**Содержание**

[Введение 5](#_Toc168922660)

[1. Теоретическая часть. Анализ предметной области 7](#_Toc168922661)

[1.1 Обобщенная постановка задачи 7](#_Toc168922662)

[1.2 Постановка задачи 7](#_Toc168922663)

[1.3 Информация о предприятии – заказчике веб-приложения 8](#_Toc168922664)

[1.4 Сравнительный анализ существующих решений 10](#_Toc168922665)

[1.5 Расширенная постановка задачи 18](#_Toc168922666)

[1.6 Инструменты для разработки сайта 18](#_Toc168922667)

[1.7 Выбор стека 20](#_Toc168922668)

[1.7.1 Фронтенд и бэкенд 20](#_Toc168922669)

[1.7.2 Архитектура клиент-сервер 26](#_Toc168922670)

[1.7.3 Фреймворк 30](#_Toc168922671)

[1.7.4 Конструкторы и CMS 32](#_Toc168922672)

[2. Практическая часть. Выбор стредств реализации 34](#_Toc168922673)

[2.1 Программное обеспечение 34](#_Toc168922674)

[2.1.1 Adobe Photoshop 34](#_Toc168922675)

[2.1.2 Figma 35](#_Toc168922676)

[2.1.3 PhpStorm 36](#_Toc168922677)

[2.1.4 Инструменты разработчика Chrome 38](#_Toc168922678)

[2.1.5 Laragon 40](#_Toc168922679)

[2.2 Аналитическое моделирование 41](#_Toc168922680)

[2.3 Проектное моделирование 44](#_Toc168922681)

[2.4 Реализация программного продукта 44](#_Toc168922682)

[2.4.1 Структура web-сайта 44](#_Toc168922683)

[2.4.2 Дизайн web-сайта 46](#_Toc168922684)

[2.4.3 Реализация блоков и секций 49](#_Toc168922685)

[3. Экономическая часть 57](#_Toc168922686)

[3.1. Цель и задачи экономического раздела 57](#_Toc168922687)

[3.2. Исходные данные для расчетов 58](#_Toc168922688)

[3.3. Экономический эффект и эффективность 61](#_Toc168922689)

[ЗАКЛЮЧЕНИЕ 63](#_Toc168922690)

[СПИСОК ЛИТЕРАТУРЫ 64](#_Toc168922691)

[ПРИЛОЖЕНИЯ 67](#_Toc168922692)

**ВВЕДЕНИЕ**

С каждым днем информационные технологии все больше внедряются в нашу жизнь. Новые изобретения в данной сфере создают возможности для эффективной коммуникации из любой точки мира. Сегодня многие привычные сферы человеческой деятельности претерпевают изменения – сфера образования не является исключением.

**Актуальность** развития системы дополнительного образования в современных условиях обусловлена резким ускорением процессов производства, потребления и накопления информации. Современная система основного образования, основанная на государственных стандартах, зачастую не успевает адаптироваться к происходящим изменениям и не может в полном объеме подготовить студентов к будущей профессиональной деятельности.

В качестве высоко адаптивного инструмента, позволяющего участвовать в процессе образования неограниченному числу лиц, независимо от их местонахождения, активно используются и разрабатываются образовательные веб-порталы.

В настоящее время в большинстве колледжей Москвы (не исключением является и «Колледж современного управления») активно развита система дистанционного образования. Она представляет собой ресурс, позволяющий студентам получать задания от преподавателей, просматривать записи лекций, обращаться к методическим рекомендациям по написанию лабораторных, практических, курсовых работ, а также выпускной квалификационной работы. Подобные системы также содержат электронный аналог «зачётной книжки», где приводится информация об успеваемости каждого студента, а также электронный журнал, куда могут быть выставлены промежуточные оценки по дисциплинам, опубликованы темы лекций и домашние задания.

Объектом исследования в данном проекте является Профессиональное образовательное учреждение «Колледж современного управления» города Москвы.

Предметом исследования – организация обучения по программам дополнительного образования в Колледже.

Цель дипломного проекта: разработать веб-приложение для организации дополнительного образования студентов и выпускников «Колледжа современного управления» города Москвы.

Для достижения этой цели необходимо решить следующие задачи:

* провести анализ предметной области и теоретических основ разработки образовательных веб-порталов;
* проанализировать системы конкурентов в сфере дополнительного образования;
* произвести стоимостную и временную оценку проекта, составить план работ;
* осуществить выбор методологии разработки ПО;
* провести сбор и анализ требований к системе;
* выбрать архитектуру системы;
* описать основные сценарии использования;
* определить сущности системы и описать поля сущностей в БД;
* составить техническое задание;
* разработать первую версию системы.

В первой главе рассмотрены основные этапы проектирования веб-сайта: обобщенная постановка задачи, анализ предметной области, сравнительный анализ существующих решений, проведена расширенная постановка задачи.

Вторая глава работы содержит выбор технических решений, включающий в себя инструменты для разработки, выбор стека, программного обеспечения.

В третьей главе выполнено аналитическое моделирование, включающее в себя диаграмму вариантов и диаграмму деятельности. Выполнено проектное моделирование, описан процесс разработки веб-приложения.

# 

# 1. Теоретическая часть. Анализ предметной области

## 1.1 Обобщенная постановка задачи

Основная задача, которую необходимо решить в ходе выполнения данной работы, заключается в проектировании и реализации веб-приложения.

1. Определение информационных потребностей заказчика и формирование структуры веб-приложения.
2. Проектирование веб-приложения. На данном этапе необходимо разработать структуру сайта, определить состав и назначение, а также архитектуру сайта в целом.
3. Реализация программных модулей. Здесь необходимо создать программную реализацию веб-приложения, которая соответствовала бы принципам простоты, гибкости и прозрачности. Соблюдение данных принципов необходимо для обеспечения низкой стоимости ПО, облегчения последующей модернизации сайта и адаптации его под конкретное место внедрения с наименьшими затратами.

**1.2 Постановка задачи**

Разрабатываемое приложение должно быть представлено в виде веб-сайта. Несмотря на то, что со временем образовательные учреждения предпочитают разрабатывать приложения, в качестве платформы был выбран именно веб, потому что:

* Код для веб-платформы не нужно переписывать под разные операционные системы — единожды написанный, он будет работать на любом устройстве.
* Приложения, как правило, размещаются в так называемых магазинах приложений (Google Play, App Store, GetApps, RuStore), требуют модерации и могут быть удалены в любой момент. На веб-сайт можно зайти из любого браузера, и он не требует установки.

Данный сайт будет представлять собой образовательный портал «Колледжа современного управления» города Москвы. Портал будет включать в себя:

* Главную страницу.
* Личную страницу участника курсов.
* Каталог курсов.
* Административный интерфейс.
* Обратную связь.

Также на разрабатываемом сайте должен быть размещен блок с информацией о колледже, особенно полезный для тех, кто попал на портал впервые, не из числа студентов и сотрудников колледжа.

**1.3 Информация о предприятии – заказчике веб-приложения**

Профессиональное образовательное учреждение «Колледж Современного Управления» основано в 2002 году с целью предоставления студентам возможности получить качественное и востребованное образование в сфере экономики и менеджмента. Образовательное учреждение имеет бессрочную лицензию на право ведения образовательной деятельности и государственную аккредитацию.

В настоящее время Колледж – это современное учебное заведение среднего профессионального образования, обладающее достаточной учебно-материальной базой, хорошо подготовленным преподавательским составом и учебно-вспомогательным персоналом.

Адрес образовательной организации, ее представительств и филиалов (при наличии): 105318, г. Москва, ул. Ибрагимова, дом 31, корпус 1, этаж 2. Режим работы: с понедельника по пятницу, с 9.00 до 20.00.

Занятия проводятся: с понедельника по пятницу, с 9.00 до 19.10.

Колледж принимает на обучение учащихся на базе 9-ти и 11-ти классов по специальностям:

− 09.02.05 Прикладная информатика (по отраслям);

− 43.02.11 Гостиничный сервис;

− 38.02.01 Экономика и бухгалтерский учет (по отраслям);

− 38.02.03 Операционная деятельность в логистике;

− 38.02.04 Коммерция (по отраслям);

− 38.02.07 Банковское дело;

− 40.02.01 Право и организация социального обеспечения;

− 43.02.10 Туризм;

− 43.02.16 Туризм и гостеприимство.

В Колледже современного управления задействованы все доступные формы обучения:

- очная;

- заочная (группы выходного дня);

- заочная (дистанционная);

- очно-заочная.

Управление Колледжем осуществляется в соответствии с нормативными правовыми актами Российской Федерации и города Москвы, Уставом Колледжа.

Единоличным исполнительным органом является директор: Чаев Иван Юрьевич.

Коллегиальными органами управления являются:

- Общее собрание работников и обучающихся.

- Попечительский совет.

- Педагогический совет.

В целях учета мнения обучающихся, родителей (законных представителей) несовершеннолетних обучающихся по вопросам управления в Колледже созданы:

- Cовет обучающихся.

- Совет родителей.

Основные документы образовательного учреждения:

I . Правоустанавливающие документы

- Устав образовательной организации.

- Выписка из реестра лицензий.

- Свидетельство о государственной аккредитации (с приложениями).

- Выписка из реестра аккредитаций.

II . Локальные акты

- Приемная комиссия.

- Общие локальные акты.

- Локальные акты, регламентирующие образовательный процесс.

**1.4 Сравнительный анализ существующих решений**

Для дальнейшего проектирования, и в следствие, реализации нашего web-приложения, стоит рассмотреть и сравнить уже существующие сайты дополнительного образования, выявить их преимущества и недостатки.

С целью объективного сравнения, в данном анализе не присутствуют государственные сайты дополнительного образования, а рассматриваются лишь частные порталы, созданные по инициативе частных лиц и образовательных организаций.

По запросу «дополнительное образование для студентов колледжей» в поисковой системе «Яндекс» предлагаются сайты следующих образовательных организаций, приведенных на рисунках 1- 6. Рассмотрим их подробнее:

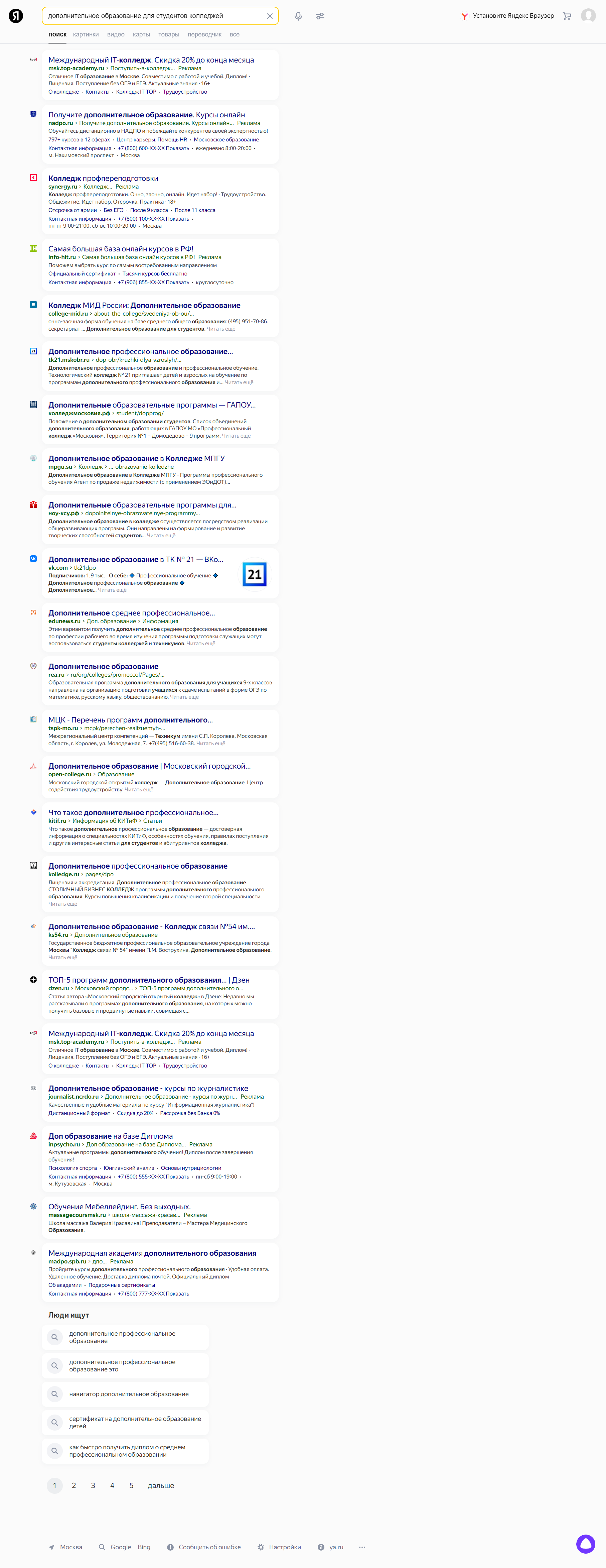


Рисунок 1 – Запрос «дополнительное образование для студентов колледжей» в поисковой системе «Яндекс»

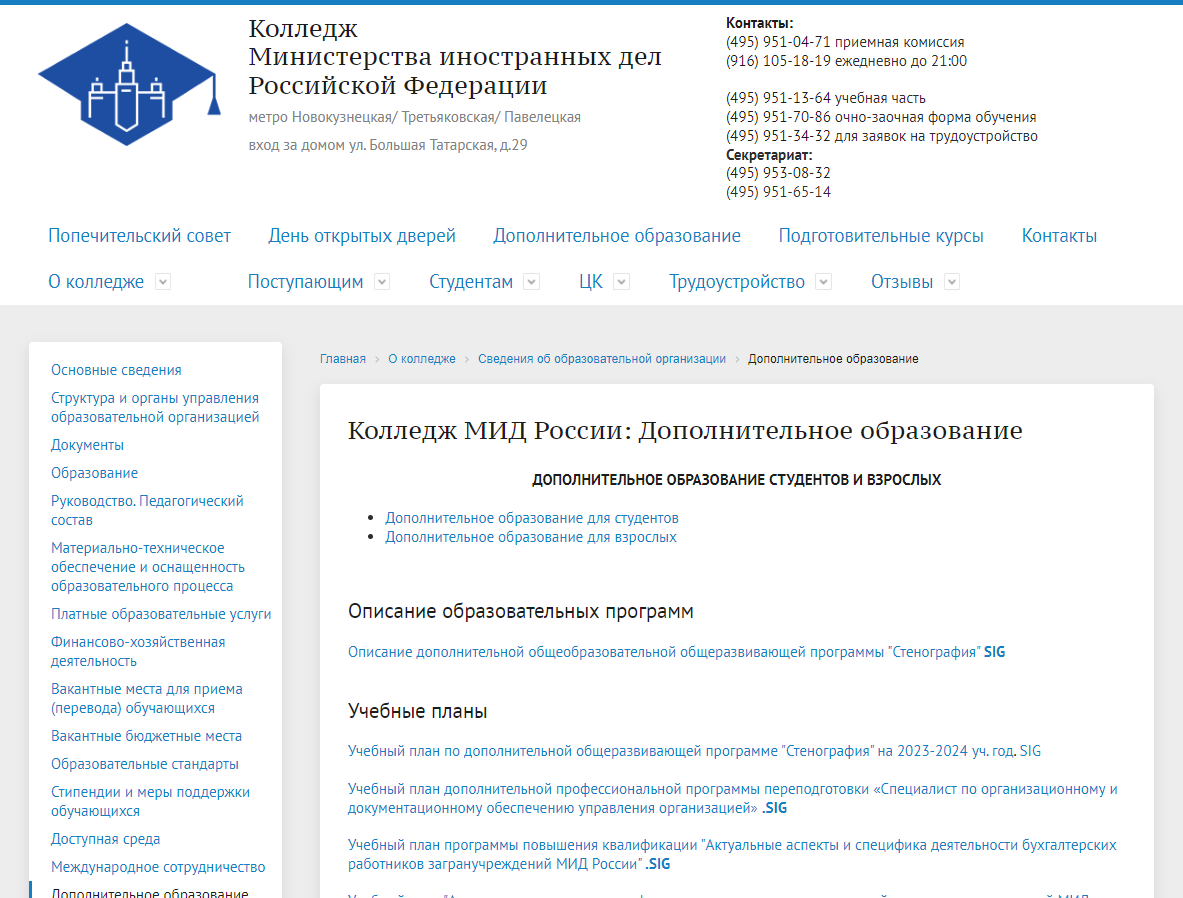


Рисунок 2 – Информация о дополнительном образовании Колледжа МИД России

Колледж Министерства иностранных дел Российской Федерации реализует дополнительные образовательные программы «Стеганография», «Специалист по организационному и документационному обеспечению управления организацией», «Деловая переписка», «Документирование управленческой деятельности в соответствии с требованиями ГОСТ Р 7.0.97-2016». Форма обучения – по большинству учебных программ очная. Портал для обучения отсутствует.

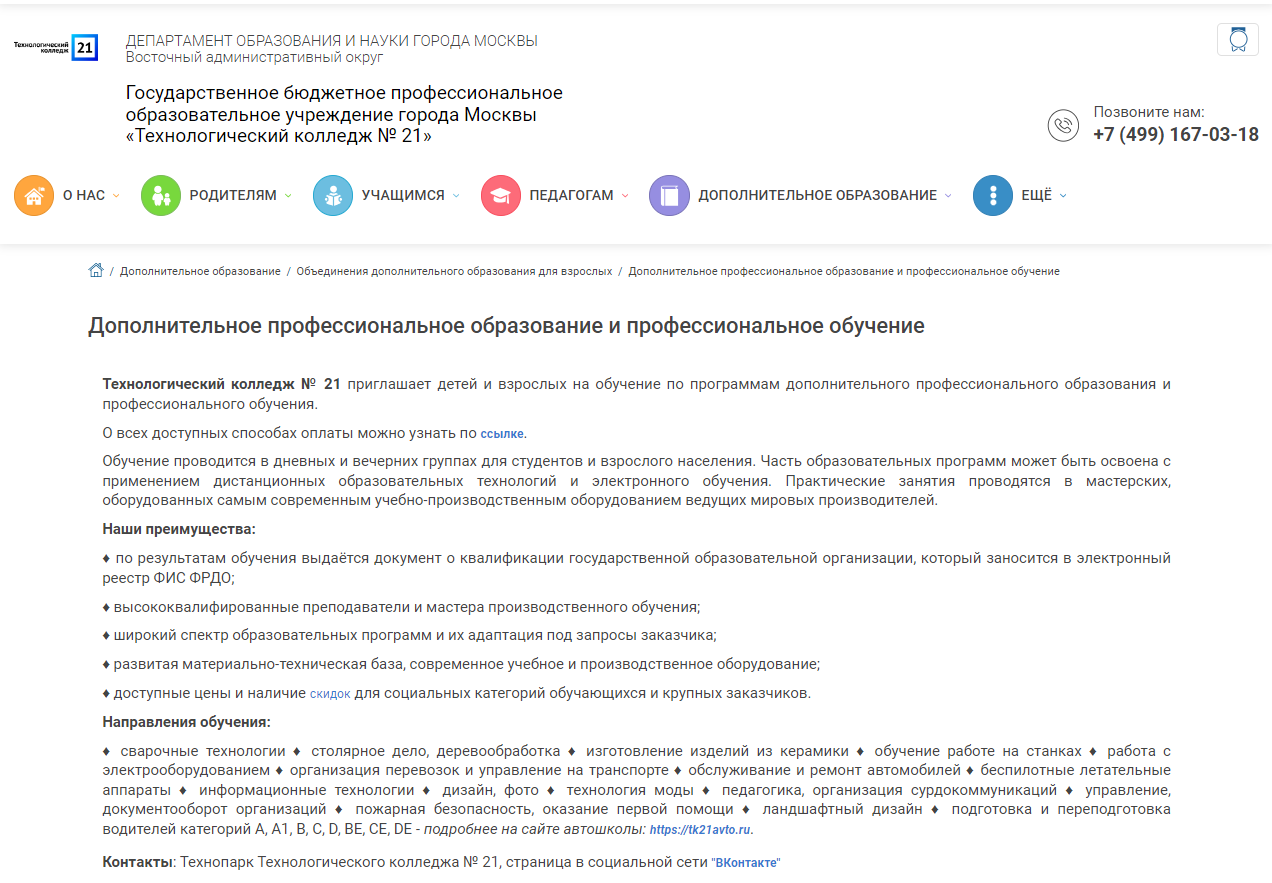


Рисунок 3 – Информация о дополнительном профессиональном образовании ГБПОУ «Технологический колледж №21» города Москвы

Государственное бюджетное профессиональное образовательное учреждение города Москвы «Технологический колледж № 21» реализует дополнительные образовательные программы по направлениям:

1. Сварочные технологии;
2. Столярное дело, деревообработка;
3. Изготовление изделий из керамики;
4. Обучение работе на станках;
5. Работа с электрооборудованием;
6. Организация перевозок и управление на транспорте;
7. Обслуживание и ремонт автомобилей;
8. Беспилотные летательные аппараты;
9. Информационные технологии;
10. Дизайн, фото;
11. Технология моды;
12. Педагогика, организация сурдокоммуникаций;
13. Управление, документооборот организаций;
14. Пожарная безопасность, оказание первой помощи;
15. Ландшафтный дизайн;
16. Подготовка и переподготовка водителей категорий А, А1, В, С, D, ВЕ, СЕ, DE.

Обучение проводится в дневных и вечерних группах для студентов и взрослого населения. Часть образовательных программ может быть освоена с применением дистанционных образовательных технологий и электронного обучения. Практические занятия проводятся в мастерских, оборудованных самым современным учебно-производственным оборудованием ведущих мировых производителей.

Форма обучения – по большинству учебных программ очная, что в первую очередь очевидно для профессий, не предусматривающих дистанционную работу. По направлению «Информационные технологии» предусмотрена очно-заочная форма, однако ресурс для обучения обнаружить не удалось.

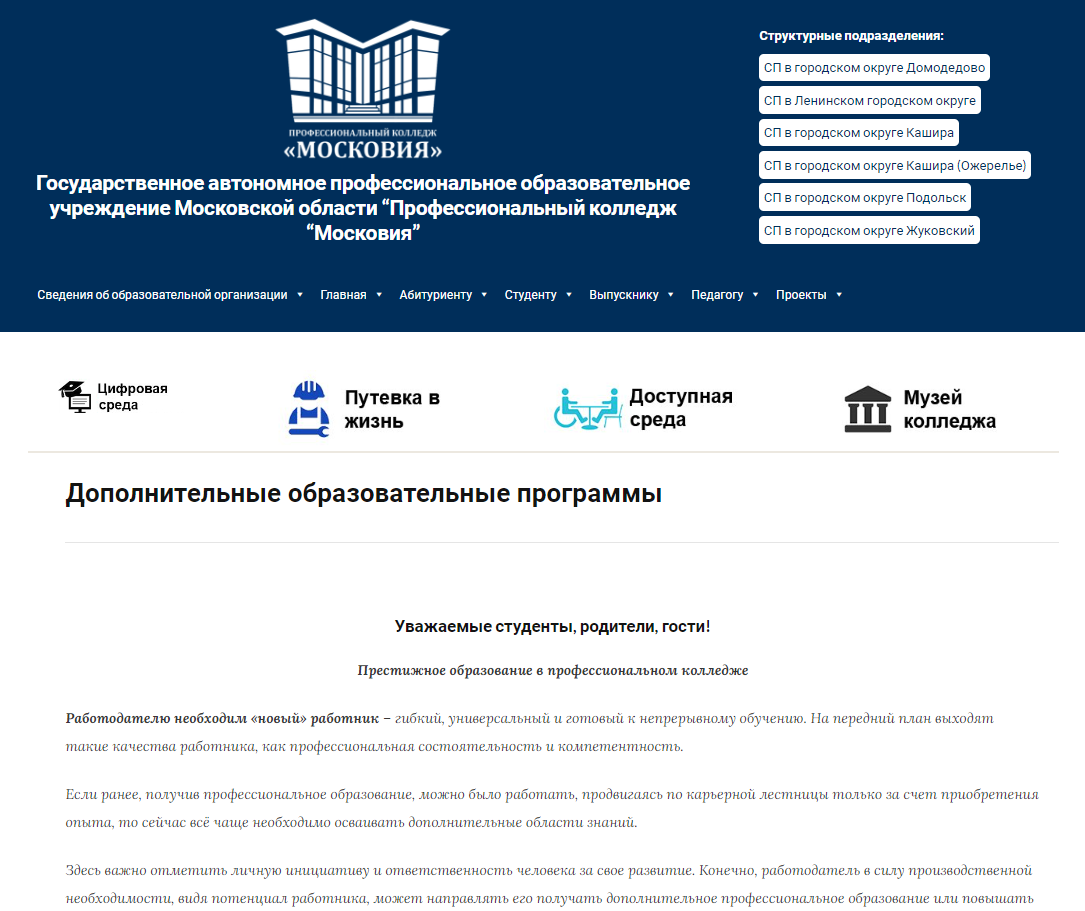


Рисунок 4 – Информация о дополнительных образовательных программах в ГАПОУ «Профессиональный колледж «Московия»

Государственное автономное профессиональное образовательное учреждение Московской области “Профессиональный колледж “Московия” реализует дополнительные образовательные программы в области патриотического воспитания, спортивные секции (волейбол, футбол, ОФП, армейский рукопашный бой), технические кружки (авиационная безопасность, компьютерный дизайн, основы программирования, управление локомотивом на ЖД-транспорте), кружки художественного творчества и самодеятельности.

Форма обучения – очная, посещение вышеперечисленных кружков и секций бесплатное, образовательный портал для обучения отсутствует.

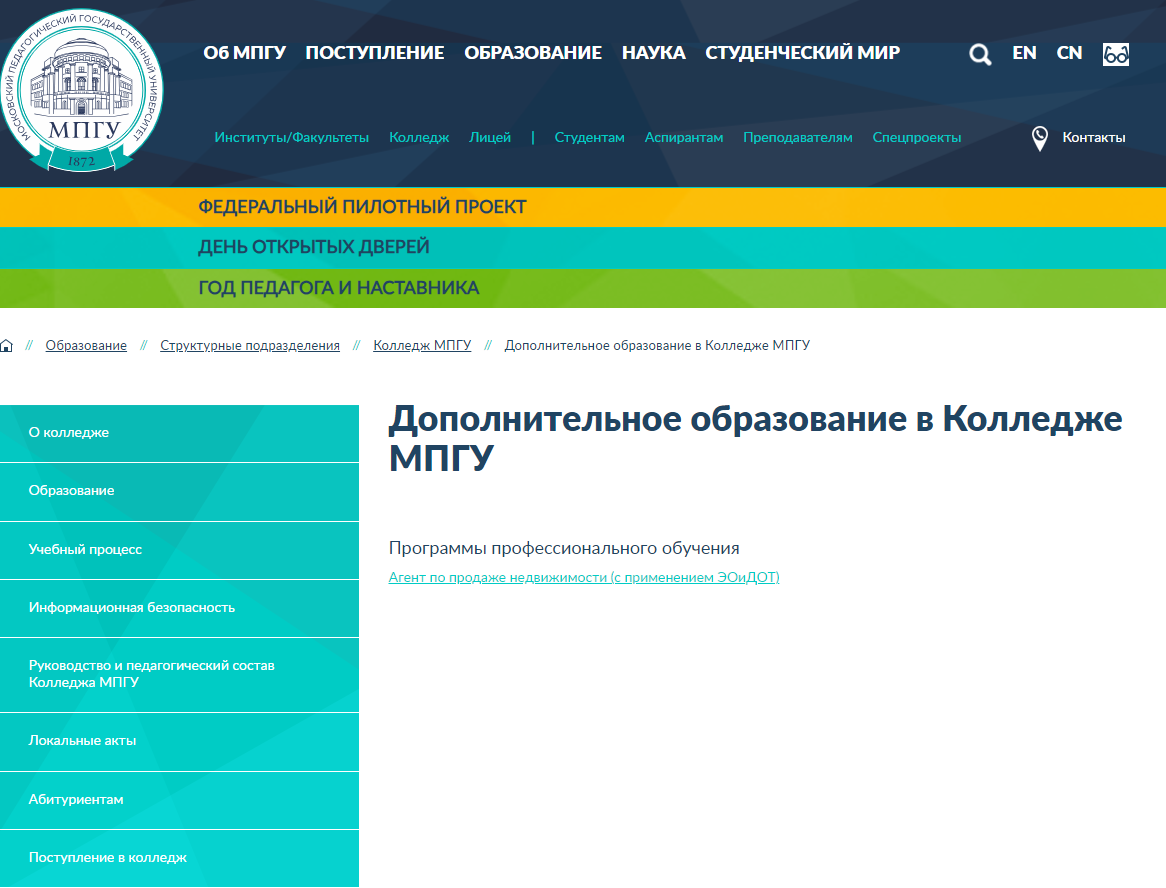


Рисунок 5 – Информация о дополнительном образовании колледжа Московского педагогического государственного университета

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский педагогический государственный университет» реализует только одну программу дополнительного образования – Агент по продаже недвижимости, включающая применение электронного обучения и обучения с использованием дистанционных технологий.

Форма обучения – очная (по информации сайта), срок обучения 2 месяца. Несмотря на упоминание ЭОиДОТ, портал для обучения также не был обнаружен.

Из проведенного анализа вышеперечисленных существующих web-сайтов колледжей, приводимых по запросу в поисковой системе «Яндекс», хотелось бы отметить Технологический колледж №21 как пример наличия разнообразных курсов дополнительного образования для всех категорий слушателей, однако поиски специального ресурса по организации дополнительного образования не увенчались успехом – в настоящее время работа системы дополнительного образования координируется исключительно в группе ВКонтакте (vk.com/tk21dpo) и при обращении лично или по электронной почте образовательной организации.

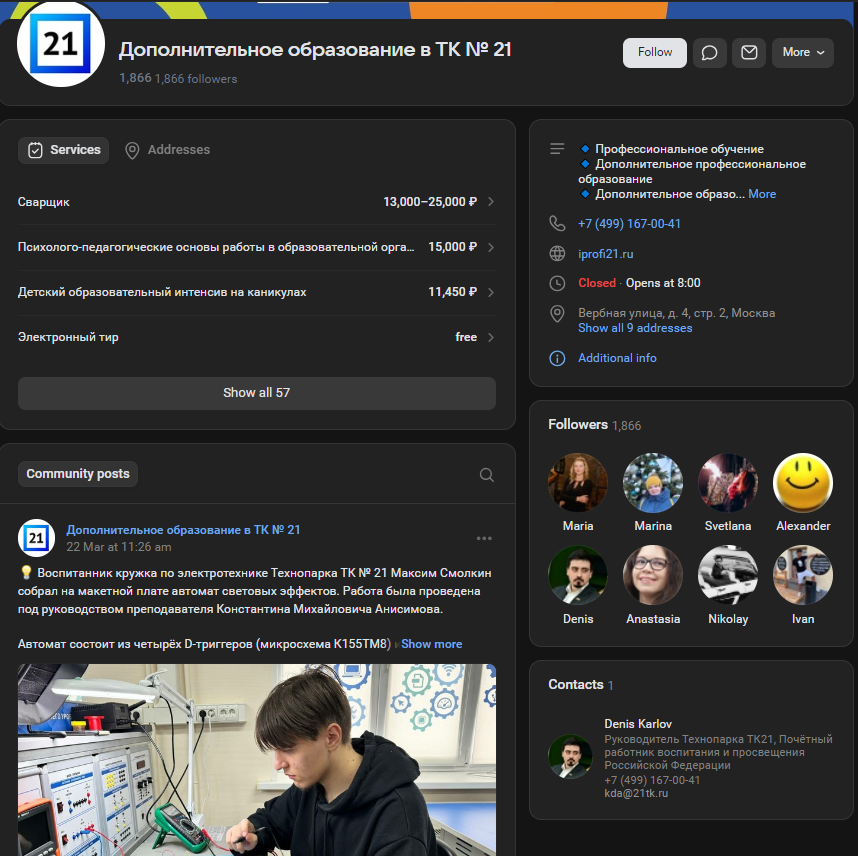


Рисунок 6 – Сообщество дополнительного образования ГБПОУ «Технологический колледж №21» города Москвы в социальной сети «ВКонтакте»

**1.5 Расширенная постановка задачи**

Конкуренция в Интернете постоянно усиливается. Это приводит к необходимости владельцев коммерческих сайтов разрабатывать наиболее эффективные ресурсы, способные увеличивать конверсию (то есть превращение посетителей в покупателей/заказчиков) и будут привлекательны для новых пользователей. Поэтому для разработки web-приложений используются передовые информационные технологии и программные продукты.

Из требований к проектированию ресурса было отмечено соблюдение дизайна в соответствии с брендбуком Колледжа современного управления, размещенным на официальном сайте с соблюдением цветовых сочетаний, начертаний шрифта и использованием официального логотипа колледжа.

Также из требований было необходимо обеспечить удобство просмотра посетителями web-сайта на различных устройствах, данный web-сайт должен быть адаптивен под различные устройства. Под смартфоны с минимальным разрешением 480×800, под планшеты или электронные книги с минимальным разрешением 768×1024, под ноутбуки и под стандартные десктопные мониторы.

Функционал обратной связи представляется в виде формы для ввода электронного адреса пользователя и кнопки «отправить» на всех блоках портала.

**1.6 Инструменты для разработки сайта**

Любое веб-приложение можно написать полностью с нуля, в целом, не прибегая к использованию иных инструментов. Однако такой подход, как правило, затратный по времени и ресурсам. Мы рассмотрим несколько типов инструментов для упрощения разработки:

1. Фреймворки

Фреймворк (от англ. Framework) – это инструмент для создания «каркаса»

сайта, обеспечения базовых возможностей для ускорения разработки.

При установке CMS вы сразу получаете готовый сайт и панель администратора для управления им. При установке фреймворка необходимо реализовывать всё самостоятельно. Обычно веб-фреймворки, вне зависимости от языка, на котором они написаны: предоставляют примерно одинаковый набор возможностей:

* Работа с запросом (то, что приходит со стороны клиента) и ответом (то, что уходит клиенту обратно). Это включает в себя установку HTTP-заголовков, работу с Cookies, обработку данных и файлов из форм.
* Роутинг — Показ пользователю нужной страницы на основе URL-адреса, который он ввёл в адресной строке.
* Работа с базами данных.
* Авторизация и регистрация пользователей, разграничение прав.
* Шаблоны для вывода HTML.
* Защита от самых распространенных атак.

1. CMS

CMS (англ. Content management system, система управления контентом) — это ПО, созданное для организации процесса создания, редактирования и управления содержимым веб-страниц. В отличие от конструкторов, для работы CMS необходима покупка и настройка веб-хостинга (хотя некоторые системы умеют автоматизировать этот процесс). Существуют как нишевые CMS (например, Wordpress для блогов или Opencart для интернет-магазинов), так и системы общего назначения (например, MODx). Большая часть CMS доступна бесплатно и имеет открытый код, что позволяет дорабатывать их.

Чаще всего CMS (как и конструкторы) используются для сайтов, где нет нестандартного функционала, например для создания блога или сайта-хранилища тематических статей.

**1.7 Выбор стека**

Стек (англ. stack – «множество») в разработке – это набор используемых технологий и языков программирования, которые будут применяться при создании веб-сайта или веб-приложения. От правильно выбранного стека зависит работоспособность сайта — то, насколько быстро он будет работать, сколько пользователей сможет выдержать одновременно.

В качестве набора инструментов разрабатываемого приложения могут быть использованы языки программирования PHP, HTML, JavaScript, язык SQL для формирования запросов к базе данных, язык оформления CSS, известный также как язык таблицы стилей.

**1.7.1 Фронтенд и бэкенд**

Веб-разработка подразделяется на 2 основных области: фронтенд и бэкенд.

* 1. Фронтенд

Фронтенд (от англ. frontend) — это всё, что связано с клиентской стороной сайта: вёрстка, дизайн и пользовательский интерфейс.

Отличие фронтенд-разработки в вебе от фронтенд-разработки в других сферах заключается в том, что для разработки можно использовать только один набор языков программирования: HTML, CSS и JavaScript.

HTML (HyperText Markup Language)– это язык разметки веб-сайтов, созданный ученым Тимом Бернерсом-Ли, создателем «всемирной паутины». Первые веб-сайты представляли из себя обычные HTML-документы. По своей структуре HTML представляет из себя набор вложенных друг в друга тегов:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Заголовок</title>

</head>

<body>

<h1>Привет, мир!</h1>

</body>

</html>

Теги, находящиеся внутри тега <head> отвечают за параметры страницы (заголовок окна, иконка, кодировка, ссылки на файлы стилей, в некоторых случаях - ключевые слова и краткое описание страницы). Те теги, что располагаются внутри тега <body> - выводятся на экран. Это могут быть тексты, картинки, таблицы, видео- и аудиоплееры и другие элементы разметки. Также чаще всего перед закрывающим тегом </body> подключаются ссылки на JavaScript-файлы (это связано с тем, что браузеры подключают сторонние файлы последовательно, и желательно подключать скрипты уже после загрузки своей страницы).

Текущая версия стандарта имеет номер 5. Он поддерживается всеми современными браузерами, вышедшими за последние несколько лет.

Существуют также так называемые «HTML-препроцессоры», позволяющие ускорять написание HTML за счёт различных сокращений, но в конечном итоге перед попаданием к пользователю все они компилируются в HTML-код.

CSS (Cascading StyleSheets)– это язык описания внешнего вида HTML-документов, созданный для того, чтобы упростить стилизацию веб-страниц и отделить логическое описание структуры документов от их внешнего вида.

CSS подключается к HTML-странице либо напрямую (прописывается внутри тега <style>), либо с помощью указания ссылки на местоположение файла с кодом с помощью тега (<link>).

CSS-код состоит из последовательного списка так называемых «селекторов» (это может быть тип HTML-элемента, его класс или идентификатор) и набора пар «свойство-значение». Так, например, следующий код:

.highlight {

color: red;

background-color: #fff;

font-weight: 100;

}

окрасит все элементы, имеющие тег highlight красным цветом, установит им белый фон и самое тонкое начертание шрифта.

Помимо этого, некоторые блоки могут быть «обернуты» в условия (также называемые медиа-запросами), например, для того, чтобы страница выглядела по-разному на компьютере и мобильных устройствах.

Текущая версия стандарта – CSS3, с 2011 года разрабатывается стандарт CSS4. Однако из-за того, что разные браузеры используют разные движки для отображения CSS, не везде стандарты поддерживаются одинаково хорошо. Это приводит либо к отказу от использования самых современных возможностей CSS, либо к необходимости написания отдельных стилей для устаревших браузеров.

Существуют CSS-фреймворки — готовые наборы стилей, которые можно подключить и свободно использовать на своих веб-страницах. Чаще всего они состоят из сброса стилей (чтобы элементы выглядели одинаково во всех браузерах), сетки колонок и готовых компонентов (например, кнопок или списков). Самым известным и популярным из таких фреймворков является Bootstrap.

Однако у использования фреймворков есть и минусы: много лишнего неиспользуемого кода, который может увеличить время загрузки страницы.

JavaScript (JS) – это интерпретируемый язык программирования, поддерживаемый браузерами (интерпретируемость языка означает то, что код на нём не нужно компилировать в машинный код).

Несмотря на своё название, не имеет отношения к Java. Изначально был создан программистом Бренданом Эйхом для браузера Netscape Navigator и использовался для придания интерактивности веб-страницам.

С годами функционал JS рос и сейчас он используется во многих сферах, но основной областью применения все еще остается веб — как на фронтенде, так и на бэкенде (с помощью серверной реализации JavaScript – NodeJS).

Самое простое применение JS – примитивные манипуляции с HTML с помощью модели DOM и событий элементов. Например, следующие строчки:

var button = document.getElementById("button");

button.addEventListener("click", function() {

document.getElementById("text").innerHTML = (new Date()).getTime();

});

добавят событие-обработчик на щелчок мыши по элементу с ID “button”, по которому в элемент с ID “text” будет записываться текущее время в формате UNIX timestamp.

JavaScript построен на базе стандарта ECMAScript (помимо него, на базе этого же стандарта создан язык ActionScript). Этот стандарт дополняется каждый год, но для того, чтобы JS-код мог поддерживаться не только самыми современными браузерами, применяются инструменты для транспиляции кода к более старым стандартам (чаще всего к ECMAScript 5, потому что он имеет максимально широкую поддержку).

В последние годы появилась возможность писать код фронтенда на других языках, например TypeScript и Dart, но все они в конечном итоге компилируются в понимаемый браузерами JS.

1. Бэкенд

Бэкенд (от англ. backend) – это серверная сторона разработки. Если код фронтенда доступен всем, кто заходит на сайт, то код бэкенда располагается на серверах и скрыт от пользователей. Фронтенд и бэкенд взаимодействуют друг с другом: бэкенд предоставляет API (программный интерфейс приложения), в свою очередь фронтенд делает запросы к нему для получения и отправки данных.

В отличие от фронтенда, для бэкенда не существует жестких ограничений по выбору используемых языков. По сути, если язык программирования поддерживает прием и отправку данных по сети, на нём уже можно написать бэкенд.

Самыми популярными языками для бэкенд-разработки являются PHP, Python, Java, Ruby, Node.JS. На разработанном web-приложении в рамках выпускной квалификационной работы используется язык PHP.

PHP (расшифровывается как PHP: Hypertext Preprocessor) – это интерпретируемый язык программирования. Изначально представлял из себя небольшой набор скриптов для обработки форм на сайтах-личных страницах, но впоследствии развился до полноценного языка.

PHP ориентирован на веб-разработку (например, «из коробки» доступны функции для работы с сессиями, cookies, формами) и редко применяется за её пределами. Является самым популярным языком для бэкендов веб-сайтов, почти все хостинг-провайдеры (кроме специализирующихся на других языках) поддерживают PHP. Используется такими крупными проектами, как, например, Facebook и Википедия.

Одной из главных особенностей языка является простота в освоении. Можно даже вставлять PHP-скрипты внутрь HTML-кода:

|  |  |
| --- | --- |
| Код на сервере: | То, что увидит пользователь: |
| <body>  <?php echo "Привет, мир"; ?>  </body> | <body>  Привет, мир!  </body> |

Однако PHP часто подвергается критике по следующим причинам:

* из-за простоты разработки появилось множество PHP-разработчиков, пишущих низкокачественный и не структурированный код;
* по скорости исполнения PHP уступает многим другим языкам;
* при разработке языка у создателей не было четкой философии.

В последние годы репутация PHP всё-таки начала улучшаться: с выходом 7 версии языка его скорость работы существенно увеличилась за счет использования кэша операция (Opcache) а для PHP-программистов появились стандарты (PSR).

В качестве альтернативы вкратце рассмотрим и другие языки программирования.

Python. В русском языке распространено как "питон". Высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Наиболее популярные фреймворки у Python – Flask и Django. Flask - легковесный и гибкий фреймворк, в нём нет жёсткой структуры. Можно поселить модели, контроллеры и инициализацию в одном файле, можно - в разных. Он пуст чуть более, чем полностью (даже ORM надо отдельно устанавливать). Это позволяет выбирать модули под конкретные задачи и устанавливать их по мере необходимости. Django - всё-в-одном фреймворк для стандартного набора задач, вроде админки, регистрации, базы данных, имейлов и т.п. Состоит из проекта, который делится на приложения. Проект - корневая папка и глобальные настройки. Приложения - это функционал сайта, разбитый по разным модулям. У каждого проекта они свои.

Ruby. В русском языке распространено как "руби". Динамический, рефлективный, интерпретируемый высокоуровневый язык программирования для быстрого и удобного объектно-ориентированного программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, сборщиком мусора и многими другими возможностями.

ASP. Разработчиком данного языка является Microsoft. Технология позволяет разрабатывать приложения для WWW. ASP легко и быстро.

Платформы для работы ASP: Windows NT и IIS (Internet Information Server). Не совсем корректно называть ASP языком, скорее, это именно технология для подключения программы к Web-страницам. Простой скриптовый язык и возможность использования внешних COM-компонентов - вот и весь секрет успеха ASP.

SQL – это информационно-логический язык, предназначенный для описания, изменения и извлечения данных, хранимых в реляционных базах данных.

DOM – это объектная модель документа (от англ. Document Object Model). Согласно DOM, документ (например, веб-страница) может быть представлен в виде дерева объектов, обладающих рядом свойств, которые позволяют производить с ним различные манипуляции, такие как: получение, изменение и удаление узлов и изменение связей между узлами.

**1.7.2 Архитектура клиент-сервер**

Веб-приложение – это клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – веб-сервер (в широком смысле).

Основная часть приложения, как правило, находится на стороне веб-сервера, который обрабатывает полученные запросы в соответствии с бизнес-логикой продукта и формирует ответ, отправляемый пользователю. На этом этапе в работу включается браузер, именно он преобразовывает полученный ответ от сервера в графический интерфейс, понятный пользователю.

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты (потребители этих функций). Практические реализации такой архитектуры называются клиент-серверными технологиями.

Двухзвенная архитектура - распределение трех базовых компонентов между двумя узлами (клиентом и сервером). Двухзвенная архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объеме (рис. 7).

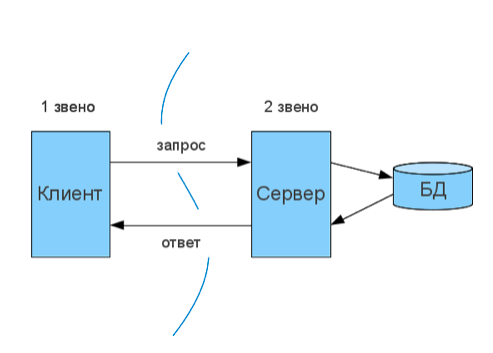


Рисунок 7 – Принцип двухзвенной архитектуры

Расположение компонентов на стороне клиента или сервера определяет следующие основные модели их взаимодействия в рамках двухзвенной архитектуры:

* Сервер терминалов — распределенное представление данных.
* Файл-сервер — доступ к удаленной базе данных и файловым ресурсам.
* Сервер БД — удаленное представление данных.
* Сервер приложений — удаленное приложение.

Клиент – это браузер, но встречаются и исключения (в тех случаях, когда один веб-сервер (ВС1) выполняет запрос к другому (ВС2), роль клиента играет веб-сервер ВС1). В классической ситуации (когда роль клиента выполняет браузер) для того, чтобы пользователь увидел графический интерфейс приложения в окне браузера, последний должен обработать полученный ответ веб-сервера, в котором будет содержаться информация, реализованная с применением HTML, CSS, JS (самые используемые технологии). Именно эти технологии «дают понять» браузеру, как именно необходимо «отрисовать» все, что он получил в ответе.

Веб-сервер – это сервер, принимающий HTTP-запросы от клиентов и выдающий им HTTP-ответы. Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает. Наиболее распространенными видами ПО веб-серверов являются Apache, IIS и NGINX. На веб-сервере функционирует тестируемое приложение, которое может быть реализовано с применением самых разнообразных языков программирования: PHP, Python, Ruby, Java, и прочих.

База данных фактически не является частью веб-сервера, но большинство приложений просто не могут выполнять все возложенные на них функции без нее, так как именно в базе данных хранится вся динамическая информация приложения (учетные, пользовательские данные и пр).

База данных — это информационная модель, позволяющая упорядоченно хранить данные об объекте или группе объектов, обладающих набором свойств, которые можно категоризировать. Базы данных функционируют под управлением так называемых систем управления базами данных (далее – СУБД). Самыми популярными СУБД являются MySQL, MS SQL Server, PostgreSQL, Oracle (все – клиент-серверные).

Рассмотрим подробнее СУБД MySQL, который был задействован при разработке веб-приложения:

MySQL — это популярная, мощная и открытая система управления базами данных (СУБД), ориентированная на работу с реляционными базами данных. Эта платформа широко используется во всем мире в качестве сервера баз данных для веб-приложений и является компонентом знаменитого стека LAMP (Linux, Apache, MySQL, PHP/Python/Perl).

Основные особенности MySQL:

- Открытый исходный код: MySQL доступна в версии с открытым исходным кодом, что позволяет сообществу участвовать в ее развитии и адаптации под свои нужды.

- Кроссплатформенность: MySQL поддерживает все самые популярные операционные системы, включая Linux, UNIX, MacOS и Windows.

- Высокая производительность: система оптимизирована для работы с большими объемами данных и предложения высокой скорости работы.

- Надежность и стабильность: MySQL известна своей стабильностью и способностью эффективно работать 24/7.

- Поддержка больших баз данных: система способна управлять очень большими объемами данных, вплоть до 50 миллионов строк и более.

- Гибкость: MySQL поддерживает множество типов данных, включая числовые, строковые, даты/время и другие.

MySQL используется как для малых, так и для крупных приложений и предлагает ряд мощных функций, таких как:

- Транзакции: поддержка автоматических транзакций для более безопасного выполнения групп операций.

- Полнотекстовый поиск: встроенные механизмы для эффективного выполнения полнотекстового индексирования и поиска.

- Поддержка различных моделей хранения: возможность выбора между разными моделями хранения (Storage Engines), такими как InnoDB (поддерживающий транзакции) или MyISAM (более быстрая производительность для чтения данных).

- Широкие возможности конфигурации: гибкие настройки конфигурации для оптимизации работы сервера под конкретные задачи.

- Масштабируемость: возможность горизонтального и вертикального масштабирования для обработки растущих объемов данных и увеличения требований к производительности.

- Безопасность: расширенные возможности безопасности, включая шифрование данных, управление доступом и разграничение прав на различные уровни.

- Интеграция с инструментами разработки: широкая поддержка и возможности интеграции с популярными веб-технологиями и инструментами программирования.

Благодаря своей надежности и преимуществам, MySQL является одним из ведущих выборов для систем управления базами данных в веб-разработке и не только.

**1.7.3 Фреймворк**

Как правило, веб-разработчики используют для написания web-страниц различные фреймворки, что существенно ускоряет работу проектирования блоков сайта. Для разработки веб-приложения был выбран фреймворк Laravel версии 10.

Laravel — это популярный фреймворк для веб-разработки, написанный на языке PHP и использующий архитектурный шаблон MVC (Model-View-Controller). Этот фреймворк является открытым программным обеспечением и позволяет разработчикам строить качественные и хорошо структурированные веб-приложения более эффективно и быстро благодаря своему богатому набору функций и инструментов.

Важные особенности Laravel включают:

- Eloquent ORM (Object-Relational Mapping): встроенная система управления объектами баз данных, позволяющая работать с базами данных через объектно-ориентированный подход.

- Blade шаблонизатор: мощная и гибкая система шаблонов, которая позволяет отделять логику приложения от кода разметки.

- Маршрутизация: простая в настройке система маршрутизации, позволяющая легко связать маршруты с классами-контроллерами.

- Система миграций баз данных: позволяет версионировать и делиться структурой базы данных между разработчиками.

- Инструменты для авторизации и аутентификации: встраиваемые механизмы для управления пользователями и полномочиями.

- Расширенные функции безопасности: в Laravel включены средства защиты от различных уязвимостей, таких как SQL-инъекции, XSS и CSRF.

- Artisan консоль: командная строка, входящая в Laravel, предоставляет множество полезных команд для ускорения разработки.

- Система управления пакетами Composer: управление зависимостями проекта с помощью Composer для легкой установки и обновления библиотек.

Laravel подходит для проектов различного масштаба, от маленьких проектов до крупных корпоративных приложений. Благодаря своему простому и интуитивно понятному синтаксису, он оказывается хорошим выбором для начинающих разработчиков, в то же время предоставляя продвинутые инструменты для более опытных профессионалов.

Использование фреймворка Laravel предлагает множество преимуществ для разработчиков при создании веб-приложений:

1. Быстрая разработка: благодаря множеству встроенных функций, Laravel значительно сокращает время разработки. Он обеспечивает инструменты для многих общих задач, таких как аутентификация, роутинг, сессии и кэширование.

2. Понятный синтаксис: Laravel известен своим чистым и выразительным синтаксисом, что делает написание кода более удобным и легко читаемым.

3. MVC-архитектура: Laravel использует модель MVC, которая способствует четкому разделению бизнес-логики и пользовательского интерфейса, улучшая тем самым структуру кода и облегчая поддержку приложения.

4. Инструменты для безопасности: предоставляет различные механизмы защиты от таких уязвимостей, как SQL-инъекции, кросс-сайтовые скриптовые атаки (XSS) и подделка межсайтовых запросов (CSRF).

5. Мощная ORM: Eloquent ORM в Laravel позволяет разработчикам использовать объектно-ориентированный подход при взаимодействии с базами данных.

6. Миграции баз данных: Система управления версиями для баз данных позволяет легко и надежно вносить изменения в структуру базы данных и делиться этими изменениями с командой разработчиков.

7. Обучающие ресурсы: существует множество ресурсов для обучения, включая официальную документацию, учебные курсы, видеоуроки, книги и статьи.

Эти и многие другие преимущества делают Laravel обширно используемым среди разработчиков и одним из лидеров среди PHP-фреймворков для разработки веб-приложений (рис. 8).

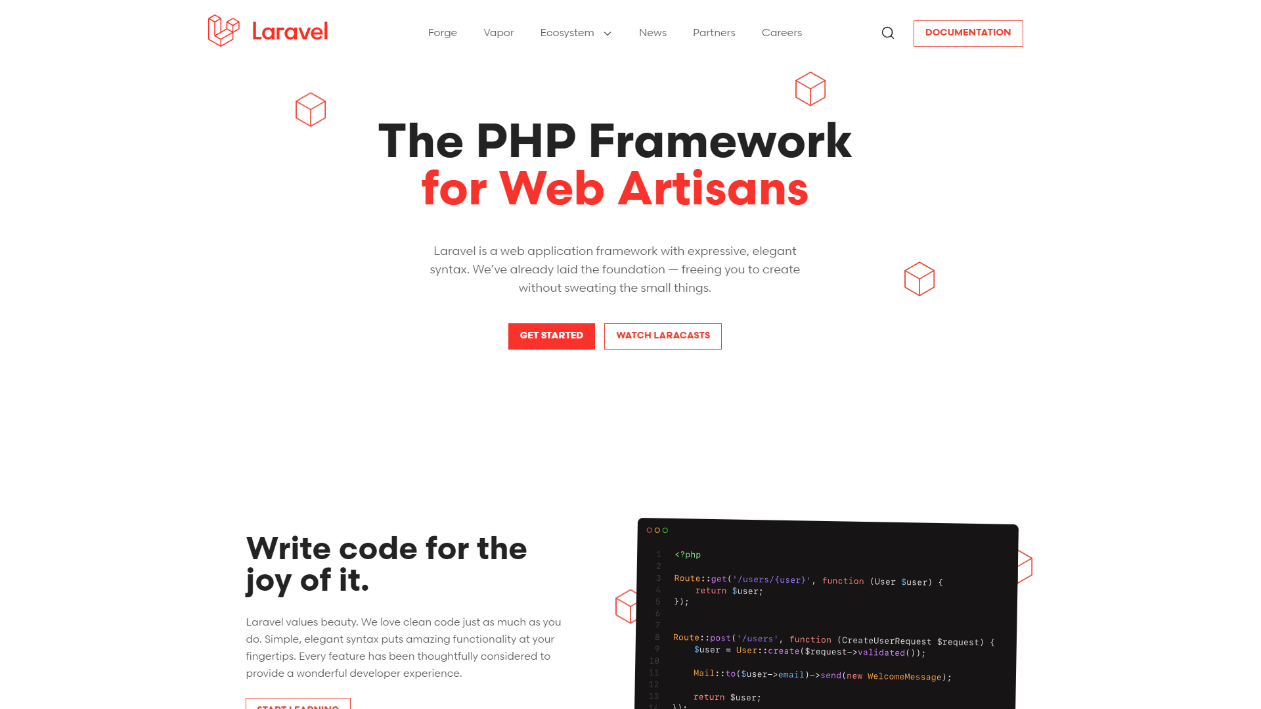
.

Рисунок 8 – Официальный сайт фреймворка Laravel

### 1.7.4 Конструкторы и CMS

Конструкторы — это сервисы для создания веб-сайтов онлайн, прямо в браузере. Как правило, они не требуют знания языков программирования и рассчитаны на людей не из сферы разработки и программирования и в основном на тех, у кого нет финансовых ресурсов на найм разработчиков в команду. С помощью конструкторов обычно создаются простые статические сайты-визитки (промо-страницы), хотя в самых современных конструкторах можно создать и полноценный сайт, к примеру интернет-магазин.

Базовый функционал конструкторов доступен бесплатно, но в большинстве случаев за любое нестандартное решение необходимо платить.

CMS, как было сказано ранее, представляет собой систему управления сайтом, существенно облегчающую его создание, добавление и редактирование материалов, организацию взаимодействия с пользователями и т.д. Все современные CMS умеют работать с базами данных, высокая функциональность обеспечивает возможность создания и развития проектов любого уровня сложности.

# 

# 2. Практическая часть. Выбор средств реализации

**2.1 Программное обеспечение**

При проектировании и разработке веб-приложения для организации дополнительного образования были использованы Adobe Photoshop и Figma на этапе проектирования, PhpStorm на этапе написания кода, инструменты разработчика Chrome на этапе отладки кода. Мы рассмотрим каждый из этих инструментов:

**2.1.1 Adobe Photoshop**

Adobe Photoshop - программа, предназначенная для работы с графикой. Другими словами, это компьютерное приложение, предназначенное для редактирования и манипуляций с цифровыми изображениями.

Данная программа является чрезвычайно мощным приложением, которым пользуются многие профессиональные фотографы и дизайнеры.

Photoshop подходит для любого вида редактирования изображений, при этом позволяя создавать высококачественную графику. Исходя из того, что Photoshop, в основном, предназначен для профессионалов, то он может быть немного сложным в использовании, особенно если пользователь только начал с ним работать. Тем не менее, пользователю не нужно понимать все его особенности и тонкости, чтобы эффективно использовать его и при этом добиться желаемого результата (рис. 9).

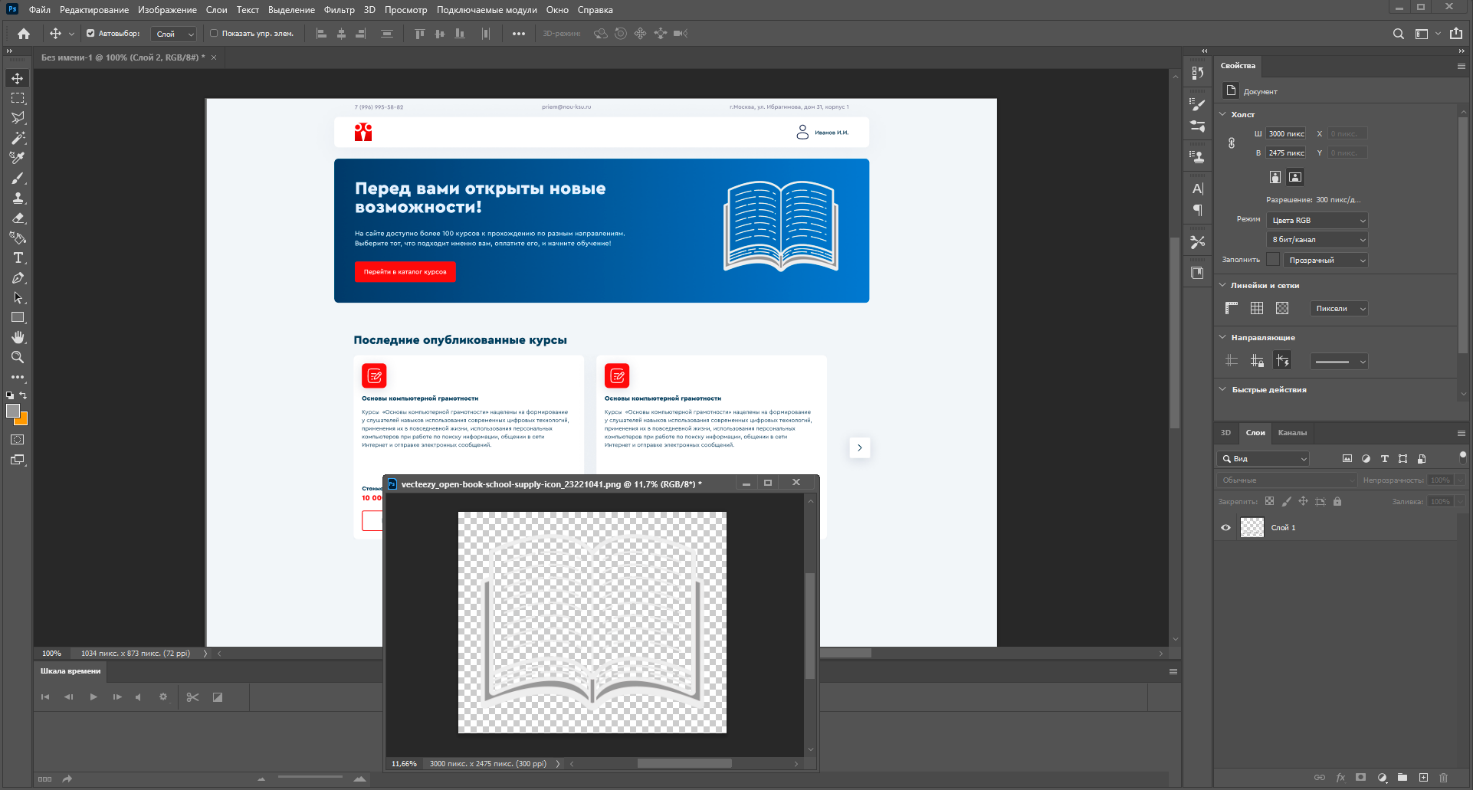


Рисунок 9 – Интерфейс программы Adobe Photoshop

**2.1.2 Figma**

Figma — кроссплатформенный онлайн-сервис для дизайнеров интерфейсов и веб-разработчиков. Разработка интерфейсов происходит в онлайн-приложении.

У Figma две ключевые особенности: доступ к макету прямо из окна браузера и возможность совместной работы над документами.

До появления Figma нескольким дизайнерам сложно было работать над одним проектом и передавать макеты разработчикам. Photoshop отказывался корректно открывать макет, пока вы пользователь не установит нужные шрифты. Или коллега вносил изменения в свою копию проекта и забывал сказать пользователю об этом.

Команда Figma учла подобные проблемы и создала продукт, который позволяет работать над проектом одновременно нескольким дизайнерам, поддерживает версионность и даёт много других возможностей. Они упрощают жизнь как дизайнеру, так и разработчику.

Все файлы хранятся в облаке Figma. Не надо вспоминать, где лежат макеты, скачивать и после того, как внесли изменения, заливать их обратно.

Изменения в Figma сохраняются автоматически.

В Figma легко просматривать ранние версии файла и восстанавливать или дублировать любую из них. Версии автоматически сохраняются, если в течение 30 минут в файле не было никаких изменений.

Фреймы в Figma очень похожи на артборды в Photoshop, но имеют большую гибкость в работе: они хорошо масштабируются и их с лёгкостью можно использовать в других фреймах. Для фреймов есть предустановленные размеры устройств: Phones, Tablet, Desktop, Watch, Paper, Social Media. Поэтому нет надобности держать их в голове и путаться в размерах (рис. 10).

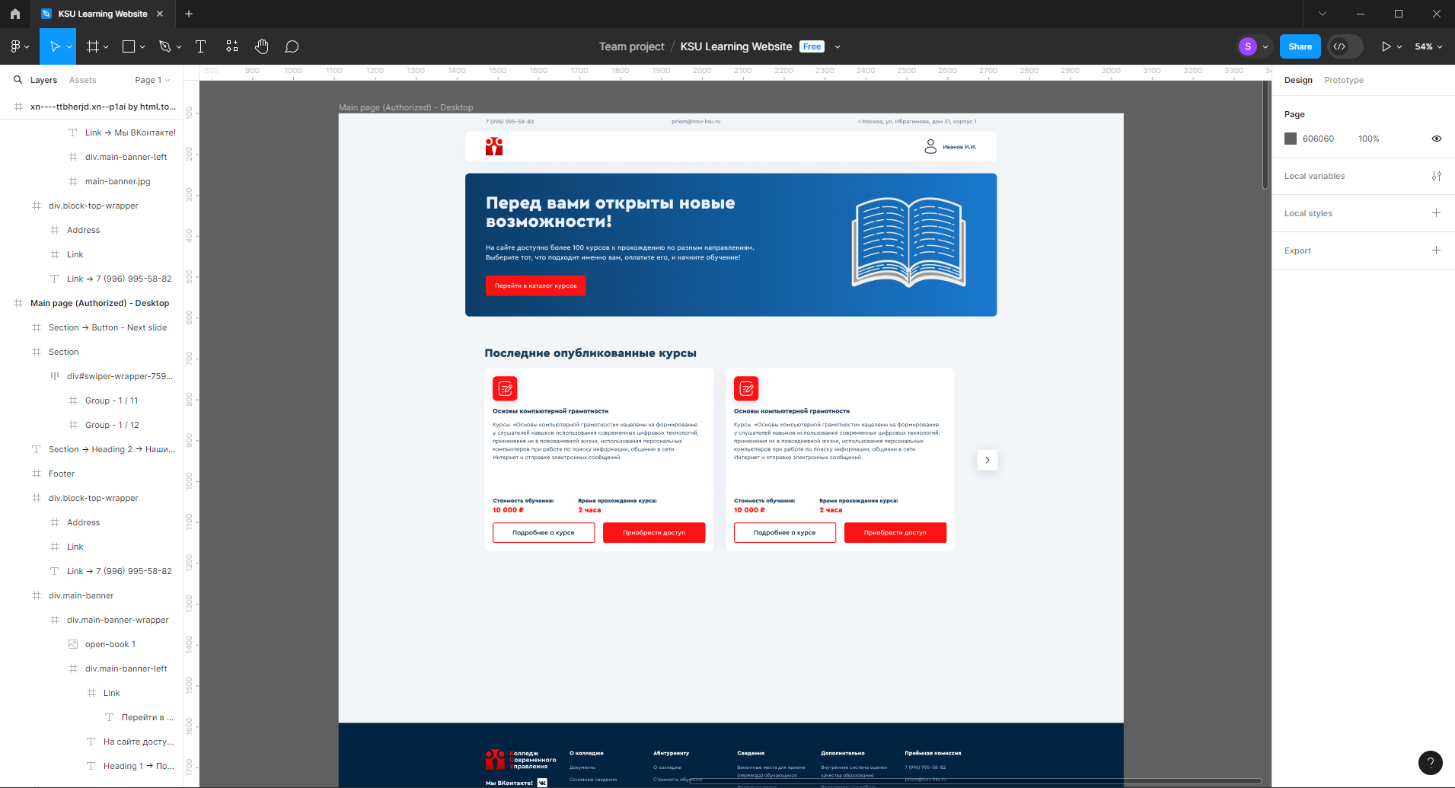


Рисунок 10 – Интерфейс программы Figma

**2.1.3 PhpStorm**

PhpStorm — это среда разработки, которая по-настоящему понимает код пользователя. IDE обеспечивает первоклассную поддержку PHP от версии 5.3 до актуальной на текущий момент (8.3), мгновенно предотвращает ошибки, предоставляет точное автодополнение и безопасные рефакторинги, а также возможность редактирования кода на HTML, CSS и JavaScript (рис. 11).

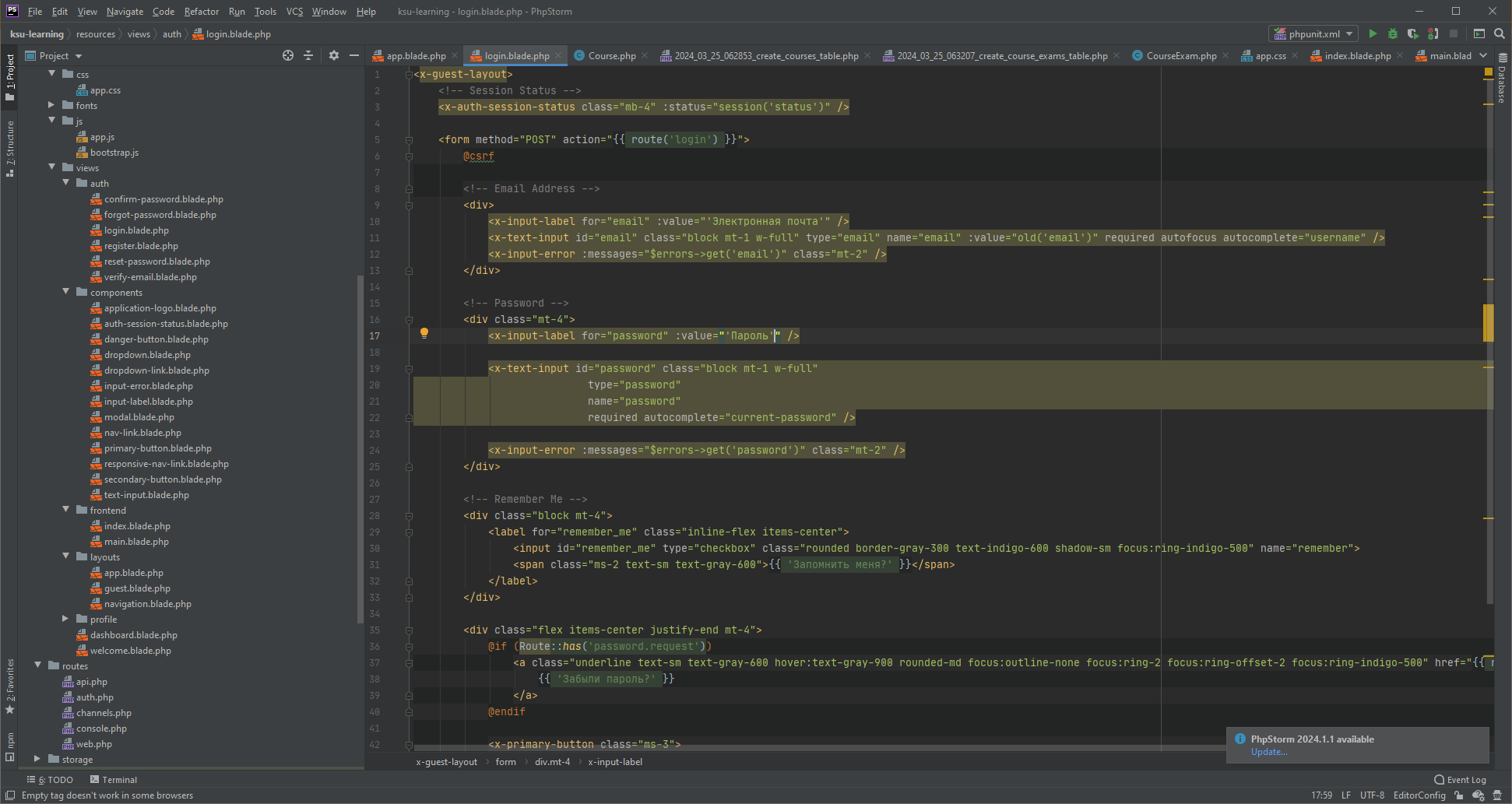


Рисунок 11 – Интерфейс программы PhpStorm

PhpStorm – умный редактор PHP-кода. IDE обеспечивает умное автодополнение, подсветку синтаксиса, расширенные настройки форматирования, проверку ошибок на лету, сворачивание фрагментов кода, поддержку языковых комбинаций и много других возможностей для продуктивной разработки. Автоматические рефакторинги позаботятся о порядке в вашем коде и помогут легко и безопасно выполнять глобальные настройки проекта.

Сотни встроенных инспекций проверяют код прямо в режиме редактирования, а также позволяют проанализировать весь проект на наличие ошибок и проблем в структуре кода. Для большинства инспекций доступны быстрые исправления, которые позволяют откорректировать код мгновенно.

В один клик можно перейти к определению любого метода, функции или переменной или найти их использования в проекте.

PhpStorm имеет встроенные инструменты для отладки, трассировки и профилирования приложений.

Программа построена на базе платформы IntelliJ, которую в JetBrains развивают и совершенствуют на протяжении 15 лет.

Продуманная стандартная конфигурация в сочетании с широкими возможностями настроить IDE делают PhpStorm удобной для всех разработчиков.

**2.1.4 Инструменты разработчика Chrome**

В каждый современный браузер встроены инструменты разработчика — они позволяют быстро отловить и исправить ошибки в разметке или в коде. С их помощью можно узнать, как построилось DOM-дерево, какие теги и атрибуты есть на странице, почему не подгрузились шрифты и многое другое.

Есть три способа получить информацию о любом элементе на странице:

* Через инспектор объектов. Способ удобен, если пользователь точно видит элемент и можно кликнуть по нему правой кнопкой мыши. Для этого кнопка Inspect или «Посмотреть код». Заодно сразу откроется панель разработчика.
* Поиск по элементам. При показанном отладчике нажимаем горячие клавиши Ctrl+F, и внизу появится окно поиска по тегу, атрибуту, классу или текстовому содержимому элемента.
* Визуальный поиск. При открытом отладчике открываем режим визуального поиска, находим нужный элемент и кликаем по нему (рис. 12).

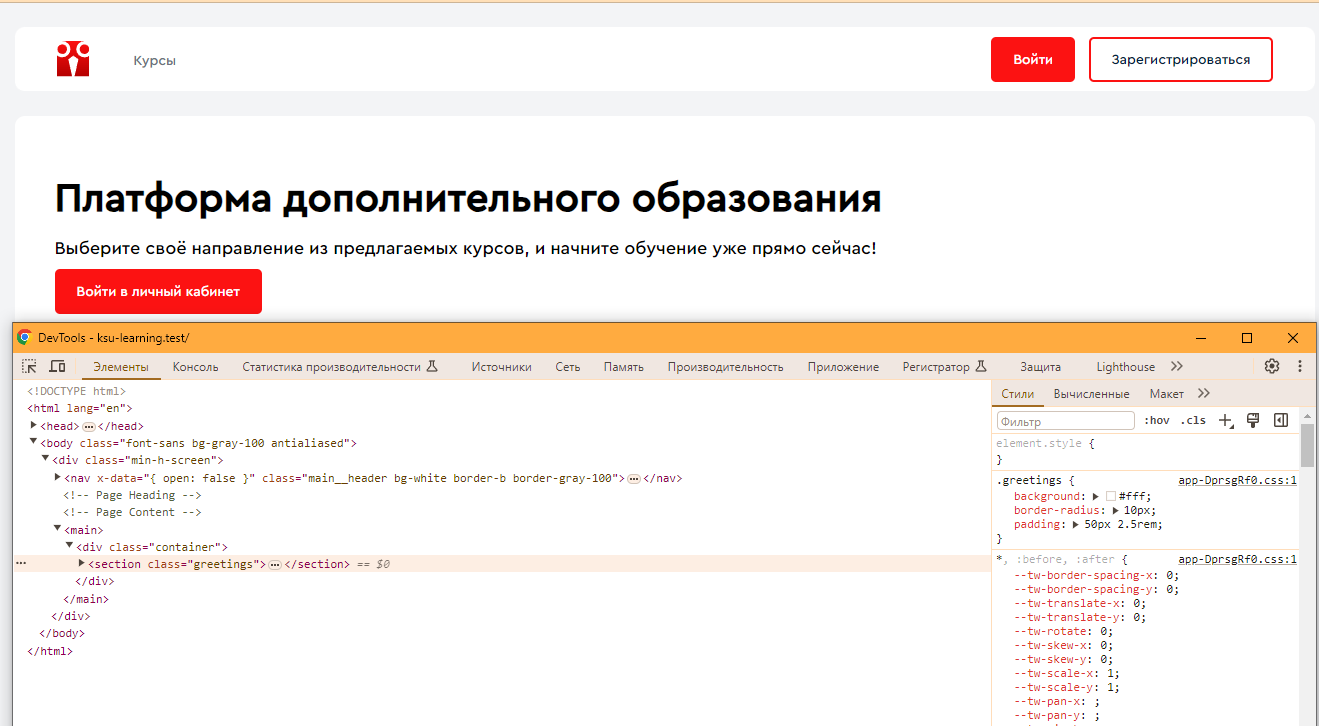


Рисунок 12 – Режим просмотра кода элемента в Chrome

Консоль Chrome — это инструмент, с помощью которого можно не только посмотреть наполнение страницы, выводимой браузером, а также существующие ошибки (что чаще всего и делают тестировщики), но и исправить эти ошибки намного проще и быстрее (что чаще всего делают девелоперы), замерять различные показатели и манипулировать страницей (рис. 13).

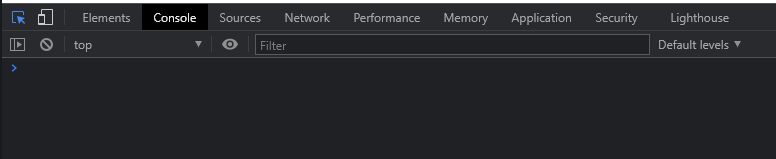


Рисунок 13 – Режим консоли в Chrome

Итого в ней есть 8 вкладок, каждая из которых отображает определенные данные:

1. Elements (содержит весь HTML/CSS код страницы и позволяет выбрать элементы для исследования, а также редактировать их)
2. Console (отображает наличие/отсутствие ошибок/предупреждений в коде)
3. Sources (позволяет выполнять операции с кодом страницы)
4. Network (отслеживает время исполнения определенных запросов и сами запросы)
5. Timeline (измеряет время загрузки страницы)
6. Profiles (позволяет создавать JavaScript, профили CPU)
7. Resources (позволяет просмотреть определенные сохраненные данные)
8. Audits (проводит проверки определенных параметров)

**2.1.5 Laragon**

Laragon – это портативная, изолированная среда окружения для работы с PHP, Node.js, Java, MySQL, Apache, Nginx и других веб-технологий на платформе Windows. Она предоставляет все необходимые инструменты для разработки веб-проектов и предназначена для упрощения и ускорения процесса веб-разработки, а также проверки конечного результата.

Из основных особенностей среды Laragon можно выделить:

* Быструю установку: Laragon быстро устанавливается, и его можно конфигурировать для работы с различными версиями PHP, MySQL, Apache и т. д.
* Простоту использования: интерфейс Laragon простой и интуитивно понятный, благодаря чему освоение программы легко даже для новичков.
* Изоляцию: Laragon работает в изолированной среде, что позволяет избежать конфликтов с уже установленными на компьютере программами и сервисами.
* Поддержку мультиверсионности: пользователи могут запускать разные версии PHP и серверных программных пакетов параллельно.
* Автоматизацию создания виртуальных хостов: Laragon автоматически создает и управляет виртуальными хостами.
* Портабельность: весь пакет Laragon можно переносить на флешку или другой носитель, сохраняя при этом все настройки и проекты.
* Быструю и эффективную систему резервного копирования и восстановления: утилиты для резервного копирования и восстановления данных интегрированы непосредственно в Laragon.
* Поддержку широкого спектра дополнений (аддонов): Laragon может быть расширен дополнительными сервисами и инструментами с помощью аддонов.

Laragon предназначен для разработчиков любого уровня квалификации, предлагая им полную и легко конфигурируемую среду, которая работает прямо «из коробки» (рис. 14).

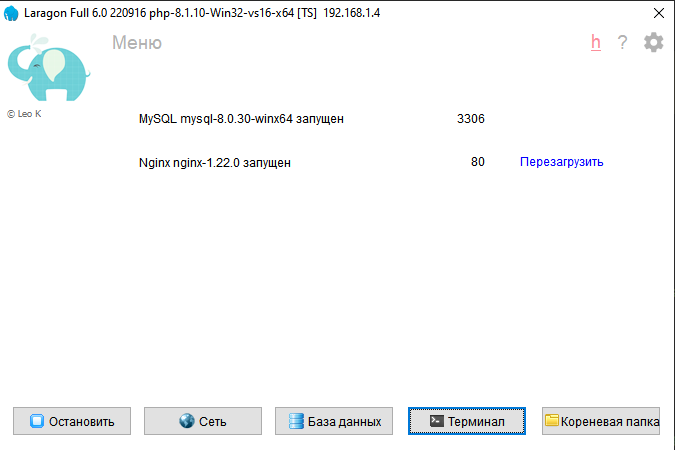


Рисунок 14 – Интерфейс программы Laragon

**2.2 Аналитическое моделирование**

На этапе аналитического моделирования строятся статическая и динамическая модели системы – статическая модель задает структурные отношения между классами предметной области.

Классы и отношения между ними отображаются на диаграммах классов. Для определения того, какие объекты следует рассматривать в аналитической модели, применяются критерии разбиения на объекты.

Затем создается динамическая модель, в которой уточняются прецеденты из модели требований, с целью показать, какие объекты участвуют в каждом прецеденте и как они взаимодействуют. Объекты и их взаимодействия отображаются на диаграммах кооперации или последовательности. В динамической модели объекты, зависящие от состояния, определяются с помощью диаграмм состояний. [1, с.128].

Диаграмма вариантов использования (use case diagram) – диаграмма, на которой изображаются отношения между пользователем и вариантами использования. Данная диаграмма – это исходное концептуальное представление или концептуальная модель системы в процессе ее проектирования и разработки. Создание диаграммы вариантов использования имеет следующие цели:

* Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы
* Сформулировать общие требования к функциональному поведению проектируемой системы
* Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей
* Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями

Назначение данной диаграммы состоит в следующем: проектируемая программная система представляется в форме так называемых вариантов использования, с которыми взаимодействуют внешние сущности или пользователи. При этом пользователем или действующим лицом называется любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определит разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет набор действий, совершаемый системой при диалоге с пользователем. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие пользователей с системой и собственно выполнение вариантов использования (рис. 15). [2, с.38]



Рисунок 15 - Диаграмма вариантов использования

Как видно из схемы, после регистрации на сайте, у обучающегося есть возможность просматривать информацию о курсах, после покупки курса доступно его изучение, после чего происходит сдача тестов по модулю и экзаменов.

Преподаватель на портале имеет возможность добавлять и редактировать созданные им курсы, а также добавлять в ручном режиме студентов на курс. В курсе имеется возможность добавления и редактирования теории, практических заданий и тестов по модулю. Те же функции доступны и администратору системы.

**2.3 Проектное моделирование**

На этапе проектного моделирования разрабатывается программная архитектура системы, при этом аналитическая модель отображается на эксплуатационную среду. Аналитическая модель, в которой наиболее важной являлась предметная область, переносится на проектную модель, где главной становится область решения. Для декомпозиции решения применяются критерии разбиения на подсистемы, последние рассматриваются как агрегированные или составные объекты. Особое внимание уделяется проектированию распределенных систем в виде конфигурируемых компонентов, которые взаимодействуют путем обмена сообщениями.

Затем проектируется каждая подсистема. Для последовательных систем акцент делается на объектно-ориентированных концепциях сокрытия информации, классов и наследования. При проектировании параллельных систем – реального времени, клиент-серверных и распределенных – помимо объектно-ориентированных концепций необходимо принять во внимание также концепции параллелизма [1, с.130].

## 2.4 Реализация программного продукта

**2.4.1 Структура web-сайта**

Структура web-сайта – это его каркас, определяющий порядок навигации. Для того, чтобы сайт осуществлял необходимые ему действия, она должна соответствовать ряду требований, а именно содержать определенные элементы и обладать некоторыми характеристиками. Глядя только на структуру сайта, должно быть понятно, каким функционалом он обладает.

Навигационная схема Web-сайта зависит от его структуры и определяет то, как пользователь будет по нему перемещаться и получать доступ к информации, которую Вы представляете. Простота и удобство навигации является одним из важных факторов, определяющих посещаемость web-сайта. Пользователи должны быстро и легко перейти на любую страницу web-сайта, в том числе на начальную. Именно на этом этапе закладываются основные принципы работы сайта, его структура, формируется общее представление о дальнейшей работе над проектом. Также на этом этапе необходимо придумать названия разделов сайта, заголовки страниц, определить переходы между ними, то есть продумать логическую структуру размещения информации. В результате третьего этапа должна быть сформирована ясная и логическая структура размещения информации на сайте - ничто так не утомляет при поиске нужной информации, как плохо структурированные сайты.

Для понимания того, какой контент требуется на сайте и что должен представлять из себя разрабатываемый сайт, была составлена структурная схема, представленная на рисунке 16.

Рисунок 16 – Общая структурная схема сайта дополнительного образования

Чаще всего структура Web-сайта представляет собой иерархию. При этом сначала создают категории высшего уровня, а затем материал в логическом порядке размещают в категории, которые находятся ниже. По созданной схеме можно создать карту сайта. Карта сайта — это своеобразная диаграмма, позволяющая визуально представить веб-страницы еще до их построения.

**2.4.2 Дизайн web-сайта**

Создание сайта для организации дополнительного образования требует особого подхода к его дизайну. И наиболее ярким путем к достижению его высокой посещаемости является использование цветовой гаммы, которая отражает все основные элементы сайта.

Создание подобных сайтов осуществляется с тщательным подбором цветов, так как неправильное цветовое сочетание сведет на нет все усилия по привлечению посетителей. Неверно подобранные цвета равносильны по негативному воздействию на посетителей написанным с ошибками заголовкам статей, низкой скорости загрузки сайта, массовым «всплывающим окнам» и огромному количеству анимационных эффектов, действующих раздражающе. [6, с.1]

В первую очередь, рассмотрим дизайн создаваемого сайта. Дизайн включает в себя цветовое оформление, наличие или отсутствие логотипа.

Для разработки главной страницы нужно составить макет, концепт дизайна. Выполнить это можно в любом графическом редакторе: Adobe Photoshop, Adobe Illustrator или Figma.

Для разработки макета был выбран инструмент Figma – наиболее распространенный в настоящее время. Преимущество данного инструмента заключается в том, что все разрабатываемые макеты и полноценные проекты доступны в веб-версии и могут редактироваться в процессе непосредственно командой людей, наблюдать за редактированием можно в режиме реального времени (рис. 17-18).

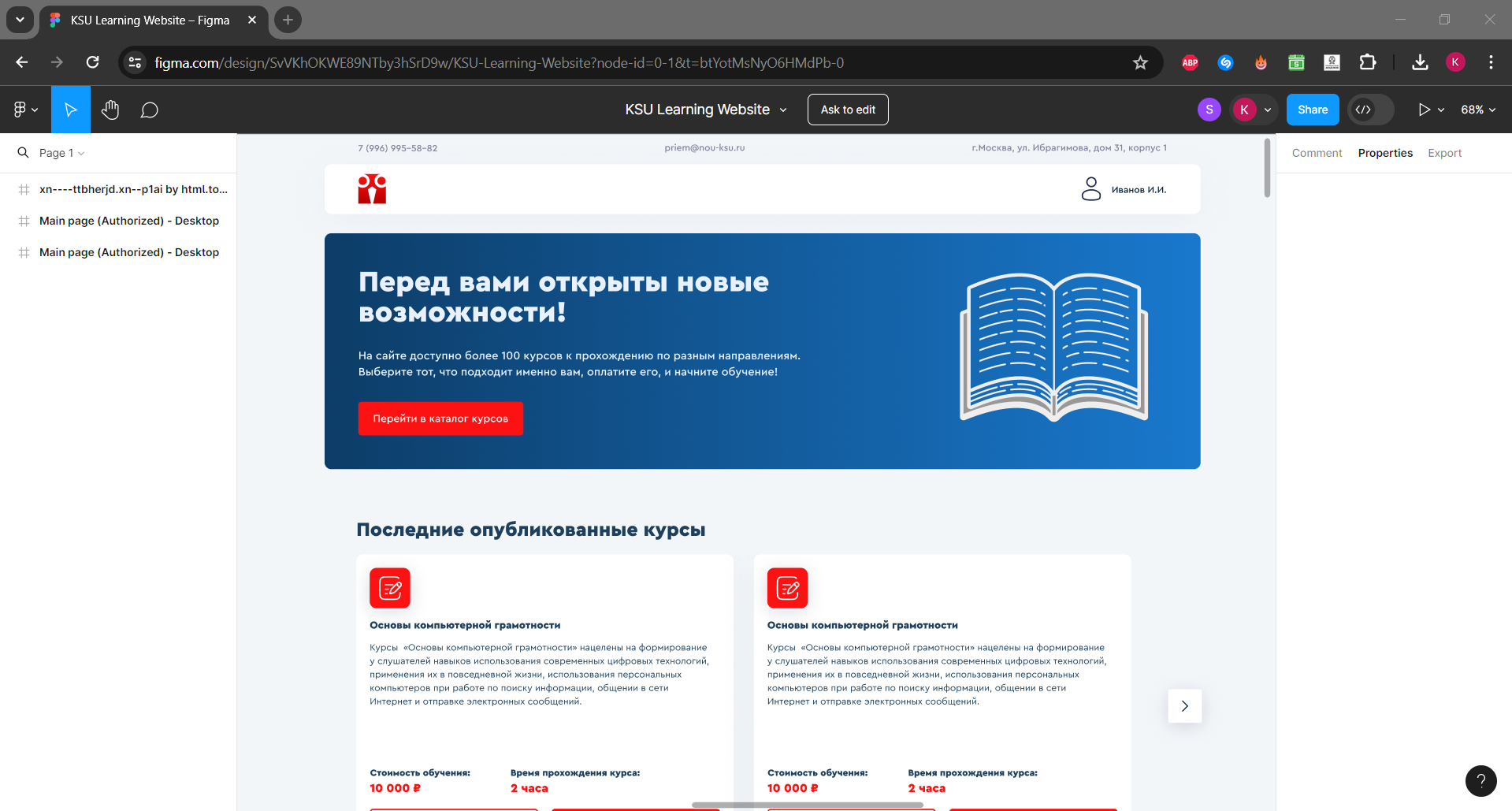


Рисунок 17 – Макет сайта дополнительного образования для НОУ КСУ, выполненный в Figma

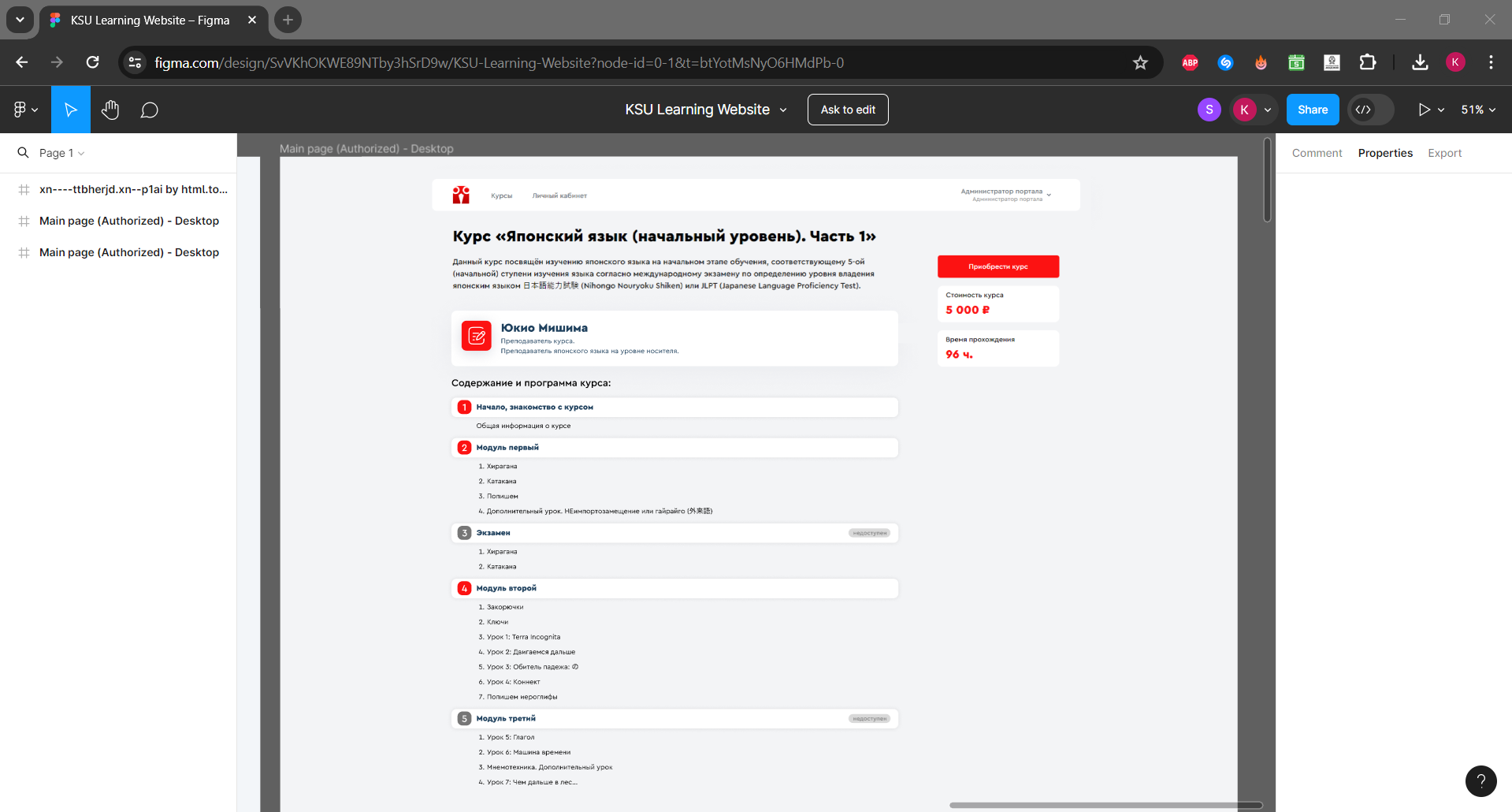


Рисунок 18 – Макет страницы курса, выполненный в Figma

Для наиболее детального ознакомления с макетом создаваемого продукта можно по адресу в интернете:

https://www.figma.com/design/SvVKhOKWE89NTby3hSrD9w/KSU-Learning-Website.

Целевой аудиторией сайта будут являться потенциальные клиенты, которые уже являются студентами колледжа и узнали о курсах от преподавателей, администрации или рекламы в учреждении. Соответственно, оформление должно быть максимально похожим на основной сайт колледжа, что позволит посетителю быстро сориентироваться в дальнейших действиях.

Любой организации важно закрепить за собой бренд и сделать запоминающееся оформление, которое будет подчёркивать статус организации. Для этого в организациях дизайнеры вводят брендбуки.

Брендбук — это официальный документ, в котором приведены стандарты визуального стиля вашей компании. В нем описаны основные ценности и особенности в продвижение и развитии бренда. Двух одинаковых брендбуков не бывает. В то время как одни компании делают упор на аспектах дизайна, другие создают более подробный документ с обзором компании и правилами взаимодействия с клиентами и деловыми партнерами.

Если у вас нет единого набора правил, то каждый дизайнер и маркетолог, которого вы будете привлекать для решения тех или иных задач, будет реализовывать свои идеи в соответствии со своим видением. В разнообразии нет ничего плохого, но в некоторых ситуациях отсутствие единства может иметь плачевные последствия. Если вы будете использовать разные дизайнерские стили и методики взаимодействия с людьми, вы не сможете построить сильный бренд и убедить потенциальных клиентов в том, что ваша компания является лучшей в своем сегменте.

Руководство по использованию фирменного стиля будет особенно полезно для новых сотрудников. Заранее ознакомившись с правилами, новички быстрее сориентируются в системе бренда. Так вам не придется проводить беседы с новыми сотрудниками и объяснять им, что они нарушили правило, о существовании которого они даже не догадывались. Таким образом, грамотно составленный брендбук экономит ваше время и сокращает количество неловких ситуаций. [7, с.1]

На официальном сайте НОУ КСУ https://ноу-ксу.рф/ размещён брендбук с описанием используемых в оформлении цветов, начертаний шрифтов и вариантов логотипа. Согласно этому брендбуку, используя его элементы, включая цвета, и будет разработан сайт для организации дополнительного образования.

**2.4.3 Реализация блоков и секций**

Для разработки сайта были скачаны и подключены необходимые файлы: CSS – отвечающие за реализацию дизайна сайта, JS – файлы JavaScript, отвечающие за реализацию скриптов сайта. Для обеспечения единого стиля страниц используется общий шаблон, от которого строятся все страницы сайта. Код шаблона страниц отдельно указан в приложении 1.

Согласно брендбуку НОУ КСУ, на сайте используется шрифт Cera Pro и указанные в нём цвета.

На главный блок было размещено меню, которое перемещает по секциям сайта и была добавлена кнопка с логотипом НОУ КСУ, которая перенаправляет на официальный сайт (рис. 19).

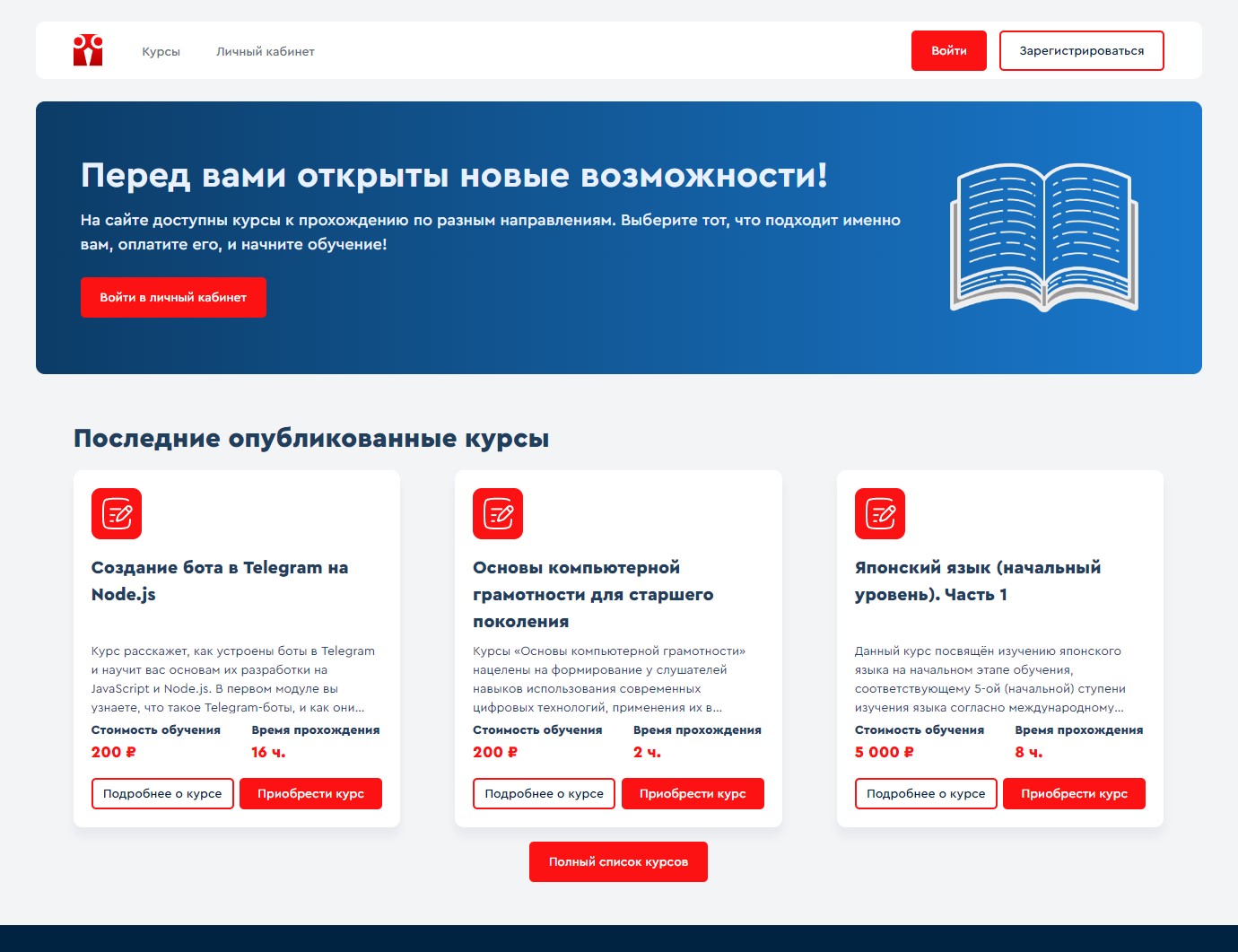


Рисунок 19 – Реализация главной страницы без пройденной авторизации

При первом заходе на сайт нас встречает главная страница ресурса. На ней мы можем увидеть основное меню сайта, а также кнопки перехода в личный кабинет посредством авторизации или регистрации.

В блоке «Последние опубликованные курсы» показываются три курса, которые были добавлены в базу данных последними. Внутри каждого блока с курсом указана краткая информация о курсе, стоимость обучения, а также время прохождения. Также есть ссылки перехода: на страницу с подробной информацией о материале и на страницу его покупки (рис. 20).

Код главной страницы отдельно размещен в приложении 2.

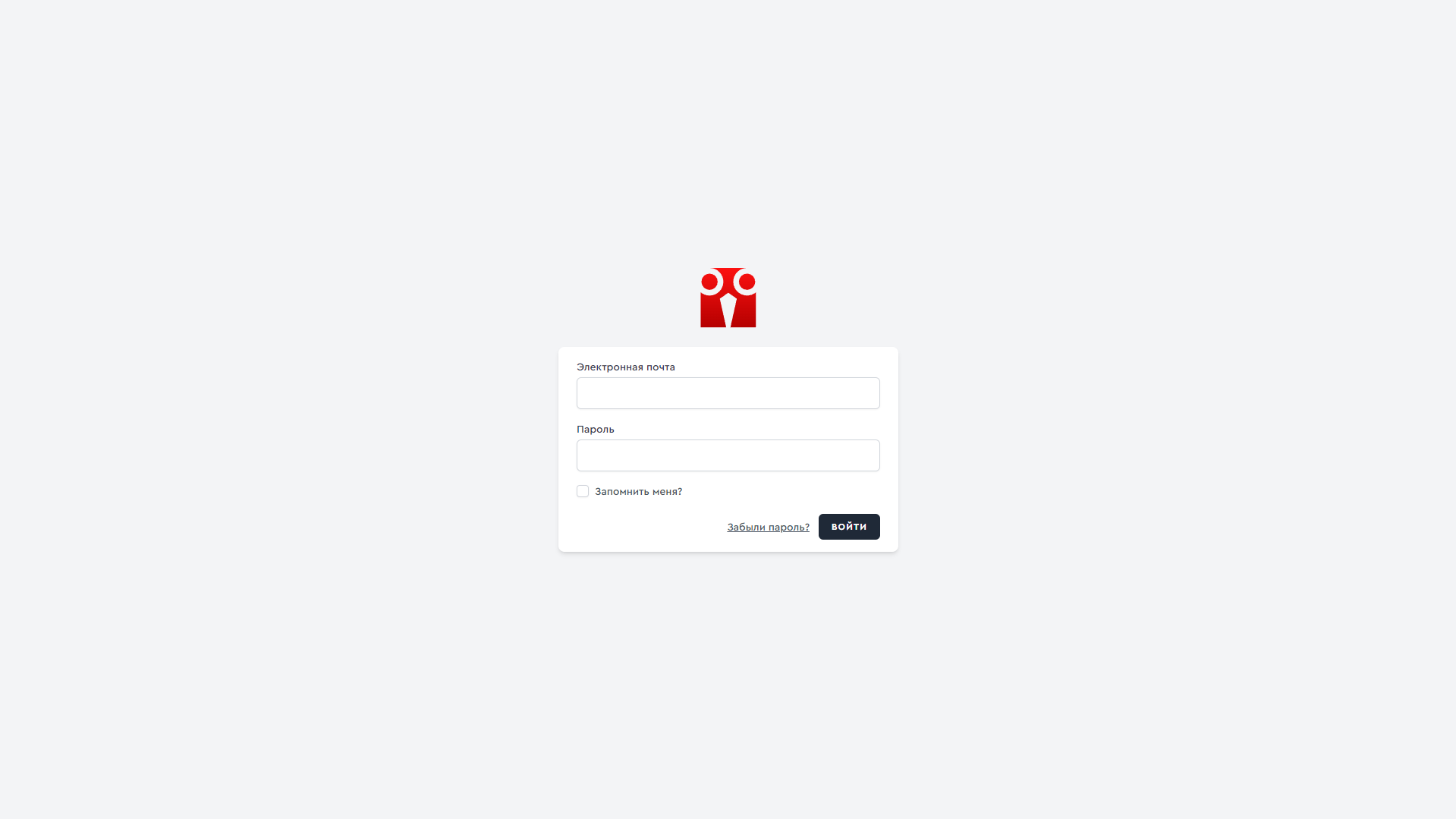


Рисунок 20 – Страница авторизации

На странице авторизации пользователю предлагается ввести электронную почту, на которую был зарегистрирован аккаунт, и пароль от этого аккаунта. По нажатии кнопки «Войти» происходит запрос в базу данных на соответствие пары «логин-пароль». Система проверяет, существует ли аккаунт с указанной электронной почтой (она выступает логином в этом случае) и, если это так, дополнительно сверяет правильность введённого пароля.

В случае успешной авторизации посетитель попадёт в раздел «Личный кабинет», в противном случае ему будет выведено сообщение об ошибке «Неверный логин или пароль».

Также предусмотрен механизм восстановления пароля, если вдруг сам пароль утерян пользователем. Для этого пользователю предлагается нажать на ссылку «Забыли пароль?».

Материалы для создания курсов были разработаны методистами колледжа или были взяты из открытых источников. Для курсов, созданным по общедоступным материалам, цена указана условно, для демонстрации возможностей сайта. Список доступных курсов и страница отдельного курса представлены на рисунках 21-22. Код страницы размещен в приложении 3.

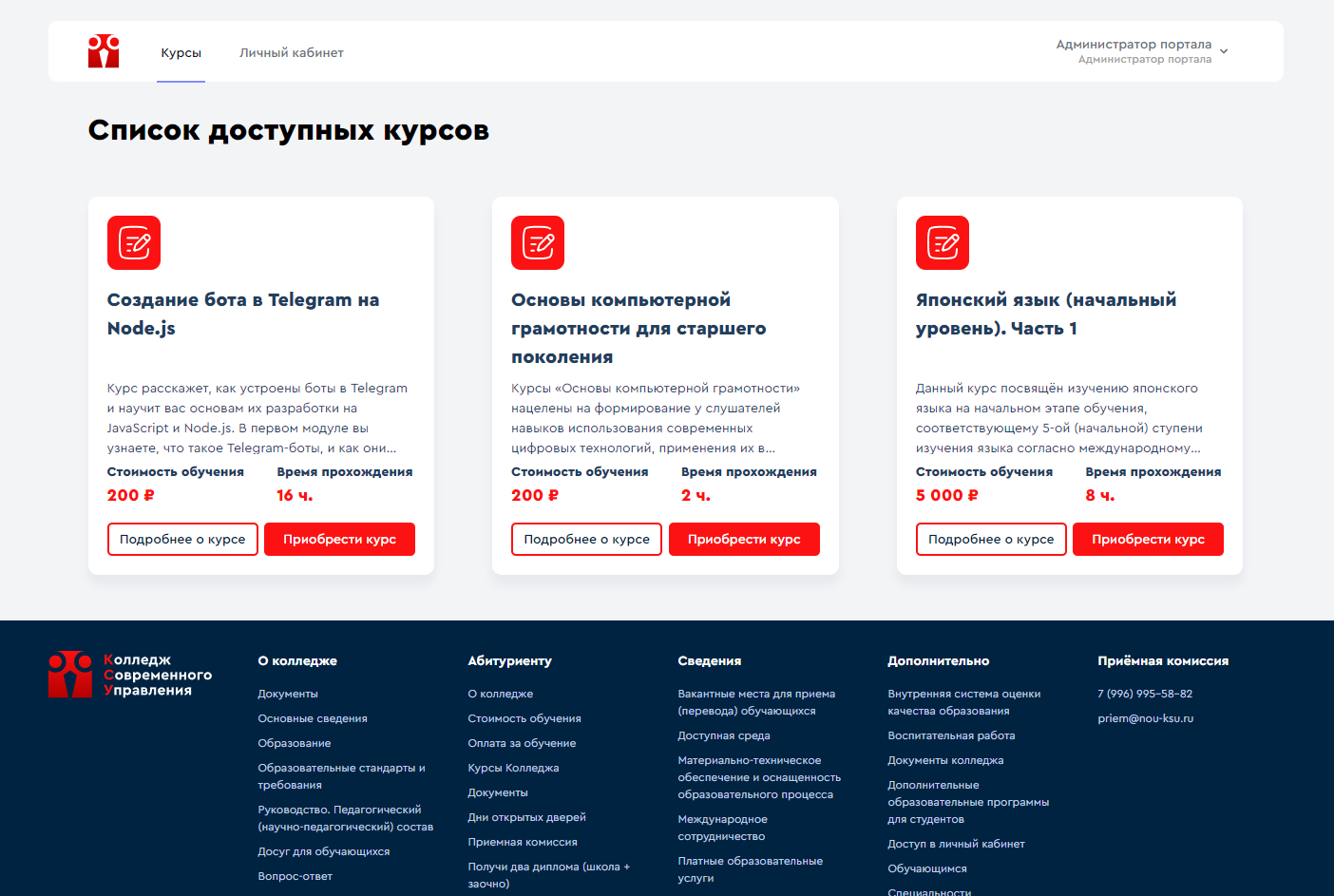


Рисунок 21 – Список доступных курсов

В разделе «Список доступных курсов» показаны все добавленные на сайт курсы, доступные для получения дополнительного образования. По аналогии с блоком «Последние опубликованные курсы» на главной странице ресурса, каждый отображаемый курс имеет краткую информацию о нём, возможность посмотреть подробную информацию на отдельной странице, а также приобрести его. Сортировка курсов в данном разделе производится в обратном хронологическом порядке, то есть сначала будут показаны те курсы, которые были добавлены на сайт последними.

По мере наполнения ресурса также планируется добавить постраничный вывод списка курсов для того, чтобы ускорить загрузку страницы раздела.

Исходный код страницы отдельно размещен в приложении 4.

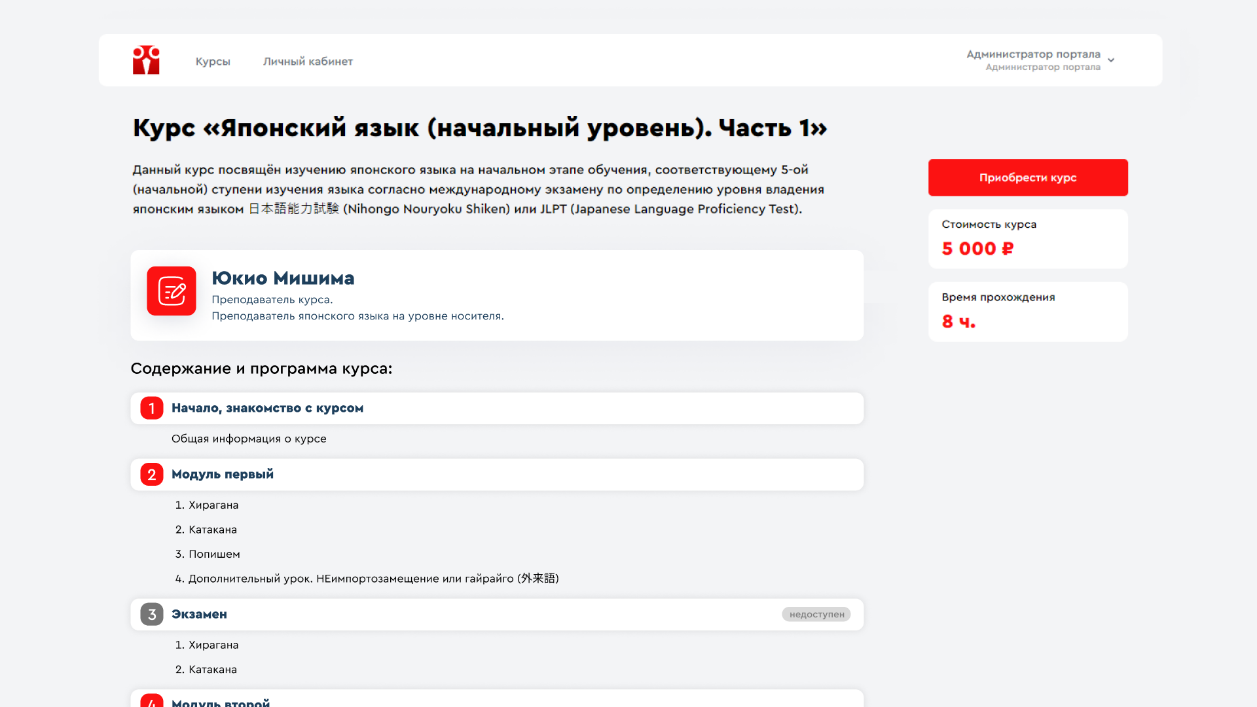


Рисунок 22 – Скриншот отдельного курса

Страница отдельно взятого курса содержит более подробную информацию об обучающем материале. Помимо вводного описания самого курса, на данной странице также описывается его содержание и информация о преподавателе, который ведёт данный курс.

В разработанной панели администратора (личном кабинете) реализованы возможности управления заказами, просмотр списка студентов, записавшихся на курсы и преподавателей, ведущих курс. Также имеется возможность управления самими курсами и просматривать список транзакций – оплаты курсов (рис. 23).

Код страницы курса размещен в приложении 5.

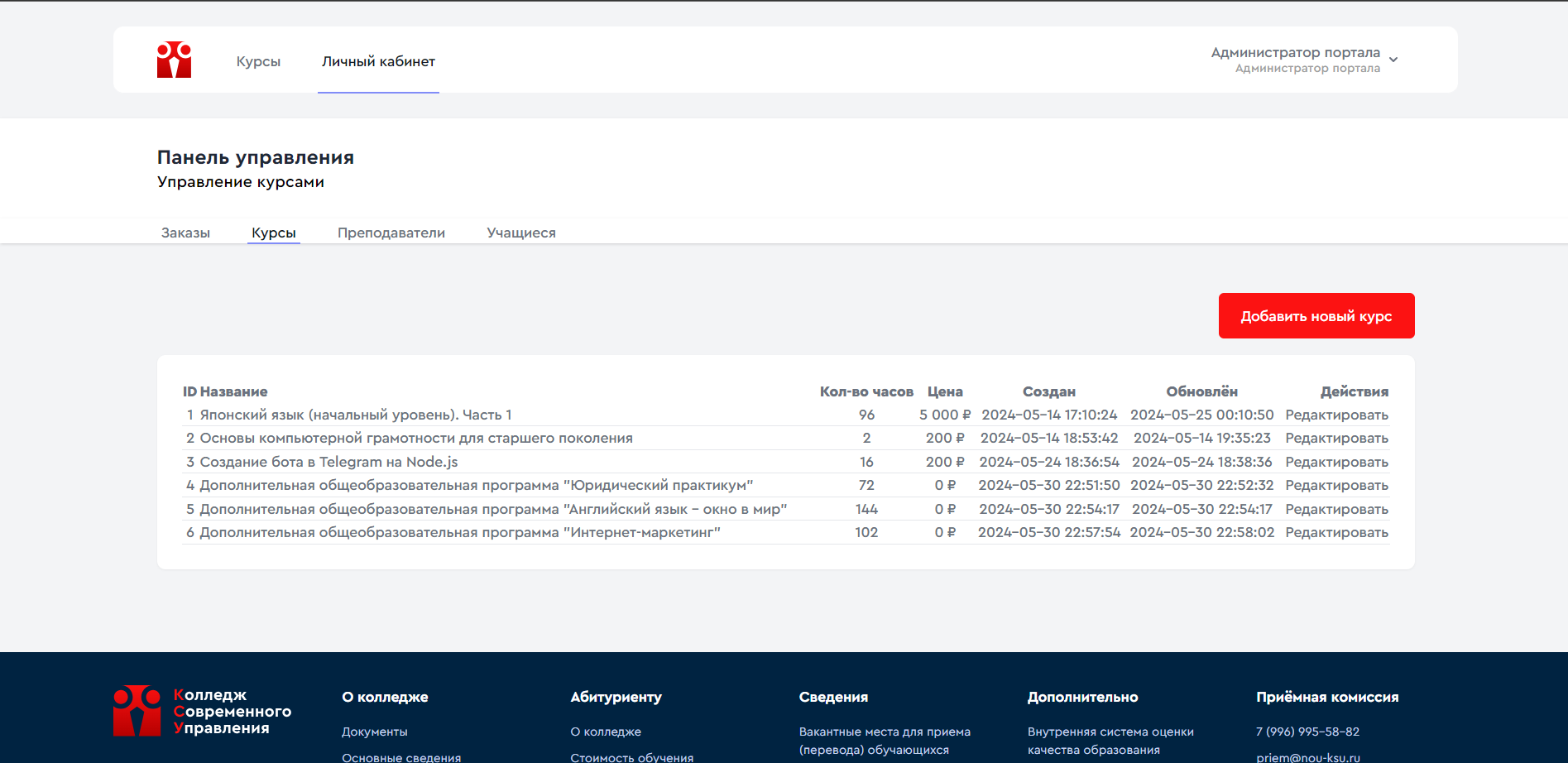


Рисунок 23 – Страница управления курсами из панели администратора

Раздел курсов в панели администратора содержит общий список всех добавленных в систему учебных материалов, с указанием информации о том, когда сам курс был создан и отредактирован.

Отдельно предусмотрено добавление нового курса по нажатии кнопки «Добавить новый курс». При этом в таблице напротив каждого добавленного в систему материала есть ссылка «Редактировать», которая ведёт на страницу редактирования курса (рис. 24).

Код административной страницы со списком курсов размещён в приложении 6.

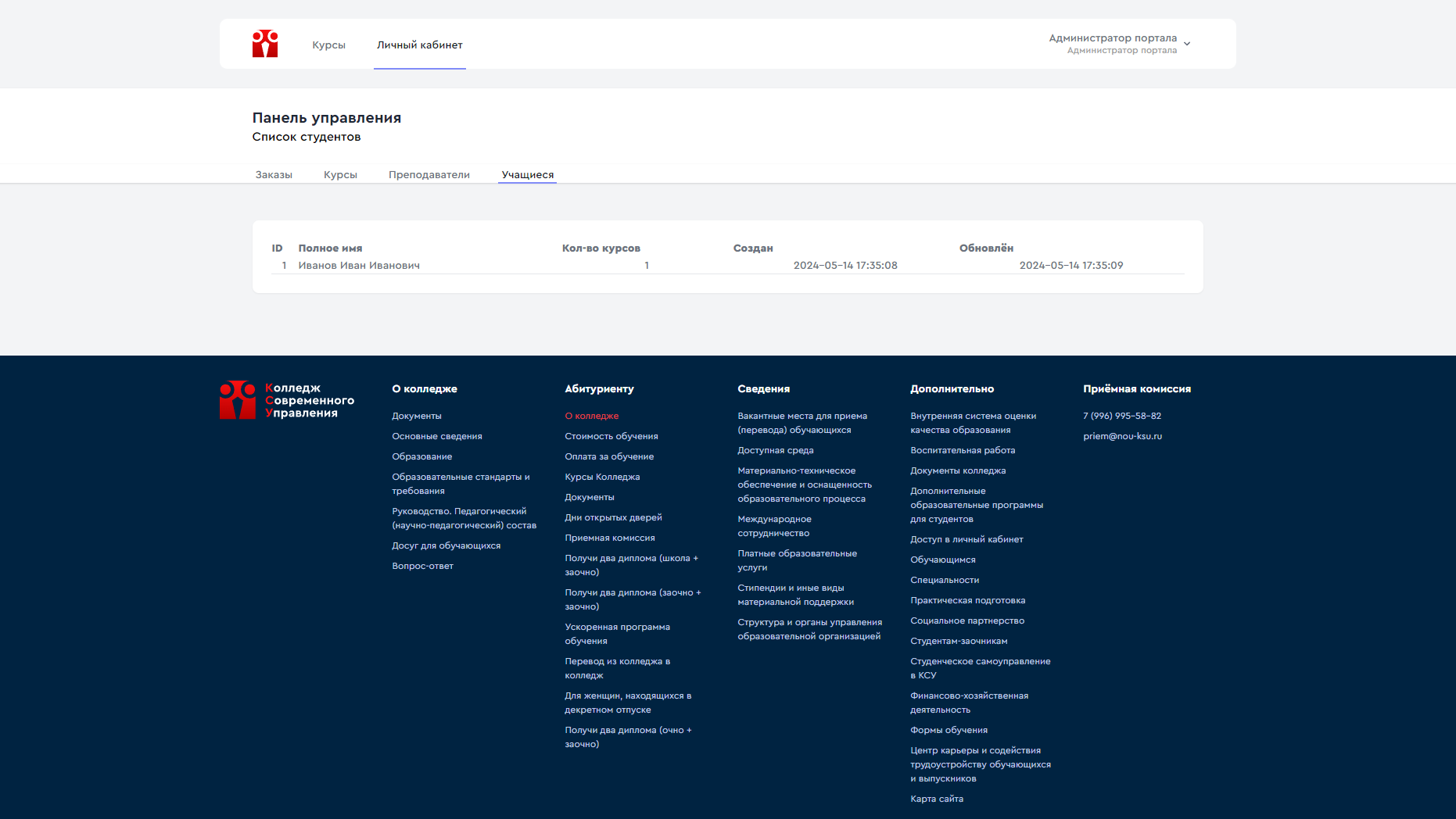


Рисунок 24 – Страница со списком студентов из панели администратора

Внутри раздела панели администратора «Учащиеся» можно посмотреть таблицу со списком всех студентов, зарегистрированных в системе. Каждая из строк данной таблицы содержит следующую информацию о студенте: его полное имя (ФИО), общее количество курсов, которые студент проходит сейчас или проходил ранее, дату создания аккаунта в системе, а также дату последнего обновления данных по нему (рис. 25).

Исходный код данной страницы отдельно размещён в приложении 7.



Рисунок 25 – Страница добавления нового курса из панели администратора

На странице добавления нового курса в панели администратора располагается форма с полями для заполнения основной информации по нему. Администратору предлагается ввести полное название курса, его описание, стоимость в рублях, а также количество часов, на которое рассчитан обучающий материал. Все поля формы являются обязательны, и подлежат заполнению перед тем, как отправить форму по нажатии кнопки «Добавить курс» (рис. 26).

Код данной страницы отдельно указан в приложении 8.

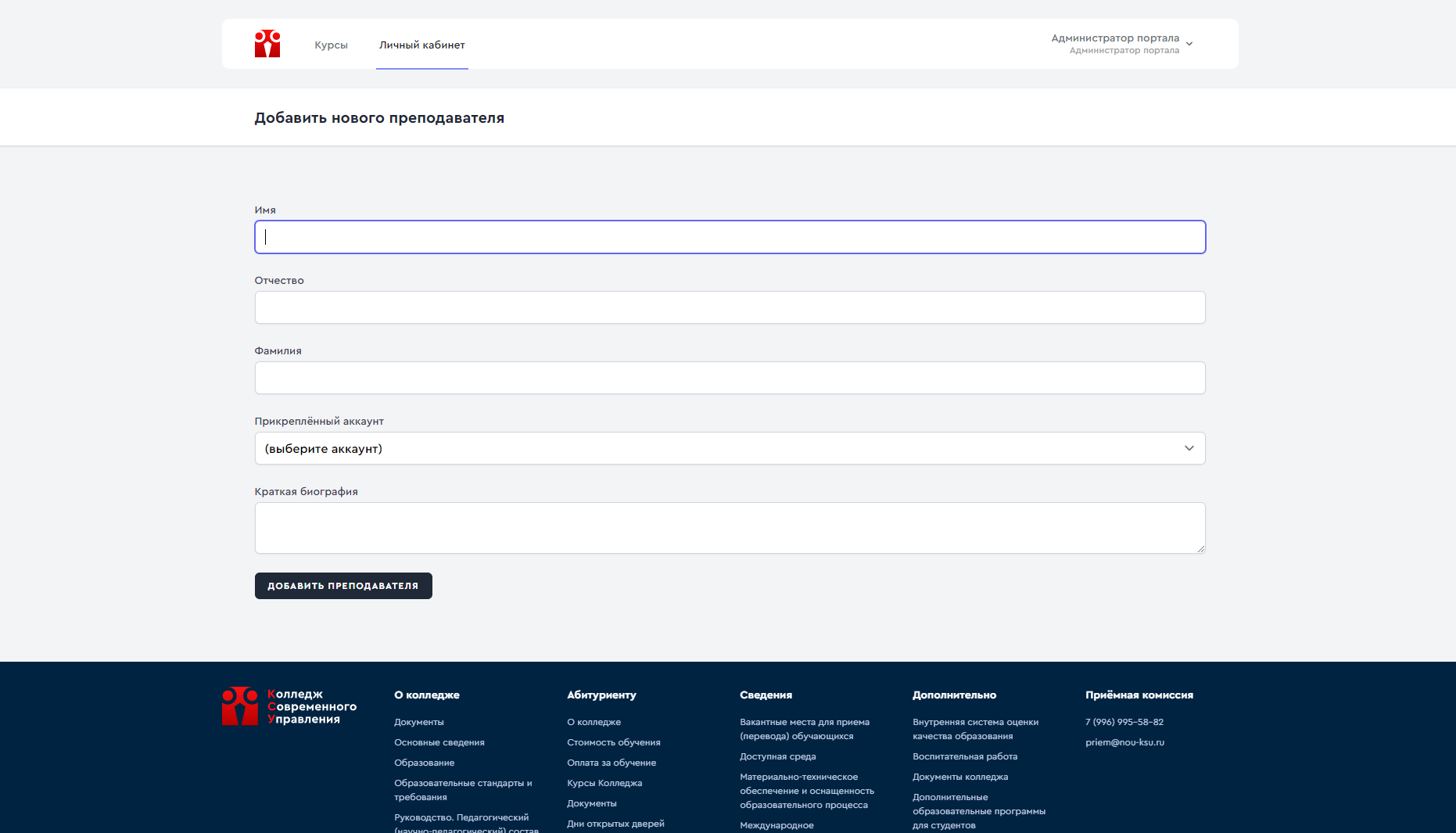


Рисунок 26 – Страница добавления нового преподавателя в систему

При добавлении нового преподавателя в наше веб-приложение администратору системы нужно заполнить его фамилию, имя и отчество, а также указать краткую биографию. Помимо этого, можно прикрепить один из заранее созданных аккаунтов в системе для того, чтобы связать доступы преподавателя с информацией о том, какие курсы он ведёт, и открыть дополнительные права доступа в личный кабинет преподавателя.

Поля «Имя», «Фамилия» и «Краткая биография» являются обязательными. Остальные поля формы заполнять не нужно, поскольку может сложиться ситуация, при которой аккаунта преподавателя ещё не создано, а у самого преподавателя может попросту не быть отчества.

По нажатии кнопки «Добавить преподавателя» отправляется запрос на добавление новой записи в таблицу преподавателей внутри базы данных приложения.

Исходный код страницы размещён в приложении 9.

# 

# 3. Экономическая часть

## 

## 3.1 Цель и задачи экономического раздела

В настоящем разделе выпускной квалификационной работы производится расчет затрат на разработку веб-приложения для организации дополнительного образования в профессиональном образовательном учреждении «Колледж современного управления» города Москвы.

Целью данного раздела является расчет:

* себестоимости проекта;
* определение цены проекта;
* экономической эффективности проекта.

Для подсчета себестоимости и цены проекта, а также экономической эффективности данного программного продукта, нужно знать следующие составляющие:

* расчет затрат на энергоресурсы;
* амортизационные отчисления;
* расчет фонда заработной платы с учетом страховых взносов во внебюджетные фонды;
* прочие (накладные) расходы.

## 

## 3.2 Исходные данные для расчетов

Таблица 1 – Исходные данные

|  |  |  |
| --- | --- | --- |
| **Показатели** | **Единицы измерения** | **Значение** |
| Страховые взносы (от заработной платы в социальные фонды), в том числе: | % | 30 |
| *- отчисления в пенсионный фонд* | *%* | *22* |
| *- отчисления в фонд социального страхования* | *%* | *2,9* |
| *- отчисления в Федеральный и Территориальный фонд обязательного медицинского страхования* | *%* | *5,1* |
| Заработная плата специалиста (техника-программиста) \* | руб./мес. | 70 000 |
| Фонд рабочего времени в 2024 году при 40-часовой рабочей неделе\*\* | дней часов | 248  1 979 |
| Стоимость компьютера | руб. | 60 000 |
| Срок полезного использования компьютера\ноутбука \*\*\* | лет | 3 |
| Амортизационные отчисления | % | 33,33 |
| Потребляемая мощность компьютера (в час) | кВт | 1,30 |
| Тариф на электроэнергию | руб./кВт-час. | 6,43 |
| Затраты на амортизацию\*\*\*\* , ежемесячно (при написании работы учитываю 1 месяц) | руб. | 1 667 |
| Затраты на услуги Интернет-провайдера | руб. | 1 000 |
| Затраты на расходные материалы (см. Таблица 2) | руб. | 2 600 |

*\*Среднее значение заработной платы у техника-программиста необходимо вычислить при помощи анализа заработной платы с нескольких предприятий по г. Москва (Московская область), обязательно делайте ссылку на источник.*

*\*\*Всего в 2024 году будет 366 дней, из которых 248 рабочих и 118 выходных. Рабочее время при 40-часовой неделе составит 1 979 часов*

*\*\*\* Ноутбуки в бухгалтерском учете относятся к 2 амортизационной группе основных средств. Объекты, включенные в эту группу, имеют срок полезного использования 2-3 года. Но предприятие вправе само установить срок, он может быть больше или меньше официального. Ноутбук можно отнести к группе «Компьютеры и периферийное оборудование». При написании ВКР я беру 3 года.*

*\*\*\*\*Затраты=Стоимость OC\*Норму амортизации/12 месяцев*

Таблица 2 – Затраты на расходные материалы

|  |  |  |
| --- | --- | --- |
| **Задача** | **Затраты** | **Ресурс** |
| Анализ и выбор ПО | 1 000 |  |
| Запуск программы | 1 600 |  |
| **Итого на создание программы** | **2 600** |  |

Для расчета стоимости проделанной работы потребуется узнать:

* Среднюю заработную плату программиста на предприятии;
* Затраты на оборудование (амортизация);
* Затраты на электроэнергию.

Среднее значение заработной платы у техника-программиста вычисляем при помощи анализа заработной платы с нескольких предприятий по г. Москва:

Заработная плата в размере от 50 000 руб.

* <https://hh.ru/vacancy/96283720>

Заработная плата в размере от 60 000 руб.

* <https://hh.ru/vacancy/95841172>

Заработная плата в размере от 120 000 руб.

* <https://hh.ru/vacancy/95508572>

Заработная плата в размере до 100 000 руб.

* <https://hh.ru/vacancy/81566025>

Заработная плата в размере от 40 000 руб.

* <https://hh.ru/vacancy/95210349>

Заработная плата в размере от 60 000 руб.

* <https://hh.ru/vacancy/95138823>

Заработная плата в размере от 60 000 руб.

* <https://hh.ru/vacancy/90024969>

Высчитываем среднее значение:

(50 000 + 60 000 + 120 000 + 100 000 + 40 000 +60 000 + 60 000) / 7 = 70 000 рублей, средняя зарплата у техника-программиста.

Вычисляем почасовую оплату. Для этого нужно разделить среднюю заработную плату за год на фонд рабочего времени в 2024 году:

70 000 *х 12 месяцев / фонд рабочего времени= 70 000 х 12 /1979 = 424,46 рублей (среднегодовая почасовая оплата)*

На создание предоставленной в выпускной квалификационной работе программы по разработке веб-приложения для организации дополнительного образования в профессиональном образовательном учреждении «Колледж современного управления» города Москвы в общем объеме ушло 80 часов.

424,46 \* 80 = 33 957 рублей оплата всех часов работы.

Общие затраты на оплаты труда составили:

Заработная плата + Страховые взносы во внебюджетные фонды (30 %)

33 957 + 33 957 \* 30% = 44 144,1 рублей

Затраты на электроэнергию вычислим, узнав тариф на электроэнергию по информации сайта ПАО Мосэнергосбыт.

Рассчитаем стоимость.

Тариф на 1 кВт/ч электроэнергии в 2024 году составляет 6,43 рублей.

Источник: mosenergosbyt.ru/individuals/tariffs-n-payments/tariffs-msk/

Примерное использование кВт в час моего компьютера составляет 1,50. Получается на электроэнергию у нас уходит: 1,50 \* 6,43 = 9.64 рублей (в час). Было затрачено 80 часов времени.

9.64 \* 80 = 771,20 рублей – затраты на электроэнергию.

Таким образом, себестоимость разработки веб-приложения для организации дополнительного образования в профессиональном образовательном учреждении «Колледж современного управления» города Москвы составила 50 182,3 рублей.

Таблица 3 – Расчет себестоимости и цены программного продукта

|  |  |  |
| --- | --- | --- |
| **Показатели** | **Условное обозначение** | **Значение** |
| 1. Прямые расходы, в том числе | | |
| *1.1 Затраты на оплату труда* | *ЗП* | *33 957* |
| *1.2 Страховые взносы во внебюджетные фонды* | *СтрВзн* | *10 187,1* |
| *1.3. Затраты на текущий и профилактический ремонт (Амортизационные отчисления (60 000\*33,33%/12) + услуги Интернет-провайдера)* | *Зпр.р* | *1 667*  *1 000* |
| *1.5. Затраты на материалы, необходимые для эксплуатации* | *Зматер* | *2 600* |
| *1.6. Затраты на электроэнергию* | *Зэл* | *771,2* |
| 2. Итого затраты на разработку программного продукта (полная себестоимость) | ОбщЗ | 50 182,3 |
| 3. Плановая прибыль (20% закладываем в среднем прибыль по предприятию) | П | 300 000 / год |
| 4. Цена РАЗРАБОТЧИКА | Ц | 60 000 |
| 5. НДС (20% от цены) | НДС | 12 000 |
| 6. Цена продажная в розницу с НДС | РознЦ | 72 000 |

## 3.3 Экономический эффект и эффективность

Стоимость поставки продукта (72 000 рублей) для конечного пользователя сопоставима с предложениями других поставщиков.

Ввиду проектно-исследовательского характера работы и отсутствия конкретных объектов внедрения приведены общие данные экономического эффекта внедрения веб-приложения для организации дополнительного образования в профессиональном образовательном учреждении «Колледж современного управления» города Москвы.

Ожидаемый экономический эффект внедрения сайта:

1. Повышение узнаваемости Колледжа, благодаря расширению базы потенциальных абитуриентов через разработанный веб-сайт и, как следствие, увеличение прибыли.

2. Непрерывное повышение профессиональных знаний, в том числе педагогов и студентов колледжа, благодаря наличию собственной платформы для обучения:

3. Совершенствование деловой и профессиональной квалификации.

4. Получение образования нового уровня.

5. Развитие Колледжа в новом направлении обучения.

# 

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы было разработано веб-приложение для осуществления дополнительного образования в образовательном учреждении НОУ «Колледж современного управления». Были получены теоретические знания в области веб-разработки и решены все поставленные задачи.

Полученные знания были применены на практике. Данный сайт доступен на любых компьютерах и мобильных устройствах с выходом в Интернет. На момент написания данной выпускной работы веб-приложение находится в режиме Альфа-тестирования, предстоит добавление новых курсов и доработки технической части. После чего планируется внедрение в систему обучения колледжа.

В настоящее время доступно добавление и редактирование учебных курсов администратором портала, регистрация студентов и запись их на курсы, наполнение курсов учебными материалами.

Функционал разработанного веб-приложения может быть расширен в дальнейшем добавлением новых функций: наличием встроенного электронного журнала, создания курсов преподавателями и студентами, связь между преподавателями и обучающимися непосредственно внутри данного сайта и других.

Создание и развитие данного веб-приложения позволит повысить узнаваемость колледжа, а также способствовать получению образования нового уровня. С экономической точки зрения, данный программный продукт в перспективе может принести заказчику дополнительную прибыль от продажи курсов всем желающим получить образование дистанционным методом.

В данный момент веб-приложение проходит стадию тестирования и размещено на сайте ksu-learning.ru, после его завершения и приема веб-приложения заказчиком оно будет размещаться по адресу до.ноу-ксу.рф.

# 

# СПИСОК ЛИТЕРАТУРЫ

1. Гома, Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений. М.: ДМК Пресс, 2011. –704 с.
2. Леоненков А.В. Нотация и семантика языка UML. - 2-е изд.: Национальный Открытый Университет «ИНТУИТ» Москва, 2016 –205 с.
3. Брюс А. Тейт Семь языков за семь недель. Практическое руководство по изучению языков программирования. — ДМК-Пресс, 2020. — 386 с.
4. Горнаков С. Осваиваем популярные системы управления сайтом (CMS). – ДМК Пресс, 2018 — 331 с.
5. Дронов В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. — БХВ-Петербург, 2020. — 408 с.
6. Клименко Р. Web-мастеринг на 100%. — Питер, 2020. — 560 с.
7. Колисниченко Д. Выбираем лучший бесплатный движок для сайта. CMS Joomla! и Drupal. — БХВ-Петербург, 2020. — 289 c.
8. Купер Н. Как создать сайт. Комикс-путеводитель по HTML, CSS и WordPress. — Манн, Иванов и Фербер (МИФ), 2019. — 266 с.
9. vc.ru – топ 20 конструкторов сайтов. [Электронный ресурс] – Режим доступа – URL: https://vc.ru/services/681602-top-20-konstruktorov-saytov-2023-goda-kak-vybrat-luchshiy-konstruktor-sayta – Дата обращения 15.01.2024.
10. html5book.ru – Основы html. [Электронный ресурс] – Режим доступа – URL: https://html5book.ru/osnovy-html/ – Дата обращения 17.01.2024.
11. intuit.ru – Основы языка HTML [Электронный ресурс] – Режим доступа – URL: https://intuit.ru/studies/courses/103/103/lecture/27971 – Дата обращения 17.01.2024.
12. seodon.ru – Приоритеты стилей в CSS [Электронный ресурс] – Режим доступа – URL: https://seodon.ru/css/prioritety-stilej.php – Дата обращения 18.01.2024.
13. HTML Academy. // Введение в Chrome DevTools. Панель Elements. [Электронный ресурс]: 2020. – URL: [https://htmlacademy.ru/blog/boost/ tools/chrome-devtools-1](https://htmlacademy.ru/blog/boost/%20tools/chrome-devtools-1) – Дата обращения 05.04.2024.
14. JetBrains. // PhpStorm. Возможности. [Электронный ресурс]: 2021. – URL: <https://www.jetbrains.com/ru-ru/phpstorm/features/> – Дата обращения 12.04.2024.
15. JOOMDom. // Дизайн – комбинирование цветов при создании лендинга. [Электронный ресурс]: 2016. – URL: <http://joomdom.com/1937-dizayn-kombinirovanie-cvetov-pri-sozdanii-lendinga.html> – Дата обращения 12.04.2024.
16. Logaster. // Что такое брендбук компании и как его создать: советы и примеры. [Электронный ресурс]: 2020. – URL: https://www.logaster.ru/blog/create-brand-book/ – Дата обращения 11.03.2024.
17. Photo & Vector Stocker. // Adobe Photoshop — самый популярный графический редактор. [Электронный ресурс]: 2013. – URL: <https://pvstoker.com/adobe-photoshop/> – Дата обращения 19.04.2024.
18. Skillbox. // Что такое Figma: возможности и принципы работы. [Электронный ресурс]: 2019. – URL : <https://pvstoker.com/adobe-photoshop/> – Дата обращения 19.04.2024.
19. TimeWeb. // Плюсы и минусы Bootstrap. [Электронный ресурс]: 2018. – URL: https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1 – Дата обращения 11.03.2024.
20. UMI.CMS. // CMS для сайта vs конструктор сайтов. Что выбрать и в чём отличия. [Электронный ресурс]: 2021. – URL: <https://www.umi-cms.ru/support/poleznye-stati/cms-saita-vs-konstruktor/> – Дата обращения 19.04.2024.
21. АО «Мосэнергосбыт» // Тарифы на электрическую энергию для населения и приравненных к нему категорий потребителей на территории г. Москвы на 2024 год. [Электронный ресурс]: 2024. – URL: <https://www.mosenergosbyt.ru/individuals/tariffs-n-payments/tariffs-msk/> – Дата обращения 15.04.2024.

# ПРИЛОЖЕНИЕ 1

**Основной шаблон сайта**

<!DOCTYPE html>

<html lang="{{ str\_replace('\_', '-', app()->getLocale()) }}">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="csrf-token" content="{{ csrf\_token() }}">

<title>{{ config('app.name', 'Laravel') }}</title>

<!-- Scripts -->

@vite(['resources/css/app.css', 'resources/js/app.js'])

</head>

<body class="font-sans bg-gray-100 antialiased">

<div class="min-h-screen">

@include('layouts.navigation')

<!-- Page Heading -->

@if (isset($header))

<header class="bg-white shadow">

<div class="max-w-7xl mx-auto py-6 px-4 sm:px-6 lg:px-8">

{{ $header }}

</div>

</header>

@endif

<!-- Page Content -->

<main>

{{ $slot }}

</main>

<!-- Page Footer -->

<footer>

<div class="container mx-auto">

<div class="flex flex-row flex-wrap">

<div class="flex-column">

<div class="footer-logo">

<svg width="173" height="50" viewBox="0 0 173 50" fill="none" xmlns="http://www.w3.org/2000/svg">

</g>

<defs>

<linearGradient id="paint0\_linear\_43\_69" x1="22.8686" y1="0" x2="22.8686" y2="49.2555" gradientUnits="userSpaceOnUse">

<stop stop-color="#FC1111"/>

<stop offset="1" stop-color="#B50000"/>

</linearGradient>

<clipPath id="clip0\_43\_69">

<rect width="173" height="50" fill="white"/>

</clipPath>

</defs>

</svg>

</div>

</div>

<div class="flex-column">

<div class="footer-menu">

<h4>О колледже</h4>

<ul class=""><li id="menu-item-500" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/dokumenty/">Документы</a></li>

<li id="menu-item-501" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/">Основные сведения</a></li>

<li id="menu-item-502" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/obrazovanie/">Образование</a></li>

<li id="menu-item-503" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/obrazovatelnye-standarty-i-trebovaniya/">Образовательные стандарты и требования</a></li>

<li id="menu-item-504" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/rukovodstvo-pedagogicheskij-nauchno-pedagogicheskij-sostav/">Руководство. Педагогический (научно-педагогический) состав</a></li>

<li id="menu-item-3372" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/dosug-dlya-obuchajushhihsya-i-ih-roditelej-zakonnyh-predstavitelej/">Досуг для обучающихся</a></li>

<li id="menu-item-4372" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/faq/">Вопрос-ответ</a></li>

</ul>

</div>

</div>

<div class="flex-column">

<div class="footer-menu">

<h4>Абитуриенту</h4>

<ul class=""><li id="menu-item-505" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/o-kolledzhe/">О колледже</a></li>

<li id="menu-item-511" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/stoimost-obucheniya/">Стоимость обучения</a></li>

<li id="menu-item-509" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/oplata-za-obuchenie/">Оплата за обучение</a></li>

<li id="menu-item-508" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/kursy-kolledzha/">Курсы Колледжа</a></li>

<li id="menu-item-507" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/dokumenty/">Документы</a></li>

<li id="menu-item-506" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/dni-otkrytyh-dverej/">Дни открытых дверей</a></li>

<li id="menu-item-510" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/priemnaya-komissiya/">Приемная комиссия</a></li>

<li id="menu-item-2899" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/priemnaya-komissiya/poluchi-dva-diploma-shkola-zaochno/">Получи два диплома (школа + заочно)</a></li>

<li id="menu-item-2898" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/priemnaya-komissiya/poluchi-dva-diploma-zaochno-zaochno/">Получи два диплома (заочно + заочно)</a></li>

<li id="menu-item-2895" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/priemnaya-komissiya/uskorennaya-programma-obucheniya/">Ускоренная программа обучения</a></li>

<li id="menu-item-2896" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/priemnaya-komissiya/perevod-iz-kolledzha-v-kolledzh/">Перевод из колледжа в колледж</a></li>

<li id="menu-item-2894" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/priemnaya-komissiya/dlya-zhenshhin-nahodyashhihsya-v-dekretnom-otpuske/">Для женщин, находящихся в декретном отпуске</a></li>

<li id="menu-item-2897" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/priemnaya-komissiya/poluchi-dva-diploma-ochno-zaochno/">Получи два диплома (очно + заочно)</a></li>

</ul>

</div>

</div>

<div class="flex-column">

<div class="footer-menu">

<h4>Сведения</h4>

<ul class=""><li id="menu-item-512" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/vakantnye-mesta-dlya-priema-perevoda-obuchajushhihsya/">Вакантные места для приема (перевода) обучающихся</a></li>

<li id="menu-item-513" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/dostupnaya-sreda/">Доступная среда</a></li>

<li id="menu-item-514" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/materialno-tehnicheskoe-obespechenie-i-osnashhennost-obrazovatelnogo-processa/">Материально-техническое обеспечение и оснащенность образовательного процесса</a></li>

<li id="menu-item-515" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/mezhdunarodnye-otnoshenie/">Международное сотрудничество</a></li>

<li id="menu-item-518" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/platnye-obrazovatelnye-uslugi/">Платные образовательные услуги</a></li>

<li id="menu-item-520" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/stipendii-i-inye-vidy-materialnoj-podderzhki/">Стипендии и иные виды материальной поддержки</a></li>

<li id="menu-item-521" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/osnovnye-svedeniya/struktura-i-organy-upravleniya-obrazovatelnoj-organizaciej/">Структура и органы управления образовательной организацией</a></li>

</ul>

</div>

</div>

<div class="flex-column">

<div class="footer-menu">

<h4>Дополнительно</h4>

<ul class=""><li id="menu-item-522" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/vnutrennyaya-sistema-ocenki-kachestva-obrazovaniya/">Внутренняя система оценки качества образования</a></li>

<li id="menu-item-523" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/vospitatelnaya-rabota/">Воспитательная работа</a></li>

<li id="menu-item-526" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/dokumenty-kolledzha/">Документы колледжа</a></li>

<li id="menu-item-527" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/dopolnitelnye-obrazovatelnye-programmy-dlya-studentov/">Дополнительные образовательные программы для студентов</a></li>

<li id="menu-item-528" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/dostup-v-lichnyj-kabinet/">Доступ в личный кабинет</a></li>

<li id="menu-item-534" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/obuchajushhimsya/">Обучающимся</a></li>

<li id="menu-item-587" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/specialties/">Специальности</a></li>

<li id="menu-item-536" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/prakticheskaya-podgotovka/">Практическая подготовка</a></li>

<li id="menu-item-538" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/socialnoe-partnerstvo/">Социальное партнерство</a></li>

<li id="menu-item-541" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/studentam-zaochnikam/">Студентам-заочникам</a></li>

<li id="menu-item-542" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/students/">Студенческое самоуправление в КСУ</a></li>

<li id="menu-item-543" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/finansovo-hozyajstvennaya-deyatelnost/">Финансово-хозяйственная деятельность</a></li>

<li id="menu-item-544" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/formy-obucheniya/">Формы обучения</a></li>

<li id="menu-item-545" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/centr-karery-i-sodejstviya-trudoustrojstvu-obuchajushhihsya-i-vypusknikov/">Центр карьеры и содействия трудоустройству обучающихся и выпускников</a></li>

<li id="menu-item-4094" class="menu-item"><a href="https://xn----ttbherjd.xn--p1ai/site-map/">Карта сайта</a></li>

</ul>

</div>

</div>

<div class="flex-column">

<div class="footer-menu">

<h4>Приёмная комиссия</h4>

<ul>

<li class="menu-item"><a href="tel:79969955882">7 (996) 995-58-82</a></li>

<!-- <li><a href="tel:79691363077">7 (969) 136-30-77</a></li> -->

<li class="menu-item"><a href="mailto:priem@nou-ksu.ru">priem@nou-ksu.ru</a></li>

</ul>

</div>

</div>

</div>

</div>

<div class="container mx-auto mt-2 p-2">

<div class="text-center">

© {{ date('Y') }} ноу-ксу.рф. Все права защищены.

</div>

</div>

</footer>

</div>

</body>

</html>

**Приложение 2**

**Главная страница**

<x-app-layout>

<div class="container">

<section class="greetings flex">

<div>

<h1 class="greetings\_\_header">Перед вами открыты новые возможности!</h1>

<p>На сайте доступны курсы к прохождению по разным направлениям. Выберите тот, что подходит именно вам, оплатите его, и начните обучение!</p>

<a href="{{ route('login') }}" class="button button\_primary mt-4">Войти в личный кабинет</a>

</div>

<div>

<img src="/images/open-book.svg"/>

</div>

</section>

<section class="courses\_\_main">

<h2 class="courses\_\_main\_header">Последние опубликованные курсы</h2>

<div class="courses\_\_main\_\_last flex">

@foreach($lastCourses as $lastCourse)

<div class="courses\_\_main\_\_item">

<div class="courses\_\_main\_\_item-icon"><svg width="40" height="41" viewBox="0 0 40 41" fill="none" xmlns="http://www.w3.org/2000/svg">

<g clip-path="url(#clip0\_43\_176)">

<g clip-path="url(#clip1\_43\_176)">

<path d="M13.3335 22.0565H17.5002" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M13.3335 15.39H24.1668" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M13.3335 28.7235H15.8335" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M33.0474 5.67613C31.0948 3.72351 27.9521 3.72351 21.6667 3.72351H18.3333C12.0479 3.72351 8.90524 3.72351 6.95262 5.67613C5 7.62875 5 10.7714 5 17.0568V23.7235C5 30.0089 5 33.1516 6.95262 35.1042C8.90524 37.0568 12.0479 37.0568 18.3333 37.0568H21.6667C27.9521 37.0568 31.0948 37.0568 33.0474 35.1042C34.6192 33.5324 34.9257 31.1894 34.9855 27.0568" stroke="white" stroke-width="2" stroke-linecap="round"/>

</g>

</g>

<defs>

<clipPath id="clip0\_43\_176">

<rect width="40" height="40" fill="white" transform="translate(0 0.390015)"/>

</clipPath>

<clipPath id="clip1\_43\_176">

<rect width="40" height="40" fill="white" transform="translate(0 0.390015)"/>

</clipPath>

</defs>

</svg>

</div>

<div class="courses\_\_main\_\_item-title">{{ $lastCourse->name }}</div>

<div class="courses\_\_main\_\_item-description">{{ $lastCourse->description }}</div>

<div class="courses\_\_main\_\_item-info">

<div class="courses\_\_main\_\_item-info\_\_price">

<div>Стоимость обучения</div>

<div>{{ number\_format($lastCourse->price, 0, '.', ' ') }} ₽</div>

</div>

<div class="courses\_\_main\_\_item-info\_\_price">

<div>Время прохождения</div>

<div>{{ number\_format($lastCourse->total\_hours, 0, '.', ' ') }} ч.</div>

</div>

</div>

<div class="courses\_\_main\_\_item-buttons">

<a href="{{ route('course.detail', ['id' => $lastCourse->id]) }}" class="button button\_primary-outline">Подробнее о курсе</a>

<a href="#" class="button button\_primary">Приобрести курс</a>

</div>

</div>

@endforeach

</div>

<div class="mt-4 text-center">

<a href="{{ route('courses') }}" class="button button\_primary">Полный список курсов</a>

</div>

</section>

</div>

</x-app-layout>

**Приложение 3**

**Страница авторизации**

<x-guest-layout>

<!-- Session Status -->

<x-auth-session-status class="mb-4" :status="session('status')" />

<form method="POST" action="{{ route('login') }}">

@csrf

<!-- Email Address -->

<div>

<x-input-label for="email" :value="'Электронная почта'" />

<x-text-input id="email" class="block mt-1 w-full" type="email" name="email" :value="old('email')" required autofocus autocomplete="username" />

<x-input-error :messages="$errors->get('email')" class="mt-2" />

</div>

<!-- Password -->

<div class="mt-4">

<x-input-label for="password" :value="'Пароль'" />

<x-text-input id="password" class="block mt-1 w-full"

type="password"

name="password"

required autocomplete="current-password" />

<x-input-error:messages="$errors->get('password')" class="mt-2" />

</div>

<!-- Remember Me -->

<div class="block mt-4">

<label for="remember\_me" class="inline-flex items-center">

<input id="remember\_me" type="checkbox" class="rounded border-gray-300 text-indigo-600 shadow-sm focus:ring-indigo-500" name="remember">

<span class="ms-2 text-sm text-gray-600">{{ 'Запомнить меня?' }}</span>

</label>

</div>

<div class="flex items-center justify-end mt-4">

@if (Route::has('password.request'))

<a class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500" href="{{ route('password.request') }}">

{{ 'Забыли пароль?' }}

</a>

@endif

<x-primary-button class="ms-3">

{{ 'Войти' }}

</x-primary-button>

</div>

</form>

</x-guest-layout>

**Приложение 4**

**Список доступных курсов**

<x-app-layout>

<div class="container">

<h1 class="page\_\_header">Список доступных курсов</h1>

<section class="courses\_\_main">

<div class="courses\_\_main\_\_last flex">

@foreach($allCourses as $oneCourse)

<div class="courses\_\_main\_\_item">

<div class="courses\_\_main\_\_item-icon"><svg width="40" height="41" viewBox="0 0 40 41" fill="none" xmlns="http://www.w3.org/2000/svg">

<g clip-path="url(#clip0\_43\_176)">

<g clip-path="url(#clip1\_43\_176)">

<path d="M30.3001 13.7887L31.0725 13.0163C32.3522 11.7366 34.4271 11.7366 35.7069 13.0163C36.9866 14.2961 36.9866 16.371 35.7069 17.6507L34.9345 18.4231M30.3001 13.7887C30.3001 13.7887 30.3966 15.4301 31.8449 16.8783C33.2931 18.3266 34.9345 18.4231 34.9345 18.4231M30.3001 13.7887L23.199 20.8898C22.7181 21.3707 22.4776 21.6112 22.2707 21.8764C22.0268 22.1892 21.8176 22.5276 21.647 22.8857C21.5023 23.1893 21.3947 23.5119 21.1796 24.1572L20.491 26.2232L20.2682 26.8916M34.9345 18.4231L27.8334 25.5242C27.3525 26.0051 27.112 26.2456 26.8468 26.4524C26.534 26.6964 26.1956 26.9056 25.8375 27.0762C25.5339 27.2209 25.2113 27.3284 24.566 27.5435L22.5 28.2322L21.8316 28.455M20.2682 26.8916L20.0454 27.56C19.9395 27.8775 20.0222 28.2276 20.2589 28.4643C20.4955 28.701 20.8456 28.7836 21.1632 28.6778L21.8316 28.455M20.2682 26.8916L21.8316 28.455" stroke="white" stroke-width="2"/>

<path d="M13.3335 22.0565H17.5002" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M13.3335 15.39H24.1668" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M13.3335 28.7235H15.8335" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M33.0474 5.67613C31.0948 3.72351 27.9521 3.72351 21.6667 3.72351H18.3333C12.0479 3.72351 8.90524 3.72351 6.95262 5.67613C5 7.62875 5 10.7714 5 17.0568V23.7235C5 30.0089 5 33.1516 6.95262 35.1042C8.90524 37.0568 12.0479 37.0568 18.3333 37.0568H21.6667C27.9521 37.0568 31.0948 37.0568 33.0474 35.1042C34.6192 33.5324 34.9257 31.1894 34.9855 27.0568" stroke="white" stroke-width="2" stroke-linecap="round"/>

</g>

</g>

<defs>

<clipPath id="clip0\_43\_176">

<rect width="40" height="40" fill="white" transform="translate(0 0.390015)"/>

</clipPath>

<clipPath id="clip1\_43\_176">

<rect width="40" height="40" fill="white" transform="translate(0 0.390015)"/>

</clipPath>

</defs>

</svg>

</div>

<div class="courses\_\_main\_\_item-title">{{ $oneCourse->name }}</div>

<div class="courses\_\_main\_\_item-description">{{ $oneCourse->description }}</div>

<div class="courses\_\_main\_\_item-info">

<div class="courses\_\_main\_\_item-info\_\_price">

<div>Стоимость обучения</div>

<div>{{ number\_format($oneCourse->price, 0, '.', ' ') }} ₽</div>

</div>

<div class="courses\_\_main\_\_item-info\_\_price">

<div>Время прохождения</div>

<div>{{ number\_format($oneCourse->total\_hours, 0, '.', ' ') }} ч.</div>

</div>

</div>

<div class="courses\_\_main\_\_item-buttons">

<a href="{{ route('course.detail', ['id' => $oneCourse->id]) }}" class="button button\_primary-outline">Подробнее о курсе</a>

<a href="#" class="button button\_primary">Приобрести курс</a>

</div>

</div>

@endforeach

</div>

</section>

</div>

</x-app-layout>

**Приложение 5**

**Страница отдельного курса**

<x-app-layout>

<div class="container">

<h1 class="page\_\_header">Курс «{{ $chosenCourse->name }}»</h1>

<div class="flex page\_\_structure">

<div class="page\_\_content">

{{ $chosenCourse->description }}

@if (count($chosenCourse->teachers))

@foreach($chosenCourse->teachers as $courseTeacher)

<div class="course\_\_teacher">

<div class="course\_\_teacher\_icon">

<svg width="40" height="40" viewBox="0 0 40 40" fill="none" xmlns="http://www.w3.org/2000/svg">

<g clip-path="url(#clip0\_119\_199)">

<g clip-path="url(#clip1\_119\_199)">

<path d="M13.3335 21.6665H17.5002" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M13.3335 15H24.1668" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M13.3335 28.3335H15.8335" stroke="white" stroke-width="2" stroke-linecap="round"/>

<path d="M33.0474 5.28612C31.0948 3.3335 27.9521 3.3335 21.6667 3.3335H18.3333C12.0479 3.3335 8.90524 3.3335 6.95262 5.28612C5 7.23874 5 10.3814 5 16.6668V23.3335C5 29.6189 5 32.7616 6.95262 34.7142C8.90524 36.6668 12.0479 36.6668 18.3333 36.6668H21.6667C27.9521 36.6668 31.0948 36.6668 33.0474 34.7142C34.6192 33.1424 34.9257 30.7994 34.9855 26.6668" stroke="white" stroke-width="2" stroke-linecap="round"/>

</g>

</g>

<defs>

<clipPath id="clip0\_119\_199">

<rect width="40" height="40" fill="white"/>

</clipPath>

<clipPath id="clip1\_119\_199">

<rect width="40" height="40" fill="white"/>

</clipPath>

</defs>

</svg>

</div>

<div class="course\_\_teacher\_info">

<div class="course\_\_teacher\_\_name">{{ $courseTeacher->fullName() }}</div>

<div class="course\_\_teacher\_\_title">Преподаватель курса. {{ $courseTeacher->bio }}</div>

</div>

</div>

@endforeach

@endif

@if (count($chosenCourse->materials))

<h2 class="course\_\_content\_\_title">Содержание курса:</h2>

@foreach($chosenCourse->materials as $courseMaterial)

<section class="course\_\_section">

<div class="course\_\_section\_header">

<span class="course\_\_section\_\_number">{{ $courseMaterial->order\_num }}</span>

<span class="course\_\_section\_\_title">{{ $courseMaterial->title }}</span>

</div>

@if ($courseMaterial->is\_section)

<div class="courses\_\_subsections">

@foreach($courseMaterial->childs as $courseMaterialChild)

<div class="course\_\_subsection"><span class="course\_\_subsection\_\_number">{{ $courseMaterialChild->order\_num }}.</span>{{ $courseMaterialChild->title }}</div>

@endforeach

</div>

@endif

</section>

@endforeach

@endif

</div>

<div class="page\_\_widgets">

<div class="page\_\_widget\_button">

<a href="#" class="button button\_primary">Приобрести курс</a>

</div>

<div class="page\_\_widget">

<div class="page\_\_widget-title">Стоимость курса</div>

<div class="page\_\_widget-content">{{ number\_format($chosenCourse->price, 0, '.', ' ') }} ₽</div>

</div>

<div class="page\_\_widget">

<div class="page\_\_widget-title">Время прохождения</div>

<div class="page\_\_widget-content">{{ number\_format($chosenCourse->total\_hours, 0, '.', ' ') }} ч.</div>

</div>

</div>

</div>

</div>

</x-app-layout>

**Приложение 6**

**Список управления курсами в административной части сайта**

<x-app-layout>

<x-slot name="header">

<h2 class="font-semibold text-xl text-gray-800 leading-tight">

Панель управления

</h2>

Управление курсами

</x-slot>

<section class="bg-white shadow">

<div class="flex max-w-7xl mx-auto dashboard\_\_menu">

<div class="hidden space-x-8 sm:-my-px sm:flex">

<x-nav-link :href="route('dashboard')" :active="request()->routeIs('dashboard')">

{{ 'Заказы' }}

</x-nav-link>

<x-nav-link :href="route('dashboard.courses')" :active="request()->routeIs('dashboard.courses')">

{{ 'Курсы' }}

</x-nav-link>

<x-nav-link :href="route('dashboard.teachers')" :active="request()->routeIs('dashboard.teachers')">

{{ 'Преподаватели' }}

</x-nav-link>

<x-nav-link :href="route('dashboard.students')" :active="request()->routeIs('dashboard.students')">

{{ 'Учащиеся' }}

</x-nav-link>

</div>

</div>

</section>

<div class="py-12">

<div class="max-w-7xl mx-auto sm:px-6 lg:px-8">

<div class="flex mb-4 justify-end">

<a href="{{ route('dashboard.courses.new') }}" class="button button\_primary">Добавить новый курс</a>

</div>

@if (session('status'))

<p

x-data="{ show: true }"

x-show="show"

x-transition

x-init="setTimeout(() => show = false, 2000)"

class="text-sm text-gray-600"

>{{ session('status') }}</p>

@endif

<div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">

<div class="p-6 text-gray-900">

<div class="relative overflow-x-auto">

<table class="w-full text-sm text-left rtl:text-right text-gray-500 dark:text-gray-400">

<thead>

<tr>

<th>ID</th>

<th class="sm:text-left">Название</th>

<th class="text-center">Кол-во часов</th>

<th class="text-center">Цена</th>

<th class="text-center">Создан</th>

<th class="text-center">Обновлён</th>

<th class="text-right">Действия</th>

</tr>

</thead>

<tbody>

@forelse($allCourses as $course)

<tr class="bg-white border-b dark:bg-gray-800 dark:border-gray-700">

<td class="text-center">{{ $course->id }}</td>

<td>{{ $course->name }}</td>

<td class="text-center">{{ $course->total\_hours }}</td>

<td class="text-center">{{ number\_format($course->price, 0, '.', ' ') }} ₽</td>

<td class="text-center">{{ $course->created\_at }}</td>

<td class="text-center">{{ $course->updated\_at }}</td>

<td class="text-right"><a href="{{ route('dashboard.courses.edit', ['id' => $course->id]) }}">Редактировать</a></td>

</tr>

@empty

<tr class="bg-white border-b dark:bg-gray-800 dark:border-gray-700">

<td class="text-center" colspan="7">Нет ни одного курса в системе!</td>

</tr>

@endforelse

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</x-app-layout>

**Приложение 7**

**Страница со списком студентов в панели администратора**

<x-app-layout>

<x-slot name="header">

<h2 class="font-semibold text-xl text-gray-800 leading-tight">

Панель управления

</h2>

Список студентов

</x-slot>

<section class="bg-white shadow">

<div class="flex max-w-7xl mx-auto dashboard\_\_menu">

<div class="hidden space-x-8 sm:-my-px sm:flex">

<x-nav-link :href="route('dashboard')" :active="request()->routeIs('dashboard')">

{{ 'Заказы' }}

</x-nav-link>

<x-nav-link :href="route('dashboard.courses')" :active="request()->routeIs('dashboard.courses')">

{{ 'Курсы' }}

</x-nav-link>

<x-nav-link :href="route('dashboard.teachers')" :active="request()->routeIs('dashboard.teachers')">

{{ 'Преподаватели' }}

</x-nav-link>

<x-nav-link :href="route('dashboard.students')" :active="request()->routeIs('dashboard.students')">

{{ 'Учащиеся' }}

</x-nav-link>

</div>

</div>

</section>

<div class="py-12">

<div class="max-w-7xl mx-auto sm:px-6 lg:px-8">

<div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">

<div class="p-6 text-gray-900">

<div class="relative overflow-x-auto">

<table class="w-full text-sm text-left rtl:text-right text-gray-500 dark:text-gray-400">

<thead>

<tr>

<th>ID</th>

<th class="sm:text-left">Полное имя</th>

<th>Кол-во курсов</th>

<th>Создан</th>

<th>Обновлён</th>

</tr>

</thead>

<tbody>

@forelse($allStudents as $student)

<tr class="bg-white border-b dark:bg-gray-800 dark:border-gray-700">

<td class="text-center">{{ $student->id }}</td>

<td>{{ $student->fullName() }}</td>

<td class="text-center">{{ count($student->courses) }}</td>

<td class="text-center">{{ $student->created\_at }}</td>

<td class="text-center">{{ $student->updated\_at }}</td>

</tr>

@empty

<tr class="bg-white border-b dark:bg-gray-800 dark:border-gray-700">

<td class="text-center" colspan="7">Нет ни одного студента в системе!</td>

</tr>

@endforelse

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</x-app-layout>

**Приложение 8**

**Страница добавления нового курса в панели администратора**

<x-app-layout>

<x-slot name="header">

<h2 class="font-semibold text-xl text-gray-800 leading-tight">

Добавить новый курс

</h2>

</x-slot>

<div class="py-12">

<div class="max-w-7xl mx-auto sm:px-6 lg:px-8">

<form method="post" action="{{ route('dashboard.courses.new') }}" class="mt-6 space-y-6">

@csrf

<div>

<x-input-label for="name" :value="'Название курса'" />

<x-text-input id="name" name="name" type="text" class="mt-1 block w-full" :value="old('name')" required autofocus autocomplete="name" />

<x-input-error class="mt-2" :messages="$errors->get('name')" />

</div>

<div>

<x-input-label for="description" :value="'Описание'" />

<x-textarea id="description" name="description" class="mt-1 block w-full" :value="old('description')" required autocomplete="description" />

<x-input-error class="mt-2" :messages="$errors->get('description')" />

</div>

<div>

<x-input-label for="price" :value="'Стоимость, в рублях'" />

<x-text-input id="price" name="price" type="number" class="mt-1 block w-full" :value="old('price')" required autocomplete="price" />

<x-input-error class="mt-2" :messages="$errors->get('price')" />

</div>

<div>

<x-input-label for="total\_hours" :value="'Количество часов, на которое рассчитан курс'" />

<x-text-input id="total\_hours" name="total\_hours" type="number" class="mt-1 block w-full" :value="old('total\_hours')" required autocomplete="total\_hours" />

<x-input-error class="mt-2" :messages="$errors->get('total\_hours')" />

</div>

<div class="flex items-center gap-4">

<x-primary-button>{{ "Добавить курс" }}</x-primary-button>

</div>

</form>

</div>

</div>

</x-app-layout>

**Приложение 9**

Страница добавления нового преподавателя в панели администратора

<x-app-layout>

<x-slot name="header">

<h2 class="font-semibold text-xl text-gray-800 leading-tight">

Добавить нового преподавателя

</h2>

</x-slot>

<div class="py-12">

<div class="max-w-7xl mx-auto sm:px-6 lg:px-8">

<form method="post" action="{{ route('dashboard.teachers.new') }}" class="mt-6 space-y-6">

@csrf

<div>

<x-input-label for="first\_name" :value="'Имя'" />

<x-text-input id="first\_name" name="first\_name" type="text" class="mt-1 block w-full" :value="old('first\_name')" required autofocus autocomplete="first\_name" />

<x-input-error class="mt-2" :messages="$errors->get('first\_name')" />

</div>

<div>

<x-input-label for="middle\_name" :value="'Отчество'" />

<x-text-input id="middle\_name" name="middle\_name" type="text" class="mt-1 block w-full" :value="old('middle\_name')" autofocus autocomplete="middle\_name" />

<x-input-error class="mt-2" :messages="$errors->get('middle\_name')" />

</div>

<div>

<x-input-label for="last\_name" :value="'Фамилия'" />

<x-text-input id="last\_name" name="last\_name" type="text" class="mt-1 block w-full" :value="old('last\_name')" required autofocus autocomplete="last\_name" />

<x-input-error class="mt-2" :messages="$errors->get('last\_name')" />

</div>

<div>

<x-input-label for="user\_id" :value="'Прикреплённый аккаунт'" />

<select id="user\_id" name="user\_id" class="border-gray-300 focus:border-indigo-500 focus:ring-indigo-500 rounded-md shadow-sm mt-1 block w-full" autofocus>

<option value="" disabled selected>(выберите аккаунт)</option>

@foreach($allUsers as $user)

<option value="{{ $user->id }}">{{ $user->name . ' (' . $user->email . ')' }}</option>

@endforeach

</select>

</div>

<div>

<x-input-label for="bio" :value="'Краткая биография'" />

<x-textarea id="bio" name="bio" class="mt-1 block w-full" :value="old('bio')" required autocomplete="bio" />

<x-input-error class="mt-2" :messages="$errors->get('bio')" />

</div>

<div class="flex items-center gap-4">

<x-primary-button>{{ "Добавить преподавателя" }}</x-primary-button>

</div>

</form>

</div>

</div>

</x-app-layout>