**Введение**

На преддипломную практику была поставлена задача: Разработка интернет-ресурса для онлайн-курсов по танцам «Elizabeth Khard». Цель курсового проекта заключается в создании приложения позволяющего просматривать видеоуроки по танцам. Создаваемое приложение будет рассчитано пользователей, которые любят танцы.

Далее приведено краткое описание каждого раздела в пояснительной записке проекта.

Раздел «Анализ задачи» включает в себя более подробное исследование предметной области, а также формулировку требований к разрабатываемому программному продукту. Описываются входные и выходные данные, необходимые для работы приложения, а также постоянная информация, используемая в процессе функционирования системы.

Раздел «Проектирование задачи» подробно описывает процесс проектирования разрабатываемого программного продукта. Здесь представлены диаграммы, концептуальные модели и структуры данных, необходимые для реализации функционала приложения.

Раздел «Реализация» представляет собой описание фактической разработки и реализации проекта. Здесь описываются основные функции и элементы управления, реализованные в рамках приложения. В данном разделе представлены технические детали разработки, используемые технологии и инструменты.

Раздел «Тестирование» посвящен проверке и тестированию разработанного программного продукта на соответствие требованиям и его корректное функционирование. Здесь описываются методы тестирования, проведенные тесты и их результаты.

Раздел «Применение» содержит информацию о применении разработанного приложения в реальной среде. Здесь описывается область применения приложения, его функционирование и реальные условия использования. Приводятся инструкции по установке, запуску и использованию приложения.

Раздел «Заключение» содержит обобщение итогов разработки проекта, сделанные выводы и рекомендации. Обсуждаются преимущества разработанного приложения по сравнению с существующими методами управления расписанием занятий в автошколе.

В разделе «Список использованных источников» приводится список литературы, статей, руководств и других источников информации, использованных при разработке проекта.

**1 Структура и общая технологическая схема предприятия. Описание структурного подразделения, где проходила практика**

Структурные подразделения Управления Государственного Комитета Судебных Экспертиз (далее – УГКСЭ) по Гродненской области:

Управление криминалистических экспертиз и учетов

1. Управление технических экспертиз
2. Управление специальных экспертиз
3. Управление судебно-медицинских экспертиз
4. Управление судебно-психиатрических экспертиз
5. Управление координации служебной деятельности
6. Управление финансов и тыла
7. Управление лабораторных исследований вещественных доказательств биологического характера
8. Отдел оперативно-дежурной службы УГКСЭ
9. Отдел кадров и идеологической работы
10. Отдел режимно-секретной деятельности и делопроизводства
11. Отделение информатизации и связи

Преддипломная практика проходила на базе управления технических экспертиз управления Государственного комитета судебных экспертиз по Гродненской области. Управление специализируется на подготовке специалистов, проведению экспертиз, а также управлению техническими экспертизами по Гродненской области. Общая технологическая схема этого заведения включает следующие ключевые объекты:

1.Офисные кабинеты, предназначенные для проведения экспертиз.

2. Лаборатории и мастерские: специализированные помещения, оснащенные необходимым оборудованием для проведения экспертиз.

**2 Описание структуры ВЦ, АСУП и производственного  
отдела, где проходит практика.**

Преддипломная практика проходила на базе управления технических экспертиз управления Государственного комитета судебных экспертиз по Гродненской области.

Вычислительный центр (ВЦ):

Вычислительный центр организации является центральным элементом информационной инфраструктуры, обеспечивающим комплексные вычислительные ресурсы и сетевую инфраструктуру.

Функциями ВЦ являются: обеспечение работоспособности компьютерной техники организации, установка, настройка и обслуживание операционных систем, серверов и прикладного программного обеспечения, мониторинг и обеспечение безопасности информационной инфраструктуры, разработка и внедрение стратегий резервного копирования данных.

Автоматизированная система управления образовательным процессом (АСУП):

АСУП представляет собой комплекс программных и аппаратных средств, обеспечивающих эффективное проведение экспертиз.

Функциями ВЦ являются: обеспечение необходимого оборудования и ресурсов для проведения экспертиз.

**3 Должностные обязанности оператора ЭВМ, техника-программиста, инженера-программиста.**

Инженер-программист обязан:

* Разрабатывать программы на основе анализа математических моделей и алгоритмов для решения различных задач.
* Участвовать в выборе языка программирования и разрабатывать программный код на основе готовых спецификаций.
* Интегрировать программные компоненты и проводить проверку программ на основе логического анализа.
* Оптимизировать программный код и проводить отладку разработанных программ.
* Создавать библиотеки стандартных программ и участвовать в разработке программ для расширения области применения вычислительной техники.
* Участвовать в сопровождении программного обеспечения и оказывать помощь специалистам на различных этапах разработки.
* Вести документацию и выполнять работы по распоряжению руководителей структурных подразделений.
* Проводить фотосьемку мероприятий и оказывать содействие в организации учебно-воспитательного процесса.
* Поддерживать функционирование и продвижение сайта колледжа, а также обеспечивать обновление информации на нем.
* Соблюдать антикоррупционное законодательство и участвовать в развитии внебюджетной деятельности.

В области охраны труда инженер-программист обязан:

* Соблюдать требования по охране труда и правила поведения в рабочих помещениях.
* Проходить медицинские осмотры и обучение по охране труда.
* Заботиться о собственной безопасности и безопасности окружающих.
* Немедленно сообщать о любых ситуациях, угрожающих жизни или здоровью работающих.
* Поддерживать рабочее место в исправном состоянии и сотрудничать с нанимателем в обеспечении безопасных условий труда.
* Обеспечивать безопасное проведение образовательного процесса и вносить предложения по его улучшению.
* Содействовать работе комиссии по охране труда.

# 4 Анализ задачи

# 4.1 Описание предметной области

Темой проекта является «Разработка интернет-ресурса для онлайн-курсов по танцам «Elizabeth Khard».

Данный проект является сайтом, который позволит пользователям в удобном и понятном формате узнать всю интересующую их информацию, касаемо танцев в стиле трайбл-фьюжн. Помимо этого, каждый пользователь будет иметь возможность купить курс видеоуроков, которые помогут ему получить навыки танцев в данном стиле.

Данный проект разрабатывается с целью его практического использования заказчиком, а также пользователями для получения доступа к видеоурокам. Интерфейс интернет-ресурса должен быть интуитивно понятен и прост для пользователей, не обладающих знаниями в сфере программирования.

Входной информацией будут являться данные, вводимые в поля при регистрации, авторизации и при заполнении формы оплаты. Условно-постоянной будет являться информация, размещенная на сайте, различные базы данных и формы. К выходной информации будет относиться запись на курс и пополнение баланса пользователя.

Сайт будет носить характер сайта-визитки, что позволит пользователю получить наибольшее количество интересующей его информации в понятном для него виде.

Разрабатываемый интернет-ресурс будет содержать:

* общую информацию о танцах данного стиля;
* информацию о педагоге;
* информацию о курсах и планируемых нововведениях;
* галерею с фото- и видеоматериалами;
* ссылки на социальные сети;

При разработке интернет-ресурса следует учесть разделение пользователей на категории. В данном случае будет выделено 3 категории пользователей: гость, авторизированный пользователь, преподаватель. В зависимости от категории, пользователю будут доступны определенные функции.

Перечень задач и функций, которые должны быть реализованы в готовом программном продукте, делятся между собой по категориям пользователей.

Функции, доступные гостю:

* просмотр информации на сайте;
* регистрация;
* просмотр бесплатного пробного урока;
* переход к социальным сетям заказчика.

Функции, доступные пользователю:

* авторизация;
* пополнение счета пользователя;
* покупка курсов;
* просмотр видеоуроков из курсов, которые приобрел пользователь;
* удаление аккаунта пользователя;
* просмотр списка существующих курсов.

Функции, доступные преподавателю:

* просмотр учеников в курсе;
* удаление учеников из курса;
* просмотр списка существующих курсов;
* просмотр списка существующих видеоуроков;
* добавление видеоуроков;
* добавление курсов;

**5 Проектирование задачи**

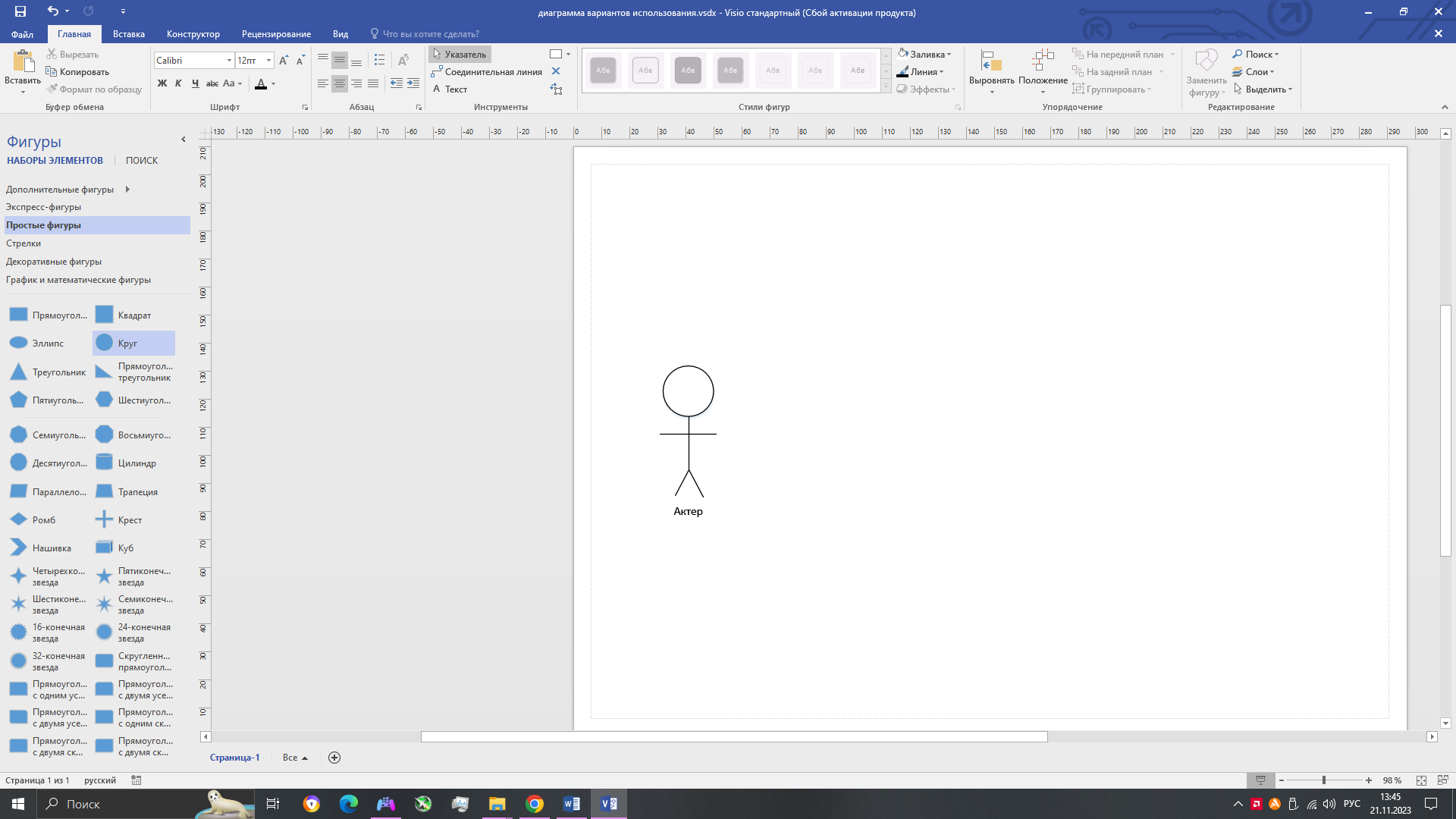
**5.1 Диаграмма вариантов использования**

Для проектирования информационного ресурса необходимо разработать диаграммы позволяющее проследить взаимодействие пользователя с программой и описать его основные действия информационного ресурса. Диаграммы вариантов использования описывают взаимоотношения и зависимость между группами вариантов использования и действующих лиц, участвующими в процессе.

Вариант использования описывает, с точки зрения действующего лица, группу действий в системе, которые приводят к конкретному результату.

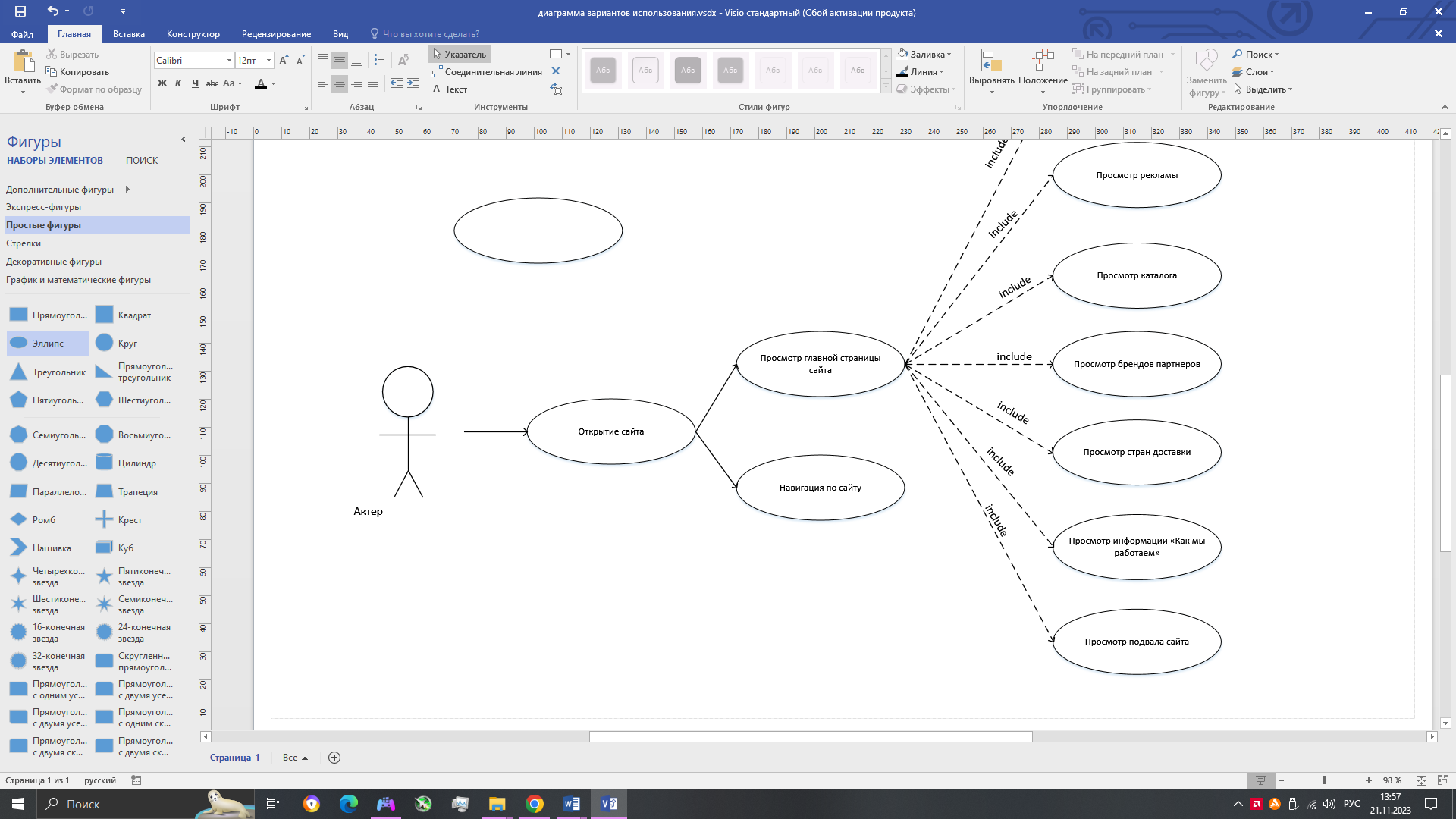
Варианты использования являются описаниями типичных взаимодействий между пользователями системы и самой системой. Они отображают внешний интерфейс системы и указывают форму того, что система должна сделать и в какой последовательности.

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. Актером или действующим лицом является любая сущность, взаимодействующая с системой извне. Изображение актера представлено на рисунке 1.



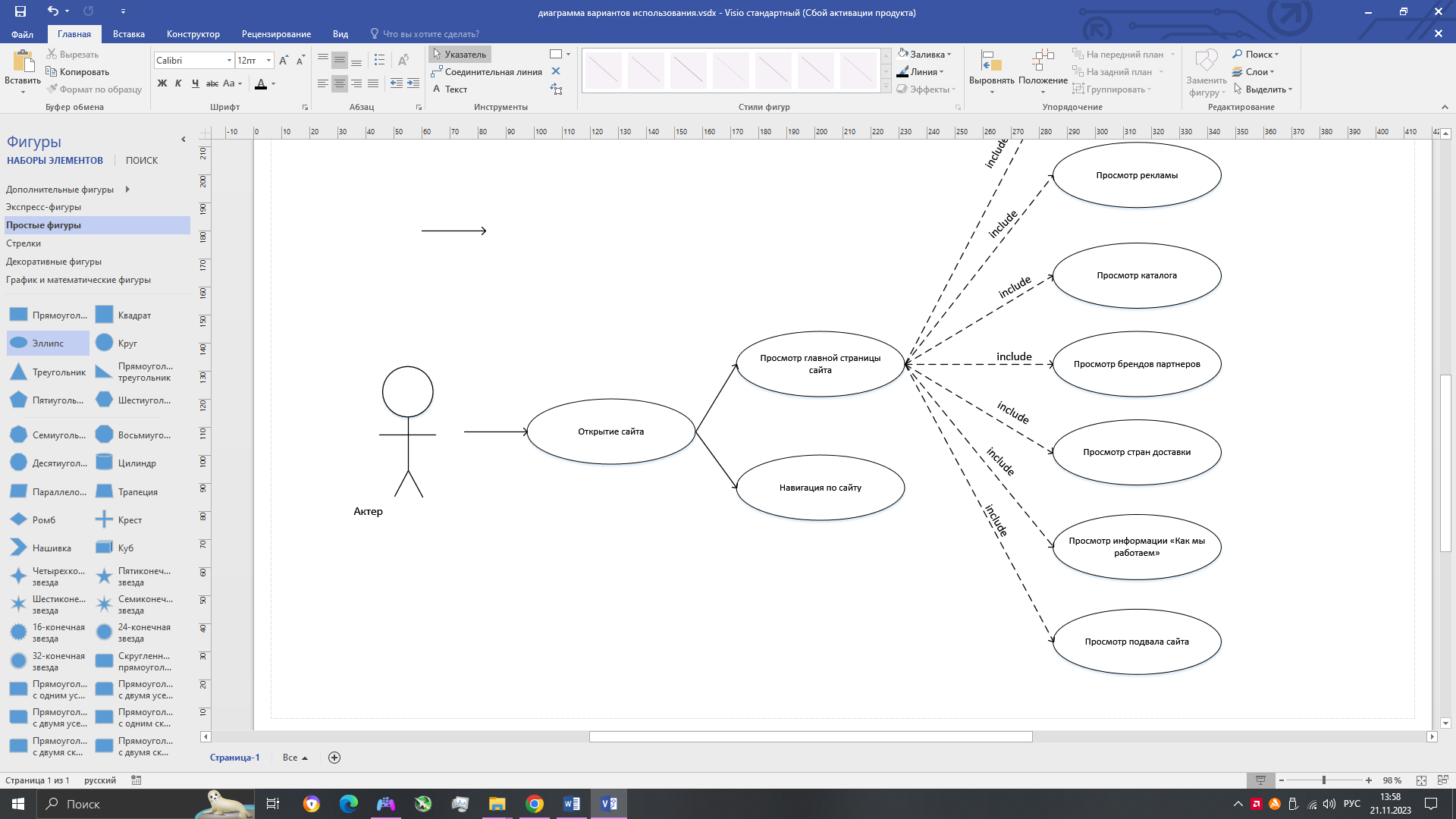
**Рисунок 1 – Актер**

Вариант использования является стандартным языком UML и применяется для спецификаций общих особенностей системы и любой другой сущности. Отдельные варианты использования обозначаются на диаграмме эллипсом, который показан на рисунке 2, в котором содержится его краткое название.



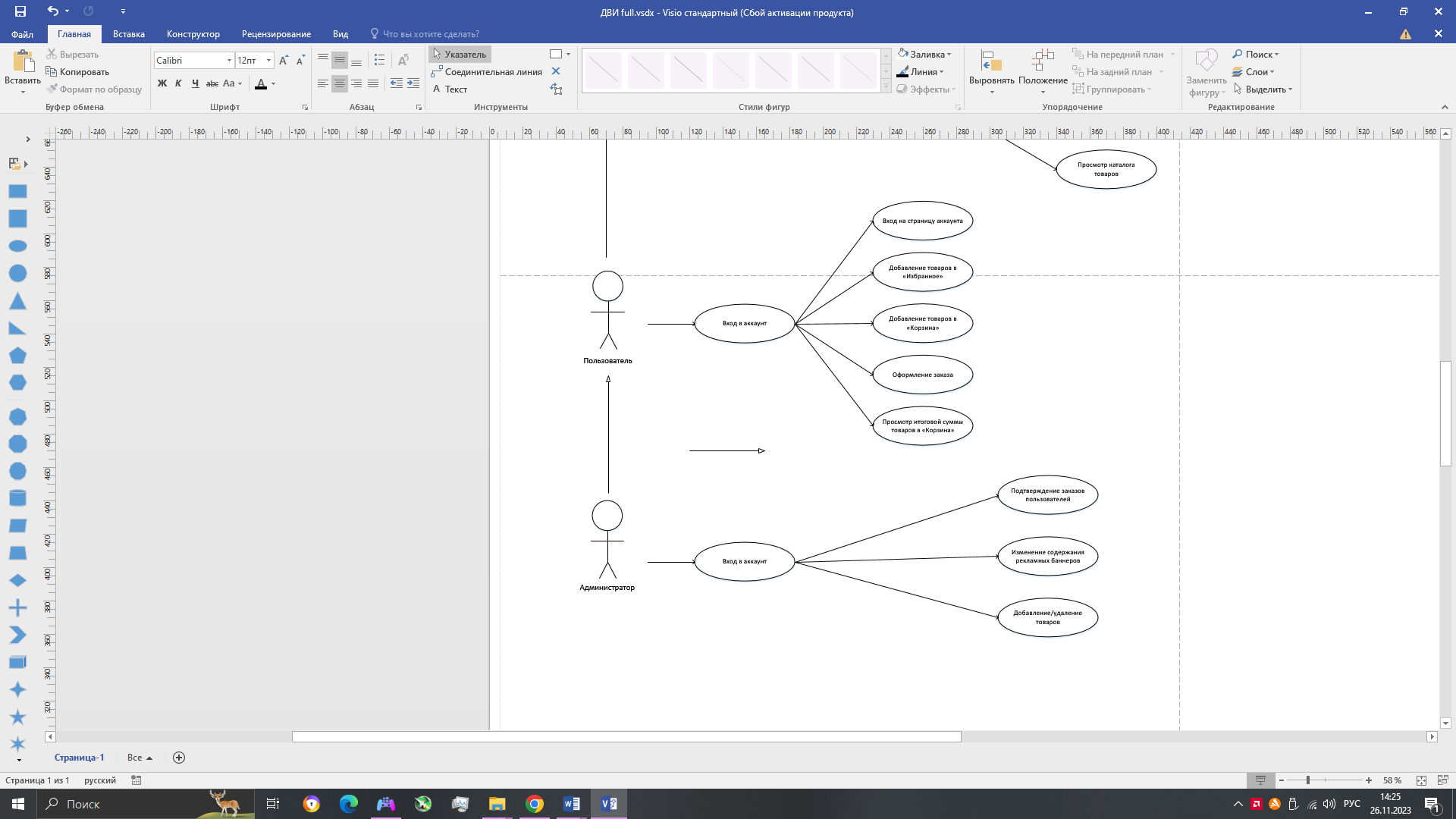
**Рисунок 2 – Вариант использования**

Отношение ассоциации является главным понятием языка UML и используется при построении всех графических моделей. Оно служит для обозначения роли актера и отдельном варианте использования. На диаграмме отношение ассоциации обозначается сплошной линией между актером и вариантом использования. Обозначение отношения ассоциации изображено на рисунке 3.



**Рисунок 3 – Обозначение отношения ассоциации**

Отношение общения в диаграмме вариантов использования – это отношение, в котором один вариант использования (дочерний) основан на другом варианте использования (родительском). Это означает, что дочерний вариант использования наследует все свойства и поведение родительского варианта использования. Обозначение отношения обобщения представлено на рисунке 4.



**Рисунок 4 – Обозначение отношения обобщения**

Определяя для выбранного актера варианты использования и устанавливая отношения между вариантами использования, получим полную диаграмму вариантов использования.

Диаграмма вариантов использования представлена в приложении А.

5.2 Модель данных

Цель моделирования данных состоит в определении концептуальной схемы совокупности данных в форме одной или нескольких локальных моделей, которые относительно легко могут быть отображены в любую форму представления данных.

Первый шаг моделирования – извлечение информации и выделение сущностей. Сущность – это реальный объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению в проекте.

\* <атрибут><имя сущности>

**Рисунок 5 — Графическое изображение сущности**

Модель данных представлена в Приложении Б.

Применимо к разрабатываемой задаче можно выделить следующие сущности:

Таблица 1 – Описание сущностей

|  |  |
| --- | --- |
| Сущность | Описание |
| Учащийся | Информация о учащемся |
| Курс | Информация о курсе |
| Видеоуроки | Информация о видеоуроках |
| Преподаватель | Информация о преподавателе |

Следующим шагом в моделировании данных является идентификация связей. Связь – это ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Степень связи и обязательность графически изображаются так, как показано на рисунке 6.

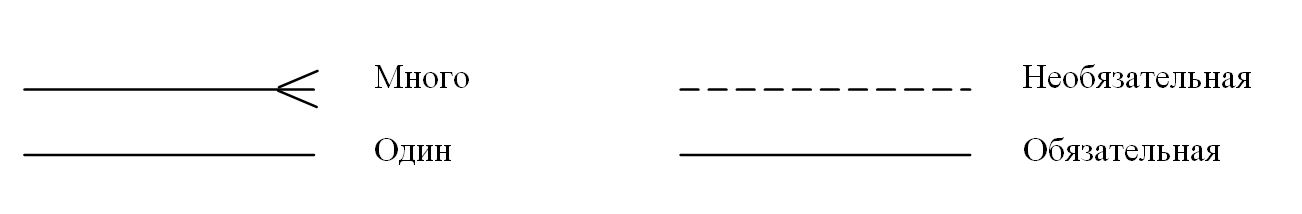


Рисунок 6 – Графическое изображение связи и обязательности

Логические зависимости (связи), которые прослеживаются между объектами нашей предметной области, будут обязательными. Причем, по характеру объединения данных установленные связи относятся к виду «Один – к – одному» и «Один – ко – многим». Связи вида «Многие – ко – многим» в рассматриваемой модели не используется.

Ознакомиться со связями в нашей модели данных можно в таблице 2.

Таблица 2 – Сущности и их связи

|  |  |  |
| --- | --- | --- |
| Сущность | Связи | Сущность |
| Учащийся | Один учащийся может быть подписан на несколько курсов | Курс |
| Крус | Один курс может содержать несколько видеоуроков | Видеоурок |
| Преподаватель | Один преподаватель может вести несколько курсов | Курс |

**5.3 Структура сайта**

Структура веб-сайта существенно влияет на восприятие контента и его значимость, а также на мотивацию, активность и самостоятельную работу пользователей.

Меню — это основная система навигации, предоставляющая возможность выбора из различных опций сайта. Оно состоит из коротких текстовых активных ссылок, направляющих пользователя на конкретные разделы.

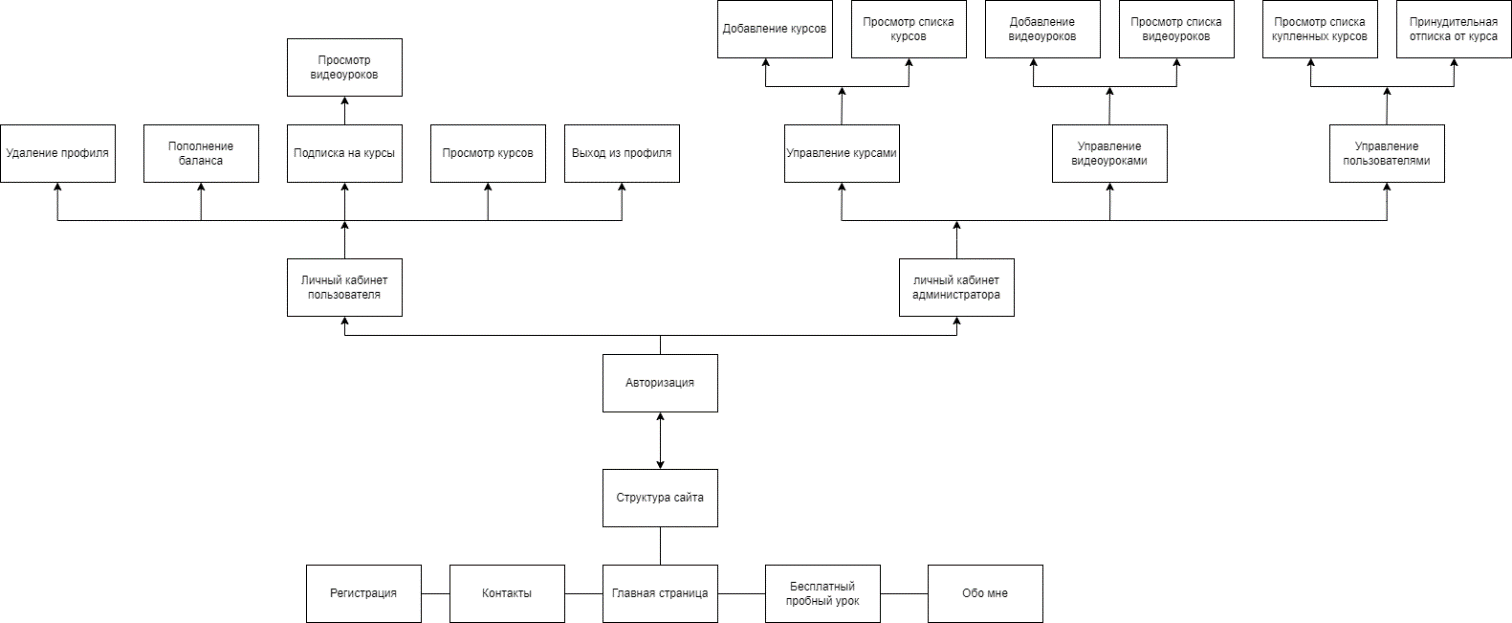
В соответствии с рисунком 7 представлена структура сайта.

Рисунок 7 – Структура сайта

**5.4 Выбор и обоснование среды разработки**

Для создания данного веб-приложения были использованы языки программирования JavaScript, PHP, а также инструмент дизайна Figma. Эти технологии широко применяются в веб-разработке, обеспечивая высокую эффективность и удобство создания интерфейсов.

HTML (Hypertext Markup Language) используется для структурирования содержимого веб-страницы. Он определяет элементы и их взаимосвязи, предоставляя основу для создания различных компонентов приложения.

CSS (Cascading Style Sheets) применяется для стилизации и внешнего оформления веб-страницы. Этот язык позволяет управлять внешним видом элементов, обеспечивая единообразный и привлекательный дизайн

JavaScript является языком программирования, который обеспечивает интерактивность веб-приложения. С его помощью реализованы динамические элементы, обработка событий и взаимодействие пользователя с интерфейсом.

PHP (Hypertext Preprocessor) — это серверный скриптовый язык программирования, который широко используется для создания динамических веб-сайтов и приложений. Он позволяет взаимодействовать с базами данных, обрабатывать формы, генерировать динамический контент и многое другое. PHP работает на стороне сервера, что обеспечивает быструю обработку запросов и генерацию HTML-страниц перед их отправкой клиенту.

Figma — это инструмент для дизайна интерфейсов, который предоставляет возможность создавать макеты, прототипы и визуализации. Он упрощает процесс дизайна и взаимодействия с элементами приложения.

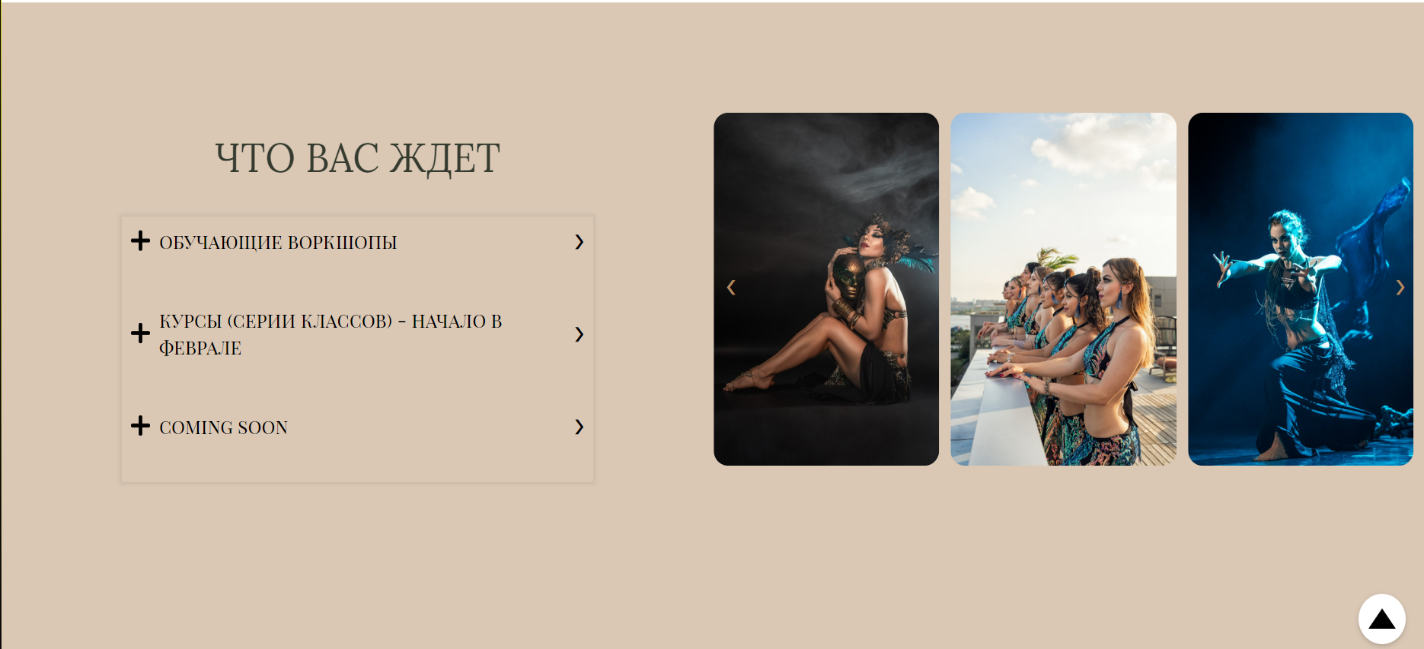
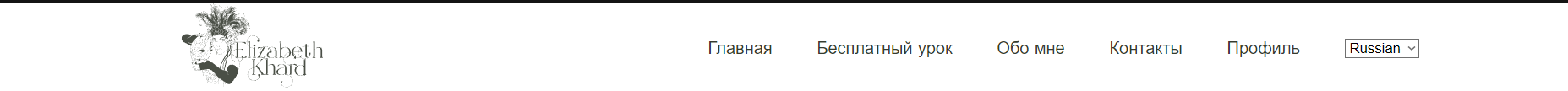
Основные преимущества использования этих технологий в веб-разработке включают высокую гибкость, удобство в использовании, а также широкие возможности для создания интерактивных и креативных пользовательских интерфейсов.

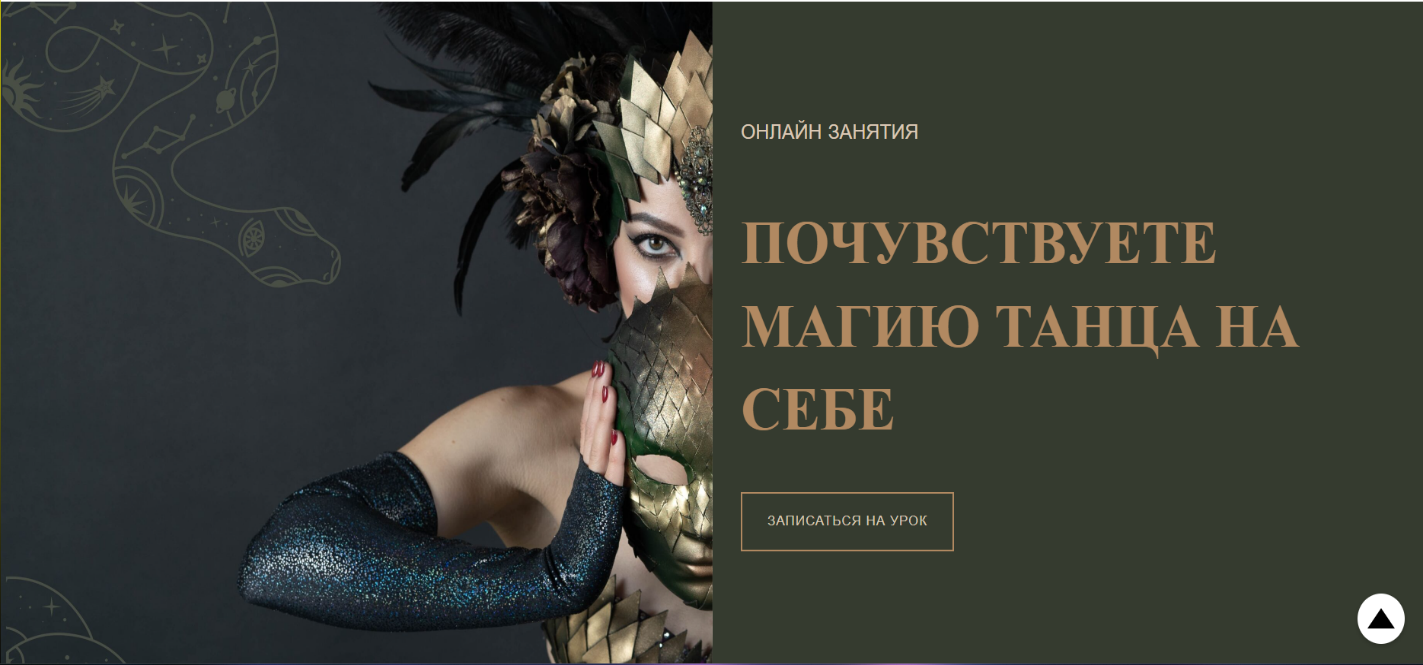
Таким образом, комбинация HTML, CSS, JavaScript, PHP и Figma обеспечивает необходимый инструментарий для эффективной разработки современных веб-приложений с привлекательным дизайном и высокой степенью взаимодействия.

**6 Реализация**

**6.1 Описание программного средства**

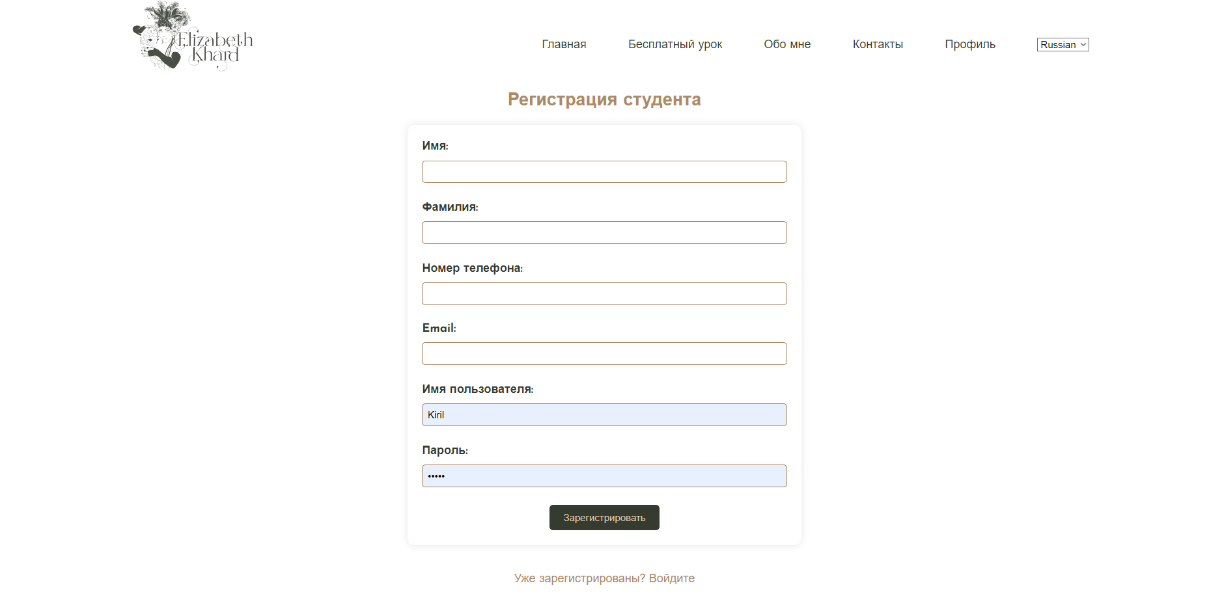
Интернет-ресурс включает в себя следующие ключевые модули:

* Главная страница (рис. 8)
* Профиль пользователя (рис.9)
* Страница регистрации (рис.10)
* Страница авторизации (рис.11)
* Страница пополнения баланса (рис.12)
* ****Профиль преподавателя (рис.13)

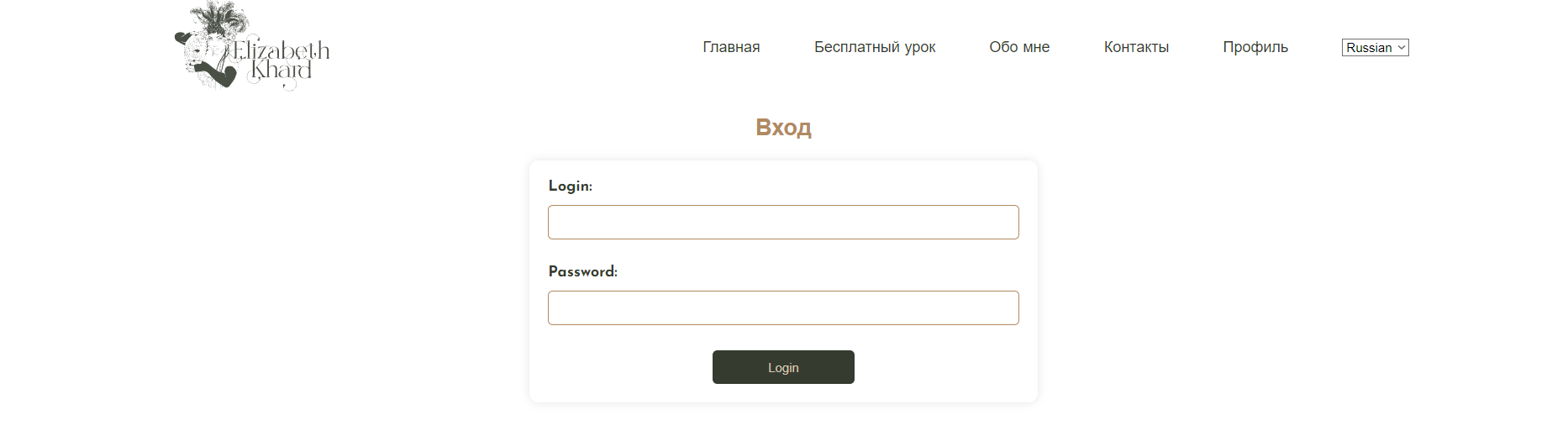
****

**Рисунок 8 – Главная страница**

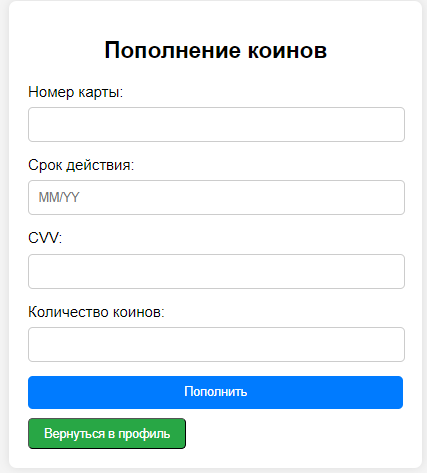
**Рисунок 9 – Профиль пользователя**

****

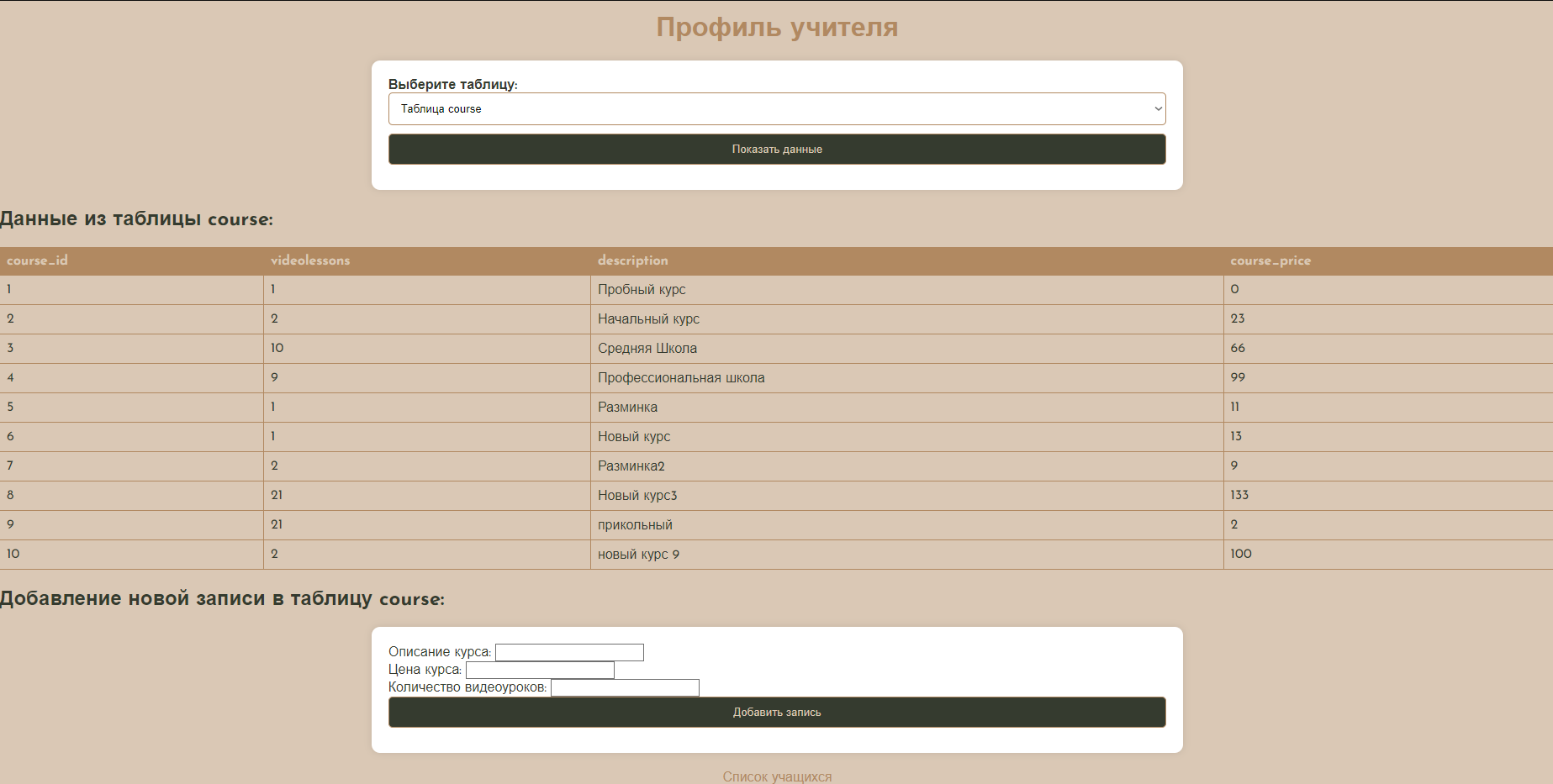
**Рисунок 10 – Страница регистрации**

****

**Рисунок 11 – Страница авторизации**

****

**Рисунок 12 – Страница пополнения баланса**

****

**Рисунок 13 – Профиль преподавателя**

**6.2 Функции и элементы управления**

Элементами управления в программе являются компоненты:

компонент <a> представляет собой ссылку. Данный компонент был использован в качестве кнопки управления.

Компонент <img> представляет собой картинку, которая заменяет собой кнопку.

Компонент <p> является обычным текстом, но обернув его в компонент <a> становится тоже элементом управления.

Спецификация проекта представлена в таблице 3.

Таблица 3 - Спецификация проекта

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Примечание |
| \*.html | Файл кода HTML | Cодержит общий код HTML. |
| \*.php | Файл кода PHP | Содержит общий код PHP |
| \*.css | Файл кода CSS | Cодержит общий код CSS. |
| \*.js | Файл кода JavaScript | Содержит общий код JavaScript. |

Таблица 3 - Продолжение

|  |  |  |
| --- | --- | --- |
| \*.svg | Файл изображений | Используются для отображения графической информации изображения. |
| \*.jpg | Файлы изображений | Используются для отображения графической информации изображения. |
| \*.ttf | Файлы шрифтов | Используется для подключения шрифтов к проекту. |
| \*.otf | Файлы шрифтов | Используется для подключения шрифтов к проекту. |

**7 Тестирование**

Этап тестирования является одним из ключевых этапов создания программного продукта. На этой стадии выявляются и устраняются ошибки, допущенные в процессе разработки.

Тест-кейс — это формально описанный алгоритм тестирования программы, специально созданный для определения возникновения в программе определённой ситуации, определённых выходных данных. Часто варианты тестирования группируют в тестовые наборы. Если к программе предъявляются определённые формальные требования, то варианты тестирования составляются таким образом, чтобы были охвачены все эти требования. Для приложений без формальных требований варианты тестирования могут быть созданы, основываясь на типичном поведении программ сходного класса. В соответствии с таблицей 4 приведены тест-кейс разрабатываемого проекта.

**Таблица 4 – Тест-кейсы**

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Описание теста | Ожидаемый результат | Отчет |
| 1 | Записать данные в форму регистрации | Ученик успешно зарегистрирован | Выполнено |
| 2 | Написать валидные данные в форму авторизации | Открывается страница профиля ученика | Выполнено |
| 3 | Нажать на кнопку «удалить аккаунт» в профиле пользователя | Ученик удален из базы данных | Выполнено |
| 4 | Нажать на кнопку «подписаться на курс» в профиле пользователя | Ученик успешно подписан на курс | Выполнено |
| 5 | Нажать на кнопку «посмотреть курс» в профиле пользователя | Открывается страница со списком видеоуроков в курсе | Выполнено |

**Таблица 4 - Продолжение**

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | Написать валидные данные преподавателя в форму авторизации | Открывается страница профиля преподавателя | Выполнено |
| 7 | В выпадающем меню выбрать таблицу и нажать кнопку «показать данные» | Данные из таблицы будут корректно отображены | Выполнено |
| 8 | В форме добавления новой записи указать валидные данные и нажать кнопку «добавить запись» | Запись успешно добавлена в таблицу | Выполнено |
| 9 | Нажать на кнопку список учащихся в профиле преподавателя | Открывается страница со списком учащихся | Выполнено |
| 10 | Выбрать курс в выпадающем меню и нажать кнопку «показать» | Будет выведена таблица с учениками, которые подписаны на данный курс | Выполнено |
| 11 | Нажать на кнопку «отписать» рядом с учащимися | Учащийся будет отписан от курса | Выполнено |
| 12 | Нажать на кнопку «Пополнить счет» | Откроется форма пополнения счета | Выполнено |
| 13 | Заполнить форму и нажать на кнопку «пополнить баланс» | Баланс пользователя будет пополнен | Выполнено |

**8 Применение**

**8.1 Инсталляция**

Для того чтобы успешно запустить разработанный программный продукт вам нужно установить open server panel. Скачать его можно с официального сайта <https://ospanel.io>.

После установки приложение вам нужно перенести папку проекта в папку проектов Open Server Panel. Для этого вы можете нажать на пункт выпадающего меню «Папка с проектами». После этого перенесите файлы в папку, открывшуюся в проводнике. Затем нажмите «Запустить» в выпадающем меню программы Open Server. Откройте браузер и в поле ввода url введите http://localhost/project.

**8.2 Обучение**

**8.2.1 Обучение для пользователя**

* Зарегистрируйтесь на сайте заполнив форму регистрации
* Авторизуйтесь на сайте введя свой логин и пароль на форме авторизации
* Вы можете подписаться на курс, если у вас достаточно средств, нажав на кнопку подписаться рядом с курсом
* Вы можете посмотреть курс, на который вы подписались, спустившись ниже по странице в разделе «Подписанные курсы», нажав на кнопку «Посмотреть курс»
* Вы можете перейти к просмотру видеоурока нажав на его url
* Вы можете удалить свой аккаунт нажав на кнопку «Удалить аккаунт» возле вашего имени и фамилии
* Вы можете пополнить баланс нажав на кнопку пополнить баланс, и заполнив данные в форм

**8.2.2 Обучение для учителя**

* С формы регистрации перейдите на форму авторизации
* Введите в поля логин admin и пароль admin
* Вы можете просмотреть записи в таблице курсов и таблице видеоуроков. Для этого выберите нужную вам таблицу в выпадающем меню и нажмите на кнопку «показать данные»
* Вы можете добавить данные в выбранную таблицу. Для этого заполните форму под таблицей и нажмите на кнопку «Добавить запись»
* Вы можете просмотреть список учащихся, для этого прокрутите страницу вниз и нажмите на текст «Список учащихся».
* Вам будет предложено выбрать курс для просмотра учащихся на этом курсе.
* Выберите нужный вам курс и нажмите на кнопку «Показать»
* В таблице учащихся вы можете нажать на кнопку «Отписать», и ученик будет отписан от курса

**Заключение**

В ходе технологической практики, проведенной в управлении техническими экспертизами УГКСЭ, был успешно реализованы поставленные задачи на преддипломную практику.

Прохождение практики в этом структурном подразделении позволило ознакомиться с организацией образовательного процесса с применением информационно-коммуникационных технологий. Были получены знания в области системного администрирования и разработке веб-ресурсов, включая языки программирования HTML, CSS, JavaScript и PHP.

В ходе работы над проектом были решены следующие задачи:

1. Структурирование и Организация Информации: С использованием HTML и CSS была создана эффективная структура ресурса, обеспечивающая удобную навигацию и визуальную привлекательность. Это включает в себя легко воспринимаемые страницы.
2. Интерактивность и Пользовательский Опыт: С помощью JavaScript были реализованы различные интерактивные элементы, обеспечивающие более углубленное взаимодействие с контентом. Это включает в себя тестирование, задания для самостоятельной работы и другие формы взаимодействия.
3. Визуализация и Графическое Оформление: При помощи CSS был разработан стильный и современный дизайн.
4. Тестирование и Отладка: Проект был подвергнут тщательному тестированию и отладке, что позволило минимизировать ошибки и обеспечить стабильную работу.

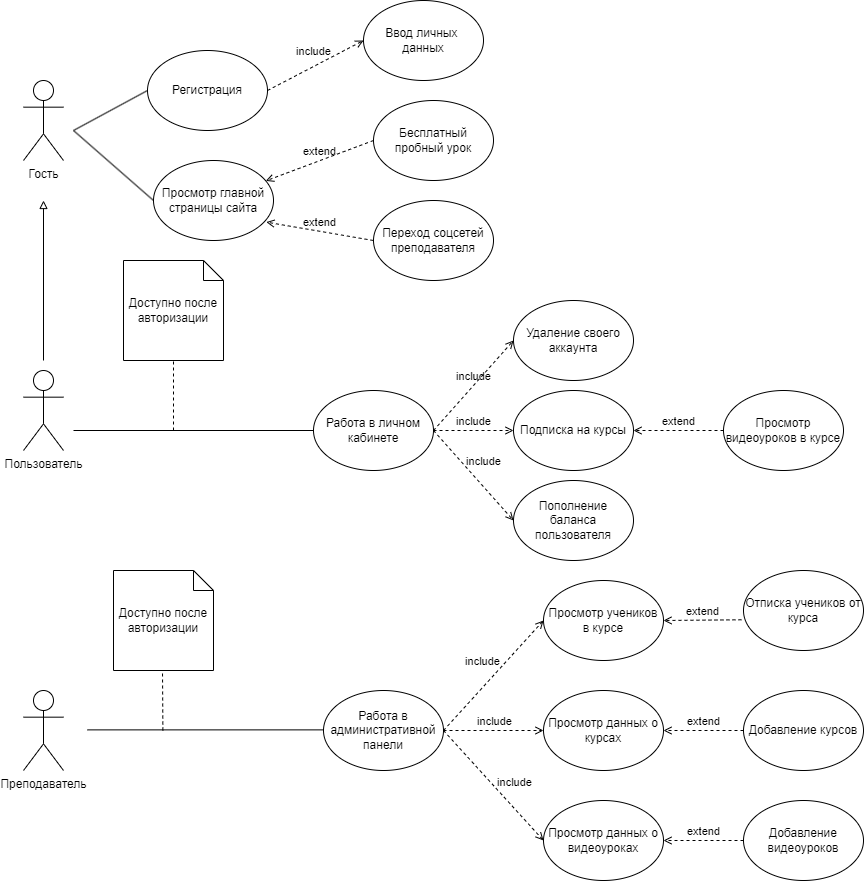
Однако продукт нуждается в дальнейшем обслуживании, добавлении и обновлении информации о курсах, новостях и создании новых видеоуроков.

**Список использованных источников**

* Основы HTML [Электронный ресурс]: Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/HTML\_basics
* Основы CSS [Электронный ресурс]: Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/CSS\_basics
* Современный учебник JavaScript [Электронный ресурс]: Режим доступа: https://learn.javascript.ru/
* Электронные образовательные ресурсы. Виды, структуры, технологии [Электронный ресурс]: Режим доступа: http://swsys-web.ru/electronic-educational-resources.html
* Что такое JavaScript? [Электронный ресурс] : Режим доступа : https://developer.mozilla.org/ru/docs/Learn/JavaScript/First\_steps/What\_is\_JavaScript
* Codepen [Электронный ресурс]: Режим доступа: https://codepen.io/
* «The Spruce: Make Your Best Home» [Электронный ресурс]: Режим доступа: https://www.thespruce.com/search?q=tile
* «PHP: Resources – Manual» [Электронный ресурс]: Режим доступа: https://www.php.net/manual/en/language.types.resource.php
* «PHP Exercises, Practice, Solution - w3resource» [Электронный ресурс]: Режим доступа: https://www.w3resource.com/php-exercises/
* «PHP Resources - Online Tutorials Library» [Электронный ресурс]: Режим доступа: https://www.tutorialspoint.com/php-resources

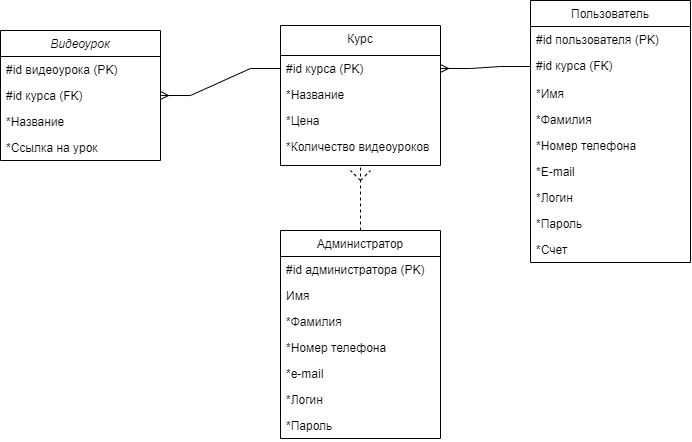
**Приложение А**

**Диаграмма вариантов использования**

****

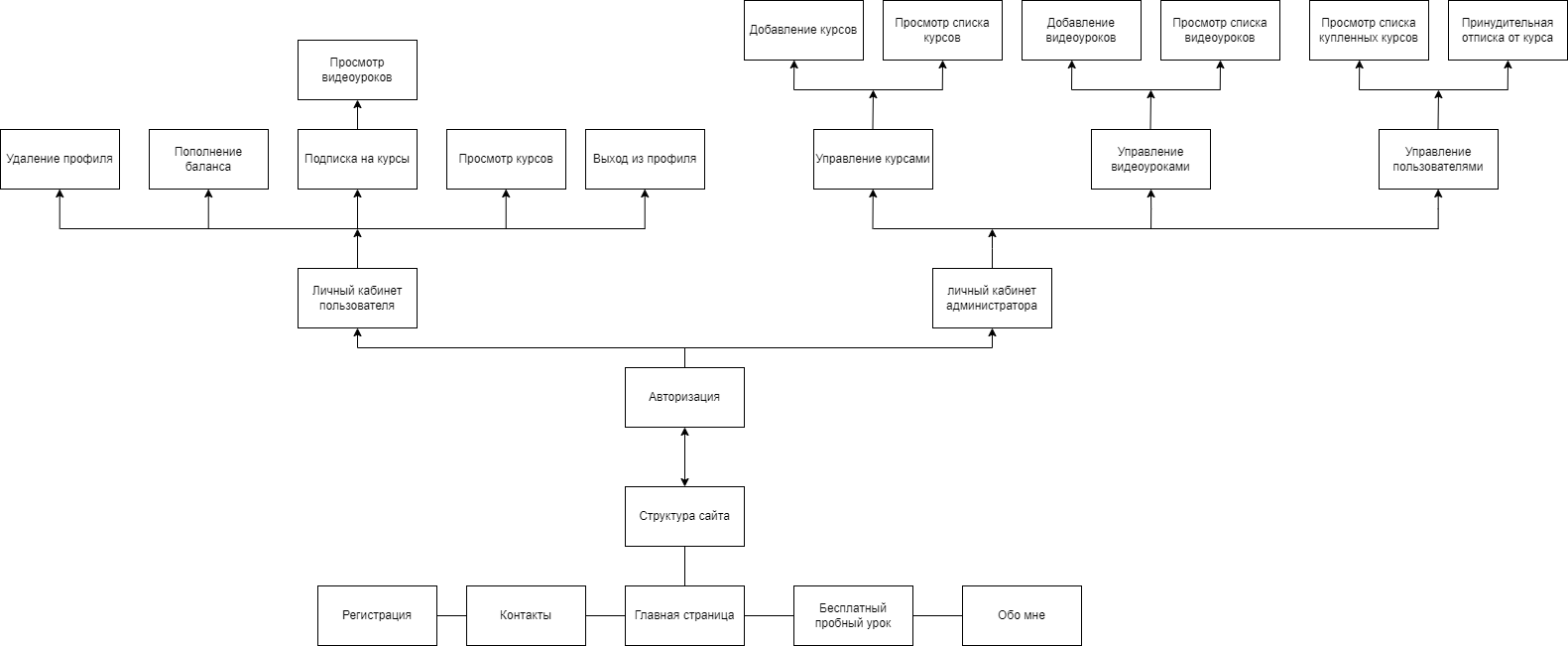
**Приложение Б**

**Модель данных**

****

**Приложение В**

**Структура сайта**

****

**Приложение Г**

**Листинг**

<?php

session\_start();

// Подключение к базе данных

$mysqli = new mysqli("localhost", "root", "", "bdcursach");

// Проверка подключения

if ($mysqli->connect\_error) {

die("Connection failed: " . $mysqli->connect\_error);

}

// Проверяем, авторизован ли пользователь

if (!isset($\_SESSION['login'])) {

header('Location: login.php');

exit;

}

// Получение данных о пользователе из базы данных

$login = $\_SESSION['login'];

$stmt = $mysqli->prepare("SELECT \* FROM student WHERE login = ?");

$stmt->bind\_param("s", $login);

$stmt->execute();

$result = $stmt->get\_result();

// Проверка наличия данных о пользователе

if ($result->num\_rows === 0) {

echo "Ошибка: Пользователь не найден";

exit;

}

$user = $result->fetch\_assoc();

// Получение списка курсов, на которые пользователь не подписан

$courses\_stmt = $mysqli->prepare("SELECT \* FROM course WHERE course\_id NOT IN (SELECT course\_id FROM student WHERE FIND\_IN\_SET(course\_id, ?))");

$courses\_stmt->bind\_param("s", $user['subscribed\_courses']);

$courses\_stmt->execute();

$courses\_result = $courses\_stmt->get\_result();

// Функция удаления аккаунта

function deleteAccount() {

global $mysqli, $login;

$delete\_stmt = $mysqli->prepare("DELETE FROM student WHERE login = ?");

$delete\_stmt->bind\_param("s", $login);

$delete\_stmt->execute();

session\_destroy();

header('Location: login.php');

exit;

}

function subscribeCourse($course\_id, $course\_price) {

global $mysqli, $user;

// Проверяем, является ли курс пробным (ID 1 в базе данных)

$is\_trial\_course = ($course\_id == 1); // Предполагая, что ID пробного курса равен

// Проверяем, хватает ли у пользователя коинов для оплаты курса, если это не пробный курс

if (!$is\_trial\_course && $user['coins'] < $course\_price) {

// Недостаточно коинов для оплаты курса

return false;

}

// Если это не пробный курс или пользователь имеет достаточно коинов, продолжаем подписку

if ($user['coins'] >= $course\_price || $is\_trial\_course) {

// Вычитаем стоимость курса из баланса пользователя

if (!$is\_trial\_course) {

$new\_coins\_balance = $user['coins'] - $course\_price;

// Обновляем баланс пользователя в базе данных

$update\_coins\_stmt = $mysqli->prepare("UPDATE student SET coins = ? WHERE login = ?");

$update\_coins\_stmt->bind\_param("is", $new\_coins\_balance, $user['login']);

$update\_coins\_stmt->execute();

// Проверяем успешность обновления баланса

if ($update\_coins\_stmt->affected\_rows !== 1) {

return false;

}

}

// Добавляем ID курса в список подписанных курсов пользователя

$subscribed\_courses = $user['subscribed\_courses'] ? explode(",", $user['subscribed\_courses']) : [];

if (!in\_array($course\_id, $subscribed\_courses)) {

$subscribed\_courses[] = $course\_id;

$subscribed\_courses\_str = implode(",", $subscribed\_courses);

$update\_subscribed\_courses\_stmt = $mysqli->prepare("UPDATE student SET subscribed\_courses = ? WHERE login = ?");

$update\_subscribed\_courses\_stmt->bind\_param("ss", $subscribed\_courses\_str, $user['login']);

$update\_subscribed\_courses\_stmt->execute();

// Проверяем успешность обновления списка подписанных курсов

if ($update\_subscribed\_courses\_stmt->affected\_rows === 1) {

return true; // Пользователь успешно подписан на курс

} else {

return false;

}

}

} else {

// Недостаточно коинов для оплаты курса

return false;

}

}

// Проверка нажатия кнопки "Удалить аккаунт"

if (isset($\_POST['delete\_account'])) {

deleteAccount();

}

// Проверка нажатия кнопки "Подписаться на курс"

if (isset($\_POST['subscribe\_course'])) {

$course\_id = $\_POST['course\_id'];

$course\_price = $\_POST['course\_price']; // Добавляем получение цены курса

if (subscribeCourse($course\_id, $course\_price)) {

// Пользователь успешно подписан на курс

header('Location: profile.php'); // Перенаправление на страницу профиля

exit;

} else {

// Недостаточно коинов для оплаты курса

echo "<script>alert('Недостаточно коинов для оплаты курса');</script>";

}

}

// Проверка нажатия кнопки "Посмотреть курс" для подписанных курсов

if (isset($\_POST['view\_subscribed\_course'])) {

$course\_id = $\_POST['course\_id'];

header("Location: course.php?course\_id=$course\_id"); // Перенаправление на страницу курса

exit;

}

// Закрытие подключения

$stmt->close();

$courses\_stmt->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Профиль</title>

<link rel="stylesheet" type="text/css" href="profile\_styles.css">

</head>

<body>

<header>

<div class="logo-holder">

<img src="images/eliyabhet.svg" alt="" style="height: 12vh; width: 12vw;">

</div>

<ul class="menu">

<li><a href="index.html">Вернуться на Главную</a></li>

</ul>

</header>

<h1>Профиль пользователя</h1>

<div>

<p>Имя: <?php echo $user['Name']; ?></p>

<p>Фамилия: <?php echo $user['Surname']; ?></p>

<div class="coin-block">

<img src="images/coin\_icon.png" alt="Coin Icon" class="coin-icon" width="16px" >

<span class="coin-count"><?php echo $user['coins']; ?></span>

<form method="post" action="payment\_simulation.php">

<button type="submit" name="simulate\_payment">Пополнить баланс</button>

</form>

</div>

<form method="post">

<button type="submit" name="delete\_account">Удалить аккаунт</button>

</form>

</div>

<h2>Доступные курсы</h2>

<table>

<thead>

<tr>

<th>Название курса</th>

<th>Цена</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

<?php while ($course = $courses\_result->fetch\_assoc()): ?>

<tr>

<td><?php echo $course['description']; ?></td>

<td><?php echo strval($course['course\_price']); ?></td>

<td>

<form method="post">

<input type="hidden" name="course\_id" value="<?php echo $course['course\_id']; ?>">

<input type="hidden" name="course\_price" value="<?php echo $course['course\_price']; ?>">

<button type="submit" name="subscribe\_course">Подписаться на курс</button>

</form>

</td>

</tr>

<?php endwhile; ?>

</tbody>

</table>

<h2>Подписанные курсы</h2>

<?php

// Получение подписанных курсов пользователя

$subscribed\_courses = explode(",", $user['subscribed\_courses']);

if (empty($subscribed\_courses)) {

echo "<p>У вас пока нет подписанных курсов</p>";

} else {

// Формируем строку с плейсхолдерами для IN оператора

$placeholders = implode(',', array\_fill(0, count($subscribed\_courses), '?'));

// Подготавливаем запрос с динамическим количеством параметров

$subscribed\_courses\_stmt = $mysqli->prepare("SELECT \* FROM course WHERE course\_id IN ($placeholders)");

if ($subscribed\_courses\_stmt) {

// Привязываем параметры

$types = str\_repeat('i', count($subscribed\_courses)); // 'i' означает integer

$subscribed\_courses\_stmt->bind\_param($types, ...$subscribed\_courses);

// Выполняем запрос

$subscribed\_courses\_stmt->execute();

$subscribed\_courses\_result = $subscribed\_courses\_stmt->get\_result();

?>

<table>

<thead>

<tr>

<th>Название курса</th>

<th>Цена</th>

<th>Действия</th>

</tr>

</thead>

<tbody>

<?php while ($course = $subscribed\_courses\_result->fetch\_assoc()): ?>

<tr>

<td><?php echo $course['description']; ?></td>

<td><?php echo $course['course\_price']; ?></td>

<td>

<form method="post">

<input type="hidden" name="course\_id" value="<?php echo $course['course\_id']; ?>">

<button type="submit" name="view\_subscribed\_course">Посмотреть курс</button>

</form>

</td>

</tr>

<?php endwhile; ?>

</tbody>

</table>

<?php } else { ?>

<p>Ошибка при выполнении запроса к базе данных</p>

<?php } ?>

<?php } ?>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Профиль учителя</title>

<link rel="stylesheet" type="text/css" href="styles\_teacher\_profile.css">

</head>

<body>

<h1>Профиль учителя</h1>

<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

<label for="table">Выберите таблицу:</label>

<select name="table" id="table">

<option value="course">Таблица course</option>

<option value="video\_lessons">Таблица video\_lessons</option>

</select>

<button type="submit">Показать данные</button>

</form>

<?php

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

$selected\_table = $\_POST["table"];

// Подключение к базе данных

$mysqli = new mysqli("localhost", "root", "", "bdcursach");

// Проверка подключения

if ($mysqli->connect\_error) {

die("Connection failed: " . $mysqli->connect\_error);

}

// SQL запрос для выбора данных из выбранной таблицы

$sql = "SELECT \* FROM " . $selected\_table;

$result = $mysqli->query($sql);

if ($result->num\_rows > 0) {

echo "<h2>Данные из таблицы $selected\_table:</h2>";

echo "<table border='1'>";

// Вывод заголовков таблицы

echo "<tr>";

$row = $result->fetch\_assoc();

foreach ($row as $key => $value) {

echo "<th>$key</th>";

}

echo "</tr>";

// Вывод данных таблицы

do {

echo "<tr>";

foreach ($row as $value) {

echo "<td>$value</td>";

}

echo "</tr>";

} while ($row = $result->fetch\_assoc());

echo "</table>";

// Показываем форму для добавления записи только для таблицы curse и video\_lessons

if ($selected\_table == 'course' || $selected\_table == 'video\_lessons') {

echo "<h2>Добавление новой записи в таблицу $selected\_table:</h2>";

echo "<form method='post' action='";

if ($selected\_table == 'course') {

echo "add\_curse.php";

echo "'>";

echo "Описание курса: <input type='text' name='description'><br>";

echo "Цена курса: <input type='text' name='course\_price'><br>";

echo "Количество видеоуроков: <input type='text' name='videolessons'><br>"; // Новое поле

} elseif ($selected\_table == 'video\_lessons') {

echo "add\_record.php";

echo "'>";

echo "Название урока: <input type='text' name='lesson\_name'><br>";

echo "ID курса: <input type='text' name='course\_id'><br>";

echo "URL видеоурока: <input type='text' name='url'><br>";

}

echo "<button type='submit'>Добавить запись</button>";

echo "</form>";

}

} else {

echo "Нет данных в таблице $selected\_table";

}

// Закрытие подключения

$mysqli->close();

}

?>

<!-- Кнопка для перехода на страницу "Список учащихся" -->

<a href="students\_list.php">Список учащихся</a>

</body>

</html>

<?php

session\_start();

// Подключение к базе данных

$mysqli = new mysqli("localhost", "root", "", "bdcursach");

// Проверка подключения

if ($mysqli->connect\_error) {

die("Connection failed: " . $mysqli->connect\_error);

}

// Проверяем, авторизован ли пользователь

if (!isset($\_SESSION['login'])) {

header('Location: login.php');

exit;

}

// Получение данных о пользователе из базы данных

$login = $\_SESSION['login'];

$stmt = $mysqli->prepare("SELECT \* FROM student WHERE login = ?");

$stmt->bind\_param("s", $login);

$stmt->execute();

$result = $stmt->get\_result();

// Проверка наличия данных о пользователе

if ($result->num\_rows === 0) {

echo "Ошибка: Пользователь не найден";

exit;

}

$user = $result->fetch\_assoc();

// Проверяем, были ли переданы данные из формы пополнения

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

// Проверяем, было ли указано количество коинов для пополнения

if (isset($\_POST["amount"])) {

// Получаем количество коинов, которое пользователь хочет добавить

$amount = intval(trim($\_POST["amount"]));

// Проверяем, чтобы количество коинов было положительным числом

if ($amount > 0) {

// Увеличиваем баланс пользователя на указанное количество коинов

$new\_coins\_balance = $user['coins'] + $amount;

// Обновляем баланс пользователя в базе данных

$update\_coins\_stmt = $mysqli->prepare("UPDATE student SET coins = ? WHERE login = ?");

$update\_coins\_stmt->bind\_param("is", $new\_coins\_balance, $user['login']);

$update\_result = $update\_coins\_stmt->execute();

// Проверяем успешность выполнения запроса и ошибки, если есть

if ($update\_result === FALSE) {

printf("Ошибка при выполнении запроса: %s\n", $mysqli->error);

exit;

}

// Проверяем количество строк, которые были затронуты запросом

if ($update\_coins\_stmt->affected\_rows === 1) {

// Пополнение коинов произведено успешно

header('Location: profile.php');

exit;

} else {

// Ошибка при обновлении баланса

$error\_message = "Ошибка при пополнении коинов. Пожалуйста, попробуйте снова.";

header('Location: payment\_simulation.php?success=false&error=' . urlencode($error\_message));

exit;

}

} else {

// Неверно указано количество коинов

$error\_message = "Пожалуйста, введите корректное количество коинов для пополнения.";

header('Location: payment\_simulation.php?success=false&error=' . urlencode($error\_message));

exit;

}

} else {

// Количество коинов не было передано

$error\_message = "Пожалуйста, укажите количество коинов для пополнения.";

header('Location: payment\_simulation.php?success=false&error=' . urlencode($error\_message));

exit;

}

}

// Закрытие подключения

$stmt->close();

$mysqli->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Пополнение коинов</title>

<link rel="stylesheet" type="text/css" href="payment\_styles.css">

</head>

<body>

<div class="container">

<h2>Пополнение коинов</h2>

<form id="paymentForm" action="process\_payment.php" method="post">

<label for="cardNumber">Номер карты:</label>

<input type="text" id="cardNumber" name="cardNumber" required>

<label for="expiryDate">Срок действия:</label>

<input type="text" id="expiryDate" name="expiryDate" placeholder="MM/YY" required>

<label for="cvv">CVV:</label>

<input type="text" id="cvv" name="cvv" maxlength="3" required>

<label for="amount">Количество коинов:</label>

<input type="number" id="amount" name="amount" required>

<button type="submit" id="submitBtn">Пополнить</button>

</form>

<script>

document.getElementById('paymentForm').addEventListener('submit', function(event) {

// Отображаем анимацию загрузки и сообщение о запросе в обработке

document.getElementById('loading').style.display = 'block';

document.getElementById('messageText').textContent = 'Запрос обрабатывается, подождите...';

document.getElementById('successMessage').style.display = 'block';

// Отладочный вывод - печатаем данные формы

console.log('Form data:', new FormData(this));

});

</script>

<!-- Добавляем кнопку для возврата на страницу профиля -->

<button id="returnLink" onclick="window.location.href = 'profile.php';">Вернуться в профиль</button>

</body>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

}

.container {

max-width: 400px;

margin: 0 auto;

background-color: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h2 {

text-align: center;

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 8px;

}

input[type="text"],

input[type="number"] {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

}

input[type="text"]:focus,

input[type="number"]:focus {

outline: none;

border-color: #007bff;

box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);

}

button[type="submit"] {

width: 100%;

padding: 10px;

border: none;

border-radius: 5px;

background-color: #007bff;

color: #fff;

cursor: pointer;

}

button[type="submit"]:hover {

background-color: #0056b3;

}

#loading,

#successMessage {

margin-top: 20px;

text-align: center;

}

#loading p {

font-style: italic;

}

#successMessage p {

margin-bottom: 10px;

}

#returnLink {

display: block;

padding: 8px 16px;

margin-top: 10px;

background-color: #28a745;

color: #fff;

text-decoration: none;

border-radius: 5px;

transition: background-color 0.3s ease;

}

#returnLink:hover {

background-color: #218838;

}