Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

ОТЧЁТ

по контрольной работе №1

по дисциплине

«Интернет-технологии и веб-программирование»

Вариант 15

|  |  |  |
| --- | --- | --- |
| Выполнил |  | ст. гр. 220604  А.И. Михнюк |
| Проверила |  | Н.В. Хаджинова |

Минск 2025

1. ЦЕЛЬ РАБОТЫ

Основной целью данной контрольной работы являлось практическое закрепление ключевых аспектов back-end разработки через создание динамических веб-приложений, взаимодействующих с базой данных. Работа была направлена на комплексное освоение всего жизненного цикла разработки - от проектирования структуры данных до развертывания готового решения в контейнеризированной среде.

В рамках работы предстояло разработать два взаимосвязанных, но функционально самостоятельных веб-приложения: базовую "Систему управления задачами" и специализированный "Каталог рецептов" по индивидуальному варианту №15. Такой подход позволил не только освоить типовые паттерны разработки, но и научиться адаптировать общую архитектуру под специфические требования предметной области.

Особое внимание уделялось проектированию реляционной структуры базы данных, где требовалось создать оптимальную схему хранения данных как для универсальных задач, так и для специализированных кулинарных рецептов с их уникальными атрибутами. Важной составляющей стала организация безопасного взаимодействия между PHP-скриптами и СУБД MySQL с использованием современных методов защиты от потенциальных уязвимостей.

Ключевой практической задачей являлась реализация полного цикла CRUD-операций, обеспечивающих базовый функционал управления данными. Это включало создание интуитивного пользовательского интерфейса для добавления, просмотра, редактирования и удаления записей, а также формирование динамических HTML-страниц на основе актуальных данных из базы.

Значимой частью работы стало освоение технологий контейнеризации приложений средствами Docker. Предстояло настроить изолированные среды выполнения для каждого приложения, обеспечив их корректное взаимодействие с базами данных и веб-серверами, что соответствует современным подходам к развертыванию веб-приложений в производственной среде.

Таким образом, работа охватывала весь спектр задач back-end разработчика - от проектирования данных и программирования бизнес-логики до обеспечения безопасности и деплоя готового решения, что способствовало формированию целостного понимания процесса создания веб-приложений.

1. ХОД РАБОТЫ
   1. Установка и настройка локального сервера, подготовка контейнеризированной среды

Первоначальная разработка веб-приложений осуществлялась в среде Windows с использованием установленного пакета PHP, интегрированного в системные переменные операционной системы. В качестве основной среды разработки была выбрана IDE Visual Studio Code, которая предоставляла широкие возможности для работы с PHP-кодом. Для организации локального сервера использовался плагин PHP Server, который позволял быстро запускать и тестировать приложения в процессе разработки. Этот подход обеспечивал простоту отладки и мгновенное применение изменений, однако имел ограничения при необходимости воспроизведения производственного окружения.

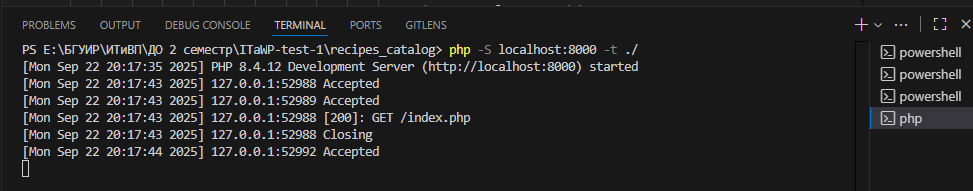


Рисунок 1 – Запуск PHP приложения через терминал

В ходе работы возникла необходимость обеспечения переносимости приложений и идентичности сред разработки и развертывания, что привело к решению о переходе на контейнеризированную среду Docker.

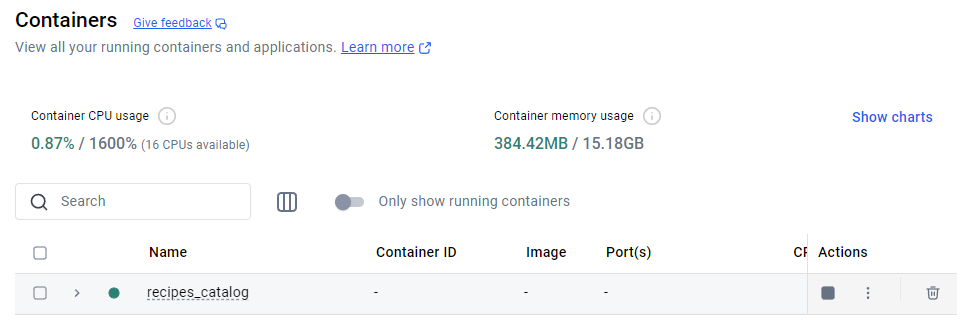
****

Рисунок 2 – Менеджер контейнеров Docker Desktop

Установка Docker Desktop позволила создать изолированные контейнеры для каждого компонента системы. Для описания инфраструктуры был разработан docker-compose файл, который определял взаимодействие между сервисами PHP-приложения, базы данных MySQL и административной панели phpMyAdmin.

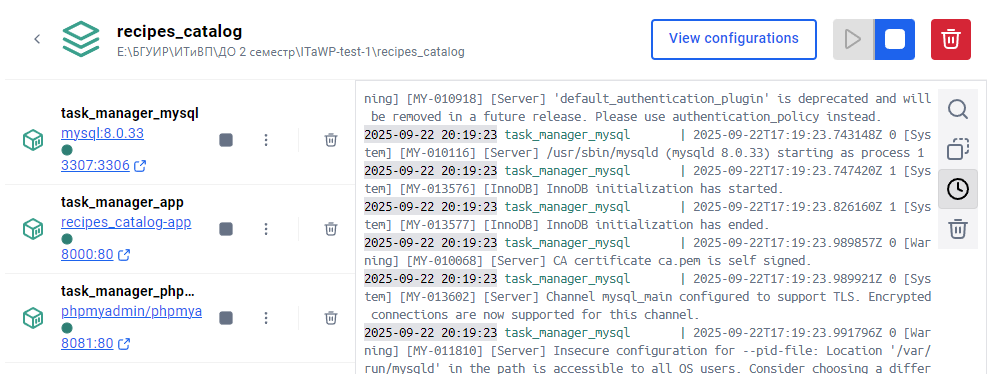


Рисунок 2 – Содержимое кластера в Docker Desktop

Конфигурация docker-compose.yml была тщательно продумана для обеспечения надежности работы системы. Для сервиса MySQL был выбран образ версии 8.0.33 с явным указанием метода аутентификации native\_password для совместимости с PHP-приложением. Особое внимание было уделено настройке healthcheck для базы данных - реализована периодическая проверка доступности MySQL через утилиту mysqladmin с интервалом 20 секунд и количеством попыток 10, что гарантировало корректный порядок запуска зависимых сервисов. Для обеспечения сохранности данных использовался named volume mysql\_data, который монтировался в стандартную директорию хранения данных MySQL /var/lib/mysql.

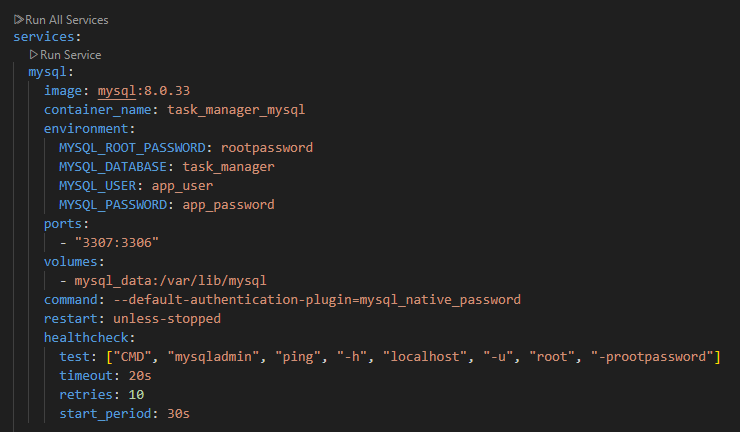


Рисунок 3 – Монтирование данных внутри Docker Desktop

Версия PHP 8.1 была выбрана как стабильная и поддерживающая все необходимые расширения для работы с базой данных. Dockerfile включал установку критически важных расширений pdo, pdo\_mysql и mysqli, без которых было невозможно организовать взаимодействие с СУБД. Особое значение имела активация модуля rewrite для Apache через команду a2enmod rewrite, что позволяло корректно обрабатывать URL-адреса в веб-приложении. Копирование файлов приложения в директорию /var/www/html/ и установка прав доступа для пользователя www-data обеспечивали правильное функционирование веб-сервера.

Важным аспектом конфигурации стало решение проблемы занятых портов в локальной среде. Стандартный порт MySQL 3306 был переназначен на 3307 для приложения управления задачами, в то время как для каталога рецептов использовался порт 3306 в изолированном окружении. Аналогично были переназначены порты для phpMyAdmin (8081 и 8082) и веб-интерфейсов приложений (8000 и 8001), что позволило избежать конфликтов и обеспечить параллельную работу обоих проектов. Зависимости между сервисами были организованы через параметр depends\_on с условием condition: service\_healthy, что гарантировало запуск PHP-приложения только после полной инициализации базы данных.

Настройка политики перезапуска restart: unless-stopped обеспечивала автоматическое восстановление работы сервисов при нештатных ситуациях. В процессе настройки возникли сложности с монтированием томов в среде Windows из-за различий в файловых системах, что потребовало перехода на использование внутренних docker volumes вместо локального монтирования директорий. Это решение повысило стабильность работы и устранило проблемы с путями, характерные для Windows-окружения.

Преимущества контейнеризации проявились в значительном упрощении процесса развертывания и обеспечения идентичности сред. Docker-контейнеры позволили упаковать приложения со всеми зависимостями, исключив проблемы совместимости версий PHP и расширений. Использование docker-compose обеспечило оркестрацию многоконтейнерных приложений и автоматизацию запуска всей инфраструктуры одной командой. Изоляция сервисов в отдельных контейнерах повысила безопасность и стабильность работы, а также упростила масштабирование приложений в будущем.

Для функционирования системы управления контентом требуется локальный сервер, обеспечивающий обработку *PHP*-кода, работу с базами данных и корректное отображение веб-страниц. В качестве программного решения был выбран *XAMPP* – комплексный пакет, включающий веб-сервер *Apache*, интерпретатор *PHP*, СУБД *MySQL* и дополнительные инструменты для разработки.

*XAMPP* обладает рядом значимых преимуществ, делающих его оптимальным выбором для учебных и коммерческих проектов. Кроссплатформенность позволяет развертывать серверную среду на операционных системах *Windows*, *Linux* и *macOS* без существенных изменений в конфигурации. Интегрированный набор компонентов избавляет от необходимости их раздельной установки и настройки взаимодействия между ними. В состав пакета включен *phpMyAdmin* – веб-приложение для визуального управления базами данных через интуитивно понятный интерфейс.

Процесс развертывания локального сервера начинается с загрузки официального дистрибутива с сайта разработчика и его установки с параметрами по умолчанию. После инсталляции осуществляется запуск основных служб – веб-сервера *Apache* и системы управления базами данных *MySQL* через панель управления *XAMPP*. Проверка работоспособности выполняется путем обращения к локальному хосту через браузер, где отображается тестовая страница и интерфейс *phpMyAdmin*.

В настройках веб-сервера указывается корневая директория размещения файлов проекта, в параметрах *PHP* активируются требуемые модули и расширения. Особое внимание уделяется настройке соединения с базой данных – создаются новые пользователи и таблицы, проверяются права доступа. После завершения скачивания будет предложено запустить панель управления, продемонстрируем ее на рисунке 1.

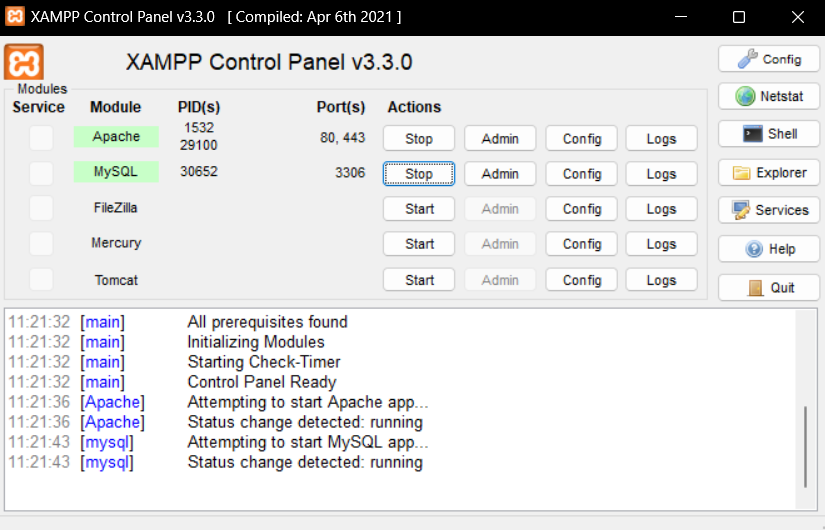


Рисунок 1 – Панель управления

Для дальнейшей работы необходимо организовать рабочего окружения. Для этого в корневой каталог веб-сервера *XAMPP* (обычно это папка *htdocs*) создается отдельная папка проекта *gift\_manager* (рисунок 2). Данное действие является необходимым, поскольку веб-сервер *Apache*, входящий в состав *XAMPP*, обрабатывает и делает доступными для браузера именно те файлы, которые расположены в его корневом каталоге и его поддиректориях. Размещение всех рабочих файлов проекта (*HTML*, *PHP*, *CSS*-скриптов) в этой специально созданной папке гарантирует, что они будут корректно интерпретироваться сервером, а само приложение будет доступно по соответствующему локальному адресу в сети.



Рисунок 2 – Организация рабочего пространства

На следующем этапе разработки была развёрнута и подготовлена база данных. В отличие от использования графических инструментов вроде *phpMyAdmin*, работа велась напрямую с сервером *MySQL* через консольный интерфейс представленной СУБД, что отображено на рисунке 3.

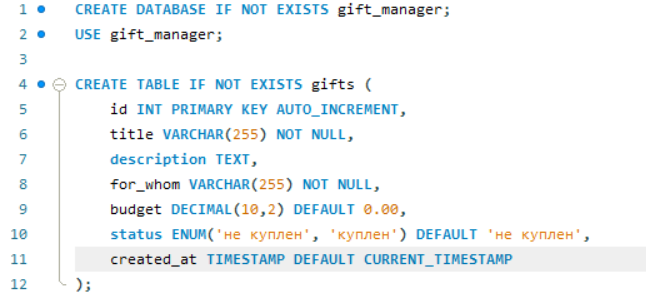


Рисунок 3 – Структура базы данных

В данном случае для поля статуса подарка было использовано перечисление, для того, чтобы избежать дополнительных проверок во время валидации полей.

После создания структуры базы данных следующим шагом является организация программного подключения к ней из *PHP*-кода. Для этого создается отдельный файл *config*.*php*, который содержит все необходимые параметры для установки соединения с СУБД *MySQL*. Этот файл служит централизованным местом для хранения учетных данных и настроек подключения, что упрощает их дальнейшее изменение и обслуживание.

В файле *config*.*php* определяются основные параметры: хост (*localhost*), указывающий на то, что сервер базы данных расположен на том же компьютере, что и веб-сервер; имя базы данных (*gift\_manager*), соответствующее созданной базе; а также логин и пароль пользователя СУБД. По умолчанию после установки *MySQL* создается пользователь *root* с полными правами доступа и без пароля, что характерно для локальной среды разработки, такой как *XAMPP*. Именно поэтому в настройках указана пустая строка в качестве пароля. Однако в реальных проектах использование учётной записи *root* не рекомендуется из-за соображений безопасности. Пароль для этого пользователя можно задать с помощью *SQL*-команды *ALTER* *USER* '*root*'@'*localhost*' *IDENTIFIED* *BY* 'новый*\_*пароль';, после чего его необходимо будет указать в соответствующей переменной файла конфигурации.

Для установки подключения используется технология *PDO* (*PHP* *Data* *Objects*), которая предоставляет универсальный интерфейс для работы с различными базами данных. В коде инициализируется новый объект *PDO*, в конструктор которого передаются строка с указанием хоста, имени базы и кодировки (*utf*8), а также логин и пароль. Дополнительно настраиваются важные атрибуты: режим ошибок устанавливается в значение *ERRMODE\_EXCEPTION*, что заставляет *PDO* генеривать исключения при возникновении проблем, что значительно упрощает отладку. Также задается режим выборки данных по умолчанию (*FETCH\_ASSOC*), при котором результаты запросов возвращаются в виде ассоциативных массивов. Весь код помещается в блок *try*...*catch* для перехвата возможных исключений, связанных с подключением. В случае ошибки выводится сообщение и выполнение скрипта прерывается.

Главная страница приложения (*index*.*php*) представляет собой динамический веб-документ, который выполняет ключевую функцию отображения всего списка подарков из базы данных в удобном и визуально организованном виде. Её работа начинается с подключения файла конфигурации *config*.*php*, который устанавливает соединение с базой данных *MySQL* через *PDO*. Это подключение является фундаментальным, так как позволяет выполнять *SQL*-запросы непосредственно из *PHP*-кода. После установки соединения скрипт проверяет параметры *URL*, переданные методом *GET*, чтобы отобразить пользователю соответствующие сообщения: например, уведомление об успешном удалении записи или об изменении статуса подарка, либо же сообщение об ошибке, если в процессе выполнения предыдущих операций возникли проблемы.

Основная серверная логика заключается в выполнении *SQL*-запроса *SELECT* \* *FROM* *gifts* *ORDER* *BY* *created\_at* *DESC*, который получает все записи из таблицы *gifts*, упорядочивая их по дате создания в порядке убывания. Это обеспечивает то, что самые свежие добавленные подарки появляются вверху списка. Результаты запроса сохраняются в переменную $*gifts* для последующего использования в *HTML*-разметке. Важно отметить, что весь код, работающий с базой данных, заключён в блок *try*...*catch*, что позволяет перехватывать и обрабатывать исключения, возникающие на уровне СУБД, и *gracefully* выводить сообщения об ошибках без прерывания работы приложения.

Для создания современного, адаптивного и эстетически привлекательного интерфейса страница активно использует внешние библиотеки и ресурсы. Фреймворк *Bootstrap* 5, подключаемый через *CDN*, предоставляет готовые компоненты и сетку для вёрстки, что позволяет быстро создавать интерфейсы, корректно отображающиеся на различных устройствах – от мобильных телефонов до настольных компьютеров. Библиотека иконок *Font* *Awesome*, также загружаемая через *CDN*, используется для добавления интуитивно понятных визуальных элементов, таких как иконки подарков, календаря, денег и действий (редактирование, удаление). Дополнительно подключается пользовательская таблица стилей (*style*.*css*), которая позволяет применять собственное оформление, дополняющее стандартные стили *Bootstrap* – например, определяет кастомные цвета для кнопок и навигационной панели.

Структура страницы включает несколько смысловых блоков: навигационную панель в верхней части экрана, которая содержит название приложения, текущую дату и кнопку для добавления нового подарка; интерактивную карусель (слайдер), выполняющую роль визуального акцента; область для вывода системных сообщений (успех или ошибка); и, наконец, сетку карточек, в которых отображаются сами данные. Каждая карточка динамически генерируется *PHP*-скриптом на основе данных, полученных из базы. В карточке выводится название подарка, его получатель, описание, бюджет и статус («куплен» или «не куплен»), причём статус визуально выделяется цветным значком.

Также в карточке присутствуют кнопки для управления: редактирования, изменения статуса и удаления записи (рисунок 4).

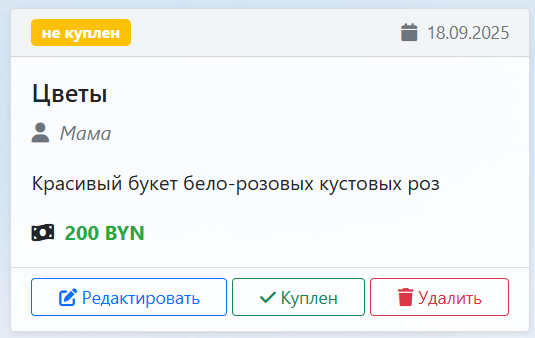


Рисунок 4 – Описание карточки подарка

Для повышения удобства использования и предотвращения случайных действий на странице реализовано модальное окно подтверждения, которое вызывается при попытке выполнить потенциально критическое действие (удаление или изменение статуса), что отображено на рисунке 5.

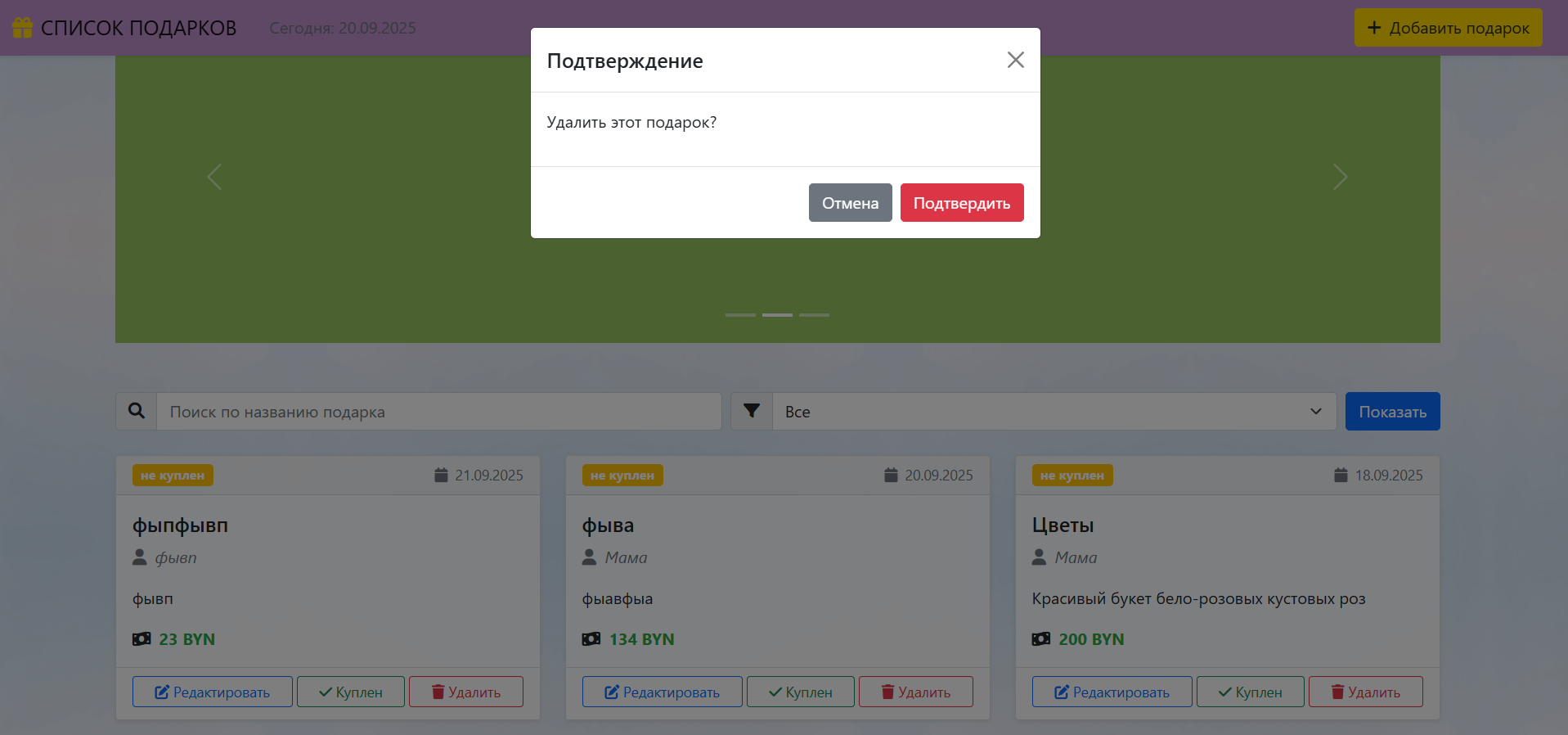


Рисунок 5 – Выполнение критического действия на странице

Стоит отметить, что на странице добавлена поисковая строка и фильтрация для улучшения пользовательского опыта.

Страница добавления подарка (*add*.*php*) логически продолжает архитектуру главной страницы, реализуя операцию «*Create*» из *CRUD* и демонстрируя принцип обработки пользовательского ввода на стороне сервера. Её работа начинается с подключения общего файла конфигурации *config*.*php*, что обеспечивает доступ к установленному соединению с базой данных через объект *PDO*. Основная логика страницы активируется при получении *POST*-запроса, что означает отправку пользователем формы. Скрипт извлекает переданные данные из суперглобального массива $*\_POST*, предварительно обрабатывая их: обрезая лишние пробелы функцией *trim* и преобразуя бюджет в число с плавающей точкой. Затем выполняется валидация обязательных полей – названия подарка и поля «для кого», а также проверка на неотрицательность бюджета. Если ошибки отсутствуют, формируется подготовленный *SQL*-запрос *INSERT* с использованием метода *prepare* объекта *PDO*, что обеспечивает безопасность от *SQL*-инъекций. Значения подставляются через плейсхолдеры и передаются методу *execute* в виде массива. При успешном выполнении запроса пользователю показывается сообщение об успехе (рисунок 6), а поля формы очищаются; в случае ошибки базы данных или валидации выводится соответствующее сообщение.

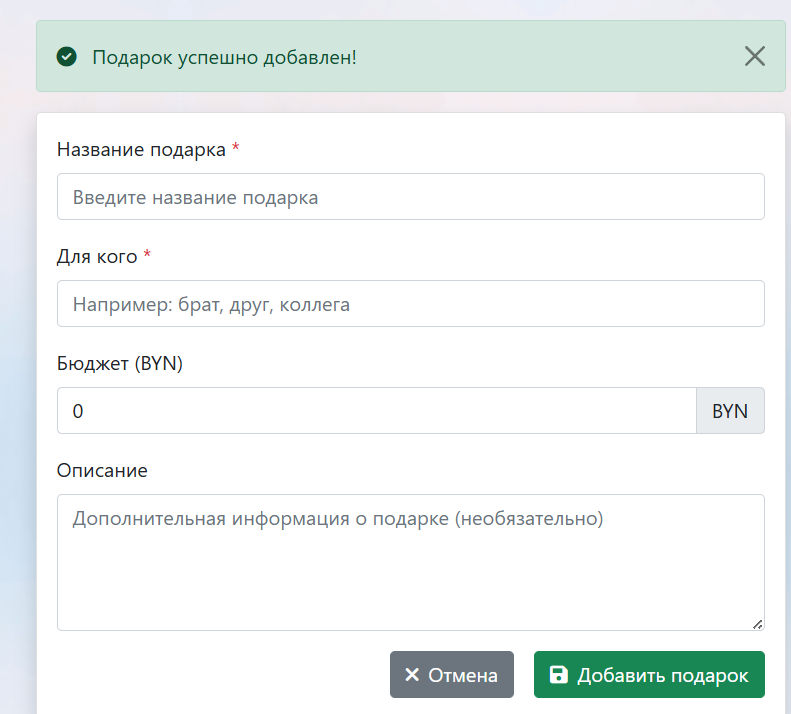


Рисунок 6 – Успешное добавление нового подарка

Визуальное оформление страницы полностью выдержано в едином стиле с главной страницей, что создает целостное восприятие приложения. Подключаются те же внешние ресурсы: *Bootstrap* 5 для адаптивной сетки и компонентов, *Font* *Awesome* для иконок и пользовательский файл стилей *style*.*css* для кастомных элементов, таких как цветовая схема навигационной панели (*bg*-*purple*-*custom*) и кнопок (*btn*-*yellow*-*custom*). Навигационная панель в верхней части экрана *mirrors* структуру главной страницы. Важно отметить, что для обеспечения безопасности вывода данные, введенные пользователем, при повторном отображении в полях формы экранируются функцией *htmlspecialchars*, что предотвращает *XSS*-атаки. Сообщения о результате операции (успех или ошибка) выводятся в стилизованных под тип *alert*-блоках, которые гармонично вписываются в общий дизайн. Таким образом, страница добавления подарка не только функционально выполняет свою задачу по сохранению данных в базу, но и обеспечивает пользователю удобный, безопасный и визуально согласованный интерфейс, что подчеркивает профессиональный подход к разработке веб-приложения.

Страница редактирования подарка (*edit*.*php*) реализует операцию «*Update*» из *CRUD* и демонстрирует комплексный подход к работе с данными: она сначала загружает существующую запись из базы данных по переданному идентификатору, валидируя его корректность и существование записи, а затем обрабатывает отправленную пользователем форму, выполняя валидацию введенных данных и обновляя запись в базе с помощью подготовленного *SQL*-запроса *UPDATE*, что обеспечивает защиту от инъекций. Визуально страница полностью соответствует общему стилю приложения: используется та же навигационная панель.

Файл .*gitignore* является важным элементом организации проекта в системе контроля версий *Git*, поскольку он позволяет целенаправленно исключать из отслеживания и последующей загрузки в удалённый репозиторий те файлы и директории, которые не должны в нём находиться. Это решает две ключевые задачи: во-первых, предотвращает избыточное копирование служебных или временных файлов, таких как локальные настройки редактора (.*vscode*/, .*idea*/), временные файлы (\*.*tmp*, \*.*temp*), логи (\*.*log*) или системные артефакты (.*DS\_Store*, *Thumbs*.*db*), что делает репозиторий чистым и содержащим только исходный код и необходимые ресурсы. Во-вторых, и это критически важно с точки зрения безопасности, он позволяет скрыть конфиденциальные данные, такие как файлы с параметрами подключения к базе данных (*config*.*php*) или *SQL*-скрипты для её инициализации (*database\_setup*.*sql*), которые могут содержать чувствительную информацию – логины, пароли, названия баз данных.

Заключительным этапом разработки проекта стало размещение всего кода в удаленном репозитории на платформе *GitHub* для обеспечения сохранности, контроля версий и возможности совместной работы. Для этого в корневой директории проекта была выполнена инициализация репозитория *Git* командой *git* *init*, которая создала необходимую скрытую структуру для отслеживания изменений. Затем все файлы проекта были добавлены в индекс (*staging* *area*) с помощью команды *git* *add* ., что подготовило их к фиксации. Следующим шагом был создан коммит с описанием выполненной работы командой *git* *commit* -*m* "*First* *commit* *gift\_manager* *application*...", который зафиксировал текущее состояние кода в истории версий. Финальным действием стало отправление всех закоммиченных изменений в удаленный репозиторий на *GitHub* командой *git* *push* *origin* *main*, что обеспечило надежное хранение кода в облаке и доступ к нему из любого места.

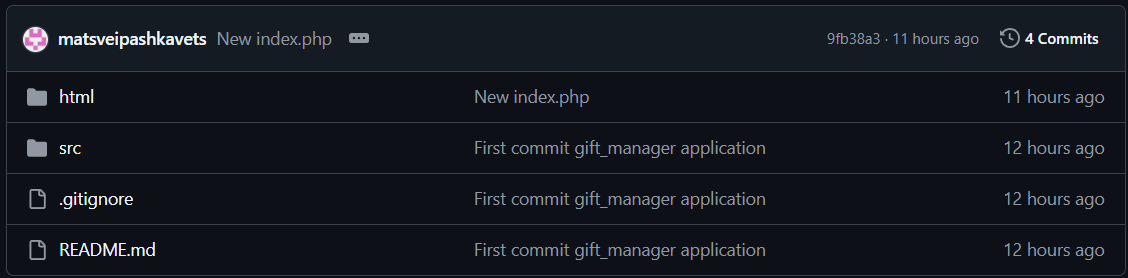


Рисунок 7 – Содержимое репозитория

Стоит отметиться, что данный коммит был выполнен в соответствии с лучшими практиками контроля версий, поскольку содержит четкое и информативное сообщение, описывающее суть вносимых изменений. Это позволяет в будущем легко идентифицировать назначение данной версии кода и упрощает навигацию по истории проекта.

* 1. Проверка функционала и адаптивности

После завершения разработки сайта фитнес центра проведена комплексная проверка его работоспособности и адаптивности на различных устройствах. Особое внимание уделялось корректному отображению всех элементов интерфейса на мобильных устройствах, так как значительная часть пользователей посещает сайт со смартфонов и планшетов.

Для достижения кроссбраузерной адаптивности был использован фреймворк *Bootstrap*, который предоставляет готовую систему адаптивных классов, основанных на медиазапросах и технологии *Flexbox*. Встроенные механизмы фреймворка автоматически обеспечивают корректное отображение всех элементов интерфейса на различных разрешениях экрана.

Фрагмент кода (класс *bootstrap*):

<*a* *href*="*delete*.*php*?*id*=<?= $*gift[*'*id*'*]* ?>"

*class*="*btn* *btn*-*outline*-*danger* *btn*-*sm* *mb*-1 *flex*-*grow*-1 *confirmation*-*link*" *data*-*message*="Удалить этот подарок?">

<*i* *class*="*fas* *fa*-*trash* *me*-1"></*i*>Удалить

</*a*>

Продемонстрируем страницы веб-приложения с учетов адаптивности, что отобразим на рисунках 8 – 13.

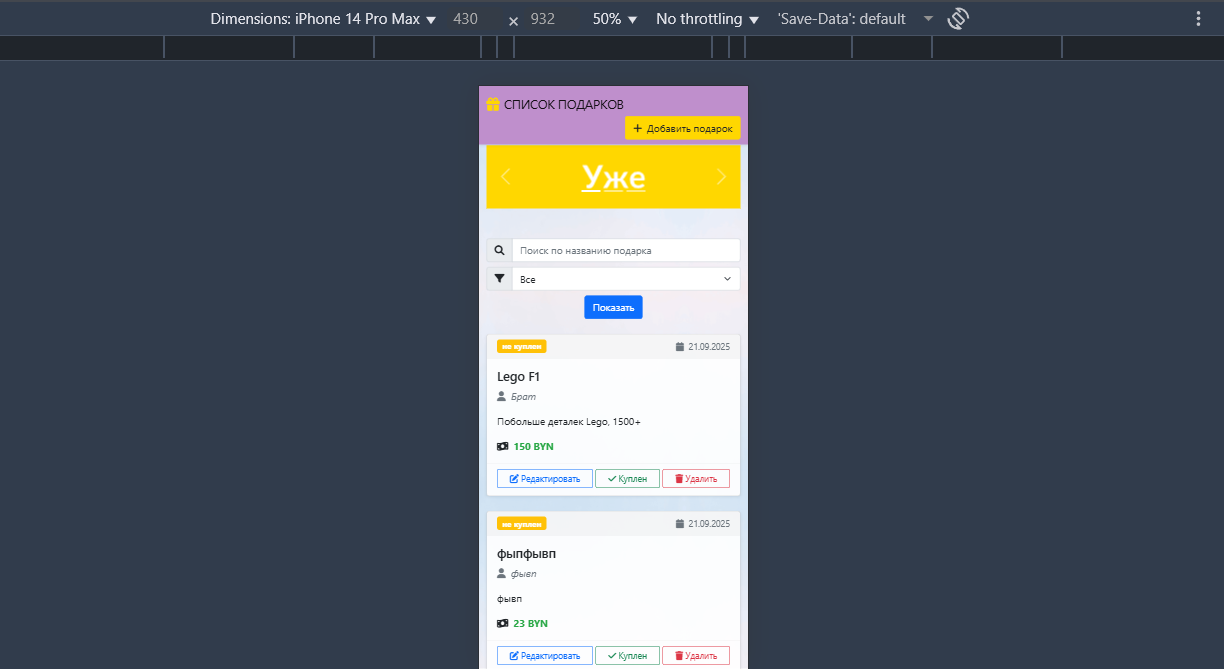


Рисунок 8 – Приветствующая страница

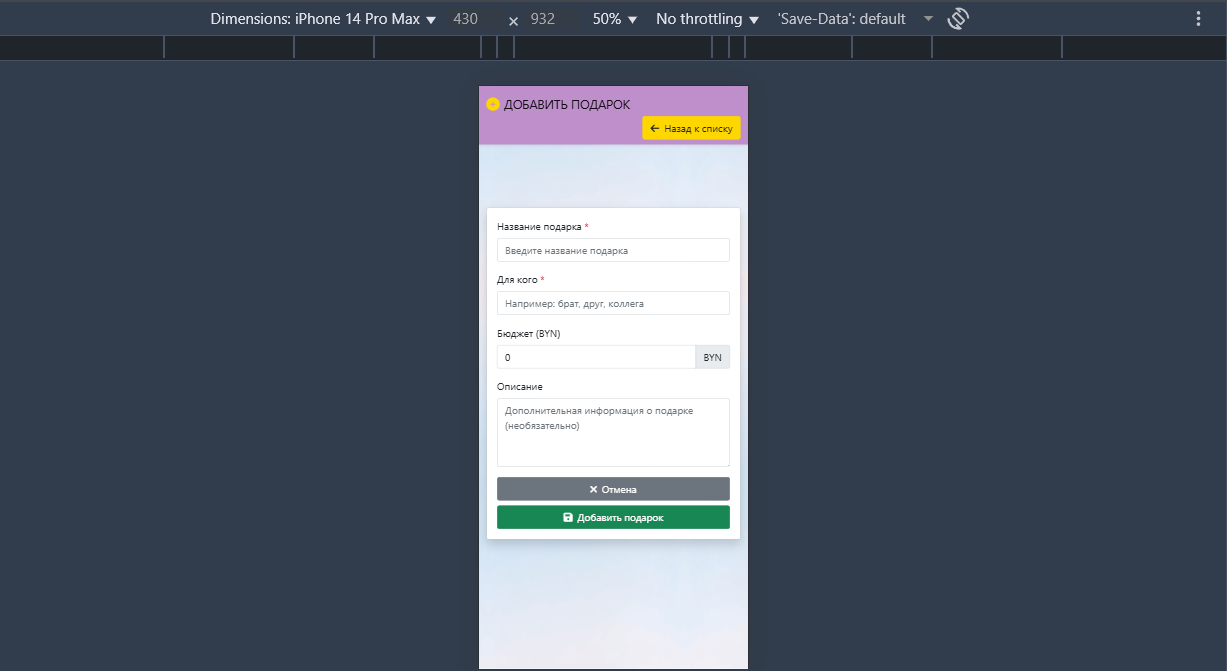


Рисунок 9 – Страница добавления подарка

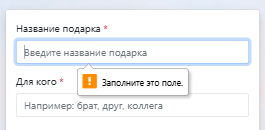


Рисунок 10 – Выполнена валидация

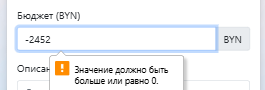


Рисунок 11 – Выполнена валидация

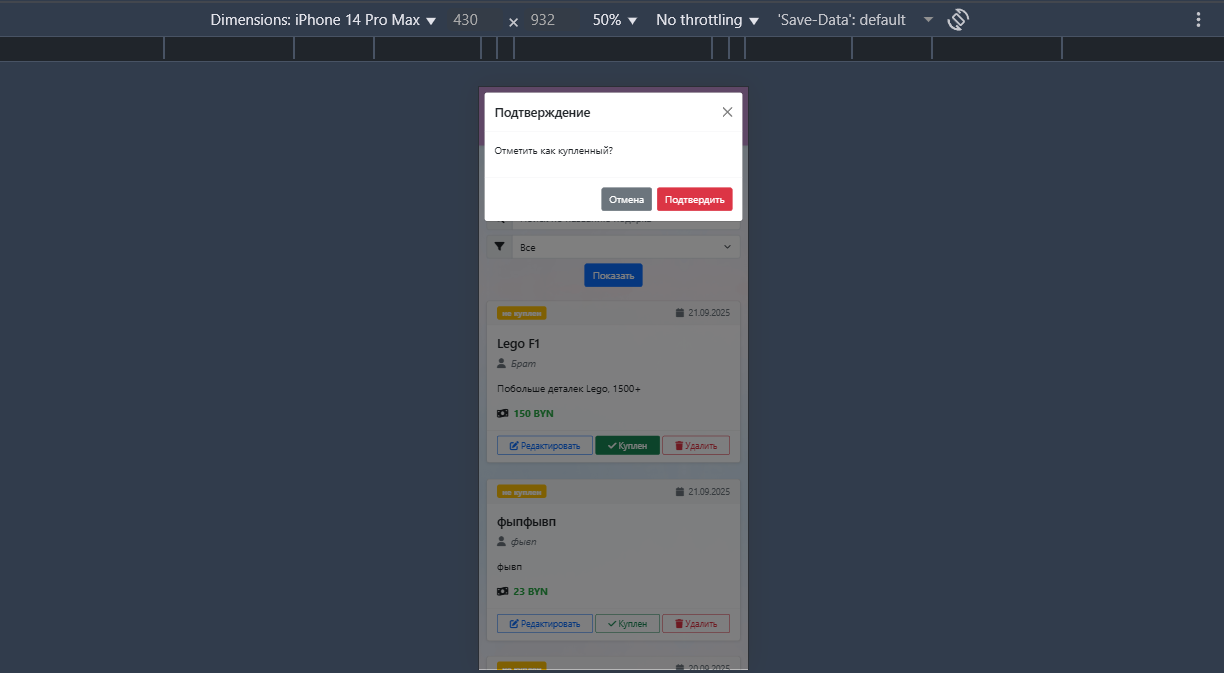


Рисунок 12 – Изменение статуса подрака

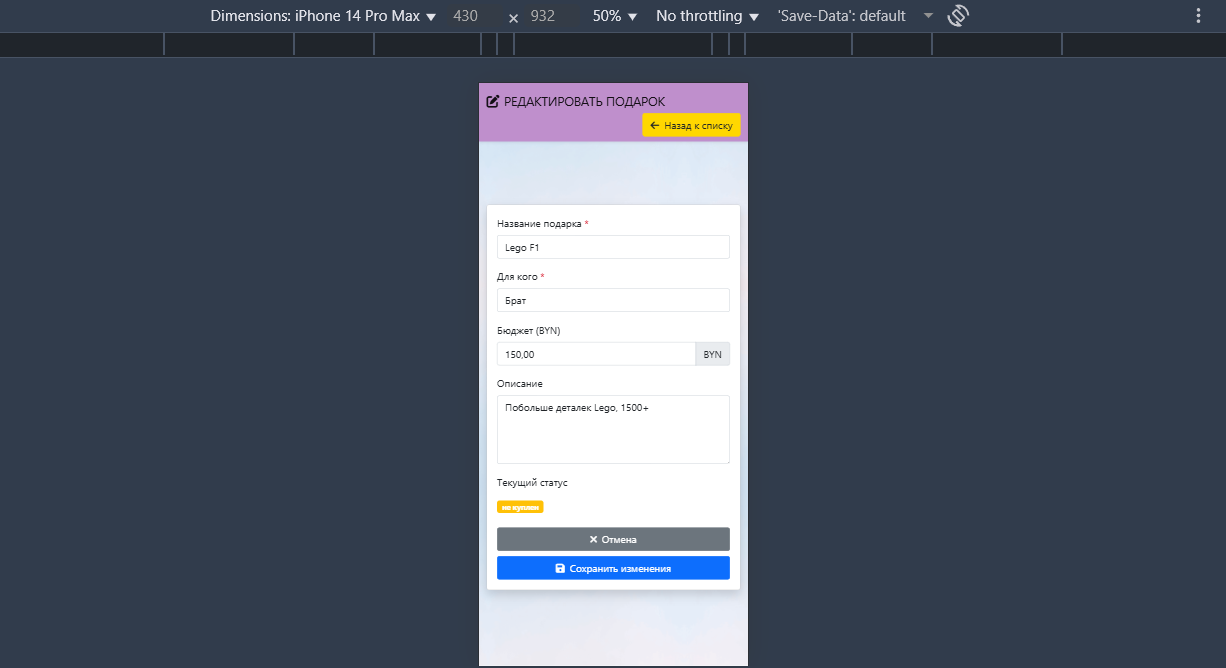


Рисунок 13 – Редактирование подрака

Тестирование проводилось на реальном устройстве *IPhone* 14 *Pro* *Max*.

* 1. Ответ на контрольный вопрос

Система управления базами данных *MySQL* предлагает разнообразные типы полей, каждый из которых служит для эффективного хранения определённого вида информации. Одним из наиболее распространенных типов является *INT* (*Integer*), предназначенный для работы с целыми числами. Этот тип часто используется для идентификаторов, счетчиков или любых числовых значений без дробной части, поддерживая как отрицательные, так и положительные числа в широком диапазоне.

Для хранения текстовых данных переменной длины применяется тип *VARCHAR*, который позволяет экономить место на диске, занимая только объем, необходимый для фактического содержимого строки. Он идеален для таких полей, как имена, адреса электронной почты или короткие описания, где длина текста может меняться.

Если же требуется сохранить большой объем текста, например, статьи, комментарии или подробные описания, лучше использовать тип *TEXT*. Он предоставляет возможность хранить до 65 тысяч символов без необходимости предварительного указания точной длины, что делает его незаменимым для работы с крупными текстовыми блоками, например, такими как описание.

Для фиксации временных меток и дат существует тип *DATETIME*, который сохраняет значения в формате «год-месяц-день час:минута:секунда». Этот тип удобен для регистрации времени событий, таких как момент создания записи или последнего обновления, обеспечивая точность и удобство при работе с временными интервалами.

Наконец, тип *ENUM* позволяет ограничить возможные значения поля заранее определённым списком вариантов. Это полезно для полей с небольшим набором допустимых значений, например, статусов подарка или простых вариантов выбора, так как он не только обеспечивает целостность данных, но и экономит пространство за счёт хранения индексов значений вместо полных строк.

Правильный выбор типа данных не только оптимизирует хранение информации, но и повышает производительность запросов, обеспечивая надежность и эффективность работы всей базы данных.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной контрольной работы успешно закреплены фундаментальные навыки *back*-*end*-разработки, а также реализовано полноценное динамическое веб-приложение, интегрированное с системой управления базами данных *MySQL*. Основной акцент сделан на освоении полного цикла работы с данными: от проектирования реляционной структуры базы данных до организации программного доступа к ней с использованием *PHP*-скриптов.

Ключевым достижением стала практическая реализация операций *CRUD* (*Create*, *Read*, *Update*, *Delete*), которые обеспечили базовый функционал для управления информацией. В процессе разработки был создан интуитивно понятный интерфейс на основе веб-форм, позволяющий пользователям взаимодействовать с приложением, а также написаны *PHP*-скрипты, корректно обрабатывающие пользовательские запросы и трансформирующие их в соответствующие *SQL*-запросы.

Важным аспектом работы стало обеспечение безопасности и устойчивости приложения: использование подготовленных выражений (*PDO*) для предотвращения *SQL*-инъекций, валидация входных данных и обработка исключений. Кроме того, применен фреймворк *Bootstrap* для создания адаптивного интерфейса, обеспечивающего корректное отображение на различных устройствах.

В рамках работы достигнуты поставленные цели и получены практические навыки, необходимые для дальнейшей разработки сложных веб-приложений.

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Программный код приложения**

С полным кодом веб-приложения можно ознакомиться по ссылке:

[*https://github.com/Matveiko0oOoo/itivp*](https://github.com/Matveiko0oOoo/itivp)

***Файл index.php***

*<?php*

*require\_once '../dbConf/config.php';*

*$success\_message = '';*

*if (isset($\_GET['deleted']) && $\_GET['deleted'] == '1') {*

*$success\_message = 'Подарок успешно удален!';*

*} elseif (isset($\_GET['status\_updated']) && $\_GET['status\_updated'] == '1' && isset($\_GET['message'])) {*

*$success\_message = htmlspecialchars($\_GET['message']);*

*}*

*$error = '';*

*if (isset($\_GET['error'])) {*

*$error = htmlspecialchars($\_GET['error']);*

*}*

*$sql = "SELECT \* FROM gifts WHERE 1=1";*

*$params = [];*

*$search\_title = $\_GET['search\_title'] ?? '';*

*if (!empty($search\_title)) {*

*$sql .= " AND title LIKE ?";*

*$params[] = '%' . $search\_title . '%';*

*}*

*$filter\_for\_whom = $\_GET['filter\_for\_whom'] ?? '';*

*if (!empty($filter\_for\_whom)) {*

*$sql .= " AND for\_whom = ?";*

*$params[] = $filter\_for\_whom;*

*}*

*$sql .= " ORDER BY created\_at DESC";*

*try {*

*$stmt = $pdo->prepare($sql);*

*$stmt->execute($params);*

*$gifts = $stmt->fetchAll();*

*$stmt\_recipients = $pdo->query("SELECT DISTINCT for\_whom FROM gifts ORDER BY for\_whom");*

*$recipients = $stmt\_recipients->fetchAll(PDO::FETCH\_COLUMN);*

*} catch(PDOException $e) {*

*$error = "Ошибка при загрузке подарков: " . $e->getMessage();*

*$gifts = [];*

*$recipients = [];*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="ru">*

*<head>*

*<meta charset="UTF-8">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<title>Список подарков</title>*

*<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">*

*<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">*

*<link rel="stylesheet" href="../src/css/style.css">*

*</head>*

*<body>*

*<nav class="navbar navbar-expand-lg navbar-light bg-purple-custom fixed-top shadow-sm">*

*<div class="container-fluid">*

*<a class="navbar-brand" href="index.php">*

*<i class="fas fa-gift me-2"></i>СПИСОК ПОДАРКОВ*

*</a>*

*<div class="d-flex align-items-center me-auto ms-3">*

*<span class="navbar-text text-muted d-none d-md-block">*

*Сегодня: <?= date('d.m.Y') ?>*

*</span>*

*</div>*

*<a href="add.php" class="btn btn-yellow-custom ms-auto">*

*<i class="fas fa-plus me-2"></i>Добавить подарок*

*</a>*

*</div>*

*</nav>*

*<div class="container mt-4">*

*<div class="row mb-5">*

*<div class="col-12 text-center">*

*<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">*

*<div class="carousel-indicators">*

*<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>*

*<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>*

*<button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-label="Slide 3"></button>*

*</div>*

*<div class="carousel-inner">*

*<div class="carousel-item active">*

*<img class="d-block w-100" src="https://placehold.co/800x200/FFD700/white?text=Уже" alt="First slide">*

*</div>*

*<div class="carousel-item">*

*<img class="d-block w-100" src="https://placehold.co/800x200/96c561/white?text=купил" alt="Second slide">*

*</div>*

*<div class="carousel-item">*

*<img class="d-block w-100" src="https://placehold.co/800x200/775494/white?text=подарок?" alt="Third slide">*

*</div>*

*</div>*

*<button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">*

*<span class="carousel-control-prev-icon" aria-hidden="true"></span>*

*<span class="visually-hidden">Previous</span>*

*</button>*

*<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">*

*<span class="carousel-control-next-icon" aria-hidden="true"></span>*

*<span class="visually-hidden">Next</span>*

*</button>*

*</div>*

*</div>*

*</div>*

*<div class="row mb-4">*

*<div class="col-12">*

*<form action="index.php" method="GET" class="d-flex flex-column flex-md-row align-items-center justify-content-between">*

*<div class="input-group mb-2 mb-md-0 me-md-2 flex-grow-1">*

*<span class="input-group-text"><i class="fas fa-search"></i></span>*

*<input type="text" name="search\_title" class="form-control" placeholder="Поиск по названию подарка" value="<?= htmlspecialchars($search\_title) ?>">*

*</div>*

*<div class="input-group flex-grow-1 me-md-2">*

*<label class="input-group-text" for="filter\_for\_whom"><i class="fas fa-filter"></i></label>*

*<select class="form-select" id="filter\_for\_whom" name="filter\_for\_whom">*

*<option value="">Все</option>*

*<?php foreach ($recipients as $recipient): ?>*

*<option value="<?= htmlspecialchars($recipient) ?>" <?= $filter\_for\_whom === $recipient ? 'selected' : '' ?>>*

*<?= htmlspecialchars($recipient) ?>*

*</option>*

*<?php endforeach; ?>*

*</select>*

*</div>*

*<button type="submit" class="btn btn-primary mt-2 mt-md-0">Показать</button>*

*<?php if (!empty($search\_title) || !empty($filter\_for\_whom)): ?>*

*<a href="index.php" class="btn btn-outline-secondary mt-2 mt-md-0 ms-md-2">Сброс</a>*

*<?php endif; ?>*

*</form>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-12">*

*<?php if ($success\_message): ?>*

*<div class="alert alert-success alert-dismissible fade show" role="alert">*

*<i class="fas fa-check-circle me-2"></i><?= $success\_message ?>*

*<button type="button" class="btn-close" data-bs-dismiss="alert"></button>*

*</div>*

*<?php endif; ?>*

*<?php if ($error): ?>*

*<div class="alert alert-danger alert-dismissible fade show" role="alert">*

*<i class="fas fa-exclamation-triangle me-2"></i><?= $error ?>*

*<button type="button" class="btn-close" data-bs-dismiss="alert"></button>*

*</div>*

*<?php endif; ?>*

*<?php if (empty($gifts)): ?>*

*<div class="text-center py-5">*

*<i class="fas fa-gift fa-5x text-muted mb-3"></i>*

*<h3 class="text-muted">Пока нет подарков</h3>*

*<p class="text-muted">Добавьте первый подарок, нажав на кнопку выше</p>*

*</div>*

*<?php else: ?>*

*<div class="row">*

*<?php foreach ($gifts as $gift): ?>*

*<div class="col-md-6 col-lg-4 mb-4">*

*<div class="card gift-card h-100 shadow-sm">*

*<div class="card-header d-flex justify-content-between align-items-center">*

*<span class="badge status-badge <?= $gift['status'] === 'куплен' ? 'bg-success' : 'bg-warning' ?>">*

*<?= htmlspecialchars($gift['status']) ?>*

*</span>*

*<small class="text-muted">*

*<i class="fas fa-calendar me-1"></i>*

*<?= date('d.m.Y', strtotime($gift['created\_at'])) ?>*

*</small>*

*</div>*

*<div class="card-body">*

*<h5 class="card-title"><?= htmlspecialchars($gift['title']) ?></h5>*

*<p class="card-text text-muted">*

*<i class="fas fa-user me-1"></i>*

*<span class="for-whom-text"><?= htmlspecialchars($gift['for\_whom']) ?></span>*

*</p>*

*<?php if (!empty($gift['description'])): ?>*

*<p class="card-text"><?= htmlspecialchars($gift['description']) ?></p>*

*<?php endif; ?>*

*<p class="card-text">*

*<i class="fas fa-money-bill-wave me-1"></i>*

*<span class="budget-text"><?= number\_format($gift['budget'], 0, ',', ' ') ?> BYN</span>*

*</p>*

*</div>*

*<div class="card-footer bg-transparent d-flex flex-wrap">*

*<a href="edit.php?id=<?= $gift['id'] ?>" class="btn btn-outline-primary btn-sm me-1 mb-1 flex-grow-1">*

*<i class="fas fa-edit me-1"></i>Редактировать*

*</a>*

*<?php if ($gift['status'] === 'не куплен'): ?>*

*<a href="update\_status.php?id=<?= $gift['id'] ?>&status=куплен"*

*class="btn btn-outline-success btn-sm me-1 mb-1 flex-grow-1 confirmation-link"*

*data-message="Отметить как купленный?">*

*<i class="fas fa-check me-1"></i>Куплен*

*</a>*

*<?php else: ?>*

*<a href="update\_status.php?id=<?= $gift['id'] ?>&status=не куплен"*

*class="btn btn-outline-warning btn-sm me-1 mb-1 flex-grow-1 confirmation-link"*

*data-message="Отметить как не купленный?">*

*<i class="fas fa-undo me-1"></i>Не куплен*

*</a>*

*<?php endif; ?>*

*<a href="delete.php?id=<?= $gift['id'] ?>"*

*class="btn btn-outline-danger btn-sm mb-1 flex-grow-1 confirmation-link"*

*data-message="Удалить этот подарок?">*

*<i class="fas fa-trash me-1"></i>Удалить*

*</a>*

*</div>*

*</div>*

*</div>*

*<?php endforeach; ?>*

*</div>*

*<?php endif; ?>*

*</div>*

*</div>*

*</div>*

*<div class="modal fade" id="confirmationModal" tabindex="-1" aria-labelledby="confirmationModalLabel" aria-hidden="true">*

*<div class="modal-dialog">*

*<div class="modal-content">*

*<div class="modal-header">*

*<h5 class="modal-title" id="confirmationModalLabel">Подтверждение</h5>*

*<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>*

*</div>*

*<div class="modal-body">*

*<p id="confirmationMessage"></p>*

*</div>*

*<div class="modal-footer">*

*<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Отмена</button>*

*<a href="#" class="btn btn-danger" id="confirmAction">Подтвердить</a>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>*

*<script>*

*document.addEventListener('DOMContentLoaded', () => {*

*// Confirmation modal logic*

*const modal = document.getElementById('confirmationModal');*

*const messageElement = document.getElementById('confirmationMessage');*

*const confirmButton = document.getElementById('confirmAction');*

*document.querySelectorAll('.confirmation-link').forEach(link => {*

*link.addEventListener('click', (e) => {*

*e.preventDefault();*

*const url = e.currentTarget.href;*

*const message = e.currentTarget.dataset.message;*

*messageElement.textContent = message;*

*confirmButton.href = url;*

*const modalInstance = new bootstrap.Modal(modal);*

*modalInstance.show();*

*});*

*});*

*});*

*</script>*

*</body>*

*</html>*

***Файл add.php***

*<?php*

*require\_once '../dbConf/config.php';*

*$message = '';*

*$message\_type = '';*

*if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*

*$title = trim($\_POST['title'] ?? '');*

*$description = trim($\_POST['description'] ?? '');*

*$for\_whom = trim($\_POST['for\_whom'] ?? '');*

*$budget = floatval($\_POST['budget'] ?? 0);*

*$errors = [];*

*if (empty($title)) {*

*$errors[] = 'Название подарка обязательно для заполнения';*

*}*

*if (empty($for\_whom)) {*

*$errors[] = 'Поле "Для кого" обязательно для заполнения';*

*}*

*if ($budget < 0) {*

*$errors[] = 'Бюджет не может быть отрицательным';*

*}*

*if (empty($errors)) {*

*try {*

*$stmt = $pdo->prepare("INSERT INTO gifts (title, description, for\_whom, budget) VALUES (?, ?, ?, ?)");*

*$stmt->execute([$title, $description, $for\_whom, $budget]);*

*$message = 'Подарок успешно добавлен!';*

*$message\_type = 'success';*

*$title = $description = $for\_whom = '';*

*$budget = 0;*

*} catch(PDOException $e) {*

*$message = 'Ошибка при добавлении подарка: ' . $e->getMessage();*

*$message\_type = 'danger';*

*}*

*} else {*

*$message = implode('<br>', $errors);*

*$message\_type = 'danger';*

*}*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="ru">*

*<head>*

*<meta charset="UTF-8">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<title>Добавить подарок</title>*

*<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">*

*<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">*

*<link rel="stylesheet" href="../src/css/style.css">*

*</head>*

*<body class="bg-light">*

*<nav class="navbar navbar-expand-lg navbar-light bg-purple-custom fixed-top shadow-sm">*

*<div class="container-fluid">*

*<a class="navbar-brand" href="add.php">*

*<i class="fas fa-plus-circle me-2"></i>ДОБАВИТЬ ПОДАРОК*

*</a>*

*<div class="d-flex align-items-center me-auto ms-3">*

*<span class="navbar-text text-muted d-none d-md-block">*

*Сегодня: <?= date('d.m.Y') ?>*

*</span>*

*</div>*

*<a href="index.php" class="btn btn-yellow-custom ms-auto">*

*<i class="fas fa-arrow-left me-2"></i>Назад к списку*

*</a>*

*</div>*

*</nav>*

*<div class="container mt-4">*

*<div class="row justify-content-center">*

*<div class="col-12">*

*<div class="form-container">*

*<?php if ($message): ?>*

*<div class="alert alert-<?= $message\_type ?> alert-dismissible fade show" role="alert">*

*<i class="fas fa-<?= $message\_type === 'success' ? 'check-circle' : 'exclamation-triangle' ?> me-2"></i>*

*<?= $message ?>*

*<button type="button" class="btn-close" data-bs-dismiss="alert"></button>*

*</div>*

*<?php endif; ?>*

*<div class="card shadow">*

*<div class="card-body">*

*<form method="POST" action="">*

*<div class="mb-3">*

*<label for="title" class="form-label">*

*Название подарка <span class="required">\*</span>*

*</label>*

*<input type="text"*

*class="form-control"*

*id="title"*

*name="title"*

*value="<?= htmlspecialchars($title ?? '') ?>"*

*placeholder="Введите название подарка"*

*required>*

*</div>*

*<div class="mb-3">*

*<label for="for\_whom" class="form-label">*

*Для кого <span class="required">\*</span>*

*</label>*

*<input type="text"*

*class="form-control"*

*id="for\_whom"*

*name="for\_whom"*

*value="<?= htmlspecialchars($for\_whom ?? '') ?>"*

*placeholder="Например: брат, друг, коллега"*

*required>*

*</div>*

*<div class="mb-3">*

*<label for="budget" class="form-label">*

*Бюджет (BYN)*

*</label>*

*<div class="input-group">*

*<input type="number"*

*class="form-control"*

*id="budget"*

*name="budget"*

*value="<?= $budget ?? 0 ?>"*

*min="0"*

*step="0.01"*

*placeholder="0.00">*

*<span class="input-group-text">BYN</span>*

*</div>*

*</div>*

*<div class="mb-3">*

*<label for="description" class="form-label">*

*Описание*

*</label>*

*<textarea class="form-control"*

*id="description"*

*name="description"*

*rows="4"*

*placeholder="Дополнительная информация о подарке (необязательно)"><?= htmlspecialchars($description ?? '') ?></textarea>*

*</div>*

*<div class="d-grid gap-2 d-md-flex justify-content-md-end">*

*<a href="index.php" class="btn btn-secondary me-md-2">*

*<i class="fas fa-times me-2"></i>Отмена*

*</a>*

*<button type="submit" class="btn btn-success">*

*<i class="fas fa-save me-2"></i>Добавить подарок*

*</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>*

*</body>*

*</html>*

***Файл delete.php***

*<?php*

*require\_once '../dbConf/config.php';*

*$id = $\_GET['id'] ?? null;*

*if (!$id || !is\_numeric($id)) {*

*header('Location: index.php');*

*exit;*

*}*

*try {*

*$stmt = $pdo->prepare("SELECT id FROM gifts WHERE id = ?");*

*$stmt->execute([$id]);*

*if (!$stmt->fetch()) {*

*header('Location: index.php');*

*exit;*

*}*

*$stmt = $pdo->prepare("DELETE FROM gifts WHERE id = ?");*

*$stmt->execute([$id]);*

*header('Location: index.php?deleted=1');*

*exit;*

*} catch(PDOException $e) {*

*header('Location: index.php?error=' . urlencode('Ошибка при удалении подарка: ' . $e->getMessage()));*

*exit;*

*}*

*?>*

***Файл edit.php***

*<?php*

*require\_once '../dbConf/config.php';*

*$message = '';*

*$message\_type = '';*

*$gift = null;*

*$id = $\_GET['id'] ?? null;*

*if (!$id || !is\_numeric($id)) {*

*header('Location: index.php');*

*exit;*

*}*

*try {*

*$stmt = $pdo->prepare("SELECT \* FROM gifts WHERE id = ?");*

*$stmt->execute([$id]);*

*$gift = $stmt->fetch();*

*if (!$gift) {*

*header('Location: index.php');*

*exit;*

*}*

*} catch(PDOException $e) {*

*$message = 'Ошибка при загрузке подарка: ' . $e->getMessage();*

*$message\_type = 'danger';*

*}*

*if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {*

*$title = trim($\_POST['title'] ?? '');*

*$description = trim($\_POST['description'] ?? '');*

*$for\_whom = trim($\_POST['for\_whom'] ?? '');*

*$budget = floatval($\_POST['budget'] ?? 0);*

*$errors = [];*

*if (empty($title)) {*

*$errors[] = 'Название подарка обязательно для заполнения';*

*}*

*if (empty($for\_whom)) {*

*$errors[] = 'Поле "Для кого" обязательно для заполнения';*

*}*

*if ($budget < 0) {*

*$errors[] = 'Бюджет не может быть отрицательным';*

*}*

*if (empty($errors)) {*

*try {*

*$stmt = $pdo->prepare("UPDATE gifts SET title = ?, description = ?, for\_whom = ?, budget = ? WHERE id = ?");*

*$stmt->execute([$title, $description, $for\_whom, $budget, $id]);*

*$message = 'Подарок успешно обновлен!';*

*$message\_type = 'success';*

*$gift['title'] = $title;*

*$gift['description'] = $description;*

*$gift['for\_whom'] = $for\_whom;*

*$gift['budget'] = $budget;*

*} catch(PDOException $e) {*

*$message = 'Ошибка при обновлении подарка: ' . $e->getMessage();*

*$message\_type = 'danger';*

*}*

*} else {*

*$message = implode('<br>', $errors);*

*$message\_type = 'danger';*

*}*

*}*

*?>*

*<!DOCTYPE html>*

*<html lang="ru">*

*<head>*

*<meta charset="UTF-8">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<title>Редактировать подарок</title>*

*<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">*

*<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">*

*<link rel="stylesheet" href="../src/css/style.css">*

*</head>*

*<body class="bg-light">*

*<nav class="navbar navbar-expand-lg navbar-light bg-purple-custom fixed-top shadow-sm">*

*<div class="container-fluid">*

*<a class="navbar-brand" href="edit.php?id=<?= $gift['id'] ?>">*

*<i class="fas fa-edit me-2"></i>РЕДАКТИРОВАТЬ ПОДАРОК*

*</a>*

*<div class="d-flex align-items-center me-auto ms-3">*

*<span class="navbar-text text-muted d-none d-md-block">*

*Сегодня: <?= date('d.m.Y') ?>*

*</span>*

*</div>*

*<a href="index.php" class="btn btn-yellow-custom ms-auto">*

*<i class="fas fa-arrow-left me-2"></i>Назад к списку*

*</a>*

*</div>*

*</nav>*

*<div class="container mt-4">*

*<div class="row justify-content-center">*

*<div class="col-12">*

*<div class="form-container">*

*<?php if ($message): ?>*

*<div class="alert alert-<?= $message\_type ?> alert-dismissible fade show" role="alert">*

*<i class="fas fa-<?= $message\_type === 'success' ? 'check-circle' : 'exclamation-triangle' ?> me-2"></i>*

*<?= $message ?>*

*<button type="button" class="btn-close" data-bs-dismiss="alert"></button>*

*</div>*

*<?php endif; ?>*

*<?php if ($gift): ?>*

*<div class="card shadow">*

*<div class="card-body">*

*<form method="POST" action="">*

*<div class="mb-3">*

*<label for="title" class="form-label">*

*Название подарка <span class="required">\*</span>*

*</label>*

*<input type="text"*

*class="form-control"*

*id="title"*

*name="title"*

*value="<?= htmlspecialchars($gift['title']) ?>"*

*placeholder="Введите название подарка"*

*required>*

*</div>*

*<div class="mb-3">*

*<label for="for\_whom" class="form-label">*

*Для кого <span class="required">\*</span>*

*</label>*

*<input type="text"*

*class="form-control"*

*id="for\_whom"*

*name="for\_whom"*

*value="<?= htmlspecialchars($gift['for\_whom']) ?>"*

*placeholder="Например: Мама, Друг, Коллега"*

*required>*

*</div>*

*<div class="mb-3">*

*<label for="budget" class="form-label">*

*Бюджет (BYN)*

*</label>*

*<div class="input-group">*

*<input type="number"*

*class="form-control"*

*id="budget"*

*name="budget"*

*value="<?= $gift['budget'] ?>"*

*min="0"*

*step="0.01"*

*placeholder="0.00">*

*<span class="input-group-text">BYN</span>*

*</div>*

*</div>*

*<div class="mb-3">*

*<label for="description" class="form-label">*

*Описание*

*</label>*

*<textarea class="form-control"*

*id="description"*

*name="description"*

*rows="4"*

*placeholder="Дополнительная информация о подарке (необязательно)"><?= htmlspecialchars($gift['description']) ?></textarea>*

*</div>*

*<div class="mb-3">*

*<label class="form-label">Текущий статус</label>*

*<div class="form-control-plaintext">*

*<span class="badge <?= $gift['status'] === 'куплен' ? 'bg-success' : 'bg-warning' ?>">*

*<?= htmlspecialchars($gift['status']) ?>*

*</span>*

*</div>*

*</div>*

*<div class="d-grid gap-2 d-md-flex justify-content-md-end">*

*<a href="index.php" class="btn btn-secondary me-md-2">*

*<i class="fas fa-times me-2"></i>Отмена*

*</a>*

*<button type="submit" class="btn btn-primary">*

*<i class="fas fa-save me-2"></i>Сохранить изменения*

*</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*<?php endif; ?>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>*

*</body>*

*</html>*

***Файл update\_status.php***

*<?php*

*require\_once '../dbConf/config.php';*

*$id = $\_GET['id'] ?? null;*

*$status = $\_GET['status'] ?? null;*

*if (!$id || !is\_numeric($id) || !in\_array($status, ['не куплен', 'куплен'])) {*

*header('Location: index.php');*

*exit;*

*}*

*try {*

*$stmt = $pdo->prepare("SELECT id FROM gifts WHERE id = ?");*

*$stmt->execute([$id]);*

*if (!$stmt->fetch()) {*

*header('Location: index.php');*

*exit;*

*}*

*$stmt = $pdo->prepare("UPDATE gifts SET status = ? WHERE id = ?");*

*$stmt->execute([$status, $id]);*

*$message = $status === 'куплен' ? 'Подарок отмечен как купленный!' : 'Подарок отмечен как не купленный!';*

*header('Location: index.php?status\_updated=1&message=' . urlencode($message));*

*exit;*

*} catch(PDOException $e) {*

*header('Location: index.php?error=' . urlencode('Ошибка при обновлении статуса: ' . $e->getMessage()));*

*exit;*

*}*

*?>*

***Файл style.css***

*body {*

*background-image: url('../img/background.png');*

*background-size: cover;*

*background-repeat: no-repeat;*

*background-attachment: fixed;*

*background-position: center center;*

*padding-top: 70px;*

*}*

*.gift-card {*

*transition: transform 0.2s;*

*background-color: rgba(255, 255, 255, 0.9);*

*}*

*.gift-card:hover {*

*transform: translateY(-2px);*

*}*

*.status-badge {*

*font-size: 0.8em;*

*}*

*.budget-text {*

*font-weight: bold;*

*color: #28a745;*

*}*

*.for-whom-text {*

*color: #6c757d;*

*font-style: italic;*

*}*

*.modal-footer .btn-danger {*

*background-color: #dc3545;*

*border-color: #dc3545;*

*}*

*.bg-purple-custom {*

*background-color: #BF8FCC !important;*

*}*

*.navbar-brand .fa-gift{*

*color: #FFD700;*

*}*

*.navbar-brand .fa-plus-circle {*

*color: #FFD700 !important;*

*}*

*.btn-yellow-custom {*

*background-color: #FFD700;*

*border-color: #FFD700;*

*color: #212529;*

*}*

*.btn-yellow-custom:hover {*

*background-color: #FFC107;*

*border-color: #FFC107;*

*color: #212529;*

*}*

*.form-container {*

*max-width: 600px;*

*margin: 0 auto;*

*margin-top: 100px;*

*}*

*.required {*

*color: #dc3545;*

*}*