**МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ «ОКЭИ»)**

**КУРСОВОЙ ПРОЕКТ**

*ОКЭИ 09.02.07. 4324. 24 ПЗ*

(код документа)

*Разработка веб-сайта для управления личными финансами*

Количество листов *34*

Дата готовности  *25.01.25*

Руководитель *Адамович Н.В.*

Разработал  *Чурсина В.И.*

Защищен\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с оценкой

(дата)

Оренбург 2024

**Содержание**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

Лист

2

*ОКЭИ 09.02.07 4324. 24 ПЗ*

*Разраб.*

*Чурсина В.И.*

*Провер.*

Адамович Н.В.

*Реценз*

*Н. Контр.*

*Утверд.*

*Отчёт по*

*курсовому проекту*

Лит.

Листов

*34*

*Отделение программирования,*

*гр. 4вб2*

|  |  |  |
| --- | --- | --- |
| Введение………………………………………………………………………........ | | 3 |
| 1 Анализ предметной области................................................................................. | | 6 |
| 2 Проектирование приложения............................................................................... | | 9 |
| 3 Разработка программного обеспечения..…………….........………………….... | | 13 |
|  | * 1. Описание технологического стека разработки........................................ | 13 |
|  | * 1. Описание алгоритма работы......................................................................   2. Описание интерфейса пользователя.......................................................... | 16  20 |
| 4 Тестирование приложения.................................................................................... | | 22 |
|  | 4.1 План тестирования......................................................................................  4.2 Оценка результатов проведения тестирования........................................ | 2223 |
| Заключение……………………………………………………………………........ | | 28 |
| Список использованных источников………………………………………..........  Приложение А (обязательное) Информационная модель…………………........ | | 30  31 |
| Приложение Б (обязательное) Диаграмма прецедентов………………….......... | | 32 |
| Приложение В (обязательное) Диаграмма последовательности......................... | | 33 |
| Приложение Г (обязательное) Диаграмма классов............................................... | | 34 |

**Введение**

Веб-сайт - это уникальный интернет-ресурс, который представляет собой свод информации о компании, организации, человеке или о чем-то другом. Веб-сайты создаются с целью предоставить пользователям информацию о продуктах или услугах, контактную информацию, новости, статьи и многое другое.

Актуальность веб-сайта заключается в том, что он является эффективным инструментом для продвижения и развития бизнеса. В настоящее время большинство людей ищут информацию в интернете, поэтому без веб-сайта компания или организация рискует потерять потенциальных клиентов. Веб-сайт является визитной карточкой бренда и позволяет привлечь новых клиентов, установить контакт с потенциальными партнерами, а также повысить узнаваемость и доверие к бренду.

Кроме того, веб-сайт позволяет эффективно информировать текущих клиентов о новых продуктах и услугах, проводимых акциях и скидках, а также дает возможность взаимодействовать с аудиторией через обратную связь и комментарии.

В современном мире, характеризующемся высокой динамикой экономической жизни и доступностью разнообразных финансовых инструментов, эффективное управление личными финансами становится все более актуальной задачей для людей всех возрастов и уровней дохода. Неспособность контролировать собственные расходы и планировать бюджет может привести к накоплению долгов, финансовой нестабильности и значительному снижению качества жизни.

Актуальность темы:

* рост финансовой доступности: Разнообразие банковских продуктов, кредитов, инвестиционных возможностей требует внимательного контроля и анализа финансовых потоков.
* необходимость планирования: Долгосрочные цели, такие как покупка недвижимости, образование или пенсия, требуют тщательного финансового планирования и отслеживания прогресса.
* повышение уровня финансовой грамотности: Все больше людей осознают важность контроля над личными финансами и ищут удобные инструменты для этого.

Традиционные методы ведения учета, такие как использование блокнотов или таблиц, также оказываются недостаточно удобными, наглядными и функциональными для полноценного анализа финансового положения. Именно поэтому создание сайта для управления личными финансами является актуальной и востребованной задачей.

Объектом является веб-сайт, профилем деятельности которого является помощь в управлении личными финансами.

Предметом является разработка автоматизированной информационной системы и базы данных сайта «KopiKopi».

Целью данной работы является создание системы для эффективного управления личными финансами.

Сайт для управления личными финансами должен решать следующие задачи:

* учет доходов и расходов;
* постановка финансовых целей;
* генерация отчетов и визуализация данных.

При создании сайта важно учитывать несколько факторов:

* целевая аудитория: необходимо определить, кому будет предназначен сайт и настроить его контент, дизайн и функционал в соответствии с интересами и потребностями пользователей.
* интуитивно понятный интерфейс: сайт должен быть понятным и легким в навигации. Важно, чтобы пользователи могли быстро и легко найти нужную информацию или функции.
* дизайн и визуальная привлекательность: сайт должен иметь привлекательный внешний вид, соответствующий бренду или целей сайта. Графика, шрифты, цветовая схема и общая композиция должны создавать приятное впечатление.
* безопасность: сайт должен быть защищен от потенциальных угроз, таких как хакерские атаки или вредоносные программы. Важно регулярно обновлять и проверять безопасность сайта.

Расширенный функционал основан на различных дополнениях, которые расширяют базовые возможности сайта и CMS. Изначально созданная система может быть ограничена в своих возможностях или уже устареть, поэтому необходимо регулярно обновлять и расширять функциональность системы.

Процесс формирования предложений о расширении начинается с анализа текущих потребностей пользователей и бизнес-задач, которые должны быть решены информационной системой. После этого идет изучение существующих решений на рынке, а также определение возможностей и ограничений текущей системы.

Сформированные предложения о расширении функциональности информационной системы:

* настраиваемые категории: разрешить пользователям создавать и редактировать собственные пользовательские категории и подкатегории, чтобы лучше отражать их потребительские привычки;
* анализ тенденций расходов: предоставлять углубленный анализ моделей расходов пользователя, выявляя тенденции и отклонения;
* мобильное приложение: разработать мобильное приложение для удобного доступа и отслеживания на ходу;
* составление бюджета: помогать пользователю создавать и отслеживать бюджет, предупреждая о превышении запланированных лимитов;
* учетные записи пользователей и совместная работа: разрешить нескольким пользователям получать доступ и управлять общим счетом (например, для пар или семей);
* сообщество или поддержка: создать форум сообщества, где пользователи могут задавать вопросы и делиться советами.

Созданный сайт для управления личными финансами позволит достичь следующих результатов:

* улучшение финансовой организации и осведомленности: получение полного обзора доходов и расходов, устранение финансовой неопределенности;
* организованный учет транзакций: простое добавление, категоризация и поиск всех финансовых операций;
* целевое накопление: постановка конкретных финансовых целей и отслеживание прогресса;
* улучшение финансовых привычек: развитие лучшей финансовой дисциплины благодаря систематическому отслеживанию и анализу;
* снижение финансового беспокойства: получение контроля над финансами, снижение стресса, связанного с управлением деньгами;
* повышение финансовой грамотности: обучение и улучшение понимания финансов благодаря инструментам и ресурсам системы.

Результатом данного проекта является разработка информационной системы, дизайна и информационное наполнение web-сайта, который предоставляет возможности управления личными финансами.

**1 Анализ предметной области**

Предметная область сайта для управления личными финансами охватывает широкий спектр вопросов и задач, связанных с планированием и контролем финансовых ресурсов пользователя.

Основные аспекты, которые могут быть включены в эту область:

* отслеживание расходов: возможность фиксировать и классифицировать ежедневные, ежемесячные и ежегодные расходы, система может автоматически группировать транзакции по категориям;
* доходы и инвестиции: раздел для учета источников доходов, сравнение доходов и расходов для понимания общей финансовой ситуации;
* финансовая аналитика и отчеты: генерация различных отчетов и графиков для визуализации финансовых данных, что поможет пользователям лучше понять свои расходы и доходы;
* цели и планирование: установка финансовых целей, таких как накопление на поездку, покупку недвижимости или создание резервного фонда.

Сайт для управления личными финансами должен быть удобным, интуитивно понятным и обеспечивать пользователям полезные инструменты для достижения их финансовых целей.

Целевая аудитория сайта для управления личными финансами достаточно широка:

* молодые люди (студенты, молодые специалисты): нуждаются в базовых функциях контроля расходов и планирования бюджета;
* люди среднего возраста (семьи, работающие специалисты): требуется более широкий функционал, включающий управление несколькими счетами, планирование долгосрочных целей (образование детей, пенсия);
* люди, стремящиеся к финансовой стабильности: основной фокус на контроле расходов и планировании бюджета;
* люди, планирующие крупные покупки: нуждаются в инструментах для планирования сбережений и отслеживания прогресса.
* люди с низким уровнем финансовой грамотности: необходим простой и понятный интерфейс, инструкции и подсказки.
* людям, стремящимся улучшить свои финансовые навыки и достичь финансовых целей.

При разработке сайта следует учитывать потребности различных сегментов целевой аудитории, предлагая различные уровни функциональности и удобства использования.

Использование сайта для управления личными финансами имеет множество преимуществ:

* улучшение финансовой организации и контроля: сайт предоставляет единое место для отслеживания всех доходов и расходов, создавая целостное представление о финансовом положении;
* простой и удобный учет: ввод данных происходит в удобном интерфейсе, избавляя от необходимости использовать громоздкие таблицы или блокноты;
* категоризация и анализ расходов: вистема позволяет группировать транзакции по категориям, что помогает выявлять основные статьи расходов и контролировать их;
* постановка и отслеживание целей: система позволяет ставить финансовые цели и отслеживать прогресс их достижения, что повышает мотивацию и помогает двигаться к намеченным финансовым целям;
* улучшение финансовых привычек: систематическое отслеживание финансов способствует формированию более ответственного отношения к деньгам и развитию хороших финансовых привычек;
* уменьшение финансового беспокойства: знание своего финансового положения и контроль над расходами снижают уровень стресса и беспокойства, связанных с деньгами.

Несмотря на многочисленные преимущества, использование сайта для управления личными финансами имеет и некоторые недостатки:

* необходимость ручного ввода данных: ручной ввод транзакций занимает достаточное количества времени, а также повышает риск ошибок;
* доверие к стороннему сервису: пользователи должны доверять стороннему сервису хранение своих конфиденциальных финансовых данных.
* зависимость от корректности данных: точность анализа и планирования напрямую зависит от правильности введенных данных.

Важно учитывать как плюсы, так и минусы при создании и продвижении сайта.

Анализ основных конкурентов в области сайтов для управления личными финансами позволяет лучше понять рынок, выявить сильные и слабые стороны конкурентов, а также определить возможности для улучшения собственного продукта.

Вот несколько ключевых конкурентов и аспекты их анализа:

Mint - один из самых популярных инструментов для управления личными финансами, предлагающий учет доходов и расходов, создание бюджета и отслеживание инвестиций.

Сильные стороны:

* бесплатный доступ с обширными функциями;
* автоматическая синхронизация с банковскими счетами;
* удобный интерфейс и визуализация данных.

Слабые стороны:

* ограниченные функции для инвесторов;
* некоторые пользователи сообщают о проблемах с безопасностью данных.

YNAB (You Need A Budget) - платный сервис, который фокусируется на методах активного бюджетирования и управлении деньгами.

Сильные стороны:

* обучающие материалы и поддержка пользователей;
* эффективная методология планирования бюджета;
* сообщество пользователей для обмена опытом.

Слабые стороны:

* платная подписка может отпугнуть некоторых пользователей;
* требует времени на привыкание к системе.

Personal Capital - платформа для управления личными финансами и инвестициями, предлагающая как бесплатные, так и платные услуги.

Сильные стороны:

* инструменты для отслеживания инвестиций и пенсионного планирования;
* бесплатные финансовые отчеты и анализ портфеля.

Слабые стороны:

* платные услуги могут быть дорогими для некоторых пользователей;
* интерфейс может быть сложным для новичков.

При анализе конкурентов важно учитывать их сильные и слабые стороны, а также уникальные предложения. Это поможет определить, какие функции и услуги можно улучшить или внедрить в вашем продукте. Также стоит обратить внимание на отзывы пользователей, чтобы понять, что именно они ценят в конкурентных сервисах и какие проблемы остаются нерешенными. Это может стать основой для разработки уникального предложения на рынке управления личными финансами.

Сайт для управления личными финансами представляет собой комплексное решение, которое помогает пользователям организовать свои финансы, повысить финансовую грамотность и достичь своих финансовых целей. Важным аспектом является удобство использования и доступность информации, что позволяет пользователям легко ориентироваться в своих финансах и принимать обоснованные решения.

**2 Проектирования приложения**

Заказчик: Адамович Н.В.

Исполнитель: Чурсина В.И.

Сроки выполнения: 15 декабря - 25 января

Основные процессы области:

* регистрация пользователей: создание учетных записей с хранением паролей;
* ввод транзакций: добавление доходов и расходов с указанием суммы и категории;
* категоризация транзакций: ручная или автоматическая категоризация транзакций для анализа расходов;
* управление счетами: добавление, редактирование и удаление банковских счетов и других финансовых источников. отслеживание баланса на каждом счете;
* управление категориями: создание, редактирование и удаление категорий расходов и доходов;
* анализ расходов и доходов: генерация отчетов по категориям за выбранный период, визуализация данных в виде графиков и диаграмм;
* постановка финансовых целей: определение долгосрочных и краткосрочных финансовых целей, мониторинг их достижения.

Основные сущности - это фундаментальные структуры данных, представляющие основную функциональность.

* пользователь: представляет зарегистрированного пользователя системы;
* счет: представляет финансовый счет;
* транзакция: представляет собой отдельную финансовую операцию;
* категория: представляет категорию для классификации транзакций;
* цель: представляет финансовую цель, которую пользователь хочет достичь.

Объектом является веб-сайт, профилем деятельности которого является помощь в управлении личными финансами.

Предметом является разработка автоматизированной информационной системы и базы данных сайта «KopiKopi».

Целью данной работы является создание системы для эффективного управления личными финансами.

Сайт для управления личными финансами должен решать следующие задачи:

* учет доходов и расходов: предоставлять пользователю удобный интерфейс для регистрации всех финансовых операций, включая доходы и расходы, с возможностью категоризации и пометками;
* постановка финансовых целей: позволять пользователю определять краткосрочные и долгосрочные финансовые цели (например, накопление на отпуск, покупка автомобиля, инвестирование) и отслеживать прогресс их достижения;
* генерация отчетов и визуализация данных: предоставлять наглядные отчеты и графики, отображающие финансовое положение пользователя, позволяющие анализировать расходы и выявлять области для оптимизации;
* современный дизайн: дизайн должен быть выполнен по современным тенденциям и быть приятным для пользователя;
* удобство использования: обеспечить интуитивно понятный и простой в использовании интерфейс, доступный пользователям с различным уровнем технической подготовки.

Функциональные требования к веб-сайту:

* управление пользователями: регистрация/авторизация пользователей с сохранением данных;
* управление транзакциями: возможность добавления транзакций и ее редактирования;
* постановка финансовых целей: возможность постановки финансовых целей и отслеживания прогресса;
* отчетность и анализ: возможность генерации различных отчетов за указанные периоды, представление данных в виде диаграмм ;
* управление счетами: возможность добавления и управления несколькими счетами, а также отслеживание их баланса.

Нефункциональные требования к веб-сайту:

* отзывчивость - сайт должен корректно отображаться на всех устройствах, включая мобильные телефоны и планшеты;
* быстродействие - сайт должен загружаться быстро, чтобы обеспечить удобство пользователей;
* безопасность - сайт должен быть защищен от взломов и кибератак;
* легкость навигации - сайт должен быть интуитивно понятным и удобным в использовании;
* совместимость с браузерами - сайт должен работать корректно в различных браузерах, таких как Chrome, Firefox, Safari и других;
* SEO-оптимизация - сайт должен быть оптимизирован для поисковых систем, чтобы привлечь больше посетителей.

Веб-сайт для управления личными финансами состоит из следующих страниц: главная страница сайта, страница регистрации/авторизации, главная страница пользователя, страница операций, страница целей и страница анализа.

Главная страница сайта:

* блок с кратким описанием назначения системы;
* блок с описанием всех возможностей платформы;
* блок с предложением создания профиля.

Страница регистрации:

* форма авторизации с возможностью перейти на форму регистрации.

Главная страница пользователя:

* блок со всеми счетами пользователя;
* блок с формой для добавления транзакции;
* блок с краткой историей транзакций за последние несколько дней.

Страница операция:

* отображение всех транзакций пользователя с возможностью поиска вручную или поиска по фильтрам.

Страница целей:

* блок с формой для добавления цели;
* список всех созданных целей пользователя.

Страница анализа:

* Различные диаграммы, отражающие данные транзакций пользователей.

Требования к техническому обеспечению:

* сеть с подключение к Интернет процессор Intel Xeon E3 шестого поколение или новее, или AMD Ryzen 1950x или новее;
* 32 Gb и более оперативной памяти;
* 2 Tb – жесткий диск SSD или HDD с поддержкой SATA 3;
* монитор – FullHD или 4K;
* клавиатура;
* манипулятор типа «мышь».

Требования, предъявляемые к конфигурации клиентских станций:

* процессор, с тактовой частотой не менее 2,4 GHz, не менее 2 ядер;
* 8 Gb оперативной памяти;
* монитор – FullHD;
* клавиатура;
* манипулятор типа «мышь»;
* доступ к подключению Интернет со скоростью не менее 100 МБ/с.

Требования к программному обеспечению:

* операционная система: Windows версии не ниже 8.1;
* доступ к широполосному подключению к сети Интернет;
* браузер для просмотра веб-страниц последней версии.

Общие требования к контенту веб-сайта:

* высококачественные изображения и видео;
* уникальный и информативный контент;
* правильная грамматика и орфография;
* оптимизация для поисковых систем (SEO);
* регулярное обновление для обеспечения актуальности и новизны.

Требования к эксплуатации:

* техническое обслуживание и устранение неполадок: оперативное устранение любых технических проблем, обеспечение безопасности и производительности сайта;

Требования к сопровождению:

* регулярные обновления программного обеспечения: обновление программного обеспечения сайта для обеспечения безопасности, производительности и функциональности;
* усовершенствование функциональности: изучение новых тенденций и технологий веб-разработки и внедрение улучшений для повышения пользовательского опыта и функциональности сайта;
* улучшение производительности: отслеживание производительности сайта и внедрение оптимизации для повышения скорости загрузки страниц и общей отзывчивости;
* адаптация к новым устройствам и технологиям: проверка совместимости сайта с новыми устройствами и технологиями, такими как мобильные устройства и голосовые помощники;
* постоянная поддержка и обслуживание: обеспечение постоянной поддержки и обслуживания сайта, чтобы он был постоянно доступным и работал без сбоев.

**3 Разработка программного обеспечения**

**3.1 Описание технологического стека разработки**

При разработке данной системы использовались такие приложения, как Figma, VS Code, SQLite.

Figma - это онлайн-инструмент для дизайна интерфейсов и прототипирования. Он позволяет дизайнерам создавать и редактировать векторные графические элементы, делиться ими с другими пользователями и просматривать изменения в реальном времени.

Преимущества Figma:

* кроссплатформенность: Figma доступен как на Windows, Mac, Linux, а также имеет мобильное приложение для iOS и Android, что позволяет работать на любом устройстве;
* совместная работа: несколько пользователей могут одновременно работать над одним проектом, делиться мнениями и комментариями, что улучшает процесс командной работы;
* облачное хранение: все проекты сохраняются в облаке, что обеспечивает доступ к ним с любого устройства и в любое время;
* простота использования: удобный и интуитивно понятный интерфейс, облегчающий процесс дизайна и прототипирования;
* возможность создания интерактивных прототипов: Figma позволяет создавать прототипы с возможностью анимации и интерактивных элементов, что помогает лучше визуализировать идеи;
* интеграция с другими инструментами: Figma поддерживает интеграцию с различными сервисами и инструментами, что упрощает процесс работы и обмена данными.

VS Code - это бесплатный редактор кода, разработанный компанией Microsoft. Он является одним из самых популярных инструментов для разработки программного обеспечения, так как обладает множеством функций и возможностей.

Преимущества VS Code:

* поддержка множества языков программирования: VS Code поддерживает большое количество языков программирования, что делает его отличным выбором для разработчиков, работающих с разными технологиями;
* наличие расширений и плагинов: VS Code имеет широкий выбор расширений и плагинов, которые добавляют новые функциональности и возможности к редактору кода;
* интеграция с Git: VS Code имеет встроенную поддержку Git, что позволяет разработчикам удобно работать с версионным контролем и управлять своими проектами;
* легкость использования: VS Code имеет интуитивный интерфейс и простые настройки, что делает его простым в использовании даже для новичков в программировании;
* поддержка платформ: VS Code доступен для Windows, MacOS и Linux, что делает его удобным выбором для разработчиков, работающих на разных операционных системах.

SQLite — это бесплатная, встраиваемая, самодостаточная, серверная база данных SQL, хранящаяся в одном файле. Это означает, что она не требует отдельного сервера баз данных (например, MySQL или PostgreSQL) для работы. База данных хранится непосредственно в одном файле на диске, к которому приложение обращается напрямую.

Преимущества SQLite:

* простота в использовании и развертывании: SQLite очень прост в использовании и интеграции в приложения. Он не требует установки и настройки отдельного сервера баз данных, что значительно упрощает разработку и развертывание. Файл базы данных можно просто скопировать вместе с приложением;
* встраиваемость: SQLite может быть встроен непосредственно в приложение, что делает его идеальным выбором для мобильных приложений, веб-приложений (с использованием Node.js, Python и других технологий), и других систем, где установка отдельного сервера баз данных нежелательна или невозможна;
* самодостаточность: База данных SQLite полностью автономна и не зависит от внешних серверов. Это обеспечивает высокую надежность и доступность, поскольку отказ сервера базы данных не повлияет на работу приложения;
* кроссплатформенность: SQLite работает на всех основных операционных системах (Windows, macOS, Linux, Android, iOS) и поддерживает большинство языков программирования;
* надежность: SQLite имеет высокую надежность и обеспечивает ACID-свойства (атомарность, согласованность, изолированность, долговечность) транзакций, гарантируя целостность данных;
* легковесность: SQLite относительно небольшой по размеру, что делает его идеальным для использования в приложениях с ограниченными ресурсами;
* бесплатное использование: SQLite распространяется под лицензией Public Domain, что означает его бесплатное использование и модификацию без каких-либо ограничений;
* поддержка SQL: SQLite поддерживает стандартный язык SQL, что упрощает разработку и делает его знакомым для разработчиков, имеющих опыт работы с другими реляционными базами данных.

HTML (HyperText Markup Language) - это язык разметки, используемый для создания веб-страниц. Он описывает структуру и содержание веб-страницы с помощью различных тегов и атрибутов.

Преимущества HTML включают:

* простота использования: HTML - достаточно простой язык, который легко изучить и понять для создания базовых веб-страниц;
* кроссплатформенность: HTML поддерживается всеми браузерами и операционными системами, что позволяет создавать универсальные веб-сайты;
* SEO-оптимизация: HTML позволяет создавать легко читаемые веб-страницы для поисковых систем, что способствует улучшению позиций сайта в выдаче результатов поиска;
* возможность создания динамических сайтов: HTML может использоваться в сочетании с CSS и JavaScript для создания интерактивных и динамических веб-страниц;
* совместимость с другими технологиями: HTML легко интегрируется с различными технологиями и инструментами для создания сложных веб-приложений.

В целом, HTML является основой для построения веб-страниц и веб-приложений, что делает его необходимым языком для всех веб-разработчиков.

CSS (Cascading Style Sheets) - это язык, который используется для оформления и стилизации веб-страниц. Он позволяет разработчикам управлять внешним видом веб-страницы, определяя цвета, шрифты, размеры, отступы и многое другое.

Преимущества использования CSS включают:

* раздельное оформление: CSS позволяет разделить содержимое и его оформление, что делает код более чистым и упорядоченным;
* легкость внесения изменений: Изменения стилей в CSS применяются ко всем элементам на странице сразу, что облегчает процесс обновления дизайна;
* кроссбраузерная совместимость: CSS позволяет создавать универсальные стили, которые одинаково хорошо отображаются на различных браузерах;
* улучшенная производительность: Поскольку CSS позволяет отделить структуру от визуального оформления, загрузка страниц происходит быстрее и производительность сайта улучшается;
* адаптивный дизайн: С помощью CSS можно создавать адаптивные и отзывчивые дизайны, которые корректно отображаются на различных устройствах и разрешениях экрана.

JavaScript - это высокоуровневый, интерпретируемый язык программирования, который является одним из основных языков веб-разработки. Он используется преимущественно для создания интерактивных элементов на веб-страницах, но его применение расширилось далеко за пределы веб-браузеров.

Основные преимущества JavaScript:

* универсальность: JavaScript — это не просто язык для веб-браузеров. Он используется в cоздание интерактивных элементов, обработкe событий, динамическом изменение содержимого страниц, анимации. С помощью Node.js JavaScript используется для создания серверных приложений, API и микросервисов;
* простота изучения: JavaScript имеет относительно простой синтаксис, что делает его сравнительно легким для изучения, особенно для начинающих программистов. Он обладает гибким динамическим типированием, что упрощает написание кода;
* широкое сообщество и поддержка: JavaScript имеет огромное и активное сообщество разработчиков, что обеспечивает богатый выбор библиотек, фреймворков и инструментов, а также множество онлайн-ресурсов для обучения и поддержки.
* постоянное развитие: Язык JavaScript постоянно развивается и совершенствуется, появляются новые стандарты и функции, что расширяет его возможности и повышает производительность.
* быстрый результат: Благодаря интерпретируемой природе JavaScript, можно быстро увидеть результат своей работы, что ускоряет процесс разработки.
* обширная экосистема: Множество библиотек и фреймворков позволяют решать самые разнообразные задачи быстро и эффективно.
* большое количество вакансий: Благодаря широкому применению JavaScript, спрос на разработчиков, владеющих этим языком, очень высок.

**3.2 Описание алгоритма работы**

Взаимодействие пользователя:

* аутентификация: Пользователь входит в систему, используя свои учетные данные (имя пользователя/адрес электронной почты и пароль). Система проверяет учетные данные, сравнивая их с хранящимися хешированными паролями и механизмами управления сессиями;
* ввод данных: Пользователь взаимодействует с интерфейсом для ввода транзакций. Это включает выбор счета, автоматическая постановка даты, суммы, категории;
* постановка целей: Пользователь определяет финансовые цели, указывая целевые суммы и сроки;
* генерация отчетов: Пользователь выбирает тип транзакции (доход или расход) диапазон дат, категории. Система генерирует отчет в виде диаграмм;
* управление счетами: Пользователи добавляют, редактируют или удаляют счета. Система обновляет остатки на счетах соответственно;

Системные процессы:

* проверка данных: При вводе данных система проверяет введенную информацию для обеспечения целостности данных;
* хранение данных: Проверенные данные надежно хранятся в базе данных;
* категоризация транзакций: При ручном вводе пользователь выбирает категорию;
* расчет баланса: Система автоматически рассчитывает и обновляет остатки на счетах на основе транзакций;
* отслеживание прогресса к цели: Система отслеживает прогресс в достижении финансовых целей и предоставляет визуальное представление прогресса;
* генерация отчетов: Система извлекает данные из базы данных, выполняет вычисления (например, суммы, средние значения) и форматирует данные в отчет.

В приложении В представлена диаграмма последовательности всех действий пользователя на сайте. Диаграмма последовательности - это визуальное представление взаимодействия между объектами в рамках определенной задачи или сценария использования. Она показывает последовательность сообщений, которые передаются между объектами или компонентами системы в определенном порядке времени.

Диаграмма последовательности включает в себя вертикальные линии, представляющие различные объекты или компоненты, а также стрелки, обозначающие передачу сообщений между ними. Порядок сообщений и временная линия помогают понять, как объекты взаимодействуют друг с другом и как изменяется состояние системы в процессе выполнения конкретного сценария.

Диаграммы последовательности часто используются в процессе анализа и проектирования системы для уточнения её функциональности и взаимодействия между компонентами. Они также могут использоваться для документирования поведения системы или как инструмент коммуникации между участниками разработки.

База данных - коллекция структурированных данных, которая хранится и обрабатывается с помощью специального программного обеспечения. Она позволяет эффективно организовывать, хранить, обновлять и анализировать информацию для различных целей, таких как управление предприятием, хранение информации о клиентах, анализ данных и многое другое.

Логика в базе данных определяется структурой данных и правилами их обработки. Взаимодействие элементов в базе данных происходит через запросы, которые позволяют получать, изменять и удалять информацию. Основные элементы базы данных, которые взаимодействуют между собой, включают таблицы (сущности), поля (атрибуты) и связи между ними.

Для взаимодействия с элементами базы данных используются SQL запросы, которые позволяют выполнять различные операции, такие как выборка данных (SELECT), вставка данных (INSERT), обновление данных (UPDATE) и удаление данных (DELETE). Логика базы данных определяет, какие данные доступны для использования и какие операции можно с ними выполнять в соответствии с правилами и ограничениями, установленными в базе данных.

В приложении Г представлена диаграмма классов. Диаграмма классов - это графическое представление структуры классов и их взаимосвязей в объектно-ориентированном программировании. В диаграмме классов классы обычно представлены в виде прямоугольников с указанием их имени, атрибутов и методов.

Отношения между классами обычно обозначаются стрелками, указывающими направления связей.

Диаграмма классов помогает разработчикам лучше понять структуру программы, увидеть связи между классами, определить иерархию классов и улучшить процесс проектирования ПО.

С помощью данных диаграмм наглядно показано взаимодействия пользователя и системы.

SQLite — это встраиваемая база данных, поэтому подключение к ней не предполагает отдельного сервера. Вместо этого, приложение напрямую обращается к файлу базы данных на файловой системе. "Подключение" заключается в открытии этого файла и выполнении операций чтения/записи. Для этого используется соответствующая библиотека для вашего языка программирования (sqlite3 в Node.js). В коде указывается путь к файлу базы данных, после чего создается объект соединения, с помощью которого выполняются SQL-запросы.

На рисунке 1 представлен код подключения бд.

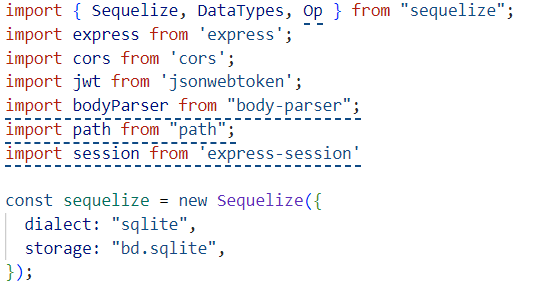


Рисунок 1 - Подключение бд

На рисунке 2 представлен скрипт для обработки формы авторизации.



Рисунок 2 - Скрипт для обработки формы авторизации

На рисунке 3 представлен скрипт для обработки формы регистрации.



Рисунок 3 - Скрипт для обработки формы регистрации

На рисунке 4 представлен скрипт для обработки формы добавления транзакций.



Рисунок 4 - Скрипт для обработки формы добавления транзакций.

На рисунке 5 представлен маршрут, возвращающий все транзакции, хранящиеся в базе данных.



Рисунок 5 - Маршрут, возвращающий все транзакции, хранящиеся в базе данных

**3.3 Описание интерфейса пользователя**

Интерфейс пользователя - способ взаимодействия человека с компьютерной программой или устройством. Это набор элементов, которые пользователь может использовать для управления программой или получения информации. Интерфейс пользователя включает в себя графические элементы, кнопки, меню, поля для ввода текста и другие функциональные элементы, которые помогают пользователю взаимодействовать с программой или устройством.

Существует несколько основных приемов реализации интерфейса пользователя, которые помогают сделать его удобным, понятным и эффективным. Основные приемы, использованные при создании данного веб-сайта:

* использование простоты и понятности: интерфейс должен быть легким для понимания и использования даже для новых пользователей. Использование понятных и интуитивно понятных элементов управления;
* соблюдение консистентности: элементы интерфейса должны быть организованы и отображены в логической и последовательной последовательности. Кнопки, меню и другие элементы должны быть размещены одинаково на всех экранах и в различных частях программы;
* понимание и контекст: пользователь всегда понимает, где находится и что происходит. Подсветка активных элементов и предоставление информации о текущем состоянии программы.

Интерфейс пользователя (UI) успешного сайта для управления личными финансами должен быть интуитивным, удобным и визуально привлекательным, одновременно обеспечивая безопасность данных. Рассмотрим основные элементы страниц сайта:

Главная страница пользователям - отражает краткое, но информативное представление о текущем финансовом состоянии :

* список счетов: список всех добавленных счетов с текущими балансами, а также возможность добавления новых.
* форма добавление транзакций: простая форма для ввода новой транзакции с полями для: суммы, категории (выбор из списка), выбор счета, выбор типа транзакции (доход или расход);
* последние транзакции: список последних финансовых операций с кратким описанием и суммой.

Транзакции:

* таблица со всеми транзакциями, сортируемая по различным параметрам (дата, сумма, категория, счет). Возможность поиска по ключевым словам в описании.

Цели:

* создание целей: функционал для добавления новых финансовых целей. Указание целевой суммы, срока и названия цели.
* список всех целей: отображение всех целей с датой окончания и собранной суммой.

Анализ - различные виды отчетов, такие как:

* диаграмма со сравнением доходов и расходов;
* диаграмма со сравнением категорий расходов/доходов;
* диаграмма с динамикой доходов/расходов за выбранные период.

Общие требования к интерфейсу:

* безопасность: защита данных пользователей с использованием шифрования и надежных методов аутентификации;
* интуитивность: простой и понятный интерфейс, легкий в освоении;
* адаптивность: корректное отображение на различных устройствах;
* современный дизайн: визуально привлекательный и современный дизайн.

Создание удобного и функционального интерфейса пользователя для сайта будет способствовать повышению удовлетворенности пользователей, увеличению количества пользователей данным сайтом.

**4 Тестирование приложения**

**4.1 План тестирования**

Тестирование - это процесс проверки программного продукта с целью обнаружения ошибок, дефектов или недочетов, которые могут повлиять на корректность работы системы.

Цель тестирования - убедиться в том, что программа соответствует заявленным требованиям к функциональности, надежности, производительности и другим аспектам качества. В процессе тестирования специально разрабатываются тестовые случаи и сценарии, которые позволяют проверить правильность работы отдельных компонентов системы, их взаимодействие между собой, а также оценить общую работоспособность продукта.

Тестирование может проводиться как вручную, когда тестировщик самостоятельно выполняет тестовые сценарии, так и автоматизированно с использованием специальных инструментов и программ.

План тестирования - это документ, в котором описаны цели, стратегии, методы и процедуры тестирования программного продукта. Он представляет собой план действий по проверке и оценке функциональности, надежности, производительности и других аспектов программного продукта.

План тестирования необходим для определения всего спектра тестов, которые будут проводиться для уверенности в качестве и корректности работы продукта перед его выпуском на рынок. Этот документ помогает организовать тестирование, ресурсы для проведения тестирования, управлять процессом тестирования и оценить его результаты.

План тестирования также помогает минимизировать риски, связанные с недостаточностью тестирования программного продукта, что может привести к ошибкам и проблемам в работе системы после ее внедрения.

План тестирования сайта обычно включает в себя следующие этапы:

1. Планирование тестирования - определение целей и задач тестирования, составление плана действий и распределение обязанностей между участниками команды.
2. Анализ требований - изучение требований к сайту и функционалу, определение критериев успешного тестирования.
3. Создание тест-кейсов - разработка набора тест-случаев для проверки функционала сайта, включающих в себя различные сценарии использования.
4. Исполнение тест-кейсов - проведение тестирования сайта в соответствии с разработанными тест-кейсами, выявление ошибок и несоответствий требованиям.
5. Отчетность - подготовка отчетов о результатах тестирования, документирование всех выявленных проблем и ошибок, их приоритизация и уточнение.
6. Устранение ошибок - передача найденных ошибок разработчикам для их исправления, повторное тестирование после устранения проблем.
7. Завершение тестирования - подготовка финального отчета о проведенном тестировании и его утверждение со стороны заказчика или куратора проекта.

Это основные этапы, которые могут включаться в план тестирования сайта.

Unit тесты - это тесты, которые проверяют работу отдельных компонентов программы (обычно функций или методов) без их зависимостей. Они помогают обнаружить ошибки в ранних стадиях разработки и уверенно делать изменения в коде, не боясь, что это повлияет на другие части программы.

Тест-кейсы - это сценарии, описывающие как нужно провести тестирование определенной логики работы программы. Включают в себя шаги, ожидаемые результаты и критерии прохождения теста. Тест-кейсы являются основой для разработки и проведения тестирования на различных уровнях.

Порядок интеграционного тестирования включает в себя последовательное объединение отдельных компонентов программы и проверку их взаимодействия. Обычно начинают с тестирования интеграции небольших компонентов, затем постепенно увеличивают сложность, включая в тестирование все больше компонентов программы.

При интеграционном тестировании важно убедиться, что все компоненты программы взаимодействуют корректно и выполняют свои функции в соответствии с требованиями. В конечном итоге, успешное интеграционное тестирование дает уверенность в работоспособности всей системы в целом.

**4.2 Оценка результатов проведения тестирования**

Оценка результатов проведения тестирования – важный этап в жизненном цикле создания программного обеспечения. После завершения тестирования необходимо анализировать полученные результаты, чтобы определить степень готовности продукта к выпуску.

Оценка результатов проведения тестирования — это важный этап в любом процессе тестирования, будь то программное обеспечение, системы или какие-либо другие продукты. Она позволяет анализировать, насколько хорошо продукт соответствует заявленным требованиям и стандартам качества.

Ключевые аспекты, которые следует учитывать при оценке результатов тестирования:

Соответствие требованиям:

* проверка того, насколько результаты тестов соответствуют предварительно определённым требованиям и спецификациям;
* оценка количества найденных дефектов и их серьёзности.

Покрытие тестами:

* анализ объема кода или функциональных возможностей, которые были протестированы;
* определение уровня покрытия тестами (например, покрытие кода, покрытие требований).

Качество продукта:

* оценка стабильности, производительности и безопасности системы;
* использование метрик, таких как количество критических и некритических дефектов на тысячу строк кода.

Анализ ошибок:

* классификация найденных ошибок по типам и частоте возникновения;
* оценка причин возникновения ошибок и их влияния на пользователей.

Сравнение с предыдущими версиями:

* если это не первая версия продукта, следует сравнить результаты тестирования с предыдущими версиями, чтобы определить динамику улучшений или ухудшений.

Пользовательский опыт:

* сбор отзывов от конечных пользователей о работе системы или продукта после тестирования;
* оценка удобства использования и соответствия ожиданиям пользователей.

Рекомендации и улучшения:

* формулирование рекомендаций по исправлению обнаруженных дефектов;
* предложение по улучшению процесса тестирования на основе выявленных проблем и их анализа.

Документация и отчетность:

* подготовка отчетов о тестировании, включающих все полученные данные, метрики, и результаты анализа;
* обеспечение доступности результатов для всех заинтересованных сторон.

Тестирование функции регистрации пользователя на сайт.

Таблица 1 - Корректное заполнение всех полей формы

|  |  |  |  |
| --- | --- | --- | --- |
| №1 | Оформление индивидуального заказа | | |
| Ссылка на ТЗ | Чурсина В. | Высокий | Заказ |
| 1.http://localhost:5173/login | | 1. Вводим корректные данные в поле «Ваше имя»  2.Вводим корректные данные в поле «Ваш E-mail»  3. Вводим данные в поле «Ваш пароль»  4. Пользователь зарегистрирован, ему открылась главная страница. | |
| Пользователь смог зарегистрироваться. | | | |

На рисунке 6 представлено верное заполнение формы регистрации.

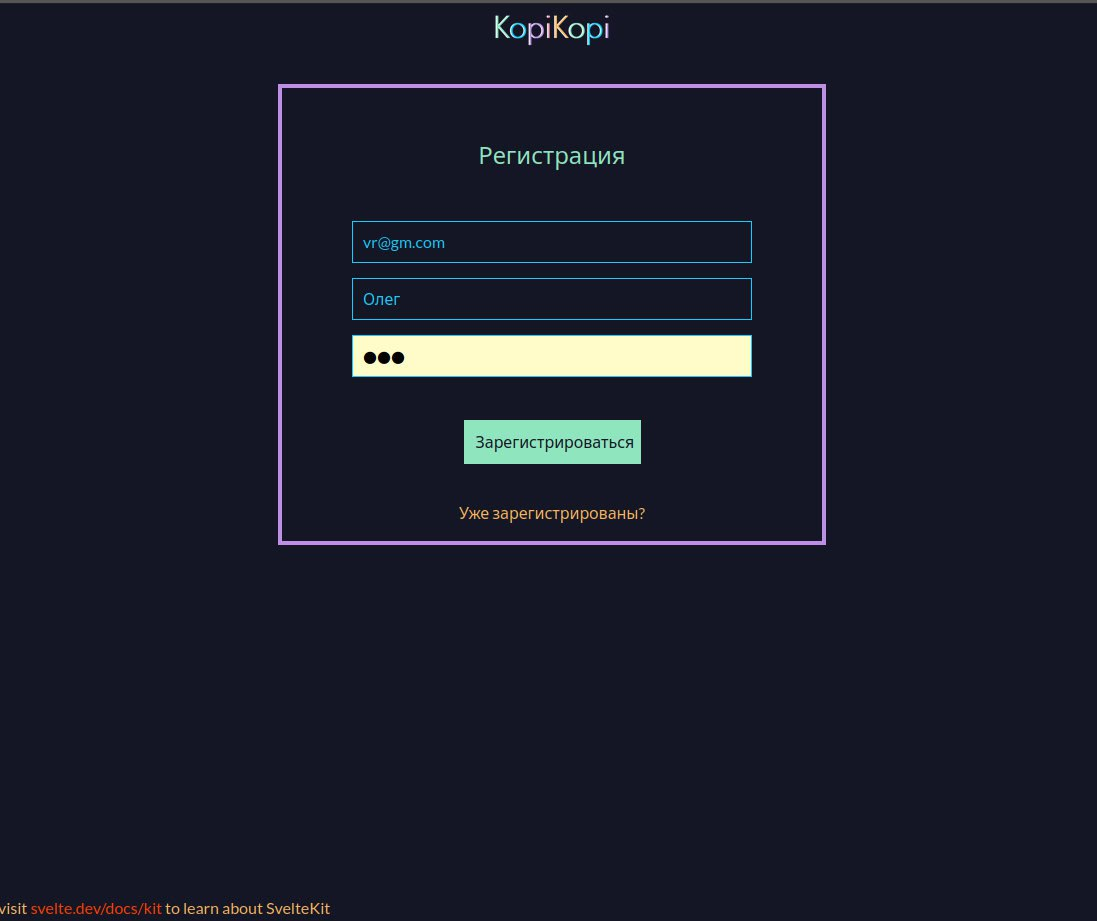


Рисунок 6 - Верное заполнение формы регистрации

Таблица 2 - Некорректное заполнения поля «Ваш E-mail»

|  |  |  |  |
| --- | --- | --- | --- |
| №2 | Оформление индивидуального заказа | | |
| Ссылка на ТЗ | Чурсина В. | Высокий | Заказ |
| 1.http://localhost:5173/login | | 1. Вводим корректные данные в поле «Ваше имя»  2.Вводим некорректные данные в поле «Ваш E-mail»  3. Вводим данные в поле «Ваш пароль»  4. Пользователь не зарегистрирован, над полем «Ваш E-mail» появилось сообщение об ошибке. | |
| Пользователь не смог зарегистрироваться, так как ввел некорректный E-mail. | | | |

На рисунке 7 представлен результат неверного заполнения поля «Ваш E-mail».

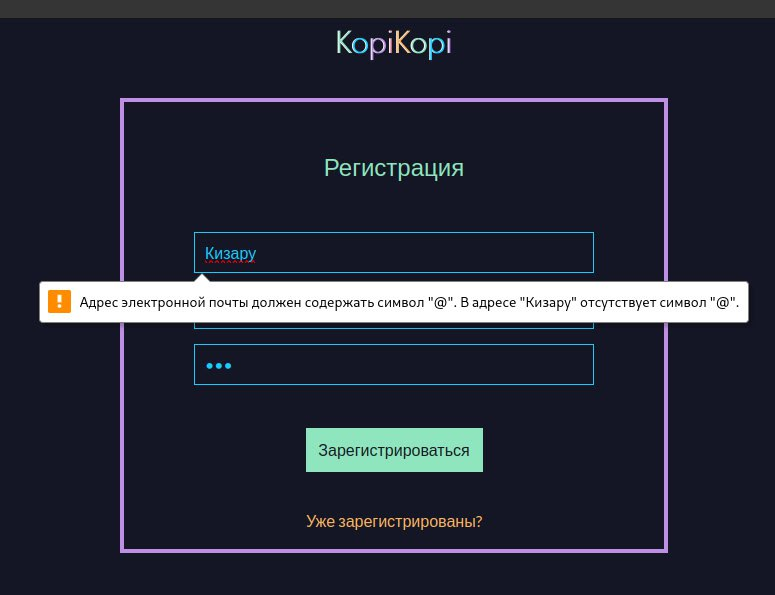


Рисунок 7 - Результат неверного заполнения поля «Ваш E-mail»

Таблица 3 - Авторизация незарегистрированного пользователя.

|  |  |  |  |
| --- | --- | --- | --- |
| №2 | Оформление индивидуального заказа | | |
| Ссылка на ТЗ | Чурсина В. | Высокий | Заказ |
| 1.http://localhost:5173/login | | 1.Вводим корректные данные в поле «Ваш E-mail»  2. Вводим данные в поле «Ваш пароль»  3. Пользователь не авторизован, появилась ошибка «Пользователь не существует». | |
| Пользователь не смог авторизоваться, так как ввел E-mail не зарегистрированный в системе. | | | |

На рисунке 8 представлен результат авторизации незарегистрированного пользователя.

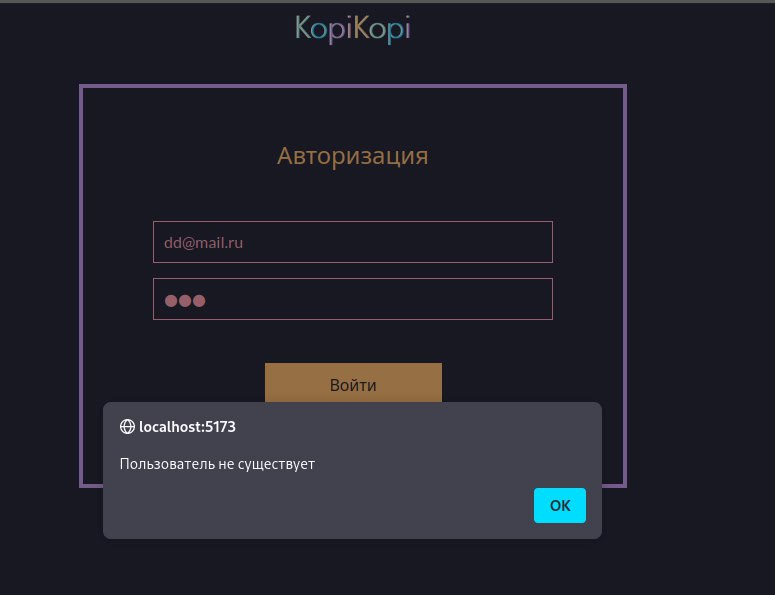


Рисунок 8 - Результат авторизации незарегистрированного пользователя

**Заключение**

Сайт - это набор веб-страниц, доступных в Интернете, которые объединены под единым доменным именем. Сайты могут выполнять различные функции и иметь разнообразные цели в зависимости от их назначения.

Для чего нужны сайты:

* информация: сайты могут использоваться для предоставления информации о компании, продукте или услуге. Это могут быть корпоративные сайты, блоги, новостные порталы и т.д;
* коммуникация: они могут служить платформой для общения между пользователями, например, через форумы, социальные сети или комментарии;
* электронная коммерция: сайты могут быть предназначены для продаж товаров и услуг через интернет – это интернет-магазины;
* образование: сайты могут предлагать образовательные материалы, курсы и ресурсы для обучения;
* развлечение: множество сайтов предлагают развлекательный контент, включая игры, видео и музыку.

Сайты могут быть разработаны для различных целей и могут сильно варьироваться по содержанию и структуре в зависимости от потребностей их создателей и пользователей.

Целью данной работы являлось создание системы для эффективного управления личными финансами.

В условиях современного мира, где финансовая грамотность и умение управлять ресурсами становятся важными навыками для каждого человека, создание сайта для управления личными финансами выглядит особенно актуальным и востребованным.

Актуальность сайта по управлению личными финансами на сегодняшний день очень высока по нескольким причинам:

* увеличение финансовой грамотности: с каждым годом растёт интерес к финансовой грамотности у широкой аудитории, и люди ищут ресурсы для обучения управлению своими финансами;
* нестабильность экономической ситуации: в условиях экономической неопределенности, кризисов и инфляции многие стремятся найти способы оптимизации своих расходов, сбережений и инвестиций;
* доступ к информации: интернет предоставляет огромное количество информации, и пользователи активно ищут практические советы, инструменты и приложения для управления своими финансами;
* рост интереса к инвестициям: в последние годы наблюдается рост интереса к инвестициям, особенно среди молодежи, что делает актуальными платформы, обучающие основам инвестирования и управления активами;
* технологические новшества: развитие финансовых технологий (финтех) приводит к появлению новых инструментов для управления личными финансами, что создает спрос на качественный функционал сайта.

При выполнении работы были осуществлены следующие задачи :

* учет доходов и расходов;
* постановка финансовых целей;
* генерация отчетов и визуализация данных.

Также в процессе работы были разработаны:

* удобный интерфейс и дизайн сайта, который соответствует предполагаемым предпочтениям целевой группы;
* база данных, хранящая информацию о пользователях и их транзакциях;

Данная практика по создания сайта для управления личными финансами научила меня следующему:

* проектирование и структурирование: создание сайта требует тщательного планирования и организации;
* верстка и дизайн: изучение основы CSS, HTML, JS которые позволяют создавать и стилизовать веб-страницы;
* работа с контентом: создание сайта требует наполнения его контентом, будь то тексты, изображения, видео или другие мультимедиа файлы. Это позволяет научитесь эффективно организовывать контент на страницах и создавать привлекательный и информативный контент;
* отладка и тестирование: создание сайта требует отладки и тестирования для устранения возможных ошибок и обеспечения хорошей функциональности.

Помимо получения новых знаний, это также дало возможность вспомнить уже изученное.

В ходе работы были рассмотрены ключевые аспекты управления личными финансами, выявлены основные проблемы, с которыми сталкиваются пользователи, и предоставлены функциональные требования к создаваемой системе. Также было уделено внимание современным технологиям и инструментам, которые использовались для реализации проекта.

Результатом работы стал разработанный web-сайт для эффективного управления личными финансами.

**Список источников**

1 Digital Web. — Текст : электронный // Digital Web : [Электронный ресурс]. — URL: https://www.digitalweb.ru/ (дата обращения: 22.12.2024);

2 HTML5 specification.. — Текст : электронный // HTML Standart : [Электронный ресурс]. — URL: https://html.spec.whatwg.org/multipage/ (дата обращения: 21.12.2024);

3 JavaScript documentation. — Текст : электронный // JavaScript documentation : [Электронный ресурс]. — URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference (дата обращения: 05.01.2025);

4 Learn JavaScript. — Текст : электронный // Learn JavaScript : [Электронный ресурс]. — URL: https://learn.javascript.ru/ (дата обращения: 07.01.2025).

5 PHP.net. — Текст : электронный // PHP.net. : [Электронный ресурс]. — URL: https://www.php.net/ (дата обращения: 24.12.2024);

6 SQL tutorial. — Текст : электронный // SQL cервер : [Электронный ресурс]. — URL: https://www.w3schools.com/sql/ (дата обращения: 25.12.2025);

7 Node.js Official Documentation. — Текст : электронный // Node.js [Электронный ресурс]. — URL: https://nodejs.org/dist/latest-v20.x/docs/api/ (дата обращения: 21.12.2024);

8 W3C CSS specification. — Текст : электронный // CSS specification : [Электронный ресурс]. — URL: https://www.w3.org/TR/CSS2/ (дата обращения: 21.12.2024);

9 W3Schools. — Текст : электронный // W3Schools. : [Электронный ресурс]. — URL: https://www.w3schools.com/ (дата обращения: 03.01.2025);

10 Web Development for Beginners.. — Текст : электронный // Web Development : [Электронный ресурс]. — URL: https://developer.mozilla.org/en-US/docs/Learn/Web/First\_steps (дата обращения: 27.12.2024).

**Приложение А**

***(обязательное)***

**Информационная модель**

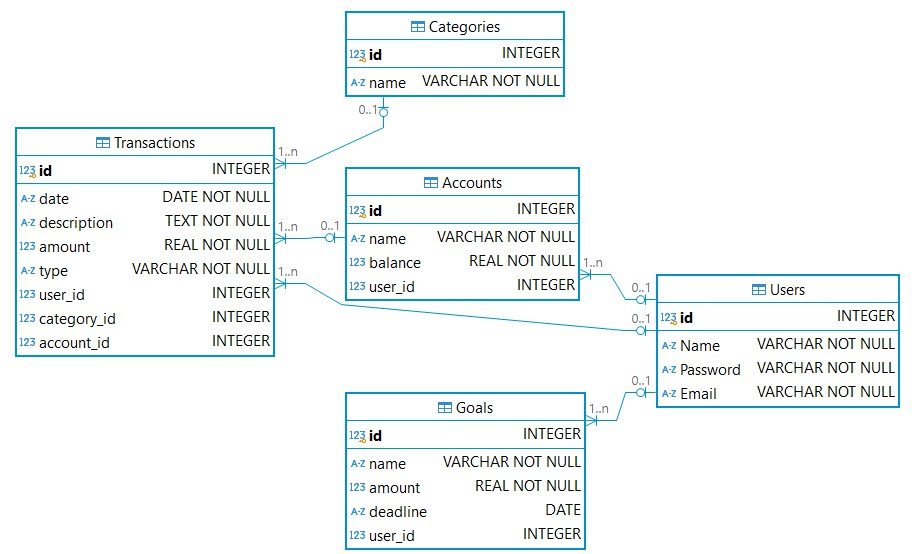


Рисунок А.1 - Информационная модель

**Приложение Б**

***(обязательное)***

**Диаграмма прецедентов**

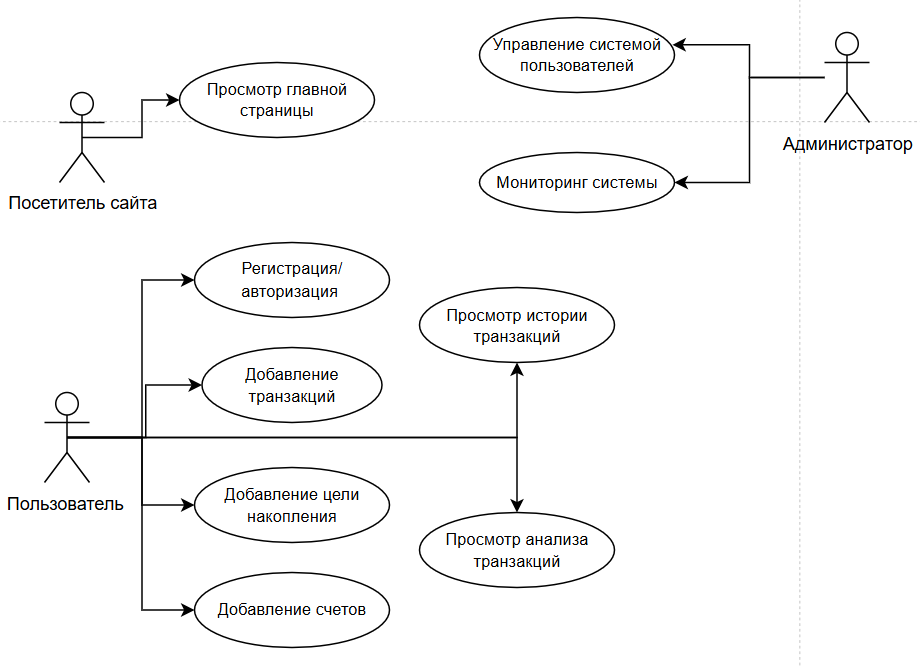
****

Рисунок Б.1 - Диаграмма прецедентов

**Приложение В**

***(обязательное)***

**Диаграмма последовательности**

****

Рисунок В.1 - Диаграмма последовательности

**Приложение Г**

***(обязательное)***

**Диаграмма классов**

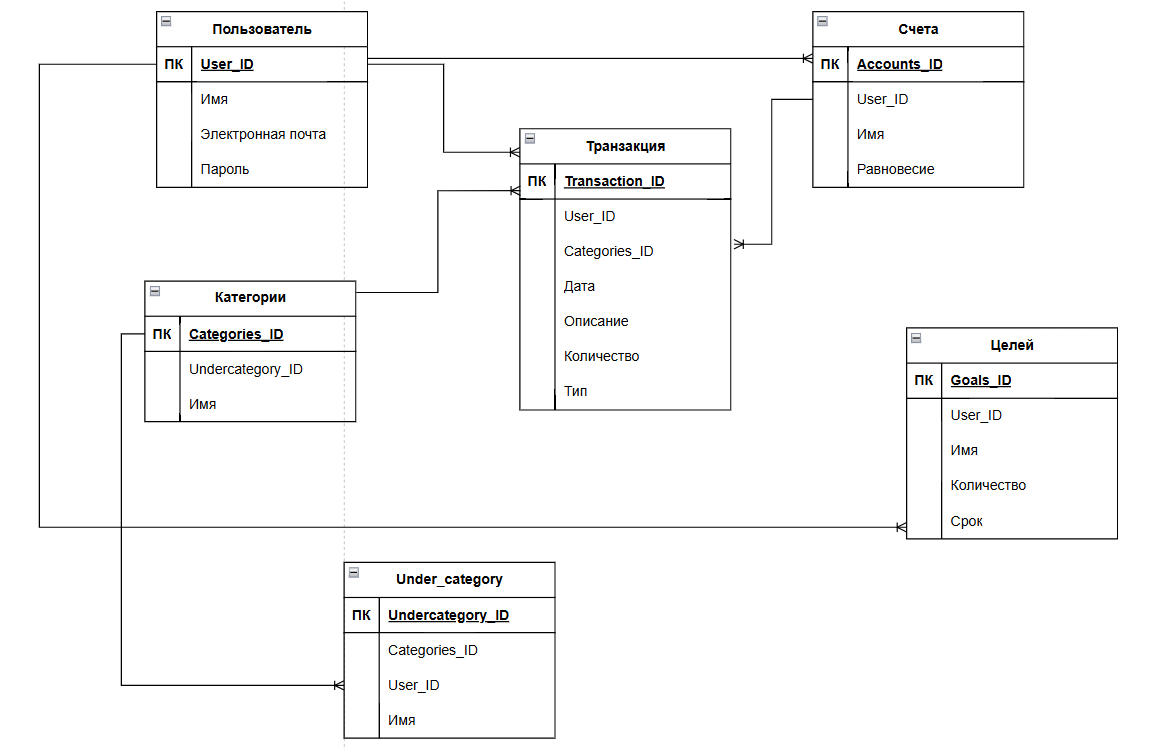
******

Рисунок Г.1 - Диаграмма классов