МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ

**ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ ОКЭИ)**

**КУРСОВОЙ ПРОЕКТ**

*ОКЭИ 09.02.07. 4 0 25. 24 О*

(код документа)

|  |
| --- |
| *ПМ.09 Разработка дизайна веб-приложения*  *Разработка веб-сайта для управления личными финансами* |

Количество листов *34*

Дата готовности *28.01.2025*

Руководитель *Адамович Никита Владимирович*

Разработал *Чурсина Валерия Игоревна*

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

Оренбург, 2025

**Содержание**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

Лист

2

*ОКЭИ 09.02.07. 4 0 25. 24 О*

*Разраб.*

*Чурсина В. И.*

*Провер.*

Адамович Н. В.

*Реценз*

*Н. Контр.*

*Утверд.*

*Отчёт по*

*курсовому проекту*

Лит.

Листов

*34*

*Отделение программирования,*

*гр. 4вб2*

[Введение](#_Toc2078) 3

[1 Анализ предметной области 6](#_Toc26006)

[2 Проектирование приложения 9](#_Toc19377)

3 Разработка программного обеспечения 12

[3.1 Описание технологического стека разработки](Чурсина_Пояснительная_записка_черновик.docx) 12

3.2 Описание алгоритма работы 16

3.3 Описание интерфейса пользователя 19

4 Тестирование веб-приложения 21

4.1 План тестирования 21

4.2 Оценка результатов проведения тестирования 22

Заключение 27

Список использованных источников 29

[Приложение А(обязательное) Информационная модель 3](#_Toc10399)1

[Приложение Б(обязательное) Диаграмма прецедентов 3](#_Toc10399)2

[Приложение В(обязательное) Диаграмма последовательности 3](#_Toc10399)3

[Приложение Г(обязательное) Диаграмма классов 3](#_Toc10399)4

**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.

Веб-сайты играют ключевую роль в современном цифровом мире. Они не только служат источником информации, но и являются важными инструментами для бизнеса, образования и социальной активности.

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 обусловлена, во-первых, ростом финансовой доступности, когда разнообразие банковских продуктов, кредитов и инвестиционных возможностей требует внимательного контроля и анализа финансовых потоков.

Во-вторых, существует необходимость планирования долгосрочных целей, таких как покупка недвижимости, образование или пенсия, что подразумевает тщательное финансовое планирование и отслеживание прогресса.

В-третьих, все больше людей осознают важность контроля над личными финансами и ищут удобные инструменты для этого, что свидетельствует о повышении уровня финансовой грамотности.

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.

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;
* интуитивно понятный интерфейс;
* дизайн и визуальная привлекательность;
* безопасность.

Целевая аудитория - необходимо определить, кому будет предназначен сайт и настроить его контент, дизайн и функционал в соответствии с интересами и потребностями пользователей.

Также важным фактором является 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, играют не менее важную роль. Сайт должен иметь привлекательный внешний вид, соответствующий бренду или целей сайта. Графика, шрифты, цветовая схема и общая композиция должны создавать приятное впечатление.

Последним фактором является 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.

Ресурсы, необходимые для реализации проекта, включают в себя технические и программные ресуры. К техническим ресурсам относится компьютерная техника, сервер для размещения веб-сайта. К программным относятся инструменты для веб-разработки (HTML, CSS, JavaScript) и системы управления контентом (CMS), которые облегчают процесс создания и обновления информации на сайте.

Расширение функционала веб-сайта для управления личными финансами может быть многообразным и охватывать разные аспекты работы как самого сайта, так и взаимодействия с клиентами.

Сформированные предложения о расширении функциональности информационной системы:

* настраиваемые категории;
* анализ тенденций расходов;
* мобильное приложение;
* составление бюджета;
* учетные записи пользователей и совместная работа;
* сообщество или поддержка.

Предложение о настраиваемых категориях предполагает разрешить пользователям создавать и редактировать собственные пользовательские категории и подкатегории, чтобы лучше отражать их потребительские привычки.

Анализ тенденций расходов означает предоставление углублённого анализа моделей расходов пользователя, выявление тенденции и отклонения.

Составление бюджета предполагает помощь пользователю в создании и отслеживании бюджета с предупреждением о превышении запланированных лимитов.

Созданный сайт для управления личными финансами позволит достичь следующих результатов:

* получение полного обзора доходов, расходов, устранение финансовой неопределенности;
* простое добавление, категоризация и поиск всех финансовых операций;
* постановка конкретных финансовых целей и отслеживание прогресса;
* развитие лучшей финансовой дисциплины благодаря систематическому отслеживанию и анализу;
* получение контроля над финансами, снижение стресса, связанного с управлением деньгами;
* 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**

Анализ предметной области включает в себя исследование различных процессов, которые происходят в этой сфере, с целью выявления возможностей для автоматизации.

Сфера финансов охватывает множество аспектов управления деньгами и активами, связанных с планированием и контролем финансовых ресурсов пользователя.

Основными аспектами, которые могут быть включены в эту область являются:

* отслеживание расходов;
* доходы и расходы;
* финансовая аналитика и отчеты;
* цели и планирование.

Отслеживание расходов означает возможность фиксировать и классифицировать ежедневные, ежемесячные и ежегодные расходы, система может автоматически группировать транзакции по категориям.

Доходы и расходы - аспект, означающий учет источников доходов, сравнение доходов и расходов для понимания общей финансовой ситуации.

Финансовая аналитика и отчеты предполагает генерацию различных отчетов и графиков для визуализации финансовых данных, что поможет пользователям лучше понять свои расходы и доходы.

Цели и планирование - это 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);
* люди среднего возраста (семьи, работающие специалисты);
* люди, стремящиеся к финансовой стабильности;
* люди, планирующие крупные покупки;
* люди с низким уровнем финансовой грамотности;
* людям, стремящимся улучшить свои финансовые навыки.

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. Во-первых, улучшение финансовой организации. Сайт предоставляет единое место для отслеживания всех доходов и расходов, создавая целостное представление о финансовом положении.

Во-вторых, простой и удобный учет. Ввод данных происходит в удобном интерфейсе, избавляя от необходимости использовать громоздкие таблицы или блокноты.

В-третьих, категоризация и анализ расходов. Система позволяет группировать транзакции по категориям, что помогает выявлять основные статьи расходов и контролировать их.

В-четвертых, постановка и отслеживание целей. Система позволяет ставить финансовые цели и отслеживать прогресс их достижения, что повышает мотивацию и помогает двигаться к намеченным финансовым целям.

В-пятых, улучшение финансовых привычек - систематическое отслеживание финансов способствует формированию более ответственного отношения к деньгам и развитию хороших финансовых привычек.

Также 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;
* доверие к стороннему сервису;
* зависимость от корректности 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: 14 декабря - 28 января

Основные процессы области:

* регистрация пользователей;
* ввод транзакций;
* категоризация транзакций;
* управление счетами;
* управление категориями;
* анализ расходов и доходов;
* постановка финансовых целей.

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

* пользователь: зарегистрированный пользователь системы;
* счет: финансовый счет пользователя;
* транзакция: отдельная финансовая операция;
* категория: категория для классификации транзакций;
* цель: финансовая цель, которую пользователь хочет достичь.

Объектом является организация «TrueStudio», профилем деятельности которой является создание веб-сайтов.

Предметом является веб-сайт для управления личными финансами.

Целью данной работы является автоматизация процесса управления личными финансами.

Для достижений цели необходимо выполнить несколько задач. Во-первых, определение целевой аудитории, что означает проведение исследования для выявления характеристик пользователей, которые могут быть заинтересованы в веб-сайте.

Во-вторых, разработка технического задания на разработку веб-сайта - сформулировать четкие требования к функциональности, дизайну и контенту веб-сайта.

В-третьих, разработка прототипа веб-сайта в соответствии с поставленными задачами. Необходимо создать интерактивный или статический прототип, который включает в себя основные страницы, навигацию и размещение контента, что позволит получить представление о взаимодействии пользователя с сайтом и внести изменения на ранних этапах разработки.

Создание дизайна - дизайн должен быть не только эстетически привлекательным, но и функциональным, соответствующим целям сайта и потребностям пользователей.

Создание интуитивно понятного интерфейса веб-сайта - разработать интерфейс, который будет удобным и доступным для понимания пользователями.

Создание базы данных для хранения данных пользователей - разработать и реализовать структуру базы данных для безопасного хранения информации о пользователях, такой как учетные записи и транзакции.

Также задачей является написание кода веб-сайта. Необходимо перевести дизайн и функциональные требования в код, используя подходящие языки программирования и технологии разработки (HTML, CSS, JavaScript, PHP, Python и др.)

Проверка работоспособности веб-сайта, наличие ошибок и недочетов -необходимо провести тестирование веб-сайта на различных устройствах и браузерах, проверяя его функциональность, производительность и соответствие заданным требованиям.

В заключении, исправление ошибок и недочетов с последующим внедрением в эксплуатацию веб-сайта и выпуск его на хостинг - внести исправления на основе тестирования и подготовить сайт к запуску.

Функциональные требования к веб-сайту:

* регистрация/авторизация пользователей с сохранением данных;
* возможность добавления транзакций и ее редактирования;
* возможность постановки финансовых целей и отслеживания прогресса;
* возможность генерации различных отчетов, представление данных в виде диаграмм;
* возможность добавления и управления несколькими счетами, отслеживание их баланса.

Нефункциональные требования к веб-сайту:

* отзывчивость;
* быстродействие;
* безопасность;
* легкость навигации;
* совместимость с браузерами;
* SEO-оптимизация.

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: оперативное устранение любых технических проблем, обеспечение безопасности и производительности сайта.

Требованиями к сопровождению данного сайта являются регулярные обновления программного обеспечения сайта, изучение новых тенденций и технологий веб-разработки, внедрение улучшений для повышения пользовательского опыта и функциональности сайта.

Также требованиями являются отслеживание производительности сайта и внедрение оптимизации для повышения скорости загрузки страниц и общей отзывчивости, адаптация к новым устройствам и технологиям, постоянная поддержка и обслуживание.

**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.

У Figma есть множество преимуществ. К преимуществам Figma относятся кроссплатформенность, совместная работа, облачное хранение, простота использования, возможность создания интерактивных прототипов, интеграция с другими инструментами. Ниже подробно расписаны все преимущества.

Кроссплатформенность означает, что Figma доступен как на Windows, Mac, Linux, а также имеет мобильное приложение для iOS и Android, что позволяет работать на любом устройстве.

Совместная работа означает, что несколько пользователей могут одновременно работать над одним проектом, делиться мнениями и комментариями, что улучшает процесс командной работы.

Облачное хранение означает, что все проекты сохраняются в облаке, что обеспечивает доступ к ним с любого устройства и в любое время.

Простота использования означает удобный и интуитивно понятный интерфейс, облегчающий процесс дизайна и прототипирования.

Возможность создания интерактивных прототипов означает, что Figma позволяет создавать прототипы с возможностью анимации и интерактивных элементов, что помогает лучше визуализировать идеи.

Интеграция с другими инструментами означает, что 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.

К преимуществам VS Code относятся:

* поддержка множества языков программировани;
* наличие расширений и плагинов;
* интеграция с Gi;
* легкость использования;
* поддержка платформ.

Поддержка множества языков программирования означает, что VS Code поддерживает большое количество языков программирования, что делает его отличным выбором для разработчиков, работающих с разными технологиями.

Наличие расширений и плагинов означает, что VS Code имеет широкий выбор расширений и плагинов, которые добавляют новые функциональности и возможности к редактору кода.

Интеграция с Git означает, что VS Code имеет встроенную поддержку Git, что позволяет разработчикам удобно работать с версионным контролем и управлять своими проектами.

Легкость использования означает, что VS Code имеет интуитивный интерфейс и простые настройки, что делает его простым в использовании даже для новичков в программировании.

Поддержка платформ означает, что 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;
* встраиваемость;
* самодостаточность;
* кроссплатформенность;
* надежность;
* легковесность;
* бесплатное использование;
* поддержка SQL.

Простота в использовании и развертывании означает, что 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.

Встраиваемость означает, что SQLite может быть встроен непосредственно в приложение, что делает его идеальным выбором для мобильных приложений, веб-приложений и других систем, где установка отдельного сервера баз данных нежелательна или невозможна.

Самодостаточность означает, что база данных SQLite не зависит от внешних серверов, это обеспечивает высокую надежность и доступность, поскольку отказ сервера базы данных не повлияет на работу приложения.

Кроссплатформенность означает, что SQLite работает на всех основных операционных системах и поддерживает большинство языков программирования.

Надежность означает, что SQLite имеет высокую надежность и обеспечивает ACID-свойства (атомарность, согласованность, изолированность, долговечность) транзакций, гарантируя целостность данных.

Легковесность означает, что SQLite относительно небольшой по размеру, что делает его идеальным для использования в приложениях с ограниченными ресурсами.

Бесплатное использование означает, что SQLite распространяется под лицензией Public Domain, что означает его бесплатное использование и модификацию без каких-либо ограничений.

Поддержка SQL означает, что 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;
* кроссплатформенность;
* SEO-оптимизация;
* возможность создания динамических сайтов;
* совместимость с другими технологиями.

Простота использования означает, что HTML - достаточно простой язык, который легко изучить и понять для создания базовых веб-страниц.

Кроссплатформенность означает, что HTML поддерживается всеми браузерами и операционными системами, что позволяет создавать универсальные веб-сайты.

SEO-оптимизация означает, что HTML позволяет создавать легко читаемые веб-страницы для поисковых систем, что способствует улучшению позиций сайта в выдаче результатов поиска.

Возможность создания динамических сайтов означает, что HTML может использоваться в сочетании с CSS и JavaScript для создания интерактивных и динамических веб-страниц.

Совместимость с другими технологиями означает, что 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.[2]

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;
* легкость внесения изменений;
* кроссбраузерная совместимость;
* улучшенная производительность;
* адаптивный дизайн.

Раздельное оформление означает, что CSS позволяет разделить содержимое и его оформление, что делает код более чистым и упорядоченным.

Легкость внесения изменений означает, что изменения стилей в CSS применяются ко всем элементам на странице сразу, что облегчает процесс обновления дизайна.

Кроссбраузерная совместимость означает, что CSS позволяет создавать универсальные стили, которые одинаково хорошо отображаются на различных браузерах.

Улучшенная производительность означает, что поскольку CSS позволяет отделить структуру от визуального оформления, загрузка страниц происходит быстрее и производительность сайта улучшается.

Адаптивный дизайн означает, что с помощью CSS можно создавать адаптивные и отзывчивые дизайны, которые корректно отображаются на различных устройствах и разрешениях экрана.[17]

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;
* простота изучения;
* широкое сообщество и поддержка;
* постоянное развитие;
* быстрый результат;
* обширная экосистема;
* большое количество вакансий.

Универсальность означает, что JavaScript используется в cоздание интерактивных элементов, обработкe событий, динамическом изменение содержимого страниц, анимации.

Простота изучения означает, что JavaScript имеет относительно простой синтаксис, что делает его сравнительно легким для изучения, особенно для начинающих программистов, также он обладает гибким динамическим типированием, что упрощает написание кода.

Широкое сообщество и поддержка означает, что 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 означает, что язык JavaScript постоянно развивается и совершенствуется, появляются новые стандарты и функции, что расширяет его возможности и повышает производительность.

Быстрый результат означает, что благодаря интерпретируемой природе JavaScript, можно быстро увидеть результат своей работы, что ускоряет процесс разработки.

Обширная экосистема означает, что множество библиотек и фреймворков позволяют решать самые разнообразные задачи быстро и эффективно.

Большое количество вакансий означает, что благодаря широкому применению JavaScript, спрос на разработчиков, владеющих этим языком, очень высок.[14]

**3.2 Описание алгоритма работы**

Варианты взаимодействие пользователя с системой:

* аутентификация;
* ввод данных;
* постановка целей;
* генерация отчетов;
* управление счетами.

Ниже подробно расписано каждое взаимодействие.

Аутентификация: пользователь входит в систему, используя свои учетные данные (имя пользователя/адрес электронной почты и пароль), далее 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;
* хранение данных;
* категоризация транзакций;
* расчет баланса;
* отслеживание прогресса к цели;
* генерация отчетов.

Каждый процесс также подробно расписан ниже.

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. [6]

SQLite с Sequelize – это удобный способ работать с базой данных в Node.js приложениях. Он позволяет абстрагироваться от SQL и использовать объекты JavaScript для работы с данными.

На рисунке 1 представлен код подключения бд.

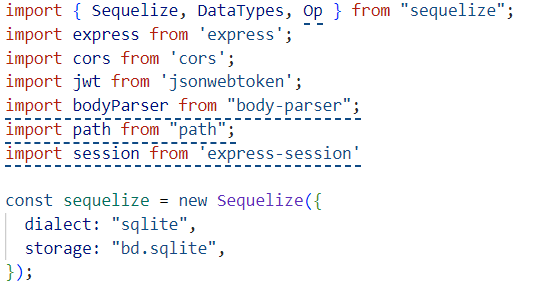


Рисунок 1 - Подключение бд

На рисунке 2 представлен скрипт для обработки формы авторизации. Данный скрипт выполняет следующие действия: получение данных из формы, валидация данных, хеширование пароля, сохранение данных в бд, отправка ответа. Ниже распишем каждое из действий.

Получение данных из формы: скрипт будет получать данные, отправленные пользователем через форму регистрации (имя пользователя, пароль, email).

Валидация данных: скрипт должен проверять, что введенные данные соответствуют требованиям (например, что email имеет правильный формат).

Хеширование пароля: пароль пользователя никогда не должен храниться в базе данных в открытом виде. Скрипт должен использовать функцию хеширования для преобразования пароля в невосстановимую строку.

Сохранение данных в базе данных: после валидации и хеширования, данные пользователя (включая хешированный пароль) будут сохранены в базе данных.

Отправка ответа: скрипт должен отправить ответ пользователю, сообщая об успешной регистрации или ошибках.



Рисунок 2 - Скрипт для обработки формы авторизации

**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. [10]

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;
* соблюдение консистентности;
* понимание и контекст.

Использование простоты и понятности означает, что интерфейс должен быть легким для понимания и использования даже для новых пользователей, использование понятных и интуитивно понятных элементов управления.

Соблюдение консистентности означает, что 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.

Прием понимание и контекст означает, что пользователь всегда понимает, где находится и что происходи, 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.

Главная страница пользователя - отражает краткое, но информативное представление о текущем финансовом состоянии.

В блоке «список счетов» представлен список всех добавленных счетов с текущими балансами, а также возможность добавления новых.

В блоке «форма добавление транзакций» представлена простая форма для ввода новой транзакции с полями для: суммы, категории (выбор из списка), выбор счета, выбор типа транзакции (доход или расход).

В блоке «последние транзакции» список последних финансовых операций с кратким описанием и суммой.

На странице транзакции представлена таблица со всеми транзакциями, сортируемая по различным параметрам (дата, сумма, категория, счет), возможность поиска по ключевым словам в описании.

На странице цели представлены блок «создание целей» и «список всех целей».

В блоке «создание целей» представлен функционал для добавления новых финансовых целей, также указание целевой суммы, срока и названия цели.

В блоке «список всех целей» представлены все цели 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;
* 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.

Тестирование направлено на выявление различных типов дефектов, включая:

* функциональные ошибки: продукт работает не так, как задумывалось;
* ошибки интерфейса: проблемы с взаимодействием с пользователем;
* ошибки производительности: продукт работает медленно, нестабильно;
* ошибки безопасности;
* ошибки совместимости с различными платформами, устройствами.

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, задач составление плана действий, распределение обязанностей между участниками команды.
2. Анализ требований - изучение требований к сайту и функционалу, определение критериев успешного тестирования.
3. Создание тест-кейсов - процесс разработки подробных инструкций для проверки конкретных аспектов функциональности приложения, системы или веб-сайта.
4. Отчетность - подготовка отчетов о результатах тестирования, документирование всех выявленных проблем и ошибок, их приоритизация и уточнение.
5. Устранение ошибок - передача найденных ошибок разработчикам для их исправления, повторное тестирование после устранения проблем.
6. 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.

Ниже представлены ключевые аспекты, которые следует учитывать при оценке результатов тестирования:

* соответствие требованиям;
* покрытие тестами;
* качество продукта;
* анализ ошибок;
* сравнение с предыдущими версиями;
* пользовательский опыт;
* рекомендации и улучшения;
* документация и отчетность.

Аспект соответствие требованиям предполагает проверку того, насколько результаты тестирования соответствуют заранее определённым требованиям и спецификациям. Важно оценить количество обнаруженных дефектов, их серьёзность и влияние на функциональность продукта. Если продукт не соответствует ключевым требованиям, это может стать причиной задержки выпуска или необходимости доработки.

Покрытие тестами включает анализ того, какой объем кода или функциональных возможностей был протестирован. Это может быть покрытие кода (сколько строк кода было проверено) или покрытие требований (сколько из заявленных требований было протестировано). Высокий уровень покрытия тестами снижает риск наличия скрытых дефектов.

Качество продукта оценивается по таким параметрам, как стабильность, производительность и безопасность. Используются метрики, такие как количество критических и некритических дефектов на тысячу строк кода, чтобы определить общее состояние продукта. Качество продукта напрямую влияет на его восприятие пользователями.

Анализ ошибок включает классификацию найденных дефектов по типам, частоте возникновения и степени влияния на пользователей. Важно понять причины возникновения ошибок и их потенциальное воздействие на работу системы. Это помогает предотвратить повторение подобных проблем в будущем.

Сравнение текущих результатов тестирования с предыдущими версиями продукта позволяет оценить динамику улучшений или ухудшений. Это особенно важно для долгосрочных проектов, где необходимо отслеживать прогресс и эффективность внесённых изменений.

Пользовательский опыт (UX) оценивается на основе отзывов конечных пользователей. Важно понять, насколько продукт удобен в использовании, соответствует ли он ожиданиям пользователей и насколько интуитивно понятен его интерфейс. Удовлетворённость пользователей является ключевым показателем успеха продукта.

На основе анализа результатов тестирования формулируются рекомендации по исправлению обнаруженных дефектов и улучшению процесса тестирования. Это может включать предложения по оптимизации тестовых сценариев, улучшению инструментов тестирования или изменению подходов к разработке.

Подготовка отчётов о тестировании является важной частью процесса. Отчёты должны включать все полученные данные, метрики, результаты анализа и рекомендации.

Область тестирования: форма авторизации.

Для тестирования данной функции используем тест-кейсы, которые представлены в таблицах 1-3.

Рассмотрим такие варианты тестирования функции регистрации пользователя на сайт, как: корректное заполнение всех полей формы, некорректное заполнения поля «Ваш E-mail», а также авторизацию незарегистрированного пользователя.

Таблица 1 - Корректное заполнение всех полей формы

|  |  |  |  |
| --- | --- | --- | --- |
| №1 | Регистрация пользователя | | |
| Ссылка на ТЗ | Чурсина В. | Высокий | Регистрация |
| 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 3 представлено верное заполнение формы регистрации. Пользователь верно указал E-mail и смог зарегистрироваться на сайт. После введения данных пользователь переходит на главную страницу сайта.

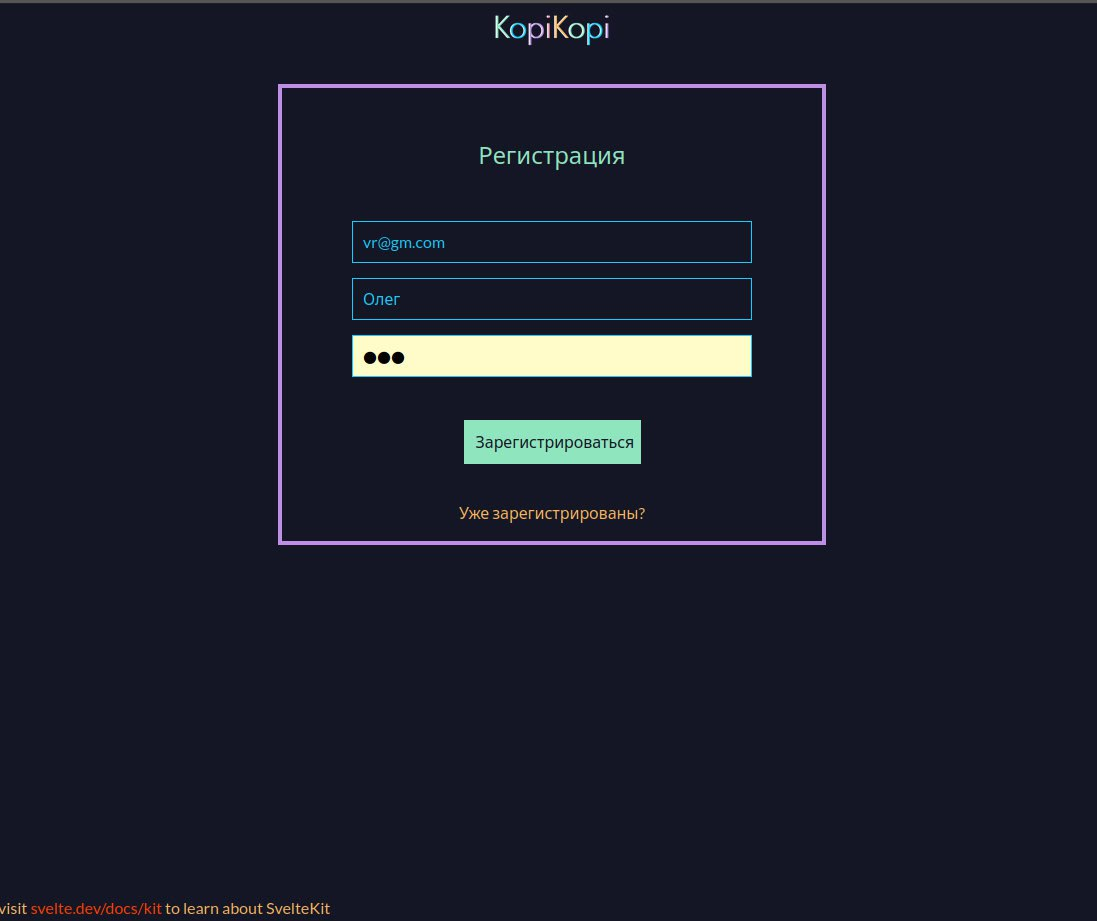


Рисунок 3 - Верное заполнение формы регистрации

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

|  |  |  |  |
| --- | --- | --- | --- |
| №2 | Регистрация пользователя | | |
| Ссылка на ТЗ | Чурсина В. | Высокий | Регистрация |
| 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 4 представлен результат неверного заполнения поля «Ваш E-mail». Пользователь не смог зарегистрироваться по причине того, что его E-mail не корректен. Система выдала пользователю ошибку.

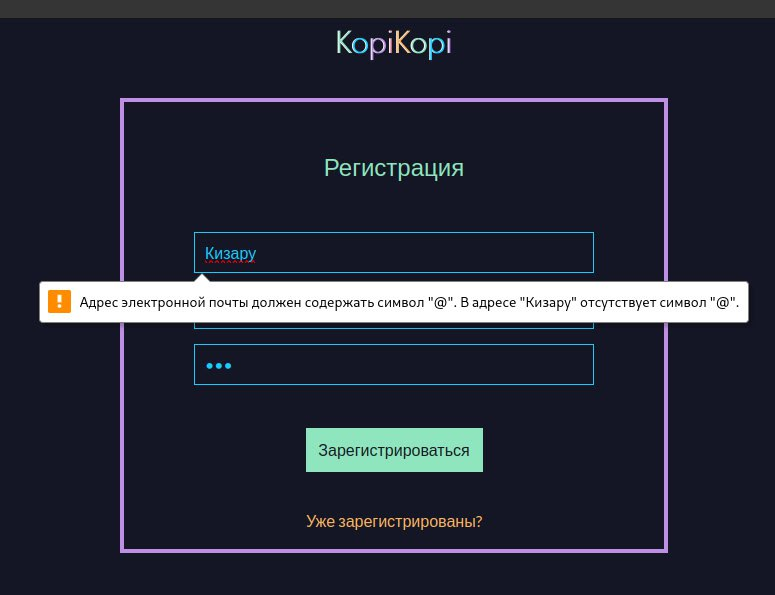


Рисунок 4 - Результат неверного заполнения поля «Ваш E-mail»

Далее рассмотрим авторизацию пользователя, который ранее не был зарегистрирован в системе.

Таблица 3 - Авторизация незарегистрированного пользователя.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №3 | Авторизация пользователя | | | |
| Ссылка на ТЗ | Чурсина В. | | Высокий | Авторизация |
| 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 5 представлен результат авторизации незарегистрированного ранее пользователя.

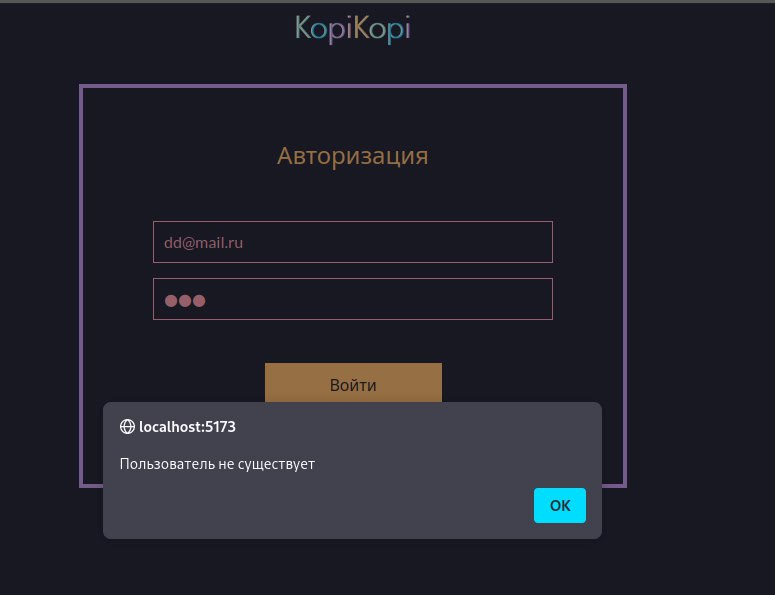


Рисунок 5 - Результат авторизации незарегистрированного пользователя

**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.[18]

Сайты используются для множества целей. Во-первых, они служат для предоставления информации о компаниях, продуктах или услугах. Во-вторых, они являются платформами для коммуникации между пользователями, обеспечивая возможность общения через форумы, социальные сети и комментарии. В-третьих, сайты активно применяются в сфере электронной коммерции, выступая в качестве интернет-магазинов для продажи товаров и услуг. Кроме того, они играют важную роль в образовании, предлагая обучающие материалы, курсы и ресурсы. Наконец, многие сайты создаются для развлечения, предоставляя пользователям 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 являлась автоматизация процесса управления личными финансами. Данная цель была выполнена.

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. Во-первых, увеличение финансовой грамотности. С каждым годом растёт интерес к финансовой грамотности у широкой аудитории, и люди ищут ресурсы для обучения управлению своими финансами.

Во-вторых, нестабильность экономической ситуации. В условиях экономической неопределенности, кризисов и инфляции многие стремятся найти способы оптимизации своих расходов, сбережений и инвестиций.

В-третьих, доступ к информации. Интернет предоставляет огромное количество информации, и пользователи активно ищут практические советы, инструменты и приложения для управления своими финансами.

В-четвёртых,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.

В-пятых, 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:

* определение целевой аудитории;
* разработка технического задания на разработку веб-сайта;
* разработка прототипа веб-сайта в соответствии с поставленными задачами;
* создание дизайна;
* создать интуитивно понятный интерфейс веб-сайта;
* создание базы данных для хранения данных пользователей;
* написание кода веб-сайта;
* проверка работоспособности веб-сайта, наличие ошибок и недочетов;
* исправление ошибок и недочетов с последующим внедрением в эксплуатацию веб-сайта и выпуск его на хостинг.

Данная практика по создания сайта для управления личными финансами научила меня тому, что создание сайта требует тщательного планирования, организации и изучение основы CSS, HTML, JS которые позволяют создавать и стилизовать веб-страницы.

Также создание сайта требует наполнения его контентом, будь то тексты, изображения, видео или другие мультимедиа файлы, это позволяет научиться эффективно организовывать контент на страницах и создавать привлекательный и информативный контент.

В заключении при создании сайта необходимы отладка и тестирование 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 Ильиных, П. Ф. Программирование на JavaScript. — Москва: Вильямс, 2022. — 300 с.

2 Кузнецов, В. Н. Разработка мультимедийных приложений. — Москва: БИОТ, 2020. — 220 с.

3 Коваленко, Т.Н. Системы управления базами данных. – Пермь: Издательство ПГТУ, 2021. – 240 с.

4 Ларионов, Р. А. Основы работы с базами данных. — Новосибирск: Сибирское университетское издательство, 2021. — 240 с.

5 Сидоров, К. Ю. Проектирование веб-сайтов на Node.js. — Омск: Омский государственный университет, 2021. — 275 с.

6 Смирнова, Е. В. Графический дизайн для веб-приложений. – Челябинск: Челябинский государственный университет, 2020. – 220 с.;

7 Сидоров, И. Н. Управление проектами в сфере информационных технологий. – Москва: Альпина Паблишер, 2020. – 290 с.

8 Тарасова, Н.К. Финансовое планирование для молодых специалистов. – Санкт-Петербург: Невская книга, 2021. – 200 с.

9 Федоров, Д. О. Разработка сайтов: практическое руководство. — Нижний Новгород: Нижегородское издательство, 2021. — 300 с.

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

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

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

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

14 MDN Web Docs. — Текст : электронный // (Mozilla Developer Network : [Электронный ресурс]. — URL: https://developer.mozilla.org/ru/ (дата обращения: 24.12.2024);

15 SQL tutorial. — Текст : электронный // SQL cервер : [Электронный ресурс]. — URL: https://www.w3schools.com/sql/ (дата обращения: 25.12.2024);

16 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);

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

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

19 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).

20 Песцова Ирина, Информационная модель / Ирина Песцова. ‒ Текст : электронный // Справочник от автор24 : [Элекронный ресурс]. ‒ URL: https://spravochnick.ru/informatika/informacionnaya\_model/ (дата обращения: 21.12.2024);

**Приложение А**

***(обязательное)***

**Информационная модель**

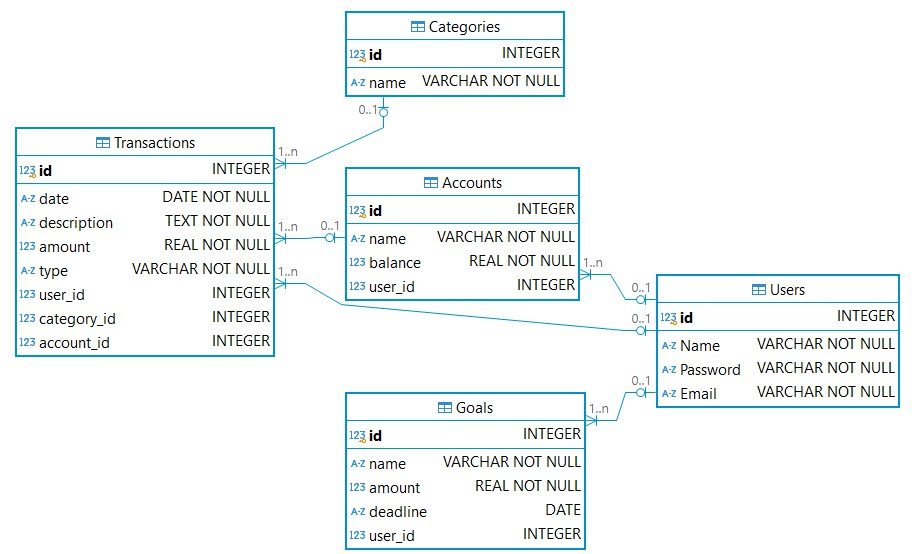


Рисунок А.1 - Информационная модель

**Приложение Б**

***(обязательное)***

**Диаграмма прецедентов**

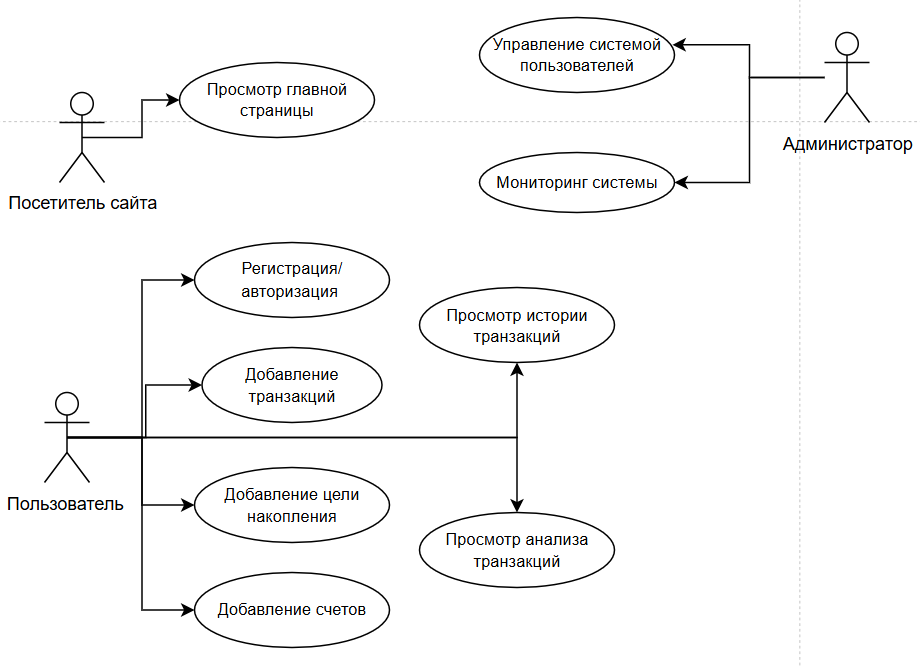
****

Рисунок Б.1 - Диаграмма прецедентов

**Приложение В**

***(обязательное)***

**Диаграмма последовательности**

****

Рисунок В.1 - Диаграмма последовательности

**Приложение Г**

***(обязательное)***

**Диаграмма классов**

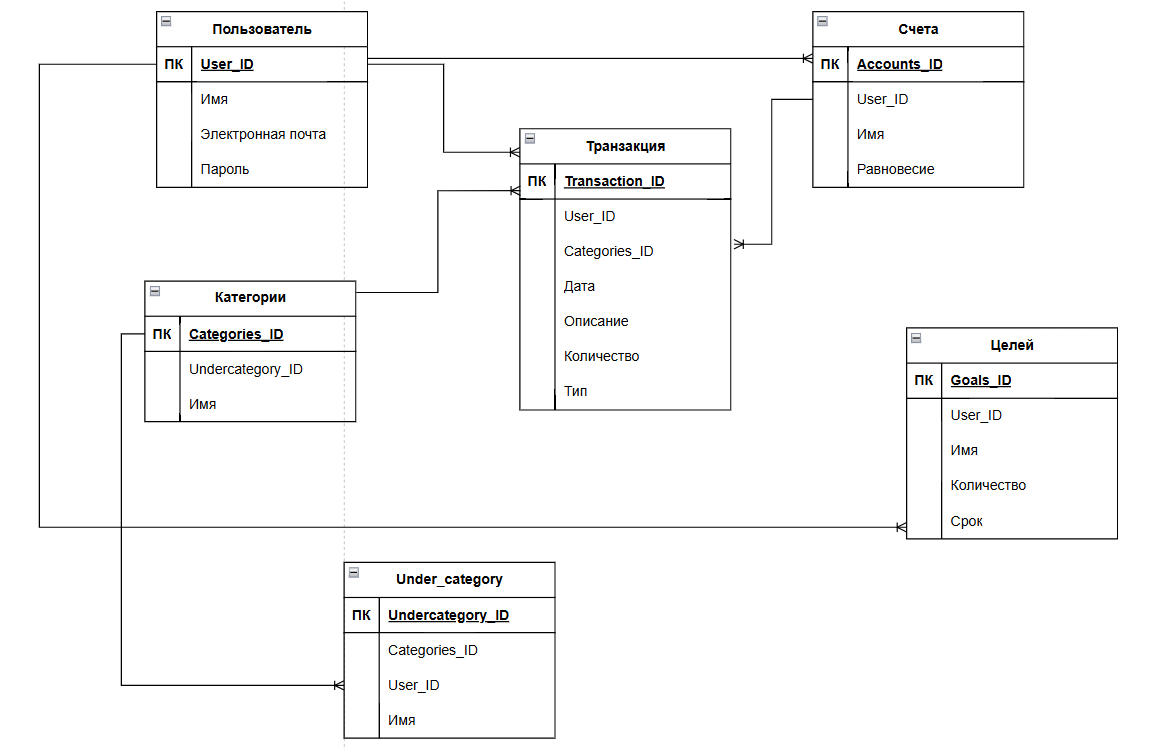
******

Рисунок Г.1 - Диаграмма классов