**Автоматизированная система работы поликлиники**

Диплом

**СОДЕРЖАНИЕ**

[Введение 5](#_Toc104843234)

[1. Аналитическая часть 7](#_Toc104843235)

[1.1 Виды и характеристика веб-сайтов, роль в работе поликлиник 7](#_Toc104843236)

[1.1.2 Роль веб-сайта в работе медицинской клиники 8](#_Toc104843237)

[1.2 Обоснование выбора средств создания веб-сайта 9](#_Toc104843238)

[1.3 Обзор систем управления базами данных 15](#_Toc104843239)

[2.Технологическая часть 18](#_Toc104843240)

[2.1 Постановка задачи 18](#_Toc104843241)

[2.1.1 Актуальность 18](#_Toc104843242)

[2.1.2 Назначение веб-сайта 18](#_Toc104843243)

[2.1.3 Характеристики оборудования 19](#_Toc104843244)

[2.2 Реализация проекта 19](#_Toc104843245)

[2.3 Средства для разработки и проектирование сайта 20](#_Toc104843246)

[2.3.1 Анализ средств для разработки и создания Web-сайта 20](#_Toc104843247)

[2.3.2 Фреймворки, используемые при разработке Web-сайтов: сравнительный анализ 26](#_Toc104843248)

[2.3.4 Разработка веб-сайта в CMS WordPress 34](#_Toc104843249)

[3. Тестирование веб-сайта и руководство пользователя 43](#_Toc104843250)

[3.1. Тестирование веб-сайта 43](#_Toc104843251)

[3.2. Руководство пользователя 45](#_Toc104843252)

[Заключение 49](#_Toc104843253)

[Список использованной литературы 50](#_Toc104843254)

[Приложение 52](#_Toc104843255)

[Написание на заказ курсовых, дипломов, диссертаций...](http://учебники.информ2000.рф/napisat-diplom.shtml)

# **Введение**

С каждым днем ​​все больше и больше людей осваивают возможности Интернета для поиска информации о тех или иных услугах, Интернет стал необходимостью для человека. Путешествуя по сети, вы сталкиваетесь с большим количеством рекламы всех видов. Часто встречаются интересные сервисы, информация и ресурсы.

На современном этапе развития глобальной сети Интернет создание веб-сайтов является актуальной задачей. Создание сайтов для компаний и организаций популярно, и эта тенденция сохранится еще долгое время. В связи с этим большинство современных компаний, учреждений, организаций имеют собственные сайты в сети Интернет, на которых раскрывается основная информация об организации, сведения о характере ее деятельности, услугах и контактные данные.

**Вернуться в каталог готовых дипломов и магистерских диссертаций –**

[**http://учебники.информ2000.рф/diplom.shtml**](http://учебники.информ2000.рф/diplom.shtml)

Качественный сайт, вобравший в себя всю полезную информацию и являющийся лучшей визитной карточкой для деловых компаний и медицинских клиник – это прекрасная возможность познакомить клиентов и пациентов со своей клиникой, ее особенностями и методами работы, а также способ поделиться результатами своей деятельности, создать информационное пространство для пациентов, клиентов и тех, кто будет посещать клинику, но еще ничего не решено.

Грамотно созданный сайт будет способствовать его дальнейшему развитию и продвижению, а также повысит эффективность взаимодействия основного корпуса медицинской клиники (пациентов и клиентов). Сайт – это визитная карточка медицинской клиники, неотъемлемая часть медицинской отрасли, используемая для взаимодействия с людьми и создания новых возможностей для взаимодействия с клиентами. Сайт медицинской клиники – это инструмент коммуникации менеджеров, пациентов и клиентов. Объект исследования дипломной работы — деятельность поликлиники.

Предмет исследования выпускной квалификационной работы — процесс создания веб-сайта поликлиники с помощью системы управления контентом WordPress.

Цель дипломной работы — разработать веб-сайт поликлиники.

В соответствии с поставленной целью в работе определены следующие задачи:

• провести анализ предметной области;

• проанализировать средства разработки веб-сайта;

• разработать модель веб-сайта;

• реализовать разработанную модель веб-сайта средствами веб-технологий;

• наполнить веб-сайт контентом.

# **1.** **Аналитическая часть**

# **Виды и характеристика веб-сайтов, роль в работе поликлиник**

Сайты поликлиник относятся к группе медицинских сайтов. Под медицинским веб-сайтом подразумевается объединение веб-страниц с повторяющимся дизайном, несущих в себе целенаправленную информацию о мед учреждении и другую информацию.

**1.1.1 Виды медицинских веб-сайтов**

Веб-сайт медицинской клиники в зависимости от своего содержания может относиться к разным видам. Это относится к многофункциональным веб-сайтам. Сайты для медицинских компаний представляют собой целую автоматизированную систему, предоставляющую полный функционал для сбора и обработки данных. По функциональному назначению, их можно разделить на следующие виды:

1. Сайты лечебных заведений. Сайты медицинских учреждений, в том числе больницы, диагностические центры, госпитали, диспансеры, женские консультации.
2. Сайты научных исследований. Сайты исследовательских работ в области медицины, врачей, преподавателей медицинских ВУЗов, научных работников.
3. Сайты справочного характера. Сайты, содержащие справочную информацию о проводимых конференциях, конкурсах, семинарах научного и медицинского направления, о предоставляемых грантах, статьи, ссылки.
   1. Сайты информационных Интернет-проектов. Сайты обществ и ассоциаций, предназначены для проведения медицинских конкурсов и предоставление информационно-развлекательных проектов медицинской тематики.
   2. Сайты типа виртуальных методических объединений. Сайты методического объединения врачей по специализациям, методам лечения, сайты для тематических врачебных конференций и чатов в online режиме по вопросам лечения пациентов.
   3. Сайты консультативного назначения. Консультативные сайты для врачей и пациентов, интернов и профессоров, предназначены для обмена информацией и ответов на возникающие вопросы.

# **1.1.2 Роль веб-сайта в работе медицинской клиники**

Веб-сайт, представляющий организацию, называется корпоративным или официальным веб-сайтом организации. Официальный сайт предъявляет более строгие требования к информативности, графическому оформлению и навигации.

Официальный сайт обычно имеет следующие разделы:

* новостная информация;
* нормативные документы, положения;
* направления деятельности;
* структура учреждения;
* кадровый состав;
* контактная информация (список ответственных лиц, их должности, координаты и часы приема).

Сайты медицинских клиник играют важную роль в организации медицинской деятельности, информационной поддержке врачей и пациентов, помощи коллегам в профильных медицинских учреждениях.

Для врачей сайт является площадкой для обмена опытом, они могут общаться с коллегами из местных и даже стран, участвовать в совместных проектах в рамках отечественных и зарубежных медицинских проектов, обмениваться медицинским опытом.

Для пациентов сайт является в первую очередь информационным ресурсом. Используя сайты клиник и других медицинских учреждений, пациенты имеют возможность сравнить их и сделать оптимальный выбор для себя, своих детей и близких.

# **1.2 Обоснование выбора средств создания веб-сайта**

Система управления контентом — это компьютерная программа, предназначенная для упрощения управления контентом веб-сайта. Эта система позволяет веб-мастерам не использовать программный код напрямую, а использовать админ-панель со всеми необходимыми функциями для редактирования контента и создания новых материалов.

Наиболее популярные и распространенные системы управления контентом:

Joomla — вторая по популярности Content Management System (CMS),

более гибкая и простая. Joomla прекрасно подходит для разработки веб-сайта

с нестандартной структурой, например, корпоративного портала.

Данная система управления контентом производная ветвь от широко

известного проекта Mambo. В системе Joomla самые разнообразные инструменты для создания сайта.

Проблема перегрузки админки бесполезными элементами устранена благодаря важной особенности Joomla - минимальному набору инструментов при первоначальной установке. В результате снижается нагрузка на сервер и экономится место на хосте.

Обычные инструменты администрирования устанавливают каталог расширений, который включает ряд языковых пакетов. Это дает возможность перевести интерфейс фронтенда и админки практически на любой язык. К минусам можно отнести то, что это, так сказать, упрощенная CMS, поэтому она слаба функционалом. Многие плагины и дополнения для CMS Joomla платные. В целом Joomla отличная платформа, которая обладает простым интерфейсом, приятным дизайном и большим набором шаблонов.

MODx — это бесплатная профессиональная система управления содержимым и фреймворк для веб-приложений, предназначенная для обеспечения и организации совместного процесса создания, редактирования и управления контентом (то есть содержимым) веб-сайтов.

Преимущества:

• поддержка ленты новостей, веб-статистики, подписок;

• платформа дает возможность осуществлять контроль над всеми элементами страницы (от платежной системы до документооборота и чата).

При хороших навыках владения языком программирования можно легко «переписать» администрирование под себя.

Возможности:

• создание разнообразных сайтов на стандартах, адаптирующихся под различные браузеры;

• настройка ключевых слов, дружественных Uniform Resource Locator(URL).

Недостатки:

• маленькое количество готовых шаблонов, но разработчики обещают увеличить коллекцию;

• чтобы восстанавливать удаленные объекты, необходимо установить специальные плагины.

Drupal — это система управления веб-сайтом, написанная на языке PHP и использующая в качестве хранилища содержания реляционную базу данных (поддерживаются MySQL, PostgreSQL, а также любые базы данных, поддерживаемые библиотекой PEAR). За счет открытого кода доступа, она удобна в использовании, настройке. Drupal является свободным программным обеспечением, защищенным лицензией GPL и создается усилиями энтузиастов со всего мира. Начал разработку голландец DriesBuytaert, который и поныне является руководителем проекта. Архитектура Drupal позволяет применять его для построения различных типов веб-сайтов — от блогов и форумов, до информационных архивов или веб-сайтов новостей. Функциональность обеспечивается подключаемыми модулями.

1С-Битрикс — автоматизированная система управления контентом, созданная компанией 1С-Битрикс. В настоящее время «1С-Битрикс» лидирует в рейтинге систем управления сайтами платного тиража России. Продукт предназначен для использования при разработке и развитии корпоративных проектов, информационных, новостных и справочных порталов, социальных сетей, интернет-магазинов и других видов сайтов для предприятий и организаций.

Недостатки:

• функциональная избыточность приводит к большому количеству неиспользуемого программного кода, что снижает эффективность;

• Высокие требования к управляемым ресурсам;

• При выходе из стандартного функционала сайта требуется глубокий анализ сложного кода системы;

• Из-за большого количества пользователей системы влияние сетевых студий на развитие системы стремится к нулю;

• Клиенты часто рассматривают покупку Битрикс вместо полной поддержки сайта, что снижает их лояльность;

• Многочисленные встроенные функции затрудняют обучение клиентов и вызывают проблемы с сайтом;

• Разработка сайта создается на основе любого готового шаблона, вылезает большое количество неиспользуемых элементов и программных модулей, что приводит к низкой скорости загрузки сайта;

• Для некоторых клиентов коммерческие лицензии неприемлемы.

• Невозможно взять на себя полную ответственность за конечный продукт: система управления разработана другой компанией, размер кода веб-студии намного меньше;

• Интеграция с «1С:Управление предприятием» обычно требует существенной доработки;

• Система архитектурно неотделима от площадки. Администрирование выполняется на рабочих экземплярах сайта.

Кратко подытожим: у системы 1С-Битрикс достаточно недостатков, чтобы вызывать только негативные отзывы разработчиков и пользователей системы. NetCat — это профессиональная система управления контентом. NetCat — одна из ведущих систем управления контентом на российском рынке. Первая версия системы была создана в 1999 году. Согласно исследованию российского рынка CMS, проведенному интернет-изданием Webinform, система NetCat является самым продаваемым инструментом управления сайтами общего назначения в России.

WordPress — это самая популярная и известная платформа. Эта платформа изначально разрабатывалась как блог-платформа, но благодаря многочисленным дополнениям часто используется для разработки и других типов веб-сайтов.

Главным преимуществом WordPress является простота и понятность: все действия можно проводить через админ-панель, не обязательно хорошо знать верстку и программирование. WordPress обладает большим набором шаблонов, тем и модулей, тем самым, создание веб-сайта становиться легче.

Функционал настолько велик, что могут возникнуть сложности с выбором тех или иных возможностей системы. Процесс установки WordPress довольно прост, поэтому для не опытного разработчика это может играть важную роль в выборе систем управления контентом.

Описание основных преимуществ системы управления контентом WordPress:

• легкость в настройке и управлении консолью. Понятно и интуитивно доходчиво, не составит большого труда разобраться с программой имея

минимальные навыки либо придерживаясь понятным урокам по WordPress;

• предоставленных возможностей разработчиками достаточно, чтобы разрабатывать сайты практически любой сложности, создавать и редактировать шаблоны;

• обширные возможности с целью оптимизации как отдельных компонентов, так и целого сайта;

• достаточно развитая система безопасности;

• плагины для расширения возможностей системы управлением контентомдают возможность настроить и представить контент более качественным и удобным методом;

• часто обновляемая система управления показатель постоянной деятельности разработчиков над усовершенствованием системы;

• новости о WordPress выходят достаточно часто, и это подтверждение тому, что система является действительно популярной.

На рисунке 1 изображено сравнение CMS систем.



Рисунок 1- Сравнение CMS систем

Таким образом, это будет идеальным вариантом для блогов, хоть и содержится много других инструментов для разработки веб-сайтов. Широкий и

простой в управлении функционал подойдет как новичку, так и опытному

веб-мастеру.

Проанализировав наиболее популярные системы управления контентом, была выбрана система WordPress. Такой выбор обусловлен тем, что среди всех имеющихся и более популярных, WordPress имеет продуманный механизм управления данными, он дает возможность администратору сайта с легкостью добавлять, редактировать и удалять информацию, позволяя не тратить много времени на операции. Для WordPress имеется много бесплатных модулей и плагинов, которые дают возможность увеличить функциональность базовой версии данной системы. Кроме того, система управления контентом WordPress является самой популярной и востребованной среди остальных систем управления. В этом можно убедиться, воспользовавшись сервисом GoogleTrends. Результаты рейтинга за март 2021 года представлен на рисунке 2. [1]

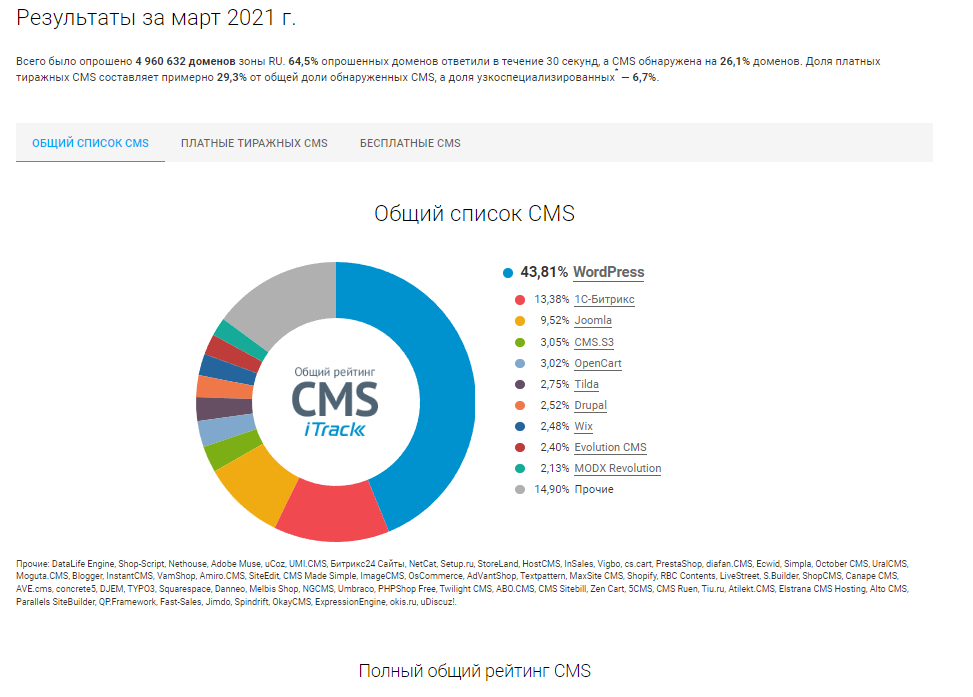


Рисунок 2 - Исследование популярности CMS за 2021 год

Популярность другого распространенного движка Joomla уменьшается, а желающих воспользоваться WordPress меньше не становится. Для тестирования и отладки используется хостинг jino.ru, с поддержкой PHP и СУБД MySQL. Что даёт возможность заказчику контролировать процесс разработки в режиме online. Внося свои корректировки и поправки.

# **1.3 Обзор систем управления базами данных**

Система управления базами данных (СУБД) — набор программного обеспечения общего или специального назначения и языковых инструментов для управления созданием и использованием баз данных.

* Основные возможности СУБД:
* Управление данными на диске;
* Управлять данными в памяти;
* Запись изменений, резервное копирование и восстановление БД после сбоя;
* Поддержка языка базы данных.
* СУБД условно можно разделить на 2 части:
* Открытый исходный код (MySQL)
* Коммерческий (Oracle, MS SQL) [5].

База данных Oracle. Система управления базами данных Oracle популярна благодаря своей надежности и безопасности. Разработанная для предприятий любого размера, Oracle Database предлагает уникальные возможности автоматической настройки и самодиагностики, которые выводят корпоративные системы на новый уровень функциональности и надежности.

Преимущества использования Oracle:

- Хранить пространственные данные в открытом стандартизированном формате;

- Данные хранятся в одной СУБД;

- Стандартизированный доступ к данным через SQL;

- Обеспечение совместной работы с данными и совместного доступа на уровне СУБД;

- Надежность, производительность;

- Расширяемость;

- Техническая поддержка и постоянное совершенствование программного обеспечения [6].

СУБД MySQL является наиболее приспособленной, для применения в среде web, системой управления базами данных. Не секрет, что для исполнения приложений клиента на большинстве хостинг-площадок провайдеры предоставляют небольшое количество ресурсов. Поэтому для данного применения необходима высокоэффективная СУБД, обладающая высокой надежностью.

Основные преимущества MySQL:

* многопоточность;
* оптимизация связей;
* записи фиксированной и переменной длины;
* ODBC драйвер;
* гибкая система привилегий и паролей;
* гибкая поддержка форматов чисел, строк переменной длины и меток времени;
* быстрая работа;
* масштабируемость;
* совместимость с ANSI SQL;
* хорошая поддержка со стороны хостинг-провайдеров;
* быстрая поддержка транзакций [7].

Microsoft SQL Server — это система управления реляционными базами данных, разработанная Microsoft. Поддерживается несколько языков SQL. Он используется для небольших и средних баз данных, для крупных корпоративных баз данных и для конкуренции с другими СУБД. Одним из основных преимуществ SQL Server являются расширенные графические и визуальные средства управления, благодаря которым вы можете значительно снизить административную нагрузку при обслуживании.

Имеет высокую производительность с удобным и простым пользовательским интерфейсом.

Основные преимущества SQL Server :масштабируемость; превосходная производительность; простота использования; готовность к использованию в Интернете; хранилища данных; интеграция с другими продуктами Microsoft [8].

Основные преимущества рассмотренных СУБД представлены в таблице 1.

Таблица 1. – Преимущества СУБД

|  |  |
| --- | --- |
| Название | Преимущества |
| Oracle Database | хранение пространственных данных; стандартизованный, доступ к данным; совместная работа с данными и разделение доступа;  надежность, производительность и масштабируемость;  техническая поддержка совершенствование ПО. |
| СУБД MySQL | многопоточность; оптимизация связей; записи фиксированной длины; ODBC драйвер; система привилегий и паролей; быстрая работа, масштабируемость; совместимость с ANSI SQL; поддержка со стороны провайдеров услуг хостинга; поддержка транзакций. |
| Microsoft SQL Server | масштабируемость; превосходная производительность;  простота использования;  готовность к использованию в Интернете;  хранилища данных;  интеграция с другими продуктами Microsoft. |

# **2.Технологическая часть**

# **2.1 Постановка задачи**

# **2.1.1 Актуальность**

Актуальность создания сайта для поликлиники заключается в том, что он позволяет лучше контролировать деятельность клиники и дает возможность дистанционно записаться к нужному врачу и управлять данными в личном кабинете, ведь у поликлиники на данный момент нет сайта и сайт будет автоматизированной системой сбора и управления информации.

Сайт поликлиники раскрывает основную информацию о клинике, информирует о характере ее деятельности и контактной информации, является визитной карточкой медицинской клиники. Создание сайта повышает авторитет поликлиники.

Для поликлиник - это уникальная возможность продемонстрировать свои услуги всем, распространяя важную информацию среди заинтересованных лиц (врачей, пациентов, клиентов, коллег в других медицинских учреждениях).

Это отличный способ поделиться результатами своей кампании и создать информационное пространство для врачей, пациентов и клиентов. На веб-сайте можно оставлять отзывы про поликлинику и врачей благодаря которым новые посетители могут ознакомится с качеством услуг.

# **2.1.2 Назначение веб-сайта**

С помощью веб-сайта можно ознакомиться со списком врачей, о самой поликлинике и её местоположению.

Назначением сайта является информировние жителей города и заинтересованных лиц, о возможности выбора в получении услуг разного медицинского характера, а также дистанционной записи к специалистам поликлинике и контролированию записями в личном кабинете.

# **2.1.3 Характеристики оборудования**

Для работы с системой управления контентом необходимо:

• операционные система Windows: Microsoft Windows 10/8/7.

Требования к компьютеру:

• IntelPentium III/800 мГц или выше (или аналогичный);

• минимум 4 Гб ОЗУ;

• 1 Тб свободного места на сервере;

Характеристики используемой системы:

• операционная система Microsoft Windows 10 x64;

• Intel Core i7;

• 16 Гб ОЗУ;

• разрешение экрана 1366x768.

# **2.2 Реализация проекта**

В соответствии с требованиями заказчика были выделены следующие этапы реализации проекта:

1. Проектирование веб-сайта.

2. Создание баз данных для сайта.

3. Разработка веб-сайта на платформе WordPress.

4. Тестирование веб-сайта.

# **2.3 Средства для разработки и проектирование сайта**

## **2.3.1 Анализ средств для разработки и создания Web-сайта**

Инструменты и программы, такие как HTML, CSS, JavaScript, FLASH, PHP, XML и т. д., обычно используются для создания современных веб-сайтов.

Конструктор сайтов

HTML — это фундаментальный язык в техническом мире веб-разработки, потому что его относительно легко выучить. Но излишняя простота также является его недостатком. HTML идеально подходил для ранней разработки технологии веб-сайтов, но по мере его дальнейшего развития возникали серьезные проблемы.

Язык HTML предназначен для формирования и оформления текстовых страниц, предназначенных для просмотра в сети Интернет с помощью специальной программы – браузера [14].

Генерация файлов HTML не требует использования мощного текстового процессора, который более удобен в использовании, например, редактор Notepad++. Содержимое HTML-страницы отображается в виде гипертекста, что означает, что в отображаемом на экране документе есть каким-то образом выбранное место, которое одним щелчком мыши может переместиться в другую часть той же страницы или на другую страницу.

Все, что нужно для чтения HTML документа — это веб-браузер, который интерпретирует HTML-разметку и отображает документ на экране в форме, предоставленной автором. HTML позволяет использовать теги для обозначения расположения заголовков или абзацев в документе, а затем позволить веб-браузеру интерпретировать эти теги. Основное преимущество HTML заключается в том, что документы можно просматривать в различных типах веб-браузеров и на различных платформах [15].

CSS (каскадные таблицы стилей) — каскадные таблицы стилей — пытаются отделить детали дизайна страницы от ее структуры и содержимого. В классическом HTML смешаны структура и оформление: рядом с абзацем текста его цвет, размер шрифта, тип шрифта и так далее. Если вам нужно изменить простые параметры, такие как размер шрифта на вашем сайте, вам придется переписать все страницы. Команды CSS позволяют назначать стили всем объектам, описания которых можно полностью хранить в отдельных файлах. Используя декодеры CSS, вы можете изменить размер шрифта всех страниц вашего сайта, исправив только один файл стиля.

Чем больше разных стилей вы примените, тем веселее будет менять дизайн. Самое главное, что вам не придется изменять готовый HTML-документ на сервере. Браузер пользователя получит доступ к файлу со стилями (по ссылке) и придаст странице желаемый вид [16].

CSS можно добавить на веб-страницу тремя способами: встроенным, вложенным и связанным с отдельным файлом CSS. Логично использовать ссылки на файлы CSS.

Таблица стилей — это отдельный файл, содержащий только правила CSS. Неограниченное количество веб-страниц может использовать такие файлы, что позволяет обеспечить надежный и идентичный внешний вид для всех страниц сайта [17].

Фреймы — элемент языка HTML, позволяющий строго разделить страницу на несколько отдельных окон и разместить в каждом окне собственную веб-страницу. В основном он используется для организации меню, которые продолжают появляться на экране, в то время как сама информация находится в другом окне. Flash — это технология, разработанная Macromedia Inc. Разнообразить обычные текстовые страницы сайта красочной и интерактивной векторной графикой. Технология Flash позволяет создавать потрясающие презентационные видеоролики и интерактивные интерфейсы, создавая новое комфортное качество на вашем сайте.

Преимущества Flash:

* + платформенная независимость;
  + распространенность и предоставление дизайнерам огромное количество инструментов;
  + правильно созданные Flash-файлы занимают мало места и достаточно быстро загружаются.

Недостатки Flash:

* + Flash-технология сложна в изучении;
  + нет желания у многих пользователей Интернет обновлять версии проигрывателей Flash, и следовательно многие новые Flash-модули не работают;
  + Долгая загрузка из-за большого объема графических файлов.

JavaScript. Это отличный инструмент, позволяющий HTML-страницам, загруженным в браузер, динамически управлять своим содержимым и самим браузером. Реализация JavaScript под названием «динамический HTML» позволяет веб-страницам иметь почти полный пользовательский интерфейс, включая многоуровневые раскрывающиеся меню, перетаскивание, анимацию и многое другое. Текст программы встраивается непосредственно в HTML-документ и интерпретируется самим браузером.

Популярность языка JavaScript связана с его широким спектром возможностей для взаимодействия с элементами веб-страницы без перезагрузки веб-страницы. Это позволяет скрывать и показывать фрагменты дизайна, перемещать их и изменять внешний вид. С помощью этих действий вы можете создавать эффекты презентации, меню, мини-игры, обрабатывать данные форм и управлять контентом [18]. PHP — это язык программирования высокого уровня, используемый для написания скриптов для различных целей. Первая версия этого языка была создана Расмусом Лердофом для разработки веб-сайтов. Отсюда и исходное название языка — PHP (Personal Homepage — Персональная домашняя страница). По мере того, как его популярность росла, PHP вырос настолько, что стал полноценным языком программирования. Теперь это называется обработка гипертекста. Высокий уровень языка означает простоту его синтаксиса и принципов программирования [19].

Интерпретатор PHP имеет открытый исходный код, что позволяет программистам по всему миру расширять и улучшать его функциональность. В разработке PHP участвовало международное сообщество программистов, что обеспечило достаточно быстрое развитие языковых средств [20].

PHP можно использовать в различных операционных системах: Windows, Linux, Unix и Mac OS. Между тем есть два варианта использования PHP — создание веб-приложений и разработка программ общего назначения. Поэтому предусмотрено два исполняемых модуля интерпретатора:

PHP CGI (Common Getwaitc Interface) — используется для выполнения скриптов через веб-сервер и отображения результатов в окне браузера;

PHP CLI (интерфейс командной строки) — для выполнения общих сценариев. Этот модуль имеет интерфейс командной строки.

Сценарии PHP хранятся в виде простых текстовых файлов, поэтому их можно писать и редактировать с помощью простого текстового редактора, такого как Блокнот. Расширение файла скрипта может быть любым, но чаще всего используется .рhp.

Необходимо обратить внимание на особенности взаимодействия PHP и веб-сервера.

PHP наиболее широко используется для веб-приложений, то есть для создания сценариев, которые выполняются на стороне сервера в Интернете, а не в браузере. Как известно, скрипты также могут выполняться на стороне клиента, но они пишутся на других языках и их код вставляется в HTML-документы.

XML (eXtensible Markup Language) — расширяемый язык разметки для описания структурированных данных в текстовой форме. Этот текстовый формат во многом похож на HTML и был разработан специально для хранения и передачи данных.

XML позволяет описывать и передавать такие структурированные данные, как:

* отдельные документы;
* метаданные, описывающие содержимое какого-либо узла Internet;
* объекты, содержащие данные и методы работы с ними;
* отдельные;
* всевозможные web-ссылки на информационные и людские ресурсы в Интернете.

Данные, описанные в XML, называются XML-документом. Язык XML легко читать и легко понимать. Исходный текст XML-документа состоит из набора XML-элементов, каждый из которых содержит открывающий и закрывающий теги. Каждая пара меток представляет один фрагмент данных. То есть, как и HTML, XML использует теги для описания данных. Однако, в отличие от HTML, XML допускает бесконечный набор пар тегов, каждый из которых представляет не содержащиеся в нем данные, а то, что они означают.

Вы можете использовать специальные директивы — таблицы стилей XSL (Extensible Style Sheet Language) для управления отображением элементов в окнах клиентских программ (например, окнах браузера). Эти XSL-таблицы позволяют определять внешний вид элемента на основе его положения в документе, то есть к двум элементам с одинаковым именем могут применяться разные правила форматирования. Кроме того, базовым языком XSL является XML, а это означает, что таблицы XSL являются более общими, и для контроля правильности таких таблиц стилей можно использовать DTD или схемы данных.

Инструменты для обработки данных, полученных в формате XML, могут быть разработаны в средах Visual Basic, Java или C++ [21].

программное обеспечение для веб-разработки

Создание, поддержка и развитие сайта осуществляется с помощью специального программного обеспечения.

HTML-документы можно создавать в любом текстовом редакторе, например в Блокноте. HTML-документы могут быть преобразованы из многих программ, таких как Microsoft Word. После преобразования страница текста может быть превращена в огромный HTML-файл размером более 100 КБ. HTML-документы можно создавать с помощью пакетов визуализации, таких как Microsoft FrontPage. Эти редакторы сами генерируют html-код документа, а разработчик просто выбирает нужные ему опции из меню.

Macromedia Corporation — один из лидеров в разработке программного обеспечения для подготовки веб-публикаций. Большой популярностью пользуются пакеты Macromedia: Dreamweaver, HomeSite и специализированные пакеты для создания компьютерной графики и анимации.

Особого внимания заслуживает создание графики для веба. Следует помнить, что существует два вида графики: растровая (при описании каждой точки изображения) и векторная (при задании формулы построения изображения). Последний способ построения графиков более компактен и идеально подходит для создания анимаций.

Для использования PHP необходимы смежные серверные компоненты, а именно:

* web-сервер;
* сервер баз данных - это серверный компонент, который позволит работать с базами данных, управляемых из PHP.

В дополнение к серверным компонентам также очень полезны инструменты, без которых работа PHP-программиста может быть затруднена. Это редакторы кода PHP, позволяющие подсвечивать синтаксис PHP, что значительно упрощает написание сценариев [22].

JavaScript — это скриптовый язык программирования. Часто используется как встраиваемый язык для программного доступа к объектам приложения. Он наиболее широко используется в браузерах в качестве языка сценариев для создания интерактивных веб-страниц.

Для написания приложений вы можете использовать NotePad++, бесплатный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса для большого количества языков программирования и разметки. Базовый функционал программы может быть расширен плагинами и сторонними модулями, такими как компиляторы и препроцессоры [23].

CSS и многие другие приложения в настоящее время поддерживаются почти всеми браузерами. Все, что вам нужно, это текстовый редактор для создания CSS, но есть много инструментов, которые облегчают этот процесс. Сегодня большинство инструментов веб-разработки обеспечивают некоторую поддержку таблиц стилей CSS.

Использовать XML в своих программах очень просто, поскольку он позволяет без особых усилий реализовать удобный формат обмена данными. Вот почему XML так популярен, потому что он прост в использовании и существует множество доступных инструментов обработки.

Для css и xml вы можете использовать HTML-Kit — редактор HTML, XHTML и XML с подключаемыми модулями, который предоставляет рекомендации по CSS и многое другое.

# **2.3.2 Фреймворки, используемые при разработке Web-сайтов: сравнительный анализ**

Фреймворк — это программный фреймворк, который реализует базовую архитектуру проекта. Самыми известными современными php-фреймворками являются Zend Framework, Yii, CodeIgniter, Symfony и многие другие.

Фреймворки предоставляют разработчикам очень мощный инструмент для разработки более гибких и менее подверженных ошибкам приложений за меньшее время. Фреймворки часто помогают ускорить процесс разработки, предоставляя необходимую функциональность. К ним относятся управление пользователями/разрешениями, доступ к данным, кэширование и многое другое. Фреймворки помогают сосредоточиться на более важных деталях дизайна и с легкостью управлять проектами.Zend Framework (ZF) - является объектно-ориентированным фреймворком с открытым исходным кодом, использует PHP5 и выпущен под лицензией New BSD License.

Zend Framework был задуман в начале 2005 года, когда много новых фреймворков набирали популярность в сообществе web-разработчиков. О ZF впервые было объявлено на первой Zend конференции в октябре 2005 года.

Функции Zend Framework включают в себя:

* большое количество модулей и библиотек;
* разрабатывается разработчиками ядра PHP;
* поддержка unit-тестов;
* достаточно низкая скорость работы с библиотеками;
* встроенный JS фреймворк DOJO и библиотеки для него.

Symfony – web-фреймворк написанный на PHP, использует платформу модель-представление-контроллер (MVC). Был выпущен под лицензией MIT. Symfony является свободным программным обеспечением.

Symfony не следует путать с Symfony CMS, Open Source XML/XSLT системой управления контентом. Symfony ускоряет создание и поддержку web-приложений. В настоящее время совместима с ORM Propel и Doctrine. Следует отметить, что на типичном хостинге, где ускоритель PHP отсутствует, Symfony может использовать собственный движок кэширования для ускорения выполнения кода.

Symfony направлен на создание надежных приложений, с целью предоставить разработчикам полный контроль над конфигурацией.

Достоинства:

* простой в освоении;
* достаточно высокая скорость работы ядра.

Yii — высокопроизводительный фреймворк, и для организации из 10+ человек имеет смысл его изучать, а с одним Yii сложно справиться. Этот фреймворк позволяет создавать очень эффективные системы.

Yii — лучший выбор между Zend Framework и Codeingiter. Он хорошо документирован, и вы всегда можете вникнуть в код, чтобы понять, что и как работает.

Фреймворк строго написан на PHP5 и строго следует концепции ООП, поэтому его легко понять, если у вас есть базовое понимание принципов ООП.

Yii более сложен (более функционален) и лучше структурирован, чем CodeIgniter.

Он меньше и более производительный, чем фреймворк Zend. Для написания приложений требуется гораздо меньше кода, в то время как у Yii масса функций.

[CodeIgniter](http://codeigniter.com/) - это PHP фреймворк, который использует платформу MVC, имеет классы для доступа к данным, классы для работы с почтой, FTP и XML-RPC.

У [CodeIgniter](http://codeigniter.com/) имеется исчерпывающая документация для начала работы [24].На рисунке 3 изображён интефейс [CodeIgniter](http://codeigniter.com/).

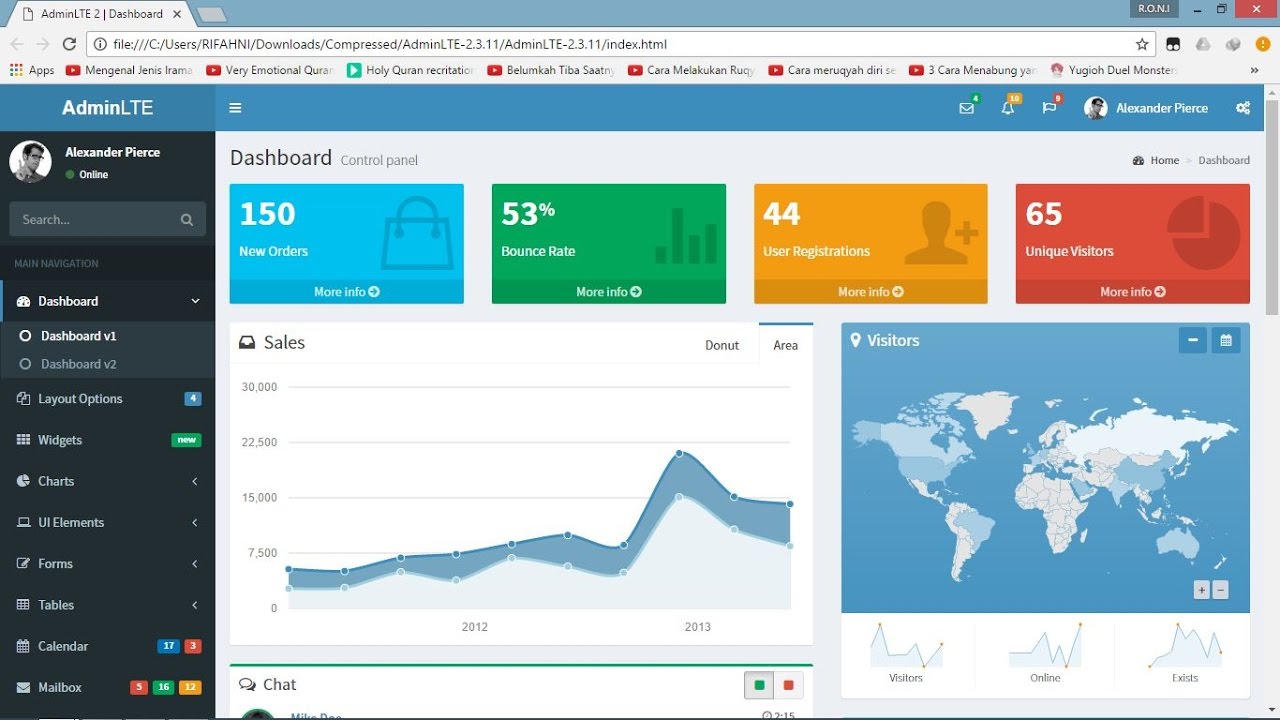


Рисунок 3- Интерфейс CodeIgniter

Таблица 2 Характерные особенности фреймворков

|  |  |  |  |
| --- | --- | --- | --- |
|  | [Zend Framework 1](http://framework.zend.com/) | [Code Igniter 2](http://codeigniter.com/) | [Symfony 2](http://www.symfony-project.com/) |
| Описание | Академически грамотный код. Очень гибок. Требует хорошего знания PHP и ООП. | Почти микрофреймворк. Очень лёгкий для изучения. Отличная документация. Гибок. Легко использовать сторонний код. | Активно использует командную строку. Мощный ORM, хорошая система view, генераторы кода. |
| Необходимый уровень знаний | PHP5, ООП, шаблоны проектирования. | PHP, Основы ООП | PHP5, ООП, ORM, консоль. |
| Платформа | [PHP](http://ru.wikipedia.org/wiki/PHP) 5 | [PHP](http://ru.wikipedia.org/wiki/PHP) 4/5 | [PHP](http://ru.wikipedia.org/wiki/PHP) 5 |
| Сложность установки и настройки | Высокая | Низкая | Высокая |
| Поддерживаемые [БД](http://ru.wikipedia.org/wiki/%D0%91%D0%94) | [MySQL](http://ru.wikipedia.org/wiki/MySQL), [Oracle](http://ru.wikipedia.org/wiki/Oracle_%28%D0%A1%D0%A3%D0%91%D0%94%29), [PostgreSQL](http://ru.wikipedia.org/wiki/PostgreSQL), [MS SQL](http://ru.wikipedia.org/wiki/MS_SQL), [SQLite](http://ru.wikipedia.org/wiki/SQLite) | [MySQL](http://ru.wikipedia.org/wiki/MySQL), [Oracle](http://ru.wikipedia.org/wiki/Oracle_%28%D0%A1%D0%A3%D0%91%D0%94%29), [PostgreSQL](http://ru.wikipedia.org/wiki/PostgreSQL), [MS SQL](http://ru.wikipedia.org/wiki/MS_SQL), [SQLite](http://ru.wikipedia.org/wiki/SQLite) | [MySQL](http://ru.wikipedia.org/wiki/MySQL), [Oracle](http://ru.wikipedia.org/wiki/Oracle_%28%D0%A1%D0%A3%D0%91%D0%94%29), [PostgreSQL](http://ru.wikipedia.org/wiki/PostgreSQL), [MS SQL](http://ru.wikipedia.org/wiki/MS_SQL), [SQLite](http://ru.wikipedia.org/wiki/SQLite) |
| Русскоязычное сообщество | [Да](http://zendframework.ru/) | [Документация](http://code-igniter.ru/), [форум](http://code-igniter.ru/forum/), блоги | Нет |
| Лицензия | [New BSD](http://www.opensource.org/licenses/bsd-license.php) | [BSD](http://ru.wikipedia.org/wiki/BSD) | [MIT](http://ru.wikipedia.org/wiki/MIT_License) |

**2.3.3 Проектирование веб-сайта**

Заказчиком является поликлиника, расположенная в городе Абиджан находящаяся по адресу 4 Bd de l'Indenié, Abidjan, Кот-д'Ивуар, поликлинику посещают жители города и ближайших населенных пунктов города Абиджан.

Деятельность клиники началась в 2015 году. В настоящее время в клинике проводят полный спектр услуг для поддержания и лечения здоровья человека. В поликлинике работают специалисты разных специальностей таких, как аллерголог, дерматолог, невролог и другие.

Структура сайта представлена на рисунке 2.

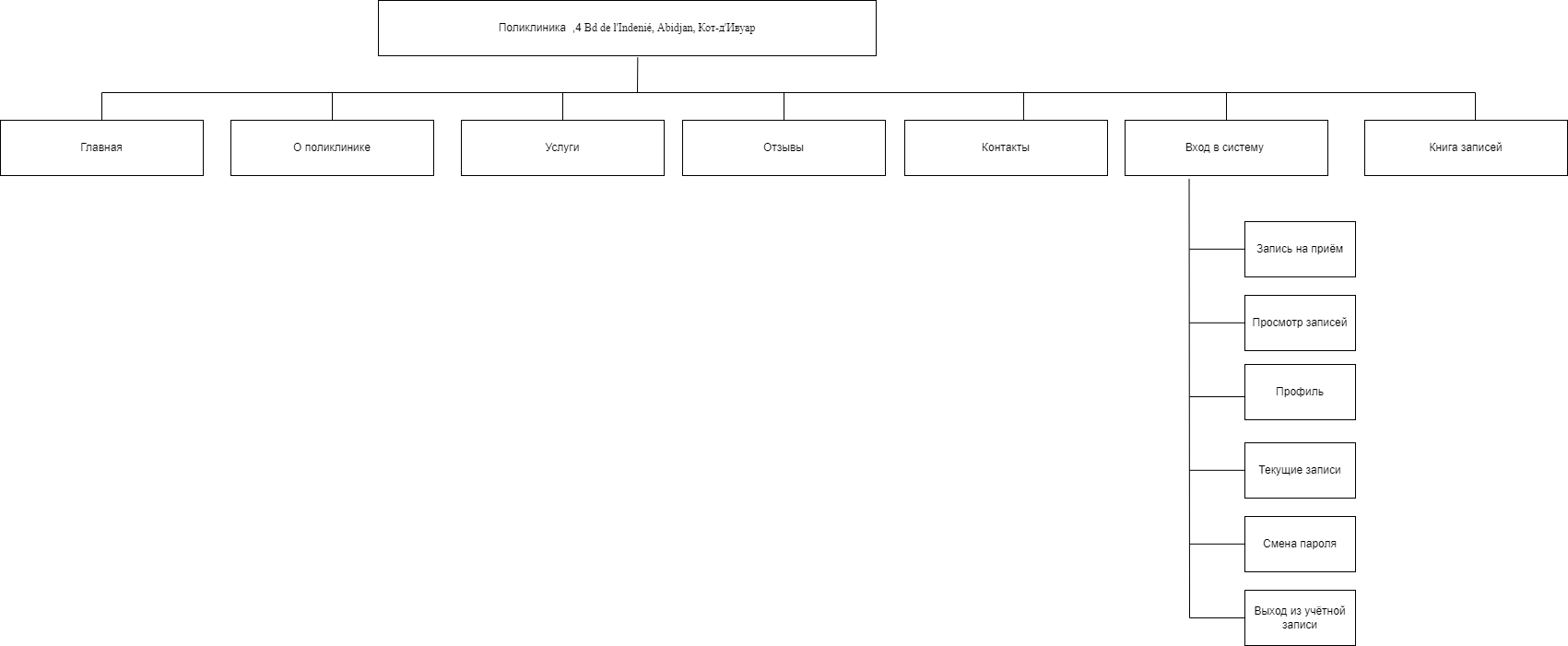


Рисунок 2- Структура сайта

1. Первая страница.

Главная страница — это лицо сайта, где обычно приводится краткая информация о направлениях деятельности поликлиники, предоставляемых услуг.

Основное назначение главной страницы web-сайта - рассказать пользователям о поликлинике, ее преимуществах. Разработанный сайт оказывает медицинские услуги. Это и отражено на главной странице путём размещения на ней прямых ссылок на главную информацию о поликлинике и её услугах.

1. О поликлинике.

В этом разделе приводится краткая информация о поликлинике.

1. Услуги.

В этом разделе описаны медицинские услуги, которые предоставляет поликлиника

1. Отзывы.

Раздел содержит отзывы посетителей поликлинике совместно с оцениванием её по 5 бальной шкале

1. Книга записей

Раздел «Книга записей» — раздел в котором показано свободное время для записей у специалистов поликлиники.

1. Вход в систему

Раздел «Вход в систему» позволяет пользователю зарегистрироваться на веб-сайте или войти под своим логином и паролем, если он уже зарегистрирован на веб-сайте.

1. После регистрации или входа на сайт пользователь может записаться на приём к специалисту, посмотреть активные записи, посмотреть предыдущие записи или сменить пароль своей учётной записи.

Каркас сайта считается одним из ключевых эскизных проектов, которые делаются перед созданием оригинал-макета сайта, и применяется при разработке дизайна сайта, для того чтобы продемонстрировать его структуру, 25 настраиваемые пути пользователя, наиболее значимые элементы интерфейса пользователя, их расположение и взаимосвязь между страницами сайта.

За управлением базами данных отвечает специальный плагин KiviCare.

KiviCare — плагин для клиники и управления пациентами. Он разработан для врачей и клиник, чтобы обрабатывать несколько назначений врачей и встреч в клинике. Это позволяет делать, управлять и отслеживать запись пациентов на прием к врачам. Сохраняя конфиденциальность на переднем плане, каждый визит к врачу может быть конфиденциальным для других. Администратор создаёт учётную запись для врача и присылает на его почту пароль, который врач может поменять перейдя по ссылке полученной по электронной почте. На рисунке 3 показана панель администратора при создании учётной записи врача.

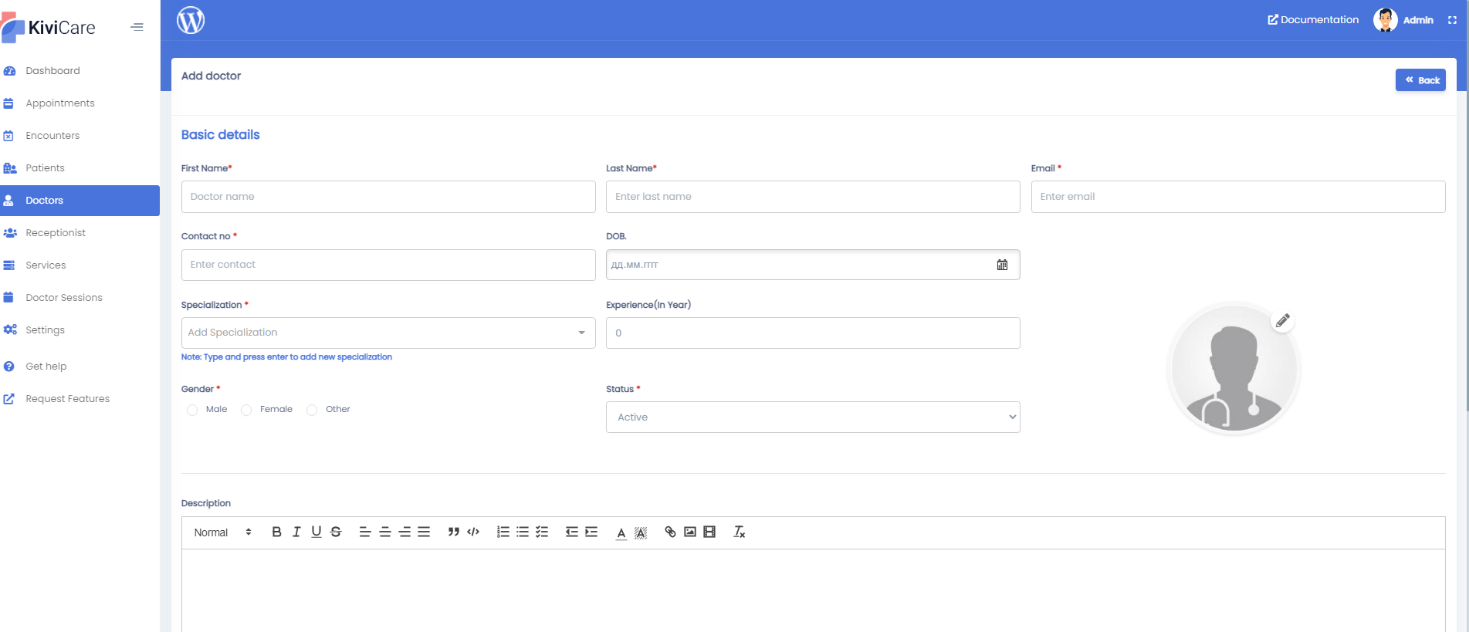


Рисунок 3- Панель администратора

Созданию шаблона веб-сайта предшествовало формирование блочной структуры, которая определяет размещение функциональных элементов вебсайта. На главной странице веб-сайта расположена шапка сайта с изображением логотипа и наименованием учреждения (этот элемент будет размещаться на всех страницах веб-сайта) и панель навигации (элемент будет размещаться на всех страницах веб-сайта). Блочная структура главной страницы представлена на рисунке 4.

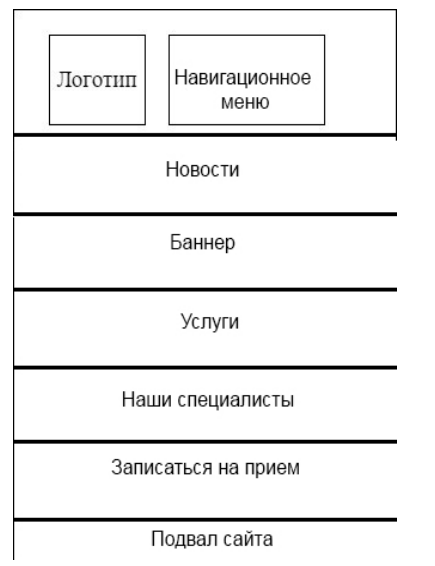


Рисунок 4- Структура главной страницы

На рисунке 5 и 6 изображены база данных пациентов и врачей , которые можно экспортировать в csv и excel.

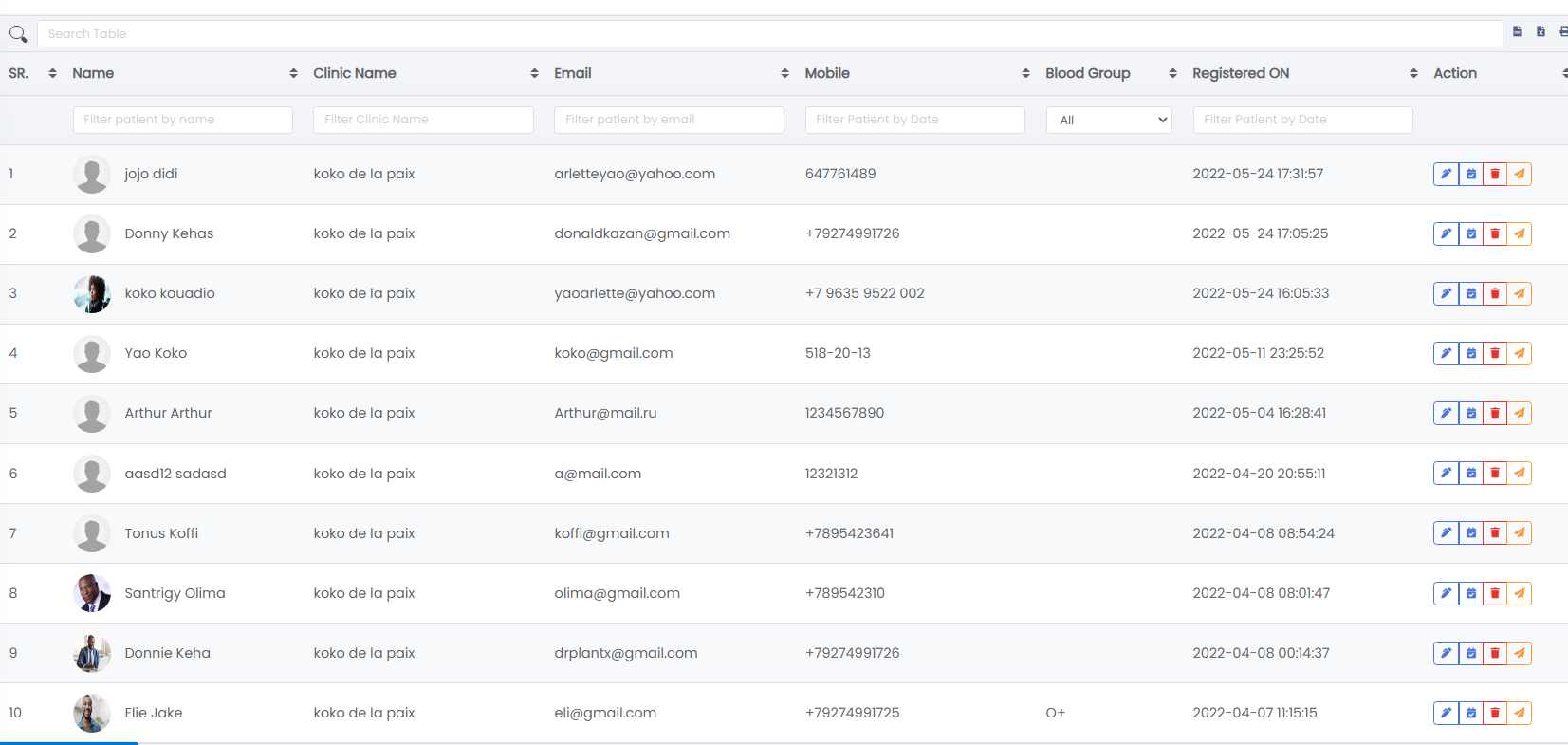


Рисунок 5- База данных пациентов

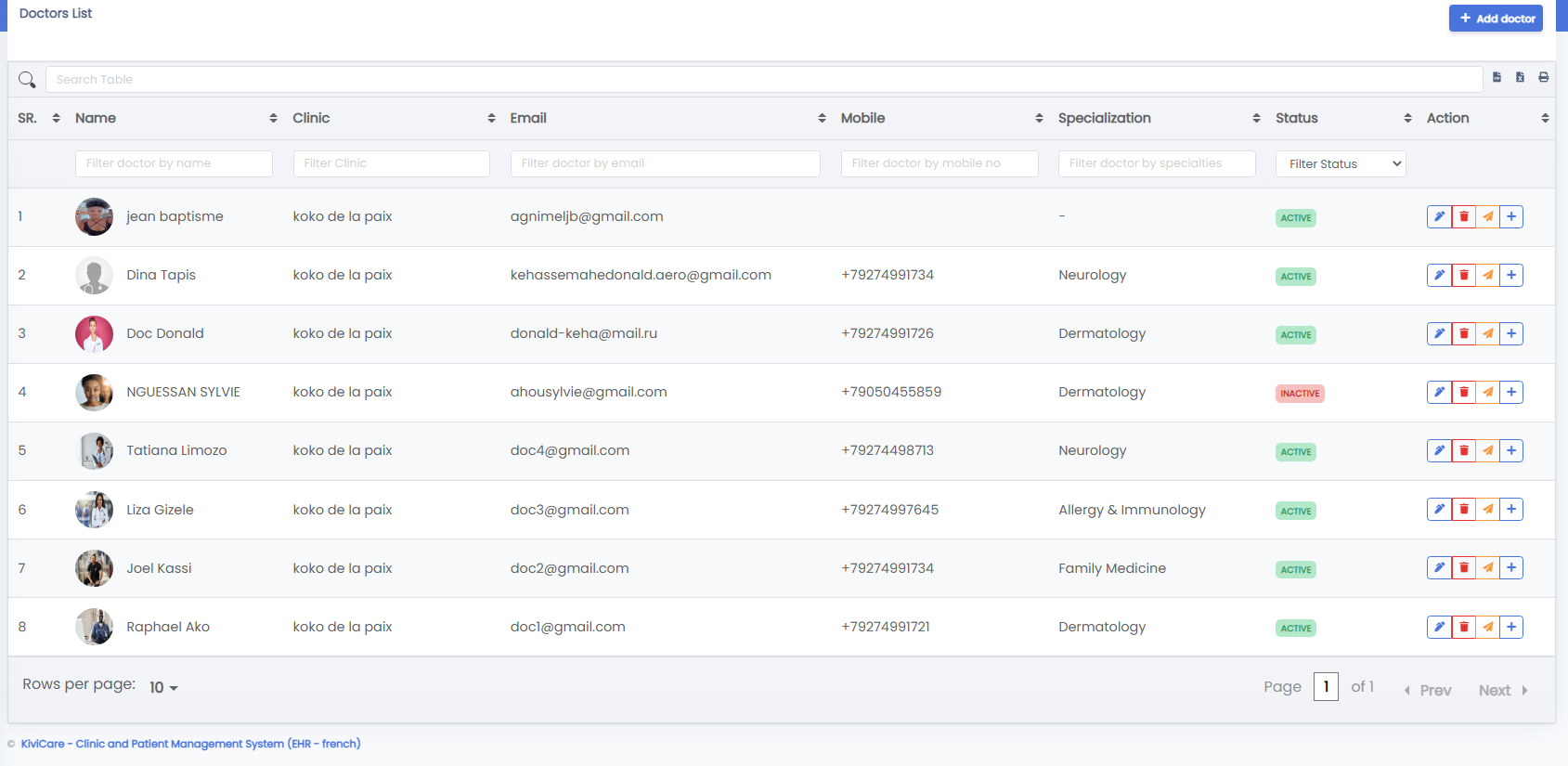


Рисунок 6- База данных врачей

# **2.3.4 Разработка веб-сайта в CMS WordPress**

Перед тем, как приступить к разработке веб-сайта в CMS WordPress, необходимо установить систему на хостинг и настроить необходимое ПО для работы с ним.

Для установки CMS WordPress был скачан дистрибутив c официального веб-сайта ru.wordpress.org. После скачивания дистрибутива был установлено программное обеспечение OpenServer. Данный программный комплекс включает в себя тщательно подобранный набор серверного программного обеспечения, а также невероятно удобную и продуманную управляющую утилиту, которая обладает мощными возможностями по администрированию и настройке всех доступных компонентов.

Базовые модули

* Apache: 2.2.31, 2.4.38, 2.4.41, 2.4.53 + auth\_ntlm, fcgid, maxminddb, xsendfile
* Bind: 9.16.28
* FTP FileZilla: 0.9.60
* Ghostscript: 9.56.1
* Nginx: 1.21.6 + ssl\_preread, image\_filter, geoip, geoip2, brotli
* NNCron Lite: 1.17
* Sendmail: 32

После установки системы необходимо подготовить будущую базу данных и панель администратора для дальнейшей работы с дистрибутивами. OpenServer Panel изображена на рисунке 7.

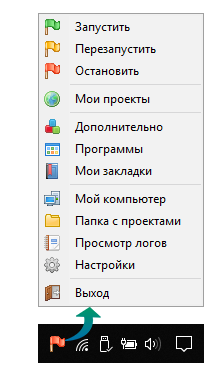


Рисунок 7- OpenServer Panel

Для начала, необходимо подготовить файлы исходного шаблона для переноса на WordPress. Нужно создать новую папку с нашим веб-сайтом в WordPress.

После всех манипуляций создаём администратора для веб-сайта и заходим в WordPress для создания сайта. На рисунке 8 изображён интерфейс Wordpress.

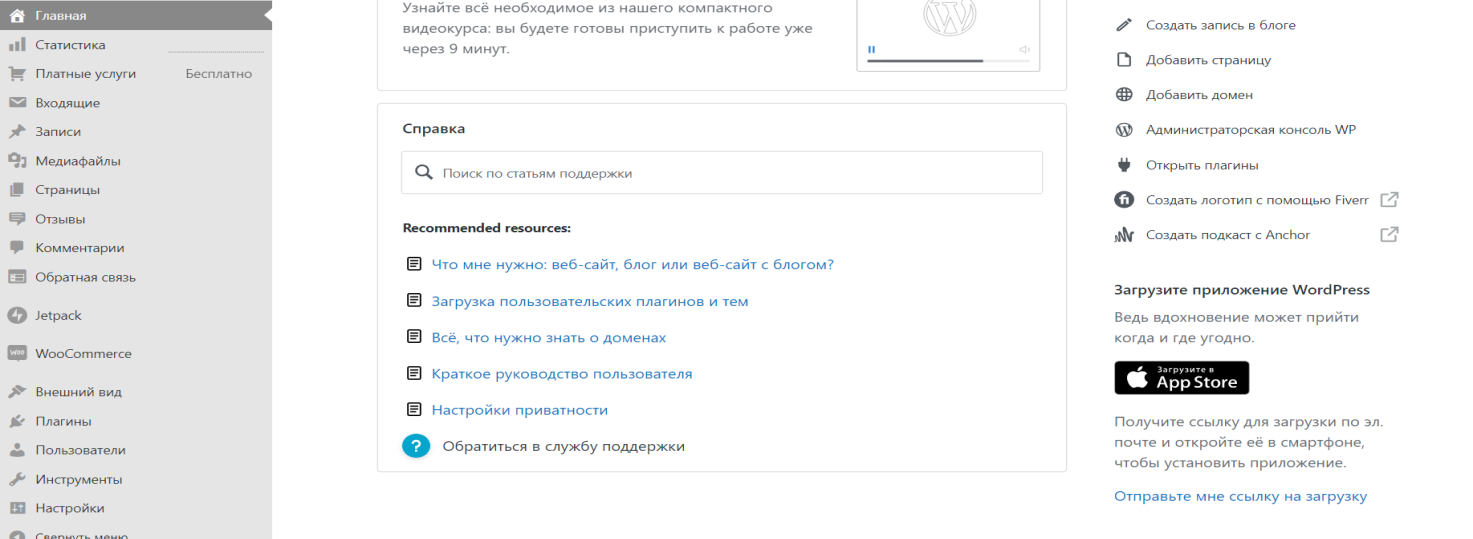


Рисунок 8- Интерфейс Wordpress

После того, как сайт был создан нужно создать домен для него, для сайта поликлиники был выбран домен <https://delya-food.online/>. Листинг сайта изображён в Приложении А.

На основе созданных шаблонов и с помощью CMS WordPress был разработан конечный продукт — веб-сайт. На главной странице расположена активная кнопка «Book Appointment», контактная информация, активные иконки оказываемых услуг и информация о руководстве медицинской клиники. Главная страница веб-сайта представлена на рисунке 9.

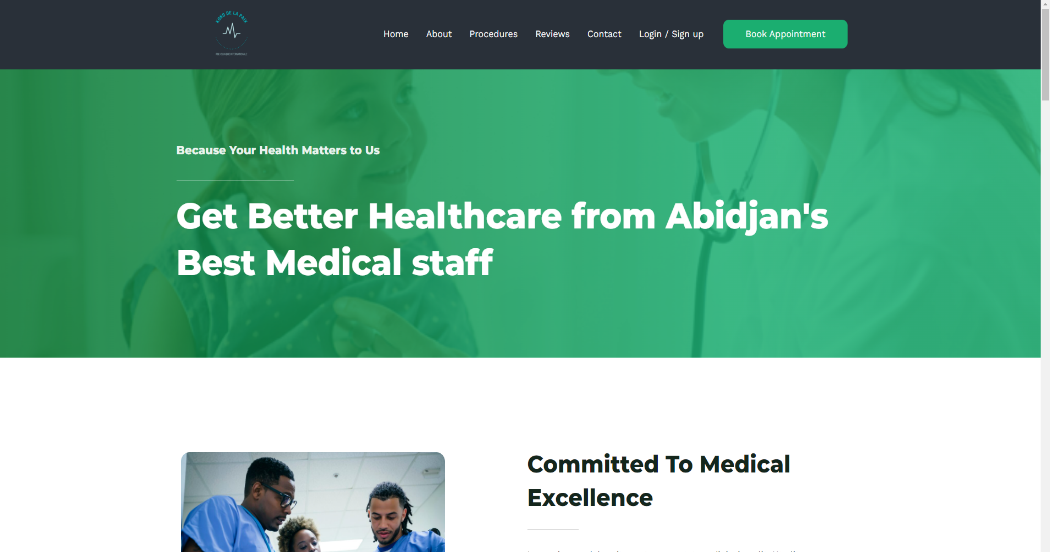


Рисунок 9- Главная страница веб-сайта поликлиники

В блоке «About» главной страницы расположена кликабельная иконка на услуги снизу блока, сверху написана информация о поликлинике. Блок «About» представлен на рисунке 10.

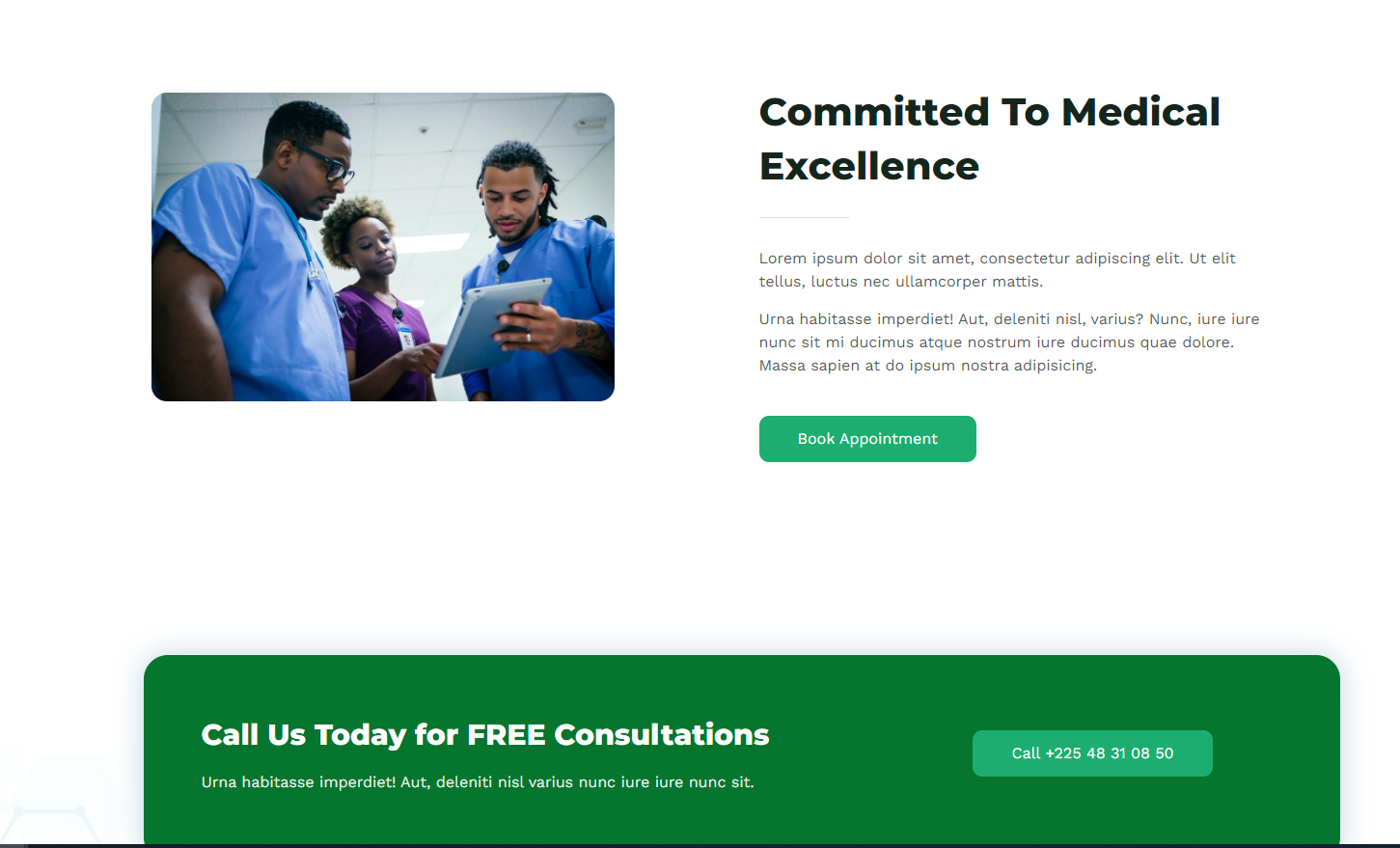


Рисунок 10- Блок «About»

В блоке «Procedures» главной страницы расположена информация об услугах поликлиники. Блок «Procedures» представлен на рисунке 11.

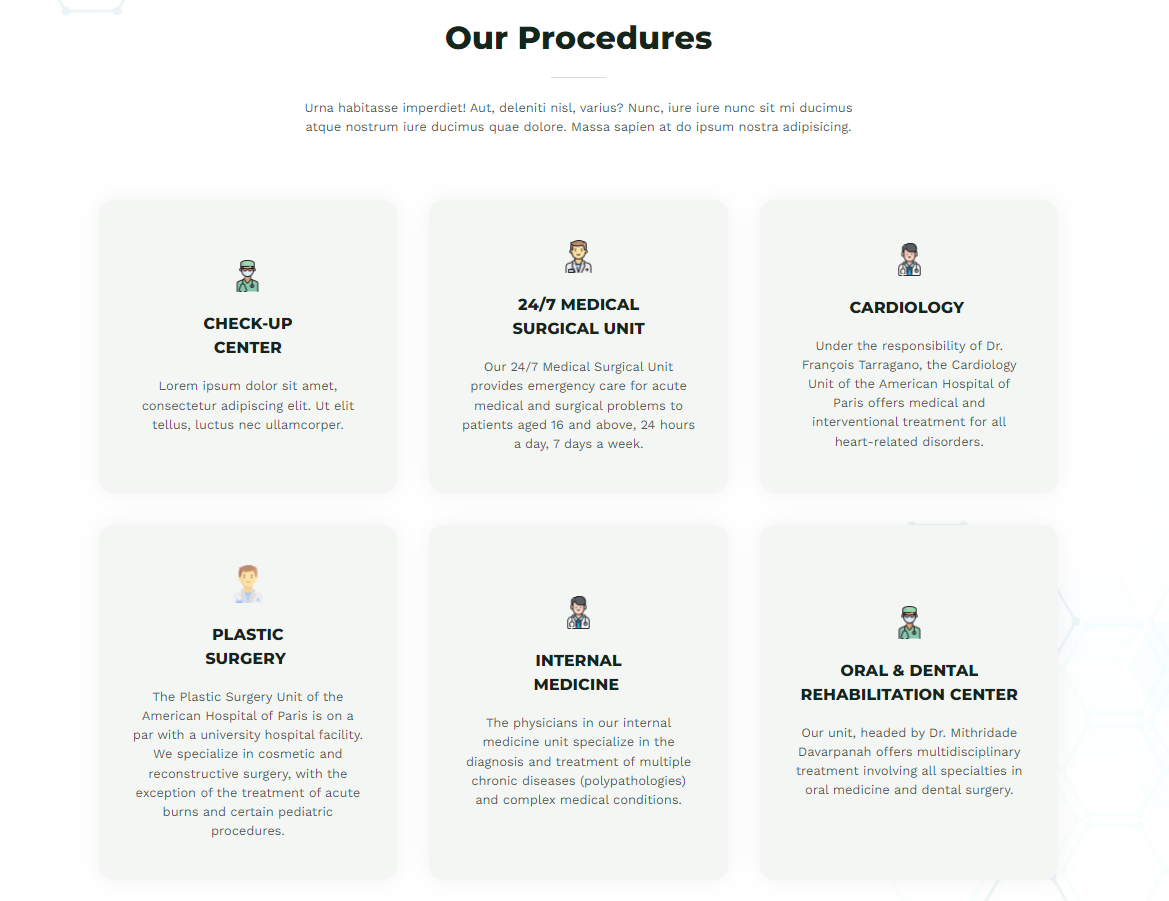


Рисунок 11- Блок «Procedures»

В блоке «Reviews» главной страницы расположены отзывы клиентов поликлиники. Блок «Reviews» представлен на рисунке 12.

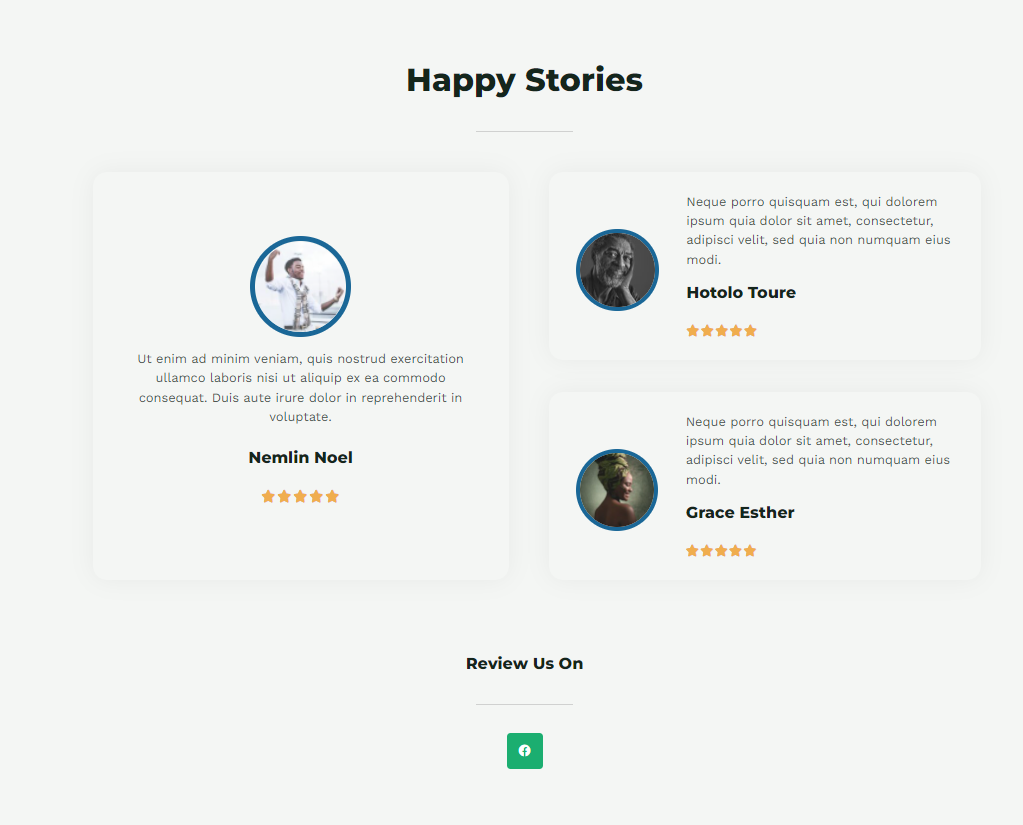


Рисунок 12 Блок «Reviews»

В блоке «Contact» главной страницы расположены адрес, контакты и местоположение поликлиники. Блок «Contact» представлен на рисунке 13.

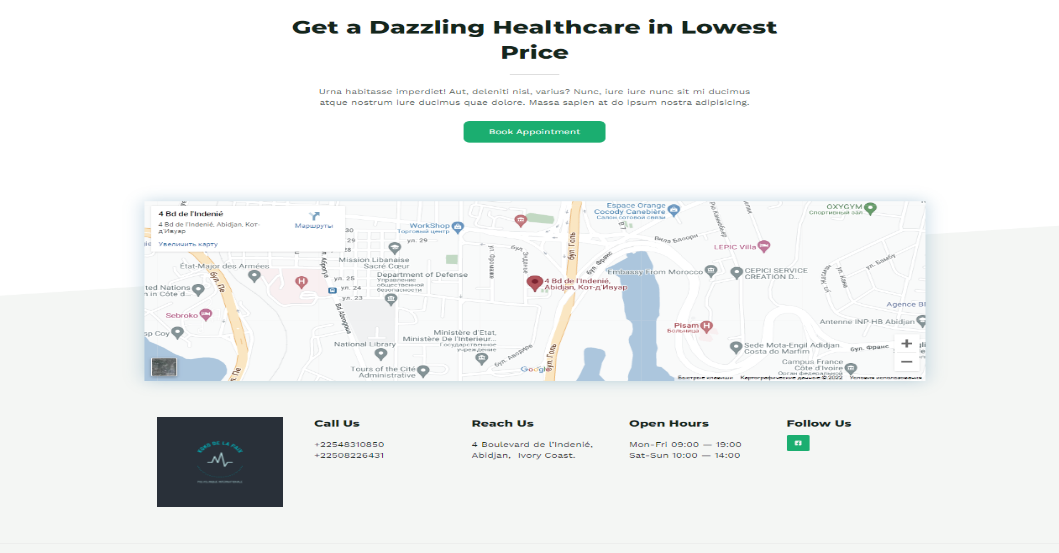


Рисунок 13- Блок «Contact»

При нажатии на кнопку главной страницы «Login/Sign Up» пользователя переносит на страницу регистрации и вхождения в аккаунт . Страница регистрации пользователя представлена на рисунке 14.

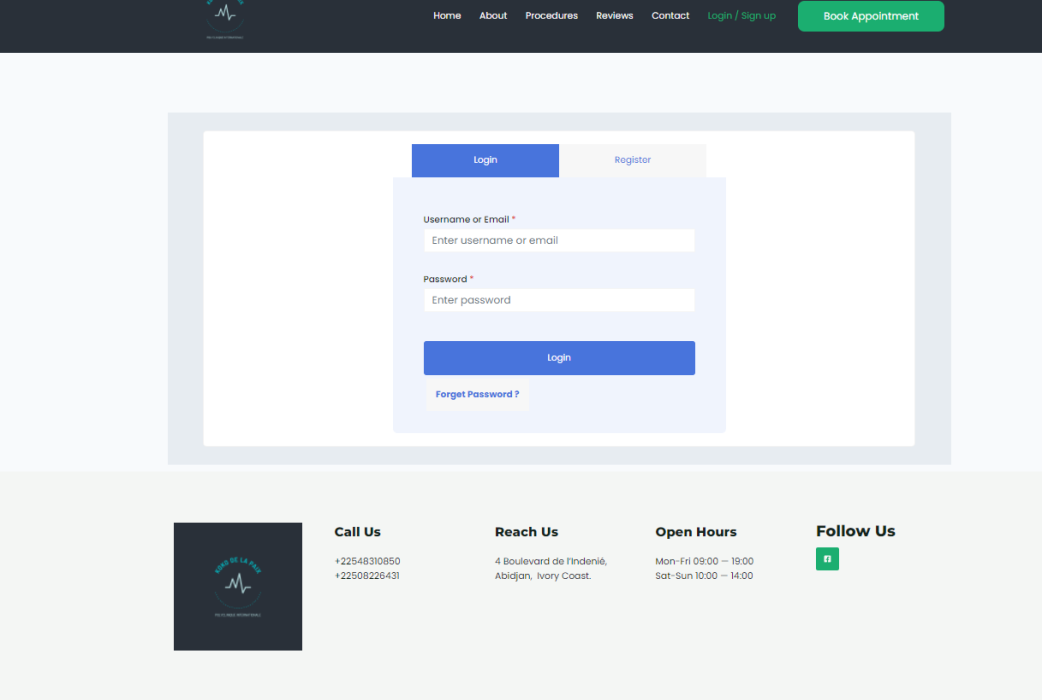


Рисунок 14 – Страница регистрации пользователя

На странице аккаунта показана информация о пользователе и о текущих записях. Страница аккаунта изображена на рисунке 15.

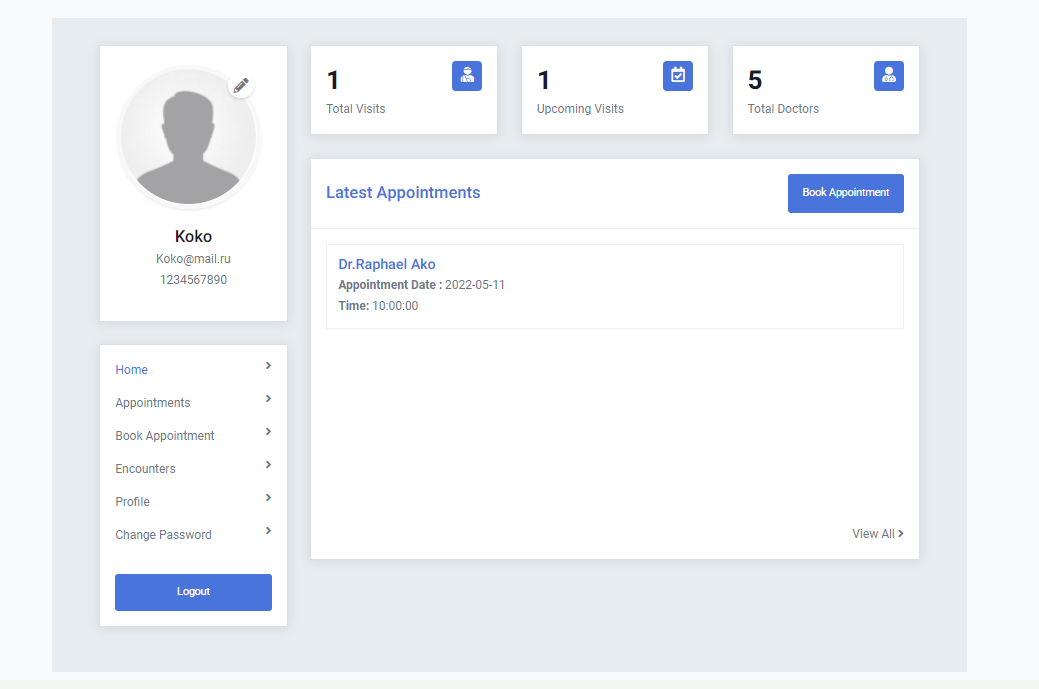


Рисунок 15- Страница аккаунта

На странице аккаунта в блоке «Appointments» страницы расположены записи пользователя к специалистам поликлиники. Блок «Appointments» представлен на рисунке 16.

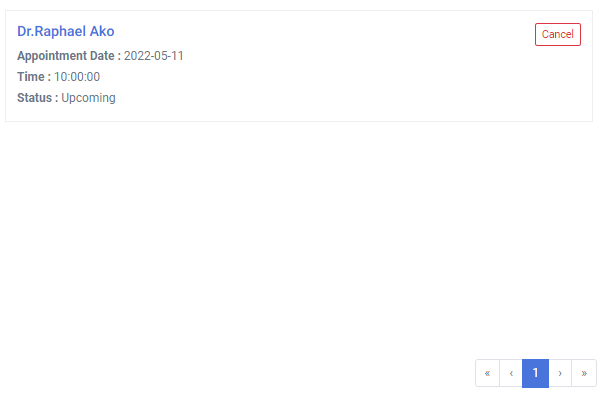


Рисунок 16- Блок «Appointments»

В блоке «Book Appointment» находится окно с выбором специалиста и временем записи к нему. Блок «Book Appointment» представлен на рисунке 17.

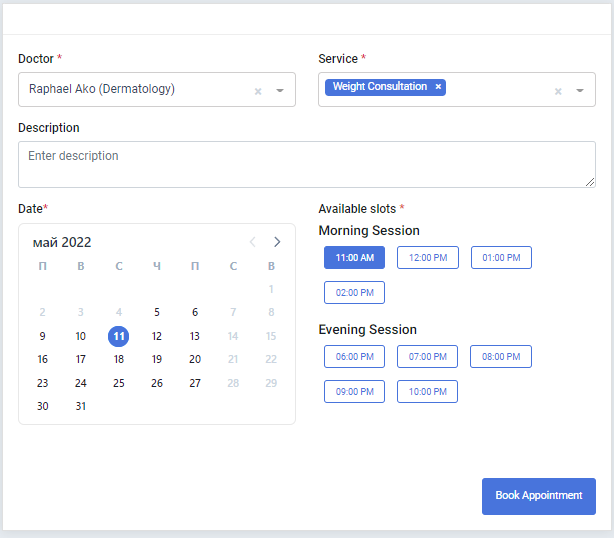


Рисунок 17- Блок «Book Appointment»

В блоке «Encounters» находятся старые записи пользователя в поликлинике. Блок «Encounters» представлен на рисунке 18.

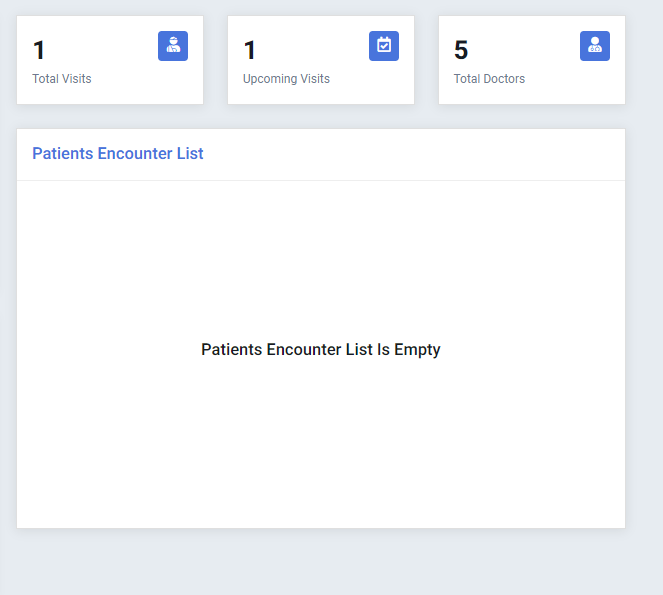


Рисунок 18- Блок «Encounters»

На странице аккаунта в блоке «Profile» страницы расположена информация о пользователе, которую пользователю может изменить в любой момент времени. Блок «Profile» представлен на рисунке 19.

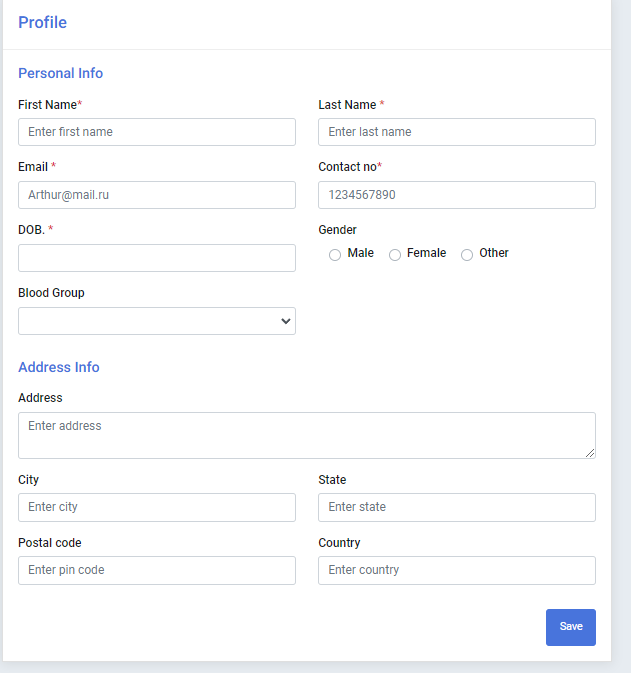


Рисунок 19 Блок «Profile»

Листинг веб сайта показан в Приложении А.

# **3. Тестирование веб-сайта и руководство пользователя**

# **3.1. Тестирование веб-сайта**

Тестирование сайта — это процесс контроля качества программного продукта.

Основная функция тестирования заключается в обнаружении и документировании дефектов (или ошибок) в программном обеспечении.

Тестирование программного обеспечения для интерактивных развлечений — это высокотехнологичная область, требующая компьютерных знаний, аналитических навыков и навыков критической оценки.

Существует несколько уровней тестирования программного обеспечения:

• Альфа-тестирование – имитация реальной работы системы штатными разработчиками;

• Бета-тестирование. Распространите бета-версию программного обеспечения среди большего числа людей, чтобы гарантировать, что продукт содержит достаточно ошибок, чтобы исправить их в будущих выпусках.

• Модульное тестирование — для тестирования используйте наименьший возможный тестовый компонент, например, один класс или функцию;

• Интеграционное тестирование – тестирование компонентов, подсистем или интерфейсов между системами;

• Системное тестирование – проверка сложных систем на соответствие требованиям;

Сайт прошел все этапы тестирования для различных версий операционных систем Microsoft Windows 7.8 и 10.

Таблица 2- Результаты функционального тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Категория  набора тестов | Ожидаемый результат | Результат |
| 1 | Проверка  работоспособности кнопок  главной страницы | Все кнопки главной страницы функционируют должным образом | Пройден |
| 2 | Проверка  работоспособности регистрация | Регистрация на веб сайте работает, данные сохраняются в базе данных | Пройден |
| 3 | Проверка валидации при регистрации | Валидация фамилии, имени и электронной почты работают | Пройден |
| 4 | Проверка  работоспособности верификации | Верификация на сайте работает, после внесения логина и пароля, пользователя переносит в его личный кабинет | Пройден |
| 5 | Запись пользователя к врачу | Выбор врача и даты записи к врачу | Пройден |
| 6 | Просмотр записей к врачам | После того, как пользователь заходит в личный кабинет он может видеть текущие и старые записи к врачам |  |

# **3.2. Руководство пользователя**

1.Общее назначение веб-сайта.

Веб-сайт предназначен для просмотра информации о поликлинике, записи к врачам, редактированию информации учётной записи и просмотру текущих и прошлых записей пользователя

2.Варианты использования веб-сайта.

В ходе анализа требований была разработана UML-диаграмма вариантов использования пользователем веб-сайта. (рис.20)

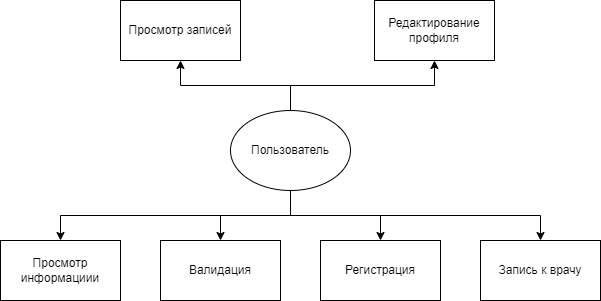


Рисунок 20- Варианты взаимодействия пользователя с веб-сайтом

Для того, чтобы зайти на веб-сайт поликлиники требуется запустить браузер и перейти по ссылке https://delya-food.online/. После перехода по ссылке на главной странице пользователь может посмотреть информацию о поликлинике и перейти на страницу регистрации нажав на кнопку Login/Sign up.

На рисунке 21 изображена страница регистрации пользователя.

После регистрации пользователя на странице регистрации пользователя переносит на профиль, в котором он может записываться на приём перейдя на вкладу Book Appointments , посмотреть свои записи к врачам на вкладке Appointment и изменить данные во вкладке Profile. Вкладки Appointments, Book Appointment и Profile показаны на рисунках 22,23,24.

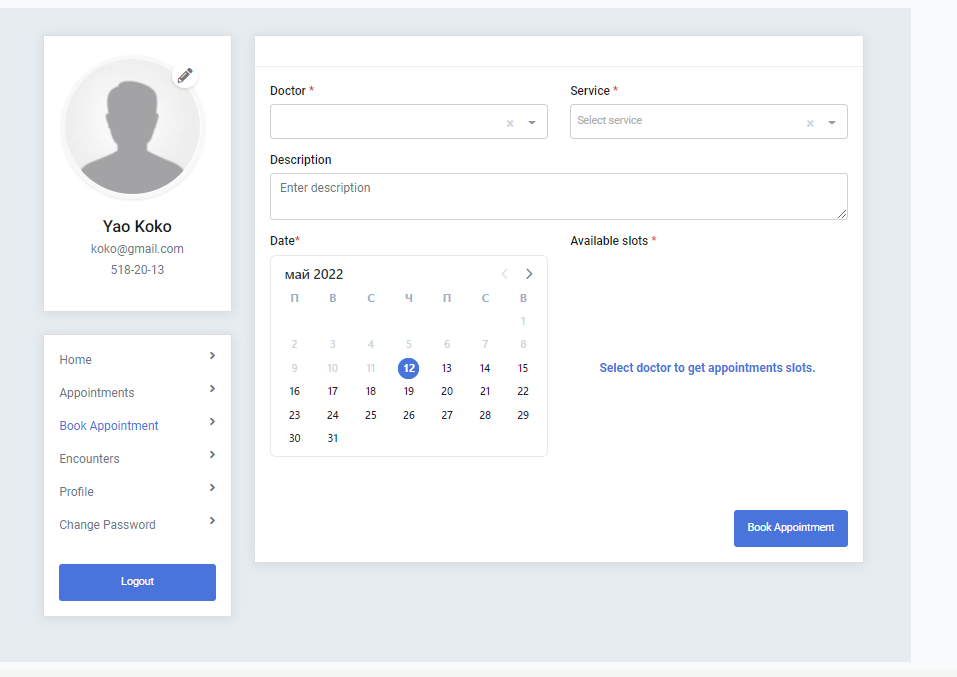


Рисунок 22- Вкладка Book Appointment

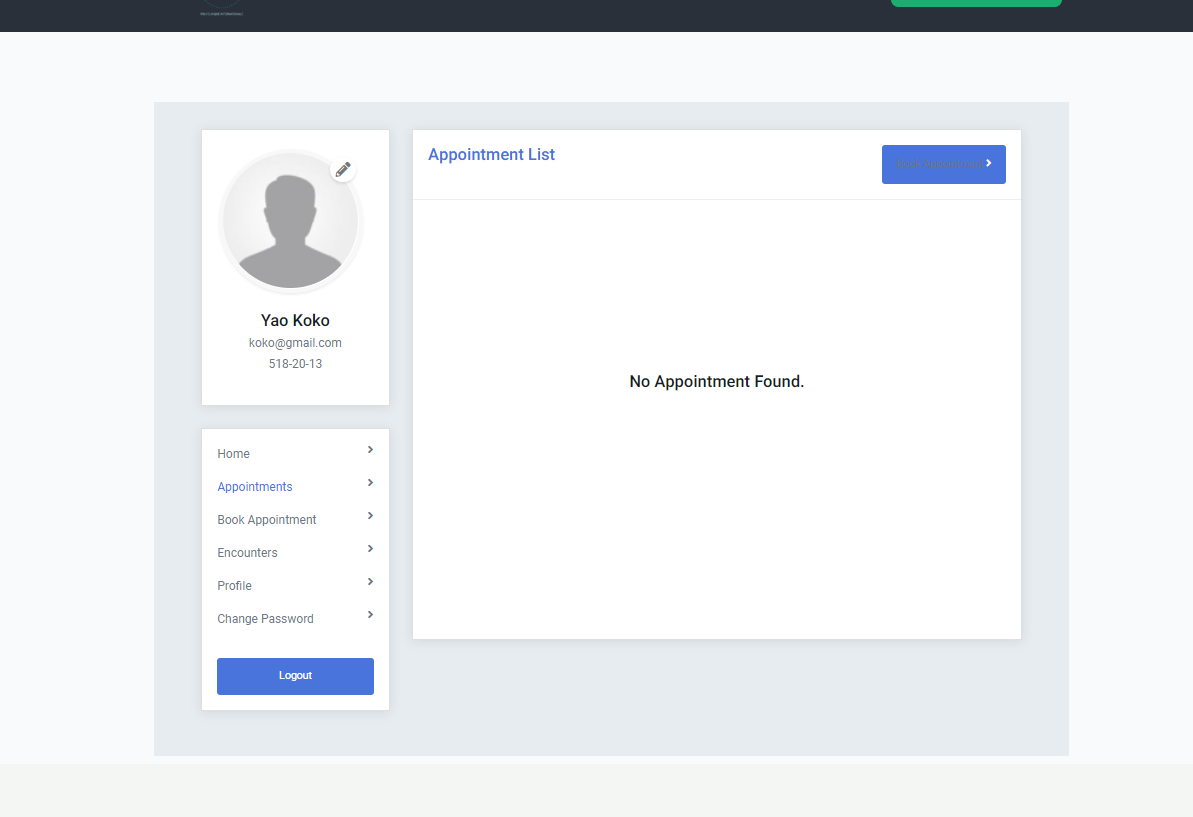


Рисунок 23- Вкладка Appointment

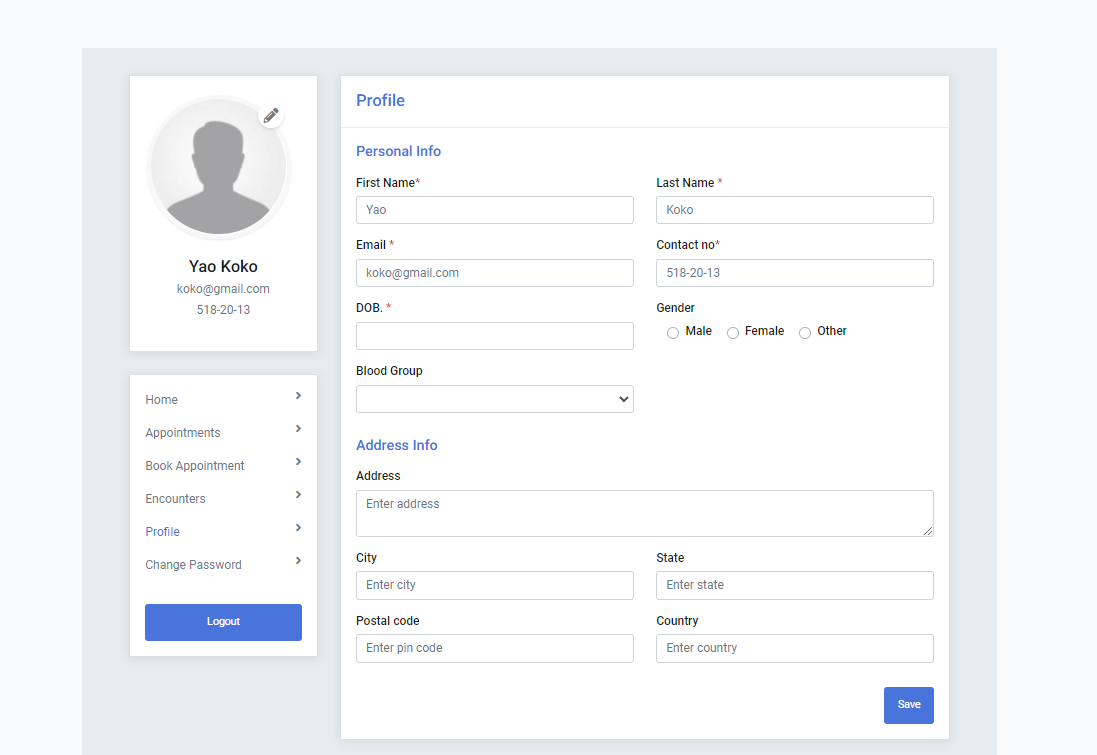


Рисунок 24- Вкладка Profile

# **Заключение**

Результатом выпускной квалификационной работы является готовый веб-сайт поликлиники.

Были решены следующие частные задачи:

· ознакомление с современными Интернет - технологиями и их использование в настоящей разработке;

· изучение программного инструментария, применяемого для разработки и создания Web-сайтов программы Wordpress;

· ознакомление со структурой поликлиники;

· определение структуры Web-страниц;

· изучены методы размещения сайтов в Интернете

Полученный в итоге веб-сайт может полностью удовлетворить все требования заказчика.

# **Список использованной литературы**

1. Голомбински К. Добавь воздуха! Основы визуального дизайна для

графики, веб и мультимедиа [Текст] / К. Голомбински, Р. Хаген. — СанктПетербург: Питер, 2013. — 272 с.

2. Городулин В. Основы CSS [Электронный ресурс]. — Режим доступа: http://css.manual.ru/articles/css\_basics (дата обращения: 20.11.2017).

3. Демченко Д. Обзор систем управления сайтом (CMS) [Электронный

ресурс]. — Режим доступа: http://www.white-windows.ru/obzor-sistemupravleniya-sajtom-cms/ (дата обращения: 18.01.2018).

4. Дакетт Джон HTML и CSS. Разработка и дизайн веб-сайтов [Текст] /

Джон Дакетт. — Москва: ЭКСМО, 2013. — 480 с.

5. Для тех, кто делает сайты [Электронный ресурс]. — Режим доступа:

http://htmlbook.ru (дата обращения: 12.03.2017).

6. Дунаев В. В. Основы WEB-дизайна [Текст]: самоучитель /

В. В. Дунае. — 2-е издание, перераб. и доп. — Санкт-Петербург: Питер, 2012.

— 480 с.

7. Как создать сайт [Электронный ресурс]. — Режим доступа:

http://myrusakov.ru (дата обращения: 20.12.2017).

8. Квентор И. Адаптивная верстка сайтов [Электронный ресурс]. —

Режим доступа: http://www.websovet.com/adaptive-layout-introduction (дата

обращения: 19.11.2017).

9. Лоусон Б. Изучаем HTML5. Библиотека специалиста [Текст] /

Б. Лоусон, Р. Шарп. — Санкт-Петербург: Питер, 2013. — 286 с.

10. Макфарланд Д. Большая книга CSS [Текст] / Д. Макфарланд. — 3-е

изд. — Санкт-Петербург: Питер, 2014. — 608 с.

11. Маркотт И. Отзывчивый веб-дизайн [Текст] / И. Маркотт. — Книга 1. — пер. Миронов П. — Москва: Манн, Иванов и Фербер, 2012. — 231 с.

12.Чебыкин Р. Разработка и оформление текстового содержания сайтов. — Санкт-Петербург: БХВ-Петербург, 2014. — 528 c.

13.Сарычев А. Идеально! Как создать и переделать свой сайт. Правильный подход и передовые техники разработки [Текст] / А. Сарычев. —

Москва: СилаУма, 2013. — 369 с.

14.Сидерхолм Д. CSS3 для веб-дизайнеров [Текст] / Д. Сидерхолм.

Книга 5. — пер. Кудашев E. — Москва: Манн, Иванов и Фербер. 2013. —

273 с.

# **Приложение**

Компетенции

**Приложение А**

if (!defined('KB\_IN\_BYTES')) { define('KB\_IN\_BYTES', 1024); }

if (!defined('MB\_IN\_BYTES')) { define('MB\_IN\_BYTES', 1024 \* KB\_IN\_BYTES); }

if (!defined('GB\_IN\_BYTES')) { define('GB\_IN\_BYTES', 1024 \* MB\_IN\_BYTES); }

if (!defined('DUPLICATOR\_PHP\_MAX\_MEMORY')) { define('DUPLICATOR\_PHP\_MAX\_MEMORY', 4096 \* MB\_IN\_BYTES); }

date\_default\_timezone\_set('UTC'); // Some machines don’t have this set so just do it here.

@ignore\_user\_abort(true);

if (!function\_exists('wp\_is\_ini\_value\_changeable')) {

/\*\*

\* Determines whether a PHP ini value is changeable at runtime.

\*

\* @staticvar array $ini\_all

\*

\* @link https://secure.php.net/manual/en/function.ini-get-all.php

\*

\* @param string $setting The name of the ini setting to check.

\* @return bool True if the value is changeable at runtime. False otherwise.

\*/

function wp\_is\_ini\_value\_changeable( $setting ) {

static $ini\_all;

if ( ! isset( $ini\_all ) ) {

$ini\_all = false;

// Sometimes `ini\_get\_all()` is disabled via the `disable\_functions` option for "security purposes".

if ( function\_exists( 'ini\_get\_all' ) ) {

$ini\_all = ini\_get\_all();

}

}

// Bit operator to workaround https://bugs.php.net/bug.php?id=44936 which changes access level to 63 in PHP 5.2.6 - 5.2.17.

if ( isset( $ini\_all[ $setting ]['access'] ) && ( INI\_ALL === ( $ini\_all[ $setting ]['access'] & 7 ) || INI\_USER === ( $ini\_all[ $setting ]['access'] & 7 ) ) ) {

return true;

}

// If we were unable to retrieve the details, fail gracefully to assume it's changeable.

if ( ! is\_array( $ini\_all ) ) {

return true;

}

return false;

}

}

@set\_time\_limit(3600);

if (wp\_is\_ini\_value\_changeable('memory\_limit'))

@ini\_set('memory\_limit', DUPLICATOR\_PHP\_MAX\_MEMORY);

if (wp\_is\_ini\_value\_changeable('max\_input\_time'))

@ini\_set('max\_input\_time', '-1');

if (wp\_is\_ini\_value\_changeable('pcre.backtrack\_limit'))

@ini\_set('pcre.backtrack\_limit', PHP\_INT\_MAX);

if (wp\_is\_ini\_value\_changeable('default\_socket\_timeout'))

@ini\_set('default\_socket\_timeout', 3600);

DUPX\_Handler::init\_error\_handler();

/\*\*

\* Bootstrap utility to exatract the core installer

\*

\* Standard: PSR-2

\*

\* @package SC\DUPX\Bootstrap

\* @link http://www.php-fig.org/psr/psr-2/

\*

\* To force extraction mode:

\* installer.php?unzipmode=auto

\* installer.php?unzipmode=ziparchive

\* installer.php?unzipmode=shellexec

\*/

/\*\*\* CLASS DEFINITION START \*\*\*/

abstract class DUPX\_Bootstrap\_Zip\_Mode

{

const AutoUnzip = 0;

const ZipArchive = 1;

const ShellExec = 2;

}

class DUPX\_Bootstrap

{

//@@ Params get dynamically swapped when package is built

const ARCHIVE\_FILENAME = '20220407\_kokodelapaix\_6aadd7dfee5437343644\_20220420094218\_archive.zip';

const ARCHIVE\_SIZE = '56847132';

const INSTALLER\_DIR\_NAME = 'dup-installer';

const PACKAGE\_HASH = '6aadd7d-20094218';

const SECONDARY\_PACKAGE\_HASH = 'c0f5587-20094218';

const VERSION = '1.4.4';

public $hasZipArchive = false;

public $hasShellExecUnzip = false;

public $mainInstallerURL;

public $installerContentsPath;

public $installerExtractPath;

public $archiveExpectedSize = 0;

public $archiveActualSize = 0;

public $activeRatio = 0;

/\*\*

\* Instantiate the Bootstrap Object

\*

\* @return null

\*/

public function \_\_construct()

{

// clean log file

self::log('', true);

//ARCHIVE\_SIZE will be blank with a root filter so we can estimate

//the default size of the package around 17.5MB (18088000)

$archiveActualSize = @file\_exists(self::ARCHIVE\_FILENAME) ? @filesize(self::ARCHIVE\_FILENAME) : false;

$archiveActualSize = ($archiveActualSize !== false) ? $archiveActualSize : 0;

$this->hasZipArchive = class\_exists('ZipArchive');

$this->hasShellExecUnzip = $this->getUnzipFilePath() != null ? true : false;

$this->installerContentsPath = str\_replace("\\", '/', (dirname(\_\_FILE\_\_). '/' .self::INSTALLER\_DIR\_NAME));

$this->installerExtractPath = str\_replace("\\", '/', (dirname(\_\_FILE\_\_)));

$this->archiveExpectedSize = strlen(self::ARCHIVE\_SIZE) ? self::ARCHIVE\_SIZE : 0 ;

$this->archiveActualSize = $archiveActualSize;

if($this->archiveExpectedSize > 0) {

$this->archiveRatio = (((1.0) \* $this->archiveActualSize) / $this->archiveExpectedSize) \* 100;

} else {

$this->archiveRatio = 100;

}

$this->overwriteMode = (isset($\_GET['mode']) && ($\_GET['mode'] == 'overwrite'));

}

/\*\*

\* Run the bootstrap process which includes checking for requirements and running

\* the extraction process

\*

\* @return null | string Returns null if the run was successful otherwise an error message

\*/

public function run()

{

date\_default\_timezone\_set('UTC'); // Some machines don't have this set so just do it here

self::log('==DUPLICATOR INSTALLER BOOTSTRAP v1.4.4==');

self::log('----------------------------------------------------');

self::log('Installer bootstrap start');

$archive\_filepath = $this->getArchiveFilePath();

$archive\_filename = self::ARCHIVE\_FILENAME;

$error = null;

$is\_installer\_file\_valid = true;

if (preg\_match('/\_([a-z0-9]{7})[a-z0-9]+\_[0-9]{6}([0-9]{8})\_archive.(?:zip|daf)$/', $archive\_filename, $matches)) {

$expected\_package\_hash = $matches[1].'-'.$matches[2];

if (self::PACKAGE\_HASH != $expected\_package\_hash) {

$is\_installer\_file\_valid = false;

self::log("[ERROR] Installer and archive mismatch detected.");

}

} else {

self::log("[ERROR] Invalid archive file name.");

$is\_installer\_file\_valid = false;

}

if (false === $is\_installer\_file\_valid) {

$error = "Installer and archive mismatch detected.

Ensure uncorrupted installer and matching archive are present.";

return $error;

}

$extract\_installer = true;

$installer\_directory = dirname(\_\_FILE\_\_).'/'.self::INSTALLER\_DIR\_NAME;

$extract\_success = false;

$archiveExpectedEasy = $this->readableByteSize($this->archiveExpectedSize);

$archiveActualEasy = $this->readableByteSize($this->archiveActualSize);

//$archive\_extension = strtolower(pathinfo($archive\_filepath)['extension']);

$archive\_extension = strtolower(pathinfo($archive\_filepath, PATHINFO\_EXTENSION));

$manual\_extract\_found = (

file\_exists($installer\_directory."/main.installer.php")

&&

file\_exists($installer\_directory."/dup-archive\_\_".self::PACKAGE\_HASH.".txt")

&&

file\_exists($installer\_directory."/dup-database\_\_".self::PACKAGE\_HASH.".sql")

);

$isZip = ($archive\_extension == 'zip');

//MANUAL EXTRACTION NOT FOUND

if (! $manual\_extract\_found) {

//MISSING ARCHIVE FILE

if (! file\_exists($archive\_filepath)) {

self::log("[ERROR] Archive file not found!");

$error = "<style>.diff-list font { font-weight: bold; }</style>"

. "<b>Archive not found!</b> The required archive file must be present in the <i>'Extraction Path'</i> below. When the archive file name was created "

. "it was given a secure hashed file name. This file name must be the <i>exact same</i> name as when it was created character for character. "

. "Each archive file has a unique installer associated with it and must be used together. See the list below for more options:<br/>"

. "<ul>"

. "<li>If the archive is not finished downloading please wait for it to complete.</li>"

. "<li>Rename the file to it original hash name. See WordPress-Admin ❯ Packages ❯ Details. </li>"

. "<li>When downloading, both files both should be from the same package line. </li>"

. "<li>Also see: <a href='https://snapcreek.com/duplicator/docs/faqs-tech/#faq-installer-050-q' target='\_blank'>How to fix various errors that show up before step-1 of the installer?</a></li>"

. "</ul><br/>"

."<b>Extraction Path:</b> <span class='file-info'>{$this->installerExtractPath}/</span><br/>";

return $error;

}

// Sometimes the self::ARCHIVE\_SIZE is ''.

$archive\_size = self::ARCHIVE\_SIZE;

if (!empty($archive\_size) && !self::checkInputVaslidInt($archive\_size)) {

$no\_of\_bits = PHP\_INT\_SIZE \* 8;

$error = 'Current is a '.$no\_of\_bits.'-bit SO. This archive is too large for '.$no\_of\_bits.'-bit PHP.'.'<br>';

self::log('[ERROR] '.$error);

$error .= 'Possibibles solutions:<br>';

$error .= '- Use the file filters to get your package lower to support this server or try the package on a Linux server.'.'<br>';

$error .= '- Perform a <a target="\_blank" href="https://snapcreek.com/duplicator/docs/faqs-tech/#faq-installer-015-q">Manual Extract Install</a>'.'<br>';

switch ($no\_of\_bits == 32) {

case 32:

$error .= '- Ask your host to upgrade the server to 64-bit PHP or install on another system has 64-bit PHP'.'<br>';

break;

case 64:

$error .= '- Ask your host to upgrade the server to 128-bit PHP or install on another system has 128-bit PHP'.'<br>';

break;

}

if (self::isWindows()) {

$error .= '- <a target="\_blank" href="https://snapcreek.com/duplicator/docs/faqs-tech/#faq-trouble-052-q">Windows DupArchive extractor</a> to extract all files from the archive.'.'<br>';

}

return $error;

}

//SIZE CHECK ERROR

if (($this->archiveRatio < 90) && ($this->archiveActualSize > 0) && ($this->archiveExpectedSize > 0)) {

self::log("ERROR: The expected archive size should be around [{$archiveExpectedEasy}]. The actual size is currently [{$archiveActualEasy}].");

self::log("ERROR: The archive file may not have fully been downloaded to the server");

$percent = round($this->archiveRatio);

$autochecked = isset($\_POST['auto-fresh']) ? "checked='true'" : '';

$error = "<b>Archive file size warning.</b><br/> The expected archive size should be around <b class='pass'>[{$archiveExpectedEasy}]</b>. "

. "The actual size is currently <b class='fail'>[{$archiveActualEasy}]</b>. The archive file may not have fully been downloaded to the server. "

. "Please validate that the file sizes are close to the same size and that the file has been completely downloaded to the destination server. If the archive is still "

. "downloading then refresh this page to get an update on the download size.<br/><br/>";

return $error;

}

}

// OLD COMPATIBILITY MODE

if (isset($\_GET['extract-installer']) && !isset($\_GET['force-extract-installer'])) {

$\_GET['force-extract-installer'] = $\_GET['extract-installer'];

}

if ($manual\_extract\_found) {

// INSTALL DIRECTORY: Check if its setup correctly AND we are not in overwrite mode

if (isset($\_GET['force-extract-installer']) && ('1' == $\_GET['force-extract-installer'] || 'enable' == $\_GET['force-extract-installer'] || 'false' == $\_GET['force-extract-installer'])) {

self::log("Manual extract found with force extract installer get parametr");

$extract\_installer = true;

} else {

$extract\_installer = false;

self::log("Manual extract found so not going to extract dup-installer dir");

}

} else {

$extract\_installer = true;

}

if ($extract\_installer && file\_exists($installer\_directory)) {

self::log("EXTRACT dup-installer dir");

$scanned\_directory = array\_diff(scandir($installer\_directory), array('..', '.'));

foreach ($scanned\_directory as $object) {

$object\_file\_path = $installer\_directory.'/'.$object;

if (is\_file($object\_file\_path)) {

if (unlink($object\_file\_path)) {

self::log('Successfully deleted the file '.$object\_file\_path);

} else {

$error .= '[ERROR] Error deleting the file '.$object\_file\_path.' Please manually delete it and try again.';

self::log($error);

}

}

}

}

//ATTEMPT EXTRACTION:

//ZipArchive and Shell Exec

if ($extract\_installer) {

self::log("Ready to extract the installer");

self::log("Checking permission of destination folder");

$destination = dirname(\_\_FILE\_\_);

if (!is\_writable($destination)) {

self::log("destination folder for extraction is not writable");

if (self::chmod($destination, 'u+rwx')) {

self::log("Permission of destination folder changed to u+rwx");

} else {

self::log("[ERROR] Permission of destination folder failed to change to u+rwx");

}

}

if (!is\_writable($destination)) {

self::log("WARNING: The {$destination} directory is not writable.");

$error = "NOTICE: The {$destination} directory is not writable on this server please talk to your host or server admin about making ";

$error .= "<a target='\_blank' href='https://snapcreek.com/duplicator/docs/faqs-tech/#faq-trouble-055-q'>writable {$destination} directory</a> on this server. <br/>";

return $error;

}

if ($isZip) {

$zip\_mode = $this->getZipMode();

if (($zip\_mode == DUPX\_Bootstrap\_Zip\_Mode::AutoUnzip) || ($zip\_mode == DUPX\_Bootstrap\_Zip\_Mode::ZipArchive) && class\_exists('ZipArchive')) {

if ($this->hasZipArchive) {

self::log("ZipArchive exists so using that");

$extract\_success = $this->extractInstallerZipArchive($archive\_filepath);

if ($extract\_success) {

self::log('Successfully extracted with ZipArchive');

} else {

if (0 == $this->installer\_files\_found) {

$error = "[ERROR] This archive is not properly formatted and does not contain a dup-installer directory. Please make sure you are attempting to install the original archive and not one that has been reconstructed.";

self::log($error);

return $error;

} else {

$error = '[ERROR] Error extracting with ZipArchive. ';

self::log($error);

}

}

} else {

self::log("WARNING: ZipArchive is not enabled.");

$error = "NOTICE: ZipArchive is not enabled on this server please talk to your host or server admin about enabling ";

$error .= "<a target='\_blank' href='https://snapcreek.com/duplicator/docs/faqs-tech/#faq-trouble-060-q'>ZipArchive</a> on this server. <br/>";

}

}

if (!$extract\_success) {

if (($zip\_mode == DUPX\_Bootstrap\_Zip\_Mode::AutoUnzip) || ($zip\_mode == DUPX\_Bootstrap\_Zip\_Mode::ShellExec)) {

$unzip\_filepath = $this->getUnzipFilePath();

if ($unzip\_filepath != null) {

$extract\_success = $this->extractInstallerShellexec($archive\_filepath);

if ($extract\_success) {

self::log('Successfully extracted with Shell Exec');

$error = null;

} else {

$error .= '[ERROR] Error extracting with Shell Exec. Please manually extract archive then choose Advanced > Manual Extract in installer.';

self::log($error);

}

} else {

self::log('WARNING: Shell Exec Zip is not available');

$error .= "NOTICE: Shell Exec is not enabled on this server please talk to your host or server admin about enabling ";

$error .= "<a target='\_blank' href='http://php.net/manual/en/function.shell-exec.php'>Shell Exec</a> on this server or manually extract archive then choose Advanced > Manual Extract in installer.";

}

}

}

// If both ZipArchive and ShellZip are not available, Error message should be combined for both

if (!$extract\_success && $zip\_mode == DUPX\_Bootstrap\_Zip\_Mode::AutoUnzip) {

$unzip\_filepath = $this->getUnzipFilePath();

if (!class\_exists('ZipArchive') && empty($unzip\_filepath)) {

self::log("WARNING: ZipArchive and Shell Exec are not enabled on this server.");

$error = "NOTICE: ZipArchive and Shell Exec are not enabled on this server please talk to your host or server admin about enabling ";

$error .= "<a target='\_blank' href='https://snapcreek.com/duplicator/docs/faqs-tech/#faq-trouble-060-q'>ZipArchive</a> or <a target='\_blank' href='http://php.net/manual/en/function.shell-exec.php'>Shell Exec</a> on this server or manually extract archive then choose Advanced > Manual Extract in installer.";

}

}

} else {

DupArchiveMiniExpander::init("DUPX\_Bootstrap::log");

try {

DupArchiveMiniExpander::expandDirectory($archive\_filepath, self::INSTALLER\_DIR\_NAME, dirname(\_\_FILE\_\_));

} catch (Exception $ex) {

self::log("[ERROR] Error expanding installer subdirectory:".$ex->getMessage());

throw $ex;

}

}

$is\_apache = (strpos($\_SERVER['SERVER\_SOFTWARE'], 'Apache') !== false || strpos($\_SERVER['SERVER\_SOFTWARE'], 'LiteSpeed') !== false);

$is\_nginx = (strpos($\_SERVER['SERVER\_SOFTWARE'], 'nginx') !== false);

$sapi\_type = php\_sapi\_name();

$php\_ini\_data = array(

'max\_execution\_time' => 3600,

'max\_input\_time' => -1,

'ignore\_user\_abort' => 'On',

'post\_max\_size' => '4096M',

'upload\_max\_filesize' => '4096M',

'memory\_limit' => DUPLICATOR\_PHP\_MAX\_MEMORY,

'default\_socket\_timeout' => 3600,

'pcre.backtrack\_limit' => 99999999999,

);

$sapi\_type\_first\_three\_chars = substr($sapi\_type, 0, 3);

if ('fpm' === $sapi\_type\_first\_three\_chars) {

self::log("SAPI: FPM");

if ($is\_apache) {

self::log('Server: Apache');

} elseif ($is\_nginx) {

self::log('Server: Nginx');

}

if (($is\_apache && function\_exists('apache\_get\_modules') && in\_array('mod\_rewrite', apache\_get\_modules())) || $is\_nginx) {

$htaccess\_data = array();

foreach ($php\_ini\_data as $php\_ini\_key=>$php\_ini\_val) {

if ($is\_apache) {

$htaccess\_data[] = 'SetEnv PHP\_VALUE "'.$php\_ini\_key.' = '.$php\_ini\_val.'"';

} elseif ($is\_nginx) {

if ('On' == $php\_ini\_val || 'Off' == $php\_ini\_val) {

$htaccess\_data[] = 'php\_flag '.$php\_ini\_key.' '.$php\_ini\_val;

} else {

$htaccess\_data[] = 'php\_value '.$php\_ini\_key.' '.$php\_ini\_val;

}

}

}

$htaccess\_text = implode("\n", $htaccess\_data);

$htaccess\_file\_path = dirname(\_\_FILE\_\_).'/dup-installer/.htaccess';

self::log("creating {$htaccess\_file\_path} with the content:");

self::log($htaccess\_text);

@file\_put\_contents($htaccess\_file\_path, $htaccess\_text);

}

} elseif ('cgi' === $sapi\_type\_first\_three\_chars || 'litespeed' === $sapi\_type) {

if ('cgi' === $sapi\_type\_first\_three\_chars) {

self::log("SAPI: CGI");

} else {

self::log("SAPI: litespeed");

}

if (version\_compare(phpversion(), 5.5) >= 0 && (!$is\_apache || 'litespeed' === $sapi\_type)) {

$ini\_data = array();

foreach ($php\_ini\_data as $php\_ini\_key=>$php\_ini\_val) {

$ini\_data[] = $php\_ini\_key.' = '.$php\_ini\_val;

}

$ini\_text = implode("\n", $ini\_data);

$ini\_file\_path = dirname(\_\_FILE\_\_).'/dup-installer/.user.ini';

self::log("creating {$ini\_file\_path} with the content:");

self::log($ini\_text);

@file\_put\_contents($ini\_file\_path, $ini\_text);

} else{

self::log("No need to create dup-installer/.htaccess or dup-installer/.user.ini");

}

} else {

self::log("No need to create dup-installer/.htaccess or dup-installer/.user.ini");

self::log("ERROR: SAPI: Unrecognized");

}

} else {

self::log("ERROR: Didn't need to extract the installer.");

}

if (empty($error)) {

$config\_files = glob('./dup-installer/dup-archive\_\_\*.txt');

$config\_file\_absolute\_path = array\_pop($config\_files);

if (!file\_exists($config\_file\_absolute\_path)) {

$error = '<b>Archive config file not found in dup-installer folder.</b> <br><br>';

return $error;

}

}

$is\_https = $this->isHttps();

if($is\_https) {

$current\_url = 'https://';

} else {

$current\_url = 'http://';

}

if(($\_SERVER['SERVER\_PORT'] == 80) && ($is\_https)) {

// Fixing what appears to be a bad server setting

$server\_port = 443;

} else {

$server\_port = $\_SERVER['SERVER\_PORT'];

}

// for ngrok url and Local by Flywheel Live URL

if (isset($\_SERVER['HTTP\_X\_ORIGINAL\_HOST'])) {

$host = $\_SERVER['HTTP\_X\_ORIGINAL\_HOST'];

} else {

$host = isset($\_SERVER['HTTP\_HOST']) ? $\_SERVER['HTTP\_HOST'] : $\_SERVER['SERVER\_NAME'];//WAS SERVER\_NAME and caused problems on some boxes

}

$current\_url .= $host;

if(strpos($current\_url,':') === false) {

$current\_url = $current\_url.':'.$server\_port;

}

$current\_url .= $\_SERVER['REQUEST\_URI'];

$uri\_start = dirname($current\_url);

$encoded\_archive\_path = urlencode($archive\_filepath);

if ($error === null) {

$error = $this->postExtractProcessing();

if($error == null) {

$bootloader\_name = basename(\_\_FILE\_\_);

$this->mainInstallerURL = $uri\_start.'/'.self::INSTALLER\_DIR\_NAME.'/main.installer.php';

$this->fixInstallerPerms($this->mainInstallerURL);

$this->archive = $archive\_filepath;

$this->bootloader = $bootloader\_name;

if (isset($\_SERVER['QUERY\_STRING']) && !empty($\_SERVER['QUERY\_STRING'])) {

$this->mainInstallerURL .= '?'.$\_SERVER['QUERY\_STRING'];

}

self::log("DONE: No detected errors so redirecting to the main installer. Main Installer URI = {$this->mainInstallerURL}");

}

}

return $error;

}

public function postExtractProcessing()

{

$dproInstallerDir = dirname(\_\_FILE\_\_) . '/dup-installer';

$libDir = $dproInstallerDir . '/lib';

$fileopsDir = $libDir . '/fileops';

if(!file\_exists($dproInstallerDir)) {

return 'Can\'t extract installer directory. See <a target="\_blank" href="https://snapcreek.com/duplicator/docs/faqs-tech/#faq-installer-022-q">this FAQ item</a> for details on how to resolve.</a>';

}

$sourceFilepath = "{$fileopsDir}/fileops.ppp";

$destFilepath = "{$fileopsDir}/fileops.php";

if(file\_exists($sourceFilepath) && (!file\_exists($destFilepath))) {

if(@rename($sourceFilepath, $destFilepath) === false) {

return "Error renaming {$sourceFilepath}";

}

}

}

/\*\*

\* Indicates if site is running https or not

\*

\* @return bool Returns true if https, false if not

\*/

public function isHttps()

{

$retVal = true;

if (isset($\_SERVER['HTTP\_X\_FORWARDED\_PROTO']) && $\_SERVER['HTTP\_X\_FORWARDED\_PROTO'] === 'https') {

$\_SERVER ['HTTPS'] = 'on';

}

if (isset($\_SERVER['HTTPS'])) {

$retVal = ($\_SERVER['HTTPS'] !== 'off');

} else {

$retVal = ($\_SERVER['SERVER\_PORT'] == 443);

}

return $retVal;

}

/\*\*

\* Fetches current URL via php

\*

\* @param bool $queryString If true the query string will also be returned.

\* @param int $getParentDirLevel if 0 get current script name or parent folder, if 1 parent folder if 2 parent of parent folder ...

\*

\* @returns The current page url

\*/

public static function getCurrentUrl($queryString = true, $requestUri = false, $getParentDirLevel = 0)

{

// \*\*\* HOST

if (isset($\_SERVER['HTTP\_X\_ORIGINAL\_HOST'])) {

$host = $\_SERVER['HTTP\_X\_ORIGINAL\_HOST'];

} else {

$host = isset($\_SERVER['HTTP\_HOST']) ? $\_SERVER['HTTP\_HOST'] : $\_SERVER['SERVER\_NAME']; //WAS SERVER\_NAME and caused problems on some boxes

}

// \*\*\* PROTOCOL

if (isset($\_SERVER['HTTP\_X\_FORWARDED\_PROTO']) && $\_SERVER['HTTP\_X\_FORWARDED\_PROTO'] === 'https') {

$\_SERVER ['HTTPS'] = 'on';

}

if (isset($\_SERVER['HTTP\_X\_FORWARDED\_SSL']) && $\_SERVER['HTTP\_X\_FORWARDED\_SSL'] === 'https') {

$\_SERVER ['HTTPS'] = 'on';

}

if (isset($\_SERVER['HTTP\_CF\_VISITOR'])) {

$visitor = json\_decode($\_SERVER['HTTP\_CF\_VISITOR']);

if ($visitor->scheme == 'https') {

$\_SERVER ['HTTPS'] = 'on';

}

}

$protocol = 'http'.((isset($\_SERVER['HTTPS']) && strtolower($\_SERVER['HTTPS']) === 'on') ? 's' : '');

if ($requestUri) {

$serverUrlSelf = preg\_replace('/\?.\*$/', '', $\_SERVER['REQUEST\_URI']);

} else {

// \*\*\* SCRIPT NAME

$serverUrlSelf = $\_SERVER['SCRIPT\_NAME'];

for ($i = 0; $i < $getParentDirLevel; $i++) {

$serverUrlSelf = preg\_match('/^[\\\\\/]?$/', dirname($serverUrlSelf)) ? '' : dirname($serverUrlSelf);

}

}

// \*\*\* QUERY STRING

$query = ($queryString && isset($\_SERVER['QUERY\_STRING']) && strlen($\_SERVER['QUERY\_STRING']) > 0 ) ? '?'.$\_SERVER['QUERY\_STRING'] : '';

return $protocol.'://'.$host.$serverUrlSelf.$query;

}

/\*\*

\* Attempts to set the 'dup-installer' directory permissions

\*

\* @return null

\*/

private function fixInstallerPerms()

{

$file\_perms = 'u+rw';

$dir\_perms = 'u+rwx';

$installer\_dir\_path = $this->installerContentsPath;

$this->setPerms($installer\_dir\_path, $dir\_perms, false);

$this->setPerms($installer\_dir\_path, $file\_perms, true);

}

/\*\*

\* Set the permissions of a given directory and optionally all files

\*

\* @param string $directory The full path to the directory where perms will be set

\* @param string $perms The given permission sets to use such as '0755' or 'u+rw'

\* @param string $do\_files Also set the permissions of all the files in the directory

\*

\* @return null

\*/

private function setPerms($directory, $perms, $do\_files)

{

if (!$do\_files) {

// If setting a directory hiearchy be sure to include the base directory

$this->setPermsOnItem($directory, $perms);

}

$item\_names = array\_diff(scandir($directory), array('.', '..'));

foreach ($item\_names as $item\_name) {

$path = "$directory/$item\_name";

if (($do\_files && is\_file($path)) || (!$do\_files && !is\_file($path))) {

$this->setPermsOnItem($path, $perms);

}

}

}

/\*\*

\* Set the permissions of a single directory or file

\*

\* @param string $path The full path to the directory or file where perms will be set

\* @param string $perms The given permission sets to use such as '0755' or 'u+rw'

\*

\* @return bool Returns true if the permission was properly set

\*/

private function setPermsOnItem($path, $perms)

{

if (($result = self::chmod($path, $perms)) === false) {

self::log("ERROR: Couldn't set permissions of $path<br/>");

} else {

self::log("Set permissions of $path<br/>");

}

return $result;

}

/\*\*

\* Compare two strings and return html text which represts diff

\*

\* @param string $oldString

\* @param string $newString

\*

\* @return string Returns html text

\*/

private function compareStrings($oldString, $newString) {

$ret = '';

for($i=0; isset($oldString[$i]) || isset($newString[$i]); $i++) {

if(!isset($oldString[$i])) {

$ret .= '<font color="red">' . $newString[$i] . '</font>';

continue;

}

for($char=0; isset($oldString[$i][$char]) || isset($newString[$i][$char]); $char++) {

if(!isset($oldString[$i][$char])) {

$ret .= '<font color="red">' . substr($newString[$i], $char) . '</font>';

break;

} elseif(!isset($newString[$i][$char])) {

break;

}

if(ord($oldString[$i][$char]) != ord($newString[$i][$char]))

$ret .= '<font color="red">' . $newString[$i][$char] . '</font>';

else

$ret .= $newString[$i][$char];

}

}

return $ret;

}

/\*\*

\* Logs a string to the dup-installer-bootlog\_\_[HASH].txt file

\*

\* @param string $s The string to log to the log file

\*

\* @return boog|int // This function returns the number of bytes that were written to the file, or FALSE on failure.

\*/

public static function log($s, $deleteOld = false)

{

static $logfile = null;

if (is\_null($logfile)) {

$logfile = self::getBootLogFilePath();

}

if ($deleteOld && file\_exists($logfile)) {

@unlink($logfile);

}

$timestamp = date('M j H:i:s');

return @file\_put\_contents($logfile, '['.$timestamp.'] '.self::postprocessLog($s)."\n", FILE\_APPEND);

}

/\*\*

\* get boot log file name the dup-installer-bootlog\_\_[HASH].txt file

\*

\* @return string

\*/

public static function getBootLogFilePath() {

return dirname(\_\_FILE\_\_).'/dup-installer-bootlog\_\_'.self::SECONDARY\_PACKAGE\_HASH.'.txt';

}

protected static function postprocessLog($str) {

return str\_replace(array(

self::getArchiveFileHash(),

self::PACKAGE\_HASH,

self::SECONDARY\_PACKAGE\_HASH

), '[HASH]' , $str);

}

public static function getArchiveFileHash()

{

static $fileHash = null;

if (is\_null($fileHash)) {

$fileHash = preg\_replace('/^.+\_([a-z0-9]+)\_[0-9]{14}\_archive\.(?:daf|zip)$/', '$1', self::ARCHIVE\_FILENAME);

}

return $fileHash;

}

/\*\*

\* Extracts only the 'dup-installer' files using ZipArchive

\*

\* @param string $archive\_filepath The path to the archive file.

\*

\* @return bool Returns true if the data was properly extracted

\*/

private function extractInstallerZipArchive($archive\_filepath, $checkSubFolder = false)

{

$success = true;

$zipArchive = new ZipArchive();

$subFolderArchiveList = array();

if (($zipOpenRes = $zipArchive->open($archive\_filepath)) === true) {

self::log("Successfully opened archive file.");

$destination = dirname(\_\_FILE\_\_);

$folder\_prefix = self::INSTALLER\_DIR\_NAME.'/';

self::log("Extracting all files from archive within ".self::INSTALLER\_DIR\_NAME);

$this->installer\_files\_found = 0;

for ($i = 0; $i < $zipArchive->numFiles; $i++) {

$stat = $zipArchive->statIndex($i);

if ($checkSubFolder == false) {

$filenameCheck = $stat['name'];

$filename = $stat['name'];

$tmpSubFolder = null;

} else {

$safePath = rtrim(self::setSafePath($stat['name']) , '/');

$tmpArray = explode('/' , $safePath);

if (count($tmpArray) < 2) {

continue;

}

$tmpSubFolder = $tmpArray[0];

array\_shift($tmpArray);

$filenameCheck = implode('/' , $tmpArray);

$filename = $stat['name'];

}

if ($this->startsWith($filenameCheck , $folder\_prefix)) {

$this->installer\_files\_found++;

if (!empty($tmpSubFolder) && !in\_array($tmpSubFolder , $subFolderArchiveList)) {

$subFolderArchiveList[] = $tmpSubFolder;

}

if ($zipArchive->extractTo($destination, $filename) === true) {

self::log("Success: {$filename} >>> {$destination}");

} else {

self::log("[ERROR] Error extracting {$filename} from archive archive file");

$success = false;

break;

}

}

}

if ($checkSubFolder && count($subFolderArchiveList) !== 1) {

self::log("Error: Multiple dup subfolder archive");

$success = false;

} else {

if ($checkSubFolder) {

$this->moveUpfromSubFolder(dirname(\_\_FILE\_\_).'/'.$subFolderArchiveList[0] , true);

}

$lib\_directory = dirname(\_\_FILE\_\_).'/'.self::INSTALLER\_DIR\_NAME.'/lib';

$snaplib\_directory = $lib\_directory.'/snaplib';

// If snaplib files aren't present attempt to extract and copy those

if(!file\_exists($snaplib\_directory))

{

$folder\_prefix = 'snaplib/';

$destination = $lib\_directory;

for ($i = 0; $i < $zipArchive->numFiles; $i++) {

$stat = $zipArchive->statIndex($i);

$filename = $stat['name'];

if ($this->startsWith($filename, $folder\_prefix)) {

$this->installer\_files\_found++;

if ($zipArchive->extractTo($destination, $filename) === true) {

self::log("Success: {$filename} >>> {$destination}");

} else {

self::log("[ERROR] Error extracting {$filename} from archive archive file");

$success = false;

break;

}

}

}

}

}

if ($zipArchive->close() === true) {

self::log("Successfully closed archive file");

} else {

self::log("[ERROR] Problem closing archive file");

$success = false;

}

if ($success != false && $this->installer\_files\_found < 10) {

if ($checkSubFolder) {

self::log("[ERROR] Couldn't find the installer directory in the archive!");

$success = false;

} else {

self::log("[ERROR] Couldn't find the installer directory in archive root! Check subfolder");

$this->extractInstallerZipArchive($archive\_filepath, true);

}

}

} else {

self::log("[ERROR] Couldn't open archive archive file with ZipArchive CODE[".$zipOpenRes."]");

$success = false;

}

return $success;

}

/\*\*

\* return true if current SO is windows

\*

\* @staticvar bool $isWindows

\* @return bool

\*/

public static function isWindows()

{

static $isWindows = null;

if (is\_null($isWindows)) {

$isWindows = (strtoupper(substr(PHP\_OS, 0, 3)) === 'WIN');

}

return $isWindows;

}

/\*\*

\* return current SO path path len

\* @staticvar int $maxPath

\* @return int

\*/

public static function maxPathLen()

{

static $maxPath = null;

if (is\_null($maxPath)) {

if (defined('PHP\_MAXPATHLEN')) {

$maxPath = PHP\_MAXPATHLEN;

} else {

// for PHP < 5.3.0

$maxPath = self::isWindows() ? 260 : 4096;

}

}

return $maxPath;

}

/\*\*

\* this function make a chmod only if the are different from perms input and if chmod function is enabled

\*

\* this function handles the variable MODE in a way similar to the chmod of lunux

\* So the MODE variable can be

\* 1) an octal number (0755)

\* 2) a string that defines an octal number ("644")

\* 3) a string with the following format [ugoa]\*([-+=]([rwx]\*)+

\*

\* examples

\* u+rw add read and write at the user

\* u+rw,uo-wx add read and write ad the user and remove wx at groupd and other

\* a=rw is equal at 666

\* u=rwx,go-rwx is equal at 700

\*

\* @param string $file

\* @param int|string $mode

\* @return boolean

\*/

public static function chmod($file, $mode)

{

if (!file\_exists($file)) {

return false;

}

$octalMode = 0;

if (is\_int($mode)) {

$octalMode = $mode;

} else if (is\_string($mode)) {

$mode = trim($mode);

if (preg\_match('/([0-7]{1,3})/', $mode)) {

$octalMode = intval(('0'.$mode), 8);

} else if (preg\_match\_all('/(a|[ugo]{1,3})([-=+])([rwx]{1,3})/', $mode, $gMatch, PREG\_SET\_ORDER)) {

if (!function\_exists('fileperms')) {

return false;

}

// start by file permission

$octalMode = (fileperms($file) & 0777);

foreach ($gMatch as $matches) {

// [ugo] or a = ugo

$group = $matches[1];

if ($group === 'a') {

$group = 'ugo';

}

// can be + - =

$action = $matches[2];

// [rwx]

$gPerms = $matches[3];

// reset octal group perms

$octalGroupMode = 0;

// Init sub perms

$subPerm = 0;

$subPerm += strpos($gPerms, 'x') !== false ? 1 : 0; // mask 001

$subPerm += strpos($gPerms, 'w') !== false ? 2 : 0; // mask 010

$subPerm += strpos($gPerms, 'r') !== false ? 4 : 0; // mask 100

$ugoLen = strlen($group);

if ($action === '=') {

// generate octal group permsissions and ugo mask invert

$ugoMaskInvert = 0777;

for ($i = 0; $i < $ugoLen; $i++) {

switch ($group[$i]) {

case 'u':

$octalGroupMode = $octalGroupMode | $subPerm << 6; // mask xxx000000

$ugoMaskInvert = $ugoMaskInvert & 077;

break;

case 'g':

$octalGroupMode = $octalGroupMode | $subPerm << 3; // mask 000xxx000

$ugoMaskInvert = $ugoMaskInvert & 0707;

break;

case 'o':

$octalGroupMode = $octalGroupMode | $subPerm; // mask 000000xxx

$ugoMaskInvert = $ugoMaskInvert & 0770;

break;

}

}

// apply = action

$octalMode = $octalMode & ($ugoMaskInvert | $octalGroupMode);

} else {

// generate octal group permsissions

for ($i = 0; $i < $ugoLen; $i++) {

switch ($group[$i]) {

case 'u':

$octalGroupMode = $octalGroupMode | $subPerm << 6; // mask xxx000000

break;

case 'g':

$octalGroupMode = $octalGroupMode | $subPerm << 3; // mask 000xxx000

break;

case 'o':

$octalGroupMode = $octalGroupMode | $subPerm; // mask 000000xxx

break;

}

}

// apply + or - action

switch ($action) {

case '+':

$octalMode = $octalMode | $octalGroupMode;

break;

case '-':

$octalMode = $octalMode & ~$octalGroupMode;

break;

}

}

}

}

}

// if input permissions are equal at file permissions return true without performing chmod

if (function\_exists('fileperms') && $octalMode === (fileperms($file) & 0777)) {

return true;

}

if (!function\_exists('chmod')) {

return false;

}

return @chmod($file, $octalMode);

}

public static function checkInputVaslidInt($input) {

return (filter\_var($input, FILTER\_VALIDATE\_INT) === 0 || filter\_var($input, FILTER\_VALIDATE\_INT));

}

/\*\*

\* this function creates a folder if it does not exist and performs a chmod.

\* it is different from the normal mkdir function to which an umask is applied to the input permissions.

\*

\* this function handles the variable MODE in a way similar to the chmod of lunux

\* So the MODE variable can be

\* 1) an octal number (0755)

\* 2) a string that defines an octal number ("644")

\* 3) a string with the following format [ugoa]\*([-+=]([rwx]\*)+

\*

\* @param string $path

\* @param int|string $mode

\* @param bool $recursive

\* @param resource $context // not used for windows bug

\* @return boolean bool TRUE on success or FALSE on failure.

\*

\* @todo check recursive true and multiple chmod

\*/

public static function mkdir($path, $mode = 0777, $recursive = false, $context = null)

{

if (strlen($path) > self::maxPathLen()) {

throw new Exception('Skipping a file that exceeds allowed max path length ['.self::maxPathLen().']. File: '.$filepath);

}

if (!file\_exists($path)) {

if (!function\_exists('mkdir')) {

return false;

}

if (!@mkdir($path, 0777, $recursive)) {

return false;

}

}

return self::chmod($path, $mode);

}

/\*\*

\* move all folder content up to parent

\*

\* @param string $subFolderName full path

\* @param boolean $deleteSubFolder if true delete subFolder after moved all

\* @return boolean

\*

\*/

private function moveUpfromSubFolder($subFolderName, $deleteSubFolder = false)

{

if (!is\_dir($subFolderName)) {

return false;

}

$parentFolder = dirname($subFolderName);

if (!is\_writable($parentFolder)) {

return false;

}

$success = true;

if (($subList = glob(rtrim($subFolderName, '/').'/\*', GLOB\_NOSORT)) === false) {

self::log("[ERROR] Problem glob folder ".$subFolderName);

return false;

} else {

foreach ($subList as $cName) {

$destination = $parentFolder.'/'.basename($cName);

if (file\_exists($destination)) {

$success = self::deletePath($destination);

}

if ($success) {

$success = rename($cName, $destination);

} else {

break;

}

}

if ($success && $deleteSubFolder) {

$success = self::deleteDirectory($subFolderName, true);

}

}

if (!$success) {

self::log("[ERROR] Problem om moveUpfromSubFolder subFolder:".$subFolderName);

}

return $success;

}

/\*\*

\* Extracts only the 'dup-installer' files using Shell-Exec Unzip

\*

\* @param string $archive\_filepath The path to the archive file.

\*

\* @return bool Returns true if the data was properly extracted

\*/

private function extractInstallerShellexec($archive\_filepath)

{

$success = false;

self::log("Attempting to use Shell Exec");

$unzip\_filepath = $this->getUnzipFilePath();

if ($unzip\_filepath != null) {

$unzip\_command = "$unzip\_filepath -q $archive\_filepath ".self::INSTALLER\_DIR\_NAME.'/\* 2>&1';

self::log("Executing unzip command");

$stderr = shell\_exec($unzip\_command);

$lib\_directory = dirname(\_\_FILE\_\_).'/'.self::INSTALLER\_DIR\_NAME.'/lib';

$snaplib\_directory = $lib\_directory.'/snaplib';

// If snaplib files aren't present attempt to extract and copy those

if(!file\_exists($snaplib\_directory))

{

$local\_lib\_directory = dirname(\_\_FILE\_\_).'/snaplib';

$unzip\_command = "$unzip\_filepath -q $archive\_filepath snaplib/\* 2>&1";

self::log("Executing unzip command");

$stderr .= shell\_exec($unzip\_command);

self::mkdir($lib\_directory,'u+rwx');

rename($local\_lib\_directory, $snaplib\_directory);

}

if ($stderr == '') {

self::log("Shell exec unzip succeeded");

$success = true;

} else {

self::log("[ERROR] Shell exec unzip failed. Output={$stderr}");

}

}

return $success;

}

/\*\*

\* Attempts to get the archive file path

\*

\* @return string The full path to the archive file

\*/

private function getArchiveFilePath()

{

if (isset($\_GET['archive'])) {

$archive\_filepath = $\_GET['archive'];

} else {

$archive\_filename = self::ARCHIVE\_FILENAME;

$archive\_filepath = str\_replace("\\", '/', dirname(\_\_FILE\_\_) . '/' . $archive\_filename);

}

return $archive\_filepath;

}

/\*\*

\* Gets the DUPX\_Bootstrap\_Zip\_Mode enum type that should be used

\*

\* @return DUPX\_Bootstrap\_Zip\_Mode Returns the current mode of the bootstrapper

\*/

private function getZipMode()

{

$zip\_mode = DUPX\_Bootstrap\_Zip\_Mode::AutoUnzip;

if (isset($\_GET['zipmode'])) {

$zipmode\_string = $\_GET['zipmode'];

self::log("Unzip mode specified in querystring: $zipmode\_string");

switch ($zipmode\_string) {

case 'autounzip':

$zip\_mode = DUPX\_Bootstrap\_Zip\_Mode::AutoUnzip;

break;

case 'ziparchive':

$zip\_mode = DUPX\_Bootstrap\_Zip\_Mode::ZipArchive;

break;

case 'shellexec':

$zip\_mode = DUPX\_Bootstrap\_Zip\_Mode::ShellExec;

break;

}

}

return $zip\_mode;

}

/\*\*

\* Checks to see if a string starts with specific characters

\*

\* @return bool Returns true if the string starts with a specific format

\*/

private function startsWith($haystack, $needle)

{

return $needle === "" || strrpos($haystack, $needle, - strlen($haystack)) !== false;

}

/\*\*

\* Checks to see if the server supports issuing commands to shell\_exex

\*

\* @return bool Returns true shell\_exec can be ran on this server

\*/

public function hasShellExec()

{

$cmds = array('shell\_exec', 'escapeshellarg', 'escapeshellcmd', 'extension\_loaded');

//Function disabled at server level

if (array\_intersect($cmds, array\_map('trim', explode(',', @ini\_get('disable\_functions')))))

return false;

//Suhosin: http://www.hardened-php.net/suhosin/

//Will cause PHP to silently fail

if (extension\_loaded('suhosin')) {

$suhosin\_ini = @ini\_get("suhosin.executor.func.blacklist");

if (array\_intersect($cmds, array\_map('trim', explode(',', $suhosin\_ini))))

return false;

}

if (! function\_exists('shell\_exec')) {

return false;

}

// Can we issue a simple echo command?

if (!@shell\_exec('echo duplicator'))

return false;

return true;

}

/\*\*

\* Gets the possible system commands for unzip on Linux

\*

\* @return string Returns unzip file path that can execute the unzip command

\*/

public function getUnzipFilePath()

{

$filepath = null;

if ($this->hasShellExec()) {

if (shell\_exec('hash unzip 2>&1') == NULL) {

$filepath = 'unzip';

} else {

$possible\_paths = array(

'/usr/bin/unzip',

'/opt/local/bin/unzip',

'/bin/unzip',

'/usr/local/bin/unzip',

'/usr/sfw/bin/unzip',

'/usr/xdg4/bin/unzip',

'/opt/bin/unzip',

// RSR TODO put back in when we support shellexec on windows,

);

foreach ($possible\_paths as $path) {

if (file\_exists($path)) {

$filepath = $path;

break;

}

}

}

}

return $filepath;

}

/\*\*

\* Display human readable byte sizes such as 150MB

\*

\* @param int $size The size in bytes

\*

\* @return string A readable byte size format such as 100MB

\*/

public function readableByteSize($size)

{

try {

$units = array('B', 'KB', 'MB', 'GB', 'TB');

for ($i = 0; $size >= 1024 && $i < 4; $i++)

$size /= 1024;

return round($size, 2).$units[$i];

} catch (Exception $e) {

return "n/a";

}

}

/\*\*

\* Returns an array of zip files found in the current executing directory

\*

\* @return array of zip files

\*/

public static function getFilesWithExtension($extension)

{

$files = array();

foreach (glob("\*.{$extension}") as $name) {

if (file\_exists($name)) {

$files[] = $name;

}

}

if (count($files) > 0) {

return $files;

}

//FALL BACK: Windows XP has bug with glob,

//add secondary check for PHP lameness

if ($dh = opendir('.')) {

while (false !== ($name = readdir($dh))) {

$ext = substr($name, strrpos($name, '.') + 1);

if (in\_array($ext, array($extension))) {

$files[] = $name;

}

}

closedir($dh);

}

return $files;

}

/\*\*

\* Safely remove a directory and recursively if needed

\*

\* @param string $directory The full path to the directory to remove

\* @param string $recursive recursively remove all items

\*

\* @return bool Returns true if all content was removed

\*/

public static function deleteDirectory($directory, $recursive)

{

$success = true;

$filenames = array\_diff(scandir($directory), array('.', '..'));

foreach ($filenames as $filename) {

$fullPath = $directory.'/'.$filename;

if (is\_dir($fullPath)) {

if ($recursive) {

$success = self::deleteDirectory($fullPath, true);

}

} else {

$success = @unlink($fullPath);

if ($success === false) {

self::log('[ERROR] '.\_\_FUNCTION\_\_.": Problem deleting file:".$fullPath);

}

}

if ($success === false) {

self::log("[ERROR] Problem deleting dir:".$directory);

break;

}

}

return $success && rmdir($directory);

}

/\*\*

\* Safely remove a file or directory and recursively if needed

\*

\* @param string $directory The full path to the directory to remove

\*

\* @return bool Returns true if all content was removed

\*/

public static function deletePath($path)

{

$success = true;

if (is\_dir($path)) {

$success = self::deleteDirectory($path, true);

} else {

$success = @unlink($path);

if ($success === false) {

self::log('[ERROR] '. \_\_FUNCTION\_\_.": Problem deleting file:".$path);

}

}

return $success;

}

/\*\*

\* Makes path safe for any OS for PHP

\*

\* Paths should ALWAYS READ be "/"

\* uni: /home/path/file.txt

\* win: D:/home/path/file.txt

\*

\* @param string $path The path to make safe

\*

\* @return string The original $path with a with all slashes facing '/'.

\*/

public static function setSafePath($path)

{

return str\_replace("\\", "/", $path);

}

}

class DUPX\_Handler

{

/\*\*

\*

\* @var bool

\*/

private static $initialized = false;

/\*\*

\* This function only initializes the error handler the first time it is called

\*/

public static function init\_error\_handler()

{

if (!self::$initialized) {

@set\_error\_handler(array(\_\_CLASS\_\_, 'error'));

@register\_shutdown\_function(array(\_\_CLASS\_\_, 'shutdown'));

self::$initialized = true;

}

}

/\*\*

\* Error handler

\*

\* @param integer $errno Error level

\* @param string $errstr Error message

\* @param string $errfile Error file

\* @param integer $errline Error line

\* @return void

\*/

public static function error($errno, $errstr, $errfile, $errline)

{

switch ($errno) {

case E\_ERROR :

$log\_message = self::getMessage($errno, $errstr, $errfile, $errline);

if (DUPX\_Bootstrap::log($log\_message) === false) {

$log\_message = "Can\'t write logfile\n\n".$log\_message;

}

die('<pre>'.htmlspecialchars($log\_message).'</pre>');

break;

case E\_NOTICE :

case E\_WARNING :

default :

$log\_message = self::getMessage($errno, $errstr, $errfile, $errline);

DUPX\_Bootstrap::log($log\_message);

break;

}

}

private static function getMessage($errno, $errstr, $errfile, $errline)

{

$result = '[PHP ERR]';

switch ($errno) {

case E\_ERROR :

$result .= '[FATAL]';

break;

case E\_WARNING :

$result .= '[WARN]';

break;

case E\_NOTICE :

$result .= '[NOTICE]';

break;

default :

$result .= '[ISSUE]';

break;

}

$result .= ' MSG:';

$result .= $errstr;

$result .= ' [CODE:'.$errno.'|FILE:'.$errfile.'|LINE:'.$errline.']';

return $result;

}

/\*\*

\* Shutdown handler

\*

\* @return void

\*/

public static function shutdown()

{

if (($error = error\_get\_last())) {

DUPX\_Handler::error($error['type'], $error['message'], $error['file'], $error['line']);

}

}

}

class DUPX\_CSRF {

/\*\*

\* Session var name prefix

\* @var string

\*/

public static $prefix = '\_DUPX\_CSRF';

/\*\*

\* Stores all CSRF values: Key as CSRF name and Val as CRF value

\* @var array

\*/

private static $CSRFVars;

/\*\*

\* Set new CSRF

\*

\* @param $key string CSRF Key

\* @param $key string CSRF Val

\*

\* @return Void

\*/

public static function setKeyVal($key, $val) {

$CSRFVars = self::getCSRFVars();

$CSRFVars[$key] = $val;

self::saveCSRFVars($CSRFVars);

self::$CSRFVars = false;

}

/\*\*

\* Get CSRF value by passing CSRF key

\*

\* @param $key string CSRF key

\*

\* @return string|boolean If CSRF value set for give n Key, It returns CRF value otherise returns false

\*/

public static function getVal($key) {

$CSRFVars = self::getCSRFVars();

if (isset($CSRFVars[$key])) {

return $CSRFVars[$key];

} else {

return false;

}

}

/\*\* Generate DUPX\_CSRF value for form

\*

\* @param string $form - Form name as session key

\* @return string - token

\*/

public static function generate($form = NULL) {

$keyName = self::getKeyName($form);

$existingToken = self::getVal($keyName);

if (false !== $existingToken) {

$token = $existingToken;

} else {

$token = DUPX\_CSRF::token() . DUPX\_CSRF::fingerprint();

}

self::setKeyVal($keyName, $token);

return $token;

}

/\*\*

\* Check DUPX\_CSRF value of form

\*

\* @param string $token - Token

\* @param string $form - Form name as session key

\* @return boolean

\*/

public static function check($token, $form = NULL) {

$keyName = self::getKeyName($form);

$CSRFVars = self::getCSRFVars();

if (isset($CSRFVars[$keyName]) && $CSRFVars[$keyName] == $token) { // token OK

return true;

}

return FALSE;

}

/\*\* Generate token

\* @param void

\* @return string

\*/

protected static function token() {

mt\_srand((double) microtime() \* 10000);

$charid = strtoupper(md5(uniqid(rand(), TRUE)));

return substr($charid, 0, 8) . substr($charid, 8, 4) . substr($charid, 12, 4) . substr($charid, 16, 4) . substr($charid, 20, 12);

}

/\*\* Returns "digital fingerprint" of user

\* @param void

\* @return string - MD5 hashed data

\*/

protected static function fingerprint() {

return strtoupper(md5(implode('|', array($\_SERVER['REMOTE\_ADDR'], $\_SERVER['HTTP\_USER\_AGENT']))));

}

/\*\*

\* Generate CSRF Key name

\*

\* @param string the form name for which CSRF key need to generate

\* @return string CSRF key

\*/

private static function getKeyName($form) {

return DUPX\_CSRF::$prefix . '\_' . $form;

}

/\*\*

\* Get Package hash

\*

\* @return string Package hash

\*/

private static function getPackageHash() {

if (class\_exists('DUPX\_Bootstrap')) {

return DUPX\_Bootstrap::PACKAGE\_HASH;

} else {

return $GLOBALS['DUPX\_AC']->package\_hash;

}

}

/\*\*

\* Get file path where CSRF tokens are stored in JSON encoded format

\*

\* @return string file path where CSRF token stored

\*/

private static function getFilePath() {

if (class\_exists('DUPX\_Bootstrap')) {

$dupInstallerfolderPath = dirname(\_\_FILE\_\_).'/dup-installer/';

} else {

$dupInstallerfolderPath = $GLOBALS['DUPX\_INIT'].'/';

}

$packageHash = self::getPackageHash();

$fileName = 'dup-installer-csrf\_\_'.$packageHash.'.txt';

$filePath = $dupInstallerfolderPath.$fileName;

return $filePath;

}

/\*\*

\* Get all CSRF vars in array format

\*

\* @return array Key as CSRF name and value as CSRF value

\*/

private static function getCSRFVars() {

if (!isset(self::$CSRFVars) || false === self::$CSRFVars) {

$filePath = self::getFilePath();

if (file\_exists($filePath)) {

$contents = file\_get\_contents($filePath);

if (!($contents = file\_get\_contents($filePath))) {

throw new Exception('Fail to read the CSRF file.');

}

if (empty($contents)) {

self::$CSRFVars = array();

} else {

$CSRFobjs = json\_decode($contents);

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

self::$CSRFVars[$key] = $value;

}

}

} else {

self::$CSRFVars = array();

}

}

return self::$CSRFVars;

}

/\*\*

\* Stores all CSRF vars

\*

\* @param $CSRFVars array holds all CSRF key val

\* @return void

\*/

private static function saveCSRFVars($CSRFVars) {

$contents = json\_encode($CSRFVars);

$filePath = self::getFilePath();

if (!file\_put\_contents($filePath, $contents, LOCK\_EX)) {

throw new Exception('Fail to write the CSRF file.');

}

}

}

try {

$boot = new DUPX\_Bootstrap();

$boot\_error = $boot->run();

$auto\_refresh = isset($\_POST['auto-fresh']) ? true : false;

if ($boot\_error == null) {

$step1\_csrf\_token = DUPX\_CSRF::generate('step1');

DUPX\_CSRF::setKeyVal('archive', $boot->archive);

DUPX\_CSRF::setKeyVal('bootloader', $boot->bootloader);

DUPX\_CSRF::setKeyVal('secondaryHash', DUPX\_Bootstrap::SECONDARY\_PACKAGE\_HASH);

DUPX\_CSRF::setKeyVal('installerOrigCall', DUPX\_Bootstrap::getCurrentUrl());

DUPX\_CSRF::setKeyVal('installerOrigPath', \_\_FILE\_\_);

DUPX\_CSRF::setKeyVal('booturl', '//'.$\_SERVER['HTTP\_HOST'].$\_SERVER['REQUEST\_URI']);

DUPX\_CSRF::setKeyVal('bootLogFile', DUPX\_Bootstrap::getBootLogFilePath());

DUPX\_CSRF::setKeyVal('package\_hash', DUPX\_Bootstrap::PACKAGE\_HASH);

}

} catch (Exception $e) {

$boot\_error = $e->getMessage();

}

?>

<html>

<?php if ($boot\_error == null) :?>

<head>

<meta name="robots" content="noindex,nofollow">

<title>Duplicator Installer</title>

</head>

<body>

<?php

$id = uniqid();

$html = "<form id='{$id}' method='post' action=".str\_replace('\\/', '/', json\_encode($boot->mainInstallerURL))." />\n";

$data = array(

'csrf\_token' => $step1\_csrf\_token,

);

foreach ($data as $name => $value) {

$html .= "<input type='hidden' name='{$name}' value='{$value}' />\n";

}

$html .= "</form>\n";

$html .= "<script>window.onload = function() { document.getElementById('{$id}').submit(); }</script>";

echo $html;

?>

</body>

<?php else :?>

<head>

<style>

body {font-family:Verdana,Arial,sans-serif; line-height:18px; font-size: 12px}

h2 {font-size:20px; margin:5px 0 5px 0; border-bottom:1px solid #dfdfdf; padding:3px}

div#content {border:1px solid #CDCDCD; width:750px; min-height:550px; margin:auto; margin-top:18px; border-radius:3px; box-shadow:0 8px 6px -6px #333; font-size:13px}

div#content-inner {padding:10px 30px; min-height:550px}

/\* Header \*/

table.header-wizard {border-top-left-radius:5px; border-top-right-radius:5px; width:100%; box-shadow:0 5px 3px -3px #999; background-color:#F1F1F1; font-weight:bold}

table.header-wizard td.header {font-size:24px; padding:7px 0 7px 0; width:100%;}

div.dupx-logfile-link {float:right; font-weight:normal; font-size:12px}

.dupx-version {white-space:nowrap; color:#999; font-size:11px; font-style:italic; text-align:right; padding:0 15px 5px 0; line-height:14px; font-weight:normal}

.dupx-version a { color:#999; }

div.errror-notice {text-align:center; font-style:italic; font-size:11px}

div.errror-msg { color:maroon; padding: 10px 0 5px 0}

.pass {color:green}

.fail {color:red}

span.file-info {font-size: 11px; font-style: italic}

div.skip-not-found {padding:10px 0 5px 0;}

div.skip-not-found label {cursor: pointer}

table.settings {width:100%; font-size:12px}

table.settings td {padding: 4px}

table.settings td:first-child {font-weight: bold}

.w3-light-grey,.w3-hover-light-grey:hover,.w3-light-gray,.w3-hover-light-gray:hover{color:#000!important;background-color:#f1f1f1!important}

.w3-container:after,.w3-container:before,.w3-panel:after,.w3-panel:before,.w3-row:after,.w3-row:before,.w3-row-padding:after,.w3-row-padding:before,

.w3-cell-row:before,.w3-cell-row:after,.w3-clear:after,.w3-clear:before,.w3-bar:before,.w3-bar:after

{content:"";display:table;clear:both}

.w3-green,.w3-hover-green:hover{color:#fff!important;background-color:#4CAF50!important}

.w3-container{padding:0.01em 16px}

.w3-center{display:inline-block;width:auto; text-align: center !important}

</style>

</head>

<body>

<div id="content">

<table cellspacing="0" class="header-wizard">

<tr>

<td class="header"> &nbsp; Duplicator - Bootloader</td>

<td class="dupx-version">

version: <?php echo DUPX\_Bootstrap::VERSION ?> <br/>

</td>

</tr>

</table>

<form id="error-form" method="post">

<div id="content-inner">

<h2 style="color:maroon">Setup Notice:</h2>

<div class="errror-notice">An error has occurred. In order to load the full installer please resolve the issue below.</div>

<div class="errror-msg">

<?php echo $boot\_error ?>

</div>

<br/><br/>

<h2>Server Settings:</h2>

<table class='settings'>

<tr>

<td>ZipArchive:</td>

<td><?php echo $boot->hasZipArchive ? '<i class="pass">Enabled</i>' : '<i class="fail">Disabled</i>'; ?> </td>

</tr>

<tr>

<td>ShellExec&nbsp;Unzip:</td>

<td><?php echo $boot->hasShellExecUnzip ? '<i class="pass">Enabled</i>' : '<i class="fail">Disabled</i>'; ?> </td>

</tr>

<tr>

<td>Extraction&nbsp;Path:</td>

<td><?php echo $boot->installerExtractPath; ?></td>

</tr>

<tr>

<td>Installer Path:</td>

<td><?php echo $boot->installerContentsPath; ?></td>

</tr>

<tr>

<td>Archive Name:</td>

<td>

[HASH]\_archive.zip or [HASH]\_archive.daf<br/>

<small>This is based on the format used to build the archive</small>

</td>

</tr>

<tr>

<td>Archive Size:</td>

<td>

<b>Expected Size:</b> <?php echo $boot->readableByteSize($boot->archiveExpectedSize); ?> &nbsp;

<b>Actual Size:</b> <?php echo $boot->readableByteSize($boot->archiveActualSize); ?>

</td>

</tr>

<tr>

<td>Boot Log</td>

<td>dup-installer-bootlog\_\_[HASH].txt</td>

</tr>

</table>

<br/><br/>

<div style="font-size:11px">

Please Note: Either ZipArchive or Shell Exec will need to be enabled for the installer to run automatically otherwise a manual extraction

will need to be performed. In order to run the installer manually follow the instructions to

<a href='https://snapcreek.com/duplicator/docs/faqs-tech/#faq-installer-015-q' target='\_blank'>manually extract</a> before running the installer.

</div>

<br/><br/>

</div>

</form>

</div>

</body>

<script>

function AutoFresh() {

document.getElementById('error-form').submit();

}

<?php if ($auto\_refresh) :?>

var duration = 10000; //10 seconds

var counter = 10;

var countElement = document.getElementById('count-down');

setTimeout(function(){window.location.reload(1);}, duration);

setInterval(function() {

counter--;

countElement.innerHTML = (counter > 0) ? counter.toString() : "0";

}, 1000);

<?php endif; ?>

</script>

<?php endif; ?>

<!--

Used for integrity check do not remove:

DUPLICATOR\_INSTALLER\_EOF -->

</html>