

Специальность 09.02.07 Информационные системы и программирование

**КУРСОВОЙ ПРОЕКТ**

**на тему:** Проектирование, дизайн и программная реализация модуля управления благотворительными организациями

по МДК 05.01 Проектирование и дизайн информационных систем,   
МДК.05.02 Разработка кода информационных систем

Студент: Сироткин Евгений Романович

(Фамилия, И.О.)

Курс 3 группа 32ИСП (до)

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

Преподаватели: Игнатьева Т.А.

(Фамилия, И.О.)

Бурнин Д.A.

(Фамилия, И.О.)

Москва 2025 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc198148008)

[1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ 6](#_Toc198148009)

[1.1 Основные принципы разработки информационных систем 6](#_Toc198148010)

[1.2 Архитектурные подходы к созданию модуля информационной системы 11](#_Toc198148011)

[1.3 Современные технологии и инструменты разработки информационных систем 13](#_Toc198148012)

[1.4 Анализ существующих решений в области управления благотворительные организациями 15](#_Toc198148013)

[1.5 Постановка задачи и определение к модулю управление телекоммуникационными услугами 17](#_Toc198148014)

[2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДУЛЯ УПРАВЛЕНИЯ Благотворительными организациями 18](#_Toc198148015)

[2.1 Выбор платформы и инструментария для разработки модуля 18](#_Toc198148016)

[2.2 Разработка технического задания на модуль управления персональными данными студентов вуза 19](#_Toc198148017)

[2.3 Проектирование базы данных и настройка серверной инфраструктуры 22](#_Toc198148018)

[2.4 Реализация основных функций модуля 24](#_Toc198148020)

[2.5 Тестирование и отладка разработанного модуля 34](#_Toc198148021)

[2.6 Создание пользовательской документации и руководства по эксплуатации 35](#_Toc198148022)

[ЗАКЛЮЧЕНИЕ 38](#_Toc198148023)

[СПИСОК ИСПОЛЬЗОВАННЫх ИСТОЧНИКОВ 40](#_Toc198148024)

# ВВЕДЕНИЕ

Современные благотворительные организации сталкиваются с необходимостью эффективного управления своими программами, базами благополучателей и доноров, а также финансовыми потоками. В условиях высокой социальной ответственности и растущих требований к прозрачности деятельности автоматизация процессов управления становится критически важной. Разработка специализированного модуля для управления благотворительными проектами позволяет оптимизировать работу организации, повысить доверие жертвователей и минимизировать административные издержки.

Актуальность данной работы обусловлена несколькими факторами. Во-первых, благотворительные организации реализуют разнообразные программы, включая помощь нуждающимся, образовательные и медицинские проекты, экологические инициативы и другие, что требует комплексного подхода к их администрированию. Во-вторых, доноры и партнеры ожидают удобства и открытости при взаимодействии с организацией, включая возможность отслеживания использования средств через личный кабинет или отчетные системы. В-третьих, автоматизация учета пожертвований и распределения ресурсов позволяет снизить риск ошибок и повысить оперативность принятия решений.

Целью курсового проекта является проектирование, дизайн и программная реализация модуля управления благотворительными программами. Для достижения этой цели необходимо решить следующие задачи: провести анализ существующих решений и конкурентов в отрасли.

1. Определить функциональные и технические требования к модулю.
2. Разработать архитектуру системы, включая базу данных и пользовательский интерфейс.
3. Создать дизайн-макеты интерфейсов для администратора и клиентов.
4. Реализовать модуль с использованием современных технологий программирования.
5. Провести тестирование и отладку системы.

Объектом исследования является процесс разработки системы управления благотворительной организацией.  
Предметом исследования выступают методы проектирования, дизайна и программирования, направленные на создание эффективного и удобного инструмента для координации благотворительной деятельности

Цели, задачи и методология исследования  
Основной целью данного курсового проекта является комплексная разработка — от проектирования до программной реализации — специализированной системы управления для благотворительных организаций. Разрабатываемое решение призвано обеспечить централизованный контроль над всеми аспектами деятельности, включая взаимодействие с благополучателями, управление пожертвованиями, координацию волонтеров и формирование отчетности.

Для достижения поставленной цели необходимо последовательно решить следующие ключевые задачи:

* Проведение анализа существующих цифровых платформ и методов управления в сфере благотворительности, а также оценка потребностей некоммерческого сектора.
* Определение функциональных и технических требований к системе, включая модули для работы с донорами, проектами, волонтерами и финансовой отчетностью.
* Проектирование архитектуры системы с учетом безопасности данных, структуры базы данных (хранение информации о пожертвованиях, проектах, участниках) и концепции пользовательского интерфейса.
* Создание дизайн-макетов интерфейсов для различных категорий пользователей (администраторы, благополучатели, доноры, волонтеры).
* Программная реализация системы с использованием современных технологий (например, облачные решения, CRM для НКО) и инструментов, обеспечивающих прозрачность и удобство взаимодействия.

# ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

# Основные принципы разработки информационных систем

Разработка информационных систем (ИС) для управления благотворительными организациями строится на основополагающих принципах, гарантирующих их эффективность, надежность и соответствие запросам пользователей.

В данном случае эти принципы были реализованы за счет внедрения современных методологий проектирования и наглядного отображения процессов. Такой подход позволяет усовершенствовать сбор, распределение и мониторинг благотворительных ресурсов, повышая прозрачность и управляемость системы.

Рассмотрим принцип модульности и декомпозиции процессов

Назначение: моделирование функциональной структуры системы, выделение основных процессов и их взаимосвязей.

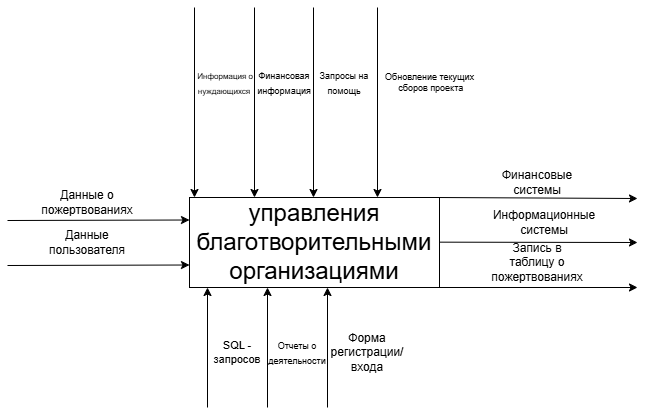


Рис. 1. IDEF0: основные процессы модуля

Описание: Диаграмма IDEF0 (Рис. 1) представляет высокоуровневые процессы модуля управления благотворительной организацией.

Входные данные включают запросы благополучателей, информацию о донорах и условия финансирования.

Выходные данные - реализованные программы, финансовые отчёты и аналитика эффективности помощи.

Взаимодействие компонентов

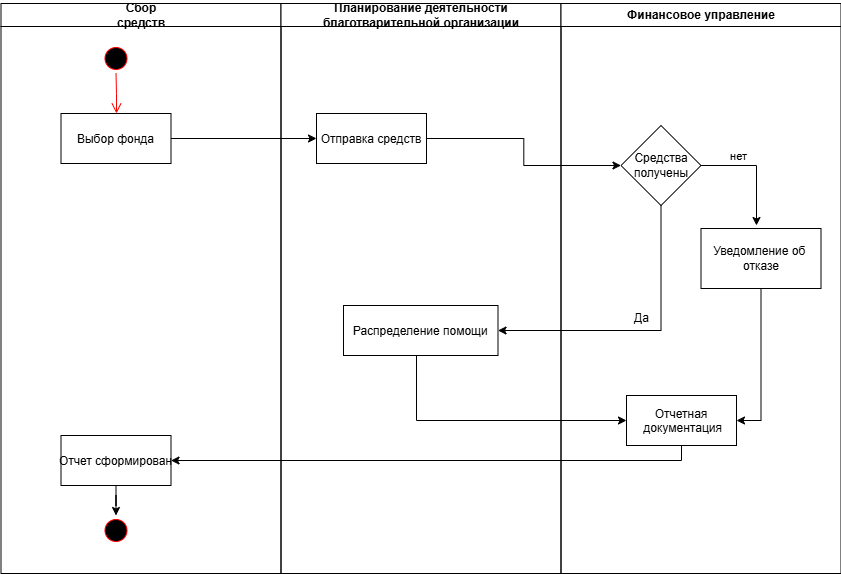
Назначение: моделирование последовательности действий в процессе, включая ветвления и параллельные потоки.

Рис. 2. UML Activity: процесс обработки запроса.

Описание: На диаграмме (Рис. 2) представлены этапы управления благотворительной организацией:

1. Сбор средств - организация проводит кампании по привлечению пожертвований.
2. Планирование деятельности- определяются цели и направления использования средств.
3. Финансовое управление- осуществляется контроль и распределение бюджета.
4. Выбор фонда - средства направляются в конкретный благотворительный фонд.
5. Отправка средств - перевод денег выбранному фонду.
6. Средства получены - подтверждение успешного перевода.

* Если перевод не удался (нет), отправляется уведомлениеоботказе.
* При успешном переводе (да) происходит распределениепомощи нуждающимся.

1. **Отчетная документация** - формируются отчеты о проделанной работе.
2. **Отчёт формирован** - завершающий этап, подтверждающий документальную прозрачность деятельности.

Ключевые элементы: взаимодействие между клиентом, модулем управления услугами и биллингом.

Состояние жизненного цикла.

Назначение: отображение жизненного цикла объекта (например, заявки) и его реакций на события.

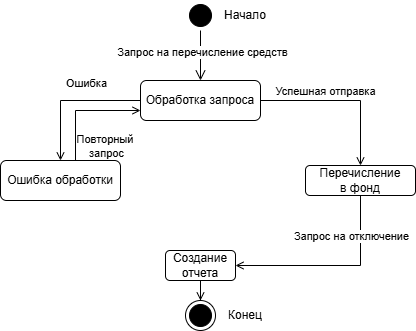


Рис. 3. UML State Machine: Жизненный цикл заявки.

Описание: Диаграмма (Рис. 3) показывает переходы состояний запроса: от «Запроснаперечисление» через «Проверку», «Успешнаяотправка», до «Перечислениев фонд» или «Ошибка». Например, при ошибке обработки запрос возвращается на этап исправления.

Прозрачность потоковых данных.

Назначение: Детализация процессов, представленных в IDEF0, для анализа подпроцессов и их взаимодействия.

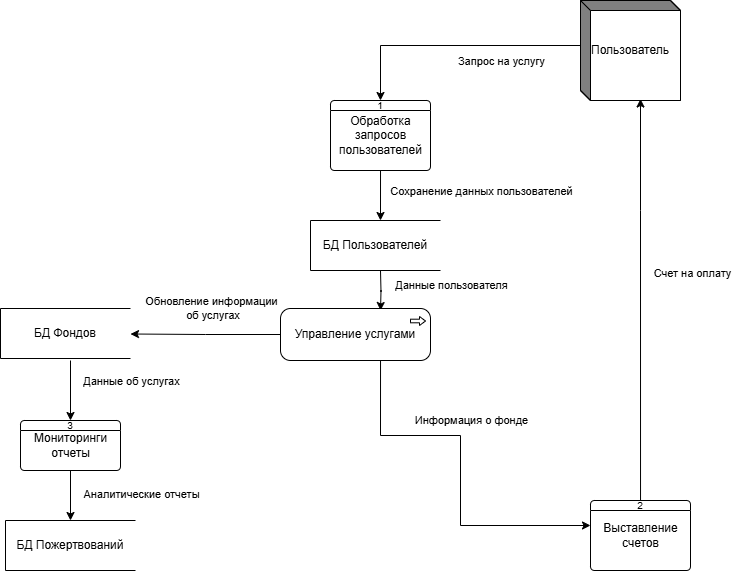


Рис. 4. Декомпозиция процессов управления услугами.

Описание: декомпозиция (Рис. 4) отображает ключевые процессы управления благотворительной организацией, включая обработку запросов пользователей, управление услугами и взаимодействие с данными.

Автоматизация и точность расчетов.

Назначение: детализация процессов, представленных в IDEF0, для анализа под процессов и их взаимодействия.

Все процессы спроектированы с учетом требований защиты данных (GDPR, PCI DSS). Например, в DFD (Рис. 4) данные пользователей шифруются при передаче в БД.

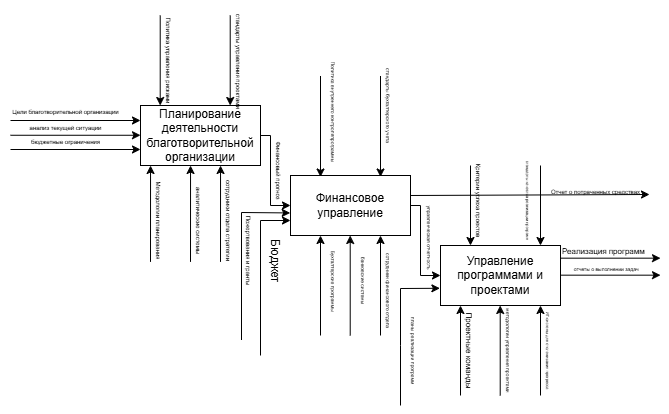


Рис. 5. Декомпозиция процессов управления услугами.

Описание: декомпозиция (Рис. 5) демонстрирует этапы управления благотворительной организацией, включая финансовое планирование, координацию проектов и контроль их выполнения. Например, показано, как распределение бюджета влияет на эффективность реализации программ и достижение социальных целей.

Архитектура системы управления благотворительными организациями, описанная в IDEF0 и UML, обеспечивает гибкость, позволяя добавлять новые функциональные модули (например, интеграцию с онлайн-платежами или CRM) без необходимости перепроектирования всей системы.

Это особенно важно для благотворительных организаций, так как позволяет постепенно расширять функционал (учет пожертвований, волонтерская деятельность, отчетность) без нарушения текущих процессов.

* 1. Архитектурные подходы к созданию модуля информационной системы

Архитектура модуля управления благотворительными организациями.

Система построена на принципах гибкости, надежности и автоматизации процессов. Основной акцент сделан на разделение функциональности на независимые компоненты, что упрощает разработку, тестирование и масштабирование.

Модульная структура

Система разделена на автономные модули, каждый из которых отвечает за конкретную задачу. Например,

* Модуль управления пожертвованиями - обрабатывает поступление средств, распределение и отчетность.
* Модуль волонтерского учета - автоматизирует регистрацию, координацию и мотивацию волонтеров.
* Модуль отчетности - формирует прозрачные отчеты для доноров и регулирующих органов.

Такой подход позволяет легко добавлять новые функции, такие как интеграция с CRM-системами или онлайн-фандрайзингом, без перестройки всей архитектуры.

Поток данных и взаимодействие

Для корректной передачи информации между компонентами используются четко определенные маршруты. Например:

1. Пожертвование поступает через веб-интерфейс или мобильное приложение.

2. Данные проходят валидацию и сохраняются в базу.

3. Информация синхронизируется с модулями учета и отчетности.

Это исключает дублирование операций и снижает риск ошибок. Например, при поступлении нового взноса система автоматически обновляет баланс и формирует благодарственное письмо.

Гибкость и обработка сценариев

Логика системы учитывает различные сценарии:

* Проверка целевого назначения пожертвования.
* Уведомление донора о статусе перевода.
* Обработка ошибок (например, отмена платежа или изменение суммы).

В случае нештатных ситуаций система предлагает варианты решения, минимизируя ручное вмешательство.

Инструменты и безопасность

Для реализации использованы технологии, обеспечивающие модульность и безопасность:

* Frontend: HTML, CSS, JavaScript - для удобного взаимодействия с донорами и волонтерами.
* Backend: PHP, SQL - для обработки данных и формирования отчетов.
* Контроль версий: Git - для прозрачности изменений.
* Тестирование: локальные среды (например, OSPanel) - для отладки перед развертыванием.

Защита данных соответствует стандартам GDPR:

* Персональные данные шифруются.
* Доступ администраторов защищен двухфакторной аутентификацией.
* Финансовые операции соответствуют требованиям PCI DSS.

Итоговые преимущества

Предложенная архитектура позволяет:

* Автоматизировать ключевые процессы (учет пожертвований, волонтерская деятельность).
* Обеспечивать прозрачность через личные кабинеты доноров и публичную отчетность.
* Быстро адаптироваться под новые требования (например, запуск новых сборов).
* Соответствовать стандартам безопасности и законодательства.

Таким образом, система не только решает технические задачи, но и помогает достичь ключевых целей: повышение доверия доноров, снижение административной нагрузки и увеличение эффективности благотворительной деятельности.

* 1. Современные технологии и инструменты разработки информационных систем

Разработка модуля управления благотворительными организациями

Для создания эффективной системы управления благотворительными проектами были применены современные технологии и инструменты, обеспечивающие удобное проектирование, разработку, управление данными и контроль версий. В рамках проекта использовались следующие решения:

Инструменты разработки

1. Draw.io

* Draw.io Использовался для визуализации ключевых процессов и архитектуры системы:
* IDEF0 - моделирование структуры системы и бизнес-процессов благотворительной организации;
* UML Activity - описание взаимодействия между модулями (например, сбор пожертвований и распределение средств);
* DFD - отображение потоков данных между донорами, волонтерами и администраторами;
* UML state machine моделирование жизненного цикла объекта.

2. Visual Studio Code

* Основная среда для написания кода (PHP, HTML, CSS, SQL). Преимущества:
* Поддержка синтаксиса и автодополнение для ускорения разработки;
* Интеграция с Git (например, через GitLens);
* Встроенные средства отладки PHP.
* Пример: в VSCode реализована логика обработки пожертвований и формирования отчетов.

3. phpMyAdmin

* Инструмент для проектирования и администрирования базы данных MySQL. С его помощью:
* Созданы таблицы Доноров (контакты, история пожертвований) и Проектов (цели, бюджеты, статусы);
* Оптимизированы SQL - запросы для быстрого формирования отчетов;
* Настроены связи между таблицами (например, распределение средств по проектам).

4. GitKraken

* Графический клиент для работы с Git, обеспечивший:
* Удобное управление ветками (разделение задач: интерфейс доноров, админ-панель);
* Визуализацию истории изменений;
* Синхронизацию с GitHub для командной работы.

5. OSPanel

* Локальный сервер для тестирования модуля в среде, близкой к реальной. Функционал:
* Быстрое развертывание веб-сервера (Apache/Nginx), PHP и MySQL;
* Создание виртуальных доменов (напр., `charity-admin.local` для тестирования панели управления);
* Интеграция с phpMyAdmin для удобной работы с БД;
* Поддержка разных версий PHP (7.4/8.2) для проверки совместимости.

Технологии программирования

1. HTML и CSS

* HTML - структура веб-интерфейсов:
* Формы для пожертвований;
* Личные кабинеты доноров и волонтеров.
* CSS - адаптивный дизайн:
* Медиа-запросы для мобильных устройств;
* Стилизация элементов (например, интерактивные графики сбора средств).

2. PHP

* Серверная логика:
* Обработка данных (проверка платежей, регистрация участников);
* Интеграция с БД (выборка информации о проектах);
* Генерация динамического контента (персонализированные благодарности донорам).

3. SQL

* Работа с базой данных:
* Запросы с JOIN для объединения данных (например, связь доноров и проектов);
* Триггеры для автоматического обновления статусов при достижении целей.

4. GitHub

* Платформа для командной разработки:
* CI/CD через GitHub Actions - автоматическое тестирование изменений;
* Хранение резервных копий проекта;
* Управление задачами через Issues (отслеживание доработок, исправление ошибок).

Такой набор инструментов и технологий обеспечил эффективное создание системы управления благотворительными организациями с удобным интерфейсом, надежной базой данных и возможностью масштабирования.

* 1. Анализ существующих решений в области управления благотворительные организациями

Современные благотворительные организации активно используют специализированные системы для автоматизации ключевых процессов: управления донорами, сбором пожертвований, отчетностью и коммуникацией с подопечными. Наиболее распространены CRM-платформы, такие как Bitrix24 и Microsoft Dynamics, которые предоставляют базовые инструменты для работы с благотворителями, но часто требуют доработки под специфику некоммерческого сектора. Например, интеграция с платежными системами для автоматического списания регулярных пожертвований или формирование отчетов для проверяющих органов.

Специализированные решения, такие как DonorPerfect или Bloomerang, предлагают готовые модули для учета пожертвований, управления кампаниями и работы с волонтерами. Однако их внедрение сопряжено с высокими затратами и сложностью адаптации под небольшие организации. Кроме того, многие системы не обеспечивают достаточной гибкости при настройке индивидуальных условий сотрудничества с донорами, что критично для НКО, стремящихся персонализировать взаимодействие.

Отдельного внимания заслуживают облачные платформы, такие как GiveLively или Donately, которые фокусируются на автоматизации сбора средств. Они упрощают прием пожертвований и интеграцию с соцсетями, но не покрывают весь спектр задач благотворительной организации — например, не поддерживают управление программами помощи или отчетность перед грантодателями.

Анализ выявил ключевые проблемы существующих решений:

* Недостаточная интеграция между модулями (например, разрыв между сбором средств и отчетностью перед донорами);
* Сложность настройки под уникальные потребности организации;
* Высокая стоимость лицензий и обслуживания.

Эти недостатки стали основой для формулирования требований к разрабатываемому решению.

* 1. Постановка задачи и определение к модулю управление телекоммуникационными услугами

Цель разработки модуля является создание гибкой системы для управления деятельностью благотворительных организаций, объединяющей работу с донорами, распределением средств, отчетностью и мониторингом эффективности помощи.

Функциональные требования:

1.Управление программами помощи:

* Возможность создания динамических программ с учетом целевых групп (дети, пожилые, бездомные животные), видов помощи (финансовая, материальная, волонтерская) и дополнительных условий (например, адресная помощь);
* Автоматическое обновление условий программ для всех участников (благополучателей, волонтеров, партнеров).

2. Мониторинг эффективности помощи:

* Сбор данных о выполненных проектах, количестве благополучателей и использованных ресурсах;
* Генерация отчетов для анализа социального воздействия и оптимизации программ.

3.Донорский и волонтерский интерфейс:

* Личный кабинет донора/волонтера с возможностью отслеживания своих пожертвований, участия в программах и подачи заявок;
* Мобильная адаптация для удобного взаимодействия.

Нефункциональные требования:

* Производительность: обработка до 10 000 транзакций в час (для крупных благотворительных фондов);
* Безопасность: шифрование персональных данных (SSL), двухфакторная аутентификация для администраторов;
* Совместимость: поддержка интеграции с существующими CRM и базами данных (MySQL);
* Масштабируемость: возможность добавления новых модулей (например, интеграция с соцсетями для сбора пожертвований).

Учет потребностей пользователей .

Для доноров и волонтеров важны:

* Минимизация шагов при внесении пожертвований или записи на мероприятия;

Прозрачность отчетности без скрытых расходов;

* Мгновенные уведомления о новых программах и результатах помощи.

Для администраторов и организаторов:

* Единая панель управления программами, финансами и волонтерами;
* Автоматизация рутинных задач (напоминания о регулярных пожертвованиях, отчетность перед регуляторами).

# ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДУЛЯ УПРАВЛЕНИЯ Благотворительными организациями

* 1. Выбор платформы и инструментария для разработки модуля

При выборе инструмента для вёрстки сайта важно подобрать удобную и функциональную среду разработки. На сегодняшний день существует множество редакторов кода, однако их возможности зачастую схожи, и не все предоставляют по-настоящему мощный инструментарий, способный значительно ускорить и упростить процесс создания веб-приложений.

В данном проекте для разработки веб-сервиса был выбран **Visual Studio Code (VS Code)** — современный, легковесный, но при этом мощный редактор, поддерживающий широкий спектр языков программирования, включая JavaScript, Python, HTML и CSS. Это делает его идеальным решением для фронтенд- и бэкенд-разработки. Среди ключевых преимуществ VS Code — встроенная интеграция с Git, удобный отладчик, а также огромное количество расширений, позволяющих кастомизировать рабочую среду под конкретные задачи.

Кроме того, VS Code отличается кроссплатформенностью, работая на Windows, macOS и Linux, что обеспечивает гибкость в выборе операционной системы. Активное сообщество разработчиков постоянно расширяет функционал редактора, предлагая новые плагины и улучшения. Благодаря этому VS Code остаётся одним из самых востребованных инструментов в среде веб-разработчиков. Интерфейс редактора (см. рисунок 6).



Рис. 6. Интерфейс Visual Studio Code

Такой набор инструментов помогает эффективно выстроить рабочий процесс и сконцентрироваться на важных аспектах разработки, сократив время на рутинные задачи

.

* 1. Разработка технического задания на модуль управления персональными данными студентов вуза

Техническое задание является основополагающим документом, который регулирует ключевые аспекты создания модуля: его функционал, архитектурные особенности и условия приемки. Согласно данным таблицы 2.1, в документе четко прописаны базовые характеристики проекта, включая сроки выполнения и распределение зон ответственности между участниками.

Таблица 2.1 Техническое задание

| Параметр | Описание |
| --- | --- |
| Назначение документа | Определение требований и порядка разработки модуля управления благотворительными организациями для благотворительных организаций. |
| Заказчик | ООО «Доброе сердце» |
| Исполнитель | Сироткин Евгений Романович, студент группы 32ИСП (до) |
| Основание для разработки | Курсовой проект по МДК 05.01 и МДК 05.02 специальности 09.02.07 «Информационные системы и программирование». |
| Плановые сроки работ | Начало: 01.02.2025 Окончание: 24.04.2025 Этапы работ указаны в разделе «Порядок контроля и приемки». |
| Порядок оформления результатов | Работы сдаются поэтапно. По завершении каждого этапа предоставляются результаты для согласования с заказчиком и преподавателем. |

Анализ таблицы 2.2 демонстрирует, что техническое задание подробно раскрывает не только бизнес-требования, такие как автоматизация управления услугами и интеграция с платежными сервисами, но и технические условия, включая производительность, безопасность и совместимость. Отдельный акцент сделан на модульность архитектуры, что подчеркивается наличием требований к масштабируемости системы.

Таблица 2.2 Назначение и цель создания модуля

| Параметр | Описание |
| --- | --- |
| Назначение модуля | Автоматизация процессов управления благотворительной организацией: учет пожертвований, распределение средств, мониторинг эффективности программ, интеграция с донорскими платформами. |
| Цель создания | Повышение эффективности работы компании за счет оптимизации управления услугами, снижения ошибок и улучшения клиентского сервиса. |

Основные требования к модулю

Функциональные требования

1. Управление программами помощи:

* Возможность создания динамических программ с учетом целевых групп (дети, пожилые, бездомные животные), видов помощи (финансовая, материальная, волонтерская) и дополнительных условий (например, адресная помощь);
* Автоматическое обновление условий программ для всех участников (благополучателей, волонтеров, партнеров).

2. Учет пожертвований и финансов:

* Интеграция с платежными системами (sber, tbank) для приема онлайн-пожертвований;

3. Мониторинг эффективности помощи:

* Сбор данных о выполненных проектах, количестве благополучателей;

4.Донорский и волонтерский интерфейс:

* Личный кабинет донора/волонтера с возможностью отслеживания своих пожертвований, участия в программах и подачи заявок;

Нефункциональные требования:

* Безопасность: шифрование персональных данных (SSL), двухфакторная аутентификация для администраторов;
* Совместимость: поддержка интеграции с существующими CRM и базами данных (MySQL);
* Масштабируемость: возможность добавления новых модулей (например, интеграция с соцсетями для сбора пожертвований).

Требования к дизайну

* + Стиль интерфейса: современный, минималистичный.
  + Адаптивность: поддержка мобильных устройств.
  + Цветовая схема: соответствие корпоративному стилю заказчика.

Информационное обеспечение

* + Языки программирования: PHP (бэкенд), HTML/CSS/JavaScript (фронтенд).
  + База данных: MySQL.
  + Инструменты: Visual Studio Code, Draw.io (для диаграмм), Git (для контроля версий).

Программное обеспечение

Клиентская часть:

* + Браузеры: Google Chrome 85+, Firefox 80+, Safari 14+.

Серверная часть:

* + Веб-сервер: Apache/Nginx.
  + PHP: версии 7.4/8.3.
  1. Проектирование базы данных и настройка серверной инфраструктуры

Для разработки веб-сервиса прежде всего необходимо создать базу данных, которая должна содержать информацию о благотворительных проектах, об конечной цели сбора и прогресса сбора средств на веб-сервисе и о пользователях системы.

Данные должны храниться в базе данных в виде связанных между собой

таблиц:

а) благотворительные проекты – данные о представляемых на сайте услуг;

б) аккаунты – данные для входа на сайт и ограничения доступа;

в) услуги – данные о пожертвованиях;

Для разработки веб-сервиса для сбора средств в благотворительные организации была выбрана свободная реляционная система управления базами данных MySQL версии 8.2 для 64-битных систем семейства Windows, используемая через веб-интерфейс администрирования phpMyAdmin, который позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL, просматривать содержимое таблиц и баз данных. Через него были созданы объекты базы данных. В качестве сервера, на котором осуществлялась работа вышеперечисленных средств, был выбран Apache версии, поддерживающей PHP 8.1 для 64-битных систем семейства Windows.

Перед созданием базы данных разработаем ER-диаграмму, отображающую структуру таблиц и связей между ними (см. рисунок 7).

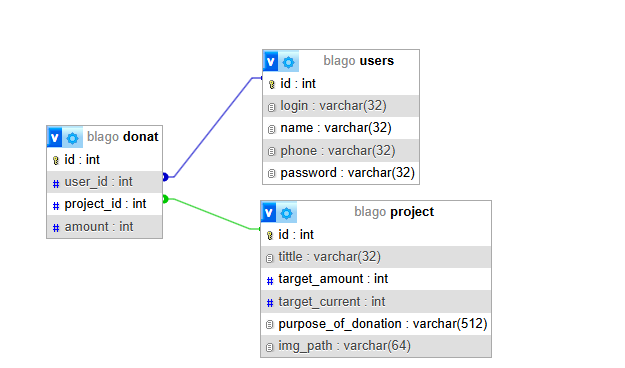
****

Рис. 7. ER-диаграмма базы данных

База данных состоит из трех таблиц (см. рисунок 8) с атрибутами:

1) таблица «users»: код пользователя, логин пользователя, имя пользователя, фамилия пользователя, телефон, пароль, права;

2) таблица «donat»: код пользователя, код проекта, сумма;

3) таблица «project»: название, цель сбора, собранная сумма, описание, фото проекта;

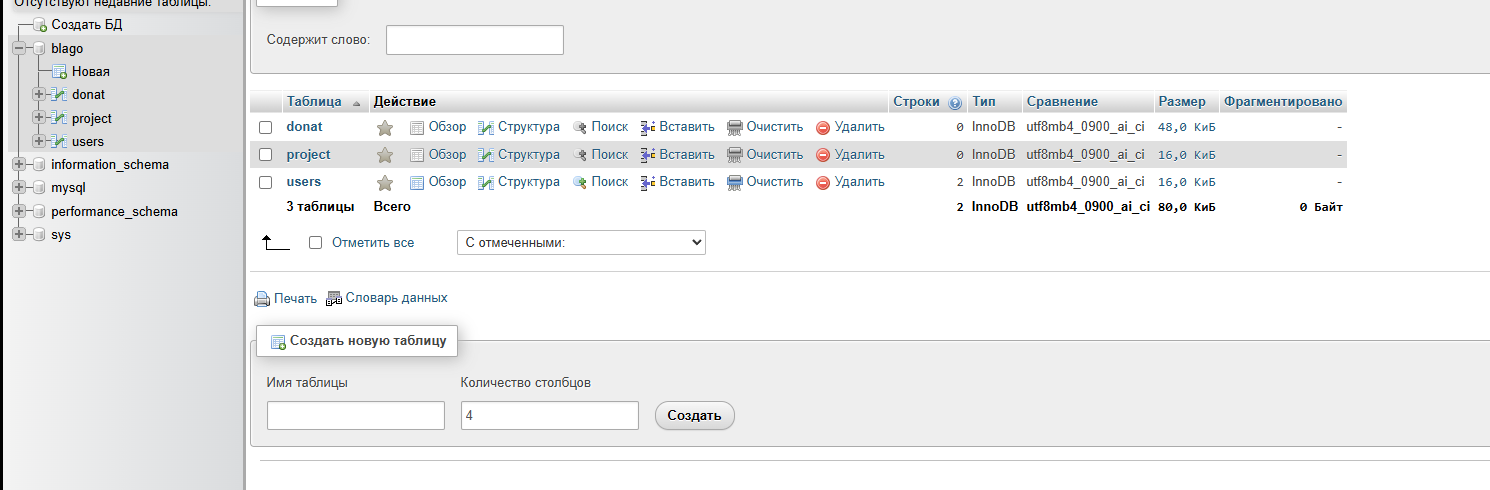
****

Рис. 8. База данных в phpMyAdmin

В данном разделе была спроектирована структура базы данных модуля, включающая три взаимосвязанные таблицы: пользователи, донат и проекты. ER-диаграмма наглядно отображает отношения между сущностями. Реализация базы данных выполнена в MySQL с использованием phpMyAdmin, что обеспечивает надежность и масштабируемость системы. Разработанная БД послужила фундаментом для последующей реализации функциональных возможностей модуля.

1. 1. Реализация основных функций модуля

Перед тем как наполнить сайт контентом, нужно разработать скрипт основных блоков – заголовок, блок отображения контента и подвал. Код для заголовка (см. рисунок 9) и подвала (см. рисунок 10) страницы идентичен для всех страниц.



Рис. 9. Код заголовка страниц

Это HTML-код шапки сайта с навигационным меню, включающим ссылки на главную, проекты, раздел помощи и контакты. В зависимости от авторизации пользователя (проверка через $\_SESSION['auth']) отображается либо кнопка входа, либо ссылка на личный кабинет.

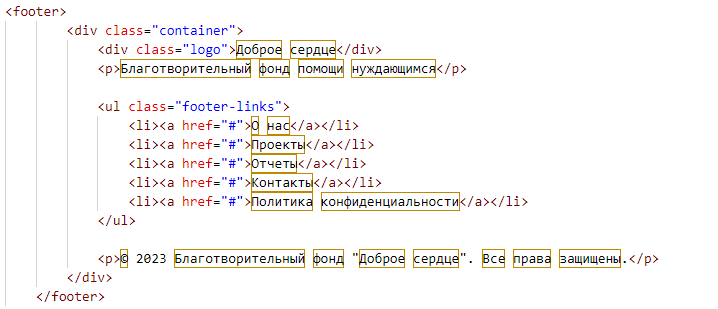


Рис. 10. Код подвала страниц

После того, как был разработан шаблон сайта и прописаны его стили (см. рисунок 10), можно посмотреть результат на главной странице (см. рисунок 11).

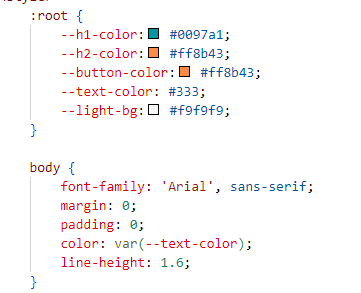
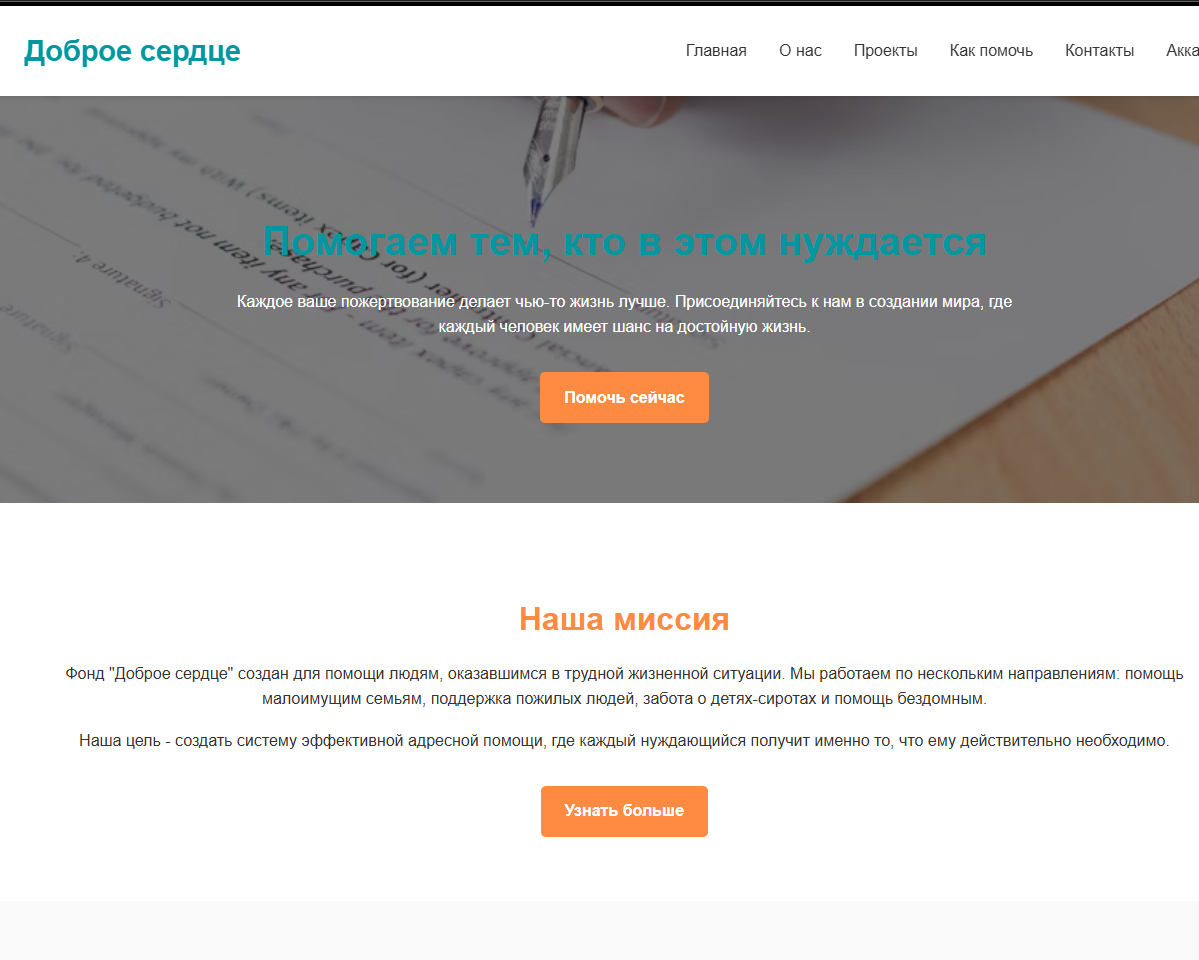


Рис. 10. Стили, прописанные в CSS



Ри. 10. Главная страница веб-сервиса

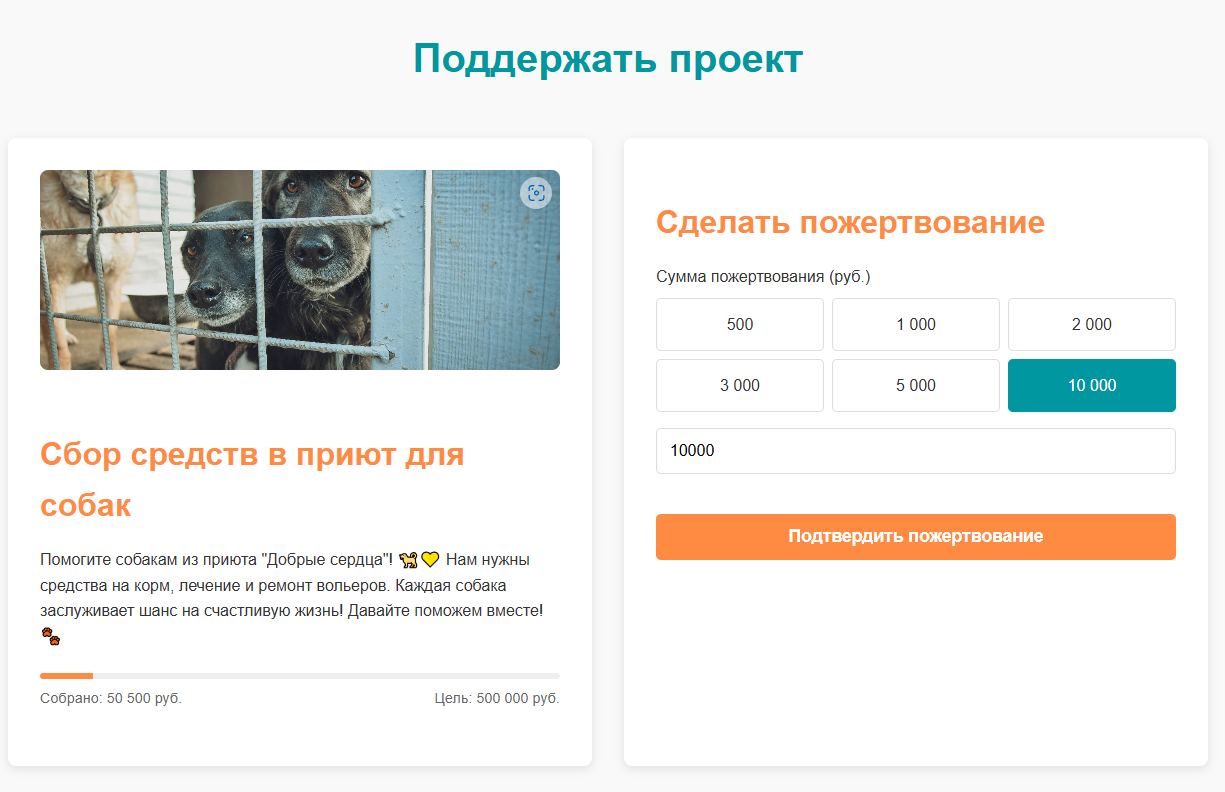
Так же на главной присутствует раздел “Предоставляемы нами проекты” (см. рисунок 11), который формируется запросом к базе данных (см. рисунок 12). Подключение к базе данных выполняется в отдельном файле, так как требуется для всех запросов (см. рисунок 13).



Риc. 11. Раздел “Предоставляемы нами проекты”



Риc. 12. код вывода записей проектов из БД



Риc. 13. Раздел пожертвования

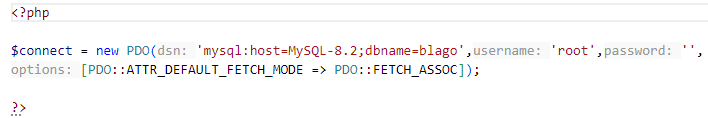
При выборе суммы пожертвования, обновляет сумму сбора в «target\_current» базы данных (см. рисунок 15).



Риc. 14. Код раздела пожертвований

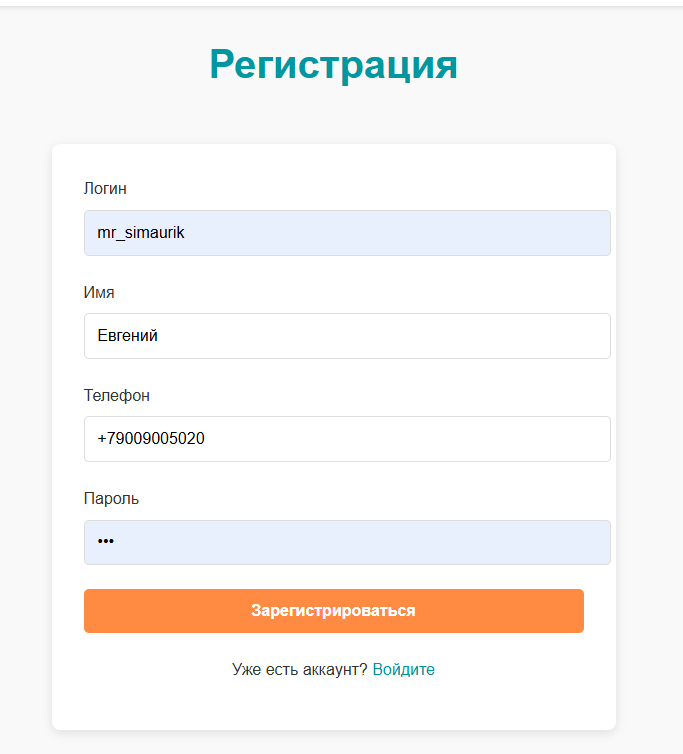


Риc. 15. Код подключения к БД пожертвований



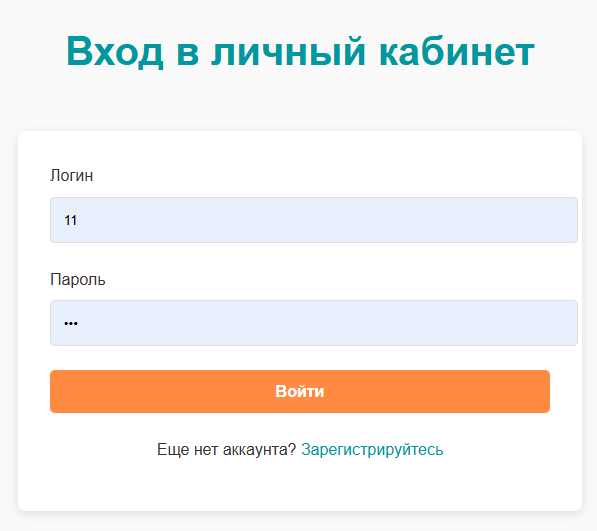
Риc. 16. Подключение к БД

Форма регистрации пользователя (см. рисунок 17).



Риc. 17. Форма регистрации.

Форма авторизации пользователя (см. рисунок 18).

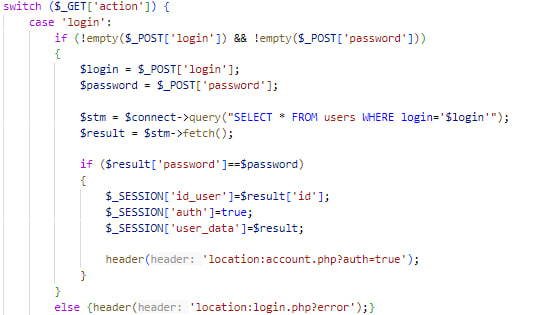


Риc. 19. Форма авторизации.

При заполнении формы «Регистрация» запускается скрипт, который проверяет наполненность полей и добавляет новую запись в таблицу «users» базы данных (см. рисунок 20).

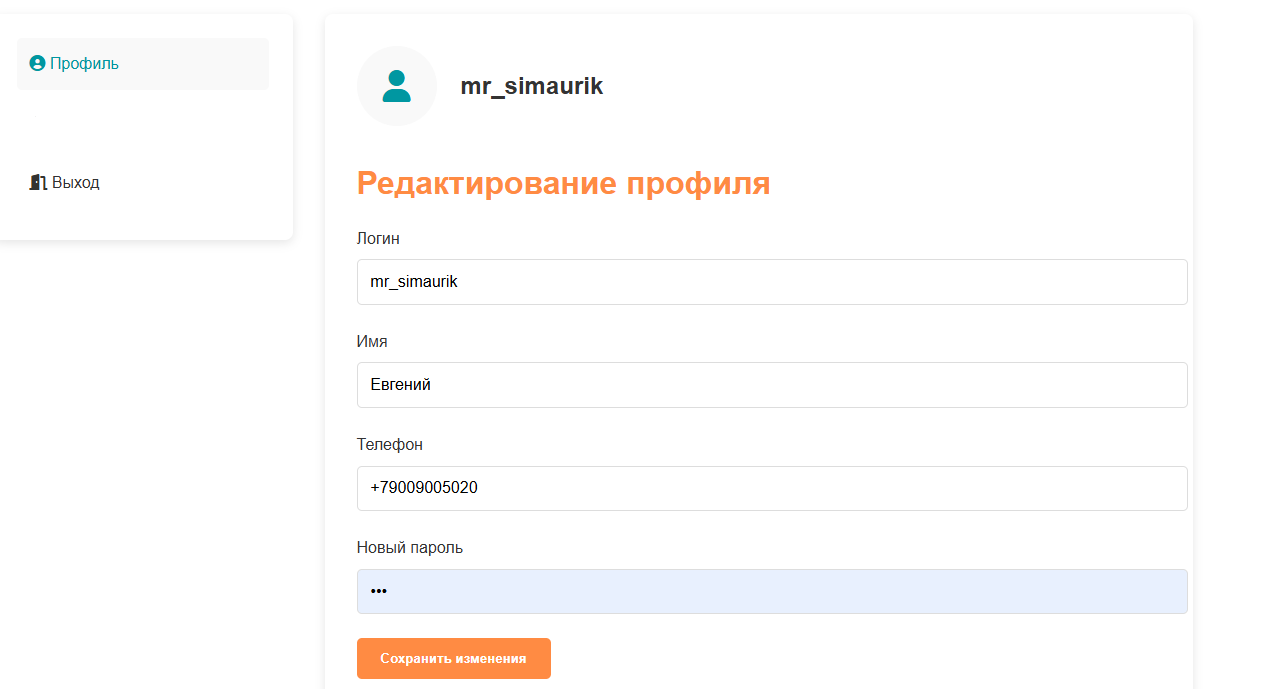
Если пользователь вводит данные в форму «Авторизация», скрипт проверяет наличие информации в полях и ищет логин в таблице «users». Если пользователь найден, происходит проверка пароля (см. рисунок 20).

Риc. 20. Код регистрации пользователя



Риc. 21. Код авторизации пользователя

После успешной авторизации/регистрации пользователю будет доступен личный кабинет, в котором доступно редактирование личного кабинета (см. рисунок 20).



Риc. 22. Личные данные

В этой форме пользователь может изменить данные код представлен (см. рисунок 23), и произвести выход из системы, код представлен (см. рисунок 24).



Риc. 23. Код редактирование данных



Риc. 24. Код выхода из системы

* 1. Тестирование и отладка разработанного модуля

Чтобы проект на выходе не содержал ошибок, необходимо проведение тестирования. Тестирование позволяет выявить недоработки проекта и осуществить работу по их устранению. Результаты тестирования представлены в таблице 2.3

Таблица 2.3 Тест-кейс «Пожертвование в благотворительные организации»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Заголовок** | **Шаги тестирования** | **Набор тестовых данных** | **Ожидаемый результат** | **Фактический результат** | **Приоритет** | **Статус** | **Описание дефекта** |
| Пожертвование в благотворительные организации | 1. Открыть страницу регистрации.  2. Ввести логин.  3. Ввести имя.  4. Ввести номер телефона.  5. Ввести пароль.  6. Нажать "Зарегистрироваться".  7. Перейти в аккаунт  8.Открыть раздел “Проекты”  9.Выбрать интересующий пользователя проект | Имя:  Евгений  Телефон: 8(966)-967-65-66  Логин:  Gg1122  Пароль: Password | Успешная регистрация с последующей авторизацией и пожертвованием в выбранный проект | Совпадает с ожидаемым |  |  |  |
| Проверка выбранного проекта | 1. Открыть страницу авторизации.  2. Ввести логин.  3. Ввести пароль.  4. Нажать "авторизоваться".  5. Перейти в раздел проекты  6. Нажать выбрать | Логин: mr\_simaurik  Пароль: 111 | Появится сообщение заполните это поле | Совпадает с ожидаемым |  |  |  |

* 1. Создание пользовательской документации и руководства по эксплуатации

Общая информация: сайт "Доброе сердце" представляет собой личный кабинет пользователя благодетеля. Система позволяет пользователям управлять своими пожертвованиями, просматривать актуальные проекты, изменять персональные данные.

Регистрация и авторизация:

Для новых пользователей доступна форма регистрации, где необходимо заполнить:

* Логин (уникальное имя для входа)
* Имя
* Номер телефона
* Пароль

После регистрации или при повторном посещении сайта используйте форму авторизации, введя свой логин и пароль. После успешного входа вы будете перенаправлены в личный кабинет.

Личный кабинет пользователя:

Личный кабинет состоит из:

1. Секции “Мои пожертвования”
2. Личных данных пользователя
3. И секции “Выход”

Выбор проекта:

После выбора благотворительного фонда вы можете перечислить средства на указанные реквизиты.

Для защиты ваших данных не передавайте логин и пароль третьим лицам. По окончании работы с личным кабинетом обязательно нажмите кнопку "Выйти", особенно при использовании общедоступных компьютеров.

Техническая поддержка:

При возникновении вопросов или проблем с работой сайта обращайтесь в техническую поддержку по контактам, указанным в заголовке сайта. В сообщении укажите суть проблемы и, по возможности, приложите скриншоты.

Функционал сайта "Доброе сердце" – пошаговое руководство

1. Авторизация и регистрация

Вход в систему: пользователь переходит на страницу авторизации, вводит логин и пароль, затем нажимает кнопку "Войти". После успешной проверки данных система перенаправляет в личный кабинет.

Регистрация нового пользователя: для создания учетной записи необходимо заполнить форму: указать логин, имя, фамилию, номер телефона и пароль. После нажатия кнопки "Зарегистрироваться" данные сохраняются, и пользователь может войти в системы

2. Личный кабинет пользователя

Личные данные: в этом разделе отображается информация о пользователе: логин, имя, фамилия, телефон и пароль. Для внесения изменений нужно отредактировать поля и нажать кнопку "Редактировать". Выход из системы осуществляется кнопкой "Выйти".

Выбор проектов: здесь представлены доступные благотворительные организации с описанием цели сбора средств.

Для пожертвования пользователю надо нажать на кнопку пожертвование и ввести сумму которую он хочет отправит.

# ЗАКЛЮЧЕНИЕ

Курсовой проект на тему «Проектирование, дизайн и программная реализация модуля управления благотворительными организациями» успешно завершён. В ходе работы достигнуты все поставленные цели и задачи. Были разработаны и внедрены ключевые компоненты системы, включая базу данных, пользовательский интерфейс и функциональные возможности для эффективного управления деятельностью благотворительных организаций.

Основные результаты проекта:

1. Анализ и проектирование: Проведен анализ существующих решений в области управления благотворительными организациями, что позволило сформулировать требования к модулю. Были разработаны диаграммы (IDEF0, UML, DFD), которые наглядно отображают архитектуру системы, процессы взаимодействия компонентов и жизненный цикл заявок.
2. Реализация модуля:

* Создана база данных на MySQL, включающая таблицы для пользователей, тарифов и услуг, с четко определенными связями между ними.
* Разработан веб-интерфейс с использованием HTML, CSS, PHP и SQL, обеспечивающий удобное взаимодействие пользователей с системой. Интерфейс включает личный кабинет, формы авторизации/регистрации.
* Реализованы функциональные возможности, такие как пожертвование, редактирование персональных данных, мониторинг новых проектов.

1. Тестирование: проведено тестирование модуля, включая проверку корректности работы форм, обработки данных и взаимодействия с базой данных. Результаты тестирования подтвердили соответствие системы заявленным требованиям.
2. Документация: подготовлены пользовательская документация и руководство по эксплуатации для облегчения внедрения и использования системы.

Выводы: Курсовой проект успешно подтвердил разработку модуля управления благотворительными организациями, который полностью соответствует современным стандартам автоматизации и обеспечивает высокий уровень пользовательского удобства.

Разработанное решение демонстрирует высокую эффективность в администрировании ключевых процессов, включая управление тарифами, ведение клиентской базы данных и контроль сервисов. Это подтверждается проведенными тестами, а также наглядной визуализацией бизнес-процессов с помощью диаграмм IDEF0, UML и DFD, которые отражают логику работы системы и её надежность.

Гибкая и модульная архитектура обеспечивает простую масштабируемость, позволяя легко адаптировать систему под новые требования и расширять её функционал. Интуитивно понятный интерфейс значительно снижает порог вхождения для пользователей и администраторов, делая работу с системой комфортной и продуктивной.

Проект полностью готов к внедрению в реальную инфраструктуру благотворительных организаций и обладает значительным потенциалом для дальнейшего развития. Среди перспективных направлений — интеграция с платежными системами для автоматизации финансовых операций, а также внедрение расширенной аналитики для глубокого анализа эффективности благотворительных программ.

Таким образом, разработанный модуль представляет собой законченное, надежное и удобное решение, способное значительно оптимизировать работу благотворительных организаций и открывающее широкие возможности для будущих улучшений.

# СПИСОК ИСПОЛЬЗОВАННЫх ИСТОЧНИКОВ

1. Котлинский С.В. Разработка моделей предметной области автоматизации: учебник для вузов/ С.В. Котлинский. – Санкт-Петербург: Лань, 2021. – 412с.
2. Вейцман В.М. Проектирование информационных систем: учебное пособие для вузов/ В.М. Вейцман. – 2-е издание, – Санкт-Петербург: Лань, 2022. – 316с.
3. Федорова Г.И. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности. Учебное пособие. Изд.: КУРС, ИнфраМ. Среднее профессиональное образование. 2020 г. 336 стр.
4. Никсон Робин. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 6-е изд. Питер, 2023. — 832 с.: ил. — (Серия «Бестселлеры O’Reilly»)
5. Изучаем РНР 7: руководство по созданию интерактивных веб-сайтов.: Пер. с англ. — СпБ.: ООО “Альфа-книга”, 2020. — 464 с.: ил. — Парал. тит. англ.
6. IT – студия SoftTime [Электронный ресурс]: Учебник по PHP 4 – Режим доступа: <http://www.softtime.ru/bookphp> – 20.03.2025.
7. HTML5 Book [Электронный ресурс]: Учебник по HTML и CSS – Режим доступа: <http://html5book.ru> – 20.03.2025.
8. Кодинг и вебмастеринг [Электронный ресурс]: Изучаем PHP и MySQL правильно PHP – Режим доступа: https://webstudio-uwk.ru/izuchaem-php-i-mysql- pravilno-php/ – 02.03.2025.
9. Яндекс Дзен [Электронный ресурс]: Обзор Visual Studio Code – Режим доступа: <https://dzen.ru/a/XUV-JQO91ACto8_d> – 04.03.2025.
10. Электронный справочник CSS [Электронный ресурс]: Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/CSS> 11.02.2025.