запроса, который связывает таблицы "Exercises" и "Exercises\_programs" и фильтрует упражнения по текущему пользователю и его программе тренировки. Создать объект BytesIO для хранения вывода в формате xlsx, после чего создать рабочую книгу xlsxwriter и добавить в нее новый лист. Определить заголовки столбцов и записать их на первую строку листа. Затем нужно перебрать список упражнений и добавить каждое упражнение на отдельной строке. Определить номер тренировки, чтобы разбить упражнения на группы по пять штук.

После добавления всех упражнений в лист, нужно закрыть рабочую книгу и сохранить вывод в объекте BytesIO. Создать объект Response с данными в формате xlsx и заголовками, устанавливающими тип контента и имя файла для загрузки. Ответ отправляется клиенту, который может использовать его для скачивания программы тренировки.

### 3.1.5 Отчет о пользователе

Для того чтобы реализовать скачивание отчета о пользователе, необходимо отправить GET запрос на URL-адрес '/Home/download'. Для того, чтобы пользователь мог скачать отчет, ему необходимо авторизоваться (быть залогиненным). Это обеспечивается декоратором @login\_required, который проверяет, авторизован ли пользователь, и, если нет, перенаправляет его на страницу входа. Когда пользователь отправляет GET запрос на URL-адрес '/Home/download', происходит выполнение функции download\_profile(), которая собирает информацию о пользователе и формирует Excel-файл. Далее, с помощью функции send\_file() и параметра as\_attachment=True, происходит отправка сформированного файла на компьютер пользователя в виде вложения (attachment). Таким образом, пользователь получает отчет о своих данных в формате Excel, который он может сохранить или распечатать. Пример отчета (см. Рисунок 22).



**Рисунок 22 – Отчет о пользователе**

## 3.2Тестирование сайта

После реализации основных функций сайта, необходимо было протестировать его работоспособность. Было проведено тестирование в рамках разработки веб-сайта. В ходе тестирования были протестированы основные функциональные требования сайта, включающие подбор программы питания, подбор программы тренировок и скачивание отчета о пользователе. Сценарии тестирования (см. Приложение В Таблица 1).

Также было проведено тестирование по методу черного ящика(см. Приложение В Таблица 2), которое позволило проверить корректность вычисления базового обмена веществ на сайте.

Ручное тестирование HTTP ответов от сервера также было проведено для проверки корректности работы функций регистрации и авторизации пользователей. В результате тестирования были выявлены ошибки, связанные с неправильными HTTP-ответами при вводе неверных email-адресов или паролей. Для неверных email-адресов должен был отправляться http ответ 403, но отправлялся 500. Код ошибки 403 означает "доступ запрещен", а код 500 - "внутренняя ошибка сервера". Эти ошибки были исправлены, чтобы улучшить работу сайта и обеспечить лучший пользовательский опыт.

## 3.3 Выводы к главе

Было разработано веб-приложение, которое может помочь пользователю начать свой путь к здоровому образу жизни, посредством подбора программы тренировок, высчитывания обмена веществ и подбора рациона питания. При запуске сайта пользователь попадает на главную страницу, где он должен зарегистрироваться или успешно аутентифицироваться. Далее пользователь может выбрать страницу “Программа тренировок” и задать предпочтительные характеристики для программы тренировок, затем ему будет предложена программа тренировок. Аналогично на странице “План питания” пользователь может рассчитать базовый обмен веществ и по нему система поможет подобрать план питания для поддержания веса. При необходимости пользователь может скачать: программу питания, программу питания или всю информацию о себе.

Программный код сайта находится на GitHub(см. https://github.com/ZakharMolchanov/Healthy\_People ):

# Заключение

Результатом данной работы стал сайт для подбора программы тренировок и плана питания, а также вычисления базового обмена веществ. Для выполнения этой цели были изучены необходимые материалы для разработки сайтов с подключением базы данных, а именно: фреймворки и библиотеки, их методы для реализации подключения к базе данных и выполнения различных необходимых функций. В результате изучения и сравнения аналогов были сформулированы и разработаны требования к веб-приложению. Затем на их основе был спроектирован пользовательский интерфейс и база данных. После реализации сайта Healthy People было произведено тестирование с помощью метода “Черный ящик”, в ходе которого было выявлены и устранены ошибки.

Таким образом, разработанный сайт будет может помочь людям поддерживать свое здоровье, но необходимо помнить, что также необходимо консультироваться со специалистами.

В будущем в данном приложении планируется реализовать создание гибкого рациона питания и настраиваемое программы тренировок из готовых упражнений.

# Библиографический список

1. MS SQL // MS SQL Server Documentation [Электронный ресурс] [Режим доступа: https://learn.microsoft.com/ru-ru/sql/?view=sql-server-ver16] [Дата обращения 15.01.2023]
2. ASP.NET // Microsoft Docs [Электронный ресурс] [Режим доступа: <https://docs.microsoft.com/en-us/aspnet/>] [Дата обращения: 15.01.2023]
3. Flask // Flask Documentation [Электронный ресурс] [Режим доступа: https://flask.palletsprojects.com/en/2.1.x/] [Дата обращения 20.02.2023]
4. Python // Python Software Foundation [Электронный ресурс] [Режим доступа: https://www.python.org/doc/] [Дата обращения 25.02.2023]
5. MySQL // Oracle Corporation [Электронный ресурс] [Режим доступа: https://dev.mysql.com/doc/] [Дата обращения 10.03.2023]
6. HTML // Mozilla Developer Network [Электронный ресурс] [Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/HTML>] [Дата обращения 10.01.2023]
7. CSS // Mozilla Developer Network [Электронный ресурс] [Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/CSS>] [Дата обращения 10.01.2023]
8. xlsxwriter // XlsxWriter Documentation [Электронный ресурс] [Режим доступа: <https://xlsxwriter.readthedocs.io/>] [Дата обращения 23.03.2023]
9. bcrypt // PyPI [Электронный ресурс] [Режим доступа: <https://pypi.org/project/bcrypt/>] [Дата обращения 20.02.2023]
10. JavaScript // Mozilla Developer Network [Электронный ресурс] [Режим доступа: https://developer.mozilla.org/en-US/docs/Web/JavaScript] [Дата обращения 15.01.2023]
11. Healthy\_People // GitHub [Электронный ресурс] [Режим доступа: https://github.com/ZakharMolchanov/Healthy\_People] [Дата обращения 20.03.2023]

# Приложение А Описание данных

**Таблица A.1– описание данных**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя атрибута | Тип данных | Значение по умолчанию | Формат ввода(маска) | Ограничение на значение атрибута | Комментарий |
| Код пользователя | Строка символов | нет | 00000000 | Атрибут является уникальным и обязательным для заполнения | 8- разрядный код, в котором используются только цифры |
| Имя | Строка символов | нет | нет | Значение атрибута обязательное для заполнения | Вводится в соответствии с документами |
| Фамилия | Строка символов | Нет | нет | Значения атрибута обязательное для заполнения | Вводится в соответствии с документами |
| Пол | Логическое значение | Мужской | нет | Значение атрибута обязательное для заполнения | Выбирается из предложенных кнопок. |
| Хэш Пароль | Строка символов | нет | нет? | Значение атрибута обязательное для заполнения | Пароль шифруется с помощью хэш-функции bcrypt. |
| Email адрес | Строка символов | нет | \_@\_.\_ | Значение атрибута обязательное для заполнения и является уникальным | Используется любой домен для регистрации. |
| Возраст | Целое число | нет | 00 | Значение атрибута является необязательным для заполнения | Значение­­ ­­­­– число,  Возраст вводится в диапазоне от 16 до 80. |
| Рост | Целое число | нет | 000 | Значение атрибута является необязательным для заполнения | Значение- целое число в диапазоне от 140 до 215. Измеряется в сантиметрах |
| Имя атрибута | Тип данных | Значение по умолчанию | Формат ввода(маска) | Ограничение на значение атрибута | Комментарий |
| Вес | Число с одним знаком после запятой | нет | 000 | Значения атрибута является необязательным для заполнения. | Мин.знач-40 кг, Макс.знач-160кг.  Значение – число, вес вводится в килограммах. (с точностью до десятых килограмм). |
| Уровень физической активности | Число с одним знаком после запятой | 1 | 0.0 | Значение атрибута является необязательным для заполнения | Уровень физической активности является коэффициентом для вычисления базового обмена веществ. Значение – число, уровень вводится с точностью до десятых |
| Базовый обмен веществ | Строка символов | Нет | Нет | Значение атрибута является автоматическим заполняемым, после заполнения других полей для вычисления БОВ. | Базовый обмен веществ является целочисленным с значением в диапазоне от 1000 до 9000.Измеряется в калориях. |
| Код программы тренировок | Строка символов | Нет | Нет | Значение атрибута является необязательным.  Может быть не уникальным | 8- разрядный код, в котором используются только цифры. Код программы тренировок, добавляется после выбора пользователем из каталога программ тренировок |
| Код Рациона питания | Строка символов | Нет | нет | Значение атрибута является уникальным и обязательным | 8- разрядный код, в котором используются только цифры. Код рациона питания, добавляется после выбора пользователем из каталога программ тренировок. |
| Код упражнения | Строка символов | нет | нет | Значение атрибута является уникальным и обязательным | 8 разрядный код, в котором используются только цифры. |
| Название упражнения | Строка символов | нет | нет | Значение атрибута обязательное для заполнения | Нет |
| Количество подходов | Целое число | 1 | нет | Значение атрибута обязательное для заполнения | Число в диапазоне от 1 до 6. |
| Количество повторений | Целое число | 1 | нет | Значение атрибута обязательное для заполнения | Число в диапазоне от 1 до 25. |
| Код программы тренировок | Строка символов | нет | нет | Значение атрибута является уникальным и обязательным | 8 разрядный код, в котором используются только цифры. |
| Название программы тренировок | Строка символов | нет | Нет | Значение атрибута является обязательным | Вводится название по нескольким характеристикам |
| Количество дней в программе | Целое число | 3 | нет | Значение атрибута является обязательным | Целое число, находится в диапазоне от 2 до 3 включая. |
| Методика тренировок | Строка символов | Нет | Нет | Значение атрибута является обязательным | Всего две методики тренировки Split и FULLBODY. |
| Место тренировок | Строка символов | Нет | Нет | Значение атрибута является обязательным | Всего существует два места тренировок, тренажёрный зал и дом. |
| Код продукта | Строка символов | нет | нет | Значение атрибута является уникальным и обязательным | 8 разрядный код, в котором используются только цифры. |
| Название продукта | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Вводится в соответствии ГОСТ? |
| Калорийность продукта | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Число в диапазоне от 1 до 900. Калорийность продукта на 100 грамм. |
| Количество белка в продукте | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Число в диапазоне от 1 до 100. Количество белка в продукте на 100 грамм. |
| Количество углеводов в продукте | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Число в диапазоне от 1 до 100. Количество углеводов в продукте на 100 грамм. |
| Количество жиров в продукте | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Число в диапазоне от 1 до 100. Количество жиров в продукте на 100 грамм. |
| Код рациона | Строка символов | нет | нет | Значение атрибута является уникальным и обязательным | 8 разрядный код, в котором используются только цифры. |
| Общая калорийность | Строка символов | нет | нет | Значение атрибута является обязательным | Число в диапазоне от 0 до 10000, измеряется в калориях. Является суммой калорий всех продуктов в рационе |
| Общее количество белка в рационе | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Количество белка во всём рационе питания |
| Количество углеводов в рационе | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Количество углеводов во всём рационе питания |
| Количество жиров в рационе | Строка символов | Нет | Нет | Значение атрибута является уникальным и обязательным | Количество жиров во всём рационе питания |

# Приложение Б Описание API

**Таблица Б.2 –Описание API**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Эндпоинт | Тип запроса | Параметры | Тело запроса | Возможные коды ответов | Описание |
| / | GET | Нет | Нет | 200(ОК) | Отображение главной страницы приложения |
| /sign.html | GET | Нет | Нет | 200(ОК) | Отображение страницы регистрации нового пользователя |
| /sign.html | POST | Нет | User\_name (string) User surname (string) Email (string), Password (string) | 201 (Created )- пользователь зарегистрирован и вошел в систему; 400 (Bad Request) - неверный формат запроса | Регистрация нового пользователя в системе |
| /delete-account | POST | Нет | Нет | 200 (OK) | Удаление учетной записи текущего пользователя |
| /login.html | GET | Нет | Нет | 200 (OK) | Отображение страницы входа в систему |
| /login.html | POST | Нет | Email (строка) - адрес электронной почты пользователя, Password (строка) - пароль пользователя | 201 (Created) - пользователь вошел в систему; 400 (Bad Request) - неверный формат запроса; (401 Unauthorized) - неверный адрес электронной почты или пароль | Вход в систему для зарегистрированных пользователей |
| /logout | GET | Нет | Нет | 302 - Редирект на  главную страницу | Выход из системы и перенаправление на главную страницу приложения. |
| /Home/<user\_id>/<user\_name> | GET | user\_id (int), user\_name (string) | Нет | 200 (OK) | Отображение главной страницы после авторизации |
| /Home/<user\_id>/diets | GET | user\_id (int) | Нет | 200(ОК) 404(Not Found) 401(Unauthorized) | Отображение информации о сохраненной диете и продуктах, а также данных пользователя. |
| /Home/<user\_id>/diets | POST | user\_id (int) | Weight(int), Height(int)), Age(int), Gender(string), Physical\_activity(string) | 200(ОК) 404(Not Found) 401(Unauthorized) | Внесение новой информации о пользователе, пересчитывает базовый обмен веществ и подбирает новую программу питания |
| /Home/<user\_id>/training | GET | user\_id (int) | Нет | 200(ОК) 404(Not Found) 401(Unauthorized) | Отображение информации о сохранной программе тренировок, а также характеристик программы |
| /Home/<user\_id>/training | POST | user\_id (int) | Location(string), Frequency(string), Method(string) | 200(ОК) 404(Not Found) 401(Unauthorized) | Внесение новой программы тренировок пользователя. |
| /download/exercises | GET | Нет | Нет | 200(ОК) 401(Unauthorized) | Получение программы тренировок пользователя, сохранение программы тренировок в xlsx формате |
| /Home/<user\_id>/diets/download | GET | user\_id (int) | Нет | 200(ОК) 401(Unauthorized) | Получение программы тренировок пользователя, сохранение программы тренировок в xlsx формате |
| /Home/<user\_id>/profile | GET | user\_id (int) | Нет | 200(ОК) 401(Unauthorized) | Получение данных пользователя, и отображение этих данных. |
| /Home/download | GET | Нет | Нет | 200(ОК) 401(Unauthorized) | Получение всей информации о пользователе, и сохранение этих данных в xlsx формате. |

# Приложение В Сценарии тестирования

|  |  |
| --- | --- |
| Заголовок тест-кейса: Подбор программы тренировок | |
| Предусловие: Пользователь авторизован, открыта страница “Программа тренировок” | |
| Шаг | Ожидаемый результат |
| Ввод корректных данных характеристик программа тренировок | Отправился запрос на сервер. |
| Нажать на кнопку "Подобрать" | Вывод программы тренировок на экран. |
| Результат тестирования: | Положительный |

**Таблица Б.1 –Тест-кейс к требованию побор программы тренировок**

|  |  |
| --- | --- |
| Заголовок тест-кейса: Подбор плана питания | |
| Предусловие: Пользователь авторизован, открыта страница “План питания” | |
| Шаг | Ожидаемый результат |
| Ввести корректные данные. | Отправился запрос на сервер, вывод БОВ. |
| Нажать на кнопку “рассчитать” | Вывод рациона питания и информации о нем на экран. |
| Результат тестирования: | Положительный |

**Таблица Б.2 – Тест-кейс к требованию о подбор плана питания**

|  |  |
| --- | --- |
| Заголовок тест-кейса: Скачивание отчета о пользователе | |
| Предусловие: Пользователь авторизован, открыта страница с личным кабинетом | |
| Шаг | Ожидаемый результат |
| Нажать на кнопку” скачать в xlsx формате отчет” | Отчет со всей информации о пользователе скачен |
| Результат тестирования: | Положительный |

**Таблица Б.3 – Тест-кейс к требованию о скачивании всей информации о пользователе**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Вес (кг) | Пол | Возраст (лет) | Уровень физической активности | Рост (см) | Ожидаемый метаболизм (ккал/день) | Текущий метаболизм (ккал/день) | Совпадение результата с ожиданием |
| 70 | М | 30 | 1 | 180 | 1670 | 1670 | Да |
| 60 | Ж | 25 | 2 | 160 | 1326 | 1326 | Да |
| 80 | М | 40 | 3 | 175 | 1922 | 1922 | Да |
| 65 | Ж | 20 | 4 | 170 | 1638 | 1638 | Да |
| 90 | М | 50 | 5 | 185 | 2440 | 2440 | Да |
| 55 | Ж | 18 | 2 | 165 | 1268 | 1268 | Да |

**Таблица Б.4 –Результаты тестирования расчета метаболизма методом**

# Приложение Г Техническое задание

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет экономики, менеджмента и бизнес-информатики  
Образовательная программа бакалавриата «Программная инженерия»



|  |  |
| --- | --- |
| СОГЛАСОВАНО  Руководитель проекта,  Преподаватель кафедры “Информационных технологий в бизнесе”  \_\_\_\_\_\_\_\_\_\_\_\_М. В. Иванова  «\_ » 2023 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы  «Программная инженерия» доцент, канд. тех. наук  В. В. Ланин  «\_ » 2023 г. |

**Разработка сайта для составления и выбора программы   
тренировок и платна питания**

**Техническое задание  
ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.03-01 ТЗ 01-1-ЛУ**

Исполнитель:

студент группы

З. В. Молчанов

«\_ » 2023г.

**УТВЕРЖДЕН**

**RU.17701729.12151-01 ТЗ 01-1-ЛУ**



**Разработка сайта для составления и выбора программы   
тренировок и платна питания**

**Техническое задание**

**RU.17701729.12151-01 ТЗ 01‑1**

**Листов 17**

**Оглавление**

[1 Введение 3](#_Toc105410780)

[1.1 Наименование программы 3](#_Toc105410781)

[1.2 Краткая характеристика области применения программы 3](#_Toc105410782)

[2 Основание для разработки 4](#_Toc105410783)

[2.1 Основание для проведения разработки 4](#_Toc105410784)

[2.2 Наименование и условное обозначение темы разработки 4](#_Toc105410785)

[3 Назначение разработки 5](#_Toc105410786)

[3.1 Функциональное назначение программы 5](#_Toc105410787)

[3.2 Эксплуатационное назначение программы 5](#_Toc105410788)

[4 Требования к программе 6](#_Toc105410789)

[4.1 Требования к функциональным характеристика 6](#_Toc105410790)

[4.2 Требования к надежности 6](#_Toc105410791)

[4.3 Условия эксплуатации 6](#_Toc105410792)

[4.4 Требования к составу и параметрам технических средств 6](#_Toc105410793)

[4.5 Требования к информационной и программной совместимости 7](#_Toc105410794)

[4.6 Требования к маркировке и упаковке 7](#_Toc105410795)

[5 Требования к программной документации 8](#_Toc105410796)

[5.1 Состав программной документации 8](#_Toc105410797)

[5.2 Специальные требования к программной документации 8](#_Toc105410798)

[6 Технико-экономические показатели 10](#_Toc105410799)

[7 Стадии и этапы разработки 11](#_Toc105410800)

[8 Порядок контроля и приемки 14](#_Toc105410801)

[9 Приложения 15](#_Toc105410802)

* 1. [Терминология 15](#_Toc105410803)

**1 Введение**

**1.1 Наименование программы**

Наименование – «Healthy People».

* 1. **Краткая характеристика области применения программы**

Программа предназначена для выбора программы тренировок и плана питания по предпочтениям.

Данная программа должна реализовывать основные функции:

* подбор программы тренировок;
* подбор рациона питания;
* рассчитывать базовый обмен веществ.

**2 Основание для разработки**

**2.1 Основание для проведения разработки**

Написать, что в соответствии с программой практики НИУ ВШЭ – Пермь и дать номер приказа.

**2.2 Наименование и условное обозначение темы разработки**

Тема – «Разработка сайта для подбора программы тренировки и плана питания».

**3 Назначение разработки**

**3.1 Функциональное назначение программы**

Сайт предоставляет возможность подобрать программу тренировок по собственным предпочтениям, рассчитывание базового обмена веществ по характеристикам пользователя, а также подбор рациона питания по обмену веществ.

**3.2 Эксплуатационное назначение программы**

Программа может быть использована людьми, которые хотят подобрать для себя программу тренировок, а также начать следить за своим питанием.

1. **Требования к программе**

**4.1 Требования к функциональным характеристикам**

Программа состоит из клиентской и серверной части. Взаимодействие с пользователем происходит через приложение.

Функциональные требования:

1. Возможность регистрации нового пользователя на сайте.
2. Возможность авторизации зарегистрированных пользователей.
3. Подбор программы тренировок по выбранным характеристикам.
4. Скачивание программы тренировок, плана питания и всей информации о пользователе в xlsx формате.
5. Подбор плана питания по базовому обмену веществ.
6. Вычисление базового обмена веществ.

**4.2 Требования к надежности**

Программа должна безотказно работать, в базе данных у пользователя должен храниться хэш пароля.

**4.3 Условия эксплуатации**

* + 1. **Климатические условия эксплуатации**

Требований к климатическим условиям эксплуатации не предъявляется.

* + 1. **Требования к видам обслуживания**

Обслуживание не требуется.

* + 1. **Требования к численности и квалификации персонала**

Техническая поддержка производится силами разработчика. Для пользования программой дополнительное обучение не требуется.

* 1. **Требования к составу и параметрам технических средств**
     1. **Требование клиентской части**

Устройство с доступом в интернет;

* + 1. **Требование серверной части**

Возможность одновременного подключения нескольких пользователей.

**4.5 Требования к информационной и программной совместимости**

**4.5.1 Требования к исходным кодам и языкам программирования**

Исходные коды программы должны быть написаны на языках Python, CSS, JavaScript, HTML и выложены на сервис для хостинга IT проектов GitHub.

**4.6 Требования к маркировке и упаковке**

Программа распространяется посредством URL ссылки на сайт.

**5 Требования к программной документации**

**5.****1 Состав программной документации**

Состав программной документации:

* Сайт для составления и выбора программы тренировок и плана питания.  
  Техническое задание (ГОСТ 19.201–78);
* Сайт для составления и выбора программы тренировок и плана питания.  
  Руководство оператора (ГОСТ 19.505–79).

**5.2 Специальные требования к программной документации**

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106–78 и ГОСТами к каждому виду документа (см. п. 5.1.).

**6 Технико-экономические показатели**

В рамках данной работы расчёт экономической эффективности не предусмотрен.

Использование разрабатываемого инструмента предназначено для помощи людям начать тренироваться и следить за питанием.

Предполагаемая потребность обуславливается тем фактом, что на данный момент трудно найти бесплатный и с полной функциональностью сайт для подбора и составления программы тренировок и плана питания.

Существующие аналоги не дают возможности работы с исходным кодом и тонкой настройки функциональных возможностей.

**Стадии и этапы разработки**

Количество и длительность проекта определяется итерационным характером модели жизненного цикла продукта и линейным характером жизненного цикла проекта.

С точки зрения управления проектом выделяются стадии:

* 1. Инициация.
  2. Планирование.
  3. Исполнение.
  4. Завершение.

С точки зрения управления жизненным циклом продукта проект состоит из трех итераций, каждая из которых состоит из пяти этапов:

1. Инициализация.
2. Проектирование.
3. Реализация.
4. Тестирование.
5. Развертывание

Более подробно этапы проекта, входящие в них задачи, их продолжительность, ресурсы и ожидаемые результаты описаны ниже в таблице 1.

Таблица 3 – Этапы проекта

| **Номер**  **ИСР** | **Описание задачи** | **Сроки**  **выполнения** | **Исполнитель** | **Результаты** |
| --- | --- | --- | --- | --- |
| 1 | **Анализ и планирование проекта** | | | |
| 1.1 | Выявить требования Заказчика | 18.01.2023 | Молчанов З. В. | Протоколы общения с Заказчиком |
| 1.2 | Формализовать требования Заказчика | 18.01.2023-25.01.2023 | Молчанов З. В. | Реестр  требований |
| 1.3 | Протестировать требования Заказчика | 26.01.2023-02.02.2023 | Молчанов З. В. | План управления рисками, план тестирования |
| 1.4 | Утвердить требования Заказчика | 03.02.2023-05.02.2023 | Молчанов З. В. | Утвержденный реестр требований |
| 1.5 | Распределить задачи по членам Проектной группы | 06.02.2023 | Молчанов З. В. | График работ |
| 2 | **Проектирование и создание прототипа продукта** | | | |
| 2.1 | Спроектировать архитектуру Системы | 07.02.2023-17.02.2023 | Молчанов З. В. | Техническое задание, руководство программиста |
| 2.2 | Разработать прототип Системы | 17.02.2023-24.02.2023 | Молчанов З. В. | Прототип Системы, руководство программиста, руководство пользователя |
| 2.3 | Протестировать прототип Системы | 24.02.2023-27.02.2023 | Молчанов З. В. | План тестирования |
| 2.4 | Актуализировать требования Заказчика | 28.02.2023 | Молчанов З. В. | Прототип Системы |
| 2.5 | Актуализировать проектные документы | 1.03.2023-  6.03.2023 | Молчанов З. В. | Реестр требований, техническое задание |
| 2.5.1 | Сформировать уточненный список требований | 7.03.2023-13.03.2023 | Молчанов З. В. | Реестр требований, техническое задание |
| 2.5.2 | Протестировать требования | 12.03.2023-13.03.2023 | Молчанов З. В. | План тестирования |
| 2.5.3 | Провести демонстрацию прототипа Системы Заказчику | 13.03.2023 | Молчанов З. В. | Документ о принятии работ |
| 3 | **Реализация и тестирование** | | | |
| 3.1 | Актуализировать требования Заказчика | 14.03.2023 | Молчанов З. В. | Протоколы общения с Заказчиком, реестр требований |
| 3.2 | Обновить документацию в соответствии с новыми требованиями | 14.03.2023- 17.03.2023 | Молчанов З. В. | Техническое задание, руководство программиста, руководство пользователя |
| 3.3 | Разработать Систему | 17.03.2023- 24.03.2023 | Молчанов З. В. | Итоговый вариант Системы |
| 3.4 | Протестировать Систему | 24.03.2023- 25.03.2023 | Молчанов З. В. | Приемо-сдаточные испытания Системы, План тестирования |
| 3.5 | Провести приемо-сдаточные мероприятия Системы | 26.03.2023 | Молчанов З. В. | Документ о принятии работ |

**8 Порядок контроля и приемки**

Написать, что в соответствии с программой практики НИУ ВШЭ – Пермь и дать номер приказа.

# Приложение Д Руководство оператора

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет экономики, менеджмента и бизнес-информатики  
Образовательная программа бакалавриата «Программная инженерия»



|  |  |
| --- | --- |
| СОГЛАСОВАНО  Руководитель проекта,  Преподаватель кафедры “Информационных технологий в бизнесе”  М. В. Иванова  «\_ » 2023 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы  «Программная инженерия» доцент, канд. тех. наук  В. В. Ланин  «\_ » 2023 г. |

**Сайт для подбора программы тренировок и плана питания**

**Руководство оператора  
ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.03-01 РО 01-1-ЛУ**

Исполнитель:

студент группы

З. В. Молчанов

«\_ » 2023 г.

**УТВЕРЖДЕН**

**RU.17701729.12151-01 РО 01-1-ЛУ**



**Сайт для подбора программы тренировок и плана питания**

**Руководство оператора**

**RU.17701729.12151-01 РО 01‑1**

**Листов 7**

**Оглавление**

[1 Назначение программы 4](#_Toc105441129)

[2 Условия выполнения программы 5](#_Toc105441130)

[3 Выполнение программы 6](#_Toc105441131)

[3.1 Установка программы 6](#_Toc105441132)

[3.2 Запуск программы 6](#_Toc105441133)

[3.3 Выполнение программы 6](#_Toc105441134)

[3.3.1 Меню настроек 6](#_Toc105441135)

[3.3.2 Каталог уроков 6](#_Toc105441136)

[3.3.3 Окно урока 6](#_Toc105441137)

[3.3.4 Управление программой 7](#_Toc105441138)

[3.4 Завершение программы 7](#_Toc105441139)

[4 Сообщения оператору. 8](#_Toc105441140)

**1 Назначение приложения**

Приложение предназначено для подбора программы питания и программы тренировок.

Данная программа должна исполнять основные функции:

* подбор программы тренировок по заданным характеристикам;
* вычисления базового обмена веществ по характеристикам;
* подбор плана питания по обмену веществ;
* скачивание всей информации о пользователе в xlsx формате.

**2 Условия выполнения приложения**

Приложение должно выполняться без ошибок на устройствах с доступом в интернет и любым браузером.

**3 Выполнение приложения**

**3.1 Запуск приложения**

Сайт запускается посредством ввода ссылки в браузер. В результате чего перед пользователем открывается главное окно сайта. Далее пользователь, для того чтобы воспользоваться функциями сайта должен зарегистрироваться.

**3.2 Выполнение программы**

**3.2.1 Регистрация или вход**

После запуска сайта пользователь должен нажать на кнопку входа и ввести email и пароль, если у него нет аккаунта, то он должен нажать на кнопку зарегистрироваться, чтобы открылась страница регистрации. Пользователь должен ввести следующие данные в форму: имя, фамилия, email, пароль, повторить пароль.

**3.2.2 Главная страница**

После регистрации пользователь находится на главной странице сайта, и он может ознакомиться со справочной информацией о функциях сайта.

**3.2.3 Страница программы тренировок**

На главной странице пользователь может перейти на страницу программа тренировок, там он может выбрать нужные характеристики программы тренировок, после ознакомления с информацией на странице. По этим характеристикам пользователю выведется подходящая программа тренировок, после чего пользователь сможет ее скачать в xlsx формате.

**3.2.4 Страница плана питания**

Пользователь на главной странице может перейти на страницу “Плана питания”. На этой странице он должен сначала ввести свои данные, такие как: рост, вес, пол, возраст, уровень физической активности. После чего пользователь нажимает кнопку рассчитать, и выводится его обмен веществ в калл, также рекомендации сколько принимать белка, жиров, углеводов в сутки. В этот же момент, выводится подходящая программа питания для пользователя, пользователь также может скачать ее в xlsx формате.

**3.2.5 Страница личного кабинета**

После всех действий пользователь может перейти на страницу личного кабинета. Там он может ознакомиться со всей его информацией. Пользователь может скачать все данные о себе в xlsx формате, с выбранной программой тренировки и плана питания. Если пользователь хочет очистить себя из базы данных, пользователь может удалить свой аккаунт.

**3.2.6 Управление приложением**

Пользователь управляет сайтом с помощью компьютерной мыши и клавиатуры, осуществляя тем переход между страницами сайта, ввод данных.

**3.7 Завершение приложения**

Приложение прекращает работу в тот момент, когда пользователь закрывает вкладку сайта, подтверждая тем самым желание завершения работы

**4 Сообщения оператору**

В ходе ввода информации оператор может получать следующие сообщения:

Сообщение об ошибке при входе в систему: "Неверный логин или пароль. Пожалуйста, попробуйте снова."

Сообщение об ошибке при регистрации: "Введенные пароли не совпадают. Пожалуйста, попробуйте снова."

Сообщение об ошибке при вводе данных в форму расчета базового обмена веществ: "Введенные данные не подходят под допустимый размер. Пожалуйста, проверьте введенные значения и попробуйте снова."