Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Магнитогорский государственный технический университет   
им Г.И. Носова»

Многопрофильный колледж

Отделение №2 Информационные технологии и транспорт

**КУРСОВОЙ ПРОЕКТ**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

по ПМ.05. Проектирование и разработка информационных систем

МДК.05.02. Разработка кода информационных систем

на тему: Разработка автоматизированной информационной системы сайта знакомств

Исполнитель: Аверин А.А. студент 4 курса, группа ИспВ-20-1

Руководитель: Зорина И.Г., преподаватель МпК

Работа допущена к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Магнитогорск, 2023

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Магнитогорский государственный технический университет   
им. Г. И. Носова»

Многопрофильный колледж

**КУРСОВОЙ ПРОЕКТ**

Тема Разработка автоматизированной информационной системы сайта знакомств

Студент Аверин Артём Александрович

Задание

Исходные данные:

по заданной предметной области:

1. выполнить моделирование информационной системы;
2. спроектировать и разработать базу данных;
3. разработать серверную и клиентскую части приложения;
4. разграничить права доступа;
5. выполнить тестирование информационной системы.

Состав и содержание проекта:

1. построение UML-диаграмм: вариантов использования, последовательности, состояния, деятельности, классов; потоков данных;
2. логическое проектирование, описание концептуальной модели данных, разработка базы данных в выбранной СУБД;
3. разработка интерфейса приложения, логики работы приложения: оформление заявок на услуги, вывод отчетной документации и т.д.;
4. администрирование приложения, разработка панели администратора;
5. разработка комплекта тестирования информационной системы.

Срок сдачи: « » 20 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И.Г. Зорина

« » 2023 г.

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.А.Аверин

« » 2023г.

Магнитогорск, 2023

Содержание

[ВВЕДЕНИЕ 4](#_Toc154569435)

[1 ТЕОРЕТИКО-ИНФОРМАЦИОННЫЙ АНАЛИЗ ПРИЛОЖЕНИЯ ПО УЧЕТУ КОНТИНГЕНГТА ОБУЧАЮЩИХСЯ 6](#_Toc154569436)

[1.1 Анализ предметной области 6](#_Toc154569437)

[1.2 Анализ и оценка существующих программных продуктов 6](#_Toc154569438)

[1.3 Постановка цели и задач 7](#_Toc154569439)

[2 ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_Toc154569440)

[2.1 Моделирование программного продукта 8](#_Toc154569441)

[2.2 Обоснование выбора средств разработки 12](#_Toc154569442)

[2.3 Логическое и физическое проектирование базы данных 13](#_Toc154569443)

[2.4 Разработка базы данных 14](#_Toc154569444)

[2.5 Логическая и физическая структура сайта 16](#_Toc154569445)

[2.7 Проектирование пользовательского интерфейса 18](#_Toc154569446)

[2.8 Администрирование приложения 21](#_Toc154569447)

[3 ОПИСАНИЕ РАБОТЫ ПРОГРАММНОГО ПРОДУКТА 23](#_Toc154569448)

[3.1 Руководство пользователя 23](#_Toc154569449)

[4 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ 26](#_Toc154569450)

[4.1 Общие положение информационной безопасности, уровни информацинной безопасности 26](#_Toc154569451)

[Федеральный закон 187 «О безопасности критической информационной инфраструктуры Российской Федерации» 28](#_Toc154569452)

[4.2 Виды угроз информационной безопасности 28](#_Toc154569453)

[4.3 Меры безопасности проекта 30](#_Toc154569454)

[5 УПРАВЛЕНИЕ ПРОЕКТОМ 32](#_Toc154569455)

[5.1 Понятие проекта. Необходимость управления проектом 32](#_Toc154569456)

[5.2 Постановка проекта по SMART 33](#_Toc154569457)

[5.4 Устав проекта 35](#_Toc154569458)

[5.5 Планирование проекта 36](#_Toc154569459)

[5.6 Управление стоимостью 38](#_Toc154569460)

[5.7 Управление рисками проекта 40](#_Toc154569461)

[ЗАКЛЮЧЕНИЕ 42](#_Toc154569462)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧИКОВ 43](#_Toc154569463)

[ПРИЛОЖЕНИЕ А 45](#_Toc154569464)

# ВВЕДЕНИЕ

В условиях развития современного общества информационные технологии глубоко проникают в жизнь людей. Всё больше процессов автоматизируются, облегчаются. уменьшения степени этого участия или трудоёмкости выполняемых операций.

Сегодняшний мир находится на пороге четвертой промышленной революции, которая будет характеризоваться внедрением и интеграцией технологий искусственного интеллекта, Интернета вещей и киберфизических систем в повседневную жизнь. Одной из главных сфер применения этих технологий является создание умных домов, способных автоматизировать повседневные процессы, экономить ресурсы и обеспечивать комфорт и безопасность.

В современном мире технологии умного дома становятся все более популярными и востребованными. Они предоставляют возможность автоматизации и оптимизации повседневных бытовых процессов, экономии времени и ресурсов, а также обеспечения комфорта и безопасности жилища. Веб-сайт умный дом является одним из ключевых элементов системы умного дома, предоставляя пользователю доступ к управлению устройствами и функциями умного дома из любой точки мира, где есть интернет-соединение.  
Настоящий проект направлен на разработку веб-сайта умного дома, который будет предоставлять пользователям широкий спектр возможностей для управления устройствами, контроля над домом и его безопасностью, а также для мониторинга и анализа данных о потреблении ресурсов.

Умный дом - это система автоматизации бытовых устройств, таких как освещение, отопление, вентиляция, кондиционирование, безопасность, мультимедиа и другие, которые работают в соответствии с заданными пользователем параметрами. Веб-сайт “Умный дом”, в свою очередь, является инструментом для управления этой системой. Он предоставляет пользователю возможность контролировать работу устройств, отслеживать их состояние, устанавливать различные сценарии и автоматизировать процессы.

Цель данного проекта - разработка веб-сайта, который будет являться удобным, функциональным и безопасным инструментом для управления системой умного дома. Для достижения этой цели необходимо решить ряд задач, связанных с разработкой дизайна, структуры, функционала и безопасности сайта, а также интеграции с различными устройствами и платформами.

Исходя из поставленной цели можно выделить следующие задачи:

* выполнить анализ предметной области;
* провести анализ и оценку существующих программных продуктов;
* выполнить постановку задачи;
* выполнить моделирование программного продукта;
* выбрать средства разработки;
* разработать базу данных;
* разработать физическую и логическую структуру сайта;
* выполнить проектирование меню и пользовательского интерфейса;
* разработать клиентскую часть приложения;
* сделать описание запросов при разработке приложения;
* спроектировать отчеты;
* выполнить администрирование программного продукта;
* разработать руководство пользователя;
* разработать руководство программисту;
* учесть меры безопасности;
* выполнить управление проектом.

# 1 ТЕОРЕТИКО-ИНФОРМАЦИОННЫЙ АНАЛИЗ ПРИЛОЖЕНИЯ ПО УЧЕТУ КОНТИНГЕНГТА ОБУЧАЮЩИХСЯ

**1.1** **Анализ предметной области**

История умного дома началась еще в XIX столетии при создании первых систем жизнеобеспечения. Все существующие на сегодняшний день технологии основываются на разработках, которые были созданы на рубеже XX столетия. Эти технологии медленно видоизменялись по мере продвижения вперед технического прогресса. Оснащение дома необходимым техническим оборудованием привело к современному виду жилого дома: кухня со встроенным оборудованием, включая газовую или электрическую плиту и холодильник, ванная и туалет с современной сантехникой и т.д. Дальнейшее усовершенствование домашнего оборудования привело к внедрению в обиход телефона, радио и телевидения. Этот процесс протекал одновременно с созданием проектов будущего жилья с принципиальной переменой представления о нем .

Термин «умный дом» возник от английского intelligent building. Понимая под этим рациональное использование агрегата «дом», не затрагивая понятие «машина».

При этом акцентирование производится на интерактивность. Эта интерактивная система состоит в наши дни из нескольких различных сетей: электросеть, телефонная и телевизионная, интернет сети. Уже сейчас производится объединение сетей, создавая оптимальный интерфейс между сетью и конечным устройством и, главное, интерфейс между человеком и машиной в виде графического пользовательского интерфейса, сенсорного экрана или идентификации голоса или жеста.

**1.2 Анализ и оценка существующих программных продуктов**

Для разработки веб-сайта “Умный дом” необходимо провести анализ и оценку существующих программных продуктов в данной области. Это позволит выявить основные тенденции, определить преимущества и недостатки существующих решений, а также учесть полученные результаты при разработке собственного продукта.

На сегодняшний день на рынке представлено множество программных продуктов для управления системами умного дома. Некоторые из них являются универсальными и могут работать с различными устройствами, другие же специализированы на определенных брендах или технологиях.

Одним из наиболее популярных универсальных решений является платформа Amazon Alexa. Она позволяет управлять устройствами различных производителей через голосовые команды и интегрируется с множеством сервисов, таких как Spotify, Netflix, YouTube и другими. Однако, недостатком этой платформы является то, что она работает только на устройствах Amazon и не имеет веб-интерфейса.

Другим популярным решением является система Google Home, которая также поддерживает управление устройствами различных производителей. Она имеет удобный веб-интерфейс и мобильное приложение, однако, как и Amazon Alexa, работает только с устройствами Google.

Также стоит отметить такие решения, как Apple HomeKit, Xiaomi Mi Home, Samsung SmartThings и другие. Все они имеют свои преимущества и недостатки, но в целом, их функционал примерно одинаков.

## **1.3 Постановка цели и задач**

Постановка целей и задач помогает структурировать работу, определить приоритеты и контролировать выполнение проекта. Без четкого понимания того, что нужно сделать, проект может превратиться в хаотичное нагромождение задач, которое в итоге не приведет к желаемому результату.

Кроме того, определение целей и задач позволяет оценить ресурсы, необходимые для реализации проекта, и определить, какие дополнительные ресурсы могут потребоваться. Это также помогает определить, какие методы и подходы будут наиболее эффективными для достижения поставленных целей.

Таким образом, постановка целей и задач является ключевым этапом в процессе управления проектами, поскольку она помогает определить направление работы, распределить ресурсы и контролировать процесс выполнения проекта.

Задачи, которые необходимо выполнить:

* разработать регистрацию пользователей;
* разработать возможность добавлять устройства;
* осуществить возможность редактировать и удалять устройства;
* разработать логистику сайта со стороны администратора.

Сроки выполнения проекта определяются в соответствии с академическим календарем учебного заведения.

Результатом проекта должно быть работающее и адаптированное к потребностям глав учебного учреждения сайт.

# 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Моделирование программного продукта

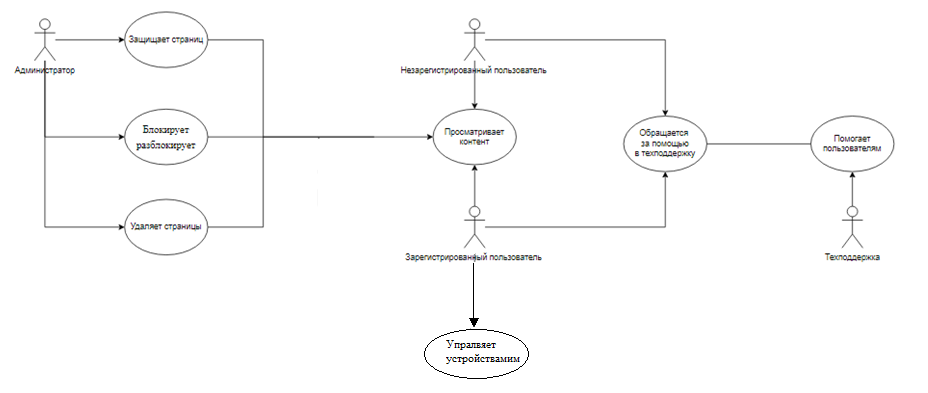
Моделирование — это представление объекта моделью для получения информации о нём путём проведения экспериментов с его моделью.

Под термином моделирование обычно понимают процесс создания точного описания системы, метод познания, состоящий в создании и исследовании моделей.

Моделирование облегчает изучение объекта с целью его создания, дальнейшего преобразования и развития. Оно используется для исследования существующей системы, когда реальный эксперимент проводить нецелесообразно из-за значительных финансовых и трудовых затрат, а также при необходимости проведения анализа проектируемой системы, то есть которая ещё физически не существует в данной организации.

Диаграмма прецедентов предоставляет визуальное представление функциональности системы и взаимодействия с актерами (рисунок 1). Она помогает команде разработчиков и заинтересованным сторонам лучше понять требования и задачи, которые должны быть реализованы в системе. Она также может служить основой для определения тестовых сценариев и функционального тестирования системы.

Важно отметить, что диаграмма прецедентов не описывает внутреннюю структуру системы или детали реализации. Она фокусируется исключительно на функциональных возможностях и взаимодействии с актерами.

Рисунок 1 – Диаграмма прецедентов

IDEF1 (рисунок 2) — это методология моделирования информационных потоков внутри системы, позволяющая отображать и анализировать их структуру и взаимосвязи. Одна из основных ценностей и причин стремительного развития информационных технологий — это высочайший темп наращивания человечеством информации. Хранимая и обрабатываемая, она даёт возможность прогресса, и потребность в том, чтобы наращивать возможности по обработке.

Стандарт IDEF1 был разработан как инструмент для анализа и изучения взаимосвязей между информационными потоками в рамках коммерческой деятельности предприятия. Целью подобного исследования является дополнение и структуризация существующей информации и обеспечение качественного менеджмента информационными потоками.

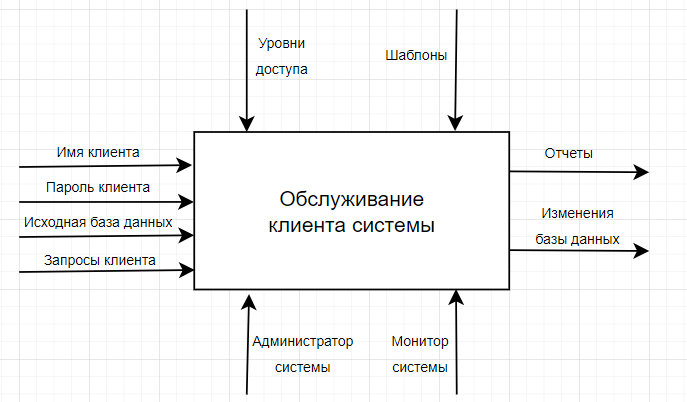


Рисунок 2 – Диаграмма IDEF1

Диаграмма последовательности (рисунок 3) — диаграмма взаимодействия, в которой основной акцент сделан на упорядочении сообщений во времени. Диаграмма последовательности — это способ описания поведения системы "на примерах". Диаграмма последовательности — диаграмма, предназначенная для представления взаимодействия между элементами модели программной системы в терминологии линий жизни и сообщений между ними.

Графически диаграмма последовательности имеет два измерения. Одно — слева направо в виде вертикальных линий, каждая из которых соответствует линии жизни отдельного участника взаимодействия. Второе — вертикальная временная ось, направленная сверху вниз. Начальному моменту времени соответствует самая верхняя часть диаграммы. Реализация взаимодействия моделируется посредством сообщений, которые передаются между различными линиями жизни. Сообщения изображаются в виде стрелок различной формы и образуют некоторый порядок относительно времени своей передачи.

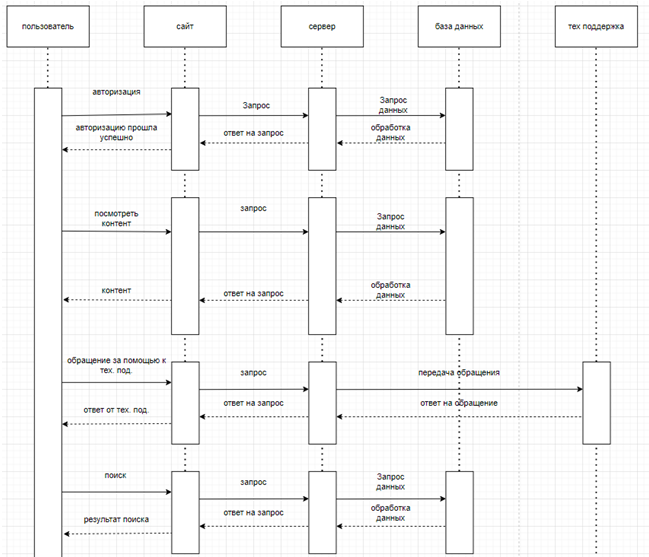
Рисунок 3 – Диаграмма последовательности

Диаграмма деятельности (рисунок 4) используется для моделирования процесса выполнения операций в языке UML. Графическая нотация во многом похожа на нотацию диаграммы состояний, поскольку на диаграммах деятельности также присутствуют обозначения состояний и переходов. Отличие заключается в семантике состояний, которые используются для представления деятельности и действий, а также в отсутствии на переходах сигнатуры событий. Каждое состояние на диаграмме деятельности соответствует выполнению некой операции, а переход в следующее состояние происходит только после завершения выполнения этой операции. Диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия или деятельности, а дугами – переходы от одного состояния действия к другому.

Диаграммы деятельности – частный случай диаграмм состояний. Они позволяют реализовать в языке UML особенности процедурного и синхронного управления, обусловленного завершением внутренних действий и деятельности. Основным направлением использования диаграмм деятельности является визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения. При этом каждое состояние может являться выполнением операции определенного класса либо ее части, позволяя использовать диаграммы деятельности для описания реакций на внутренние события системы.

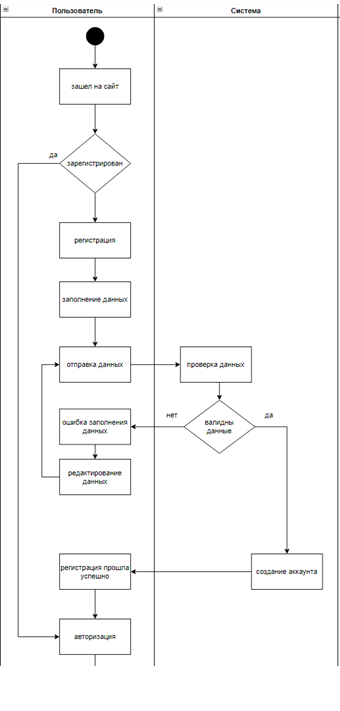


Рисунок 4 – Диаграмма деятельности

## **2.2 Обоснование выбора средств разработки**

В процессе создания веб–приложения были выбранные различные средства для разработки.

Выбранные средства для разработки:

* MySQL;
* Laravel.

Laravel является одним из самых популярных в мире PHP–фреймворков, предназначенный для разработки веб–приложений. Имя при себе ряд плюсов, стоит выделить некоторые их низ:

* Laravel – полнофункциональный фреймворк. Он содержит все программные подсистемы необходимые для разработки веб–приложения. После его установки нет необходимости доустанавливать ещё какие–либо программы или приложения;
* у Laravel низкий порог вхождения. Для программирования веб–приложений хватает базовых знаний PHP и основ веб–разработки. Он не требует сложного конфигурирования и готов к работе сразу после установки. Отдельные модули, составляющие код, не требуется явно связывать друг с другом, достаточно «разложить» их по нужным папкам и веб–приложение будет прекрасно работать;
* Laravel содержит множество дополнительных библиотек, расширяющих функциональность фреймворка, и программ, помогающих в работе.

Таким образом, фреймворк Laravel представляет из себя качественный продукт, которые облегчит разработку веб–приложения, поможет сжать код и обеспечить разработку всего требуемого функционала для веб–приложения.

MySQL – это свободно распространяемая система управления базами данных. MySQL имеет клиент–серверную архитектуру: к серверу MySQL могут обращаться различные клиентские приложения, в том числе с удаленных компьютеров.

MySQL имеет ряд плюсов, за счет которых она была выбрана для разработки:

* MySQL представляет из себя открытый код. Любой желающий может бесплатно скачать программу и при необходимости доработать ее;
* MySQL – кроссплатформенная система. Ее можно использовать практически во всех современных операционных системах;
* MySQL имеет множество программных интерфейсов, благодаря которым к базе данных MySQL могут подключаться приложения, созданные с помощью различных языков, таких как: C/C++, PHP Python и так далее;
* MySQL имеет отличные технические характеристики: многопоточность, многопользовательский доступ, быстродействие, масштабируемость
* MySQL имеет развитую систему обеспечения безопасности и разграничения доступа на основе системы привилегий

MySQL представляет из себя прекрасное средство для разработки веб–приложения, которая содержит сбалансированный набор функций, возможностей и преимущества. MySQL идеально подходит для быстродействия при взаимодействии данных, он понятен и прост в управлении системе управления базами данных.

Bootstrap – это бесплатный CSS–фреймворк с открытым исходным кодом, предназначенный для быстрой верстки адаптивных интерфейсов и веб–приложений с ориентацией на мобильные устройства.

На данный момент Bootstrap является одним из самых распространённых фреймворков для проектирования веб–приложений. Он пользуется популярностью не только у разработчиков сайтов, но и у заказчиков.

**2.3 Логическое и физическое проектирование базы данных**

База данных — это программа, которая позволяет хранить и обрабатывать информацию в структурированном виде. База данных — это отдельная независимая программа, которая не входит в состав языка программирования. В базе данных можно сохранять любую информацию, чтобы позже получать к ней доступ.

Проектирование базы данных заключается в многоступенчатом описании будущей базы данных с различной степенью детализации и формализации, в ходе которого производится уточнение и оптимизация ее структуры.

Логическое проектирование — это процесс создания модели используемой на предприятии информации на основе выбранной модели организации данных, но без учета типа целевой системы управления базами данных и других физических аспектов реализации.

Второй этап проектирования базы данных называется логическим проектированием базы данных. Его цель состоит в создании логической модели данных для исследуемой части предприятия. Концептуальная модель данных, созданная на предыдущем этапе, уточняется и преобразуется в логическую модель данных. Логическая модель данных учитывает особенности выбранной модели организации данных в целевой системе управления базами данных (например, реляционная модель).

Если концептуальная модель данных не зависит от любых физических аспектов реализации, то логическая модель данных создается на основе выбранной модели организации данных целевой системы управления базами данных. Иначе говоря, на этом этапе уже должно быть известно, какая система управления базами данных будет использоваться в качестве целевой — реляционная, сетевая, иерархическая или объектно-ориентированная. Однако на этом этапе игнорируются все остальные характеристики выбранной системы управления базами данных, например, любые особенности физической организации ее структур хранения данных и построения индексов.

В процессе разработки логическая модель данных постоянно тестируется и проверяется на соответствие требованиям пользователей.

Созданная логическая модель данных является источником информации для этапа физического проектирования и обеспечивает разработчика физической базы данных средствами поиска компромиссов, необходимых для достижения поставленных целей, что очень важно для эффективного проектирования. Логическая модель данных играет также важную роль на этапе эксплуатации и сопровождения уже готовой системы. При правильно организованном сопровождении поддерживаемая в актуальном состоянии модель данных позволяет точно и наглядно представить любые вносимые в базу данных изменения, а также оценить их влияние на прикладные программы и использование данных, уже имеющихся в базе.

Физическое проектирование базы данных — это процесс создания описания конкретной реализации базы данных, размещаемой во вторичной памяти. На этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты.

На основе ситуации описания проекта проектируется модель базы данных и составляется схема логической модели. Концептуальная модель

**2.4** **Разработка базы данных**

Создание базы данных - это процесс разработки и организации структурированного хранилища данных, которое позволяет эффективно хранить, управлять и извлекать информацию. Это включает в себя определение схемы базы данных, создание таблиц и связей между ними, а также установку правил целостности данных.

Создание таблицы «Пользователь» Характеристики таблицы представлены в таблице1

public function up(): void

    {

        Schema::create('users', function (Blueprint $table) {

            $table->id();

            $table->string('firstname',255);

            $table->string('lastname',255);

            $table->string('patronymic',255);

            $table->enum('gender', ['М', 'Ж']);

            $table->string('image',255)->default('images/static/default.jfif');

            $table->string('role',255)->default('user');

            $table->string('email')->unique();

            $table->timestamp('email\_verified\_at')->nullable();

            $table->string('password');

            $table->rememberToken();

            $table->timestamps();

        });

    }

Таблица 1 Характеристики полей таблицы «users»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор пользователя (первичный ключ) |
| name\_first | varchar(255) | имя пользователя |
| name\_last | varchar(255) | фамилия пользователя |
| patronymic | varchar(255) |  |
| password | char(32) | пароль пользователя |
| role | varchar(255) |  |
| gender | Enum['М', 'Ж'] | пол |
| image | varchar(255) |  |
| email | varchar(255) |  |
| email\_verified\_at | timestamp |  |
| password | varchar(255) | пароль |
| remember\_token | varchar(100) | токен |
| created\_at | timestamp | дата создания |
| updated\_at | timestamp | дата обновления |

Создание таблицы «Категория» Характеристики таблицы представлены в таблице2

public function up(): void

    {

        Schema::create('categories', function (Blueprint $table) {

            $table->id();

            $table->string('name',255);

            $table->longText('description',255);

            $table->string('image',255)->default('images/static/device.jpg');

            $table->timestamps();

        });

    }

Таблица 2 Характеристики полей таблицы «categories»

|  |  |  |
| --- | --- | --- |
| id | int | идентификатор пользователя (первичный ключ) |
| name | varchar(255) | название |
| description | varchar(255) | описание |
| image | varchar(255) | изображение |
| created\_at | timestamp | дата создания |
| updated\_at | timestamp | дата обновления |

**2.5 Логическая и физическая структура сайта**

Структура сайта — это логическое построение всех страниц сайта, категорий и подкатегорий. Это логическая схема, в соответствии с которой все страницы и разделы сайта расположены относительно друг друга и принцип, по которому они друг с другом взаимосвязаны.

От того, насколько проста и понятна структура сайта, зависит то, как быстро посетитель найдет нужную ему информацию.

Основные модели логической структуры:

* Строгая линейная структура ресурса. Используется достаточно редко, поскольку не обеспечивает предоставление необходимого удобства – пользователи могут переходить только от фактической страницы к следующей. Очевидная простота в данном случае противопоставлена свободе выбора, что не каждому человеку может понравиться.
* Логическая структура ресурса, так называемая «решетка». Наиболее часто встречается при создании интернет-магазинов и представляет собой линейную структуру, функционирующую в двух направлениях, подразумевающих присутствие вертикальной и горизонтальной связи между страницами.
* Логическая структура сайта, так называемое «дерево». Наиболее распространенная модель, выстраиваемая в определенной иерархической последовательности. Посетители получают полную свободу управления глубиной анализа ресурса и могут заходить на страницы любого уровня.

Логическая структура, так называемая «паутина». Ее может быть связано с желанием обеспечить выразительность предоставления информации или обычным стечением обстоятельств. Определенная страница может ссылаться на другой источник информации, что и обеспечивает максимальную полноту получения данных.

Физическая структура сайта описывает размещение файлов и папок на диске компьютера. Правильная организация файлов и папок поможет в дальнейшей работе с сайтом.

Физическая и логическая структуры не всегда совпадают, потому что обычно физическая структура сайта разрабатывается, основываясь на удобстве размещения файлов. Но более или менее точное сохранение логических разделов в физической структуре сайта позволяет упростить работу по изменению обновлению информационных материалов (рисунок 5).

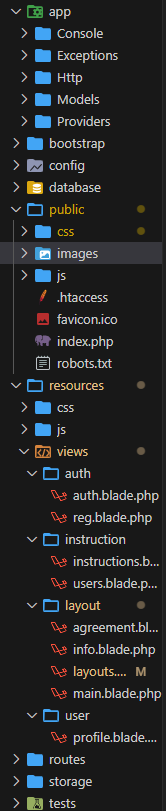


Рисунок 5 – Физическая структура

**2.7 Проектирование пользовательского интерфейса**

Пользовательский интерфейс — это набор программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. Основу такого взаимодействия составляют диалоги. Под диалогом в данном случае понимают регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи. Каждый диалог состоит из отдельных процессов ввода/вывода, которые физически обеспечивают связь пользователя и компьютера. Обмен информацией осуществляется передачей сообщения.

Интерфейс — это место соприкосновения двух функциональных объектов. Если говорить более узко, то интерфейс — это «мост», посредник между человеком, программами и машинами, иными системами. Это инструменты взаимодействия, с помощью которых одна система контактирует и строит диалог с другой.

Качество и удобство интерфейса пользователя играют важнейшую роль в определении уровня удовлетворенности пользователей. Что же делает хороший интерфейс?

Во-первых, он должен быть интуитивно понятным. Пользователь должен легко понимать, как использовать интерфейс и выполнять нужные действия без особых усилий. Отображение информации и доступ к функционалу должны быть логичными и простыми.

Во-вторых, интерфейс должен быть информативным. Вся необходимая информация должна быть предоставлена пользователю в понятном и доступном виде. Важные элементы интерфейса, такие как кнопки, ссылки и подсказки, должны быть ясно видимыми и четко описанными.

В-третьих, интерфейс пользователя должен быть эстетически приятным и привлекательным. Дизайн должен соответствовать общему стилю и образу компании или бренда. Цветовая гамма, шрифты и графические элементы должны гармонично сочетаться и быть привлекательными для глаз пользователя.

В-четвертых, интерфейс должен быть адаптивным и отзывчивым. Он должен быть способен адаптироваться к различным устройствам и экранам, чтобы обеспечить комфортное использование на мобильных телефонах, планшетах и компьютерах. Кроме того, интерфейс должен реагировать на действия пользователя быстро и без задержек.

В-пятых, интерфейс пользователя должен обеспечивать хорошую функциональность и навигацию. Пользователь должен легко находить нужные функции и возможности, а также иметь возможность легко возвращаться к предыдущим шагам. Навигационное меню и разделение информации на логические блоки помогают пользователям легко перемещаться по интерфейсу и быстро находить нужное.

Хороший интерфейс пользователя должен постоянно обновляться и улучшаться на основе обратной связи от пользователей. Слушание и анализ отзывов пользователей позволяет выявлять недостатки интерфейса и вносить изменения, чтобы сделать его еще более удобным и функциональным.

Интерфейс пользователя является важным элементом любой системы, который влияет на ее успешность и эффективность. Создание хорошего интерфейса требует тщательного анализа требований и потребностей пользователей, а также применения лучших практик в области дизайна и удобства использования.

Разработка интерфейса пользователя является одной из важных задач в сфере создания программного обеспечения. UI представляет собой то, через что пользователи взаимодействуют с приложением или веб-сайтом, поэтому его дизайн и удобство использования имеют большое значение

Перед началом разработки интерфейса пользователя необходимо провести исследование и анализ целевой аудитории. Это поможет понять, какие функции необходимо предоставить пользователям, какие проблемы они могут испытывать и как сделать интерфейс максимально интуитивным и удобным в использовании.

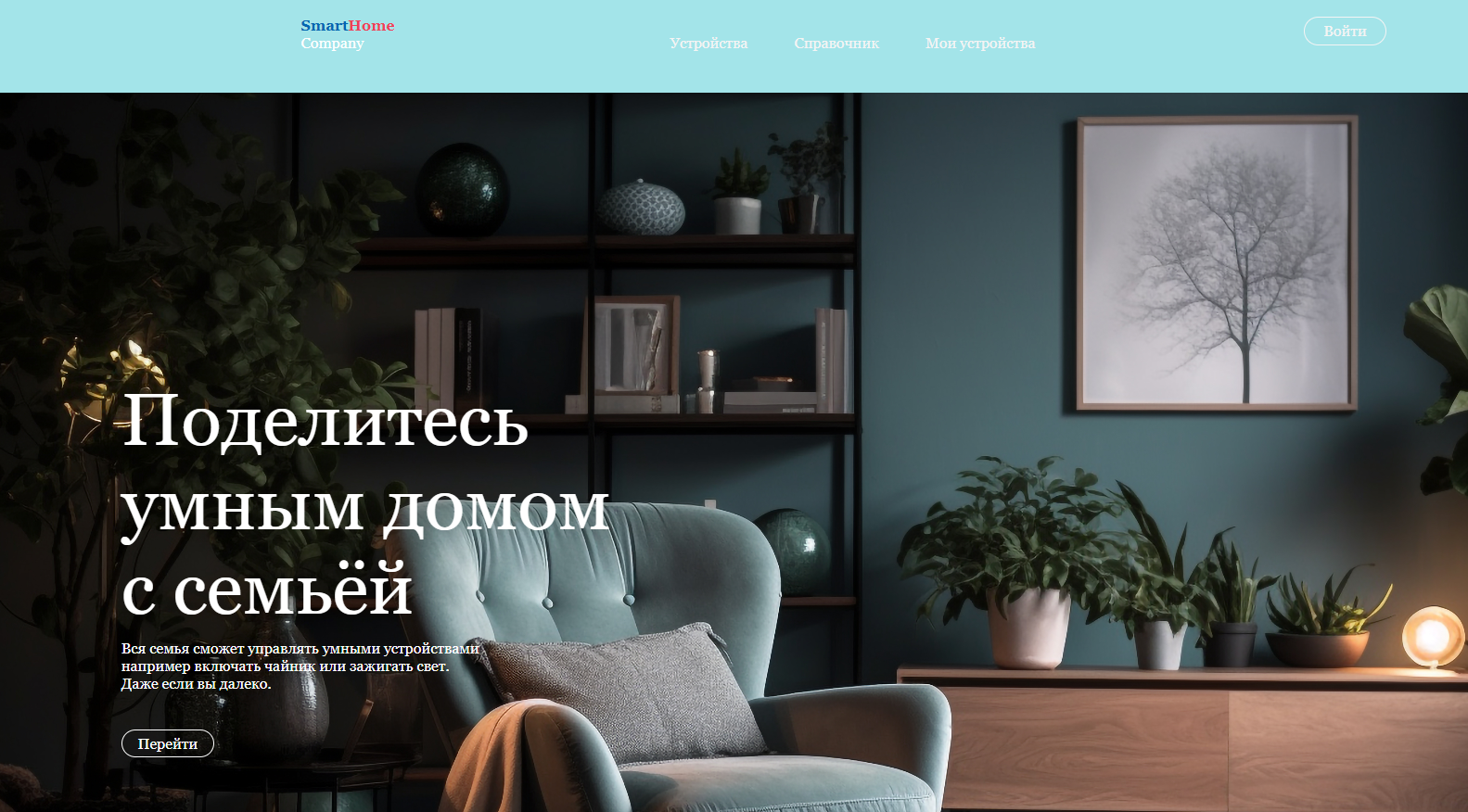
Далее начинается проектирование интерфейса. Оно включает разработку визуальной составляющей, такой как цветовая гамма, типографика, иконки и графика. Учитывается анимация и стилизация элементов интерфейса, которые помогают пользователям взаимодействовать с приложением.

Разработка интерфейса - это не только визуальные аспекты. Важным является также разработка архитектуры интерфейса, которая определяет, какие элементы будут включены в интерфейс и как они будут располагаться на странице или экране. Это включает в себя размещение кнопок, форм, меню и других элементов, которые пользователи будут использовать при взаимодействии с приложением.

Одной из важных задач при разработке интерфейса пользователя является тестирование и улучшение. В процессе разработки необходимо проводить тестирование с реальными пользователями, чтобы выявить проблемы и недостатки интерфейса. На основе обратной связи пользователей можно вносить изменения и улучшать интерфейс, чтобы сделать его более удобным и интуитивным.

В современных технологиях разработки пользовательских интерфейсов широко используется методика адаптивного дизайна. Это позволяет создавать интерфейсы, которые оптимально отображаются на различных устройствах и экранах, таких как компьютеры, смартфоны и планшеты.

Разработка интерфейса пользователя является неотъемлемой частью процесса создания программного обеспечения. Она включает в себя не только визуальные аспекты, но и архитектуру и функциональность. Качественный и удобный интерфейс помогает пользователям эффективно взаимодействовать с приложением и достигать своих целей(Рис.6).

Рисунок 6 – Интерфейс на примере курсового сайта

**2.8** **Администрирование приложения**

Права доступа определяют набор действий (например, чтение, запись, выполнение), разрешённых для выполнения субъектам (например, пользователям системы) над объектами данных. Для этого требуется некая система для предоставления субъектам различных прав доступа к объектам. Это система разграничения доступа субъектов к объектам, которая рассматривается в качестве главного средства защиты от несанкционированного доступа к информации или порчи самой системы.

Функции системы разграничения доступа:

* реализация правил разграничения доступа субъектов и их процессов к данным;
* реализация правил разграничения доступа субъектов и их процессов к устройствам создания твёрдых копий;
* изоляция программ процесса, выполняемого в интересах субъекта, от других субъектов.

Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, с носителем, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации.

В таблице 3 представлено разграничение прав доступа данного проекта.

Таблица 3 – Разграничение прав доступа

|  |  |  |  |
| --- | --- | --- | --- |
| Пользователи | Email | Пароль | Права |
| Admin | admin@mail.ru | admin123 | Редактирование контента на сайте |
| User | user@mail.ru | user123 | Использование функционала для управления умным домом |
| Guest |  |  | Просмотр главной страницы |

# 3 ОПИСАНИЕ РАБОТЫ ПРОГРАММНОГО ПРОДУКТА

**3.1**  **Руководство пользователя**

"Инструкция по использованию программы - это краткое руководство, предназначенное для пользователей программы, чтобы они могли корректно и эффективно использовать все ее функции и возможности.

В инструкции пользователю должна содержаться всесторонняя и пошаговая информация, которая поможет пользователю понять основную функциональность программы и научиться работать с ней без необходимости обращаться к дополнительным материалам или спецификациям. Вся необходимая информация для этого извлекается из полной документации, но представляется в удобной форме пользователю.

Первая часть инструкции - описательная. В ней должны присутствовать следующие пункты:

* Наименование программы: указывается название программы, чтобы пользователь мог легко ее идентифицировать.
* Краткое описание программы: дается краткое описание основных функций программы и ее назначения.
* Перечень выполняемых функций: перечисляются все функции, которые может выполнять программа, чтобы пользователь знал, какие возможности у него есть.
* Характеристика метода решения задачи: предоставляется краткое описание метода (или методов), которые использовались для решения поставленной задачи, а также указываются их достоинства и недостатки.
* Библиографическая ссылка: указывается полная библиографическая ссылка на описание метода, чтобы пользователь мог ознакомиться с ним подробнее, если потребуется.
* Входные и выходные данные: описываются данные, которые программа принимает на входе и данные, которые она выдает на выходе.
* Структура базы данных: если программа использует базу данных, то описывается структура базы данных и все ее таблицы.

Вторая часть инструкции должна содержать описание процесса работы с программой. В ней должны присутствовать следующие пункты:

* Запуск программы: описывается, как запустить программу, чтобы пользователь мог начать работу с ней.
* Режимы работы программы: описываются все режимы работы программы, чтобы пользователь мог выбрать нужный.
* Печати и диагностические сообщения: описываются все печати и сообщения, которые могут появляться в процессе работы программы, чтобы пользователь мог корректно на них реагировать.
* Подготовка и ввод данных: описывается, как подготовить и ввести начальные данные в программу.
* Реакция на запросы программы: описывается, как пользователь должен реагировать на запросы программы, чтобы получить желаемый результат.
* Работа в исключительных ситуациях: описывается, как пользователь должен действовать в случае возникновения исключительных ситуаций или ошибок.
* Восстановление работы программы: описывается, как восстановить работу программы в случае ее аварийного завершения.
* Получение требуемого результата: объясняется, как получить конечный результат работы с программой.
* Правильное завершение работы с программой: описывается, как корректно закончить работу с программой.
* Другие сведения: предоставляются дополнительные сведения, которые могут быть полезными пользователю во время работы с программой.

Инструкция пользователю представляет собой независимый документ, содержащий всю необходимую и понятную информацию для пользователя программы, чтобы он мог успешно и эффективно ею пользоваться.

Руководство для обычного пользователя

При посещении веб-сайта обычным пользователем возникает некоторое количество ограничений. Например, неавторизованный пользователь не имеет возможности отправить заявку на создание свое странички продукта или оставить свой отзыв на сайте. Также у него отсутствует доступ к личному кабинету, поскольку учетной записи просто не существует. Чтобы получить возможность зарегистрироваться или войти в систему, необходимо нажать на соответствующую кнопку "Войти" или "Регистрация" в меню сайта (Рисунок 7).



Рисунок 7 – Личный кабинет

После того, как пользователь зарегистрировался, он может зайти в личный кабинет (Рисунок 8)

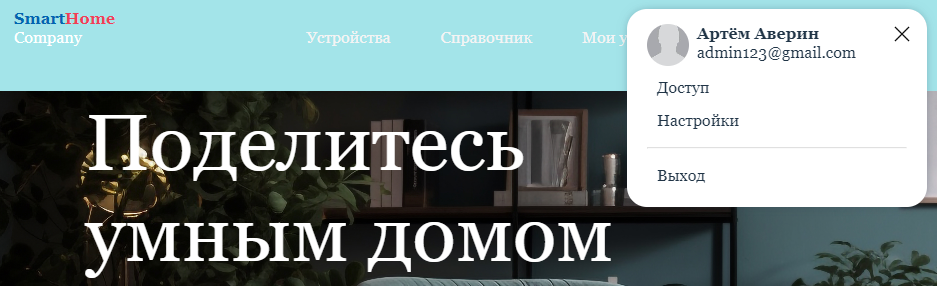


Рисунок 8 – Личный кабинет

Функционал во вкладке “личном кабинете” представляет собой набор возможностей, предоставляемых пользователю для удобного управления своим личными данными. Такой функционал дает пользователю доступ к различным инструментам, позволяющим контролировать свои данные и обновлять персональную информацию.

# 4 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

**4.1** **Общие положение информационной безопасности, уровни информацинной безопасности**

Информационная безопасность – это сохранение и защита информации, а также ее важнейших элементов, в том числе системы и оборудование, предназначенные для использования, сбережения и передачи этой информации. Иначе, это набор технологий, стандартов и методов управления, которые необходимы для защиты информационной безопасности.

Главной целью информационной безопасности является защита информационных данных и поддерживание инфраструктуру от случайного или специального вмешательства.

Информационная безопасность придерживается трех главных принципов:

* конфиденциальность. Это гарантирует достаточный уровень безопасности с данными пользователей, которые пользуются веб–приложением;
* целостность. Она обеспечивает предотвращение искажения информации;
* доступность. Доступность обеспечивает надежный и эффективный доступ к информации уполномоченных лиц.

Угроза объекту информационной безопасности есть совокупность факторов и условий, возникающих в процессе взаимодействия различных объектов (их элементов) и способных оказывать негативное воздействие на конкретный объек информационной безопасности. Негативные воздействия различаются по характеру наносимго вреда, а именно: по степени изменения свойств объекта безопасности и возможности ликвидации последствий проявления угрозы.

Выделяется несколько уровней информационной безопасности:

* законодательный. Он включается в себя совокупность мер, которые направлены на формирование и поддержку в обществе отрицательного отношения к киберприступникам и нарушителям информационной безопасности. Так же включается в себя подготовку отечественных норм и стандартов в соответствие с международными регламентами.
* административный. Административной безопасностью занимаются руководство разработчиков. Главный документ на данном уровне – политика безопасности, состоящая из решений по управлению, необходимых для защиты данных, техсредств, которые с ней ассоциированы.
* процедурный. Сюда входят некоторые меры безопасности, такие как обеспечение физической защиты техсредств, программ, документации, поддержка работоспособности, реагирование на негативные события в режиме безопасности и стратегия восстановительных работ.
* программно–технический уровень. Здесь используются основные механизмы безопасности, такие как обеспечение максимальной доступности, протоколирование, управление доступом, идентификация и проверка подлинности пользователей.

В России действуют законы, где описано, как правильно работать с информацией: кто отвечает за ее сохранность, как ее собирать, обрабатывать, хранить и распространять.

Федеральный закон "Об информации, информационных технологиях и о защите информации" является главный закон об информации в России. Он определяет ключевые термины, например, говорит, что информация — это любые данные, сведения и сообщения, представляемые в любой форме. Также там описано, что такое сайт, электронное сообщение и поисковая система. Именно на этот закон и эти определения нужно ссылаться при составлении документов по информационной безопасности.

В 149 федеральном законе сказано, какая информация считается конфиденциальной, а какая – общедоступной, когда и как можно ограничивать доступ к информации, как происходит обмен данными. Также именно здесь прописаны основные требования к защите информации и ответственность за нарушения при работе с ней.

Ключевые моменты закона об информационной безопасности:

* нельзя собирать и распространять информацию о жизни человека без его согласия;
* все информационные технологии равнозначны — нельзя обязать компанию использовать какие-то конкретные технологии для создания информационной системы;
* есть информация, к которой нельзя ограничивать доступ, например сведения о состоянии окружающей среды;
* некоторую информацию распространять запрещено, например ту, которая пропагандирует насилие или нетерпимость;
* тот, кто хранит информацию, обязан ее защищать, например, предотвращать доступ к ней третьих лиц.

Федеральный закон «О персональных данных» регулирует работу с персональными данными – личными данными конкретных людей. Его обязаны соблюдать те, кто собирает и хранит эти данные.

Ключевые моменты закона:

* перед сбором и обработкой персональных данных нужно спрашивать согласие их владельца;
* для защиты информации закон обязывает собирать персональные данные только с конкретной целью;
* если вы собираете персональные данные, то обязаны держать их в секрете и защищать от посторонних;
* если владелец персональных данных потребует их удалить, вы обязаны сразу же это сделать;
* если вы работаете с персональными данными, то обязаны хранить и обрабатывать их в базах на территории Российской Федерации. При этом данные можно передавать за границу при соблюдении определенных условий, прописанных в законе — жесткого запрета на трансграничную передачу данных нет.

Федеральный закон "О безопасности" от 28.12.2010 N 390 определяет основные принципы и содержание деятельности по обеспечению безопасности государства, общественной безопасности, экологической безопасности, безопасности личности, иных видов безопасности, предусмотренных законодательством Российской Федерации (далее - безопасность, национальная безопасность), полномочия и функции федеральных органов государственной власти, органов государственной власти субъектов Российской Федерации, органов местного самоуправления в области безопасности, а также статус Совета Безопасности Российской Федерации.

## Федеральный закон 187 «О безопасности критической информационной инфраструктуры Российской Федерации»

Этот закон касается компаний, которые работают в сферах, критически важных для жизни государства — таких, что сбой в их работе отразится на здоровье, безопасности и комфорте граждан России.

К таким сферам относится здравоохранение, наука, транспорт, связь, энергетика, банки, топливная промышленность, атомная энергетика, оборонная промышленность, ракетно-космическая промышленность, горнодобывающая промышленность, металлургическая промышленность и химическая промышленность. Также сюда относят компании, которые обеспечивают работу предприятий из этих сфер, например, предоставляют оборудование в аренду или разрабатывают для них программное обеспечение.

Если на предприятии из этой сферы будет простой, это негативно отразится на жизни всего государства. Поэтому к IT-инфраструктуре и безопасности информационных систем на этих предприятиях предъявляют особые требования.

**4.2 Виды угроз информационной безопасности**

Угрозы информационной безопасности — это разнообразные способы, которыми злоумышленники могут пытаться получить несанкционированный доступ к информации, нарушить работу компьютерных систем или использовать данные в своих интересах.

Важнейшей стороной обеспечения информационной безопасности является определение и классификация угроз. Угрозы безопасности информации — это некая совокупность факторов и условий, которые создают опасность в отношении защищаемой информации.

Для того чтобы определить угрозы, от которых необходимо обезопасить информацию, нужно определить объекты защиты. Ведь информация — это некоторые данные, носителями которых могут быть как материальные, так и нематериальные объекты. К примеру, носителями конфиденциальной информации могут быть документы, технические средства обработки и хранения информации и даже люди.

Утечки информации можно разделять на умышленные и случайные. Случайные утечки происходят из-за ошибок оборудования, программного обеспечения и персонала. Умышленные, в свою очередь, организовываются преднамеренно с целью получить доступ к данным, нанести ущерб. Потерю данных можно считать одной из основных угроз информационной безопасности. Нарушение целостности информации может быть вызвано неисправностью оборудования или умышленными действиями людей, будь то сотрудники или злоумышленники.

Документационными носителями информации могут быть проекты, бизнес-планы, техническая документация, контракты и договора, а также картотеки отдела кадров (персональные данные) и отдела по работе с клиентами. Отличительной их особенностью является зафиксированность данных на материальном объекте — бумаге.

Техническими средствами обработки и хранения информации являются персональные компьютеры, ноутбуки, серверы, сканеры, принтеры, а также съемные носители (переносные жесткие диски, флеш-карты, CD-диски, дискеты) и тому подобное. Информация в технических средствах хранится и обрабатывается в цифровом виде. Зачастую конфиденциальные данные отправляются через Интернет, например, по электронной почте. В сети они могут быть перехвачены злоумышленниками. Кроме того, при работе компьютеров из-за их технических особенностей обрабатываемые данные преобразуются в электромагнитные излучения, распространяющиеся далеко за пределы помещения, которые также могут быть перехвачены и использованы в недобросовестных целях.

Атака типа «человек посередине» (MitM) также известна как атака с перехватом. Она происходит, когда злоумышленник пытается перехватить связь между двумя сторонами (веб-сервером и клиентским браузером), чтобы следить за жертвой, украсть личную информацию или учетные данные, передаваемые по сети. Для выполнения атаки хакер ищет незащищенные сетевые соединения в общедоступных сетях Wi-Fi. Для предотвращения атак MitM при доступе к сети из незащищенной общедоступной точки доступа Wi-Fi используют виртуальную частную сеть (VPN), обеспечивающую безопасное соединение с шифрованием данных.

Атаки типа «отказ в обслуживании» (DoS, DDoS) перегружают системы, серверы, сайты трафиком для нарушения или прекращения их обслуживания, чтобы сделать его недоступным для посетителей. В результате перегруженного сервера система не может отвечать на запросы пользователей и прекращает свою работу.

Распределенная атака «отказ в обслуживании» DDoS – тот же метод, что и традиционный DoS, за исключением того, что хакер использует несколько взломанных устройств для запуска атаки трафиком на целевой веб-ресурс в более крупном масштабе. Эта атака сложнее и опаснее, так как выполняется одновременно с разных IP-адресов по всему миру, что затрудняет определение её источника для сетевых администраторов.

Существуют разные типы DoS- и DDoS-атак, наиболее распространенные из них – TCP SYN flood, smurf, ping-of-death, ботнеты. Злоумышленники часто используют Dos-, DDos-атаки вместе с другими атаками, чтобы отвлечь автоматизированные системы защиты от реагирования на проблему.

Межсайтовый скриптинг (XSS), атака с использованием межсайтовых сценариев – атака, при которой злоумышленник загружает в базу данных веб-сайта вредоносный клиентский код (скрипт JavaScripts, реже HTML, VBScript, ActiveX, Flash). Атаки XSS разделяют на три категории: сохраненные (постоянные), отраженные (непостоянные), основанные на DOM.

Атака с использованием SQL-инъекции нацелена на получение доступа к базам данных, хранящихся на сервере. Хакеры загружают скрытые запросы, содержащие вредоносный код, в веб-формах (например, в форме данных пользователя при входе на сайт – логин, пароль). Когда пользователь вводит свои данные в веб-форму и нажимает кнопку «Войти», он отправляет запрос SQL в базу данных на сервере для извлечения его данных и отражения их в пользовательском интерфейсе.

**4.3** **Меры безопасности проекта**

Для безопасности веб-приложения была проведена проверка вводимых данных со стороны пользователя. Это позволяет предотвратить неправильные или нежеланные вводимые данные, включая те, которые могут быть использованы злоумышленниками для SQL, CSRF, других атак.

Так же было использовано хеширование паролей. Все пароли пользователей будут хешироваться и храниться в зашифрованном виде, который предотвращает кражу данных со стороны злоумышленников.

Так же были использованы подготовленные запросы, которые позволяют предотвратить SQL-инъекции, поскольку запросы выполняются отдельно от данных, вводимых пользователем.

При этом были использованы защищённые протоколы передачи данных, которые защищают пользовательские сессии от несанкционированного доступа к данным.

Вместе с тем производиться регулярное резервное копирование веб-приложения и всех важных данных. И так же проводиться тестирование веб-приложения, которое позволяет убедиться, что не были внедрены вредоносные ссылки.

**5 УПРАВЛЕНИЕ ПРОЕКТОМ**

**5.1 Понятие проекта. Необходимость управления проектом**

Проект — это временное усилие, предпринимаемое для достижения конкретной цели или создания уникального продукта, услуги или результата. Он отличается от рутинных операций, так как имеет четкую начальную и конечную точки, ограниченные ресурсы и специфические задачи.

Проект как система деятельности существует ровно столько времени, сколько требуется для получения конечного результата. Однако данное обстоятельство не противоречит концепции фирмы: проект все чаще становится основной формой деятельности фирмы.

Любой проект включает в себя замысел, который представлен проблемой или задачей, требующей решения. Следующей составной частью проекта являются средства его реализации, которые представляют собой способы решения имеющихся проблем и задач. В процессе реализации проекта получают результаты в виде решения проблемы и задачи.

В самом общем виде проектом называют то, что задумывается или планируется. Под проектом еще понимают некоторую задачу, которая обладает определенными исходными данными и требуемыми результатами (целями). Многие проекты по своему характеру являются инвестиционными, что означает вложение определенного количества ресурсов для получения через отведенное необходимого (как правило, финансового) результата. В то же время проект может быть направлен на создание или модернизация физических объектов, технологических процессов, технической и организационной документации.

Таким образом, проекты в современном понимании представляют собой то, что изменяет наш мир. В этом случае примерами проектов будут строительство жилого или промышленного объекта, создание нового предприятия или реконструкция уже существующего, разработка новой технологии, создание кинофильма и так далее.

Управление проектом — это процесс планирования, организации, контроля и координации ресурсов и задач, связанных с проектом, с целью достижения его целей и успешной реализации. Оно включает в себя принятие решений, определение приоритетов, распределение ресурсов, управление временем, стоимостью, качеством и рисками проекта.

Необходимость управления проектом обусловлена рядом факторов:

* Комплексность проектов: проекты могут быть сложными и включать множество задач, ресурсов, заинтересованных сторон и переменных. Управление проектом помогает структурировать и организовать эти элементы, чтобы достичь успешных результатов.
* Ограничения ресурсов: проекты обычно имеют ограниченные ресурсы, такие как бюджет, время, персонал и материалы. Управление проектом позволяет эффективно распределить и использовать эти ресурсы, чтобы максимизировать результаты проекта.
* Риски и неопределенность: проекты часто связаны с рисками и неопределенностью. Управление проектом включает в себя идентификацию, анализ и управление рисками, а также планирование и адаптацию к изменяющимся условиям проекта.
* Целевая ориентация: проекты имеют конкретные цели и результаты, которые нужно достичь. Управление проектом помогает определить эти цели, разработать планы действий и следить за их достижением.
* Командная работа: проекты обычно вовлекают команду людей с разными ролями и задачами. Управление проектом обеспечивает координацию и коммуникацию между участниками проекта, что способствует синхронизации и эффективности работы.

Управление проектом позволяет улучшить планирование, контроль и исполнение проекта, минимизировать риски и неопределенность, обеспечивать достижение поставленных целей и результатов проекта. Оно играет важную роль в обеспечении успеха проекта и оптимального использования ресурсов.

## **5.2 Постановка проекта по SMART**

Технология SMART — современный подход к постановке работающих целей. Система постановки smart — целей позволяет на этапе целеполагания обобщить всю имеющуюся информацию, установить приемлемые сроки работы, определить достаточность ресурсов, предоставить всем участникам процесса ясные, точные, конкретные задачи.

Расшифровка каждого компонента SMART:

* Specific (Специфичные): цель должна быть конкретно сформулирована и ясно определена. Должно отвечать на вопросы «что», «кому», «когда» и «где».
* Measurable (Измеримые): цель должна быть измеримой, чтобы можно было оценить ее достижение. Это означает определение конкретных показателей или метрик, которые позволят оценить прогресс.
* Achievable (Достижимые): цель должна быть достижимой и реалистичной. Должно основываться на доступных ресурсах, знаниях и возможностях.
* Relevant (Релевантные): цель должна быть связана с общими стратегическими целями и приоритетами. Должно иметь значение и быть согласованной с другими целями и направлениями компании или проекта.
* Time-bound (Ограниченные по времени): цель должна быть ограничена по времени, иметь четкий срок выполнения или дедлайн. Это помогает устанавливать ясные временные рамки и создавать чувство срочности.

Технология SMART помогает:

* сфокусироваться на конкретном результате и не распыляться по сторонам;
* отбросить на старте нерелевантные цели, сэкономить таким образом время и ресурсы;
* не откладывать реализацию задуманного на потом, придерживаться заранее установленных временных рамок;
* увидеть путь достижения, наметить шаги и составить список задач;
* отказаться от нереалистичных, завышенных требований;
* отслеживать прогресс по цели в понятных и легко измеримых этапах.

Цель любого проекта — создание уникального продукта, не имеющего аналогов. Так что можно смело сказать, что проектный подход — это основа внедрения инноваций. С помощью грамотного управления проектами компания может повысить свою эффективность.

Таблица 4 — Постановка цели проекта по SMART

|  |  |  |
| --- | --- | --- |
| Обозначение | Название | Значение |
| S | Specific (Конкретный) | Создание веб–сайт по управлению “Умным домом” |
| M | Measurable (Измеримый) | Обеспечить пригодность к эксплуатации в указанные сроки |
| A | Attainable (Достижимый) | Исходя из общедоступных ресурсов и сроков реализации проекта создать готовое веб–сайт по управлению “Умным домом” |
| R | Relevant (Актуальный) | Разработать веб–сайт основываясь на современных и популярных источниках. |
| T | Time–bound (Ограниченный по времени) | Разработать готовое веб–сайт в указанные сроки |

Таким образом постановка SMART для веб–приложения показывает ясность цели, определит её актуальность и достижимость, и организацию сроков для создания качественного веб–приложения.

**5.3** **Анализ заинтересованных сторон проекта**

Анализ заинтересованных сторон проекта позволяет охарактеризовать заинтересованные стороны, а также оценить их интерес к проекту. Это является действенным инструментом управления, так как он дает возможность задавать стратегические вопросы, помогает понять сложность рассматриваемого вопроса и представит в результате перечини основных характеристик каждой из заинтересованных сторон.

Заинтересованные стороны:

* Пользователи (владельцы умного дома): возможность дистанционного управления устройствами, получение информации о работе систем, интеграция с другими устройствами.

–Разработчики устройств умного дома: возможность интеграции своих устройств с веб-сайтом, получение обратной связи от пользователей, улучшение своих продуктов.

–Производители программного обеспечения: интеграция со своими системами, расширение клиентской базы, получение отзывов о работе своих продуктов.

Потребности заинтересованных сторон:

* Пользователи: удобный и функциональный веб-сайт для управления умным домом, возможность расширения функционала за счет интеграции с другими устройствами и программным обеспечением.
* Разработчики устройств: возможность быстрой и простой интеграции своих устройств с веб-сайтом, доступ к обратной связи от пользователей и возможность улучшения своих продуктов.
* Производители программного обеспечения: доступ к данным пользователей для улучшения своих систем, возможность интеграции с другими системами и устройствами, расширение клиентской базы.

## **5.4** **Устав проекта**

Устав проекта - это документ, который определяет основные правила и положения проекта. Он включает в себя информацию о целях и задачах проекта, этапах его реализации, руководстве проектом, финансировании и других важных аспектах. Устав проекта помогает обеспечить эффективное управление проектом и координацию работы всех участников.

В уставе проекта обычно приводятся следующие элементы:

* Название проекта и его краткое описание.
* Цель и задачи проекта.
* Участники проекта и их роли.
* Этапы реализации проекта.
* Требования к результатам проекта.
* Критерии успешности проекта.
* Ресурсы, необходимые для реализации проекта.
* Процедуры контроля и мониторинга проекта.
* Порядок изменения устава проекта.

Таблица 5 — Устав проекта

|  |  |
| --- | --- |
| Назначение проекта | Разработка веб–сайта “Умный дом” |
| Цели проекта | Создать веб–сайт по управлению “Умным домом” простым, доступным, многогранным и объёмным в своих возможностях |
| Описание проекта | Веб–сайт, которое предоставляет возможность управлять своим “умным домом” |
| Задачи проекта | 1. Провести анализ предметной области;  2. Определить проблемы схожих сайтов и найти варианты решении этих проблем  3. Разработать базу данных для хранения информации  4. Разработка интерфейса веб–приложения |
| Результаты проекта | Создание работоспособного веб–сайта |
| Критерии успешности выполнения проекта | Добавление своих устройств на сайте и управление ими посредством сайта |

## **5.5** **Планирование проекта**

Планирование проекта - это важный этап в процессе его реализации. Оно включает в себя определение целей и задач проекта, выбор ресурсов и команды, а также разработку плана действий. Планирование помогает организовать работу и контролировать процесс выполнения проекта. В тексте на тему “Планирование проекта” можно рассмотреть следующие аспекты:

* Цели и задачи проекта: определение основных целей и задач, которые должны быть достигнуты в ходе реализации проекта.
* Ресурсы: определение необходимых ресурсов для выполнения проекта, таких как люди, оборудование, материалы и т.д.
* Команда проекта: подбор команды, которая будет реализовывать проект, распределение ролей и ответственности между участниками.
* План действий: разработка плана реализации проекта, определение этапов и сроков выполнения задач, установление контрольных точек и показателей эффективности.
* Риски и проблемы: выявление возможных рисков и проблем, которые могут возникнуть в процессе реализации проекта, и разработка мер по их минимизации.
* Мониторинг и контроль: определение методов мониторинга и контроля выполнения проекта, а также системы отчетности и анализа результатов.
* Корректировка плана: возможность внесения изменений и корректировок в план действий в зависимости от возникающих обстоятельств и условий.  
  В целом, планирование проекта помогает обеспечить его успешную реализацию и достижение поставленных целей

Планирование представляет собой совокупность связанных между собой взаимными отношениями процедур. Первым этапом планирования проекта является разработка первоначальных планов, являющихся основой для разработки бюджета проекта, определения потребностей в ресурсах, организации обеспечения проекта, заключения контрактов и прочее. Планирование проекта предшествует контролю по проекту и является основой для его применения, так как проводится сравнение между плановыми и фактическими показателями.

Процессы планирования могут повторяться и входить в состав итерационной процедуры, выполняемой до достижения определенного результата. Например, если первоначальная дата завершения проекта неприемлема, то требуемые ресурсы, стоимость, а иногда и содержание проекта должны быть изменены. Результатом в этом случае будут согласованные сроки, объемы, номенклатура ресурсов, бюджет и содержание проекта, соответствующие целям. Сам процесс планирования не может быть полностью алгоритмизирован и автоматизирован, так как содержит много неопределенных параметров и часто зависит от случайных факторов. Поэтому предлагаемые в результате планирования варианты плана могут отличаться, если они разрабатываются разными командами, специалисты в которых по-разному оценивают влияние на проект внешних факторов.

Основные процессы планирования могут повторяться несколько раз, как в течение всего проекта, так и его отдельных фаз.

В процессе осуществления проекта могут происходить изменения как внутри проекта, так и вне него. Поэтому основное назначение планирования заключается в непрерывном поддержании курса осуществления проекта на пути к его успешному завершению.

Объектами планирования в проекте являются:

* предметная область;
* время;
* стоимость;
* качество;
* организация;
* коммуникации;
* риски;
* изменения;
* прочие компоненты проекта;
* интеграционный план.

Предметная область: Веб-разработка, технологии умного дома.

Время: 1 месяца.

Стоимость: Зачет.

Качество: Обеспечение функциональности и надежности веб-сайта.

Организация: Команда из 3 специалистов: веб-разработчика, дизайнера и тестировщика.

Коммуникации: Регулярные встречи и обсуждения с командой, а также общение с пользователями.

Риски: Технические проблемы, изменение требований пользователей.  
Изменения: Возможность внесения изменений в процессе разработки.  
Прочие компоненты проекта: Интеграция с устройствами умного дома, безопасность данных пользователей.

Интеграционный план: После разработки веб-сайта будет проведена его интеграция с устройствами умного дома для обеспечения функциональности.

**5.6** **Управление стоимостью**

**Управление стоимостью проекта объединяет процессы, выполняемые в ходе планирования, разработки бюджета и контролирования затрат, обеспечивающие завершение проекта в рамках утвержденного бюджета.**

Важной особенностью процессов управления стоимостью проекта является их очень тесная связь с иными процессами планирования. Информация о рисках проекта также может значительно повлиять как на величину, так и на структуру проекта.

Одним из первых шагов, выполняемых менеджером проекта в ходе управления стоимостью проекта, является разработка концепции управления стоимостью. Эта концепция должна содержать общие правила организации управления затратами проекта, принципы учета и документирования, рекомендуемые методики и технологии.

Разработка концепции управления стоимостью и финансированием проекта:

* выработка стратегии управления стоимостью и финансами проекта (определение целей и задач, критериев успеха и неудач, ограничений и допущений);
* проведение экономического анализа и обоснования проекта (маркетинг, оценка стоимости и источников финансирования, прогноз выполнения);
* общая экономическая оценка проекта;
* разработка укрупненного графика финансирования;
* определение требований к системе управления стоимостью и финансированием в проекте;
* утверждение концепции.

Согласно статье 129 Трудового Кодекса России заработная плата (оплата труда работника) — это вознаграждение за труд в зависимости от квалификации работника, сложности, количества, качества и условий выполняемой работы, а также компенсационные выплаты (доплаты и надбавки компенсационного характера, в том числе за работу в условиях, отклоняющихся от нормальных, работу в особых климатических условиях и на территориях, подвергшихся радиоактивному загрязнению, и иные выплаты компенсационного характера) и стимулирующие выплаты (доплаты и надбавки стимулирующего характера, премии и иные поощрительные выплаты).

Тарифная ставка – фиксированный размер оплаты труда работника за выполнение нормы труда определенной сложности (квалификации) за единицу времени без учета компенсационных, стимулирующих и социальных выплат определяется по формуле (11):

|  |  |
| --- | --- |
| *Тст = МРОТ/Вф,* | (11) |

где *МРОТ* – минимальный размер оплаты труда с 30 ноября 2023 г. (оклад сотрудника). В Челябинской области с 30 ноября 2023 года при расчете выплат работникам применяют новый российский минимальный размер оплаты труда в размере 16242 рублей;

*Вф* – фактически отработанное время (168 час. в месяц т.е. 21 смена по 8 часов).

*Тст* = 16242 / 168 = 96,68 руб.

Время работы персонального компьютера при создание программного продукта определяется по формуле Время работы персонального компьютера при создание программного продукта определяется по формуле (12)

|  |  |
| --- | --- |
| *Фв = 1,15  (Тп + Тд + Тотл)  kкор* | (12) |

Фв = 1,15 ** 133,87 + 34,46 + 33,47) ** 0,9 = 208,86 час

Себестоимость продукции (услуги) — это сумма затрат на производство и реализацию (сбыт) этой продукции (услуг).

Затраты на производство продукции — это расходы по обычным видам деятельности, которые связаны с производством продукции, выполнением работ, оказанием услуг. В совокупности с расходами на продажу они формируют затраты на производство и реализацию продукцию.

**5.7 Управление рисками проекта**

Управление рисками проекта — это процесс идентификации, анализа, оценки, реагирования и мониторинга рисков, которые могут повлиять на достижение целей проекта.

Цель управления рисками проекта — повышение вероятности возникновения и воздействия благоприятных событий и снижение вероятности возникновения и воздействия неблагоприятных для проекта событий.

Цель управления рисками проекта заключается в идентификации, анализе, оценке и управлении рисками, которые могут повлиять на достижение целей проекта. Главная цель состоит в минимизации негативных воздействий рисков на проект и максимизации вероятности успешного завершения проекта.

Весь процесс управления рисками проекта разделяется на отдельные подпроцессы, которые возникают в каждом проекте, а некоторые из них реализуются на нескольких стадиях проекта.

Так, на стадии планирования проекта происходит интегрирующий процесс выбора подхода, планирования и выполнения операций по управлению рисками проекта. Также на стадии планирования проекта происходит процесс идентификации рисков, т.е. определение того, какие риски могут повлиять на проект, и документальное оформление их характеристик.

Необходимым условием управления рисками проекта является планирование реагирования на риски, т.е. разработка возможных вариантов и действий, способствующих повышению благоприятных возможностей и снижению угроз для достижения целей проекта.

Процесс управления рисками проекта включает следующие шаги:

* Идентификация рисков: на этом шаге производится определение потенциальных рисков, которые могут повлиять на проект. Риски могут быть связаны с различными аспектами, включая технические, временные, финансовые, организационные и другие.
* Анализ рисков: для каждого идентифицированного риска проводится анализ его вероятности возникновения и воздействия на проект. Это позволяет определить наиболее значимые риски, требующие особого внимания.
* Оценка рисков: в этом шаге риски оцениваются с точки зрения их влияния на проект и возможности реализации мер по управлению ими. Оценка рисков помогает определить приоритеты и разработать стратегии по их управлению.
* Планирование реагирования на риски: на основе анализа и оценки рисков разрабатывается план действий по управлению рисками. В этом плане определяются меры по предотвращению рисков, смягчению их последствий, а также планы контроля и реагирования в случае возникновения рисков.
* Реализация плана управления рисками: план управления рисками становится основой для принятия решений и осуществления действий по управлению рисками во время выполнения проекта. Ресурсы и ответственность распределяются с учетом плана.
* Мониторинг и контроль рисков: Весь процесс управления рисками требует постоянного мониторинга и контроля. Риски могут изменяться, новые риски могут возникать, поэтому важно следить за состоянием рисковой ситуации и принимать соответствующие меры по обновлению и корректировке плана управления рисками.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данного курсового проекта был разработан веб-сайт, который отвечает современным требованиям и стандартам веб-разработки. Проект был основан на анализе потребностей и целей заказчика, а также учете требований пользователей.

В процессе разработки были использованы современные технологии и инструменты, такие как HTML, CSS, JavaScript, а также фреймворки и библиотеки, позволяющие создать удобный и функциональный интерфейс. Веб-сайт был разработан с учетом адаптивного дизайна, что позволяет его корректное отображение на различных устройствах, включая планшеты.

Также в процессе разработки был проведен тестирование функциональности веб-сайта. Были выявлены и исправлены возможные ошибки и недочеты, что позволило достичь высокой стабильности сайта.

В результате выполнения данного курсового проекта был создан веб-сайт. Он предоставляет удобный и интуитивно понятный интерфейс, а также функциональность, необходимую для достижения поставленных целей.

Разработка данного веб-сайта позволила применить и углубить знания в области веб-разработки, а также приобрести опыт работы и управлении проектом.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧИКОВ

1. **Агальцов, В.П.** Базы данных. В 2-х кн. Книга 2. Распределенные и удаленные базы данных [Электронный ресурс]: учебник / В.П. Агальцов. – Москва: ИД «ФОРУМ»: ИНФРА-М, 2017. – 271 с. – Режим доступа: <https://new.znanium.com/catalog/document?id=155080> (дата обращения: 04.02.2023).
2. **Ананьева, Т.Н.** Стандартизация, сертификация и управление качеством программного обеспечения [электронное пособие]: учеб. пособие / Т.Н. Ананьева, Н.Г. Новикова, Г.Н. Исаев. – Москва: ИНФРА-М, 2017. – 232 с. – (Высшее образование: Бакалавриат). —Режим доступа: <https://new.znanium.com/catalog/product/792682>  
   (дата обращения: 03.04.2023).
3. **Коваленко, В. В.** Проектирование информационных систем [Электронный ресурс] : учебное пособие / В. В. Коваленко. — Москва : ФОРУМ : ИНФРА-М, 2018. — 320 с. — Режим доступа: <https://znanium.com/read?id=329691> (дата обращения: 10.03.2023).
4. **Кроксен-Джон Д.** Оптимизация интернет-магазина: Почему 95% посетителей вашего сайта ничего не покупают и как это исправить [Электронный ресурс] : Справочное пособие / Кроксен-Джон Д., Ван Тондер Й. - Москва :Альпина Паблишер, 2018. - 318 с.: ISBN 978-5-9614-7131-1 - Режим доступа: <https://znanium.com/read?id=333470>  
   (дата обращения: 20.01.2023).
5. Курсы по программированию: [Электронный ресурс] // URL: <https://htmlacademy.ru/> (дата обращения: 28.04.2023).
6. **Лаврентьев, А. Н.** Цифровые технологии в дизайне. История, теория, практика [Электронный ресурс] : учебник и практикум для вузов / А. Н. Лаврентьев [и др.] ; под редакцией А. Н. Лаврентьева. - 2-е изд., испр. и доп. - Москва : Издательство Юрайт, 2019. — 208 с. — (Авторский учебник). — ISBN 978-5-534-07962-3. — Режим доступа: <https://www.urait.ru/bcode/424029> (дата обращения: 06.02.2023).
7. **Лисьев, А.В.** Программное обеспечение компьютерных сетей и web-серверов: учеб. пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. – Москва: ИНФРА-М, 2018. – 145 с. – (Высшее образование: Бакалавриат) Режим доступа: <https://new.znanium.com/catalog/document?id=328870> (дата обращения: 13.04.2023).
8. **Назаров, С.В.** Архитектура и проектирование программных систем [электронное пособие]: монография / С.В. Назаров. – 2-е изд., перераб. и доп. – Москва: ИНФРА-М, 2017. – 374 с. – (Научная мысль). Режим доступа: <https://new.znanium.com/catalog/product/907016> (дата обращения: 05.03.2023).
9. **Немцова, Т. И.** Компьютерная графика и web-дизайн [Электронный ресурс] : учебное пособие / Т. И. Немцова, Т. В. Казанкова, А. В. Шнякин ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. + Доп. материалы. - Режим доступа: <https://znanium.com/read?id=329728> (дата обращения: 10.02.2023).
10. Основы языка гипертекстовой разметки HTML и CSS [электронный ресурс]: учебное пособие /Сост. С. М. Наместников. – Ульяновск: УлГТУ, 2014. // Режим доступа: <http://sernam.ru/book_html.php> (дата обращения: 10.03.2023).
11. Портал по php, MySQL и другим веб-технологиям [Электронный ресурс] ‑ Режим доступа : <http://www.php.su> (дата обращения: 15.04.2023).
12. Ресурсы по программированию на php: [Электронный ресурс] // URL: <http://www.php.net> (дата обращения: 29.04.2023).
13. Ресурсы по программированию: [Электронный ресурс] // URL: <http://www.realcoding.net/> (дата обращения: 01.05.2023).
14. **Шаньгин, В. Ф.** Информационная безопасность компьютерных систем и сетей [Электронный ресурс] : учебное пособие / В. Ф. Шаньгин. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 416 с. — (Среднее профессиональное образование). - Режим доступа: <https://znanium.com/read?id=336332> (дата обращения: 10.04.2023).
15. **Шаньгин, В. Ф.** Комплексная защита информации в корпоративных системах [Электронный ресурс] : учебное пособие / В.Ф. Шаньгин. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 592 с. — (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/read?id=330966> (дата обращения: 11.03.2023).

# ПРИЛОЖЕНИЕ А

Листинг 1 – Регистрация пользователя в базе данных.

public function create(Request $request){

        $this->authUser();

        $data=(object)[

            'id'=>$this->user\_id,

            'name'=>$this->user\_name,

            'image'=>$this->user\_image,

            'email'=>$this->user\_email,

            'role'=>$this->user\_role,

        ];

        return view('auth.reg')->with(['data'=>$data]);

    }

public function store(Request $request){

        $validator=Validator::make($request->all(),[

            'firstname'=>['required','alpha'],

            'lastname'=>['required','alpha'],

            'patronymic'=>['nullable'],

            'email'=>['email','unique:users'],

            'password'=>['required','min:6','confirmed']

        ]);

        if($validator->fails()){

            return redirect()->route('create')->withErrors($validator)->with('success','Ошибка регистрации');

        }

        else{

            User::create($validator->validated());

            return redirect()->route('login')->with('success','Регистрации прошла успешно');

        }

    }

Листинг 2 – Авторизация пользователя в системе.

public function login(Request $request){

        $this->authUser();

        $data=(object)[

            'id'=>$this->user\_id,

            'name'=>$this->user\_name,

            'image'=>$this->user\_image,

            'email'=>$this->user\_email,

            'role'=>$this->user\_role,

        ];

        return view('auth.auth')->with(['data'=>$data]);

    }

    public function signup(Request $request){

        if(Auth::attempt($request->only(['email','password']))){

            return redirect()->route('home')->with('success','Вы авторизованы');

        }

        else{

            return redirect()->route('login')->with('success','Ошибка авторизации');

        }

    }

Листинг 3 – Выход пользователя из системы.

public function logout(Request $request){

        if(Auth::check()){

            Auth::logout();

        }

        return redirect()->route('home')->with('success','Вы вышли');

    }