Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ УПРАВЛЕНИЯ БИЗНЕС-ПРОЦЕССАМИ

КАФЕДРА БИЗНЕС-ИНФОРМАТИКИ И МОДЕЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ

**КУРСОВАЯ РАБОТА**

Создание web-приложения с применением современных web-технологий на собственную тему "форум для обсуждения JDM-машин"

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Никитин

подпись, дата

Студент УБ23-08Б, 432320869 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А. Короткая

подпись, дата

Красноярск 2024

**Содержание**

[Введение 3](#_Toc184307266)

[1. Теоретическая часть 4](#_Toc184307267)

[1.1. HTML – язык разметки гипертекста 4](#_Toc184307268)

[1.2. CSS – каскадные таблицы стилей 6](#_Toc184307269)

[1.3. JavaScript – язык интерактивности 7](#_Toc184307270)

[1.4. PHP – серверный язык обработки данных 8](#_Toc184307271)

[1.5. MySQL – система управление СУБД 10](#_Toc184307272)

[1.6. phpMyAdmin и OpenServer – инструменты для СУБД 11](#_Toc184307273)

[2. Практическая часть 12](#_Toc184307274)

[2.1. Базовые требовании для конфигурации и базы данных 12](#_Toc184307275)

[2.2. Подключение к базе данных 15](#_Toc184307276)

[2.3. Шапка и футер для главной страницы 16](#_Toc184307277)

[2.4. Главная страница 19](#_Toc184307278)

[2.5. Авторизация в личный аккаунт 25](#_Toc184307279)

[2.6. Регистрация нового аккаунта 29](#_Toc184307280)

[2.7. Личный кабинет 32](#_Toc184307281)

[2.8. Управление профилями 35](#_Toc184307282)

[2.9. Просмотр проектов 39](#_Toc184307283)

[2.10. Вкладка контактов – «Обо мне» 45](#_Toc184307284)

[2.11. Вкладка выхода из аккаунта 47](#_Toc184307285)

[Заключение 48](#_Toc184307286)

[Список использованных источников 50](#_Toc184307287)

# **Введение**

Цель данного проекта – разработка веб-приложения, представляющего собой специализированный форум для обсуждения JDM (Japanese Domestic Market) автомобилей. В последние десятилетия JDM-автомобили завоевали огромную популярность среди автолюбителей по всему миру благодаря своей высокой производительности, уникальному дизайну и инновационным инженерным решениям. Главная идея нашего проекта заключается в создании удобной и интуитивно понятной платформы, которая объединит энтузиастов JDM-автомобилей, предоставляя им возможность делиться опытом, обсуждать новинки и обмениваться советами по эксплуатации и модификации. В отличие от общих автомобильных форумов, данное приложение будет сосредоточено на культуре JDM, создавая уникальное сообщество для автолюбителей с общими интересами и страстью к японским автомобилям.

Актуальность данного проекта обусловлена растущим интересом к JDM-автомобилям и необходимостью создания специализированного пространства для их обсуждения. В современном мире, где информация распространяется с невероятной скоростью, автолюбители ищут платформы, которые позволят им не только получать актуальную информацию о новинках, но и находить единомышленников, обмениваться опытом и получать советы по модификации своих автомобилей. Форумы становятся не просто местом для обсуждения, но и настоящими сообществами, где участники могут обмениваться знаниями, идеями и поддержкой. Наш форум призван удовлетворить эту потребность, предлагая функциональный и безопасный инструмент для общения и обмена информацией, что особенно важно в условиях современного интернета.

В процессе разработки нашего веб-приложения будут использованы современные веб-технологии, такие как HTML5, CSS, JavaScript, PHP и MySQL. Эти технологии обеспечат высокую производительность, безопасность и масштабируемость приложения, что является критически важным для обеспечения надежной работы форума и защиты данных пользователей. Использование HTML и CSS позволит создать интуитивно понятный и привлекательный интерфейс, который будет способствовать удобству пользователей и улучшению их взаимодействия с форумом. JavaScript добавит динамичности и отзывчивости приложению, а PHP и MySQL обеспечат эффективное управление данными и надежную работу серверной части.

Ожидаемый результат – полноценное веб-приложение, позволяющее пользователям эффективно взаимодействовать, обсуждать актуальные темы и находить поддержку в своем увлечении JDM-автомобилями. Мы стремимся создать форум, который будет отличаться интуитивным интерфейсом, удобной навигацией и возможностью персонализации пользовательского опыта, чтобы каждый участник чувствовал себя комфортно и мог легко находить нужную информацию.

В данном отчете подробно описаны этапы разработки, использованные технологии и инструменты, а также представлены результаты и перспективы развития проекта. Мы уверены, что создание такого форума станет важным шагом в развитии сообщества автолюбителей, интересующихся JDM-автомобилями, и поможет укрепить связи между участниками, способствуя обмену идей и формированию новых связей.

1. **Теоретическая часть**
   1. **Что такое HTML**

Язык гипертекстовой разметки сайта, или HTML (HyperText Markup Language), — код, помогающий структурировать содержание каждой веб-страницы.

HTML создает иерархическую структуру веб-страницы, используя заголовки, абзацы, списки и таблицы. Такая структура помогает пользователю легче ориентироваться на сайте.

Также с помощью HTML отображается текст, изображения, таблицы, видео и аудио. А еще благодаря коду на HTML мы можем переходить по ссылкам в интернете с одного сайта на другой.

HTML-документ создаётся в обычном текстовом редакторе — чаще всего в «Блокноте». Есть и специализированные приложения вроде Notepad++ или Visual Studio Code — с подсветкой кода.

С помощью языка разметки HTML браузер делает запрос по адресу, который ввёл пользователь, и получает файл в формате «.html». Браузер распознаёт код, выбирает знакомые для себя сигналы: понимает, что написать словами, где поставить заголовок и какой именно. Таким образом код из файла преобразуется в необходимые визуальные объекты.

С помощью HTML можно:

* Делать текстовую разметку — форматировать текст, выделять фрагменты, создавать списки, добавлять сноски.
* Встраивать медиа. HTML позволяет размещать на сайте изображения, аудио, видео, карты.
* Создавать ссылки и навигацию. Гиперссылки и списки меню помогают быстрее найти информацию и сориентироваться на странице.
* Создавать таблицы. Нередко информацию удобно представить в табличном виде. HTML умеет работать с таблицами.
* Создавать формы. Формы нужны для регистрации посетителей сайта по телефону и электронной почте, оформления заказов, опросов и сбора обратной связи — отзывов, комментариев, предложений.

В HTML можно даже создавать простой дизайн: например, устанавливать цвет и шрифт текста или фоновый цвет блока. Но более сложный дизайн страницы разработчики делают с помощью CSS — языка стилей, который создали специально для работы в связке с HTML.

Возможностей HTML не хватает, чтобы «оживлять» сайты, делать их функциональными и интерактивными. Здесь приходит на помощь язык программирования JavaScript, благодаря которому мы можем взаимодействовать с формами на сайте, видеть динамические элементы и анимации.

Основные элементы HTML:

**<**html**>:** Корневой элемент, содержащий весь код страницы.

**<**head**>:** содержит метаданные страницы, такие как заголовок (<title>), описание (<meta name="description">), ссылки на внешние стили (<link rel="stylesheet">) и скрипты (<script>). Эти данные не отображаются непосредственно на странице, но важны для поисковых систем и функциональности сайта.

**<**body**>:** содержит видимый контент страницы: текст, изображения, ссылки, видео и другие элементы.

Основные теги: HTML использует множество тегов для форматирования текста и организации контента. К наиболее распространенным относятся: <h1>, <p>, <br>, <ul>, <a href= «URL»> текст ссылки</a>, <table>, <tr>, <div>.

Теги часто имеют атрибуты, которые модифицируют их поведение. Например, атрибут href в теге <a> указывает URL-адрес ссылки, а атрибут src в теге <img> указывает путь к изображению. Атрибут alt в теге <img> предоставляет альтернативный текст для изображений, важный для доступности сайта и SEO.

Современный HTML уделяет большое внимание семантической разметке – использованию тегов, которые описывают не только внешний вид, но и смысл контента. Например, <article> обозначает отдельную статью, <aside> – боковую панель, <nav> – навигационное меню, <header> и <footer> – соответственно, заголовок и подвал страницы. Семантическая разметка улучшает доступность сайта для людей с ограниченными возможностями и повышает его рейтинг в поисковых системах.

* 1. **Что такое CSS**

CSS — это язык описания внешнего вида документа, то есть он отвечает за то, как выглядят веб-страницы: цвет фона и декоративных элементов, размер и стиль шрифтов. Термин расшифровывается как Cascading Style Sheets (каскадные таблицы стилей). CSS взаимодействует с другим языком разметки — HTML, который отвечает за размещение элементов на странице.

Основные и самые часто встречающиеся задачи, которые выполняются с использованием CSS:

* оформление текста (задание цвета, размера шрифта, выравнивания, высоты строки и т. д.);
* оформление html-элементов (размеры, отступы, цвет фона, тени, скругления);
* построение сеток для расположения контента (об этом подробнее ниже);
* анимации (с CSS мы можем создавать красивые анимации без использования JavaScript);
* создание адаптивного дизайна (мы можем менять отображение в зависимости от размеров устройства, на котором просматривают сайт).

CSS (каскадные таблицы стилей) – это язык, определяющий визуальное оформление веб-страниц: шрифты, цвета, расположение элементов и многое другое. Его ключевое преимущество – разделение структуры (HTML) и стиля, что делает код чище, понятнее и проще в поддержке. CSS обеспечивает гибкость и масштабируемость, позволяя легко адаптировать дизайн под различные устройства и быстро изменять внешний вид сайта. Возможности CSS простираются и на добавление анимаций и трансформаций, значительно повышая привлекательность веб-страниц [4].

Однако, при работе с CSS необходимо учитывать возможные проблемы совместимости между различными браузерами, что может потребовать дополнительной работы по обеспечению кросс браузерной совместимости. Создание сложных анимационных эффектов требует глубокого понимания CSS и может быть достаточно трудоемким. Наконец, избыточное использование CSS-стилей и больших изображений может негативно сказаться на скорости загрузки страницы, что ухудшит пользовательский опыт. Поэтому, эффективное использование CSS предполагает баланс между желаемой визуальной привлекательностью и производительностью сайта.

* 1. **Что такое JavaScript**

На JavaScript написан практически весь фронтенд — всё, что мы можем увидеть в интернете. А относительно недавно он обосновался в бэкенде, на смартфонах и даже в прошивках умных чайников. Язык востребован и регулярно занимает первое место в рейтинге [Stack Overflow](https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies).

Где JS используют чаще всего:  
Веб-сайты и веб-приложения. Самая популярная область применения языка JavaScript — это написание кода для сайта. Практически на каждом современном сайте используют код, написанный на JS.  
Расширения для браузера. Небольшие простые скрипты, которые добавляют дополнительный функционал — блокируют рекламу, позволяют сохранять аудио, отправляют уведомления о новых письмах или меняют цветовую схему сайта.

Важно понимать, что современные фреймворки и библиотеки (React, Angular, Vue.js) значительно упрощают разработку и минимизируют риски. Они предоставляют готовые компоненты, структуры, и инструменты для эффективного управления состоянием и взаимодействия с пользователем, автоматизируя многие рутинные задачи. Это позволяет разработчикам сосредоточиться на логике приложения, а не на технических аспектах JavaScript. Также, использование модулей и организации кода в файлы с использованием ES6 (и выше) методов минимизирует глобальную загрязненость и улучшает понимание и поддержку кода. Таким образом, JavaScript, в сочетании с современными инструментами, становится мощным средством для создания сложных и эффективных веб-приложений.

* 1. **Что такое PHP**

PHP (Hypertext Preprocessor) – это широко распространенный серверный язык программирования, используемый для создания динамических веб-страниц и веб-приложений. Его ключевое преимущество – способность генерировать контент на сервере, взаимодействуя с базами данных и обрабатывая пользовательские запросы, что позволяет создавать сложные и функциональные веб-ресурсы, от простых сайтов до крупных интернет-магазинов. PHP просто интегрируется с популярными СУБД (MySQL, PostgreSQL и др.), позволяя эффективно управлять данными и создавать динамическое содержимое. Его гибкость и обширная экосистема библиотек и фреймворков открывают возможности для решения разнообразных задач веб-разработки [7].

Однако, использование PHP требует внимательного отношения к безопасности. Неправильное обращение с данными, например, незащищенные от SQL-инъекций запросы к базе данных, может привести к серьёзным уязвимостям [6]. Кроме того, не оптимизированный код может снизить производительность приложения, особенно при обработке больших объемов данных или при большом количестве запросов. Также, по мере роста сложности проекта, структура кода может становиться сложной в поддержке и масштабировании, требуя применения принципов объектно-ориентированного программирования и использования более структурированных подходов к архитектуре.

Рассмотрим задачи, которые выполняет PHP:

* Обработка форм. Практически у каждого сайта есть определенная форма, с помощью которой пользователи регистрируются, входят в личный кабинет, оформляют или оплачивают заказ. PHP помогает обработать данные, которые вносятся в эту форму. Например, если вы хотите [сделать форму обратной связи на сайте](https://pro-promotion.ru/articles/konversiya/chto-takoe-forma-obratnoy-svyazi-i-kak-eye-sdelat-na-sayte), то можно использовать не только языки программирования, но этот способ будет более надежным.
* Базы данных. PHP используется для сохранения информации о пользователе в единой базе данных. Например, при авторизации или регистрации на сайте нужно вводить логин и пароль, с помощью PHP можно понять есть ли такая комбинация в базе данных.
* Динамичные страницы. На PHP достаточно создать один файл, который будет отвечать за все страницы сайта. Например, если вы захотите внести какие-то изменения в дизайн и оформление сайта. Это экономит кучу времени и не нужно каждый раз прописывать код заново.
* Сессии и куки. Еще одна задача, для которой используется PHP. Многие сайты помещают в куки пользователя данные о посещениях и просмотренных товарах, услугах. Преимущество этого в том, что база данных не забивается и часть информации хранится на компьютере пользователя.

Это лишь основные задачи, для которых используется данный язык программирования, он имеет много преимуществ и достаточно прост для изучения и применения.

В заключение, можно сказать, что PHP остается актуальным и популярным языком для веб-разработки, однако его эффективное использование требует знания лучших практик программирования и применения современных инструментов и фреймворков для обеспечения безопасности, масштабируемости и производительности разрабатываемых приложений.

* 1. **Что такое MySQL**

MySQL — это реляционная система управления [базами данных](https://blog.skillfactory.ru/glossary/baza-dannyh/) (СУБД), которая распространяется как свободное программное обеспечение. Является одной из наиболее популярных, так как отличается гибкостью, легкостью, удобством в использовании.

Слово «реляционный» означает, что базы представлены в виде связанной информации и описываются как набор связей. MySQL работает с языком запросов [SQL](https://blog.skillfactory.ru/glossary/sql/), который традиционно используется в базах данных.

Кто использует MySQL и в каких сферах:

* Бэкенд-разработчики, задача которых — управлять базой данной на серверной стороне сайта или приложения.
* Разработчики десктопных и мобильных приложений, в которых есть база данных для хранения информации.
* Администраторы баз данных — специалисты, которые занимаются построением и настройкой базы.

В основном MySQL используется для работы со средними и небольшими проектами. Она имеет открытый исходный код, поэтому любой разработчик может с ним ознакомиться. Сейчас систему поддерживают известная компания Oracle и [независимое сообщество](https://dev.mysql.com/community/) разработчиков-энтузиастов.

MySQL — это популярная система управления базами данных (СУБД), широко используемая для хранения и обработки структурированных данных в веб-приложениях. Она обеспечивает высокую производительность при работе с большими объемами информации, поддерживает различные типы данных, и обладает гибкой масштабируемостью [3]. Это делает MySQL хорошим выбором для многих проектов, начиная от небольших сайтов и заканчивая крупными интернет-магазинами и корпоративными системами.

Однако, MySQL имеет определенные ограничения. Некоторые критики отмечают, что возможности работы с хранимыми процедурами и триггерами в MySQL могут быть ограниченными по сравнению с другими СУБД. Это может влиять на сложность реализации сложных бизнес-логик и оптимизации запросов. Также, в отдельных случаях, при работе с крайне большими объемами данных возможна недостаточная эффективность, требующая тщательной настройки и оптимизации базы данных, а также возможных переходов к более масштабируемым решениям.

1. **Практическая часть**
   1. **Программы, которые были использованы в процессе работы**

Visual Studio Code (или VS Code) от Microsoft — это удобный текстовый редактор для разработчиков, который работает на Windows, macOS и Linux. Инструмент поддерживает множество языков программирования от JavaScript до Python и предлагает широкий функционал для написания, отладки и тестирования кода. С помощью VS Code можно настраивать среду разработки под индивидуальные задачи, используя расширения и встроенные инструменты.

VS Code позволяет решать разные задачи: от простого редактирования файлов до разработки сложных веб- и мобильных приложений. Он поддерживает работу с такими языками программирования как:

* JavaScript, TypeScript, HTML, CSS — для фронтенд- и веб-разработки.
* Python, R — для анализа данных и машинного обучения.
* Java, Kotlin, Go, Rust — для серверной и высокопроизводительной разработки.
* C, C++ — для системного программирования.
* SQL — для работы с базами данных.

Кроме того, можно добавить поддержку дополнительных языков и фреймворков через расширения.

Редактор используют разработчики для создания веб-приложений, API и других программных продуктов; системные администраторы для написания и выполнения скриптов автоматизации; и аналитики данных для анализа больших объемов данных и машинного обучения.

В процессе разработки нашего веб-приложения для специализированного форума о JDM-автомобилях мы выбрали Visual Studio Code (VS Code) в качестве основного инструмента для написания и обработки кода. Этот мощный редактор кода, разработанный компанией Microsoft, стал незаменимым помощником благодаря своим широким возможностям и гибкости, что позволило нам эффективно работать с различными языками программирования, такими как PHP, HTML, CSS и MySQL.

Одним из ключевых преимуществ Visual Studio Code является его интуитивно понятный интерфейс, который позволяет разработчикам сосредоточиться на написании кода, не отвлекаясь на сложные настройки. Поддержка множества расширений предоставляет возможность кастомизации рабочего пространства под индивидуальные потребности каждого разработчика. Мы использовали расширения для PHP, HTML и CSS, что значительно упростило процесс разработки, обеспечив автоматическое дополнение кода, подсветку синтаксиса и встроенные инструменты для отладки.

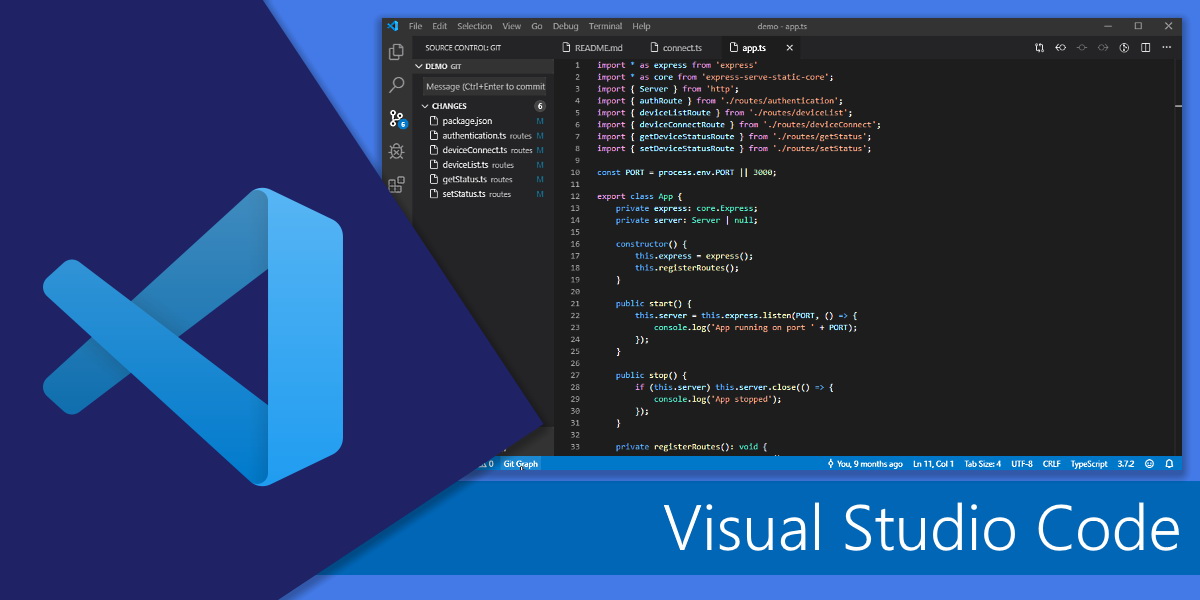


Рисунок 2.1.1 – Visual Studio Code

Open Server Panel — это портативная программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

На сегодняшний день OSPanel широко используется веб-разработчиками по всему миру с целью разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях. Наибольшую популярность проект завоевал у начинающих веб-мастеров, поскольку позволяет быстро создать рабочее окружение и сразу начать изучение веб-технологий без сложных манипуляций по установке и настройке большого количества незнакомых программных компонентов.

Основные характеристики Open Server Panel:

1. Удобство установки и настройки: Open Server Panel позволяет пользователям быстро установить и настроить локальный сервер без необходимости глубоких знаний в администрировании. Все компоненты уже собраны в одном пакете, что значительно упрощает процесс подготовки рабочего окружения.
2. Поддержка различных технологий: Open Server Panel поддерживает множество технологий, таких как PHP, MySQL, Apache, Nginx и другие. Это позволяет разработчикам создавать и тестировать приложения на разных стэках технологий, не устанавливая каждую из них отдельно.
3. Интуитивно понятный интерфейс: Панель управления имеет простой и удобный интерфейс, который позволяет пользователям легко управлять сервером, настраивать параметры и контролировать работу приложений. Все настройки доступны через графический интерфейс, что делает работу с Open Server Panel доступной даже для начинающих разработчиков.
4. Гибкость и расширяемость: Open Server Panel позволяет добавлять и настраивать дополнительные модули и расширения, что делает его подходящим для различных проектов. Пользователи могут легко изменять конфигурацию сервера в соответствии с требованиями своих приложений.
5. Локальная разработка и тестирование: Использование Open Server Panel позволяет разработчикам тестировать свои веб-приложения в локальной среде, что значительно ускоряет процесс разработки и позволяет избежать проблем, связанных с развертыванием на удалённых серверах.
6. Сообщество и поддержка: Open Server Panel имеет активное сообщество пользователей, что обеспечивает доступ к множеству руководств, примеров и поддержки. Это позволяет разработчикам находить решения для возникающих проблем и делиться опытом.

Open Server Panel включает в себя MySQL, что позволяет студентам создавать и управлять базами данных непосредственно из панели управления. С помощью phpMyAdmin, который поставляется с Open Server, можно легко создавать таблицы, выполнять SQL-запросы и управлять данными. Это особенно полезно для курсовых работ, где требуется работа с динамическими данными.

Одним из ключевых преимуществ Open Server Panel является возможность тестирования веб-приложений в локальной среде. Студенты могут запускать свои проекты в браузере, используя локальный адрес (например, http://localhost/имя\_проекта), и проверять функциональность приложения. Это позволяет быстро выявлять и исправлять ошибки, а также тестировать различные сценарии использования.

Использование Open Server Panel в курсовой работе предоставляет студентам мощный и удобный инструмент для разработки веб-приложений. Он позволяет создать локальное окружение, управлять базами данных, тестировать и отлаживать код, а также интегрироваться с современными инструментами разработки. Все эти возможности делают Open Server Panel незаменимым помощником для студентов, работающих над своими проектами, и способствуют успешному выполнению курсовых работ.



Рисунок 2.1.2 – Open Server Panel

Веб-приложение phpMyAdmin — это программа, которая позволяет управлять базами данных через удобный графический интерфейс. Ее устанавливают на сервер, где хранится сайт, а затем открывают в браузере для удаленного администрирования СУБД. Проще говоря, можно отдавать команды путем нажатия на кнопки в пользовательском веб-интерфейсе, а не вручную расписывать их в терминале.

phpMyAdmin предназначена для работы с проектами, которые используют веб-сервер Apache, язык PHP и системы MySQL или MariaDB. Установка и настройка незначительно отличается в зависимости от используемой операционной системы. Приложение постоянно улучшают и развивают, выходят новые версии, хотя внешне панель управления практически не меняется. phpMyAdmin имеет открытый исходный код и распространяется бесплатно, что в совокупности с многофункциональностью и удобством интерфейса является основой ее популярности.

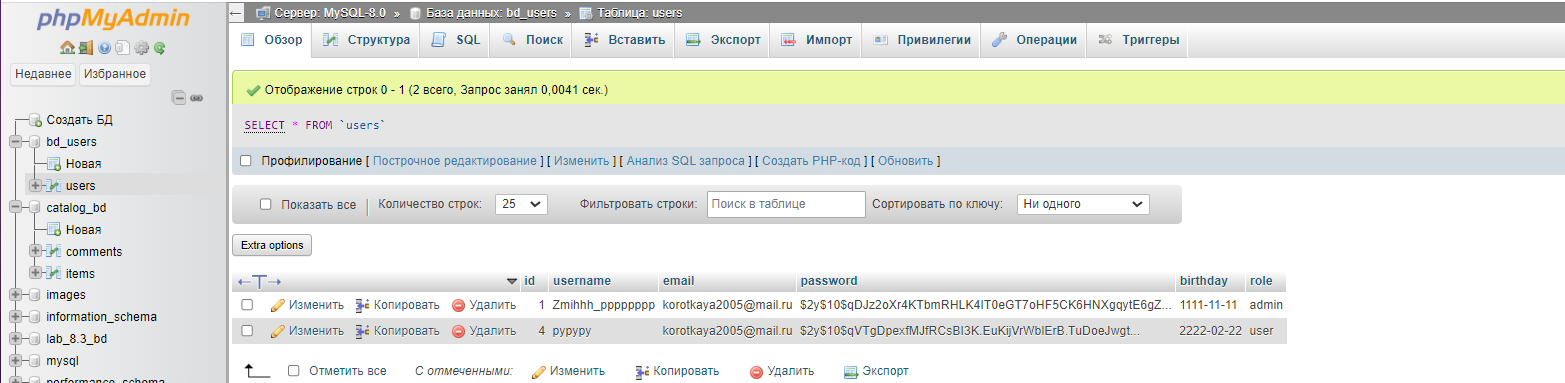


Рисунок 2.1.3 – phpMyAdmin

* 1. **Реализация главной страницы**

На главной странице входа, представленной файлом **index.php**, пользователи получают первичное взаимодействие с веб-ресурсом, посвящённым культуре JDM (Japanese Domestic Market) и автомобильному миру в целом. Дизайн страницы выполнен в минималистичном и интуитивно понятном стиле, чтобы каждый посетитель мог легко ознакомиться с тематикой сайта и принять решение о дальнейшем взаимодействии.

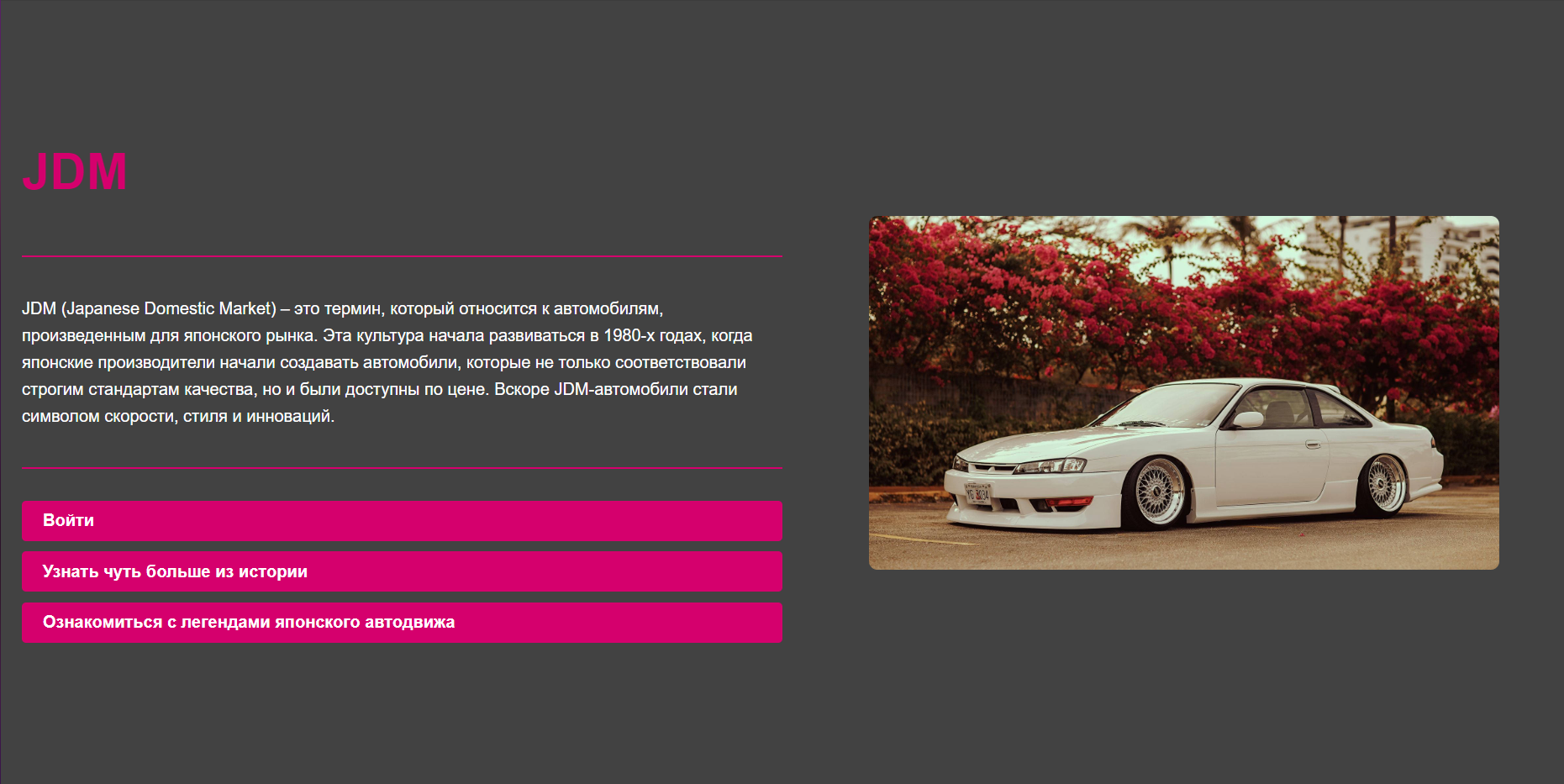


Рисунок 2.2.2 – Внешний вид главной страницы

В правой части страницы расположено яркое и привлекательное описание, которое кратко вводит в атмосферу JDM культуры, акцентируя внимание на её уникальных аспектах, таких как стиль, технические особенности японских автомобилей и влияние на мировую автомобильную индустрию. Эта информация помогает пользователям сразу понять, что сайт посвящён автолюбителям и поклонникам японских марок.

Главное меню страницы предлагает несколько ключевых направлений:

1. **Авторизация** — для зарегистрированных пользователей доступна возможность войти в личный кабинет, где они смогут управлять своим контентом, обсуждать модели автомобилей, а также участвовать в различных форумах и сообществах.
2. **История JDM культуры** — отдельная страница, которая погружает пользователя в увлекательную историю японского автомобильного мира, начиная с первых моделей и заканчивая современными достижениями в автомобильной промышленности Японии. Это отличное место для тех, кто хочет узнать больше о культуре, которая вдохновляет тысячи фанатов по всему миру.
3. **Форма для обсуждения** — уникальная возможность для пользователей поделиться своими мыслями, комментариями и обсуждениями различных моделей автомобилей. Даже для неавторизованных посетителей доступна опция комментировать, что делает сайт более открытым и доступным для широкого круга пользователей. На этой странице представлена форма для оставления комментариев, позволяющая каждому внести свой вклад в обсуждение и обмен мнениями.

Таким образом, главная страница входа на сайт является отправной точкой для дальнейшего погружения в увлекательный мир JDM культуры, предоставляя пользователям удобный интерфейс для изучения информации, общения и обмена опытом.

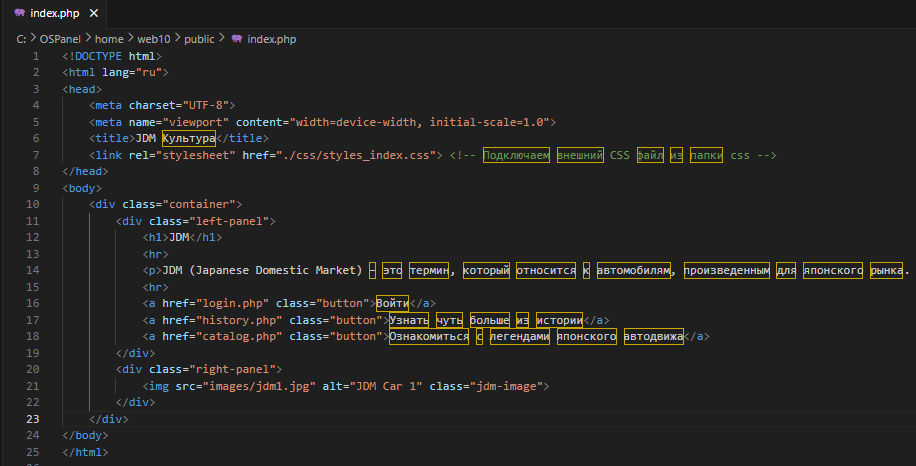


Рисунок 2.2.2 – Код главной страницы index.php

* 1. **Авторизация пользователя и регистрация**

Форма авторизации — это важный элемент взаимодействия пользователей с веб-ресурсом, обеспечивающий безопасность и персонализацию опыта на сайте. В данном случае форма авторизации представлена в виде простого и интуитивно понятного интерфейса, где пользователю необходимо ввести логин и пароль для доступа к закрытым разделам сайта.

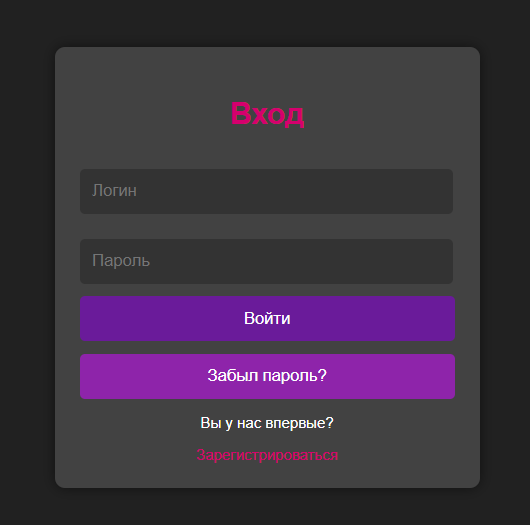


Рисунок 2.3.1 – Форма авторизации

На странице авторизации пользователь видит два поля для ввода данных: одно — для логина, второе — для пароля. Эти поля выполнены в строгом, но удобном дизайне, что позволяет легко ориентироваться даже пользователям с минимальным опытом работы в интернете. В случае, если введённые данные не соответствуют зарегистрированным на сайте, система отображает ошибку, предлагая повторно ввести информацию или восстановить пароль.

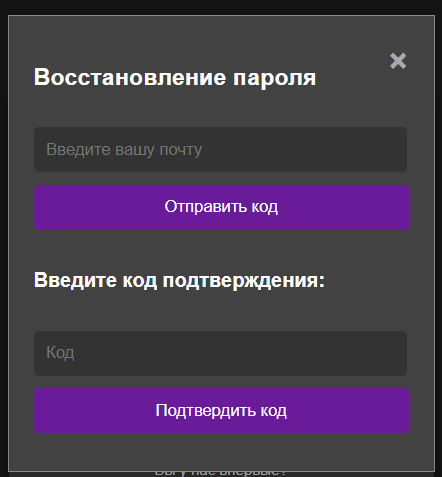


Рисунок 2.3.2 – Форма восстановления пароля

Для удобства пользователей рядом с полем для ввода пароля размещена ссылка «Забыли пароль?», которая ведёт на страницу восстановления учётной записи, где можно легко вернуть доступ к аккаунту, следуя пошаговой инструкции.

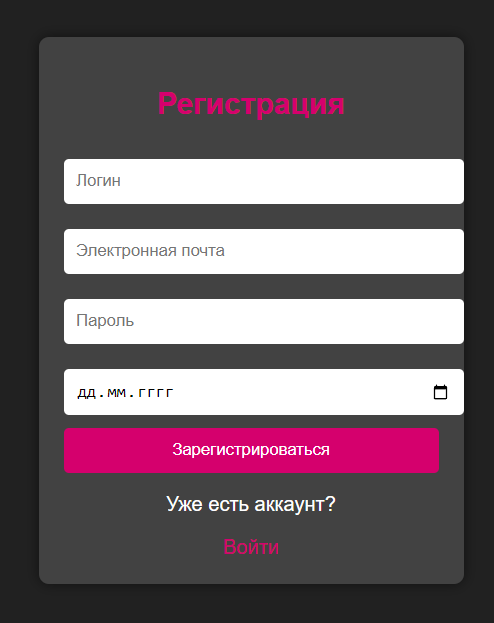


Рисунок 2.3.3 – Форма регистрации

Кроме того, внизу формы авторизации находится ссылка на страницу **регистрации**, которая предназначена для новых пользователей, желающих создать учётную запись на сайте. Перейдя на эту страницу, пользователь попадает на форму регистрации, где ему предстоит ввести несколько обязательных данных:

1. **Логин** — уникальное имя, которое будет использоваться для входа на сайт. Логин должен быть простым, но в то же время отличаться от уже существующих в системе.
2. **Электронная почта** — необходима для связи с пользователем, а также для подтверждения регистрации и восстановления доступа к учётной записи.
3. **Пароль** — для обеспечения безопасности аккаунта, пароль должен быть достаточно сложным и содержать как минимум одну заглавную букву, цифры и специальные символы.
4. **Дата рождения** — этот пункт является обязательным для подтверждения возраста пользователя, а также для выполнения ряда функциональных требований, связанных с правовыми аспектами доступа к определённым разделам сайта.

Форма регистрации продумана таким образом, чтобы пользователь мог без труда заполнить все необходимые поля. Для предотвращения ошибок и недочётов на странице предусмотрены подсказки, объясняющие требования к каждому из вводимых данных. Также после успешной регистрации пользователю будет предложено перейти в раздел авторизации для дальнейшего входа в систему.

Таким образом, форма авторизации и регистрационная форма создают удобный и безопасный процесс доступа к ресурсу, гарантируя, что каждый пользователь сможет легко создать аккаунт и начать полноценное использование всех возможностей сайта.

Вход на сайт осуществляется только в случае, если пользователь вводит правильные данные для авторизации — логин и пароль. При этом для повышения уровня безопасности пароль не сохраняется в базе данных в открытом виде, а хранится в виде хэша. При вводе пароля пользователем система использует функцию **password\_verify()** для сравнения введённого пароля с хэшированным значением, которое хранится в базе данных.

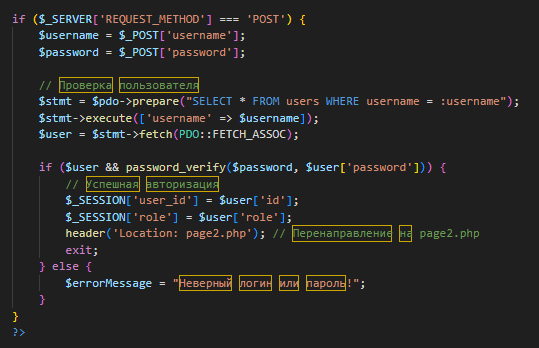


Рисунок 2.3.4 – Фрагмент кода для авторизации пользователя из файла login.php

Пример кода на PHP показывает, как происходит процесс проверки данных при входе. После отправки формы авторизации через метод **POST** система извлекает логин и пароль, введённые пользователем. Далее происходит запрос к базе данных с использованием подготовленного выражения **$stmt->prepare()**, чтобы найти пользователя по введённому логину. Если пользователь найден, система проверяет, совпадает ли введённый пароль с хэшем, сохранённым в базе данных. Для этого используется функция **password\_verify()**, которая выполняет безопасное сравнение пароля с его хэшированной версией.

В случае успешного сравнения данных, пользователю предоставляется доступ к системе, и создаются сессионные переменные, такие как **$\_SESSION['user\_id']** и **$\_SESSION['role']**, которые позволяют идентифицировать пользователя и его роль на сайте. После этого пользователя перенаправляют на защищённую страницу. В случае ошибки, например, при неверно введённом логине или пароле, выводится соответствующее сообщение об ошибке, информируя пользователя о проблеме.

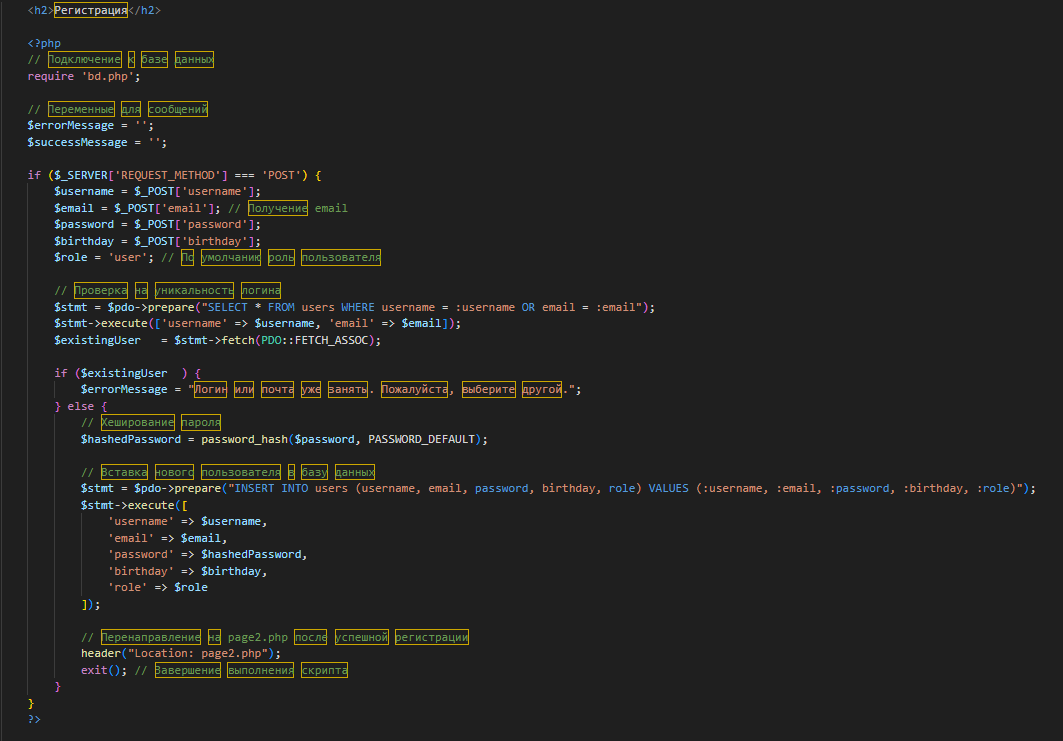


Рисунок 2.3.5 – Фрагмент кода register.php, отвечающий за регистрацию пользователей и передачу данных в bd\_users

Форма регистрации на сайте работает следующим образом. Когда пользователь отправляет данные через форму методом **POST**, скрипт получает введённые значения: логин, электронную почту, пароль и дату рождения. Вначале происходит проверка на уникальность логина и email: выполняется запрос в базу данных, чтобы убедиться, что эти данные ещё не зарегистрированы другими пользователями. Если хотя бы один из параметров уже занят, выводится сообщение об ошибке, и пользователь просит выбрать другой логин или почту.

Если логин и email уникальны, то пароль пользователя проходит процесс хэширования с использованием функции **password\_hash()**, чтобы повысить безопасность хранения пароля. После этого данные пользователя (логин, почта, хэшированный пароль, дата рождения и роль по умолчанию — «пользователь») добавляются в базу данных с помощью SQL-запроса **INSERT INTO**.

После успешной регистрации, пользователю автоматически предоставляется доступ к системе и он перенаправляется на другую страницу (например, **page2.php**), где может продолжить использовать сайт.

* 1. **Главная страница авторизованного пользователя**

Главная страница авторизованного пользователя представляет собой центральный элемент взаимодействия с сайтом, где он может сразу погрузиться в увлекательный мир JDM культуры и автомобилей. После успешного входа в систему, пользователю открывается доступ к функционалу сайта, ориентированному на активное взаимодействие с контентом и другими участниками сообщества.

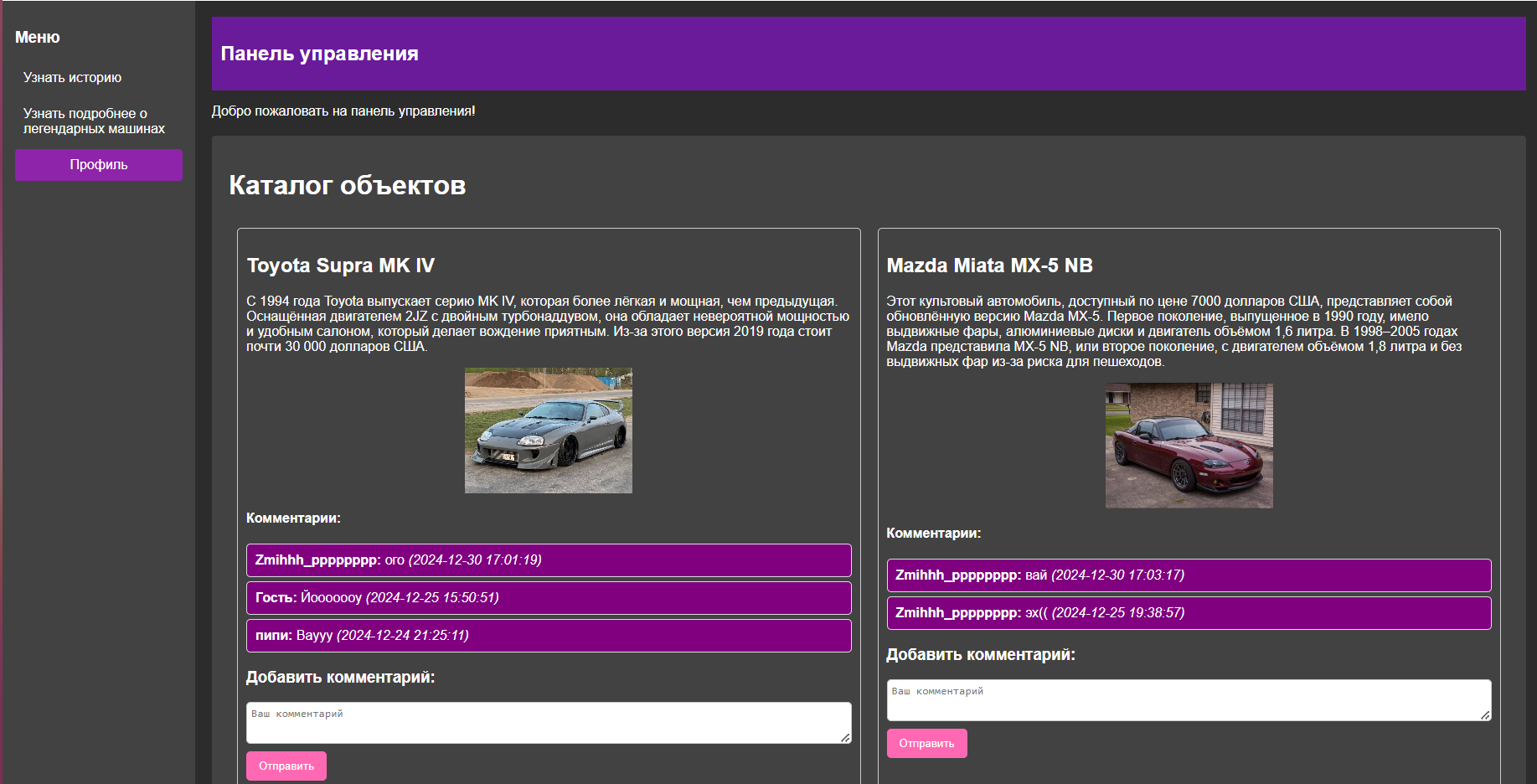


Рисунок 2.4.1 – Главная страница авторизованного пользователя (с ролью user)

**Основная часть страницы** содержит каталог японских автомобилей JDM, где каждый автомобиль представлен с подробным описанием и изображениями. Пользователь может не только ознакомиться с характеристиками автомобилей, но и принять участие в обсуждении. Под каждым автомобилем расположена форма для добавления комментариев, что позволяет пользователям обмениваться мнениями, обсуждать технические характеристики, уникальные особенности моделей и делиться личными впечатлениями от взаимодействия с машиной. Каждый комментарий отображается с указанием имени пользователя, который его оставил, а также с точной датой и временем отправки сообщения. Это создаёт прозрачность общения и позволяет отслеживать хронологию обсуждений.

Кроме того, в **левой части страницы** находится удобное меню навигации, которое помогает пользователю быстро ориентироваться в структуре сайта. Меню включает несколько разделов:

1. **Каталог машин JDM** — прямой доступ к списку всех доступных для просмотра автомобилей с возможностью фильтрации или сортировки по различным критериям.
2. **История JDM культуры** — раздел, посвящённый историческому контексту и развитию японской автомобильной культуры, где пользователи могут узнать о значимых событиях, марках и моделях, которые сыграли ключевую роль в формировании JDM движения.
3. **Профиль пользователя** — раздел, где пользователь может просматривать и редактировать свои личные данные, включая имя, email, дату рождения, а также видеть свою активность на сайте, такие как оставленные комментарии и рейтинги.

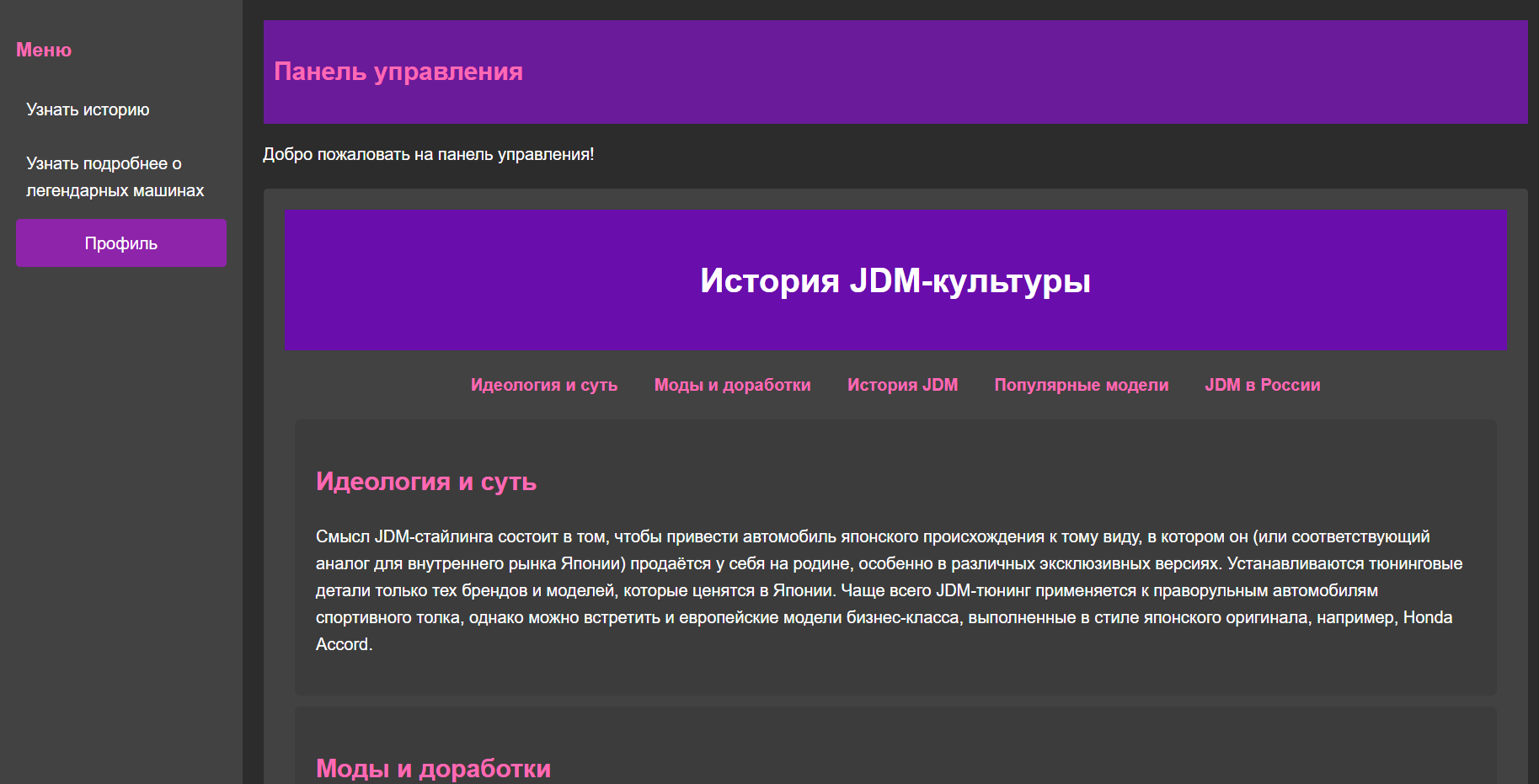


Рисунок 2.4.2 – Вкладка файла history.php для авторизованных пользователей

Страница выполнена в удобном и современном дизайне, что позволяет пользователю легко ориентироваться и быстро находить нужную информацию. Все элементы интерфейса, от каталога машин до меню, адаптированы под различные устройства, обеспечивая комфортное использование как на десктопах, так и на мобильных устройствах.

Таким образом, главная страница авторизованного пользователя не только служит отправной точкой для взаимодействия с контентом, но и создает пространство для общения и обмена мнениями среди единомышленников, делая сайт живым и динамичным.

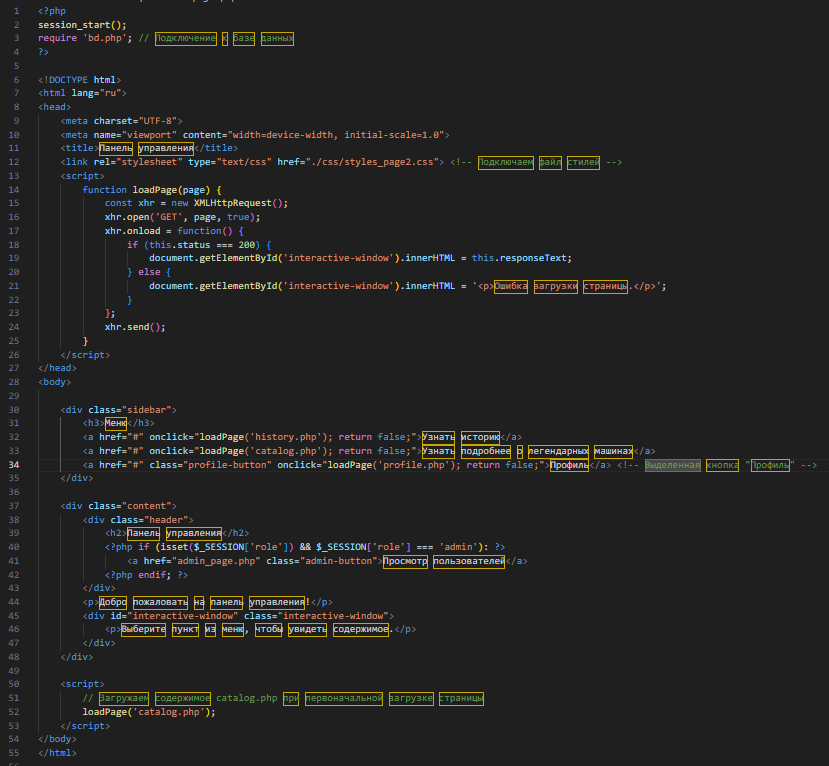


Рисунок 2.4.3 – Код страницы page2.php главной страницы сайта после авторизации

Если пользователь имеет роль администратора (проверяется через сессионную переменную **$\_SESSION['role']**), ему предоставляется доступ к кнопке **"Просмотр пользователей"**, которая перенаправляет на страницу администрирования, где можно управлять пользователями сайта.

Таким образом, данная страница обеспечивает удобную и динамичную навигацию по сайту, позволяя пользователям взаимодействовать с различными разделами без необходимости перезагружать страницу, улучшая пользовательский опыт.

* 1. **Профиль пользователя и редактирование данных**

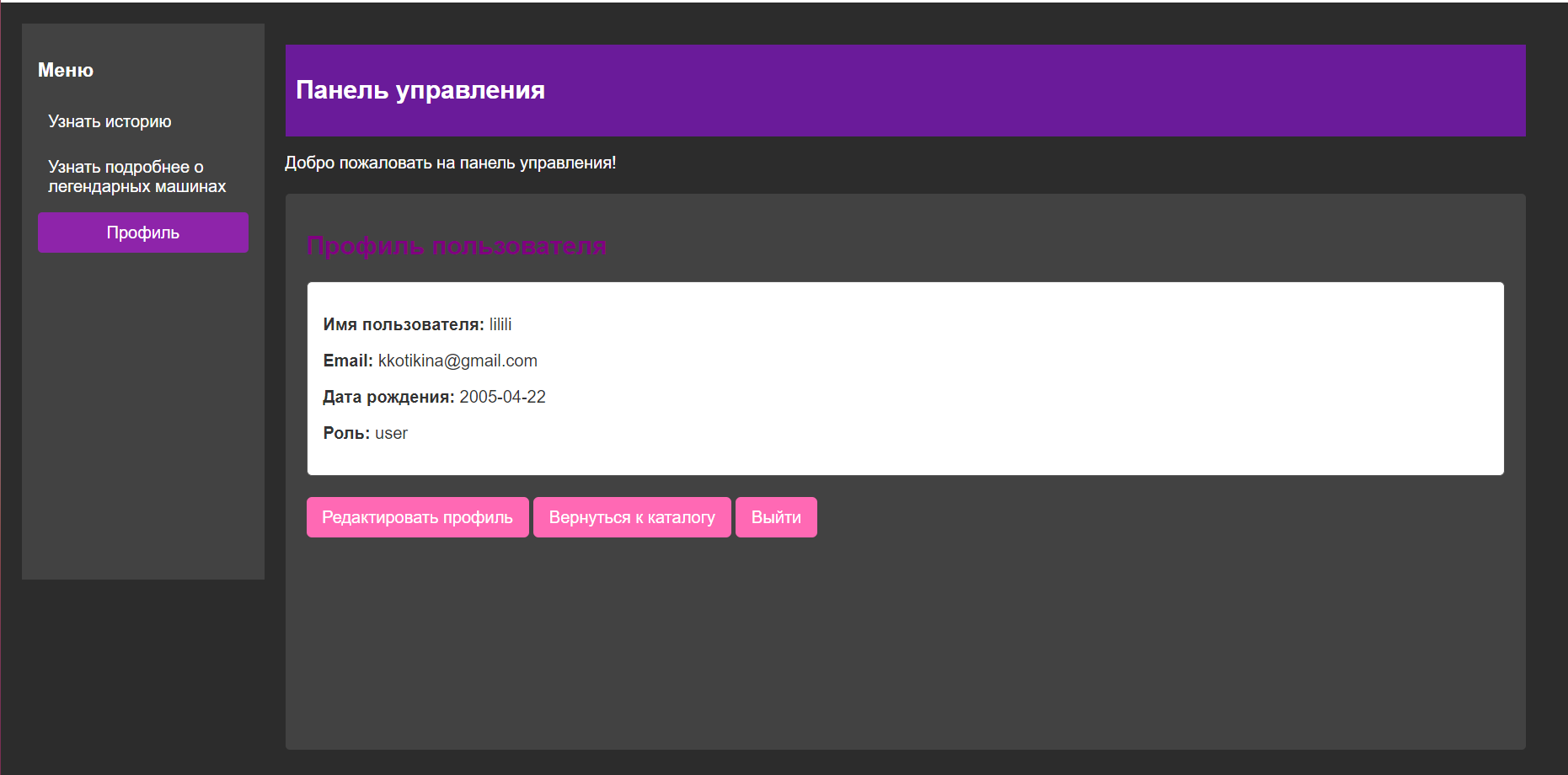
****

Рисунок 2.5.1. – Страница профиля пользователя

Страница **профиля пользователя** представляет собой личный кабинет, в котором отображается основная информация о зарегистрированном пользователе, а также предоставляет функциональные возможности для дальнейшего взаимодействия с аккаунтом. На этой странице пользователи могут увидеть свои данные, такие как:

* **Логин** — уникальное имя, которое используется для входа в систему.
* **Электронная почта** — адрес, связанный с учётной записью, через который пользователю могут быть отправлены уведомления или восстановление пароля.
* **Дата рождения** — информация о возрасте пользователя, которая была указана при регистрации.
* **Роль на сайте** — отображается роль пользователя, например, «пользователь» или «администратор», в зависимости от его прав на сайте.

В верхней части страницы находятся **кнопки** для выполнения следующих действий:

1. **Редактировать профиль** — кнопка, которая позволяет пользователю перейти к форме редактирования, где можно изменить логин, почту, пароль и другие данные профиля.
2. **Вернуться к каталогу** — кнопка для перехода обратно в каталог машин, где пользователь может продолжить изучать автомобили и их характеристики.
3. **Выйти с сайта** — кнопка для выхода из учётной записи, что завершает сессию пользователя и перенаправляет его на страницу входа.

Все эти функции делают страницу профиля не только информационной, но и удобной для пользователя, обеспечивая ему возможность управлять своими данными, а также легко возвращаться к основным разделам сайта или выходить из системы. Таким образом, страница профиля играет ключевую роль в создании персонализированного опыта для пользователей, предлагая простые и понятные средства управления учётной записью.

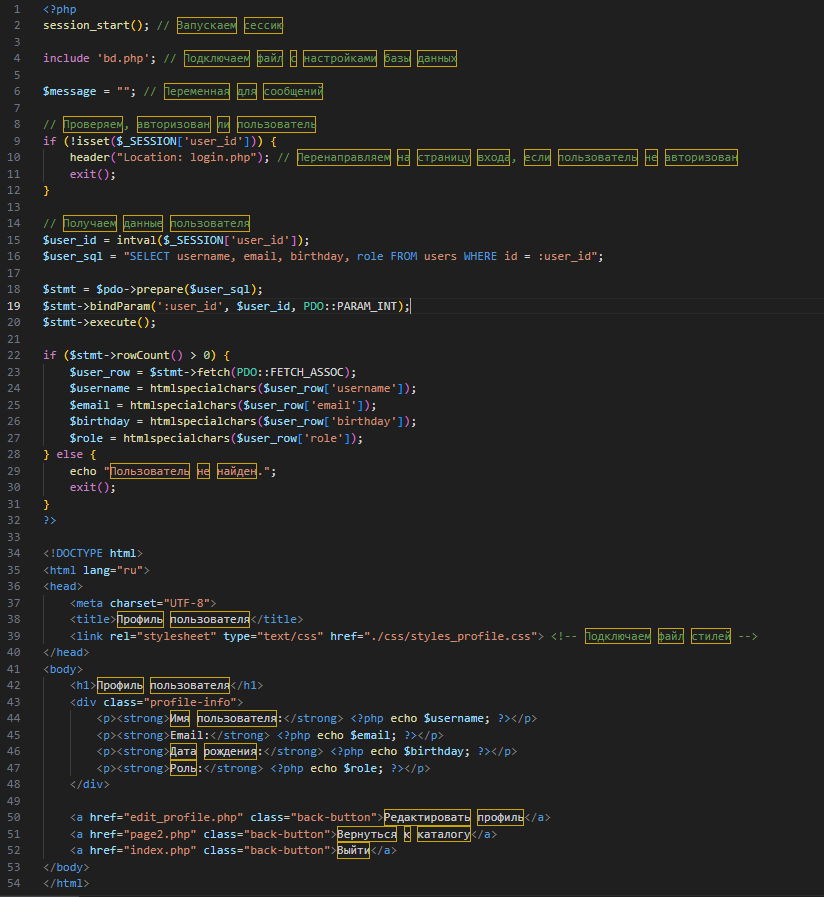


Рисунок 2.5.2 – Код страницы профиля пользователя

Код описывает работу страницы профиля пользователя на сайте. Вот краткое описание его работы:

1. **Запуск сессии**: В начале с помощью session\_start() запускается сессия, чтобы работать с данными, сохранёнными для текущего пользователя, такие как его идентификатор (**user\_id**).
2. **Проверка авторизации**: Далее проверяется, авторизован ли пользователь. Если переменная **$\_SESSION['user\_id']** не установлена, значит пользователь не вошёл в систему, и его перенаправляют на страницу входа (**login.php**) с помощью header().
3. **Получение данных пользователя**: Если пользователь авторизован, выполняется запрос к базе данных, чтобы извлечь информацию о пользователе, используя его **user\_id**. Для этого используется подготовленный SQL-запрос, где извлекаются данные: имя пользователя (**username**), почта (**email**), дата рождения (**birthday**) и роль (**role**).
4. **Вывод данных**: Если пользователь найден в базе данных, эти данные выводятся на страницу в виде HTML, с использованием **htmlspecialchars()** для защиты от XSS-атак. В противном случае выводится сообщение об ошибке.
5. **Кнопки навигации**: На странице профиля отображаются ссылки, которые позволяют:
   * Перейти на страницу редактирования профиля (**edit\_profile.php**).
   * Вернуться к каталогу машин (**page2.php**).
   * Выйти с сайта, вернувшись на страницу входа (**index.php**).

Таким образом, код обеспечивает безопасный доступ к информации пользователя и предоставляет удобные функции для управления аккаунтом.

****

Рисунок 2.5.3 – Форма редактирования данных пользователя

Страница **редактирования профиля** предоставляет пользователю возможность обновить свои личные данные, такие как логин, email и дата рождения. Страница доступна только авторизованным пользователям, что проверяется через сессию.

### Основной функционал:

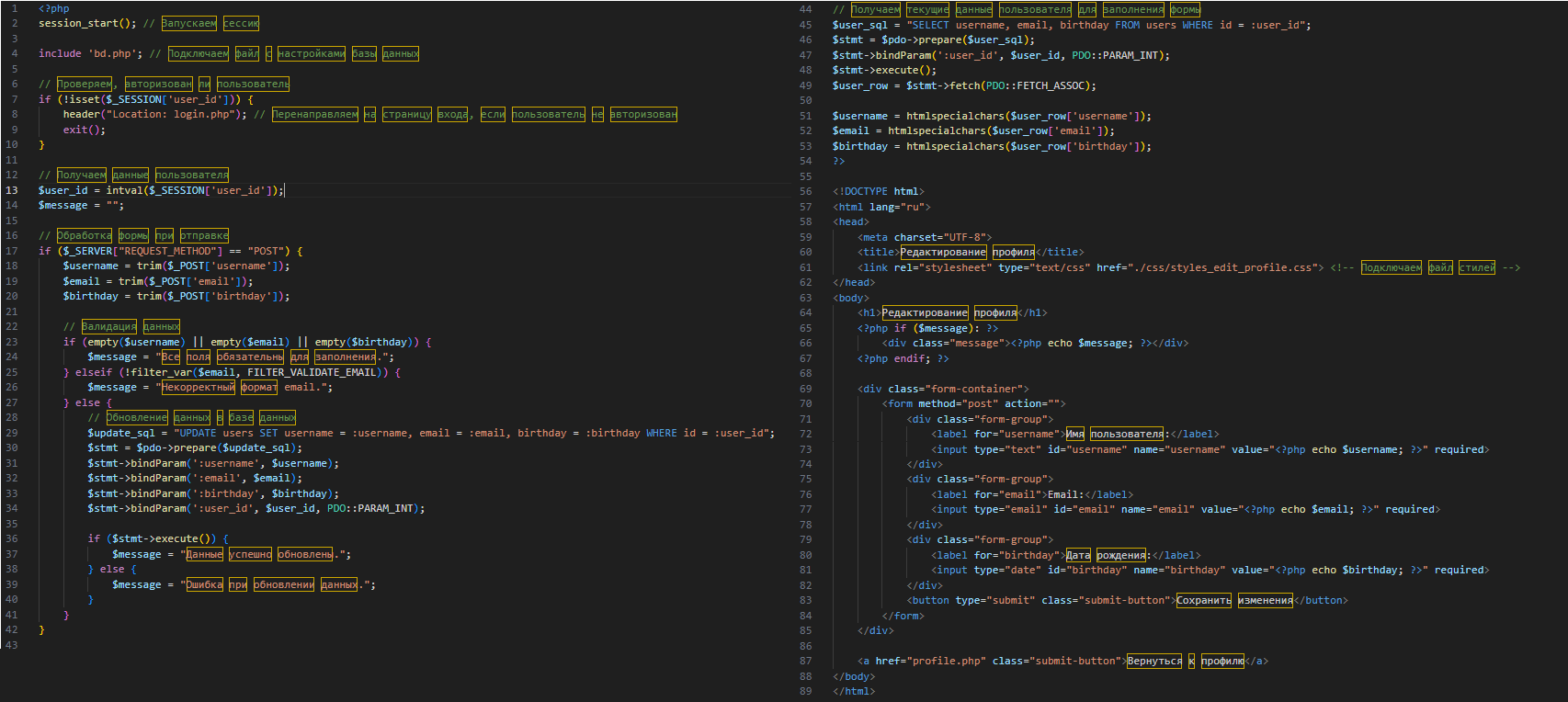
1. **Проверка авторизации**: При попытке попасть на страницу редактирования, сначала проверяется, авторизован ли пользователь. Если нет, происходит перенаправление на страницу входа (**login.php**).
2. **Получение текущих данных**: Перед отображением формы редактирования, из базы данных извлекаются текущие данные пользователя (логин, почта и дата рождения), которые затем подставляются в поля формы для редактирования.
3. **Форма редактирования**: На странице выводится форма, содержащая три поля:
   * **Имя пользователя** — для изменения логина.
   * **Email** — для изменения почты.
   * **Дата рождения** — для обновления даты рождения.

Каждое поле заполняется текущими значениями, взятыми из базы данных.

1. **Валидация и обработка данных**: При отправке формы проверяется корректность введённых данных:
   * Все поля должны быть заполнены.
   * Email должен соответствовать корректному формату.

Если данные прошли валидацию, они сохраняются в базе данных. Для этого выполняется SQL-запрос **UPDATE**, который обновляет информацию о пользователе.

1. **Сообщения об успехе или ошибке**: После отправки формы на странице отображается сообщение, информирующее пользователя о результате операции. Если обновление прошло успешно, выводится сообщение **"Данные успешно обновлены"**. В случае ошибок, например, при неправильном формате email или незаполненных полях, отображается соответствующее сообщение об ошибке.
2. **Кнопки навигации**: Внизу страницы есть кнопка для сохранения изменений и кнопка для возврата к **профилю**. При нажатии на кнопку **"Вернуться к профилю"** пользователь может вернуться к странице своего профиля, не сохраняя изменений.

****

**Рисунок 2.5.4 – Файл edit\_profile.php с кодом для реализации и функционирования страницы редактирования профиля**

Таким образом, страница редактирования профиля позволяет пользователям легко обновлять свои личные данные на сайте, обеспечивая удобство и безопасность за счёт валидации и обработки информации на сервере.

* 1. **Функционал пользователей с ролью admin**

Администратор сайта имеет расширенные права для управления контентом и пользователями. Эти возможности обеспечивают эффективное администрирование и поддержку порядка на платформе. В частности, администратор обладает следующими функциями:

1. **Удаление комментариев**  
   Каждый комментарий, оставленный пользователями на странице с автомобилями, может быть удалён администратором. Для этого рядом с каждым комментарием присутствует кнопка **"Удалить"**, которая позволяет моментально удалить нежелательный или неприемлемый комментарий. Это предоставляет администратору возможность контролировать качество контента и следить за соблюдением правил поведения на платформе. Удаление комментариев происходит с помощью AJAX-запросов, что позволяет обновить список комментариев на странице без её перезагрузки.
2. **Добавление новых записей**  
   Администратор также имеет возможность добавлять новые записи в базу данных, такие как описание новых автомобилей в каталоге или другие важные обновления, которые могут быть интересны пользователям. Эта функция позволяет поддерживать актуальность информации на сайте, делая его более интересным и функциональным для посетителей.
3. **Просмотр и управление пользователями (Рис. 2.6.1)**  
   Для обеспечения эффективного управления пользователями на сайте, администратор имеет доступ к специальной странице **"Просмотр пользователей"**. На этой странице администратор может просматривать список всех зарегистрированных пользователей, видеть их данные (логин, email, дата рождения, роль) и при необходимости:
   * **Изменять данные пользователя**: Например, администратор может обновить информацию о пользователе, если он изменит свой логин, email или другие данные.
   * **Удалять пользователей**: В случае нарушения пользователем правил или по другим причинам, администратор может удалить пользователя из системы. Это гарантирует, что на платформе останется только активное и позитивное сообщество.
   * **Добавлять новых пользователей**: Администратор имеет возможность добавлять новых пользователей в систему вручную, что может быть полезно для массовой регистрации или при восстановлении аккаунтов пользователей.

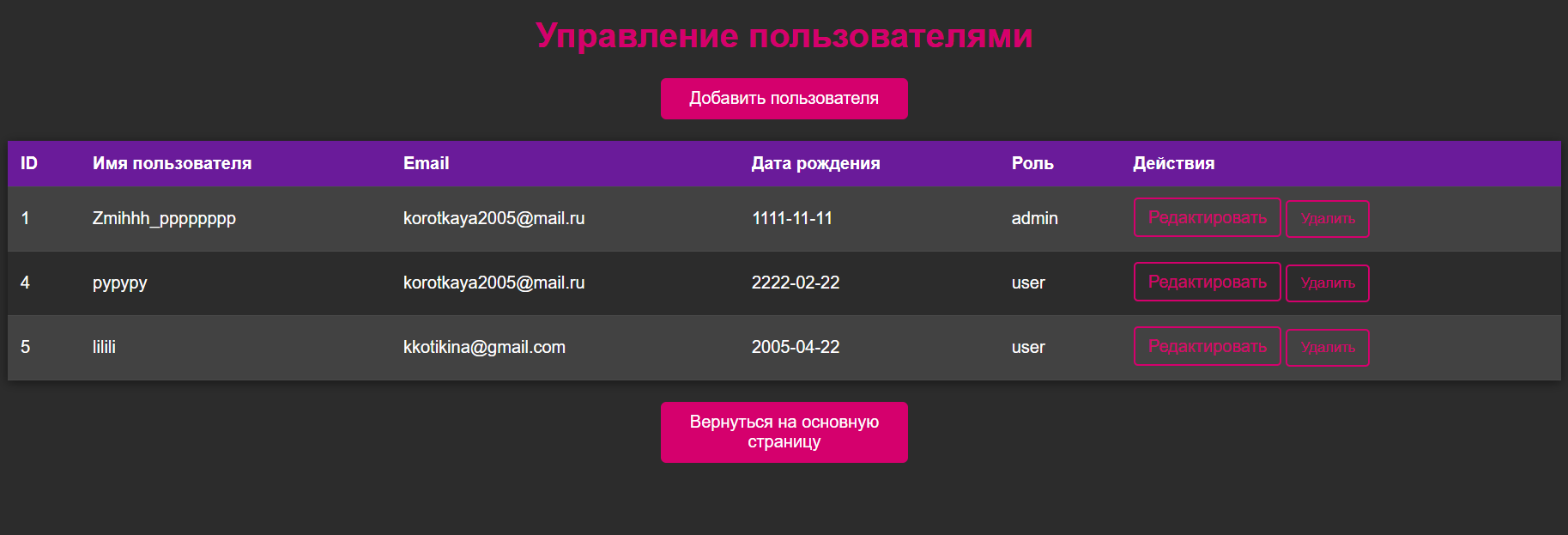


Рисунок 2.6.1 – Страница управления пользователями

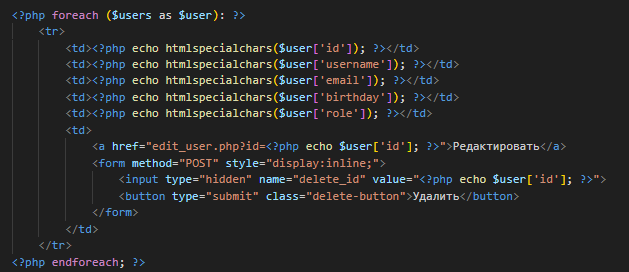


Рисунок 2.6.2 – фрагмент кода admin\_page.php, отвечающий за вывод данных всех пользователей

Таким образом, функционал, доступный администратору, направлен на поддержание порядка и удобства на сайте. Возможности управления контентом и пользователями, такие как удаление комментариев, добавление записей и администрирование аккаунтов, позволяют эффективно управлять платформой, обеспечивая её безопасность и высокое качество предоставляемых услуг.

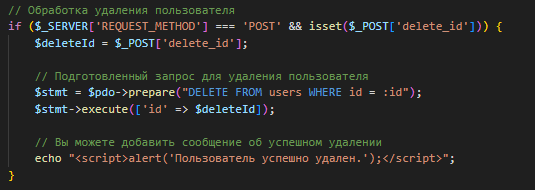


Рисунок 2.6.3 – фрагмент кода admin\_page.php, отвечающий за удаление пользователей

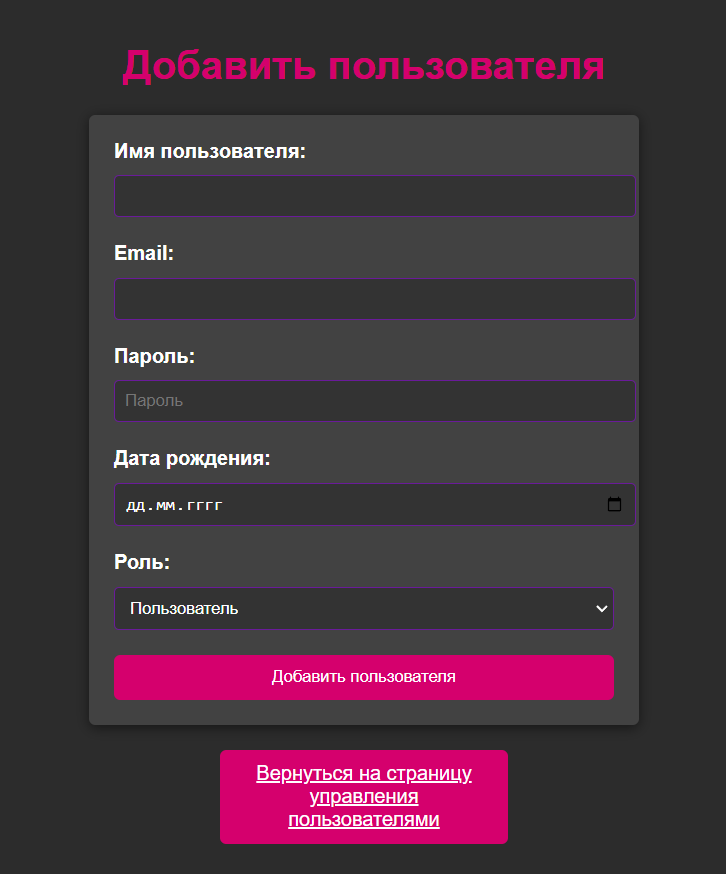
****

Рисунок 2.6.4 – Форма добавления новых пользователей

Форма добавления новых пользователей в систему выполнена аналогично стандартной форме регистрации, с той лишь разницей, что администратор имеет дополнительные возможности для управления ролями пользователей. В отличие от обычной регистрации, где роль по умолчанию устанавливается как **"user"**, в форме администрирования предусмотрен выбор роли для нового участника. Это позволяет администратору гибко назначать роли, такие как **"user"**, **"admin"** или другие, в зависимости от уровня доступа, который он желает предоставить пользователю.

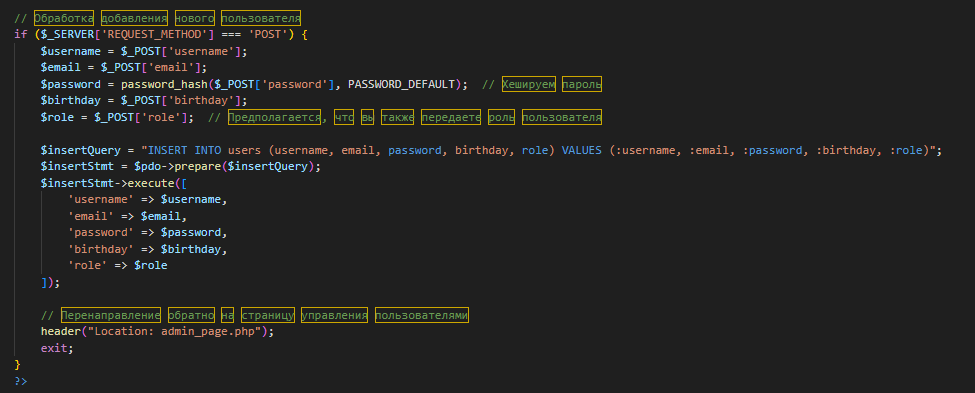
****

Рисунок 2.6.5 – фрагмент кода add\_user.php, отвечающий за добавление пользователей

Когда администратор добавляет нового пользователя, ему необходимо заполнить стандартные поля, такие как **имя пользователя**, **email**, **пароль** и **дата рождения**, точно так же, как и при обычной регистрации. Однако, в дополнение к этим полям, администратор может выбрать подходящую **роль пользователя** из выпадающего списка. Это позволяет администратору легко управлять правами доступа, обеспечивая необходимый уровень контроля над платформой.

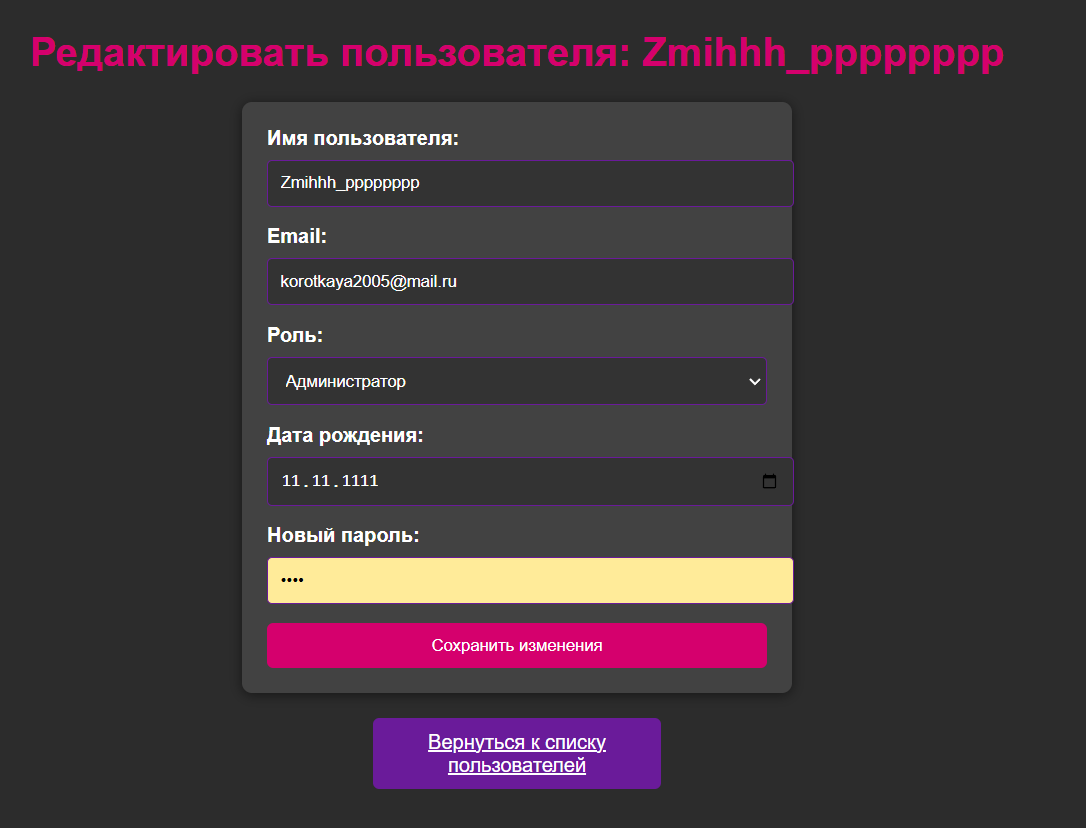


Рисунок 2.6.6 – Форма для редактирования данных пользователя

Здесь доступна та же информация, что и при добавлении пользователя, но поля уже заполнены. Исходя из этого, админ может изменять любые данные пользователя, после чего они сохранятся в базе данных.

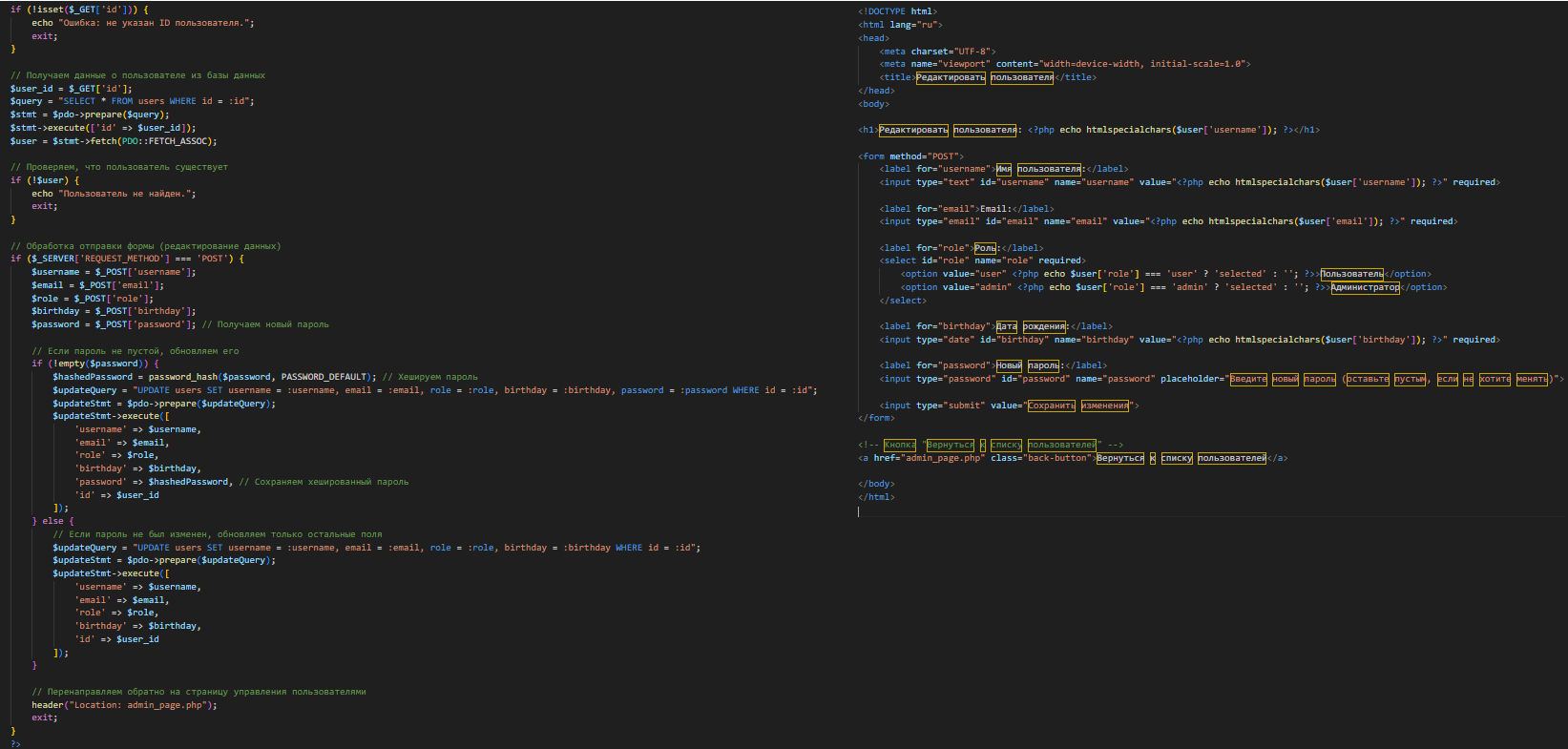


Рисунок 2.6.7 – Файл edit\_user.php, в котором прописан код для реализации админской формы редактирования пользователей

* 1. **Реализация каталога и основного функционала**

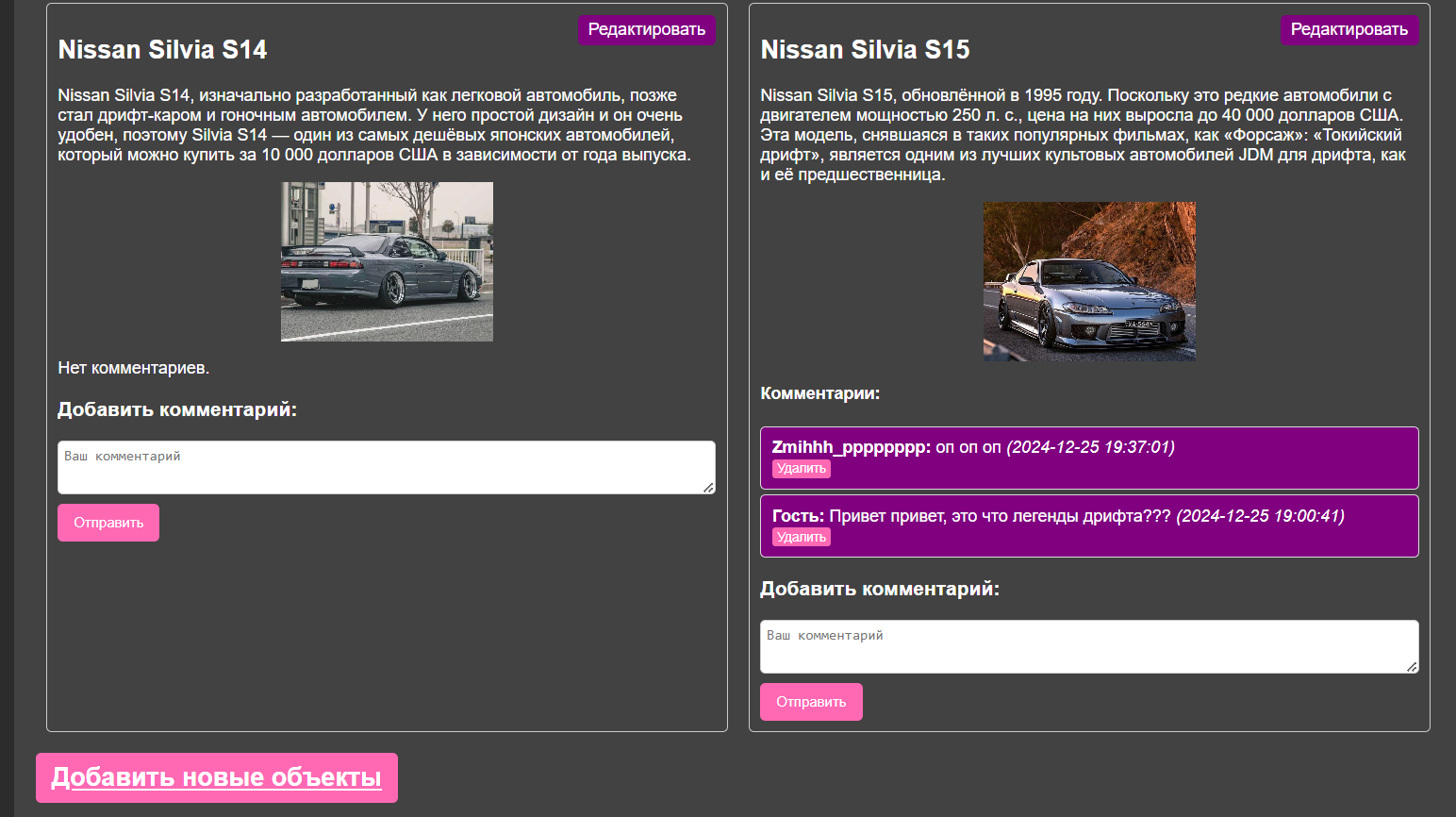


Рисунок 2.7.1 – Реализация каталога

Страница каталога на сайте представляет собой раздел, где пользователи могут ознакомиться с различными объектами, представленными в базе данных. Каждый объект отображается с названием, описанием, изображением (если оно доступно), а также имеет возможность взаимодействовать с другими пользователями через систему комментариев.

1. **Список объектов:** Каждый объект каталога отображается в виде карточки, содержащей:
   * **Название объекта**, представленное заголовком второго уровня.
   * **Описание объекта**, которое выводится в виде текстового абзаца.
   * **Изображение**, если оно привязано к объекту, показывается под его описанием с выравниванием по центру. Изображения адаптируются под размеры, указанные в базе данных, обеспечивая корректное отображение на странице.
2. **Комментарии к объектам:** Для каждого объекта отображается список комментариев, оставленных другими пользователями. Комментарии сортируются по дате отправки, начиная с самых новых. Каждый комментарий включает:
   * Имя автора комментария.
   * Текст комментария.
   * Дату и время его отправки.

Если автор комментария — администратор, он имеет возможность удалить комментарий, нажав кнопку **"Удалить"** рядом с комментарием. Это доступно только тем пользователям, чья роль в системе указана как **"admin"**.

1. **Добавление комментариев:** Все пользователи могут оставить свой комментарий под объектами каталога. Для этого существует форма с текстовым полем для ввода комментария. Каждый пользователь может добавить свой комментарий, указав свое имя, которое автоматически подставляется из данных сессии, если пользователь авторизован. В случае, если пользователь не авторизован, в поле "Автор" будет установлено значение **"Гость"**.
2. Добавление новых объектов: для всех пользователей предусмотрена кнопка для добавления новых объектов в каталог, доступная на странице.

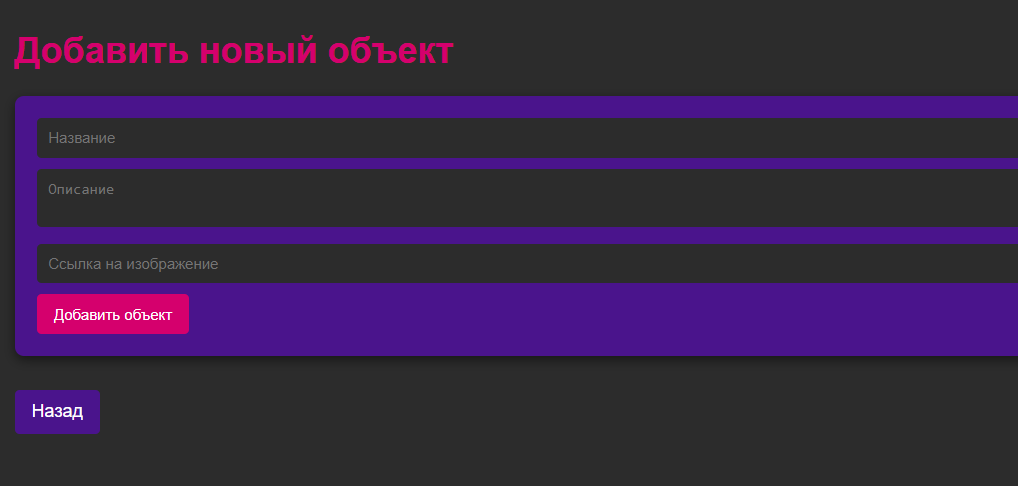


Рисунок 2.7.2 – Добавление новых объектов

1. **Админские функции:** Если пользователь является администратором (роль **"admin"**), ему открывается дополнительный функционал:
   * Редактирование объектов: Администратор может редактировать информацию об объекте, нажав на кнопку "Редактировать" рядом с его описанием.

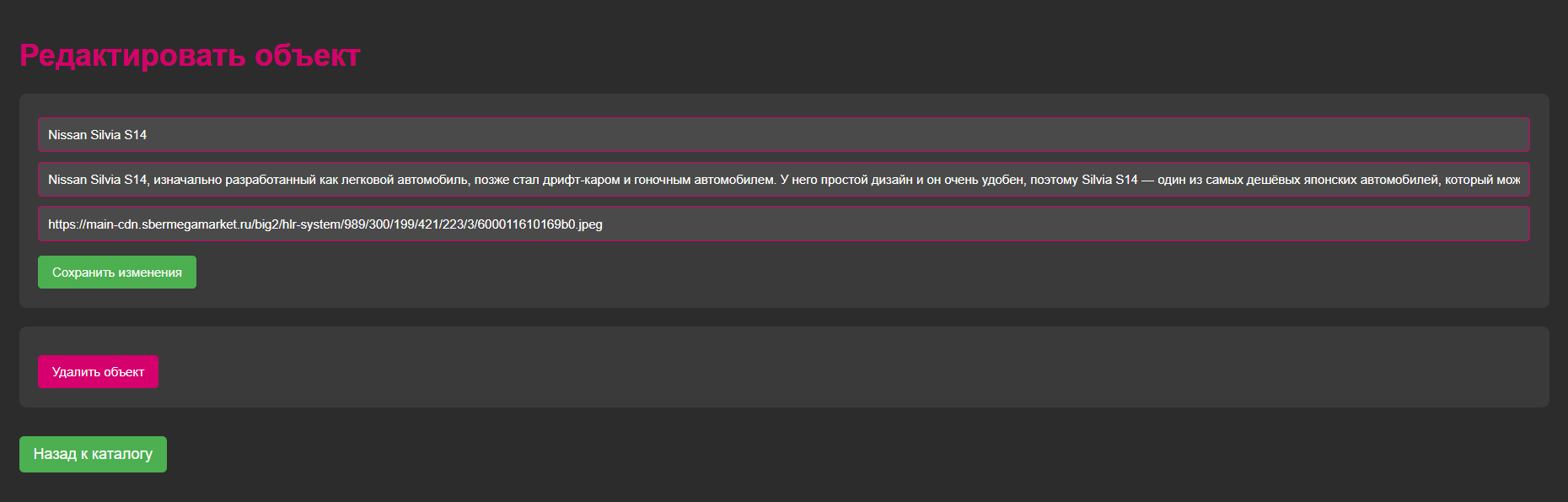


Рисунок 2.7.3 – Редактирование объектов



Рисунок 2.7.4 – Фрагмент кода functions.php с обработкой формы редактирования объектов

* + Удаление комментариев: Администратор имеет возможность удалить любой комментарий, используя кнопку "Удалить", расположенную рядом с каждым комментарием. Это помогает поддерживать порядок на сайте и исключать нежелательные сообщения.

1. **Обработка данных и взаимодействие с базой данных:** Все данные о товарах, комментариях и пользователях извлекаются из базы данных с помощью SQL-запросов. Код работает следующим образом:
   * Выполняется запрос на получение всех объектов из базы данных, и каждый объект выводится в виде карточки на странице.
   * Для каждого объекта также запрашиваются комментарии, которые выводятся ниже объекта.
   * После отправки комментария, форма передает данные на сервер для обработки, и комментарий добавляется в базу данных. В случае успешного добавления, комментарий появляется на странице без перезагрузки.

Страница каталога предоставляет пользователям удобный и интуитивно понятный интерфейс для взаимодействия с содержимым сайта. Она позволяет не только просматривать объекты, но и оставлять отзывы, комментировать, а в случае наличия прав администратора — управлять контентом, редактировать объекты и модерировать комментарии.

# **Заключение**

В ходе выполнения данного проекта была успешно разработана веб-платформа для специализированного форума, посвященного обсуждению JDM (Japanese Domestic Market) автомобилей. Основной целью проекта было создание удобного и функционального инструмента для общения автолюбителей, интересующихся японскими автомобилями, и обмена опытом, советами и новинками.

В процессе разработки веб-приложения были тщательно изучены и применены различные веб-технологии. Использование HTML позволило создать структурированный и доступный интерфейс, а CSS обеспечил его визуальное оформление, сделав его интуитивно понятным и привлекательным для пользователей. JavaScript добавил динамичности в работу сайта, что значительно улучшило взаимодействие пользователей с платформой. Работа с PHP и MySQL позволила эффективно управлять серверной частью приложения и обеспечивать хранение и обработку данных пользователей, а также динамическое взаимодействие с базой данных, что является основой для функционала форума.

Кроме того, в рамках разработки проекта были освоены важные инструменты и программы для создания локальной среды разработки, такие как Open Server Panel и phpMyAdmin. Open Server Panel обеспечил запуск и настройку локального сервера, а phpMyAdmin значительно упростил управление базой данных, предоставив удобный интерфейс для работы с MySQL. Эти инструменты позволили эффективно реализовать проект, тестировать и оптимизировать функционал веб-приложения на каждом этапе его разработки.

Созданное веб-приложение успешно решает задачу создания специализированного форума для обсуждения JDM-автомобилей. Оно предоставляет пользователям не только возможность общения, но и пространство для обмена знаниями и опытом в области модификации автомобилей, новинок японской автомобильной культуры и технических решений. Система комментирования, регистрации пользователей и управления контентом на форуме отвечает современным требованиям безопасности и функциональности.

В результате проделанной работы был создан инструмент, который способствует формированию сообщества единомышленников и укреплению связей между автолюбителями, увлеченными JDM-автомобилями. Форум предоставляет всем участникам возможность получить актуальную информацию и поделиться своими достижениями и идеями, создавая уникальное пространство для людей, стремящихся развивать свою страсть к японским автомобилям.

Таким образом, в рамках данного проекта были успешно реализованы ключевые аспекты разработки современного веб-приложения, что позволяет утверждать, что поставленные цели и задачи были достигнуты. В будущем проект может быть расширен и дополнен новыми функциями, такими как система личных сообщений, рейтинги пользователей, а также возможность обмена фотографиями и видео, что позволит создать еще более привлекательную и насыщенную платформу для автолюбителей.

# **Список использованных источников**

1. Главная страница OpenServer // operserver URL: <https://ospanel.io> (дата обращения 01.12.2024).
2. Главная страница phpMyAdmin // phpMyAdmin URL: <https://www.phpmyadmin.net/downloads/> (дата обращения 01.12.2024).
3. Документация MySQL. // metanit URL: <https://metanit.com/sql/mysql/> (дата обращения: 04.12.2024).
4. Руководство по HTML и CSS // metanit URL: <https://metanit.com/web/html5/> (дата обращения: 04.12.2024).
5. Руководство по JS // metanit URL: <https://metanit.com/web/javascript/> (дата обращения: 09.12.2024).
6. Руководство по PHP // metanit URL: <https://metanit.com/php/tutorial/> (дата обращения: 07.12.2024).
7. PHP, MySQL, JavaScript, CSS и HTML5 / Робин Никсон 6-е издание, 2023. – 832 с. (дата обращения: 07.12.2024).
8. **PHP и MySQL. Разработка веб-приложений** / Д. Колисниченко 6-ое издание, 2017 – 640 с. (дата обращения 03.12.2024).