

ГБПОУ СО «ЧАПАЕВСКИЙ ГУБЕРНСКИЙ КОЛЛЕДЖ им. О. Колычева»

ОТЧЕТ

о прохождении преддипломной практики
студента 4 курса, 44а группы

Ф. И. О. Григоренко Даниила Владимировича

Специальность: 09.02.01 Компьютерные системы и комплексы

Период прохождения практики: 20.04.2022 – 17.05.2022гг.

Место прохождения практики: ООО «Фотон»

Руководитель практики от организации:

Аксенов Андрей Рустямович, системный администратор ООО «Фотон»
(фамилия, имя, отчество, должность)

Руководители практики от колледжа: Сидоров С.А., Суворова Л.Е.

 5 (отлично)
(оценка)
17.05.2022 год
(подпись, дата)

г. Чапаевск
2022

Содержание

Введение

1. Описание базы практики
2. Выполнение работ по программному модулю 01
3. Выполнение работ по программному модулю 02
4. Выполнение работ по программному модулю 03
5. Выполнение работ по программному модулю 04
6. Выполнение индивидуального задания на ВКР

Вывод

Приложение

Введение

Преддипломная практика была пройдена в Обществе с ограниченной ответственностью «Фотон» (далее - ООО «Фотон», организация).

ООО «Фотон» расположено по адресу: Самарская область, г.о. Чапаевск, ул. Кирова, 34.

Руководителем является Семендяй Владимир Яковлевич.

Руководитель (наставник) практики от ООО «Фотон»: Системный администратор, Аксёнов Андрей Рустямович.

Руководители практики от колледжа: преподаватели профессиональных модулей: Сидоров Сергей Александрович, Суворова Любовь Евгеньевна.

Описание базы практики

Общество с ограниченной ответственностью (ООО) «Фотон»

Местонахождение и почтовый адрес Общества: Россия, 446100, Самарская область, г. Чапаевск, ул.Кирова, д.34.

Основными видами деятельности ООО «Фотон» являются:

- а) Деятельность по трансляции телерадиоканалов по сетям кабельного телерадиовещания;
- б) Деятельность в области радиовещания
- в) Деятельность в области телевизионного вещания
- г) Деятельность по предоставлению услуг связи для целей открытого эфирного вещания

Целью деятельностью Общества является получение прибыли.

Для достижения поставленной цели, Общество в установленном законодательством порядке, осуществляет виды деятельности указанные выше.

**Выполнение работ по
программному модулю 01
«Проектирование цифровых
устройств»**

На первом этапе преддипломной практики, по программному модулю 01, в соответствии с заданием были выполнены следующие задачи:

Были составлены принципиальная и структурная схемы телевизионного ВЧ модулятора (рисунок 1, рисунок 2):

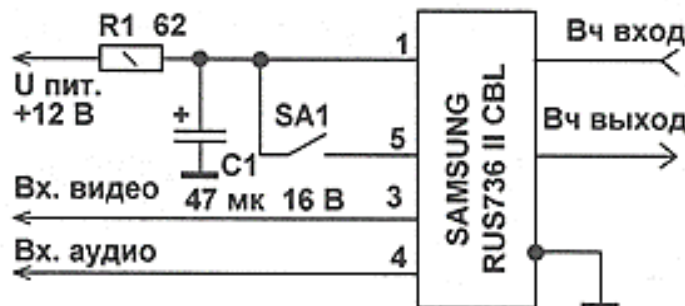


Рисунок 1. Принципиальная схема телевизионного ВЧ модулятора

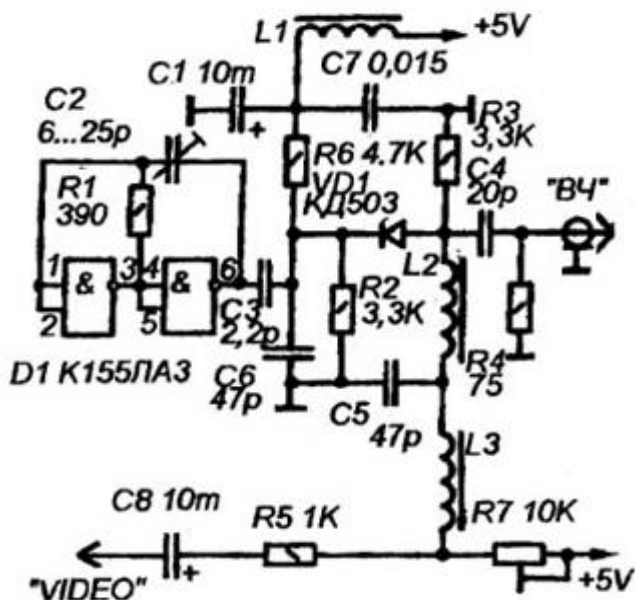


Рисунок 2. Структурная схема телевизионного ВЧ модулятора

Далее, было произведено моделирование схемы Мультиплексора в программной среде автоматизированного проектирования Multisim.

Multisim - средство разработки и моделирования электронных схем. NI Multisim позволяет создать схему, используя обширную библиотеку компонентов, и эмулировать поведение интегральной схемы с помощью стандартного промышленного симулятора SPICE. Начиная с версии 10.1 в Multisim интегрирован MCU Module, позволяющий добавить в SPICE-эмулированную интегральную схему микроконтроллер и запрограммировать его

на С или Ассемблере. Модуль позволяет эмулировать работу интегральной схемы с микроконтроллером и различными дополнительными устройствами: RAM, ROM, клавиатурой, а также графическими и буквенно-цифровыми ж-к дисплеями.

Была смоделирована схема мультиплексора в среде Multisim. Схема представлена на рисунке 3.

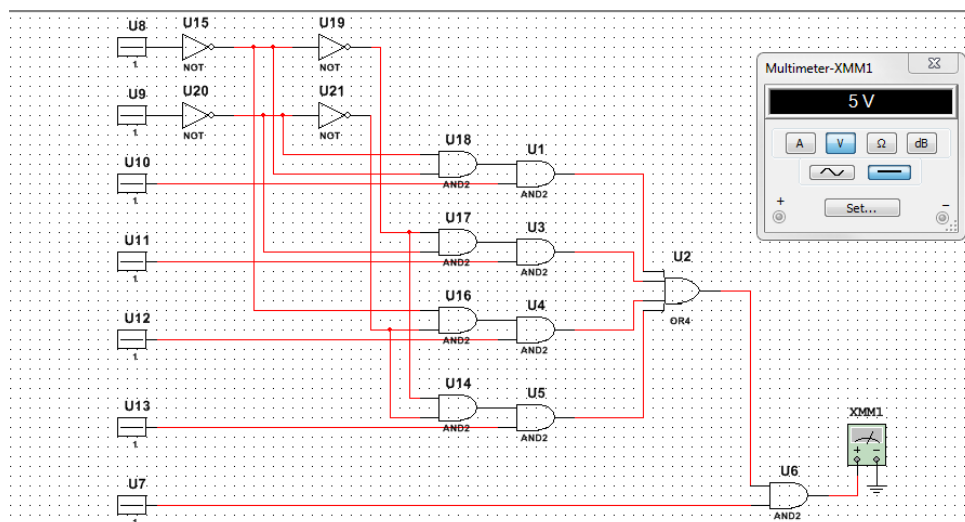


Рисунок 3. Схема мультиплексора в среде Multisim

Схема состоит из двух линий, в каждой из которых по два последовательно включенных элемента «НЕ».

Также была выполнена работа по определению показателей надежности и качества проектируемых цифровых устройств. В частности, были изучены измерительные приборы – мультиметр и осциллограф. Изучены функции, назначение и характеристики данных приборов.

Мультиметр (тестер, авометр) - комбинированный электроизмерительный прибор, объединяющий в себе несколько функций. Внешний вид представлен на рисунке 4.

В минимальном наборе включает функции вольтметра, амперметра и омметра. Иногда выполняется мультиметр в виде токоизмерительных клещей.

Существуют цифровые и аналоговые мультиметры.

Мультиметр может быть, как лёгким переносным устройством, используемым для базовых измерений и поиска неисправностей, так и сложным стационарным прибором со множеством возможностей.



Рисунок 4. Внешний вид цифрового мультиметра DT-9928T

Таблица 1. Характеристики мультиметра

Модель	DT-9928T
Постоянное напряжение	200мВ, 2В, 20В, 200В, 500В
Переменное напряжение	200В, 500В
Защита от перегрузки	500В (Действующее значение)
Сопротивление	200Ом, 2000Ом, 20кОм, 200кОм, 2000кОм
Постоянный ток	2000мкА, 20мА, 200мА
Тестовый ток диодов	1,6мА (макс)
Тестовое напряжение диодов	3.2В (макс)
Диапазон батареи	9В
Тестовый ток батареи	6мА

Осциллограф - прибор, предназначенный для исследования (наблюдения,записи,измерения) амплитудных и временных параметров электрического сигнала, подаваемого на его вход, и наглядно отображаемого (визуализации) непосредственно на экране либо регистрируемого на фотоленту.

Внешний вид представлен на рисунке 5.

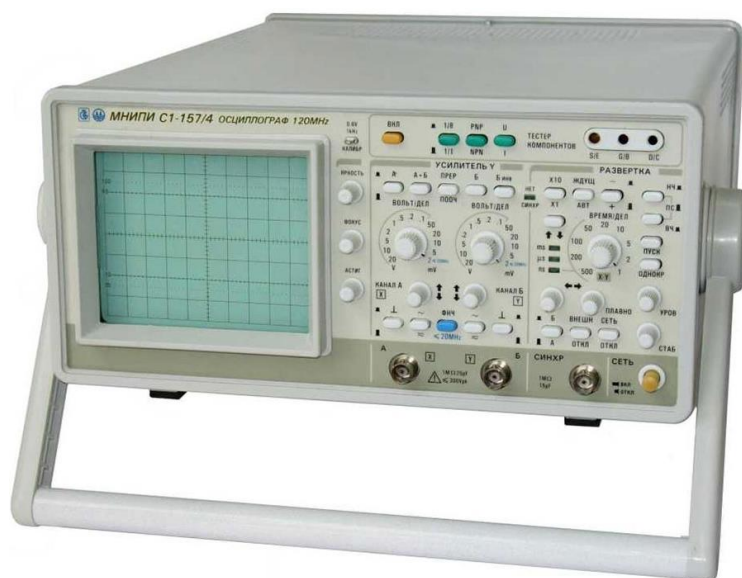


Рисунок 5. Внешний вид осциллографа МНИПИ С1-157

Таблица 1. Характеристики осциллографа МНИПИ С1-157

Модель	МНИПИ С1-157
Число каналов	2
Полоса пропускания:	0-100 мГц
Коэффициенты отклонения	5 мВ/дел-5 В/дел (шаг 1-2-5)
Погрешность установки	$\pm 3\%$
Время нарастания ПХ	$\leq 3,5$ нс
Входной импеданс	1 МОм/25 пФ, (с делителем 1:10 10 МОм/12 пФ)
Входное напряжение:	
с открытым входом	≤ 90
с закрытым входом	≤ 100
с делителем 1:10	≤ 250

**Выполнение работ по
программному модулю 02
«Применение
микропроцессорных систем,
установка и настройка
периферийного
оборудования»**

На втором этапе преддипломной практики, по программному модулю 02, в соответствии с заданием были выполнены следующие задачи:

Было произведено программирование микроконтроллера модулятора. На Рисунке 6 изображен внешний вид платы модулятора.



Рисунок 6. Внешний вид платы модуля (модулятора)

Программирование производилось с помощью специализированного устройства программирования модулей (модуляторов). Внешний вид программатора представлен на Рисунке 7.



Рисунок 7. Внешний вид программатора

Процесс программирования состоял из нескольких этапов: установки соединения программатора с модулятором; проверки состояния устройства

(технических параметров); загрузки файла обновления прошивки; настройки компонентов модулятора.

Далее, согласно заданию, было произведено тестирование и отладка микропроцессорных систем. В частности, была произведена настройка цифрового демодулятора DVB.

На рабочем месте сотрудника организации был установлен и сконфигурирован персональный компьютер.

При установке были применены приемы обеспечения устойчивой работы компьютерных систем и комплексов с применением правил и нормы охраны труда, техники; безопасности, промышленной санитарии и противопожарной защиты.

Далее после установки ПК, к системному блоку были подключены периферийные устройства: монитор, клавиатура, мышь, аудиосистема, многофункциональное устройство. В системном блоке был смонтирован накопитель информации (жёсткий диск).

После установки и подключения устройств была произведена инсталляция операционной системы на ПК – Windows 10 (Рисунок 8).

После установки ОС было проведено конфигурирование и настройка операционной системы, установка драйверов (Рисунок 9).

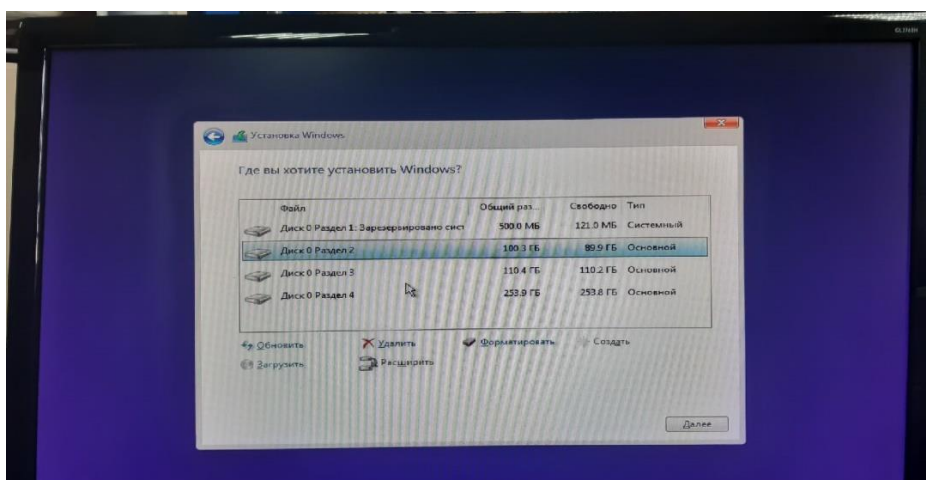


Рисунок 8. Инсталляция ОС Windows 10 на ПК

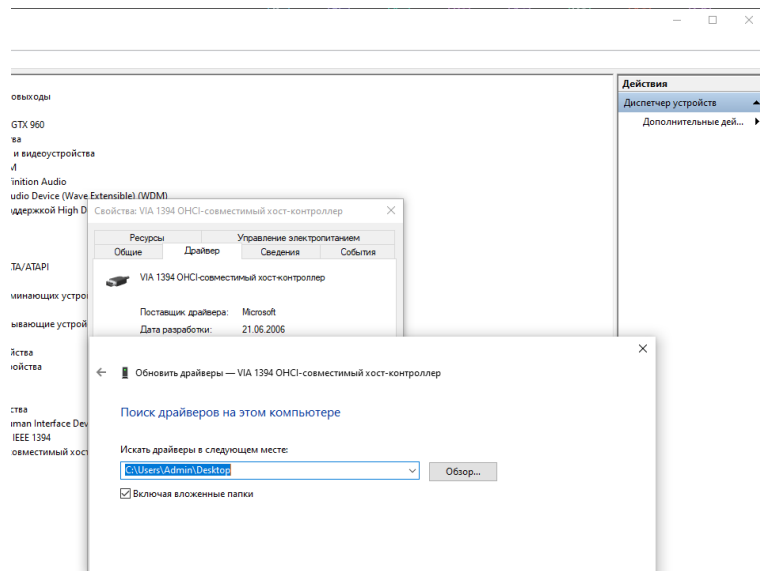


Рисунок 9. Установка драйверов

В персональном компьютере сотрудника организации в порт PCI Express была установлена плата ввода/вывода аудио/видеосигнала (рисунок 10), а также было установлено программное обеспечение для функционирования этого оборудования.

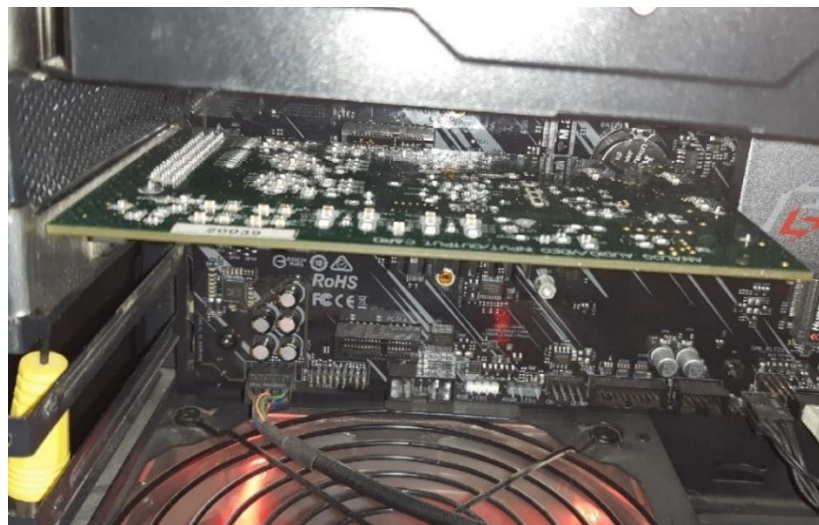


Рисунок 10. Установленная плата ввода/вывода аудио/видеосигнала

После того как плата была установлена в компьютер, в дисковод был вставлен компакт-диск входящий в комплект поставки, с фирменным программным обеспечением.

В соответствии с инструкцией входящий в комплект поставки, была выполнена регистрация приобретенного продукта.

Далее, было установлено программное обеспечение (драйвер) устройства с помощью установочного исполняемого файла (.exe) (рисунок 11).

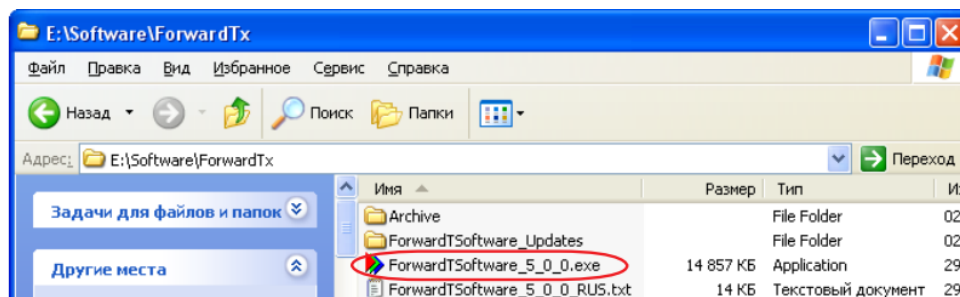


Рисунок 11. Установочный исполняемый файл

В ходе установки драйвера, программа установки предложила дополнительно установить пакет NET.Framework, а также компоненты DirectX (рисунок 12).

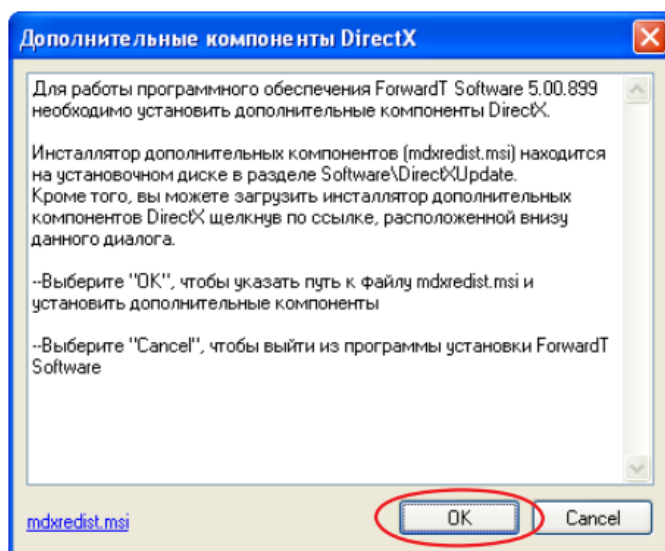


Рисунок 12. Окно установщика дополнительных компонентов DirectX

В завершении установки, программа предложила перезагрузить компьютер (рисунок 13).

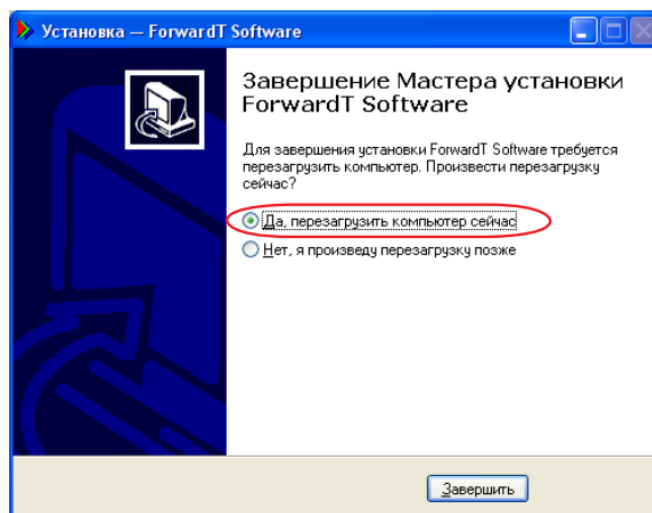


Рисунок 13. Окно установщика дополнительных компонентов DirectX

На завершающем этапе прохождения практики был разработан проект микропроцессорной системы с использованием пневматических элементов.

При разработке проекта использовалось следующее оборудование:

- Компрессор
- Распределитель Camozzi
- Модуль Camozzi с релейно-контактными устройствами управления
- Модуль Camozzi с логическим контроллером Siemens LOGO!
- Пневмоцилиндр двустороннего действия
- Отведения, шланги, соединители

Внешний вид модулей Camozzi представлен на Рисунке 14.

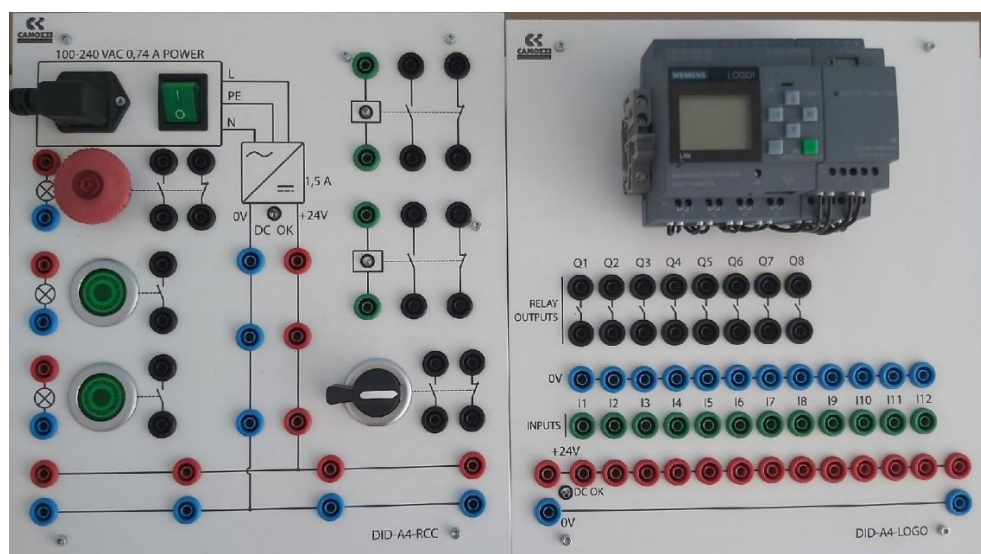


Рисунок 14. Внешний вид модулей Camozzi

Контроллер Siemens LOGO! предназначен для управления технологическим оборудованием в соответствии с загруженной в него программой.

Также Siemens LOGO! применяется в различных областях, такие как: управление насосами, вентиляторами, компрессорами, прессами, эскалаторами, подъемниками, в системах отопления; управление коммутационной аппаратурой; в системах управления дорожным движением и конвейерных системах.

Внешний вид контроллера Siemens LOGO! представлен на Рисунке 15.



Рисунок 15. Внешний вид контроллера Siemens LOGO!

По схеме контроллер Siemens LOGO! управляет распределителем воздуха через специальный интерфейс.

Распределитель в свою очередь получив сигнал от контроллера подает воздух на пневмоцилиндр.

Пневмоцилиндры делятся на два типа: пневмоцилиндры одностороннего действия и пневмоцилиндр двухстороннего действия. В цилиндрах одностороннего действия поршень движется под силой сжатого воздуха только в ту сторону, в которую подается воздух. Возврат поршня производится за счет силового действия возвратной пружины или действиями инерции. При всем этом воздух, который подается в пневмоцилиндр, должен обязательно сбрасываться в атмосферу. В этих цилиндрах пружина должна снижать усилие прямого хода, которое развивается цилиндром под действием сжатого воздуха.

В цилиндрах, которые работают по принципу двухстороннего действия, шток движется под действием сжатого воздуха, во всех заданных направлениях. Для цилиндров такого типа прямой и обратный ход, являются рабочими. Стоит отметить что усилие, которое развивает пневмоцилиндр в обратном ходе существенно ниже усилия, развиваемого им при прямом ходе.

Пневмоцилиндром можно управлять, подсоединив его к распределителю.

В зависимости от типа используемого распределителя (нормально открытый или нормально закрытый) его шток будет выдвигаться или втягиваться при включении распределителя и возвращаться обратно при прекращении воздействия. Это так называемое «прямое управление». Такое управление применяется в пневмоцилиндрах с малым расходом воздуха.

Функционирование схемы осуществляется двумя кнопками без фиксации. При однократном нажатии кнопки «Пуск» начинается работа пневмоцилиндра, заключающаяся в полном выдвигении штока. При однократном нажатии кнопки «Стоп» система возвращается в исходное состояние, а её работа прекращается.

С помощью компьютера с предустановленной средой программирования была разработана схема управления пневмоцилиндром и программа работы микроконтроллера Siemens LOGO!.

Схема представлена на Рисунке 16.

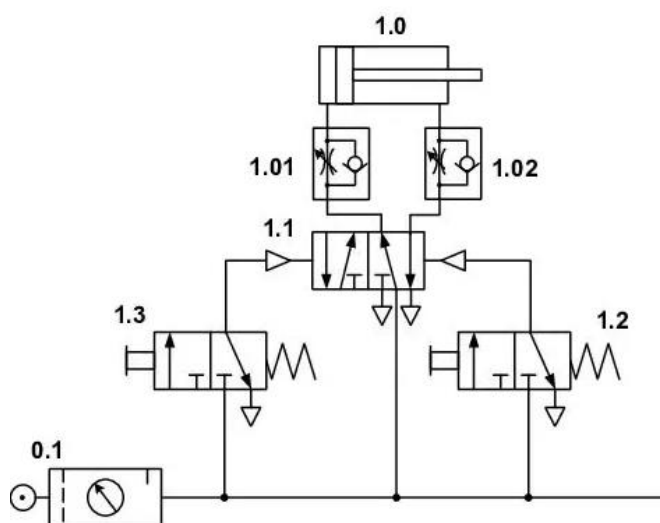


Рисунок 16. Схема управления пневмоцилиндром

**Выполнение работ по
программному модулю 03
«Техническое обслуживание
и ремонт компьютерных
систем и комплексов»**

Был проведен контроль, диагностика и восстановление работоспособности персонального компьютера и многофункциональных устройств организации (Рисунок 17,18). Были применены основные методы диагностики, использованы аппаратные и программные средства функционального контроля.



Рисунок 17. Восстановление работоспособности персонального компьютера



Рисунок 18. Восстановление работоспособности многофункционального устройства

Было проведено системотехническое обслуживание компьютерных систем и комплексов. Для систематического обслуживания была применено сервисное средство – программа AIDA64 Extreme Edition (Рисунок 19).

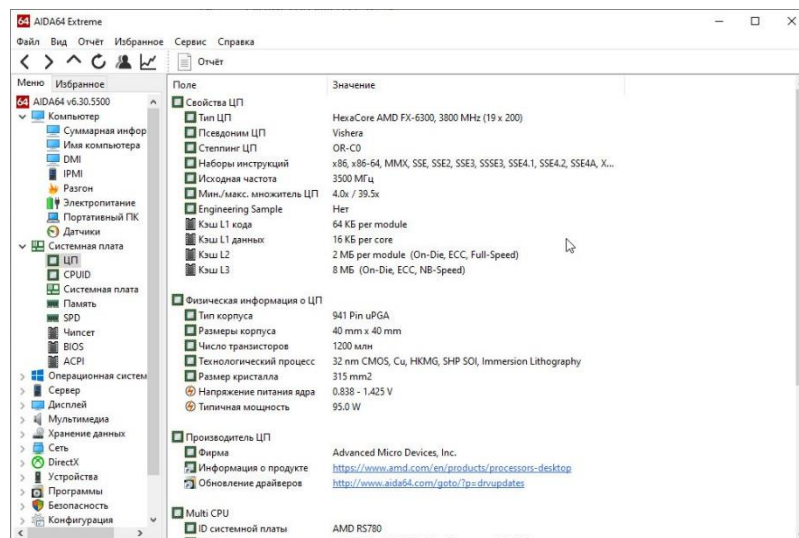


Рисунок 19. Работа с программой AIDA64 Extreme Edition

Осуществлена инсталляция операционной системы на ПК – Windows 10 (Рисунок 20).

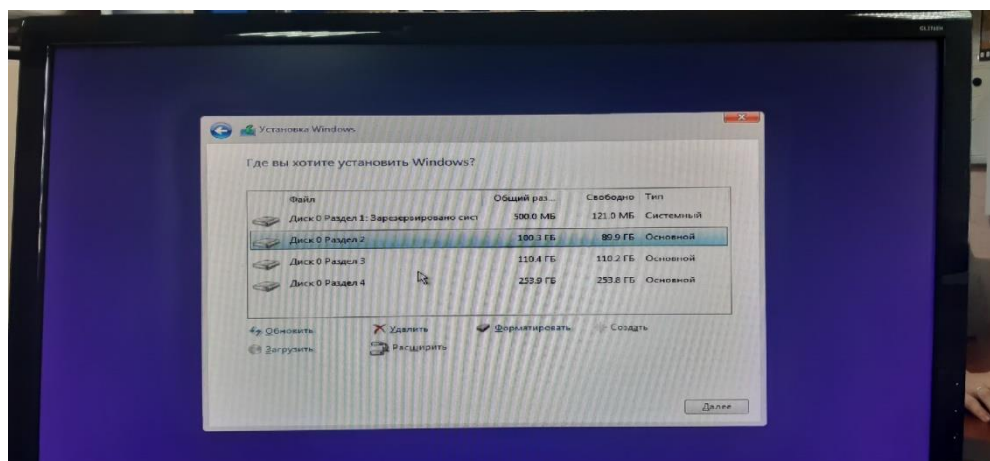


Рисунок 20. Инсталляция ОС Windows 10 на ПК

Проведено конфигурирование и настройка операционной системы (Рисунок 21), установка драйверов (Рисунок 22). Применены приемы обеспечения устойчивой работы компьютерных систем и комплексов с применением правил и нормы охраны труда, техники; безопасности, промышленной санитарии и противопожарной защиты.

Проведено системотехническое обслуживание компьютерных систем и комплексов.

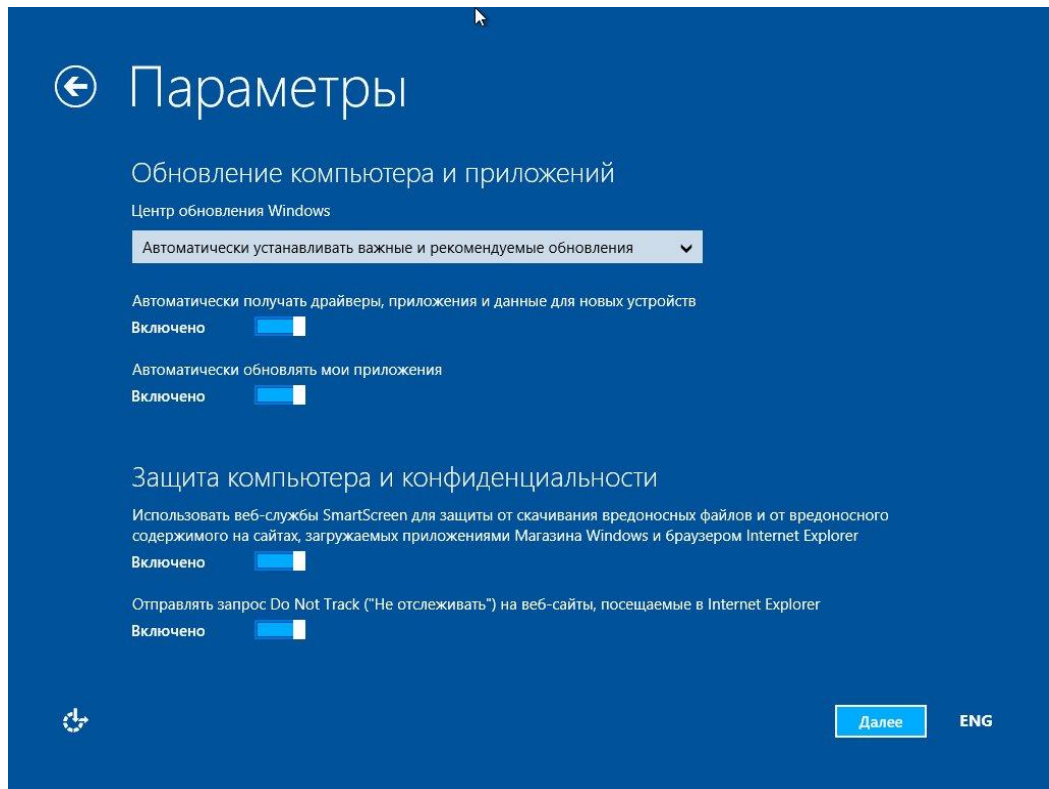


Рисунок 21. Конфигурирование и настройка ОС

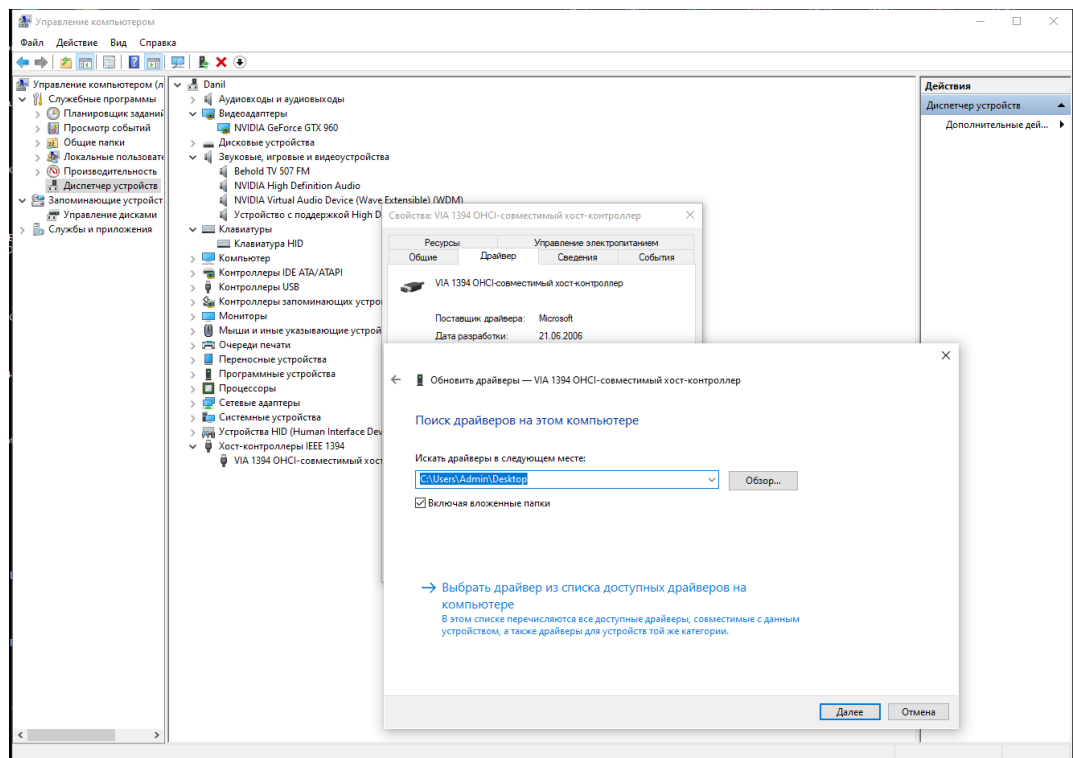


Рисунок 22. Установка драйверов

А также прошел обучение по разработке и программированию манипуляторов или мобильных роботов, по работе с беспаячной макетной платой, и по подключению электронных компонентов.



Рисунок 23. Изучение материала по теме разработки и программирования манипуляторов или мобильных роботов, работы с беспаячной макетной платой, и по подключению электронных компонентов

**Выполнение работ по
программному модулю 04
«Выполнение работ по
профессии 16199 Оператор
электронно-
вычислительных и
вычислительных машин»**

На данном этапе прохождения практики были выполнены работы по профессии 16199 Оператор электронно-вычислительных и вычислительных машин».

Была проведена работа по конвертированию файлов с цифровой информацией в различные форматы. По заданию руководителя практики от организации было выполнено конвертирование видеофайлов с помощью приложения «Adobe Media Encoder». Интерфейс программы представлен на рисунке 24. Исходные файлы содержат формат AVI (Audio Video Interleave), скорость передачи данных – 15 Мбит/сек; далее файлы были конвертированы в формат MP4 (MPEG4), скорость передачи данных – 9 Мбит/сек. Причиной конвертирования файлов из одного формата в другой является большой размер файла. После проделанной работы видеофайлы не потеряли качество и стали более лёгкими по сравнению с AVI форматом.

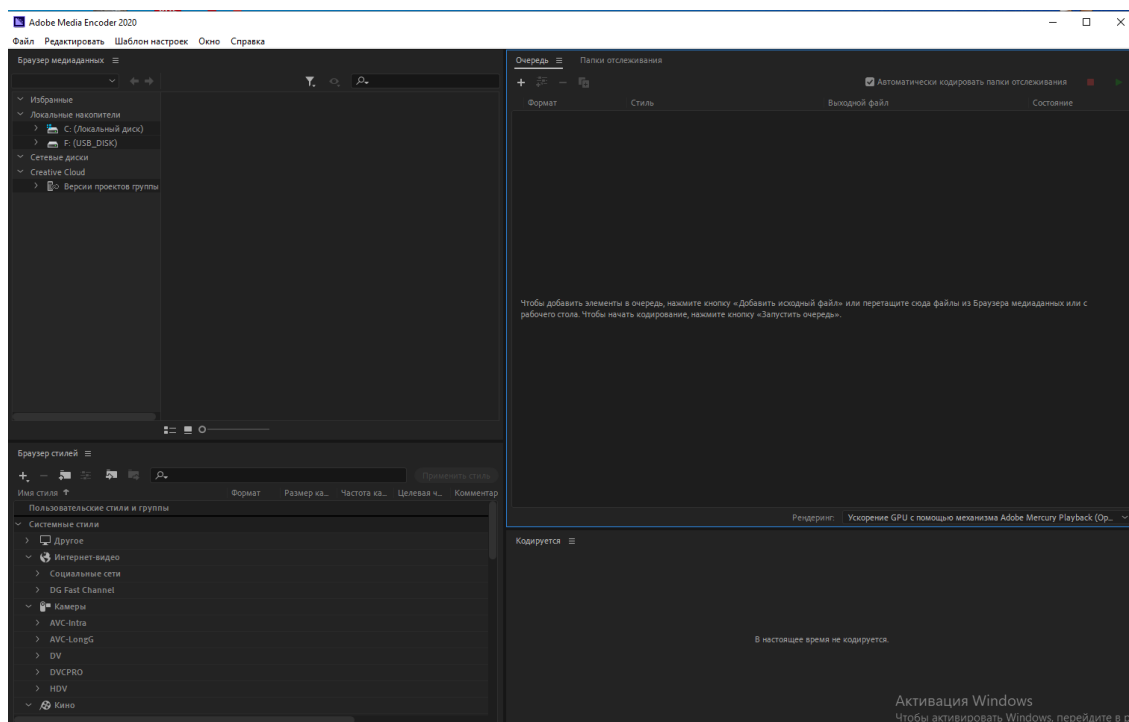
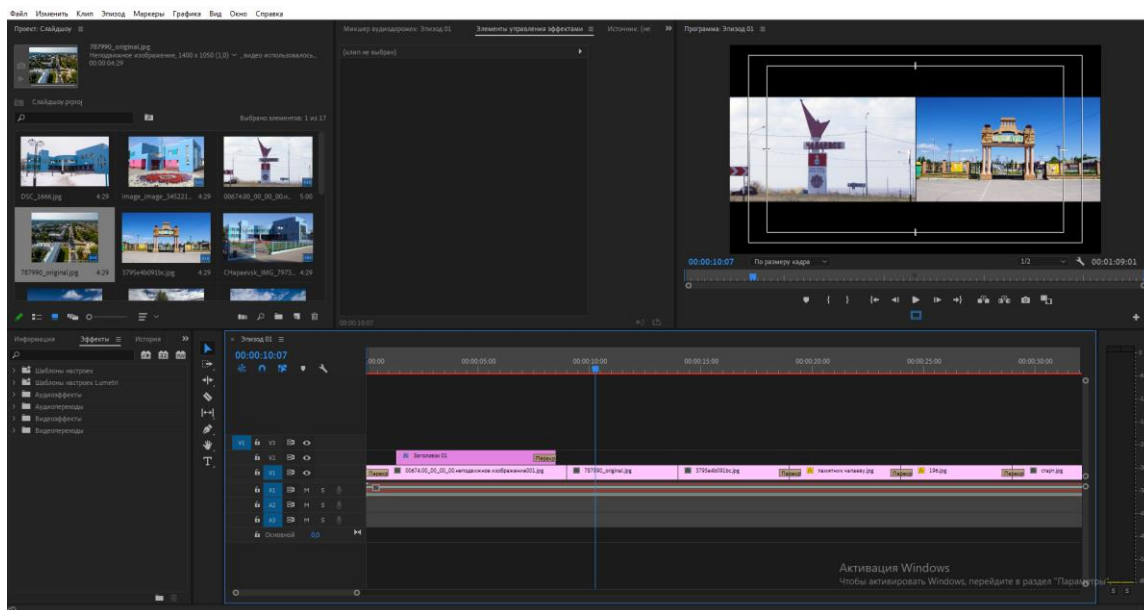


Рисунок 24. Интерфейс программы Adobe Media Encoder

Далее в соответствии с заданием было произведено создание видеоролика при помощи приложения «Adobe Premiere Pro» на тему «Чапаевск. Городские зарисовки». Процесс создания представлен на рисунке 25.



**Рисунок 25. Процесс создания видеоролика в приложении
Adobe Premiere Pro**

Далее готовый видеоролик был размещен на сервисе хранения данных «Яндекс.Диск».

Также было выполнено задание по обработке графических изображений средствами программ растровой и векторной графики. Была обработана фотография из семейного архива. На фотографии были обнаружены дефекты: линии, обрезы, неровные края. Средствами обработки изображений в программе Adobe Photoshop все вышеперечисленные дефекты были устранены.

Вначале фотография была отсканирована в высоком качестве при помощи сканера. Далее, фотография, переведенная в цифровой формат была полностью обработана инструментами программы: точечной восстанавливающей кистью, штампом, кривыми и уровнем. (Рисунок 27)

Результат обработки фотографии представлен на рисунке 28.

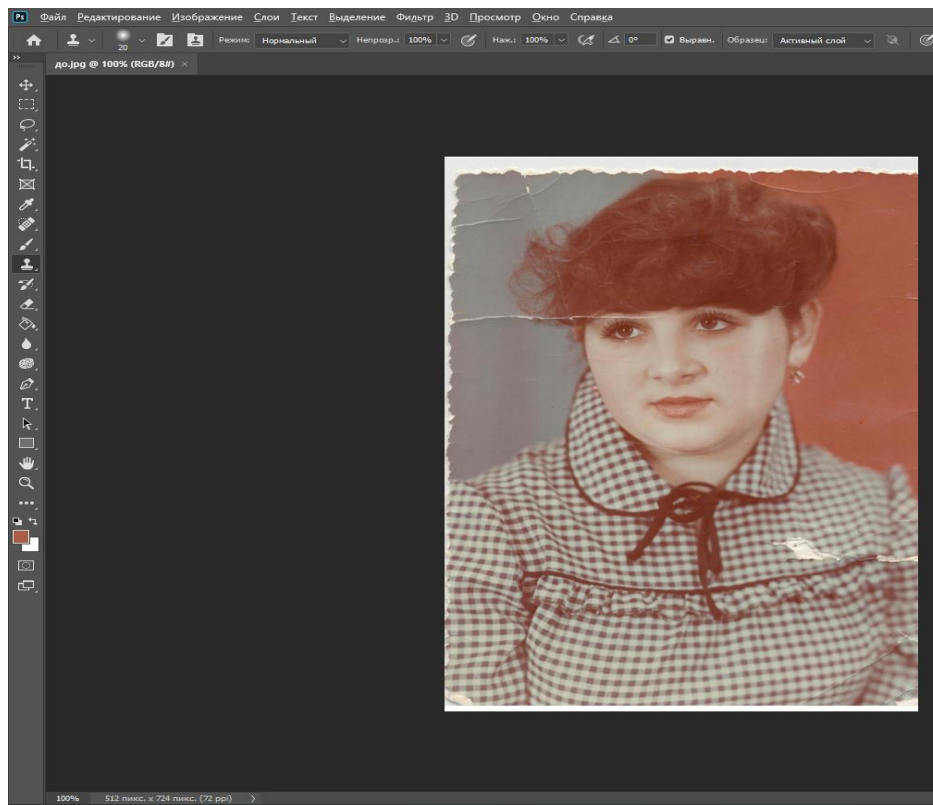


Рисунок 27. Оригинал фотографии в цифровом формате



Рисунок 28. Результат обработки фотографии

Завершающим заданием по данному программному модулю являлось создание анимационных объектов. Были разработаны электронные цифровые часы с помощью приложения Macromedia Flash.

Процесс создания состоял из 4 частей:

Создание рабочей области (создание фона (циферблата) часов), создание слоёв interface, sec, min, hour и actions

Работа с кадрами

Написание кода для кадров.

Создание дополнительных кадров

Результат работы анимационный объект «Часы» представлен на рисунке 29.

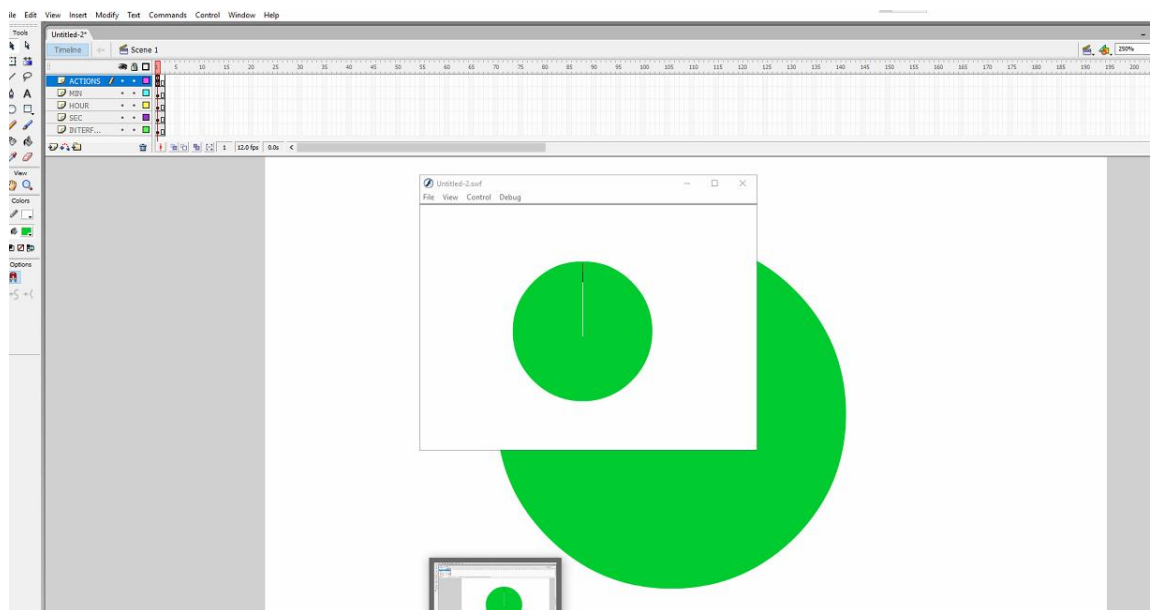


Рисунок 29. Созданный анимационный объект «Часы»

**Выполнение
индивидуального задания на
ВКР**

В соответствии с индивидуальным заданием, выданным на выполнение выпускной квалификационной работы в период прохождения преддипломной практики было выполнено следующее:

1. Проведен обзор литературы и Интернет – источников по теме выпускной квалификационной работы.
2. Изучены технологии разработки Web-приложений и использование баз данных в разработке Web-приложений.
3. Продолжена разработка Web-приложения «Электронная система расписаний занятий образовательной организации».

После проведения обзора литературы и Интернет – источников по теме выпускной квалификационной работы также был проведен обзор основных вопросов по теме разработки Web-приложения

При создании Web-приложений возникают следующие вопросы:

1. Какие языки программирования будут использоваться при разработке серверной и клиентской части?
2. Какую выбрать программную среду для разработки Web-приложения?
3. Какое программное обеспечение (HTTP-сервер, PHP-сервер) использовать в работе серверной части?
4. Какую систему управления базами данных использовать в работе?
5. Какую структуру будет иметь Web-приложение?
6. Какой пользовательский интерфейс будет иметь Web-приложение?

За всеми инструментами веб-разработки стоит язык. Язык программирования является формальным языком, предназначенным для построения связи с компьютером что бы программист мог удобно и максимально быстро создавать программы. При разработке данного проекта будут использоваться следующие популярные языки программирования:

HTML: язык гипертекстовой разметки

PHP: популярный язык сценариев общего назначения, который особенно подходит для веб-разработки.

Javascript: язык программирования HTML и веб.

HTML5: язык разметки, последняя версия HTML и XHTML.

CSS3: последняя версия каскадных таблиц стилей, используемых при фронтальной разработке сайтов и приложений.

SQL: обозначает язык структурированных запросов, используемый с реляционными базами данных.

Также была выбрана программная среда для разработки Web-приложения: ей является локальный веб-сервер OpenServer.

Локальный веб-сервер OpenServer это программная платформа для создания локальных сайтов на базе Windows.



Рисунок 30. Логотип веб-сервера OpenServer

Работа с OpenServer на своём компьютере позволяет создавать сайты без выхода в Интернет и без использования сторонних серверов. Многие веб-мастера именно так создают и тестируют веб-ресурсы. Это удобно. Вы экономите средства, не зависите от работы сторонних серверов, компьютерная помощь на дому решает возникающие технические проблемы, можно вести сколько угодно проектов и т.д.

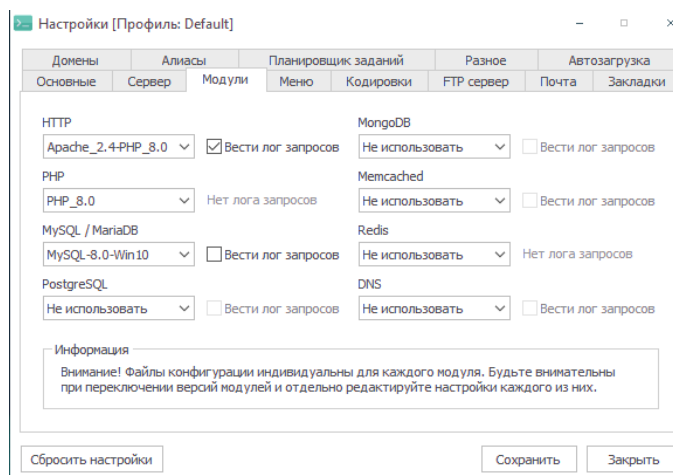


Рисунок 31. Панель управления веб-сервером OpenServer

Для функционирования серверной части (обработки запросов, функционирования модулей) на сегодняшний день используются два наиболее

популярные веб-сервера с открытым исходным кодом для работы в Интернете: HTTP-сервер Apache и NGINX.

Для выбора подходящего для разрабатываемого Web-приложения также приведены сравнительные характеристики.

Более 50% веб-сайтов в мире работают на этих двух веб-серверах. В течение почти двух десятилетий веб-сервер Apache обслуживал около 60 процентов веб-сайтов в мире, пока не появился его конкурент NGINX. В связи с резким ростом объемов трафика данных и количества пользователей всемирной паутины NGINX был создан для преодоления ограничений производительности веб-серверов Apache. NGINX, разработанный для обеспечения более высокого уровня параллелизма, может быть развернут как автономный веб-сервер и как внешний прокси-сервер для Apache и других веб-серверов.

Обзор сервера Apache



Рисунок 32. Логотип веб-сервера Apache

Apache был разработан для доставки веб-контента, доступ к которому осуществляется через Интернет. Он известен тем, что играл ключевую роль в начальном росте Всемирной паутины. Apache - это программное обеспечение с открытым исходным кодом, разработанное и поддерживаемое открытым сообществом разработчиков и работающее в самых разных операционных системах. Архитектура включает в себя ядро Apache и модули. Основной компонент предоставляет базовую серверную функцию, поэтому он принимает соединения и управляет параллелизмом. Различные модули соответствуют различным функциональным возможностям, которые выполняются для каждого запроса. Конкретное развертывание Apache может быть сконфигурировано для включения различных модулей, таких как функции

безопасности, управление динамическим контентом или для базовой обработки HTTP-запросов.

Модель «один сервер делает все» стала ключом к успеху Apache. Однако по мере увеличения уровней трафика и увеличения количества веб-страниц работа Apache стала усложняться.

Обзор NGINX



Рисунок 33. Логотип веб-сервера NGINX

NGINX был разработан для устранения ограничений производительности веб-серверов Apache. Производительность и масштабируемость NGINX обусловлены архитектурой управления событиями. Он значительно отличается от подхода Apache к процессу или потоку на соединение. В NGINX каждый рабочий процесс может одновременно обрабатывать тысячи HTTP соединений.

Следовательно, NGINX - это легковесная, масштабируемая и высокопроизводительная альтернатива. Архитектура NGINX делает обработку больших нагрузок на данные гораздо более предсказуемой с точки зрения использования ОЗУ, использования ЦП и задержки.

NGINX также имеет богатый набор функций и может выполнять различные роли сервера:

Обратный прокси-сервер для протоколов HTTP, HTTPS, SMTP, POP3 и IMAP

Балансировщик нагрузки и HTTP-кеш

Интерфейсный прокси для Apache и других веб-серверов, сочетающий гибкость Apache с хорошей производительностью статического содержимого NGINX.

Простота

Разрабатывать и обновлять приложения на Apache очень просто. Модель «одно соединение на процесс» позволяет очень легко вставлять модули в любой точке логики веб-обслуживания. Разработчики могут добавлять код

таким образом, что в случае сбоев будет затронут только рабочий процесс, выполняющий код. Обработка всех других соединений будет продолжаться без помех.

NGINX, с другой стороны, имеет сложную архитектуру, поэтому разработка модулей не легка. Разработчики модулей NGINX должны быть очень осторожны, чтобы создавать эффективный и точный код, без каких-либо сбоев, и соответствующим образом взаимодействовать со сложным ядром, управляемым событиями, чтобы избежать блокирования операций.

Производительность

Производительность измеряется тем, как сервер доставляет большие объемы контента в браузер клиента, и это важный фактор. Контент может быть статическим или динамическим.

Статический контент

NGINX работает в 2,5 раза быстрее, чем Apache, согласно тесту производительности, выполняемому до 1000 одновременных подключений. Другой тест с 512 одновременными подключениями показал, что NGINX примерно в два раза быстрее и потребляет меньше памяти. Несомненно, NGINX имеет преимущество перед Apache со статическим контентом. Поэтому, если вам нужно обслуживать одновременный статический контент, NGINX является предпочтительным выбором.

Динамический контент

Результаты тестов Speedemy показали, что производительность динамического контента была одинаковой для серверов Apache и NGINX. Вероятная причина этого заключается в том, что почти все время обработки запросов тратится в среде выполнения PHP, а не в основной части веб-сервера. Среда выполнения PHP довольно похожа для обоих веб-серверов.

Apache также может обрабатывать динамический контент, встраивая процессор языка, подобного PHP, в каждый из его рабочих экземпляров. Это позволяет ему выполнять динамический контент на самом веб-сервере без

необходимости полагаться на внешние компоненты. Эти динамические процессы могут быть включены с помощью динамически загружаемых модулей.

NGINX изначально не имеет возможности обрабатывать динамический контент. Для обработки PHP и других запросов на динамический контент NGINX должен перейти на внешний процессор и дождаться отправки отрендеренного контента. Однако этот метод также имеет некоторые преимущества. Поскольку динамический интерпретатор не встроен в рабочий процесс, его накладные расходы будут присутствовать только для динамического содержимого.

Поддержка ОС

Apache работает во всех операционных системах, таких как UNIX, Linux или BSD, и полностью поддерживает Microsoft Windows.

NGINX также работает на нескольких современных Unix-подобных системах и поддерживает Windows, но его производительность в Windows не так стабильна, как на платформах UNIX.

Безопасность

И Apache, и NGINX являются безопасными веб-серверами. Команда безопасности Apache существует, чтобы предоставлять помощь и советы проектам Apache по вопросам безопасности и координировать обработку уязвимостей безопасности. Важно правильно настроить серверы и знать, что делает каждый параметр в настройках. Существует множество рекомендаций по обеспечению безопасности серверов для предотвращения атак безопасности.

Гибкость

Веб-серверы могут быть настроены путем добавления модулей. Apache долгое время имел динамическую загрузку модулей, поэтому все модули Apache поддерживают это.

NGINX Plus (NGINX Plus - это программный балансировщик нагрузки, веб-сервера и кэш контента, построенный на основе открытого исходного кода NGINX) также использует модульную архитектуру. Новые функции и

возможности могут быть добавлены с программными модулями, которые по требованию могут быть подключены к работающему экземпляру NGINX Plus. Динамические модули добавляют в NGINX Plus такие функции, как геолокация пользователей по IP-адресу, изменение размеров изображений и встраивание сценариев Lua в модель обработки событий NGINX Plus. Модули создаются как самой NGINX, так и сторонними разработчиками.

Большинство необходимых функциональных возможностей основного модуля (например, прокси, кэширование, распределение нагрузки) поддерживается обоими веб-серверами.

Поддержка и документация

Важным моментом, который следует учитывать, является доступная справка и поддержка веб-серверов среди прочего программного обеспечения. Поскольку Apache был долгое время популярен, поддержка сервера довольно распространена повсеместно.

Наряду с документацией многие веб-проекты содержат инструменты для начальной загрузки в среде Apache. Оно может быть включено в сами проекты или в пакеты, поддерживаемые вашим дистрибутивом.

Apache, как правило, получает большую поддержку от сторонних проектов просто из-за своей доли рынка и продолжительности его доступности.

Apache

Простота. Легко разрабатывать и внедрять новые функции благодаря своей модели «одно соединение на весь процесс»

Производительность. Медленно в отображении статического контента, но быстрая в отображении динамического контента

Поддержка ОС. Поддерживает все ОС - Unix, в том числе и Windows

Гибкость. Можно настроить, добавив модули. Apache имел самую долгую динамическую загрузку модулей

Поддержка и документация. Отличная поддержка и документация, как это было на рынке в течение очень долгого времени.

Ngnix

Простота. Сложный в разработке, поскольку он имеет сложную архитектуру для одновременной обработки нескольких соединений.

Производительность. В 2,5 раза быстрее, чем Apache, и потребляет меньше памяти

Поддержка ОС. Поддерживает все ОС - как Unix, так и Windows, однако производительность в Windows сравнительно менее стабильна.

Гибкость. NGINX версии 1.11.5 и NGINX Plus Release R11 представили совместимость для динамических модулей.

Поддержка и документация. В начале у NGINX была слабая поддержка, но с ростом популярности поддержка и документация стала доступной.

Для надежного хранения данных приложения, выбор системы управления базами данных является одним из важных этапов при разработке. Наиболее правильный подход при выборе СУБД основан на оценке того, в какой мере существующие системы удовлетворяют основным требованиям создаваемого проекта информационной системы.

В данной работе хранение данных будет производится в реляционной базе данных. Реляционные базы данных используются уже очень давно. Они стали популярными благодаря успешным реализациям реляционных моделей в системах управления, оказавшимся весьма удобными для работы с данными. Далее приведены три самые популярные реляционные системы управления базами данных (РСУБД): SQLite, MySQL и PostgreSQL.

Реляционные системы реализуют реляционную модель работы с данными, которая определяет всю хранимую информацию как набор связанных записей и атрибутов в таблице.

СУБД такого типа используют структуры (таблицы) для хранения и работы с данными. Каждый столбец (атрибут) содержит свой тип информации. Каждая запись в базе данных, обладающая уникальным ключом, передаётся в строку таблицы, и её атрибуты отображаются в столбцах таблицы.

Отношения можно определить как математические множества, содержащие наборы атрибутов, отображающие хранящуюся информацию.

Каждый элемент, формирующий запись, должен удовлетворять определённому типу данных (целое число, дата и т.д.). Различные РСУБД используют разные типы данные, которые не всегда взаимозаменяемы.

Такого рода ограничения обычны для реляционных баз данных. Фактически, они и формируют суть отношений.

SQLite

SQLite — это библиотека, встраиваемая в приложение, которое её использует. Будучи файловой БД, она предоставляет отличный набор инструментов для более простой (в сравнении с серверными БД) обработки любых видов данных.



Рисунок 34. Логотип СУБД SQLite

Когда приложение использует SQLite, их связь производится с помощью функциональных и прямых вызовов файлов, содержащих данные (например, баз данных SQLite), а не какого-то интерфейса, что повышает скорость и производительность операций.

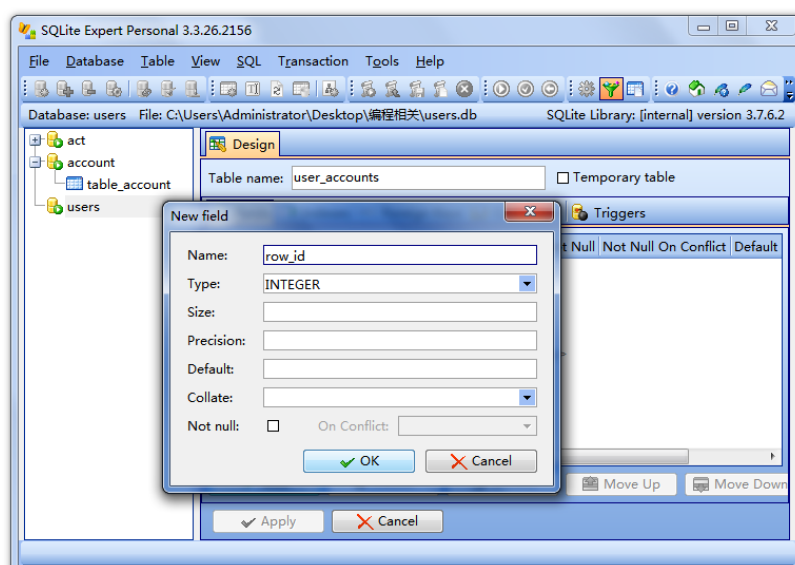


Рисунок 35. Консоль управления СУБД SQLite

Преимущества:

Файловая: вся база данных хранится в одном файле, что облегчает перемещение.

Стандартизированная: SQLite использует SQL; некоторые функции опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), однако, есть и некоторые новые.

Отлично подходит для разработки и даже тестирования: во время этапа разработки большинству требуется масштабируемое решение. SQLite, со своим богатым набором функций, может предоставить более чем достаточный функционал, при этом будучи достаточно простой для работы с одним файлом и связанной библиотекой.

Недостатки:

Отсутствие пользовательского управления: продвинутые БД предоставляют пользователям возможность управлять связями в таблицах в соответствии с привилегиями, но у SQLite такой функции нет.

Невозможность дополнительной настройки: SQLite нельзя сделать более производительной.

Когда стоит использовать SQLite:

Встроенные приложения: все портируемые не предназначенные для масштабирования приложения — например, локальные однопользовательские приложения, мобильные приложения или игры.

Система доступа к дисковой памяти: в большинстве случаев приложения, часто производящие прямые операции чтения/записи на диск, можно перевести на SQLite для повышения производительности.

Тестирование: отлично подойдёт для большинства приложений, частью функционала которых является тестирование бизнес-логики.

Когда не стоит использовать SQLite:

Многопользовательские приложения: если вы работаете над приложением, доступом к БД в котором будут одновременно пользоваться

несколько человек, лучше выбрать полнофункциональную РСУБД — например, MySQL.

Приложения, записывающие большие объёмы данных: одним из ограничений SQLite являются операции записи. Эта СУБД допускает единовременное исполнение лишь одной операции записи.

MySQL

MySQL — это самая популярная из всех крупных серверных БД. Разобраться в ней очень просто, да и в сети о ней можно найти большое количество информации. Хотя MySQL и не пытается полностью реализовать SQL-стандарты, она предлагает широкий функционал. Приложения общаются с базой данных через процесс-демон.



Рисунок 36. Логотип СУБД MySQL

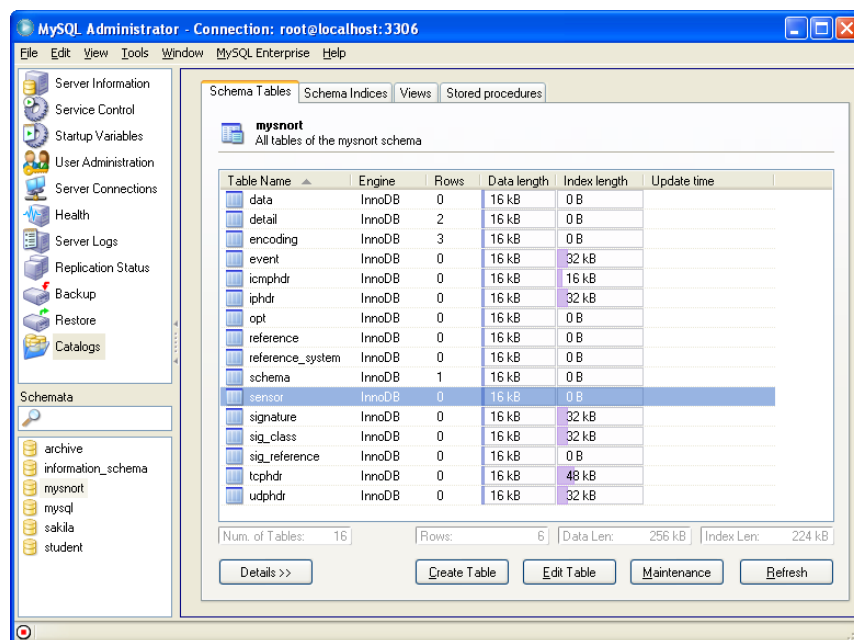


Рисунок 37. Консоль администрирования СУБД MySQL

Преимущества:

Простота: MySQL легко устанавливается. Существует много сторонних инструментов, включая визуальные, облегчающих начало работы с БД.

Много функций: MySQL поддерживает большую часть функционала SQL.

Безопасность: в MySQL встроено много функций безопасности.

Мощность и масштабируемость: MySQL может работать с действительно большими объёмами данных, и неплохо подходит для масштабируемых приложений.

Скорость: пренебрежение некоторыми стандартами позволяет MySQL работать производительнее.

Недостатки

Известные ограничения: по определению, MySQL не может сделать всё, что угодно, и в ней присутствуют определённые ограничения функциональности.

Вопросы надёжности: некоторые операции реализованы менее надёжно, чем в других РСУБД.

Застой в разработке: хотя MySQL и является open-source продуктом, работа над ней сильно заторможена. Тем не менее, существует несколько БД, полностью основанных на MySQL (например, MariaDB).

Когда стоит использовать MySQL:

Распределённые операции: когда вам нужен функционал больший, чем может предоставить SQLite, стоит использовать MySQL.

Высокая безопасность: функции безопасности MySQL предоставляют надёжную защиту доступа и использования данных.

Веб-сайты и приложения: большая часть веб-ресурсов вполне может работать с MySQL, несмотря на ограничения. Этот инструмент весьма гибок и прост в обращении, что только на руку в длительной перспективе.

Кастомные решения: если вы работаете над очень специфичным продуктом, MySQL подстроится под ваши потребности благодаря широкому спектру настроек и режимов работы.

Когда не стоит использовать MySQL:

SQL-совместимость: поскольку MySQL не пытается полностью реализовать стандарты SQL, она не является полностью совместимой с SQL. Из-за этого могут возникнуть проблемы при интеграции с другими РСУБД.

Конкурентность: хотя MySQL неплохо справляется с операциями чтения, одновременные операции чтения-записи могут вызвать проблемы.

Недостаток функций: в зависимости от выбора движка MySQL может не предоставлять некоторых функций.

PostgreSQL

PostgreSQL — это самая продвинутая РСУБД, ориентирующаяся в первую очередь на полное соответствие стандартам и расширяемость. PostgreSQL, или Postgres, пытается полностью соответствовать SQL-стандартам ANSI/ISO.



Рисунок 38. Логотип СУБД PostgreSQL

PostgreSQL отличается от других РСУБД тем, что обладает объектно-ориентированным функционалом, в том числе полной поддержкой концепта ACID (Atomicity, Consistency, Isolation, Durability).

Будучи основанным на мощной технологии Postgres отлично справляется с одновременной обработкой нескольких заданий. Поддержка конкурентности реализована с использованием MVCC (Multiversion Concurrency Control), что также обеспечивает совместимость с ACID.

Хотя эта РСУБД не так популярна, как MySQL, существует много сторонних инструментов и библиотек для облегчения работы с PostgreSQL.

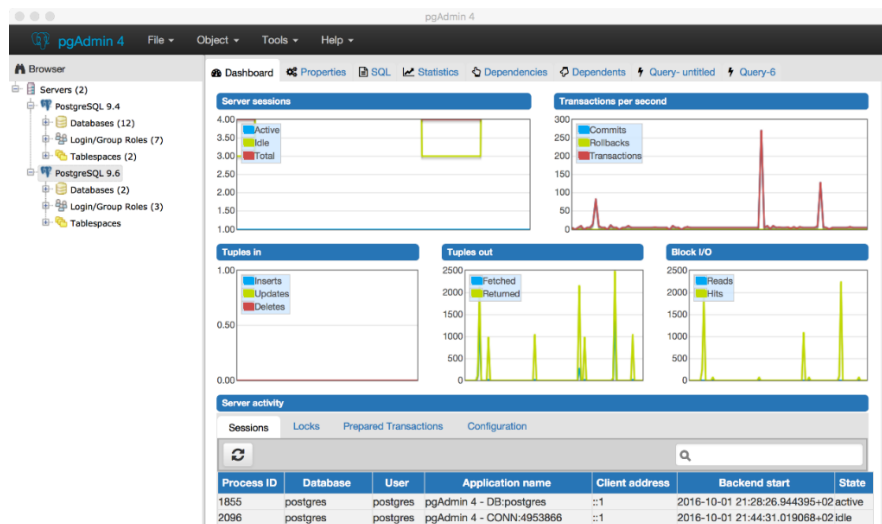


Рисунок 39. Консоль администрирования СУБД PostgreSQL

Преимущества:

Полная SQL-совместимость.

Поддержка сторонними организациями: несмотря на очень продвинутые функции, PostgreSQL используется в многих инструментах, связанных с РСУБД.

Расширяемость: PostgreSQL можно программно расширить за счёт хранимых процедур.

Объектно-ориентированность: PostgreSQL — не только реляционная, но и объектно-ориентированная СУБД.

Недостатки:

Производительность: В простых операциях чтения PostgreSQL может уступать своим соперникам.

Популярность: из-за своей сложности инструмент не очень популярен.

Хостинг: из-за вышеперечисленных факторов проблематично найти подходящего провайдера.

Когда стоит использовать PostgreSQL:

Целостность данных: если приоритет стоит на надёжность и целостность данных, PostgreSQL — лучший выбор.

Сложные процедуры: если ваша БД должна выполнять сложные процедуры, стоит выбрать PostgreSQL в силу её расширяемости.

Интеграция: если в будущем вам предстоит перемещать всю базу на другое решение, меньше всего проблем возникнет с PostgreSQL.

Когда не стоит использовать PostgreSQL:

Скорость: если всё, что нужно — это быстрые операции чтения, не стоит использовать PostgreSQL.

Сравнивая преимущества и недостатки вышеперечисленных систем управления базами данных, для разрабатываемого проекта Web-приложения была выбрана СУБД – MySQL.

После определения языков программирования и подбора необходимого в работе серверной и клиентской части приложения была разработана структурная схема Web-приложения которая представлена на рисунке 40.

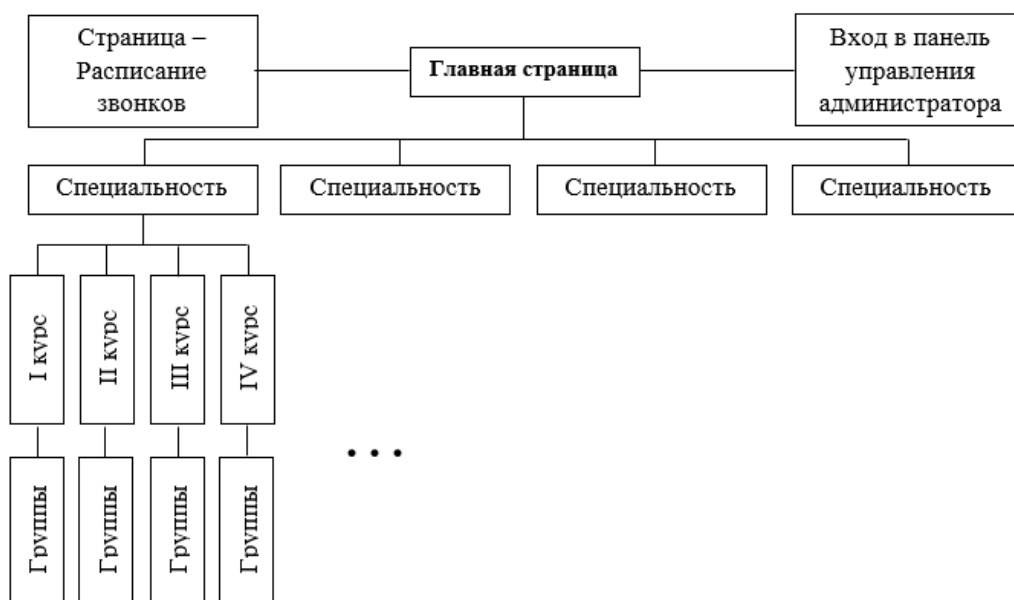


Рисунок 40. Структурная схема Web-приложения

Также для клиентской части (стороны пользователя) разработан дизайн-макет приложения. Макет в процессе разработки помогает - заранее увидеть, как будут выглядеть продукт, включая мобильную версию сайта и элементы анимации; разработать единый дизайн для всех страниц сайта и всех его типовых элементов, а также продумать наполнение элементов — текста и изображений.

Макет разрабатываемого Web-приложения состоит из блоков «Логотип», «Заголовок», «Меню», «Подменю», «Поле ввода», «Контактная информация». Макет представлен на рисунке 41.

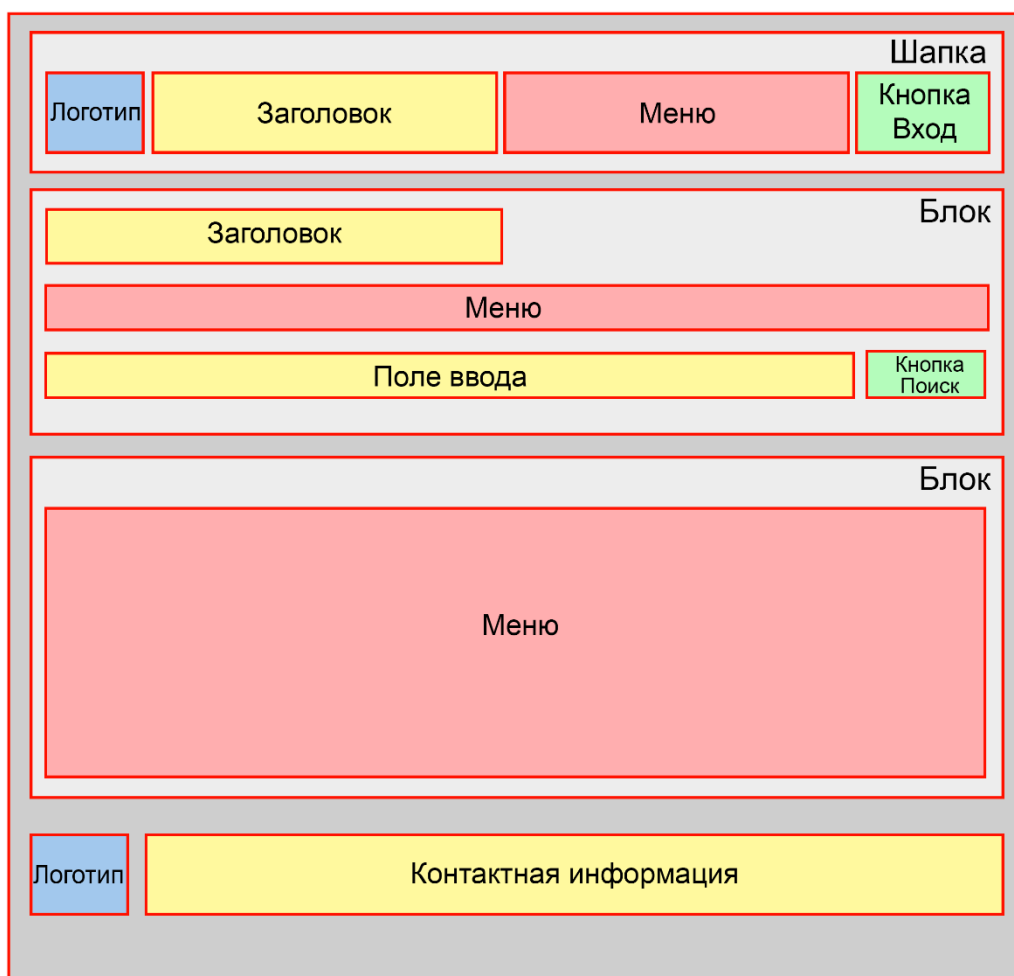


Рисунок 41. Макет Web-приложения «Электронная система расписания занятий образовательной организации»

Вывод

На базе ООО «Фотон» была пройдена преддипломная практика.

Сроки практики: с 20.04.2022г. по 17.05.2022г.

Целью практики являлось закрепление, расширение, углубление и систематизация знаний, полученных в ходе обучения по специальности 09.02.01 Компьютерные системы и комплексы, приобретение и закрепление практического опыта, овладение видами профессиональной деятельности, такими как: проектирование цифровых устройств, применение микропроцессорных систем, установка и настройка периферийного оборудования, техническое обслуживание и ремонт компьютерных систем и комплексов, а также по выполнению работ по профессии 16199 Оператор электронно-вычислительных и вычислительных машин.

На вводном занятии было произведено ознакомление с инструктажем по техники безопасности и производственной санитарии, с организацией и получением индивидуальных заданий.

Инструктаж провел руководитель практики от организации: Аксёнов Андрей Рустямович.

В ходе практики была изучена история создания и развития организации, организационная структура управления.

Были выполнены все задания по программным модулям, согласно заданию по преддипломной практике.

Также было выполнено индивидуальное задание по написанию дипломной работы на тему «Разработка электронной системы расписания занятий образовательной организации».

При выполнении заданий и подготовке отчета использовались различные источники – справочная информация, интернет-ресурсы, информация от руководителей практики, поисковой браузер, и свои познания в некоторых заданиях.

Цель и задачи практики выполнены полностью.