## Введение

Современный ритм жизни требует от человека грамотного управления личными финансами, однако постоянный рост разнообразия расходов и финансовых обязательств усложняет процесс их контроля. В таких условиях актуальным становится использование специализированных цифровых решений, способных автоматизировать учет и анализ бюджета.

Представленное в курсовой работе Web API приложение SmartWallet предназначено для решения задач персонального финансового менеджмента. Программа позволяет клиентам систематизировать ежедневные расходы, устанавливать лимиты бюджета, а также анализировать финансовые привычки с помощью статистики.

SmartWallet ориентировано на широкий круг пользователей, включая студентов, фрилансеров, семьи и всех, кто стремится к осознанному управлению своими средствами. Приложение может быть интегрировано с мобильными или веб-клиентами, что обеспечивает доступ к данным с любых устройств. Ключевая особенность решения — гибкость API, позволяющая адаптировать функционал под индивидуальные потребности.

Объектом внедрения SmartWallet являются персональные финансовые системы, где требуется автоматизация рутинных операций и повышение прозрачности расходов. Программа не только упрощает ведение учета, но и способствует формированию здоровых финансовых привычек за счет анализа динамики трат и своевременных рекомендаций.

Разработка SmartWallet в формате Web API обеспечивает масштабируемость, безопасность данных и возможность дальнейшего расширения функционала, что делает приложение перспективным инструментом в сфере персональной экономики.

## 1 Основания для разработки

## Разработка ведётся на основании задания к курсовому проекту по профессиональному модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» МДК 01.01 «Разработка программных модулей».

## **2 Назначение разработки**

Цель создания Web API приложения SmartWallet — предоставить пользователям инструмент для автоматизированного учета личных расходов, планирования бюджетов и анализа финансового поведения. Функциональное назначение включает: регистрацию транзакций, категоризацию расходов, настройку лимитов и сбор статистики. Эксплуатационное назначение заключается в обеспечении доступности, безопасности данных и масштабируемости решения, что позволяет интегрировать SmartWallet с различными клиентскими приложениями (веб, мобильные) для удобного управления финансами в режиме реального времени.

### 

## **3 Требования к программе**

### **3.1 Требования к функциональным характеристикам**

Данный программный продукт должен удовлетворять следующим функциональным требованиям:

* Добавление, получение, изменение и удаление денежных хранилищ. Денежное хранилище должно обладать названием;
* Добавление, получение, изменение и удаление областей трат. Область трат должна обладать названием;
* Добавление, получение и удаление транзакций;
* Авторизация и контроль доступа. Все данные привязаны к пользователю, и только он может с ними взаимодействовать;
* Регистрация в системе. Пользователь должен указать своё ФИО, электронную почту и пароль. Пароль должен удовлетворять следующим условиям: быть больше 8 и меньше 34 символов, содержать спецсимволы, содержать заглавную букву.

Входные данные:

Файл в формате json с кодировкой UTF-8.

Выходные данные:

Файл в формате json с кодировкой UTF-8.

### **3.2 Требования к надёжности**

Программа должна соответствовать следующим требованиям к надёжности:

1. **Устойчивое функционирование:**

* Корректная обработка ошибок при попытке получения несуществующих данных.

1. **Контроль входной информации:**

* Отклонение запроса на добавление дубликатов данных в базу;
* Валидация данных: проверка формата, диапазонов и логической корректности полученных из запроса данных, и отклонение запроса в случае неудачной валидации.

1. **Контроль выходной информации:**

* Гарантия целостности данных: выходной json формируется только на основе актуальных и проверенных данных.

**3.3 Требования к информационной и программной совместимости**

Программа должна соответствовать следующим требованиям информационной и программной совместимости:

* Операционная система: Windows 10 или выше;
* Доступ к интернету;
* Программа должна быть написана на языке программирования C# на платформе ASP.NET Core.

## 4 Требования к программной документации

В процессе проектирования должны быть разработаны следующие документы:

* Пояснительная записка (с необходимыми приложениями).

## 5 Перечень графических материалов

Для данной разработки следует подготовить следующие графические материалы:

* Структуру программы;
* Диаграмма классов;
* Диаграмму базы данных.
* Диаграмма последовательности.