

ВВЕДЕНИЕ

Информационные технологии – это область, которая занимается использованием компьютерных систем для сбора, хранения, обработки и передачи данных. Благодаря информационным технологиям создается программное обеспечение, которое автоматизирует процессы в различных сферах жизни, улучшая их эффективность и удобство. В области информационных технологий существует множество направлений: веб-разработка, мобильные технологии, базы данных, искусственный интеллект, кибербезопасность и многое другое.

Веб-приложения играют важную роль в обучении детей, предоставляя им доступ к образовательным ресурсам в удобном и интерактивном формате, обогащая их знания и навыки с помощью современных технологий. Они помогают детям учиться в своём собственном темпе и в соответствии с их индивидуальными потребностями. Благодаря визуальным и аудиоэлементам, веб-приложения делают учебный процесс более привлекательным и запоминающимся для детей. Веб-приложения обеспечивают доступ к обучающим материалам в любое время и из любого места, что делает их удобным инструментом для образования как в школе, так и дома.

Целью дипломного проекта является разработка веб-приложения «Учимся финансовой грамотности. Налоги для школьников» с использованием современных технологий разработки, направленного на повышение финансовой грамотности среди школьников, предоставляя интерактивные задания и тесты по основам налоговой системы.

Веб-приложение предлагает различные разделы, которые помогут детям усвоить основные понятия налоговой системы и научиться применять их на практике. Благодаря удобному интерфейсу и понятной структуре, дети могут легко найти интересующий их материал и изучать его в своё удовольствие.

Одним из ключевых компонентов веб-приложения являются интерактивные задания и видеоматериалы, предоставленные для закрепления теоретической части.

Создание веб-приложения, нацеленного на обучение финансовой грамотности и налоговой системе, имеет важное значение для формирования у детей финансовых навыков с ранних лет и является необходимым элементом образования, который помогает подготовить их к самостоятельной жизни. Приложение предоставляет доступный и интерактивный способ изучения этих сложных концепций, что помогает родителям и учителям обеспечить дополнительное обучение детей в домашних условиях.

Пояснительная записка дипломного проекта включает в себя пять глав: аналитическая часть, проектирование программного средства, разработка программного

обеспечения, охрана труда, техника безопасности, противопожарные мероприятия и охрана окружающей среды, экономическая часть.

Первая глава содержит в себе следующие пункты: описание предметной области, анализ существующих разработок и обоснование необходимости, постановка задачи, формализация и документирование требований к программному средству.

Во второй главе рассматриваются: разработка модели данных, входная и выходная информация, функциональная модель, структура данных, иерархическая структура и ее характеристика, проектирование пользовательского интерфейса, выбор программных и аппаратных средств.

Третья глава включает в себя: разработка программного средства, описание основных алгоритмов, методов и приемов разработки программных модулей, реализация взаимосвязи компонентов программного средства, защита информации, тестирование программного средства, разработка документации к программному средству.

Четвертая глава содержит следующие пункты: организация рабочего места для работающих на компьютере, производительность труда и рабочее место, зоны досягаемости горизонтальной плоскости, требования безопасности при работе на персональных электронно-вычислительных машинах, гигиенические требования к правильной посадке при работе на компьютере.

В пятой главе представлены: экономическая часть, характеристика проекта, определение трудоемкости создания программного продукта, определение полной себестоимости и отпускной цены программного продукта, определение экономического эффекта от внедрения программного продукта.

Последним пунктом пояснительной записки является заключение, в котором содержатся выводы о разработке веб-приложения и об экономической эффективности его эксплуатации.

					ДП.2-400101. 43817.ПЗ-24	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1 Аналитическая часть

1.1 Описание предметной области

Предметная область – это конкретная область знаний, проблем или задач, которые решает или описывает программа или система. [1]

При проектировании программного обеспечения важно понимать предметную область, чтобы разработать соответствующую архитектуру и функциональность, что позволяет программистам создавать эффективные и удобные для использования приложения, которые решают конкретные задачи или удовлетворяют потребности пользователей в определенной сфере.

Понимание предметной области помогает определить структуру данных, методы и алгоритмы, необходимые для эффективной работы программы. Умение анализировать и моделировать предметные области является важным навыком для разработчиков программного обеспечения.

Предметной областью дипломного проекта является Инспекция Министерства по налогам и сборам Республики Беларусь по Сморгонскому району.

Инспекция занимается учетом причитающихся к уплате и фактически уплаченных сумм налогов и других обязательных платежей, разрабатывает предложения по вопросам регулирования и управления в сфере налогообложения.

Инспекция МНС по Сморгонскому району занимается осуществлением контроля за соблюдением налогового законодательства в пределах своей компетенции, включающей проверку правильности исчисления, своевременности и полноты уплаты налогов, сборов и других обязательных платежей в бюджет, в государственные целевые бюджетные фонды. Она контролирует сферу игорного бизнеса, обеспечивая соблюдение соответствующего законодательства.

В числе основных функций входит осуществление камерального контроля за полнотой и своевременностью уплаты налогов, что включает использование современных автоматизированных информационных систем.

Важной составляющей работы инспекции является обеспечение применения различных информационных систем, таких как автоматизированная информационная система «Расчет налогов» и прием и внесение в системы налоговых деклараций и отчетов, предусмотренных законодательством.

Неотъемлемой частью ее деятельности является взыскание неуплаченных налогов и иных обязательных платежей в бюджет, включая применение мер по обеспечению исполнения налогового обязательства и уплаты пеней.

					ДП.2-400101. 43817.ПЗ-24	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

Инспекция МНС по Сморгонскому району выполняет разнообразные функции в соответствии с ее уставными обязанностями.

Она осуществляет взыскание денежных средств, полученных от реализации имущества, обращенного в доход государства или взыскание на неуплаченные налоги и пени и возмещает расходы, связанные с таким имуществом в случае его возврата собственнику. Обеспечивает взыскание расходов по имуществу, которое подверглось административному взысканию, но не было конфисковано или оценено в целях взыскания и принимает решения о признании задолженности безнадежным долгом и ее списании.

Организация осуществляет электронное взаимодействие с плательщиками и иными обязанными лицами, в том числе через системы электронного декларирования, и предоставляет услуги плательщикам, используя электронные очереди. Проведение разъяснительной работы по вопросам применения налогового законодательства дополнительно входит в ее компетенцию.

Инспекция рассматривает обращения по вопросам налогообложения и проводит анализ отчетных данных, чтобы разрабатывать предложения о совершенствовании деятельности налоговой инспекции, выполняет административные процедуры, занимается подбором и обучением кадров, обеспечивает защиту сотрудников во время исполнения ими служебных обязанностей.

1.2 Анализ существующих разработок и обоснование необходимости

Для разработки веб-приложения, необходимо изучить существующие аналоги, их интерфейс и функциональные особенности.

Для анализа выбраны следующие веб-приложения: «Финансовые грабли», «Вклад в будущее» раздел «Игры и игровые занятия по финансовой грамотности» и интеллектуальная онлайн-игра «Первые шаги в мире финансов».

Веб-приложение «Финансовые грабли» предлагает игры, которые объединены в три тематические модуля. В каждом модуле три игры, справочный материал и проверочное задание. Игры представляют собой викторины с выбором ответа и накоплением баллов. Предоставлено неограниченное число попыток прохождения игр для тренировки перед прохождением тестирования.

Несмотря на свои преимущества, веб-приложение «Финансовые грабли» имеет недостатки. Ограниченная глубина материала и охват тем могут не обеспечивать полноценного понимания финансовых концепций. Недостаточное разнообразие заданий может быстро утратить интерес пользователей, а ограниченное число попыток проверки знаний не учитывает индивидуальные потребности, требующие дополнительного времени и повторений для усвоения материала.

					ДП.2-400101. 43817.ПЗ-24	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

Интеллектуальная онлайн-игра «Первые шаги в мире финансов» – это браузерная игра, которая рассказывает о минимальном наборе знаний для создания собственного бизнеса. Игра построена по принципу викторины, где сначала дается теоретическая информация, а затем она проверяется с помощью вопросов. Если ответ неверный, участник возвращается к изучению теории. Игра состоит из 5 блоков: получение статуса индивидуального предприятия, страхование, инвестиционные операции, защита от мошенничества, пенсионное обеспечение.

Можно выделить следующие недостатки онлайн-игры «Первые шаги в мире финансов».

Односторонность подхода – игра сконцентрирована исключительно на процессе открытия индивидуального предпринимателя, оставляя за пределами внимания другие формы бизнеса, что приводит к узкому и поверхностному пониманию финансовых вопросов.

Отсутствие интерактивности – хотя игра предлагает вопросы для проверки знаний, но в ней отсутствует дополнительная интерактивность, которая могла бы увлечь игроков и углубить их понимание. Ограниченность в способах взаимодействия сделать процесс обучения менее эффективным.

Раздел «Игры и игровые занятия по финансовой грамотности» веб-приложения «Вклад в будущее» – это инструмент обучения, предлагающий восемь захватывающих игр для детей в возрасте от 5 до 13 лет и четыре игры для подростков в возрасте от 13 до 18 лет. Приложение разработано специально для использования в учебных помещениях, оборудованных проектором и экраном, и предлагает захватывающий способ ознакомления с финансовой грамотностью через интерактивные викторины.

Недостатками раздела «Игры и игровые занятия по финансовой грамотности» веб-приложения «Вклад в будущее» можно выделить: приложение, ориентированное на детей и подростков, малоэффективно для старших возрастных групп и взрослых, интересующихся финансовой грамотностью через игры. Его полноценное использование требует проектора и экрана, что ограничивает доступность в условиях ограниченных ресурсов учебных заведений. Интерактивные викторины, хотя и увлекательные, ограничивают возможности глубокого взаимодействия и обсуждения темы среди участников.

Актуальность веб-приложения «Учимся финансовой грамотности. Налоги для школьников» заключается в том, что оно предоставляет целенаправленный и структурированный подход к обучению школьников основам финансовой грамотности и налоговой системы. В приложении информация о налогах систематизирована и доступна

для усвоения в удобном формате, что делает процесс обучения более эффективным и увлекательным для учащихся.

Веб-приложение обладает удобным интерфейсом, который позволит пользователям быстро и легко находить необходимую информацию. Возможность быстрого доступа к различным разделам и вкладкам обеспечит эффективное использование приложения как преподавателями, так и учащимися.

Для преподавателей веб-приложение является хорошим инструментом для организации учебного процесса и автоматизации контроля знаний в сфере налогообложения. Оно обеспечивает структурирование учебного материала, предоставление учащимся понятных и наглядных ресурсов и упрощает процесс проверки их знаний.

Для учащихся приложение станет ценным инструментом для самостоятельного изучения учебного материала по финансовой грамотности в сфере налогообложения. Они смогут легко осваивать новые знания и проверять их через встроенные тесты и задания, получать ответы на интересующие вопросы в удобном и доступном формате.

1.3 Постановка задачи

Целью дипломного проекта является разработка веб-приложения «Учимся финансовой грамотности. Налоги для школьников».

Веб-приложение позволит учащемуся:

- пройти регистрацию/авторизацию перед началом работы с веб-приложением;
- изучить теорию с помощью информации по темам и видео;
- закрепить теоретический материал игрой;
- проверить свои знания с помощью теста;
- ознакомиться с историческими датами в сфере налогообложения;
- осуществить поиск термина в налоговом словаре.

1.4 Формализация и документирование требований к программному средству

Требования – это четко определенные и описанные потребности, функции, условия и характеристики, которым должен удовлетворять продукт, чтобы соответствовать ожиданиям пользователей, заказчиков и других заинтересованных сторон. [2]

Функциональные требования для веб-приложения «Учимся финансовой грамотности. Налоги для школьников».

Раздел изучения основ налогообложения: возможность ознакомиться с информационными материалами о различных видах налогов. Предоставление учебных

материалов, интерактивные заданий, которые помогут пользователю закрепить полученные знания. Материалы представлены в понятной и доступной форме.

Раздел тестирования знаний: тесты позволят проверить уровень знаний, а система оценки результатов и обратной связи предоставит пользователю информацию о его успехах.

Раздел интерактивных упражнений: интерактивные элементы: анимации, звуковые эффекты и возможности взаимодействия с системой.

Раздел доступа к дополнительным ресурсам: предоставление ссылок на полезные образовательные материалы: налоговый календарь, налоговый словарь и исторические даты, которые помогут углубить знания по теме финансовой грамотности и налогов.

При проектировании предъявлены следующие требования к производительности:

- система должна поддерживать стабильную работу и обеспечивать отсутствие сбоев при использовании;
- веб-приложение должно обеспечивать быстрый отклик на запросы пользователей, как при загрузке страниц, так и при выполнении игр и тестов, минимизировав время загрузки ресурсов, ускорив выполнение запросов к базе данных и обработку пользовательского ввода.

Требования к интерфейсу веб-приложения:

- пользователь может использовать меню навигации для перехода между различными страницами приложения;
- на страницах приложения представлены ссылки и кнопки, позволяющие пользователю переходить к дополнительной информации, выполнять различные действия: прохождение тестов и заданий, возвращаться на предыдущие страницы;
- в приложении должны быть использованы изображения и графика, иллюстрирующие концепции налогов и финансовой грамотности;
- для облегчения понимания сложных понятий пользователю предоставлены аудио- и видео-материалы, поясняющие основные принципы налогообложения и финансовой грамотности.

Интерфейс веб-приложения обеспечивает удобное и эффективное взаимодействие пользователя с информацией о налогах и финансовой грамотности, делает процесс обучения интересным для школьников.

Техническое задание веб-приложения «Учимся финансовой грамотности. Налоги для школьников» представлено в Приложении А.

2 Проектирование программного средства

2.1 Разработка модели данных

Модель данных – это абстрактное представление о данных и их отношениях в системе или приложении. Она определяет структуру, типы данных, правила и ограничения для хранения, обработки и передачи информации. Модель данных помогает организовать данные так, чтобы они были легко доступны, понятны и могли быть эффективно использованы программами и пользователями. [3]

Существует несколько видов моделей данных: реляционная, иерархическая, сетевая, объектно-ориентированная, документоориентированная.

Реляционная модель данных: организует данные в виде таблиц, где каждая строка представляет собой запись, а каждый столбец – атрибут или поле. Отношения между таблицами определяются ключами.

Иерархическая модель данных: представляет данные в виде древовидной структуры, где каждая запись имеет одного родителя и может иметь несколько детей.

Сетевая модель данных: расширяет иерархическую модель, позволяя записям иметь несколько родителей, что создает более сложные отношения.

Объектно-ориентированная модель данных: представляет данные в виде объектов, которые могут иметь свойства и методы.

Документоориентированная модель данных: хранит данные в виде документов, обычно в формате JSON или XML.

Каждая модель данных имеет свои преимущества и недостатки, и выбор конкретной модели зависит от требований и характеристик конкретного проекта или системы.

При разработке веб-приложения «Учимся финансовой грамотности. Налоги для школьников» использовалась реляционная модель данных.

Реляционная модель данных позволит эффективно структурировать данные и установить связи между ними с помощью ключевых полей. Одной из основных особенностей реляционной модели является использование SQL для работы с данными. Ее преимущества включают простоту использования, гибкость, поддержку целостности данных и возможность расширения.

2.1.1 Входная и выходная информация

Входная информация – это данные, поступающие на обработку и анализ из внешних источников [4]. Она может быть представлена в различных форматах: текст, аудио, видео, изображения или сигналы от датчиков, и служит основой для дальнейших действий или обработки в системе.

					ДП.2-400101. 43817.ПЗ-24	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Входной информацией веб-приложения «Учимся финансовой грамотности. Налоги для школьников» является: личные данные пользователей, обучающие материалы, игры, тесты.

Обучающие материалы: веб-приложение предоставляет пользователям доступ к различным информационным ресурсам, статьям, видеороликам и интерактивным урокам по финансовой грамотности и налогам. В приложении представлена теоретическая информация о основных налогах, налоговых льготах, налоговых вычетах, налоговых обязанностях, налоговом контроле, бюджете и налогах.

Личные данные пользователей: имя, фамилия, отчество, дата рождения, город, название школы, логин, пароль.

Игры: различные игровые механики, спроектированные для эффективного закрепления знаний о налоговой системе.

Тесты: приложение предлагает разнообразные тесты, спроектированные для проверки понимания и усвоения учебного материала о налоговой системе.

Выходная информация – это результат обработки или анализа входных данных, который предоставляется пользователю, используется для принятия решений или дальнейших действий [4].

Выходная информация может представлять собой различные формы информации: текстовые сообщения, графики, таблицы, диаграммы, отчеты и любые другие форматы данных, которые передают пользователю необходимую информацию или выводы.

Выходной информацией в веб-приложении «Учимся финансовой грамотности. Налоги для школьников» является:

- результаты тестов и игр, предоставляемые пользователю для просмотра его уровня усвоения материала;
- сообщения и уведомления, возникающие при неправильном вводе налоговых терминов и ошибках во время регистрации и авторизации.
- отзывы, являющиеся источниками обратной связи.

Взаимодействие входной и выходной информации создаёт цикл непрерывного обучения и совершенствования, обеспечивая пользователям эффективное и удобное освоение финансовой грамотности и налоговых знаний.

2.1.2 Функциональная модель

Функциональное моделирование – это методология, которая используется для анализа и проектирования систем с точки зрения их функциональных аспектов. Этот подход фокусируется на определении функций, которые выполняет система, и их взаимосвязях, а не на конкретных деталях реализации или технических аспектах. [4]

Целью создания функциональной модели процесса является представление основных функций, выполняемых в рамках этого процесса, и их взаимосвязей, что позволяет лучше понять, как процесс работает, определить его ключевые компоненты и оптимизировать его выполнение. Создание функциональной модели процесса способствует более глубокому пониманию процесса и его улучшению для достижения оптимальных результатов.

Примерный перечень требований, которые могут использоваться при построении функциональной модели сети процессов обработки информации, включает идентификацию информационных потоков, определение различных видов информации, которые обрабатываются в сети процессов, и их потоков от источников до конечных получателей. Необходимо установить связи между процессами, определив взаимосвязи и зависимости между различными процессами обработки информации, включая последовательность выполнения и передачу данных между ними.

Для создания функциональной модели использовался инструмент моделирования Draw.io.

Draw.io – это онлайн-инструмент для создания диаграмм, схем, блок-схем, графиков и других типов визуальных представлений. Он предоставляет удобный интерфейс, позволяющий пользователям создавать и редактировать различные типы диаграмм непосредственно в веб-браузере без необходимости установки дополнительного программного обеспечения. [5]

Обладает большим набором инструментов и элементов для создания диаграмм, включая блоки, стрелки, формы, текстовые блоки и другие объекты. Пользователи могут создавать диаграммы различной сложности.

Draw.io предоставляет простой в использовании интерфейс, который делает создание и редактирование диаграмм максимально интуитивным. Он обладает обширной библиотекой предустановленных элементов и шаблонов для разнообразных типов диаграмм. Импорт и экспорт диаграмм поддерживается в различных форматах файлов, включая PNG, JPEG и PDF. Помимо этого, Draw.io интегрируется с популярными облачными хранилищами, такими как Google Drive, OneDrive и Dropbox, что обеспечивает удобство доступа к диаграммам из любой точки, где есть интернет.

При построении функциональной модели сети процессов обработки информации важно постепенно раскрывать структуру и детали системы. Начальным шагом является создание верхнего уровня модели, который предоставляет общее представление о системе или процессе. Верхний уровень служит основой для дальнейшей детализации и помогает понять общую архитектуру и основные функции системы.

Верхний уровень изображен на рисунке 1.

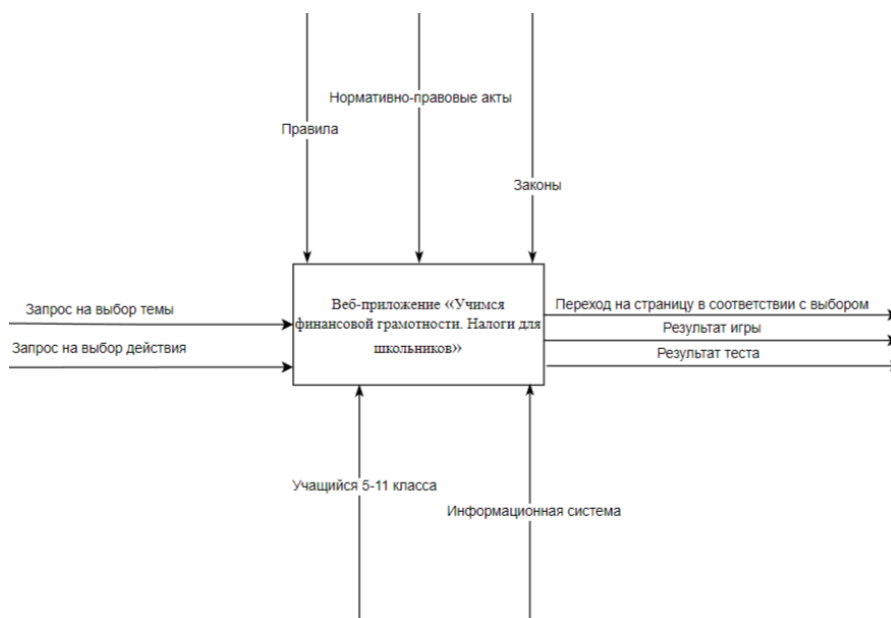


Рисунок 1 – Верхний уровень функциональной модели

На втором уровне функциональной модели, более детально показаны основные типы деятельности программы и их взаимосвязи, что позволяет лучше понять, взаимодействие различных компонентов друг с другом и как информация обрабатывается на разных этапах процесса.

Второй уровень модели представлен на рисунке 2.

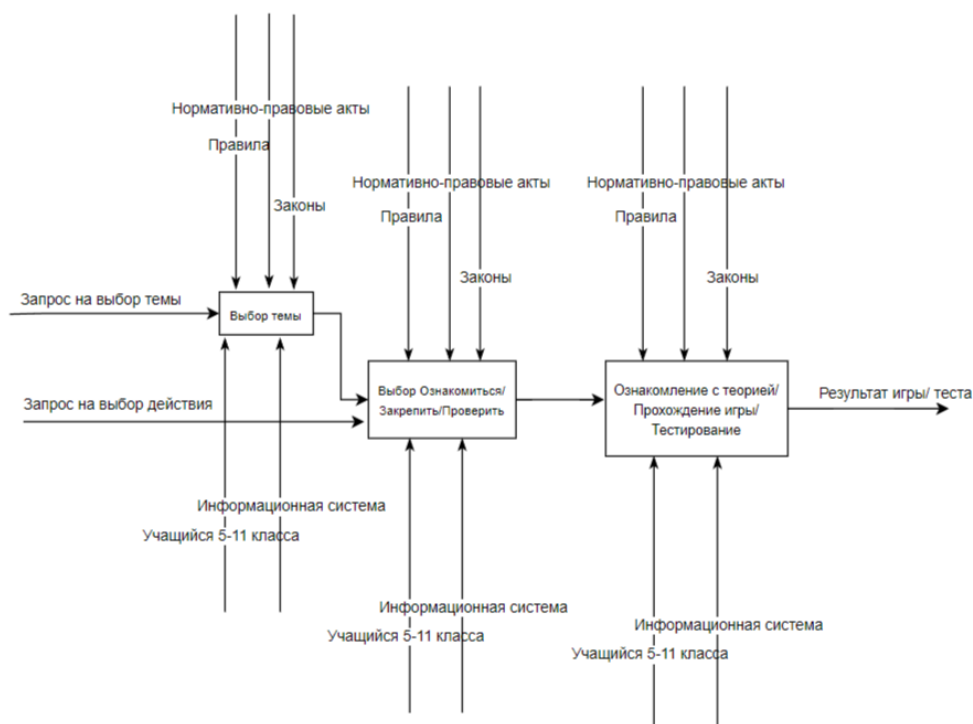


Рисунок 2 – Второй уровень функциональной модели

Построенная диаграмма IDEF0 наглядно показывает основные процессы и их взаимосвязи. Она структурирует приложение, выделяя ключевые модули и их взаимодействие. Показывая потоки данных и управление, она помогает оптимизировать обработку информации и уменьшить ошибки.

2.1.3 Структура данных

Структура данных – это способ организации и хранения информации в компьютере или другой вычислительной системе. Она определяет формат данных, их типы и взаимосвязи между ними, обеспечивая эффективный доступ, обработку и управление информацией. [6]

Для удобного хранения и обработки больших объемов информации используются базы данных.

База данных – это организованная коллекция данных, хранящаяся в компьютерной системе и обеспечивающая эффективный доступ, управление и обновление информации. [6]

При разработке базы данных для веб-приложения «Учимся финансовой грамотности. Налоги для школьников» использовался бесплатный веб-интерфейс, предназначенный для управления базами данных MySQL с помощью веб-браузера – PhpMyAdmin.

База данных представлена в едином файле, что обеспечивает компактность и удобство в управлении данными.

Основным структурным компонентом базы данных является таблица. В таблицах хранятся вводимые данные. Каждая таблица состоит из столбцов, называемых полями, и строк, называемых записями. Каждая запись таблицы содержит всю необходимую информацию об отдельном элементе базы данных.

Для веб-приложения «Учимся финансовой грамотности. Налоги для школьников» создана база данных «registr-bg», которая состоит из следующих таблиц:

Таблица 1 – Структура таблицы «users»

Название поля	Тип данных	Размер поля
Id	int	11
name	varchar	255
fam	varchar	255
otch	varchar	11

Продолжение таблицы 1

Название поля	Тип данных	Размер поля
date	date	–
gorod	varchar	255
school	varchar	255
login	varchar	255
pass	varchar	255

Таблица 2 – Структура таблицы «gorod»

Название поля	Тип данных	Размер поля
Id	int	11
name	varchar	255

Таблица 3 – Структура таблицы «school»

Название поля	Тип данных	Размер поля
Id	int	11
name	varchar	255

База данных «registr-bg» создана с целью хранения и управления информацией, связанной с процессом регистрации пользователей веб-приложения «Учимся финансовой грамотности. Налоги для школьников». Она обеспечивает структурированное хранение данных о пользователях.

2.2 Иерархическая структура и ее характеристика

Иерархическая структура – это тип организации данных или объектов, где каждый элемент имеет один или несколько непосредственных родителей и может иметь одного или нескольких потомков. Элементы располагаются на разных уровнях иерархии, где верхний уровень представляет собой корень или начальный элемент, а нижние уровни – это дочерние элементы, которые могут иметь свои собственные подэлементы. [6]

Иерархическая структура программной системы не описывает процедурные аспекты, такие как последовательность операций, ветвления и повторения. Она скорее представляет собой организацию модулей или компонентов программы в иерархическом порядке, где каждый модуль может иметь один или несколько подмодулей, образуя древовидную структуру.

При анализе иерархической структуры программы первичные характеристики включают количество вершин и количество ребер. Эти параметры помогают понять сложность программы и ее организацию.

Высота иерархии определяется максимальным числом уровней модулей, начиная от корневого модуля и заканчивая листьями. Ширина же характеризует количество подмодулей, напрямую связанных с родительским модулем на каждом уровне иерархии.

Эти характеристики помогают оценить сложность и структуру программной системы, что в свою очередь может помочь в оптимизации процесса разработки и сопровождения программного обеспечения.

2.3 Проектирование пользовательского интерфейса

Пользовательский интерфейс – это способ взаимодействия пользователя с программным обеспечением или устройством, представляющий собой систему элементов управления, графических компонентов и визуальных элементов, которые обеспечивают пользователю доступ к функциональности продукта и позволяют ему взаимодействовать с ним. [7]

Проектирование пользовательского интерфейса важно для создания удобных и эффективных в использовании программ и веб-приложений. Проектирование интерфейса помогает выявить потребности пользователей и адаптировать продукт под их ожидания, что способствует его успешному внедрению на рынок и увеличению его конкурентоспособности.

При проектировании интерфейса были учтены следующие основные принципы: золотое сечение, кошелек Миллера, принцип группировки, бритва Оккама, умственное заимствование.

На главной странице веб-приложения визуальные элементы – кнопки, изображения расположены с учетом пропорций золотого сечения.

Пропорции золотого сечения обеспечивают баланс и гармонию между элементами интерфейса, делая его более приятным для восприятия пользователем. Они способствуют лучшему пониманию и использованию функциональности приложения.

Принцип «Кошелек Миллера» означает учет ограниченной емкости внимания и памяти пользователя. Исходя из этого принципа, важно создавать интерфейс, который минимизирует нагрузку на внимание пользователя и обеспечивает простоту восприятия информации.

В веб-приложении «Учимся финансовой грамотности. Налоги для школьников» главное окно содержит шесть основных функциональных элементов, которые удовлетворяют потребностям пользователей.

Использовано выпадающее меню для скрытия дополнительных функций, что позволяет сделать интерфейс более компактным и удобным для пользователей, не перегружая главное окно лишней информацией (рисунок 3).



Рисунок 3 – страница «Главная»

Принцип группировки при проектировании интерфейса заключается в объединении связанных элементов и функций в логические группы и категории. Принцип помогает создать структурированный и удобный для пользователя интерфейс, упрощает навигацию и повышает понятность информации.

Веб-приложение «Учимся финансовой грамотности. Налоги для школьников» разбито на ясно выделенные блоки элементов.

Каждая из категорий представляет определенную тему, в пределах которой сгруппированы пять игр и один тест, связанные с этой темой.

Для удобства навигации пользователей категории представлены в виде отдельных элементов на главном окне приложения. При выборе категории пользователь может увидеть список доступных игр, относящихся к данной теме.

Применение принципа группировки в веб-приложении «Учимся финансовой грамотности. Налоги для школьников» помогает создать структурированный интерфейс, который упрощает поиск и доступ к необходимым играм и тестам, повышая удобство использования приложения и понимание предлагаемой информации.

Принцип «Бритва Оккама» подразумевает, что интерфейс создается с использованием наименьшего количества элементов, необходимых для выполнения задачи или предоставления функциональности. Принцип основывается на философской концепции, которая призывает к простоте и минимализму.

Элементы интерфейса, предназначенные для обучения через игровой процесс и интерактивные упражнения реализованы с учетом минимализма и простоты, чтобы сосредоточить внимание пользователей на сути задач и обучающих материалах.

По принципу «Умственного заимствования» выбор ярких цветов помогает создать игровую и дружелюбную атмосферу, что в свою очередь делает процесс обучения более увлекательным и эффективным для детей. Использование знакомых детям элементов дизайна: анимация, иконки и символы, которые ассоциируются с играми или развлечениями, упрощают процесс ознакомления с приложением и улучшают его восприятие.

2.4 Выбор программных и аппаратных средств

При разработке веб-приложения «Учимся финансовой грамотности. Налоги для школьников» использовались стандартный язык разметки HTML, динамический язык программирования JavaScript, формальный язык декодирования и описания внешнего вида документа CSS, приложение для создания интерактивных заданий разных уровней сложности LearningApps.org, онлайн-редактор визуального контента Canva, онлайн-редактор Figma. База данных была разработана при помощи веб-приложения с открытым кодом, написанного на языке PHP и представляющего собой веб-интерфейс для администрирования систем управления базами данных MySQL PhpMyAdmin.

HTML применялся для разметки текстов и заголовков, что позволило структурировать информацию о налогах и финансовой грамотности в удобной для восприятия форме. Использование HTML позволило четко организовать контент, используя семантические теги, такие как <h1> для заголовков, <p> для параграфов и для списков, что способствует улучшению читабельности и восприятия материала школьниками.

HTML предоставил возможность внедрять мультимедийные элементы, такие как изображения и видео, с помощью тегов и <video>, что обеспечило интерактивность и наглядность учебного материала. HTML-формы являются основными в реализации функций регистрации и авторизации пользователей. С помощью тегов <form>, <input>, <label> и <button> были созданы удобные и безопасные формы для ввода данных, что позволило пользователям легко создавать учетные записи и входить в систему.

Динамический язык программирования JavaScript применяется для: добавления интерактивности и динамического поведения на страницах, обработка пользовательских действий и обновление контента без перезагрузки страницы, обработки пользовательских вводов, создания анимаций и эффектов, делая интерфейс более привлекательным и удобным для школьников. Дополнительно, JavaScript обеспечивал валидацию данных при

					ДП.2-400101. 43817.ПЗ-24	Лист
						21
Изм.	Лист	№ докум.	Подпись	Дата		

регистрации и авторизации, что улучшает пользовательский опыт и повышает безопасность приложения.

Формальный язык декодирования и описания внешнего вида документа CSS играет ключевую роль в разработке веб-приложения «Учимся финансовой грамотности. Налоги для школьников». CSS отвечает за стиль и внешний вид веб-приложения.

CSS позволил точно позиционировать элементы на странице, используя различные схемы компоновки, такие как Flexbox и Grid, что помогло разработчику организовать контент так, чтобы он был легко читаемым и интуитивно понятным для пользователей.

Приложение для создания интерактивных заданий LearningApps.org использовалось для интеграции различных учебных игр и заданий в веб-приложение «Учимся финансовой грамотности. Налоги для школьников». С помощью LearningApps.org созданы интерактивные упражнения, кроссворды, викторины, которые делают процесс обучения более увлекательным и эффективным.

Для оформления страниц использовались различные конструкторы, в том числе онлайн-инструмент для графического дизайна Canva и онлайн-редактор Figma, в котором удобно проектировать интерфейсы, создавать макеты сайтов, презентации, иллюстрации, логотипы и анимацию.

PhpMyAdmin предоставляет удобный веб-интерфейс для администрирования систем управления базами данных MySQL. Он облегчает создание, изменение и управление базами данных, позволяя разработчикам выполнять такие операции, как создание таблиц, редактирование записей, выполнение SQL-запросов, настройка индексов и управление пользователями и правами доступа.

При разработке веб-приложения «Учимся финансовой грамотности. Налоги для школьников» с помощью PhpMyAdmin создана база данных, обеспечивающая функциональность регистрации и авторизации пользователей. Процесс включает в себя создание таблиц для хранения информации о пользователях, таких как их логины, хэшированные пароли, имена, фамилии, даты рождения, города, названия школ.

Для разработки программного средства использована следующая конфигурация компьютера:

- процессор: IntelCore i3-6006M с тактовой частотой 2.00 ГГц;
- оперативная память: 4 Гб;
- жесткий диск: 1000 Гб;
- видеокарта: Intel HD Graphics 520;
- Windows 10.

3 Разработка программного средства

3.1 Описание основных алгоритмов, методов и приемов разработки программных модулей

Разработка программных модулей – это процесс создания независимых и используемых компонентов программного обеспечения, которые выполняют определенные функции или задачи. Модули разрабатываются для решения конкретных подзадач или функциональных требований программы. [8]

Алгоритм – это последовательность шагов или инструкций, которые описывают процесс выполнения определенной задачи или решения определенной проблемы. Он представляет собой формальное описание порядка действий, необходимых для достижения конечного результата. [8]

Блок-схема – это графическое представление последовательности операций или действий в процессе, алгоритме или системе. [9]

Основные элементы блок-схем – это графические символы и формы, которые представляют различные аспекты процесса или алгоритма.

Прямоугольники – представляют действия или операции в процессе. Например, они могут содержать инструкции, команды или функции, которые должны быть выполнены.

Ромбы – используются для представления ветвлений или принятия решений в процессе. Обычно внутри ромба записывается условие, и в зависимости от его истинности или ложности, происходит переход к определенным действиям.

Параллелограммы – отображают операции ввода или вывода данных. Например, это может быть ввод данных пользователем через клавиатуру или вывод результатов на экран.

Стрелки – используются для указания направления потока выполнения действий в блок-схеме. Они показывают, какие операции должны быть выполнены после других и как происходит переход между различными блоками.

Текстовые аннотации – дополнительные пояснения или комментарии, которые могут быть добавлены к блокам или соединительным линиям для облегчения понимания процесса.

Блок-схема работы веб-приложения «Учимся финансовой грамотности. Налоги для школьников» изображена на рисунке 4.

Благодаря структурированному подходу блок-схемы, стало понятно, какие основные функциональные элементы будут включены в приложение и как они будут взаимодействовать между собой.

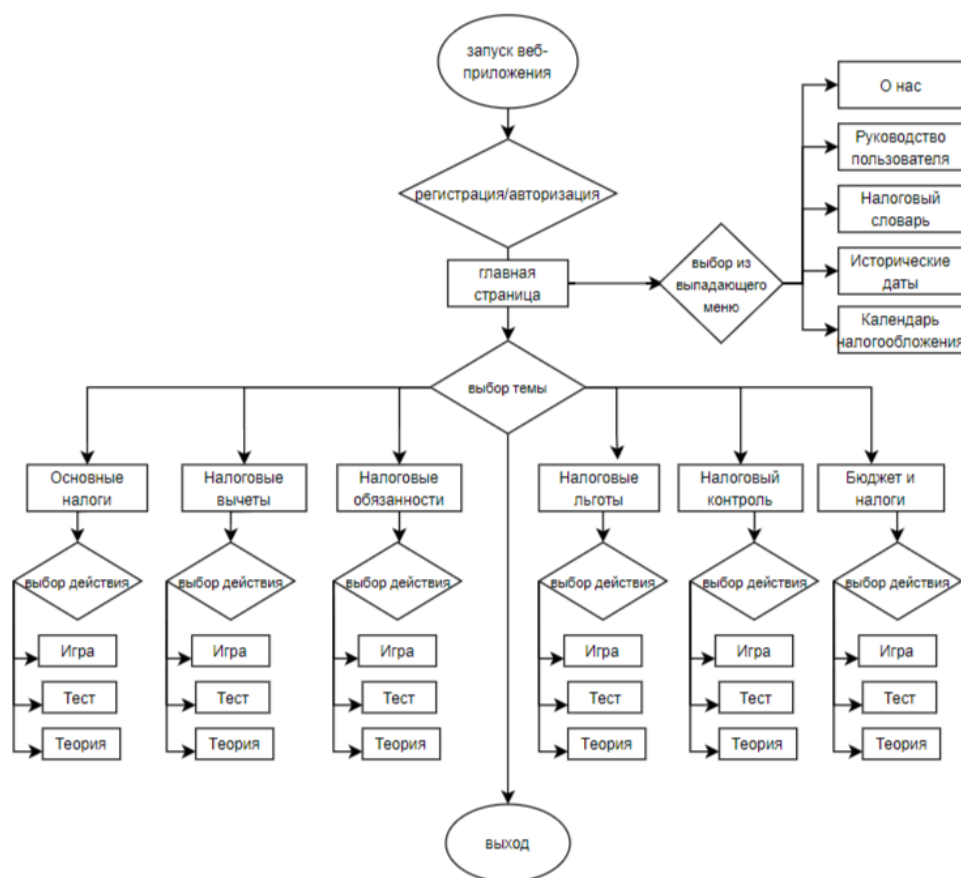


Рисунок 4 – Блок-схема работы веб-приложения «Учимся финансовой грамотности. Налоги для школьников»

Каждый блок, представляющий определенную функциональность, четко определен и сопровождается соответствующими стрелками, указывающими на направление потока данных и управления, что обеспечивает понимание логики работы приложения и последовательность его действий.

3.2 Реализация взаимосвязи компонентов программного средства

Взаимосвязи компонентов программного средства играют важнейшую роль в обеспечении его целостности и эффективности. Взаимосвязи обеспечивают поток данных, согласованность операций и последовательность выполнения задач.

Благодаря этому компоненты могут обмениваться информацией, реагировать на события и совместно реализовывать сложные бизнес-логики. Гармоничная интеграция всех частей программного средства способствует его надежности, масштабируемости и легкости в поддержке, что в конечном итоге улучшает пользовательский опыт и удовлетворяет требования бизнеса.

Веб-приложения состоят из множества компонентов, каждый из которых выполняет специфическую роль и совместно обеспечивает полноценное функционирование системы.

Основными компонентами являются фронтенд, бэкенд, базы данных, серверы, пользовательские интерфейсы и API-интерфейсы.

Фронтенд веб-приложения отвечает за то, как пользователи взаимодействуют с приложением в браузере или на устройстве. Это включает в себя все элементы, которые пользователь видит и с которыми может взаимодействовать, такие как интерфейс, графика, анимация и другие аспекты визуального представления.

Для создания фронтенда веб-приложения «Учимся финансовой грамотности. Налоги для школьников» использовались HTML, CSS, Java Script.

HTML – это стандартизированный язык разметки, используемый для создания веб-страниц. Он состоит из различных элементов, называемых тегами, каждый из которых имеет свою уникальную функцию. [10]

Основные теги HTML, использованные при создании веб-приложения «Учимся финансовой грамотности. Налоги для школьников»:

- <html> определяет начало и конец HTML документа;
- <head> содержит метаинформацию о документе, такую как заголовок, метатеги и ссылки на внешние ресурсы;
- <title> устанавливает заголовок документа, который отображается в заголовке веб-браузера;
- <body> определяет основное содержимое документа, такое как текст, изображения, ссылки и другие элементы;
- <h1>, <h2> определяют заголовки различного уровня;
- <p> определяет абзац текста;
- <a> создает ссылку на другую веб-страницу или ресурс, указывая атрибут href;
- вставляет изображение на веб-страницу, указывая атрибут src;
- , , определяют маркированные и нумерованные списки, а – элементы списка;
- <div>, используются для создания блочных и строчных контейнеров, которые обеспечивают структурирование и стилизацию содержимого страницы;
- <table>, <tr>, <td> определяют таблицу, строки и ячейки таблицы соответственно;
- <button> используется для создания интерактивной кнопки на веб-странице;
- <input> используется для создания интерактивных элементов управления в форме, таких как текстовые поля, кнопки и флажки;
- <form> используется для создания формы для отправки данных;
- <label> используется для связывания текстовой метки с элементом формы;
- <script> используется для включения JavaScript в HTML-документ;

– <div> используется как контейнер для других элементов, JavaScript может изменять содержимое и стили этого контейнера динамически.

Дерево сайта предоставлено в графической части на Листе 2.

3.3 Защита информации

Защита компьютерной информации становится все более важной в условиях растущих угроз со стороны хакеров, вирусов и кибератак. Недостаточная защита может привести к утечке конфиденциальных данных, нарушению целостности информации и прерыванию работы системы, подчеркивая необходимость эффективных мер безопасности.

Защита информации – это комплекс мер и технологий, направленных на обеспечение конфиденциальности, целостности и доступности информации, защиту от несанкционированного доступа, модификации или уничтожения данных. Что является важным аспектом в обеспечении безопасности информационных систем и данных, который включает в себя технические, организационные и правовые меры. [11]

Целью защиты информации является обеспечение конфиденциальности, целостности и доступности данных, а также защита от несанкционированного доступа, модификации и уничтожения информации с использованием соответствующих мероприятий и технологий.

Информационная безопасность должна обеспечивать конфиденциальность, целостность, доступность, аутентификацию, авторизацию, защиту от угроз, мониторинг и аудит.

Конфиденциальность гарантирует доступ к информации только уполномоченным лицам. Целостность обеспечивает защиту информации от несанкционированных изменений и повреждений. Доступность гарантирует своевременный доступ к информации. Аутентификация подтверждает личности пользователей и устройств, авторизация управляет доступом на основе прав пользователей. Защита от угроз предотвращает утечки данных и вредоносные атаки. Мониторинг и аудит отслеживают и анализируют действия пользователей для предотвращения инцидентов безопасности.

При регистрации и входа пользователей в веб-приложение обеспечивается безопасность и защита учетных данных с помощью хеширования паролей и добавления к ним уникальный «солей», чтобы предотвратить атаки по словарю. Передача данных осуществляется через защищенное HTTPS-соединение и ограничиваем количество попыток входа.

Для дополнительной безопасности установлен пароль на базу данных с помощью систем управления базами данных.

					ДП.2-400101. 43817.ПЗ-24	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

3.4 Тестирование программного средства

Тестирование программного средства – это процесс, в ходе которого программное обеспечение проверяется на соответствие требованиям, спецификациям и ожиданиям пользователей. [12]

Виды тестирования включают в себя функциональное тестирование, тестирование производительности, тестирование безопасности и множество других. Каждый вид тестирования направлен на проверку определенных аспектов программного средства.

Тестирование веб-приложений имеет свои особенности, обусловленные их сложностью и многообразием взаимодействий. Тестирование производительности важно для оценки скорости загрузки страниц и стабильности приложения под нагрузкой. Безопасность играет ключевую роль, поэтому необходимо проводить тесты на уязвимости, такие как XSS и SQL-инъекции.

Функциональное тестирование проверяет корректность работы всех функций и пользовательских сценариев, а юзабилити-тестирование помогает оценить удобство интерфейса для конечных пользователей. Тестирование адаптивности гарантирует, что интерфейс правильно отображается и функционирует на устройствах с различными разрешениями экрана. Автоматизация тестирования помогает сократить время и усилия, затрачиваемые на регрессионное тестирование и повторяющиеся задачи.

Тестирование веб-приложения «Учимся финансовой грамотности. Налоги для школьников» проводится с помощью тестового сценария, включающего несколько ключевых этапов.

Сначала выполняется функциональное тестирование для проверки всех функций: регистрация, вход, доступ к обучающим материалам, прохождение игр и тестов. Затем проводится кроссбраузерное тестирование для проверки работы на разных браузерах и устройствах, тестирование производительности для оценки скорости загрузки и стабильности под нагрузкой, тестирование безопасности для выявления уязвимостей, и тестирование юзабилити для проверки удобства интерфейса для школьников. Завершается процесс автоматизацией тестирования.

Тестовый сценарий – это документированная последовательность действий, которые тестирующий должен выполнить для проверки определенного аспекта программного продукта. Он описывает шаги, которые следует выполнить, ожидаемый результат или поведение программы на каждом этапе тестирования.

Веб-приложение «Учимся финансовой грамотности. Налоги для школьников» было протестировано в соответствии с составленным тестовым сценарием. Выявленные ошибки были изучены в ходе тестирования и были устранены.

					ДП.2-400101. 43817.ПЗ-24	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

Листинг кода представлен в Приложении Д.

Результаты тестирования представлены в Приложении Б.

3.5 Разработка документации к программному средству

Документация к программному средству представляет собой совокупность информационных материалов, которые предназначены для описания, объяснения и руководства по использованию программного продукта.

Документация к программному обеспечению включает различные типы документов, которые помогают пользователям и разработчикам понять, использовать и поддерживать программу. Среди таких документов можно выделить: пользовательские руководства, технические спецификации, API-документацию, документацию по тестированию, руководство по разработке и поддержке.

Существует четыре основных вида документации по программному обеспечению.

Архитектурная/проектная – тип документации, представляющий собой обзор программного продукта, включающий описание окружающей среды и основных принципов, используемых при его разработке.

Техническая – вид документации, содержащий информацию о коде программы, алгоритмах, используемых интерфейсах и API. Она предназначена для разработчиков и содержит детальное описание внутренней структуры и функциональности программного продукта.

Пользовательская – тип документации, включающий в себя руководства для конечных пользователей, администраторов системы и другого персонала.

Маркетинговая – вид документации, предназначенный для привлечения потенциальных клиентов и продвижения продукта на рынке.

Руководство пользователя описывает основные шаги и действия, которые пользователь должен выполнить, чтобы освоить программу и начать использовать ее в повседневной работе. Этот документ является ключевым ресурсом для новых пользователей, помогая им быстро освоиться с программным обеспечением и решить возникающие проблемы.

Оно может включать в себя советы по оптимизации работы с программой, подсказки по решению типичных проблем и рекомендации по дальнейшему обучению и развитию навыков использования приложения.

Руководство пользователя представлено в приложении В.

Аннотация к веб-приложению представлена в Приложении Г.

					ДП.2-400101. 43817.ПЗ-24	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

4 Охрана труда, техника безопасности, противопожарные мероприятия и охрана окружающей среды

В Республике Беларусь наблюдается интенсивное внедрение персональных компьютеров в различные сферы человеческой деятельности, что привело к существенному увеличению производительности труда и изменению характера и содержания работы в различных отраслях.

Охрана труда включает в себя комплекс мероприятий и средств, таких как законодательные нормы, социально-экономические и организационные меры, технические и гигиенические стандарты, направленных на обеспечение безопасности, сохранение здоровья и работоспособности человека в процессе труда.

Функции охраны труда включают исследования в области санитарии и гигиены труда, реализацию мер, направленных на снижение вредного воздействия факторов на здоровье работников в процессе их трудовой деятельности. Основным методом охраны труда является применение безопасной техники.

Работа с персональным компьютером и программированием связана с необходимостью длительного пребывания в фиксированной рабочей позе, что может привести к различным заболеваниям опорно-двигательного аппарата, таким как остеохондроз, сколиоз и туннельный синдром. Длительное сидение за компьютером вызывает напряжение мышц шеи, спины и плеч, что со временем может привести к хронической боли и дискомфорту.

В большинстве случаев работа с дисплеем требует минимизации звуковых раздражений, вызванных посторонними шумами. Такие шумы могут отвлекать и снижать концентрацию, что особенно критично при выполнении задач, требующих высокой точности и внимательности. Поэтому для создания оптимальных условий труда важно обеспечить тишину в рабочем помещении, использовать шумопоглощающие материалы.

Персональные электронные вычислительные машины являются электроустановками, и в помещениях, где они используются, могут возникнуть различные аварийные ситуации. Одной из наиболее распространенных проблем является короткое замыкание, которое может произойти из-за изношенности или повреждения проводки, перегрузки электрической сети или неисправности оборудования. Короткое замыкание может привести к возгоранию проводки и оборудования, создавая риск пожара.

Поражение электротоком является еще одной значительной угрозой при работе с ЭВМ. Электрический ток может причинить серьезные травмы или даже привести к летальному исходу.

4.1 Организация рабочего места для работающих на компьютере

Приступая к работе на компьютере, необходимо выполнить ряд действий.

Проверить рабочее место: убедиться, что рабочее место организовано эргономично: монитор находится на уровне глаз, клавиатура и мышь расположены удобно для рук. Обеспечить достаточное освещение рабочего места, избегая бликов на экране.

Проверить оборудование: проверить состояние проводов и соединений, убедиться в отсутствии повреждений или износа. Обязательно убедиться, что все устройства подключены правильно и находятся в исправном состоянии.

Настроить оборудование: отрегулировать высоту стула, чтобы ноги стояли на полу или на подставке под углом 90 градусов.

Провести разминку: сделать несколько простых упражнений для разминки шеи, плеч и рук, чтобы подготовить тело к длительному сидению.

Следить за положением тела: сидеть прямо, с поддержкой для спины. Держать запястья на одном уровне с клавиатурой и избегать чрезмерного сгибания.

Планировать перерывы: организовать рабочий график с регулярными короткими перерывами каждые 45-60 минут. В эти перерывы вставать, ходить, делать упражнения для глаз и тела.

Проверить электробезопасность: убедиться, что все электрические устройства подключены к розеткам через защитные устройства (например, сетевые фильтры).

Обеспечить пожарную безопасность: знать местоположение огнетушителей и аварийных выходов. Убедиться, что доступ к ним не заблокирован.

Психологическая подготовка: создать благоприятную рабочую атмосферу, уменьшив уровень шума и отвлекающих факторов. Убедиться в наличии всех необходимых материалов и информации для выполнения задач.

Ознакомиться с инструкциями и правилами: прочитать и соблюдать инструкции по охране труда и технике безопасности, относящиеся к работе на компьютере.

Выполнение этих шагов поможет создать безопасные и комфортные условия для работы на компьютере, что снизит риск профессиональных заболеваний и повысит общую продуктивность.

Каждый, работающий на ПК, должен соблюдать противопожарный режим, избегать действий, приводящих к пожару, использовать исправные инструменты и оборудование, следовать инструкциям и указаниям по пожарной безопасности, убирать рабочие места и отключать электроприёмники после работы, уметь использовать средства пожаротушения и при обнаружении пожара немедленно сообщать в пожарную охрану, уведомлять руководство и принимать меры к его ликвидации.

					ДП.2-400101. 43817.ПЗ-24	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

При включении компьютера необходимо соблюдать следующую последовательность включения оборудования: включается стабилизатор напряжения (если он используется), включаются периферийные устройства, включается системный блок.

Требования безопасности при работе на компьютерах включают: отсутствие лишних предметов на рабочем месте; избегание касания токопроводящих проводов и задних панелей устройств; соблюдение правильной осанки и дистанции не менее 0,5 метра от экрана; недопустимость снятия защитного экрана и перемещения устройств; запрещение разборки корпуса монитора, системного блока и их компонентов.

По окончании работы необходимо завершить все активные задачи, извлечь съемные носители, выключить системный блок и все периферийные устройства, отключить блок питания и стабилизатор напряжения, если он используется. Затем следует осмотреть и привести в порядок рабочее место, и сообщить о любых неисправностях и замечаниях, выявленных в процессе работы.

4.2 Производительность труда и рабочее место

Хорошо организованное рабочее место, адаптированное под конкретную трудовую деятельность, обеспечивает удобное положение при работе и повышает производительность труда при минимальном физическом и психическом напряжении.

Оптимальное размещение оборудования, достаточное рабочее пространство и правильное освещение являются ключевыми факторами, влияющими на эффективность работы. Правильно организованное пространство позволяет легко осуществлять все необходимые движения и перемещения, что способствует улучшению рабочего процесса.

Кроме того, уровень акустического шума должен быть на уровне, не превышающем допустимых значений, чтобы обеспечить комфортные условия работы и повысить концентрацию.

Правильная организация рабочего места существенно увеличивает производительность труда работника.

Главными элементами рабочего места являются письменный стол и кресло. Основным рабочим положением является положение сидя. Рабочее место для выполнения работ в положении сидя организуется в соответствии с ГОСТ 12.2.032-78.

4.3 Зоны досягаемости рук в горизонтальной плоскости

Зоны досягаемости рук в горизонтальной плоскости играют важную роль при организации рабочего места с персональным компьютером. Зоны определяют участки, которые пользователь может легко достичь, не изменяя положения своего тела или слишком сильно наклоняясь.

Клавиатура должна располагаться в такой позиции, чтобы руки находились в зоне комфортного досягаемости, без излишних усилий или перегибов.

Важно учитывать расположение мыши, чтобы она находилась рядом с клавиатурой и была легко доступна для использования. Регулярное обновление расположения элементов на рабочем столе может помочь уменьшить нагрузку на мышцы и суставы, снизить риск возникновения травм и повысить общую эргономику рабочего места.

Оптимальное размещение предметов труда и документации в зонах досягаемости рук:

- дисплей размещается в зоне а (в центре);
- клавиатура находится в зоне г/д;
- системный блок размещается в зоне б (слева);
- принтер находится в зоне а (справа).

При проектировании письменного стола необходимо учитывать несколько важных аспектов, чтобы обеспечить комфортное и эффективное рабочее место.

Во-первых, высота стола должна быть выбрана таким образом, чтобы программист мог сидеть свободно и удобно, опираясь при необходимости на подлокотники стула, что позволит предотвратить напряжение в спине и шее, снизить риск возникновения дискомфорта при длительной работе за столом.

Во-вторых, нижняя часть стола должна быть сконструирована таким образом, чтобы программист мог сидеть без необходимости поджимать ноги.

Поверхность стола должна быть матовой или иметь специальное покрытие, которое исключает появление бликов в поле зрения программиста.

Конструкция стола должна предусматривать наличие выдвижных ящиков для хранения различных предметов, таких как документация, листинги, канцелярские принадлежности и личные вещи.

4.4 Требования безопасности при работе на персональных электронно-вычислительных машинах

Требования безопасности при работе на персональных электронно-вычислительных машинах представляют собой комплекс мер и правил, направленных на предотвращение возможных опасностей и обеспечение безопасного использования компьютерной техники. Эти требования охватывают широкий спектр аспектов, включая физическую безопасность пользователей, защиту от возможных технических проблем и обеспечение конфиденциальности данных.

Во-первых, важно поддерживать чистоту и порядок на рабочем месте с ПК. Лишние предметы и материалы могут создавать препятствия или повышать риск возгорания,

особенно вблизи электронной техники. Поэтому рабочее место должно быть свободно от мусора и легковоспламеняющихся материалов.

Важно избегать контакта с токопроводящими проводами и обратными панелями монитора и системного блока компьютера, чтобы предотвратить поражение электрическим током. Также следует избегать перегрева компонентов, поэтому не рекомендуется закрывать вентиляционные отверстия компьютера и монитора.

Правильная осанка играет ключевую роль в предотвращении болей в спине и шее при длительной работе за компьютером. Регулярные перерывы и упражнения для разминки мышц также помогают снизить нагрузку на позвоночник и предотвратить развитие мускульных болей.

Следует обращать внимание на правильное расположение монитора. Верхняя часть экрана должна быть на уровне глаз или чуть ниже, чтобы избежать излишнего напряжения в шее.

Негативное влияние компьютеров на здоровье пользователей выражается в повышенном зрительном напряжении, психологической нагрузке, длительном неизменном положении тела в процессе работы, а также воздействии некоторых физических факторов (электромагнитных излучений, статического электричества, ультрафиолетового и рентгеновского излучения).

Важнейшее значение в возникновении зрительного перенапряжения имеет качество более двадцати визуальных параметров изображения на дисплее. Поэтому выполнение требований, установленных действующими стандартами к ним, имеет первостепенное значение в профилактике ухудшения зрения пользователей компьютера.

Компьютеры с жидкокристаллическим экраном не имеют источников мощного электромагнитного излучения и не наводят статического электричества. Однако при использовании блока питания возникает некоторое превышение уровня на частоте 50 Гц, поэтому рекомендуется работать больше с использованием аккумулятора.

4.5 Гигиенические требования к правильной посадке при работе на компьютере

Гигиенические требования к правильной посадке при работе на компьютере играют важную роль в поддержании здоровья и предотвращении профессиональных заболеваний.

Для правильной организации рабочего места используйте регулируемый по высоте стул с хорошей поддержкой для поясницы. Спинка стула должна повторять естественные изгибы позвоночника. Сиденье должно быть удобным, с закругленными краями, и его высота должна позволять ногам стоять на полу или на подставке под углом 90 градусов.

					ДП.2-400101. 43817.ПЗ-24	Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

Высота стола должна быть такой, чтобы клавиатура располагалась чуть ниже уровня локтей, что позволяет держать руки и запястья в естественном положении. Столешница должна быть достаточно большой, чтобы разместить все необходимые предметы, избегая излишнего вытягивания.

При правильном положении тела спина должна быть прямой, с небольшой естественной поддержкой поясницы. Используйте стул с регулируемой спинкой.

Ноги должны стоять на полу или на специальной подставке, образуя угол в 90 градусов в коленях и бедрах. Избегайте скрещивания ног или поджимания их под себя. Руки должны быть расслабленными и лежать на столе или подлокотниках стула, образуя угол около 90 градусов в локтях. Запястья должны быть прямыми и находиться на одной линии с предплечьями.

Важным аспектом является правильное положение монитора. Верхняя часть экрана должна быть на уровне глаз или чуть ниже, чтобы не приходилось наклонять голову вперед или назад, что поможет избежать напряжения шеи и плеч. Расстояние до экрана должно составлять 50-70 см, что примерно соответствует длине вытянутой руки. Экран должен быть наклонен под углом 10-20 градусов, чтобы избежать бликов и обеспечить комфортное чтение.

Освещение рабочего места имеет большое значение. Рабочее место должно быть хорошо освещено, чтобы избежать напряжения глаз. Идеально подходит рассеянное освещение, которое не создает бликов на экране. Используйте настольные лампы с регулируемым светом для освещения рабочей зоны, при этом свет не должен направляться прямо в глаза или на экран монитора.

Не забывайте делать регулярные перерывы каждые 45-60 минут. Вставайте, разминайтесь и делайте упражнения для глаз, чтобы снять напряжение. Каждые 20 минут смотрите на объект, находящийся на расстоянии 20 футов (около 6 метров), в течение 20 секунд, чтобы снизить усталость глаз. В течение рабочего дня выполняйте простые упражнения для шеи, плеч, рук и спины, чтобы улучшить кровообращение и снять мышечное напряжение.

На основании вышеизложенного можно заключить, что правильно организованное рабочее место пользователя компьютера значительно способствует повышению работоспособности и производительности труда, а также сохранению здоровья пользователей компьютерной техники.

5 Экономическая часть

5.1 Характеристика проекта

Веб-приложение «Учимся финансовой грамотности. Налоги для школьников» предназначено для обучения школьников основам финансовой грамотности и налоговой системы. Оно предоставляет доступ к информации о различных видах налогов и их особенностях, позволяет ознакомиться, закрепить и проверить знания в сфере налогообложения.

Для успешной реализации любого проекта необходимо провести его экономическое обоснование. Оно включает в себя оценку экономической эффективности проекта. Для проведения оценки требуется сравнить затраты на разработку проекта с ожидаемыми результатами от его внедрения.

Главной целью дипломного проекта является разработка веб-приложения «Учимся финансовой грамотности. Налоги для школьников».

Для достижения цели необходимо рассчитать расходы на оплату труда и социальные отчисления, материальные затраты, трудоемкость создания программного продукта, а также определить экономический эффект от его разработки и внедрения.

Все расчеты выполнены на 01.05.2024 года с учетом цен, ставки первого разряда, существующих налогов и ценовых отчислений.

5.2 Определение трудоемкости создания программного продукта

Создание программного продукта – сложный процесс, требующий тщательной оценки трудоемкости. Для этого необходимо составить перечень всех видов и этапов работ в соответствии с таблицей 5, которые выполняются в ходе исследования. Особое внимание уделяется упорядочению выполнения задач.

Таблица 4 – План проведения научно-исследовательской работы

Наименование этапов и видов работ	Исполнитель (должность, квалификация)	Количество исполнителей	Трудоемкость, человеко-дни,
1 Составление и согласование технического задания	Техник- программист	1	1
2 Сбор и изучение научно- технической литературы	Техник- программист	1	2
3 Формулирование возможных направлений решения задач	Техник- программист	1	3

Продолжение таблицы 4

4 Разработка методики проведения исследований	Техник-программист	1	3
5 Исследование задач наблюдения и возможность их сведения к задачам	Техник-программист	1	3
6 Разработка и реализация алгоритма решения задачи с большим количеством ограничений	Техник-программист	1	3
7 Оценка некоторого количества задач и решение их	Техник-программист	1	2
8 Решение ряда задач наблюдения	Техник-программист	1	4
10 Составление плана мероприятий по технике безопасности	Техник-программист	1	3
11 Оформление документации по выполненному программному продукту	Техник-программист	1	5
12 Оформление графического материала	Техник-программист	1	3
13 Обобщение результатов исследований	Техник-программист	1	3
14 Составление и оформление отчета	Техник-программист	1	2
Всего:	—	—	37

5.3 Определение полной себестоимости и отпускной цены программного продукта

Цель планирования себестоимости программного продукта – это экономически обоснованное определение всех затрат на его разработку. В плановую себестоимость включаются все расходы, связанные с выполнением проекта, независимо от источника их финансирования. Определение этих затрат осуществляется посредством составления калькуляции плановой себестоимости.

Все расчеты выполнены на 01.05.2024 года с учетом цен, ставки первого разряда, существующих налогов и ценовых отчислений.

Калькуляцию плановой себестоимости программного средства составим по следующим статьям затрат: основная заработная плата, дополнительная заработная плата, отчисления на социальные нужды, материальные затраты, содержание и эксплуатация основных фондов, налоги и сборы, включаемые в себестоимость, накладные расходы, прочие затраты.

Основная заработная плата техника-программиста рассчитывается по формуле:

$$З_О = \sum_{i=1}^n T_{qi} \cdot T_q \cdot \Phi P B_i, \quad (1)$$

где T_{qi} – часовая тарифная ставка i -го разряда;

T_q – количество часов работы в день ($T_q = 8$ ч);

$\Phi P B_i$ – фонд рабочего времени i -го исполнителя.

Часовая тарифная ставка i -го разряда определяется следующим образом:

$$T_{qi} = \frac{B_1}{\mathcal{C}_H} \cdot K_{Ti} \quad (2)$$

где B_1 – базовая ставка 1-го разряда, установленная законодательно (на 01.05.2024 – 250 руб);

\mathcal{C}_H – нормативное количество рабочих часов в месяц. Для пятидневной рабочей недели с выходными днями в субботу и воскресенье расчетная норма рабочего времени в 2024 году составляет 2016 часов (168) часов в месяц).

K_{Ti} – тарифный коэффициент i -того разряда.

Работа техника-программиста тарифицируется 4 тарифным разрядом ($K_{Ti} = 1,21$).

Определим часовую тарифную ставку техника-программиста 4 разряда:

$$T_{qi} = \frac{250}{168} \cdot 1,21 = 1,80 \text{ руб.}$$

Рассчитаем на основе часовой тарифной ставки основную заработную плату исполнителя:

$$З_О = 1,80 \cdot 8 \cdot 37 = 532,8 \text{ руб.}$$

Дополнительная заработная плата определяется по нормативу в процентах к основной заработной плате:

$$З_Д = \frac{З_О \cdot H_Д}{100}, \quad (3)$$

где $H_Д$ – норматив дополнительной заработной платы в целом по организации (30%).

Определим размер дополнительной заработной платы:

					ДП.2-400101. 43817.ПЗ-24	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

$$З_{\text{д}} = \frac{532,8 \cdot 30}{100} = 159,84 \text{ руб.}$$

Отчисления в Фонд социальной защиты населения определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной заработной платы исполнителей. Сумма отчислений в фонд социальной защиты населения ($З_{\text{сз}}$) определяется по формуле:

$$O_{\text{ФСЗН}} = \frac{(З_{\text{о}} + З_{\text{д}}) \cdot H_{\text{ФСЗН}}}{100}, \quad (4)$$

где $H_{\text{ФСЗН}}$ – норматив отчислений в Фонд социальной защиты населения ($H_{\text{сз}} = 34\%$).

$$O_{\text{ФСЗН}} = \frac{(532,8 + 159,84) \cdot 34}{100} = \frac{692,64 \cdot 34}{100} = 235,50 \text{ руб.}$$

Кроме отчислений в Фонд социальной защиты населения в учреждения образования производятся отчисления в Белгосстрах. Определим их следующим образом:

$$O_{\text{БГС}} = \frac{(З_{\text{о}} + З_{\text{д}}) \cdot H_{\text{БГС}}}{100},$$

$$O_{\text{БГС}} = \frac{(З_{\text{о}} + З_{\text{д}}) \cdot H_{\text{БГС}}}{100}, \quad (5)$$

где $H_{\text{БГС}}$ – норматив отчислений в Белгосстрах ($H_{\text{БГС}} = 0,09\%$).

$$O_{\text{БГС}} = \frac{(532,8 + 159,84) \cdot 0,09}{100} = \frac{692,64 \cdot 0,09}{100} = 0,62 \text{ руб.}$$

Расходы по статье «Материальные затраты» включают стоимость израсходованных материалов и затраты на электроэнергию.

Стоимость израсходованных материалов я определила исходя из фактического расхода и цены приобретения (таблица 6).

Таблица 5 – Расчет стоимости израсходованных материалов

Наименование материалов	Ед. изм.	Расход	Цена за единицу, руб.	Стоимость, руб.
Бумага	уп. (500 листов)	1	13,90	13,90
CD-диски	Шт	1	3,50	3,50
Краска для картриджа	Заправка	1	21,00	22,00
Итого:	—	—	—	39,40

Затраты на электроэнергию ($З_э$) определим следующим образом:

$$3_{\mathfrak{I}} = C_{\mathfrak{I}} \cdot (T_{PK} \cdot W_{PK} + T_{ПРИНТ} \cdot W_{ПРИНТ}), \quad (6)$$

где C_3 – тариф за 1 кВт/ч электроэнергии, руб.;

$T_{ПК}$, $T_{ПРИНТ}$ – время работы персонального компьютера и принтера соответственно, час.;

$W_{ПК}$, $W_{ПРИНТ}$ – номинальная потребляемая мощность персонального компьютера и принтера соответственно, кВт/ч.

$$3_3 = 0,45 \cdot (130 \cdot 0,4 + 1 \cdot 0,04) = 23,41 \text{ pyб.}$$

Расходы на содержание и эксплуатацию основных фондов включают суммы амортизационных отчислений от стоимости ЭВМ и от стоимости площадей, занятых ЭВМ.

Годовая величина амортизационных отчислений от стоимости ЭВМ определяется по формуле:

$$A_{\mathcal{A}BM} = \frac{U_{\mathcal{A}BM} \cdot k_Y \cdot H_{\mathcal{A}BM}}{100} = \frac{U_{\mathcal{B}BM} \cdot H_{\mathcal{A}BM}}{100}, \quad (7)$$

где $C_{ЭВМ}$ – цена ЭВМ на момент приобретения (руб.);

k_v – коэффициент удорожания ЭВМ ($k_v = 1,007$);

$H_{AЭВМ}$ – норма амортизационных отчислений от стоимости ЭВМ ($H_{AЭВМ} = 10\%$);

$C_{БЭВМ}$ – балансовая стоимость ЭВМ.

$$A_{\text{ЭБМ}} = \frac{3200 \cdot 1,007 \cdot 10}{100} = 322,24 \text{ руб.}$$

Величина амортизационных отчислений за период разработки программного продукта (37 дней):

$$A_{\text{ЭВМ}} = 322,24 \div 366 \cdot 37 = 32,58 \text{ руб.}$$

Величина амортизационных отчислений от стоимости производственных площадей, занятых ЭВМ определяются по формуле:

$$A_{\Pi\Lambda} = \frac{S \cdot \varrho_{\Pi\Lambda} \cdot H_{\Lambda\Pi\Lambda}}{100} = \frac{\varrho_{\text{БП\Lambda}} \cdot H_{\Lambda\Pi\Lambda}}{100}, \quad (8)$$

где S – площадь, занимаемая ЭВМ ($S = 4\text{м}^2$);

$C_{пл}$ – цена 1 м² производственной площади;

$H_{АПЛ}$ – норма амортизационных отчислений на производственные площади ($H_{АПЛ} = 1\%$);

$C_{\text{БПД}}$ – балансовая стоимость площадей, занимаемых ЭВМ.

$$A_{\text{пл}} = \frac{4 \cdot 128,60 \cdot 1}{100} = \frac{514,4}{100} = 5,14 \text{ руб.}$$

Величина амортизационных отчислений от стоимости производственных площадей занятых ЭВМ за период разработки программного продукта (37 дней):

$$A_{\text{пл}} = 5,14 \div 366 \cdot 37 = 0,51 \text{ руб.}$$

К налогам и сборам, включаемым в себестоимость, относится налог на недвижимость. Величина налога на недвижимость определяется по формуле:

$$H_H = \frac{C_{\text{БПЛ}} \cdot C_{\text{НН}}}{100}, \quad (9)$$

где $C_{\text{НН}}$ – ставка налога на недвижимость ($C_{\text{НН}} = 1\%$).

$$H_H = \frac{514,4 \cdot 1}{100} = 5,14 \text{ руб.}$$

$$H_H = \frac{5,14}{366 \cdot 37} = 0,38 \text{ руб.}$$

Накладные расходы (P_H) – затраты, связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств, а также с расходами на общехозяйственные нужды, вычисляются по формуле:

$$P_H = \frac{3_O \cdot H_{PH}}{100}, \quad (10)$$

где H_{PH} – норматив накладных расходов в целом ($H_{PH} = 13\%$).

$$P_H = \frac{532,8 \cdot 13}{100} = 69,26 \text{ руб.}$$

Прочие затраты на конкретное программное средство (P_3) включают в себя затраты на приобретение и подготовку специальной литературы и специальной научно-технической информации, и определяются по формуле:

$$P_3 = \frac{3_O \cdot H_{ПЗ}}{100}, \quad (11)$$

где $H_{ПЗ}$ – норматив прочих затрат ($H_{ПЗ} = 5\%$).

$$P_3 = \frac{532,8 \cdot 5}{100} = 26,64 \text{ руб}$$

Общую сумму расходов по всем статьям определим в таблице 6.

Таблица 6 – Калькуляция плановой себестоимости программного продукта

№ п/п	Статья затрат	Сумма, руб.
1	Основная заработная плата	532,8
2	Дополнительная заработная плата	159,84

Продолжение таблицы 6

3	Отчисления на социальные нужды – всего	243,05
	– в государственный фонд социальной защиты населения	236,12
	– в Белгосстрах	0,62
4	Материальные затраты – всего	62,81
	в том числе:	39,40
	– материалы	
	– электроэнергия	23,41
5	Содержание и эксплуатация основных фондов – всего	37,72
	в том числе:	32,58
	– амортизационные отчисления от стоимости ЭВМ	
	– амортизационные отчисления от стоимости производственных площадей, занимаемых ЭВМ	5,14
6	Налоги и сборы, включаемые в себестоимость	5,14
7	Накладные расходы	69,26
8	Прочие затраты	26,64
	ИТОГО (C_{Π})	1137,26

Расчет прогнозируемой прибыли (Π_{Π}) по разрабатываемому ПС по формуле:

$$\Pi_{\Pi} = \frac{C_{\Pi} \cdot U_{\Pi\Pi}}{100}, \quad (12)$$

где $U_{\Pi\Pi}$ – уровень рентабельности ПС ($U_{\Pi\Pi} = 10\%$).

$$\Pi_{\Pi} = \frac{1137,26 \cdot 10}{100} = 113,72 \text{ руб.}$$

Прогнозируемая цена без налога на добавленную стоимость (Π) складывается из полной себестоимости программного средства и прогнозируемой прибыли. Рассчитывается по формуле:

$$\Pi = C_{\Pi} + \Pi_{\Pi}, \quad (13)$$

$$\Pi = 1137,26 + 113,72 = 1250,98 \text{ руб.}$$

На реализацию создаваемого программного средства льготы по налогу на добавленную стоимость (НДС) не распространяются. В связи с этим НДС рассчитывается по формуле:

$$НДС = \frac{Ц \cdot C_{НДС}}{100}, \quad (14)$$

где $C_{НДС}$ – ставка налога на добавленную стоимость ($C_{НДС} = 20\%$).

$$НДС = \frac{1250,98 \cdot 20}{100} = 250,20 \text{ руб.}$$

Прогнозируемая отпускная цена ($Ц_0$) определяется по формуле:

$$Ц_0 = Ц + НДС \quad (15)$$

$$Ц_0 = 1250,98 + 250,20 = 1501,18 \text{ руб.}$$

Прибыль, остающаяся в распоряжении организации-разработчика, определяется по формуле:

$$П_ч = П_п - \frac{П_п \cdot H_{НП}}{100}, \quad (16)$$

где $H_{НП}$ – норматив налога на прибыль ($H_{НП} = 18\%$).

$$П_ч = 113,72 - \frac{113,72 \cdot 18}{100} = 93,25 \text{ руб.}$$

Чистая прибыль от реализации ПС остается организации-разработчику и представляет собой экономический эффект от создания нового программного средства.

5.4 Определение экономического эффекта от внедрения программного продукта

Определение экономии времени отражено в таблице 7.

Таблица 7 – Определение экономии времени

Перечень услуг программного продукта	Время до автоматизации, час.	Время после автоматизации, час.	Экономия времени, час.	Количество операций в год	Годовая экономия времени, час.
Поиск налоговых терминов	0,5	0,1	0,4	300	120
Поиск информации по определенной теме налогообложения	0,5	0,1	0,4	400	160

Продолжение таблицы 7

Просмотр видеоролика по определенной теме налогообложения	0,5	0,1	0,4	403	161,2
Просмотр налогового календаря за текущий год	0,4	0,1	0,3	250	75
Просмотр исторических дат в сфере налогообложения	0,5	0,1	0,4	200	80
Итого:	—	—	—	—	596,2

Внедрение разработанного программного средства приведёт к условной годовой экономии фонда рабочего времени исполнителя в размере 596,2 часа, т.е. снизит трудоёмкость отдельных операций исполнителя. Сэкономить можно приблизительно 0,29 тарифной ставки исполнителя.

Исполнителем программного средства является классный руководитель.

Фонд заработной платы за месяц классного руководителя:

- $З_О$ – 547,60 руб. (основная заработная плата, формула 2);
- $З_Д$ – 164,28 руб. (дополнительная заработная плата, формула 3);
- $З_{СЗ}$ – 242,40 руб. (отчисления в ФСЗН, формула 4);
- $З_Б$ – 0,65 руб. (отчисления в Белгосстрах, формула 5).

Тогда годовой фонд заработной платы равен:

$$\text{ГДФЗП} = (З_О + З_Д + О_{сн}) * 12 = 954,93 * 12 = 11459,16 \text{ руб.} \quad (17)$$

Условная экономия по заработной плате составит:

$$\text{ЭК.ЗП} = \text{ГДФЗП} * К = 11459,16 * 0,29 = 3323,16 \text{ руб.} \quad (18)$$

Экономический эффект от внедрения веб-приложения «Учимся финансовой грамотности. Налоги для школьников» составит:

$$\text{ЭК.ЭФФЕКТ} = \text{ЭК.ЗП} - Ц_О = 3323,16 - 1501,18 = 1821,98 \text{ руб.} \quad (19)$$

Подводя итог можно сказать, что разработка веб-приложения приложения «Учимся финансовой грамотности. Налоги для школьников» позволит получить экономический эффект от его реализации в сумме 1821,98 рублей.

5.5 Выводы

В дипломном проекте создано, протестировано и подготовлено к эксплуатации веб-приложение «Учимся финансовой грамотности. Налоги для школьников». Оно предоставляет обучение основам налогообложения и финансовой грамотности с помощью трех этапов: ознакомиться, закрепить, проверить.

Экономический эффект от реализации веб-приложения «Учимся финансовой грамотности. Налоги для школьников» составляет 1821,98 рублей.

					ДП.2-400101. 43817.ПЗ-24	Лист
						44
Изм.	Лист	№ докум.	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

В результате дипломного проектирования разработано веб-приложение «Учимся финансовой грамотности. Налоги для школьников», позволяющее школьникам изучать основы налогообложения через интерактивные материалы, тесты и упражнения. Программный продукт прошел всестороннее тестирование и готов к эксплуатации.

Внедрение информационной системы «Учимся финансовой грамотности. Налоги для школьников» позволит получить экономический эффект в размере 1821,98 рублей за счет сокращения затрат на печатные материалы и учебные пособия. Разработанное веб-приложение обеспечит доступность образовательных ресурсов, повышение уровня финансовой грамотности школьников и улучшение качества обучения за счет использования современных интерактивных технологий.

В качестве дальнейшего совершенствования веб-приложения планируется разработка дополнительных разделов, улучшение пользовательского интерфейса для повышения его информативности и привлекательности, а также интеграция с внешними образовательными ресурсами для расширения возможностей обучения.

					ДП.2-400101. 43817.ПЗ-24	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Предметная область и ее структура [Электронный ресурс]/ Файловый архив студентов – Режим доступа: <https://studfile.net/preview/7028484/page:2/> – Дата доступа: 18.05.2024.
2. Требования к программному обеспечению [Электронный ресурс]/ Системная инженерия ЛЕКЦИЯ 4 – Режим доступа: <https://studfile.net/preview/9507549/page:9/> – Дата доступа: 20.05.2024.
3. Лекция №5 Модели данных Терминология [Электронный ресурс]/ Файловый архив студентов. – Режим доступа: <https://studfile.net/preview/1705291/> – Дата доступа: 27.05.2024.
4. Проектирование информационного обеспечения [Электронный ресурс]/ Studbooks.net – Режим доступа: https://printsipy_proektirov_informatsionn_obespecheniya – Дата доступа: 22.05.2024.
5. Бесплатный сервис Draw.io – обзор инструмента, работа с сервисом, возможности [Электронный ресурс]/ Практикум – Режим доступа: <https://blog/vozmozhnosti-drawio/> – Дата доступа: 20.05.2024.
6. Основные понятия и определения баз данных [Электронный ресурс]/ Файловый архив студентов. – Режим доступа: <https://studfile.net/preview/2918106/page:2/> – Дата доступа: 21.05.2024.
7. Пользовательский интерфейс [Электронный ресурс]/ Фоксфорд – Режим доступа: <https://foxford.ru/wiki/informatika/polzovatelskij-interfejs> – Дата доступа: 20.05.2024.
8. Структуры данных: основы алгоритмов [Электронный ресурс]/ Медиум – Режим доступа: <https://medium.com/nuances-of-programming/структуры-данных-основы-алгоритмов-72b1a4021907> – Дата доступа: 20.05.2024.
9. Блок-схема [Электронный ресурс]/ Диаграммы и блок-схемы aspose – Режим доступа: <https://products.aspose.app/diagram/ru/pages/what-is-flowchart> – Дата доступа: .
10. HTML [Электронный ресурс]/ W3schools – Режим доступа: <https://osnovy-html/> – Дата доступа: 21.05.2024.
11. Информационная безопасность [Электронный ресурс]/ Министерство образования и науки Республики Беларусь – Режим доступа: <https://mathsec/course-mathsec-basic-concepts/> – Дата доступа: 20.05.2024.
12. Тестирование программного средства [Электронный ресурс]/ Медиум – Режим доступа: <https://blog/testirovanie-programmnogo-obespecheniya/> – Дата доступа: 21.05.2024.