**Техническое задание**

**Яндекс.Fitness**

**Что делает сайт?**

Ежедневник для здоровья и фитнеса: приложение, которое позволяет пользователям вести записи о своем питании, тренировках и общем состоянии здоровья.

Они могут отслеживать свой прогресс и устанавливать цели для улучшения здоровья.

**Функциональность:**

1. **Учет физической активности**:

Пользователи могут добавлять свои тренировки, указывать тип активности (бег, йога, плавание и т.д), длительность, интенсивность и другие параметры.

1. **Учет питания**:

пользователи могут вводить информацию о своем рационе, включая приёмы пищи, продукты, потребляемые калории, питательные вещества и т.д.

1. **Учет здоровья:**

Возможность отмечать основные показатели здоровья, такие как вес, давление и другие.

1. **Установка и отслеживание целей:**

Пользователи могут устанавливать цели, такие как потеря веса, повышение физической выносливости, улучшение своего питание. Затем можно отслеживать прогресс выполнения целей.

1. **Рекомендации и советы:**

Возможность предоставления пользователю рекомендаций и советов по здоровому образу жизни на основе их целей и прогресса.

1. **Аутентификация и авторизация**

**Стек технологий**

1. Flask
2. Python 3.10
3. Sqlalchemy
4. Json
5. Bootstrap, Materialize
6. Flask-login
7. Wtf-forms, flask-wtf

**Технические характеристики**

Рабочая среда

Операционная система

OC Windows 11

СУБД: SQLite 3

Язык программирования:

Python с использование встроенных библиотек, а также указанных в стеке технологий.

**Общее положение**

1. Главная страница

**1.2)** Авторизация и аутентификация

**2)** Учёт физической активности

**3)** Учёт питания

**4)** Отслеживание целей

**5)** Личный кабинет

**6)** Учёт здоровья.

1. **Главная страница – адрес(“/”)**

Основная страница с описанием возможностей сайта, контактами организации и кнопками для входа и регистрации. Для стилизации этой страницы будет использоваться materialize.

1. **Авторизация и аутентификация**

(“/register” и “/login”).

* Попасть на эти страницы можно будет с главной через кнопки регистрации и входа соответственно.
* Форма регистрации и логина будут реализованы с помощью wtforms, а пользователь будет храниться в базе данных users.
* Работа с базой данных будет проводиться с помощью sqlalchemy.
* Поля модели пользователя:
  + Имя
  + Фамилия
  + Хэшированный пароль
  + Почта
  + Дата рождения
  + Словарь с характеристиками (рост, вес и др.)

При регистрации будет необходимо указать свои имя и фамилию, а также почту и пароль (с повторением). Их можно будет изменить в профиле.

1. **Учёт физической активности (“/activities”)**

Пользователи могут добавлять свои тренировки на неделю, указывать тип активности, а также дату и время начала и конца.

Будет реализовано с помощью модели Activity со следующими полями:

1.Название активности (Обязательное поле)

2. Тип активности

3.Время начала (Обязательное поле)

4.Время окончания (Обязательное поле)

5. День недели (По умолчанию равно текущему дню недели)

Работа с данным будет реализована через собственное api.

Будет возможность удалять и изменять активности.

1. **Страница профиля (“/profile”)**

Основная страница. Содержит весь необходимый интерфейс для получения доступа ко всем возможность сайта.

По мимо этого будут отображать его данные в примерно следующем формате:

* Фамилия
* Имя
* Дата рождения
* Вес
* Рост

1. **Учёт здоровья (“/health\_condition”)**

Пользователь может вводить свои параметры здоровья в специальную

форму.

1. Уровень сахара

2. Рост

3. Вес

4. Пульс

5. Давление

Здесь можно будет увидеть график изменений показателей с течением времени. Работа с графиками будет реализовано с помощью библиотеки matpotlib.

1. **Установка и отслеживание** целей(“/goals”)

У целей будет описание, приоритет и сроки выполнения. Также будет раздел с выполненными целями (“complete\_goals”)

Цели будут храниться в таблице goals и будут привязаны к пользователю по id.

Поля модели цели (Goal):

1. Краткое название
2. Id пользователя с данной целью
3. Описание цели(Необязательно)
4. Приоритет (по умолчанию 0)
5. Ожидаемая дата выполнения(необязательно)
6. Учёт питания (“/diet”)

Пользователь может выбирать продукты, которые он съел за день.

На этой основе строится статистика по полученным полезным веществам и каллориям. Еду пользователь будет искать по названию в списке. А также выбирать примерный вес/кол-во. Все параметры еды будет прибавляться к суммарным показателям за сутки.

**Распределение обязанностей**

Егор - авторизация/аутентификация, цели, api

Кирилл - тимлид; учёт физической активности и питание.

Андрей – дизайн, учёт здоровья, рекомендации и советы, главная страница