Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ

**ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ**

РАЗРАБОТКА ОБУЧАЮЩЕЙ ПЛАТФОРМЫ Weber

Курсовой проект по учебной дисциплине

«Проектирование и разработка приложений»

по специальности 09.02.07 Информационные системы и программирование

ПТК.КП 2994 11.000ПЗ

|  |
| --- |
| Руководители  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |
| Студент группы 2994  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 года |

Содержание

[Введение 3](#_Toc732469090)

[1 Общая часть 4](#_Toc251429818)

[1.1 Постановка задачи 4](#_Toc1226589138)

[1.2 Обоснование проектных решений 7](#_Toc118287061)

[1.3 Обзор и анализ существующих программных систем 13](#_Toc1401518701)

[2 Специальная, практическая, опытно-экспериментальная часть 15](#_Toc133847113)

[2.1 Анализ задачи 15](#_Toc1593486195)

[2.2 Описание логической структуры 16](#_Toc1022833104)

[2.3 Описание работы программы 20](#_Toc1527941974)

[Заключение 28](#_Toc2135376068)

[Список литературы 29](#_Toc1965071321)

[Приложение А 30](#_Toc889518387)

[Приложение Б 31](#_Toc322911230)

[Приложение В 32](#_Toc1379683225)

[Приложение Г 33](#_Toc1745637062)

[Приложение Д 40](#_Toc1107111296)

# Введение

В соответствии с заданием на курсовое проектирование требуется разработать платформу для обучения веб-программированию.

Актуальность исследования обусловлена тем, что проблема выбора профессии является неотъемлемой частью жизни человека, особенно в подростковом возрасте. Веб-программирование является одной из самых востребованных профессий в современном мире. С учетом быстрого развития технологий и перехода многих компаний в онлайн-пространство, специалисты в области веб-программирования становятся все более ценными. Они способны создавать и поддерживать веб-сайты и приложения, которые становятся ключевыми для успешной работы бизнеса.

Таким образом, задача данного курсового проектирования является актуальной.

Целькурсового проекта – разработать обучающую интернет-платформу.

Задачи курсового проекта:

* знакомство с предметной областью исследуемой задачи;
* разработка модели базы данных (реляционную);
* создание структуры базы данных средствами веб-интерфейса phpmyadmin для администрирования СУБД MySQL;
* проектирование функциональных возможностей и создание веб-сайта;
* организация поиска необходимой по критерию информации;
* разработка руководства пользователя.

Гипотеза: для людей, которые ищут профессию с большим спросом и перспективами на будущее, веб-программирование может стать отличным выбором. Обучение на нашей платформе поможет вам приобрести необходимые навыки и знания для начала успешной карьеры в этой области. Так же обучающая платформа для веб-программирования способна значительно улучшить навыки и компетенции студентов обучающихся в этой области.

1. Общая часть
   1. Постановка задачи
      1. Обоснование необходимости разработки

Задачей данного курсового проектирования является проектирование и разработка платформы для обучения веб-программированию. Требуется разработать информационную систему для отображения обучающих курсов и связанных с ними функций (просматривания активных курсов, просматривания завершенных курсов и т.д.).

* + 1. Технико-математическое описание задачи

Для реализации были использованы следующие средства:

HTTP (от англ. HyperText Transfer Protocol) — протокол передачи гипертекста. Это набор правил, по которым данные в интернете передаются между разными источниками, обычно между компьютерами и серверами. Интернет-протокол HTTP — это шаблон, по которому формируется запрос на передачу данных, а затем передаются интернет-страницы, видео, аудио и текст. Чаще всего с помощью HTTP передают веб-страницы, то есть контент сайтов, которые отображаются в интернете.

Протокол HTTP нужен для стандартизации. Благодаря ему все компьютеры в интернете могут расшифровать присланные данные и отправлять их в виде, понятном другим компьютерам.

Laravel – это бесплатный PHP-фреймворк с открытым исходным кодом. Laravel разработали в качестве помощника при создании сложных веб-ресурсов и приложений. С его помощью специалисты упрощают процесс аутентификации, а также работу с БД, кэширование, сессии, структуру приложения, маршрутизацию и другие не менее важные процессы.

Возможностей у платформы Laravel немало. Одна из них – построение логичной архитектуры для проектов любой сложности и типа. Платформа характеризуется:

* высокой производительностью;
* возможностью интеграции с другими платформами, а также библиотеками;
* немалым количеством интересных возможностей для разработчиков сайтов и приложений.

HTML (HyperText Markup Language) переводится, как язык разметки гипертекста. HTML применяется для верстки веб-страниц. Она нужна браузерам, которые преобразуют гипертекст и выводят на экран страницу в удобном для человека формате. Т.е. при переходе на веб-сайт браузер скачивает HTML-документ с сервера, имеющий специальные метки, называемые тегами, которые представляют собой парную (как правило) конструкцию, которая нужна для того, чтобы задать определенные свойства тексту либо каким-то другим информационным данным, находящимся внутри этой конструкции, впоследствии опознаются браузером и используются им для отображения содержимого файла на экране компьютера. С помощью этих меток можно выделять заголовки документа, изменять цвет, размер и начертание букв, вставлять графические изображения и таблицы. Но основным преимуществом гипертекста перед обычным текстом является возможность добавления к содержимому документа гиперссылок -- специальных конструкций языка HTML, которые позволяют щелчком мыши перейти к просмотру другого документа.

JavaScript - oдин из самых распространенных языков. Часто начинающие программисты путают Java и JavaScript. Несмотря на созвучное название, это два совершенно разных языка. Область его применения обширна и практически безгранична. На JavaScript пишут серверные, мобильные и компьютерные приложения. Любой браузер и любая операционная система хорошо знакома с JavaScript. Все сценарии выполняются непосредственно в браузере устройства, пользователю не нужно предпринимать каких-либо действий. В большинстве случаев он используется для создания простых анимаций, скриптов и объектов пользовательского интерфейса. Позволяет подключать интерактивные элементы: карты, графику, кнопки, формы обратной связи. Именно JS отвечает за то, что сайт будет реагировать на ваши действия: обновлять страницу, запускать видео или отправлять запрос на обратный звонок.

CSS или Cascading Style Sheets переводится как «каскадная таблица стилей» — язык описания внешней страницы. Он отвечает за визуальную красоту и оформление сайта.

Стили — это визуальные особенности, которые нужно придать странице, например цвет текста, размер подзаголовка, тип шрифта или оформление иллюстрации.

Стили или свойства CSS называются правилами. Каждое правило состоит как минимум из одного селектора и одной пары «свойство — значение».

Селектор нужен для того, чтобы браузер понимал, к какому элементу CSS будет применяться свойство: заголовку, абзацу, фоновому цвету.

Свойство показывает, что именно будет меняться в этом элементе: цвет, размер, выравнивание. Значение определяет, как именно изменится элемент: станет красным, увеличится вдвое, примет круглую форму.

* + 1. Характеристика бизнес-процессов

Так как целью проекта является проектирование и разработка платформы для обучения веб-программированию, следовательно, необходимо определить основные функции, которые будут выполняться приложением у разных пользователей в системе.

Разобьём пользователей информационной системы на две группы: пользователи, которые будут иметь неограниченный и ограниченный тип доступа (приложение А).

Для графического изображения того, как организованны бизнес–процессы организации построена диаграмма бизнес – прецедентов (приложение Б).

Исходя из вышеизложенного, создание информационной системы сводится к следующим этапам:

* описание предметной области, то есть созданием модели предметной области;
* проектирование информационной модели базы данных;
* разработка базы данных в Laravel;
* разработка обучающей веб-платформы;
* подключение базы данных к приложению;
* описание справочного руководства пользователя.
  + 1. Требования к программе

Программный продукт должен быть разработан на языке высокого уровня программирования. Программа должна обеспечивать:

* удобный интерфейс;
* лёгкость в использовании.
  + - 1. Аппаратные требования

Для создания и использования программы необходимы следующие минимальные аппаратные требования:

* процессор с частотой 1 ГГц;
* оперативная память – 512 Mb;
* жесткий диск - 20 Гб;
* видеокарта 512 Мб.
  + - 1. Указание системного программного обеспечения

Для создания и использования программы необходимо следующее программное обеспечение:

* Windows XP и выше, Linux и выше;
* браузеры: Internet Explorer 7.0 и выше, Firefox 3.6 и выше, Chrome.
  1. Обоснование проектных решений
     1. Обоснование выбора языков программирования

Для языков разметки были выбраны: HTML и CSS.

Преимущества HTML:

* оригинальность (Полностью весь дизайн и функционал будет подогнан именно под ваши нужды и пожелания, а не адаптирован из какого-либо общедоступного шаблона для обобщенных задач);
* SEO оптимизация (В отличие от CMS и конструкторов, продвижение в поисковиках таких сайтов происходит намного легче ввиду отсутствия шаблонности данных, дублей страниц и продолжительной загрузки);
* быстродействие (На сервере хранятся реальные HTML файлы, которые тот сразу выдает при соответствующем запросе посетителя. Причем выдает настолько быстро, насколько это только возможно! И никаких генераций! Вы не упустите посетителя из-за длительной загрузки веб-страниц);
* безопасность (Никому уже не секрет, что все динамические CMS страдают от найденных уязвимостей, в то же время статичные сайты не боятся попыток хакеров найти лазейки в коде. Не нужно устанавливать патчи безопасности. К слову, 79% всех сайтов на WordPress уязвимы перед вредоносными программами, такими как эксплойты);
* надежность (Чем проще – тем лучше. И надежнее! Нет нужды в никаких PHP и MySQL компонентах. Когда ваш сайт вырастет – не нужно менять тарифный план из-за большей базы данных. Да и сам сайт не начнет открываться через раз, потому что PHP просит слишком много памяти);
* недорогой хостинг (Чтобы выложить свое детище на хостинг не придется влететь в копеечку и переплачивать за больший объем необходимого пространства);
* резервное копирование (Если придется переехать на новый хостинг – достаточно всего на всего скопировать папки с HTML-файлами в пункт назначения. Никаких дамп базы данных и последующего воссоздания, настроек PHP и MySQL);
* ничего лишнего (Не нужно абсолютно никаких плагинов и расширений, которые требуют постоянного и своевременного обслуживания, к тому же ещё и утяжеляют веб-ресурс. Не нужно устанавливать очередное обновление движка или отдельных модулей из-за дыр в системе безопасности);
* высокая производительность (Отсутствие фрагментов лишнего или же незадействованного кода, который присущ всем CMS, не будет тормозить и мешать вашему сайту выполнять нужные действия);
* высокая стабильность (В случае, если была допущена ошибка на одной из страниц сайта – на общей работе всего ресурса это не отразится, и он будет дальше уверенно и стабильно работать).

CSS (Cascading Style Sheets – каскадные таблицы стилей) – язык описания внешнего вида документа, созданного при помощи языка разметки.

Концепция каскадных таблиц стилей была предложена Хоконом Виум Ли – норвежским ученым и специалистом в области информационных технологий, работавшим в то время на консорциум W3C.

Как правило, CSS применяется при работе с HTML и XHTML языками, и используется для задания цветов, параметров шрифтов, расположения блоков и иных элементов представления веб - страниц.

К преимуществам использования CSS относятся:

* централизованное управление отображением множества документов при помощи одной таблицы стилей;
* упрощенный контроль внешнего вида веб - страниц;
* наличие разработанных дизайнерских техник;
* возможность использования различных стилей для одного документа, в зависимости от устройства, при помощи которого осуществляется доступ к веб - странице.

Стандарт HTML 4.0 помимо CSS утвердил и объектную модель браузера (Browser Object Model – BOM), о которой следует сказать отдельно.

Объектная модель браузера описывает содержимое веб - документа, т.е. сама модель является набором объектов, описывающих указанное содержимое. Поскольку BOM уникальна для каждого браузера, возникали проблемы с межплатформенными приложениями. В конечном итоге, на место объектной модели браузера пришла объектная модель документа (Document Object Model – DOM), описывающая стандарт представления веб - страниц в виде набора объектов.

Для языков программирования были выбраны:

JavaScript.

Преимущества JavaScript:

* незаменимость в веб-разработке. JS – это основная «технология» для клиент-серверных моделей и программирования «для интернета»;
* наличие полной интеграции с версткой страниц и серверной частью;
* рациональность применения и простоту. Решить элементарную задачу при помощи JS удается за несколько минут. Сам ЯП обладает понятным синтаксисом, который не требует углубленного изучения;
* скорость работы. JS может сделать итоговое приложение более быстрым и удобным;
* производительность;
* комфортность использования пользовательских интерфейсов;
* наличие собственной мощной экосистемы (инфраструктуры). Особо заметно в последние годы. Пример – появление огромного количества полезных фреймворков, которые подойдут для любого случая.
  + 1. Инструментальные средства

Laravel - это мощный и гибкий фреймворк для разработки веб-приложений на языке программирования PHP. Laravel предлагает богатый набор функций, которые значительно упрощают разработку веб-приложений.

Он предоставляет интуитивную и понятную структуру, которая помогает облегчить многие общие задачи, связанные с разработкой веб-приложений, такие как маршрутизация, аутентификация, управление сессиями, кеширование и многое другое.

Laravel следует архитектурному паттерну MVC (Model-View-Controller), что позволяет разработчикам организовывать свой код логически и эффективно.

Кроме того, Laravel обладает мощным ORM (Object-Relational Mapping), интеграцией с front-end библиотеками и фреймворками, удобными инструментами для работы с базами данных и многочисленными другими возможностями, делающими его выбором номер один для многих разработчиков PHP.

Open Server Panel - это мощный инструмент для управления сервером. Он предоставляет широкий спектр функций, позволяющих настраивать и контролировать различные аспекты сервера.

С его помощью можно устанавливать и удалять программное обеспечение, включая операционные системы и приложения. Он также позволяет мониторить использование ресурсов сервера, включая процессор, память и дисковое пространство, чтобы обеспечить эффективное использование ресурсов.

Важной особенностью Open Server Panel является возможность управления безопасностью. Это включает в себя установку и обновление брандмауэров, антивирусного программного обеспечения и других мер безопасности, а также мониторинг за угрозами безопасности.

В общем, Open Server Panel - это универсальный инструмент, который может значительно облегчить управление сервером и повысить его производительность и безопасность.

* + 1. Обоснование выбора среды программирования

Visual Studio Code (VS Code) — это редактор [кода](https://blog.skillfactory.ru/glossary/programmnyj-kod/" \t "_blank) для разных [языков программирования](https://blog.skillfactory.ru/glossary/yazyk-programmirovaniya/" \t "_blank). Он относительно немного весит, гибкий и удобный. В нем можно писать, форматировать и редактировать код на разных языках.

VS Code не стоит путать с Visual Studio — это [IDE](https://blog.skillfactory.ru/glossary/ide/" \t "_blank), очень мощная и масштабная, но одновременно с этим тяжеловесная. Названия похожи, потому что у обоих средств разработки один и тот же создатель, но продукты разные по своей сути. VS Code меньше весит, проще в освоении и подходит в том числе для начинающих разработчиков.

Редактор кода существует для всех популярных [операционных систем](https://blog.skillfactory.ru/glossary/operaczionnaya-sistema/" \t "_blank): Windows, Linux и macOS. Он бесплатный, в отличие от большинства версий «старшего брата» Visual Studio.

VS Code позволяет легко писать, форматировать и редактировать код на разных языках. С его помощью можно быстро создать проект и структуру файлов в нем, он подсвечивает [синтаксис](https://blog.skillfactory.ru/glossary/sintaksis/" \t "_blank) кода и помогает автоматически править ошибки. В нем есть возможности для [отладки](https://blog.skillfactory.ru/glossary/otladka-debugging/" \t "_blank) и запуска кода на некоторых языках.

Редактор легко расширяется, поэтому к перечисленным функциям можно добавить новые — достаточно просто скачать нужное дополнение из официального каталога. Дополнения тоже распространяются бесплатно.

Преимущества:

* простота освоения. У редактора интуитивно понятный интерфейс, его легко установить и настроить. А после этого им сразу можно начинать пользоваться. Единственной проблемой поначалу могут стать горячие клавиши и дополнительные настройки — они необычно организованы. Но со временем легко разобраться, особенно если вначале держать под рукой памятку;
* малый вес. По сравнению с IDE редактор быстро работает и мало весит, не так требователен к ресурсам компьютера. При этом он довольно функционален, а огромное количество расширений позволяет добавлять в него новые возможности;
* кроссплатформенность. Visual Studio Code есть для всех популярных операционных систем, так что с ним можно работать вне зависимости от ОС, которую вы предпочитаете. К тому же это позволяет быстрее и удобнее переносить проекты с одного устройства на другое;
* поддержка разных языков. Есть инструменты, которые предназначены специально для какого-то языка, например Python или C++. Visual Studio Code поддерживает огромное количество языков программирования, и они легко подключаются. Поэтому с его помощью можно вести проекты, написанные на нескольких языках одновременно;
* гибкость. У VS Code множество расширений и удобных функций. Помимо вещей, стандартных для редактора кода, вроде подсветки синтаксиса, он «умеет» много чего еще. Его можно легко и гибко настроить под себя в зависимости от потребностей разработчика и технологий, с которыми он работает.
  + 1. Информационное обеспечение

При написании курсовой работы была использована программа Microsoft Word. Это тестовый процессор, предназначенный для создания, просмотра, редактирования и форматирования текстов статей. Для защиты курсовой работы и отображения результата проекта использована программа Microsoft Power Point, которая служит для подготовки презентаций и их просмотра. При проектирование реляционной схемы базы данных и схемы функциональной модели использовалась программа diagrams.net. Это бесплатное приложение для построения диаграмм, позволяющее пользователям создавать схемы, модели и диаграммы и обмениваться ими в веб-браузере.

* 1. Обзор и анализ существующих программных систем

Перед разработкой программного продукта требуется произвести поиск аналогов.

Codecademy - онлайн-платформа для обучения кодированию. Она предлагает интерактивные уроки по различным языкам программирования, включая HTML, CSS, JavaScript, Python и Ruby.

Плюсы: интерактивные уроки помогают понять материал глубже, обучение с нуля идеально подходит для новичков, широкий выбор языков программирования позволяет изучить разные направления.

Минусы: Большинство курсов доступны только по платной подписке, что может быть преградой для некоторых пользователей.

FreeCodeCamp - это сообщество, которое помогает людям учиться кодировать, создавать проекты для некоммерческих организаций, и получать опыт работы.

Плюсы: Бесплатность обучения делает его доступным для всех, общество поддержки помогает преодолевать проблемы и учиться на опыте других, реальные проекты для портфолио помогают набрать опыт и показать свои навыки.

Минусы: может быть сложным для новичков без предварительных знаний, так как материалы иногда требуют базового понимания кодирования. Mozilla Developer Network - один из наиболее авторитетных ресурсов для веб-разработчиков. Включает всю необходимую информацию по HTML, CSS и JavaScript.

Плюсы: отличный источник информации для веб-разработчиков, содержит подробные руководства и документацию по множеству тем.

Минусы: не предлагает структурированных курсов или интерактивного обучения, что может затруднить изучение для новичков.

1. Специальная, практическая, опытно-экспериментальная часть
   1. Анализ задачи
      1. Информационное моделирование предметной области

В соответствии с выданным заданием на курсовую работу требуется разработать обучающую платформу Weber. Сервис предназначен для обучения веб-программированию.

* + - 1. Построение информационной модели

Работа над сервисом началась с проектирования и разработки базы данных.

Для графического изображения того, как организованна база данных построена IDEF1x модель (приложение В).

* + 1. Проектирование пользовательского интерфейса

Дизайн пользовательского интерфейса был разработан в приложении Figma.

Figma (Фигма) – это онлайн-сервис для дизайнеров и веб-разработчиков. В нем прорабатывают интерфейсы для мобильных приложений и сайтов. У него невысокие требования к подготовке пользователей и большой потенциал при относительной простоте. Ключевой особенностью является возможность совместной работы в режиме реального времени.

#### Интерфейс

Логотип имеет переход на главную страницу веб-приложения.

Пункт меню «Каталог курсов» имеет переход на страницу каталог, на которой представлены все курсы, взятые из базы данных.

Пункт бургер меню «Профиль» имеет переход на страницу «Профиль», на которой представлены все данные пользователя, активные и выполненные курсы, взятые из базы данных.

Кнопка «Авторизация» открывает диалоговое окно авторизации и выполняет вход пользователя в систему.

Кнопка «Регистрация» открывает диалоговое окно регистрации и выполняет регистрацию пользователя в систему, если у клиента не имеется аккаунта.

Кнопка «Выход» имеет переход на главную страницу не вошедшего в систему пользователя.

Кнопка «Панель администратора» имеет переход на страницу администратора, на которую может попасть только пользователь, имеющий права администратора в веб-приложении.

* + - 1. Интерфейс приложения

Приложение осуществляет различные процедуры с курсами, такие как: просмотр, переход между ними, добавление их в активные, просмотр уроков, находящихся в них.

Посмотреть наброски и готовый результат страниц интерфейса можно в приложении (рисунок Г1-Г18, приложение Г).

* 1. Описание логической структуры

Алгоритм программы — это точное предписание (совокупность последовательных шагов, схема действий), которое определяет процесс перехода от первичных данных к желаемому результату.

Программа обрабатывает получаемые через интерфейсы управления команды оператора, а также получаемые через сетевые интерфейсы Ethernet-кадры. Обработка выполняется с учетом стандартов на используемые протоколы, настроек Программы и состояния аппаратных модулей технического средства. В случае возникновения необходимости модифицируется состояние аппаратных модулей технического средства, генерируются, отправляются и пересылаются через сетевые интерфейсы технического средства Ethernet-кадры.

* + 1. Используемые методы

PHP:

* view() - отображение view шаблона;
* validated() - валидация формы;
* attempt() - проверяет, есть ли пользователь с предоставленными учетными данными;
* back() - переадресация на предыдущую страницу;
* withErrors() - вывод ошибки;
* session() – запуск сессии пользователя;
* regenerate() - воссоздание сессии пользователя;
* user() - используется для получения аутентифицированного пользователя;
* redirect() - переадресация;
* route() - используется для генерации URL для именованного маршрута (пример кода представлен в приложении Е.1);
* logout() - используется для завершения сессии аутентифицированного пользователя;
* input() - получения данных из input в blade шаблоне;
* where() - используется для создания SQL WHERE-условий в построителе запросов;
* first() - получение данных первого элемента в базе данных;
* create() - создание записи в базе данных (пример кода представлен в приложении Е.2);
* distinct() - используется, чтобы убедиться, что запрос базы данных возвращает только уникальные результаты;
* pluck() - используется для извлечения определенного значения из коллекции;
* compact() - используется для создания массива из переменных и их значений;
* find() - используется для поиска записи в базе данных по ее идентификатору;
* save() - сохранение изменений в базе данных;
* array\_filter() - используется для фильтрации элементов массива, используя функцию обратного вызова;
* all() - используется для получения всех элементов коллекции;
* update() - используется для обновления записей в базе данных;
* get() - используется для получения всех записей из таблицы базы данных;
* post() - используется для создания новой записи в базе данных;
* put() - используется для обновления записи в базе данных;
* groupBy() - используется для группировки элементов коллекции по определенному критерию;
* array\_filter() - используется для фильтрации элементов массива, используя callback-функцию;
* whereDoesntHave() - используется для фильтрации элементов коллекции, которые не имеют определенной связи;
* delete() - используется для удаления элементов из коллекции или массива.
* now() - используется для получения текущего времени;
* diffInDays() - используется для определения разницы в днях между двумя датами;
* round() - используется для округления числа;
* createFromFormat() - используется для создания экземпляра Carbon из строки, представляющей дату и время в определенном формате

JavaScript:

* + getElementById() - получения данных элемента из html шаблона по id;
  + addEventListener() - метод, который настраивает функцию, которая будет вызываться всякий раз, когда указанное событие доставляется цели;
  + stopPropagation() - предотвращает дальнейшее распространение текущего события в дереве DOM.

Были разработаны blade шаблоны, предназначенные для отрисовки элементов на странице, которые являются многоразовыми автономными блоками кода имея свой HTML, CSS и JavaScript код.

* + 1. Составные части программы и связи между ними

Разбивка файлов:

* weber папка сгенерирована Laravel, в ней находится весь проект (название создается на основе введенного вами);
* app - содержит множество важных директорий и файлов, таких как Http, где находятся контроллеры и промежуточное ПО;
* bootstrap - содержит директории и файлы фреймворка для css;
* config - содержит все конфигурационные файлы вашего приложения Laravel;
* database содержит все файлы миграции и сидирования базы данных;
* node\_modules — это место, куда помещаются ваши пакеты npm и двоичные файлы;
* public - является корневой директорией, которая служит входной точкой в ваше приложение Laravel;
* resources - содержит ваши представления, необработанные ассеты, а также локализованные сообщения;
* routes - содержит все файлы маршрутов вашего приложения Laravel;
* storage - содержит все файлы, которые ваше Laravel приложение генерирует в процессе работы;
* tests - содержит все тестовые случаи для вашего приложения Laravel;
* vendor - содержит все ваши composer зависимости;
* .env;
* gitignore;
* composer.json - представляет собой файл, который содержит список всех зависимостей проекта;
* composer.lock - содержит информацию о конкретных версиях всех установленных пакетов, которые были последними при последнем обновлении;
* package.json содержит ваши зависимости и использует собственные модули JavaScript с ключевыми словами и устаревшими файлами CommonJS;
* package-lock.json - отслеживает версию зависимостей npm;
* phpunit.xml - конфигурационный файл для PHPUnit, инструмента для тестирования в PHP;
* vite.config.js содержит конфигурацию Vite.

Программный продукт построен на клиент-серверной архитектуре.

Серверная и клиентская часть взаимодействует между собой благодаря протоколу передачи данных HTTPS. Но также у каждой части есть взаимодействие между своими компонентами.

Серверная часть обрабатывает бизнес-логику приложения, то есть программный код, находящийся на сервере. Код должен обеспечивать обработку запросов от клиента, обмен данными с базой данных, выполнение любых необходимых изменений или модификаций, а затем возврат результата клиенту.

Клиентская часть производит реализацию пользовательского интерфейса, а также формирует запросы к серверу обрабатывает полученные ответы от него и отображает их содержимое.

* 1. Описание работы программы

Сервис предоставляет обучающие уроки по веб-программированию.

* + 1. Функциональное назначение

Сервис предоставляет возможность пользователю:

* взаимодействие с верхней панелью;
* просматривать обучающие уроки;
* взаимодействие с курсами.
  + 1. Тестирование

После разработки приложения необходимо протестировать модули на корректную работу. Для этого напишем тест-кейсы. Тест-кейс — это форма записи проверки, которую проводит тестировщик. По сути, это алгоритм действий, по которому предполагается тестировать уже написанную программу. В нём подробно прописаны шаги, которые нужно сделать для подготовки к тесту, сама проверка и ожидаемый результат. Тест-кейс может выглядеть по-разному.

У него есть стандартные поля, но каждая компания оформляет его как ей удобно.

2.3.2.1 Тест-кейсы

Таблица 1 - Тест-кейс успешной отправки формы регистрации

|  |  |
| --- | --- |
| Наименование | Описание |
| Заголовок/название теста | Тест-кейс №1 |
| Краткое изложение теста | Успешная отправка формы регистрации |
| Этапы теста | 1. Запустить сайт 2. На главном экране открыть диалоговое окно при нажатии на кнопку «Регистрация» 3. Ввести данные в форму 4. Нажать на кнопку «Зарегистрироваться»   (рисунок Д1 – Д3, приложение Д) |
| Тестовые данные | Login: KiKO  Email: kir890987@gmail.com  Password: 1234567 |
| Ожидаемый результат | В базе данных в таблице users создастся новая запись со значениями полей «login», «email», «password» и переход на главную страницу с авторизованным пользователем |
| Фактический результат | Пользователь вошел в учетную запись |
| Предварительное условие | Наличие персонального компьютера и установленного браузера последней версии |
| Постусловие | Переход на главную страницу с авторизованным пользователем |
| Статус (зачет/незачет) | Зачет |

Таблица 2 - Тест-кейс успешной отправки формы авторизации

|  |  |
| --- | --- |
| Наименование | Описание |
| Заголовок/название теста | Тест-кейс №2 |
| Краткое изложение теста | Успешная отправка формы авторизации |
| Этапы теста | 1. Запустить сайт 2. На главном экране открыть диалоговое окно при нажатии на кнопку «Авторизация» |
|  | 1. Ввести данные в форму 2. Нажать на кнопку «Войти»   (рисунок Д4 – Д6, приложение Д) |
| Тестовые данные | Login: KiKO  Password: 1234567 |
| Ожидаемый результат | Из базы данных таблицы users будет взята информация из полей «login», «password», если пользователь существует, то происходит вход в учетную запись |
| Фактический результат | Пользователь вошел в учетную запись |
| Предварительное условие | Наличие персонального компьютера и установленного браузера последней версии |
| Постусловие | Переход на главную страницу с авторизованным пользователем |
| Статус (зачет/незачет) | Зачет |

Таблица 3 - Тест-кейс добавления курса из панели администратора

|  |  |
| --- | --- |
| Наименование | Описание |
| Заголовок/название теста | Тест-кейс №3 |
| Краткое изложение теста | Успешное добавление курса в бд с помощью панели администратора |
| Этапы теста | 1. Запустить сайт 2. Авторизоваться 3. Перейти на страницу панели администратора 4. Ввести данные в форму 5. Нажать на кнопку «Добавить курс» 6. Проверить, что курс добавлен в бд   (рисунок Д7 – Д9, приложение Д) |
| Тестовые данные | Название курса: HTML  Напрвление курса: frontend  Дата начала: 21.01.24  Дата окончания: 21.02.24 |
| Ожидаемый результат | Курс успешно добавлен в бд |
| Фактический результат | Курс успешно добавлен в бд |
| Предварительное условие | Наличие персонального компьютера и установленного браузера последней версии, иметь доступ к панели администратора |
| Постусловие | После нажатия на кнопку «Добавить курс» запись добавляется в бд |
| Статус (зачет/незачет) | Зачет |

Таблица 4 - Тест-кейс добавления урока из панели администратора

|  |  |
| --- | --- |
| Наименование | Описание |
| Заголовок/название теста | Тест-кейс №4 |
| Краткое изложение теста | Успешное добавление урока в бд с помощью панели администратора |
| Этапы теста | 1. Запустить сайт 2. Авторизоваться 3. Перейти на страницу панели администратора 4. Ввести данные в форму 5. Нажать на кнопку «Добавить урок» 6. Проверить, что выбранная запись добавлена в бд   (рисунок Д10 – Д12, приложение Д) |
| Тестовые данные | Название урока: Введение  Название курса: HTML |
| Ожидаемый результат | Урок добавлен в бд |
| Фактический результат | Урок добавлен в бд |
| Предварительное условие | Наличие персонального компьютера и установленного браузера последней версии, иметь доступ к панели администратора |
| Постусловие | После нажатия на кнопку «Добавить урок» запись добавлена в бд |
| Статус (зачет/незачет) | Зачет |

Таблица 5 - Тест-кейс выхода из учетной записи

|  |  |
| --- | --- |
| Заголовок/название теста | Тест-кейс №5 |
| Краткое изложение теста | Успешный выход из учетной записи |
| Этапы теста | 1. Запустить сайт 2. Авторизоваться 3. Нажать кнопку «Выйти» 4. Проверить, что пользователь вышел из учетной записи   (рисунок Д13 – Д15, приложение Д) |
| Тестовые данные | Наличие учетной записи |
| Ожидаемый результат | Пользователь вышел из учетной записи |
| Фактический результат | Пользователь вышел из учетной записи |
| Предварительное условие | Наличие персонального компьютера и установленного браузера последней версии, зарегистрироваться в системе |
| Постусловие | Перезагрузить страницу |
| Статус (зачет/незачет) | Зачет |

* + 1. Вызов и загрузка

Программный продукт требует установки composer, node.js и Open Server Panel (или другой локальный сервер). Для того чтобы запустить сайт нужно:

1. Запустить Open Server или другой локальный сервер;
2. Ввести следующие команды в терминале в директории weber: php artisan serve, npm run dev;
3. Перейти по пути http://localhost:8000 в адресной строке вашего браузера;
   * 1. Назначение программы

Обучающая платформа(веб-приложение) Weber предназначена для обучение пользователей веб-программированию.

* + - 1. Работа пользователя

Для того, чтобы начать работать с программным продуктом, необходимо запустить приложение, зайти с персонального компьютера в браузер, после этого ввести в адресную строку ссылку с доменным именем приложения. После чего пользователь может ознакомиться с продуктом, зарегистрироваться или авторизироваться и начать обучение.

Заключение

В результате курсового проектирования были изучены основы проектирования баз данных, работа со специализированными программами, были улучшены знания в области вёрстки и проектирования баз данных. Были также, улучшены навыки программирования.

На этапе формирования требований к системе были сформулированы цели и задачи курсового проекта.

В данной работе были решены все поставленные во введении задачи, начиная со знакомством с предметной областью и заканчивания созданием сайта для музыкального веб-приложения.

В ходе работы на этапе проектирования была разработана базы данных, средствами Draw.io. Также создана структура базы данных средствами Laravel и декларативного языка программирования MySQL, также были приобретены новые навыки работы с языками программирования такими как: JavaScript, а также получен новый опыт работы с фреймворками Laravel и приложением VSCode.

На этапе реализации был осуществлён ввод данных в базу данных и организован поиск необходимой информации по критерию и отображение результирующего списка данных.

В приложении к пояснительной записке курсового проекта помещены разработанные схемы базы данных, выполненные в соответствии с требованиями ГОСТ 7.32-2017.

Таким образом можно сделать вывод, что задание на курсовое проектирование выполнено успешно и в полном объеме.

Список литературы

1. ГОСТ 7.32-2017. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: Стандартинформ, 2017, дата обращения к источнику 21.02.2024;
2. ГОСТ Р 2.105-2019. Единая система конструкторской документации. Общие требования к текстовым документам. – М.: Стандартинформ, 2019, дата обращения к источнику 23.02.2024;
3. Глушаков С.В. Базы данных / С.В. Глушаков, Д.В. Ломотько. – М.: Харьков: Фолио, 2018. - 504 c, дата обращения к источнику 01.03.2023;
4. Документация Laravel [Электронный ресурс]

https://laravel.com/docs/11.x/documentation, дата обращения к источнику 05.03.2024;

1. Документация с Vite [Электронный ресурс]

https://vitejs.dev/guide/, дата обращения к источнику 05.03.2024;

1. Документация JavaScript [Электронный ресурс]

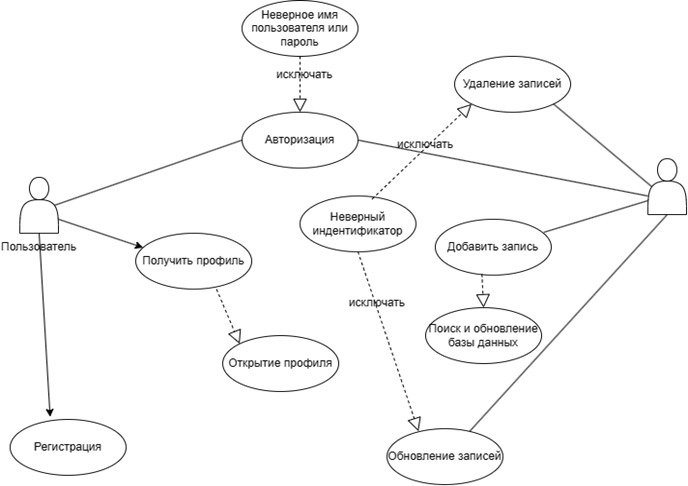
<https://developer.mozilla.org/ru/docs/Web/JavaScript>, дата обращения к источнику 09.03.2024;

1. Документация CSS [Электронный ресурс]

<https://developer.mozilla.org/ru/docs/Web/CSS/Reference>, дата обращения к источнику 05.03.2024;

# Приложение А

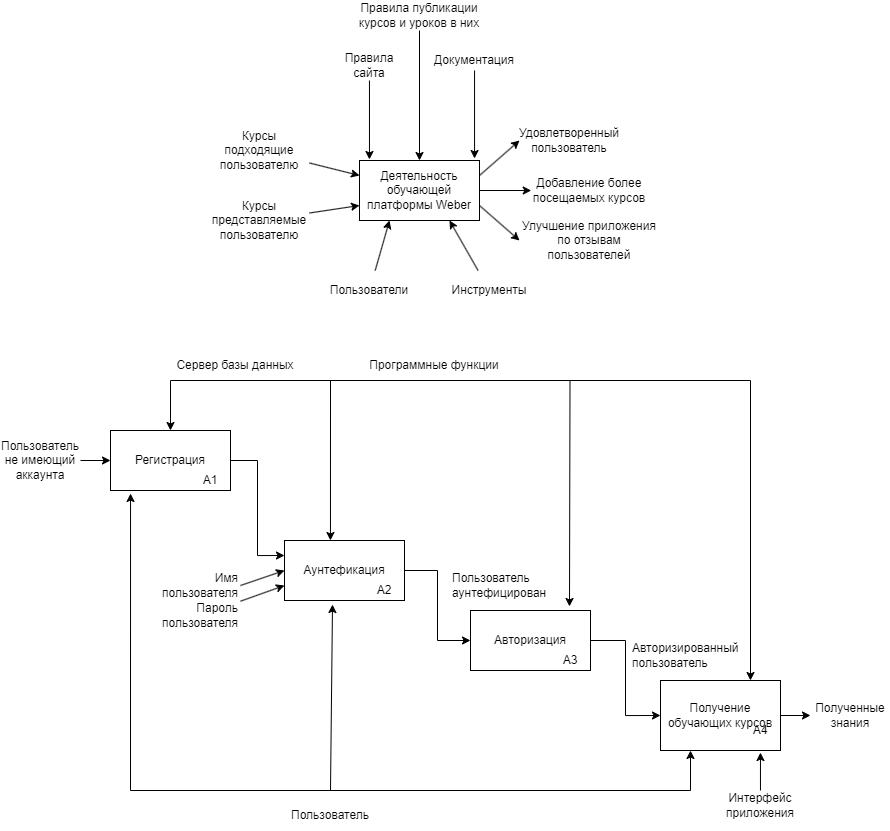
(обязательное)



А.1 – Схема типов доступа пользователей к сайту

# Приложение Б

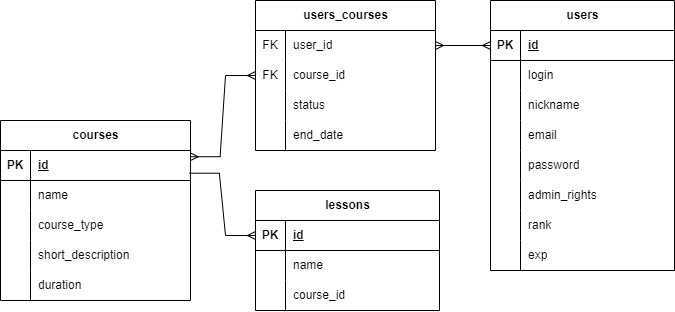
(обязательное)



Б.1 – Диаграмма бизнес – прецедентов

# Приложение В

(обязательное)



В.1 – IDEF1x модель

# Приложение Г

(обязательное)



Рисунок Г.1 - шапка пользователя не вошедшего в систему



Рисунок Г.2 – пример главной страницы

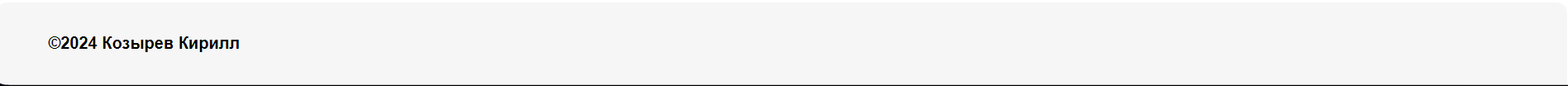


Рисунок Г.3 – подвал

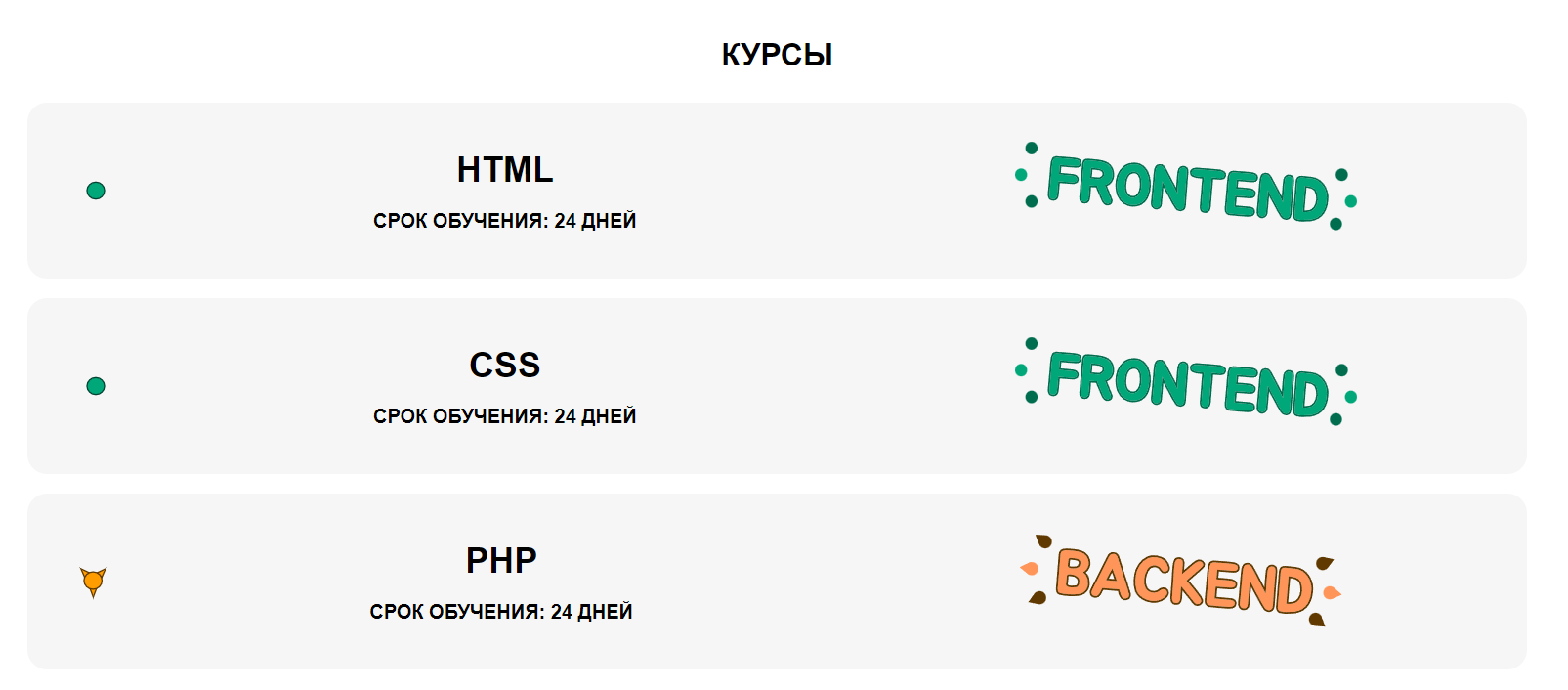


Рисунок Г.4 – каталог курсов

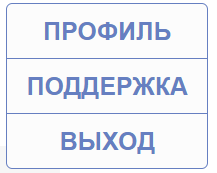


Рисунок Г.5 – выпадающая панель при клике на иконку

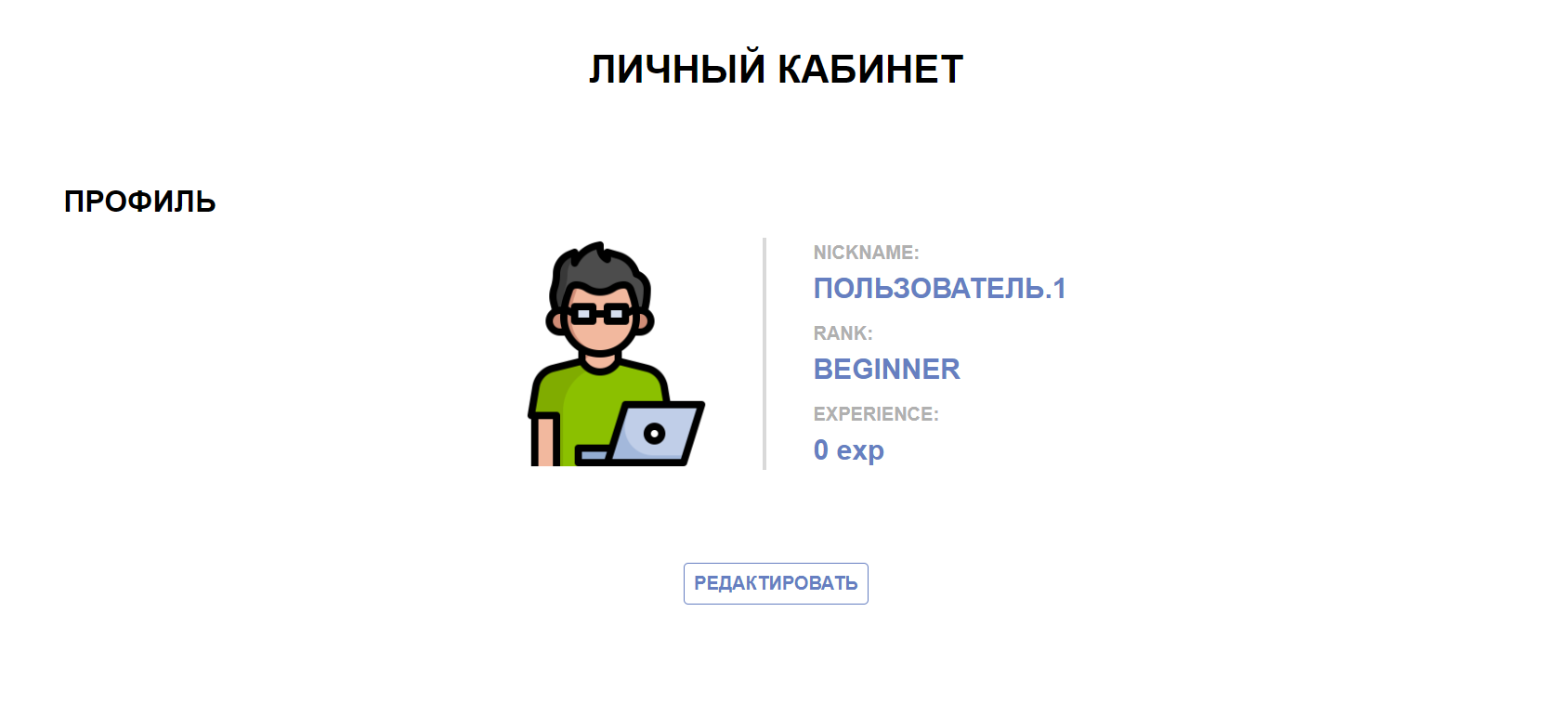


Рисунок Г.6 – пример страницы профиля

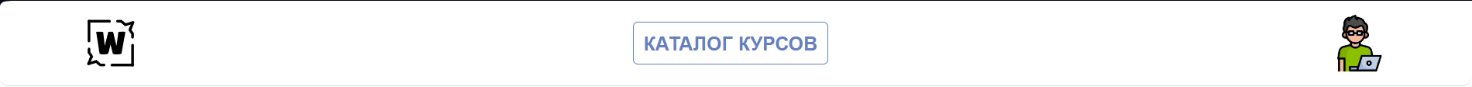


Рисунок Г.7 – шапка пользователя вошедшего в систему

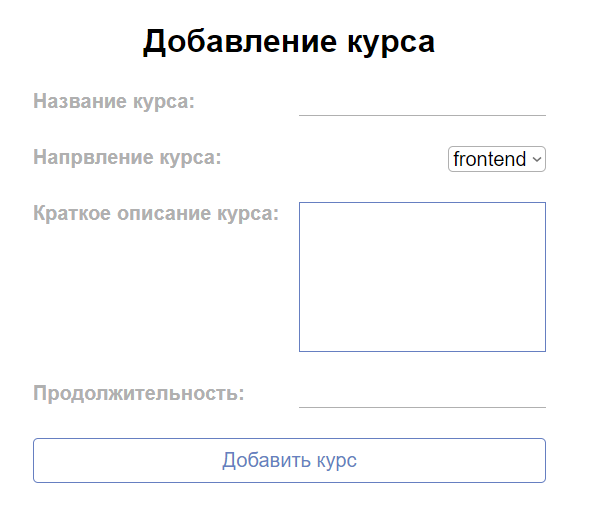


Рисунок Г.8 – добавления курсов на странице админа

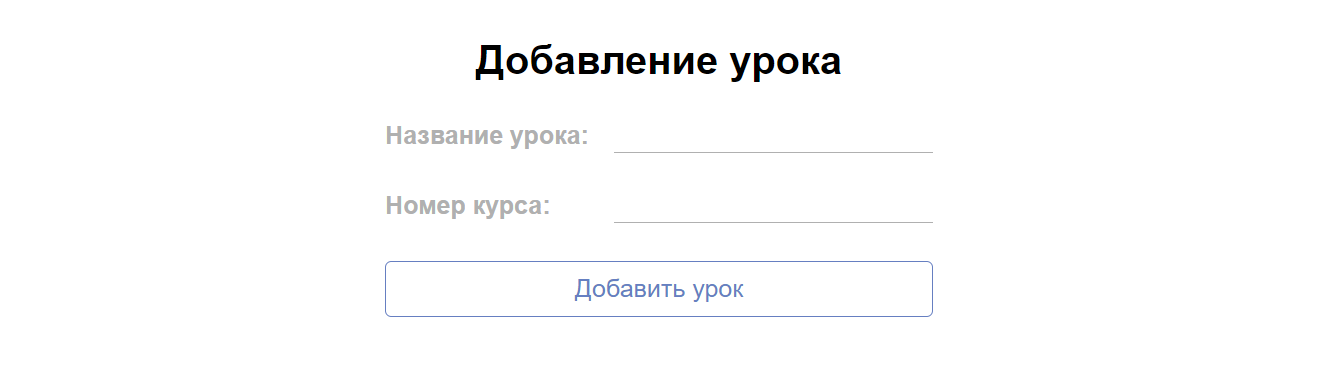


Рисунок Г.9 – добавления уроков на странице админа

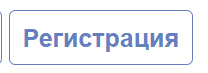
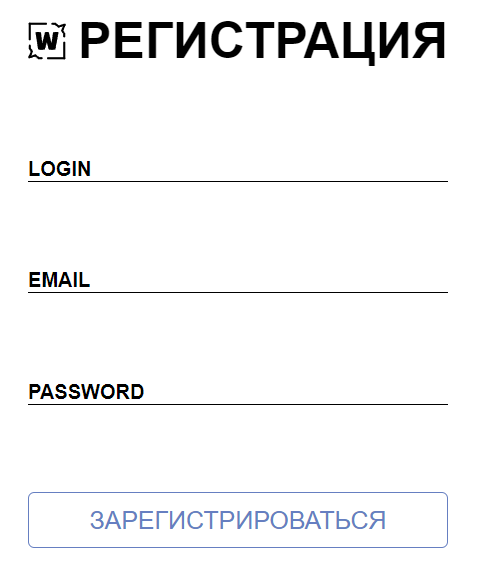


Рисунок Г.10 – кнопка вызова диалогового окна регистрации пользователя в системе

  
Рисунок Г.11 – диалоговое окно регистрации пользователя в системе

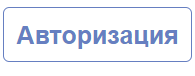


Рисунок Г.12 – кнопка вызова диалогового окна входа пользователя в систему

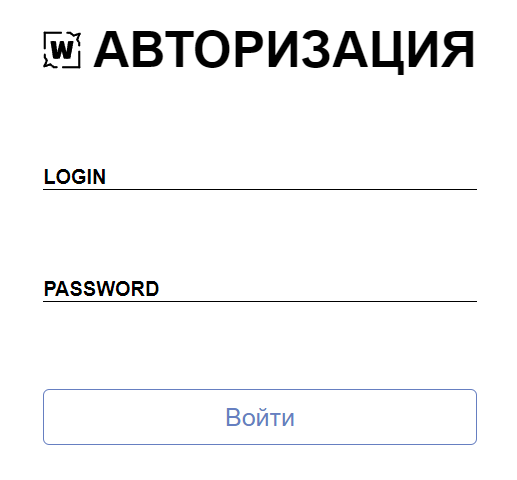


Рисунок Г.13 – диалоговое окно авторизации пользователя в системе

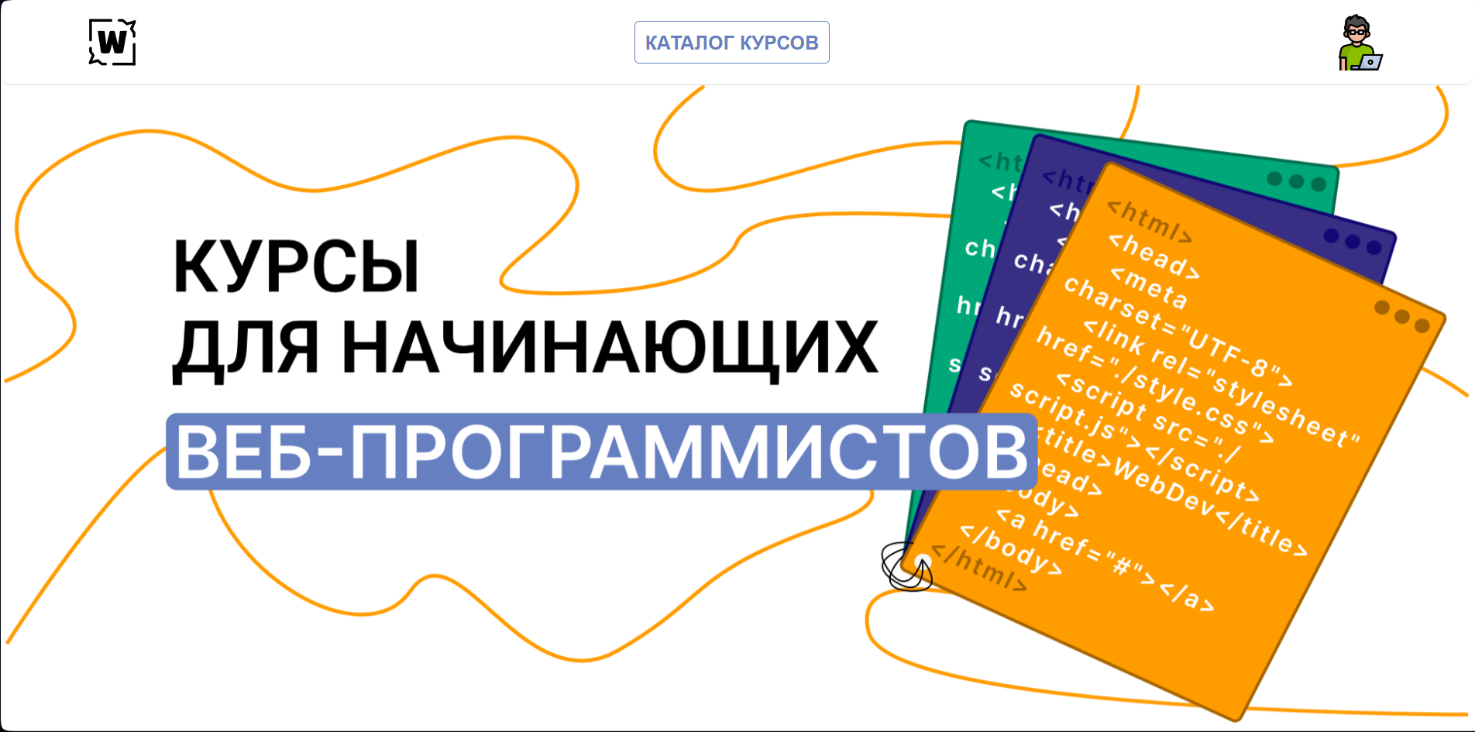


Рисунок Г.14 – страница после авторизации пользователя в системе



Рисунок Г.15 – страница после авторизации пользователя с правами админа в системе

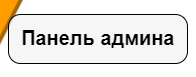


Рисунок Г.16 – кнопка перехода на страницу администратора

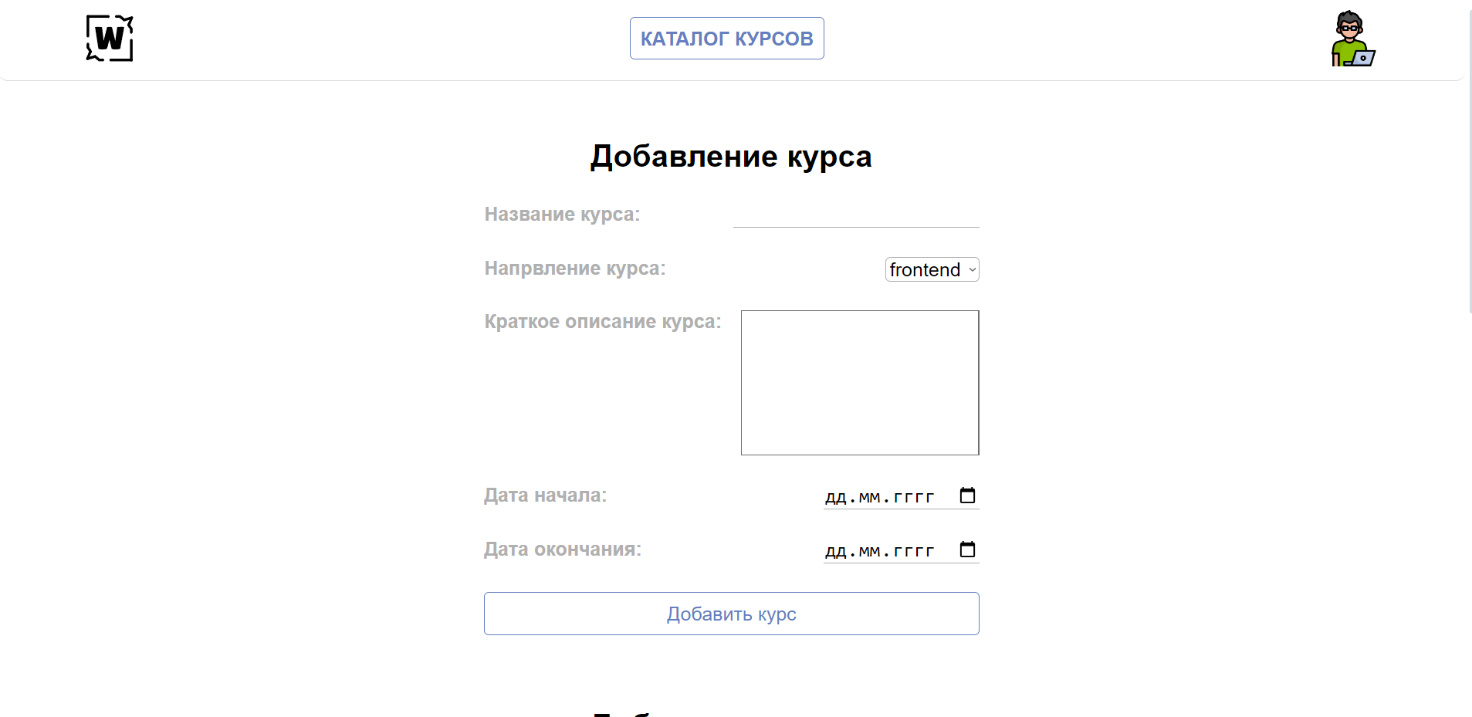


Рисунок Г.17 – страница администратора

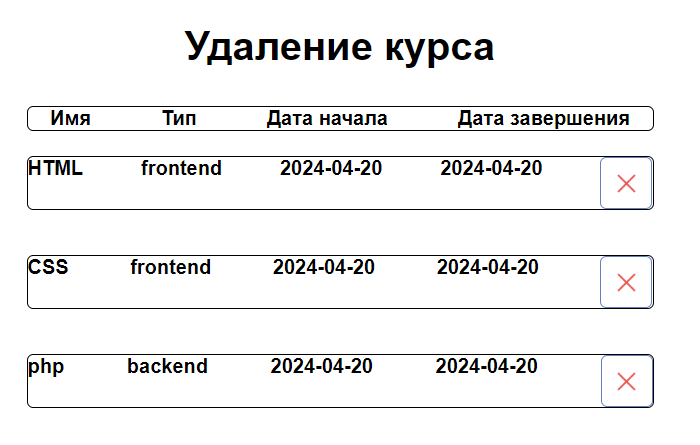


Рисунок Г.18 – форма удаления курса

# Приложение Д

(обязательное)

public function destroy\_lesson($lesson\_id)

{

$lesson = Lesson::find($lesson\_id);

$lesson->delete();

return redirect()->route('admin');

}

Д.1 – Пример кода с использованием функции route()

public function description ($id) {

$user\_id = auth()->user()->id;

$course = Course::find($id);

$course = User\_course::create(

[

'user\_id'=> $user\_id,

'course\_id'=> $course->id,

'status' => 'processing',

]

);

return redirect()->route('profile');

}

Д.2 – Пример кода с использованием функции create()