

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра систем информационной безопасности

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой к.п.н., доцент Р. Ш. Бердибаев
_____ « _____ » _____ 2019 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка электронного торгового портала с защитой пользовательских данных

Специальность 5В100200 – «Системы информационной безопасности»

Выполнил Жумахан Дамир Бауржанулы

Группа СИБ-15-2

Научный руководитель Бердибаев Рат Шындалиевич

Консультант:

по экономической части:

К. Ф. И., профессор Аренбаева М. Г.

(ученая степень, звание, Ф.И.О)

М. Аренбаева « 10 » июня 2019 г.
(подпись)

по безопасности жизнедеятельности:

А. Т. И. С. прел. Бекбасаров И. И.

(ученая степень, звание, Ф.И.О)

И. Бекбасаров « 12 » июня 2019 г.
(подпись)

по применению вычислительной техники:

К. п. н. И., доцент Бердибаев Р. Ш.

(ученая степень, звание, Ф.И.О)

Р. Ш. Бердибаев « 12 » июня 2019 г.
(подпись)

Нормоконтролер:

Ст. преподаватель Асмадова А. Б.

(ученая степень, звание, Ф.И.О)

А. Асмадова « 11 » июня 2019 г.
(подпись)

Рецензент:

PhD Бекмембаева Е. Е.

(ученая степень, звание, Ф.И.О)

Е. Е. Бекмембаева « 12 » июня 2019 г.
(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Институт систем управления и информационных технологий
Кафедра систем информационной безопасности
Специальность 5В100200 – «Системы информационной безопасности»

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Жумахан Дамиру Бауржанулы

Тема проекта Разработка электронного торгового портала с защитой пользовательских данных

Утверждена приказом по университету № 124 от «26» 10 2018 г.

Срок сдачи законченного проекта «10» июня 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): работа подразумевает создание и проектирование электронного торгового портала с системой защиты пользовательских данных. В качестве инструмента для улучшения безопасности использовались различные методы, разграничение ролей пользователей, администраторов. В системе защиты будут использоваться шифрование и дешифрование. А также применяться различные методы защиты при помощи программного обеспечения «Laravel».

Дипломный проект включает в себя несколько глав, разделенные на подглавы, каждая из которых освещает определенную тематику, связанную с созданием электронного торгового портала и проектированием системы защиты.

В первой главе дипломного проекта представлена информация об анализе электронных торговых площадок: виды, назначение и функции порталов, сравнение, исследование и защита пользовательских данных.

Во второй главе дипломного проекта представлена информация об мерах защиты пользовательских данных: внутренние меры защиты и внешние меры защиты.

В третьей главе подробно описывается проектирование и конфигурация электронно-торгового портала с системой защиты пользовательских данных.

на электроэнергию, затраты на оплату труда, затраты на социальный налог и затраты на амортизацию.

Перечень графического материала (с точным указанием обязательных чертежей):

- 1 топология;
- 2 скрины электронно-торгового портала;
- 3 скрины с примененными методами защиты;
- 4 скрины с реализации электронно-торгового портала;
- 5 скрины с шифрованием.

Основная рекомендуемая литература:

- 1 Demina N.V, Chistova M.V To the question on specifics of work on electronic trading facilities in terms of cybercrime prevention // Modern science.2017
- 2 Классификация ЭТП. URL: http://xn____6kcbaifbn4di5abenic8aq
- 3 Чистова М.В., Концевич Г.Е., Демина Н.В. Возможности внедрения информационных технологий для реформирования жилищно-коммунального хозяйства. 2014 6 с.95-101
- 4 Торговые площадки URL: <http://www.tfc72.ru/torgoviyploshadki.html>
- 5 Тупикина А.В Электронные торговые площадки. // Информационные технологии в экономике, образовании и бизнесе.
- 6 Электронные торговые площадки: обзор рынка. URL: <http://b2blogger.com/articles/review>

Конструкции по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономика	Арибаева М.Г.	04.03-10.06	М.Арибаева
БНН	Бекбаева Ш.Ш	04.03-10.06	Ш.Бекбаева
Юридический отдел	Бекбаева Р.А.	14.01-03.03.	Р.А.Бекбаева

**График
подготовки дипломного проекта**

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Электронный тематический портал	1.02.19 - 6.02.19	
FTP. Виды и их функции	11.02.19 - 15.02.19	
Измерение и анализ по CANAL	18.02.19 - 22.02.19	
Настройка маршрутного обеспечения	25.02.19 - 28.02.19	
Хранение конфиденциальных данных	04.03.19 - 08.03.19	
Векторные коллекции	11.03.19 - 15.03.19	
Фреймворк. Виды и их функции	18.03.19 - 22.03.19	
Решения FTP	25.03.19 - 29.03.19	
Решение задач	01.04.19 - 05.04.19	
Задачи и требования к созданию	08.04.19 - 12.04.19	
Проектирование структуры	15.04.19 - 19.04.19	
Видение и настройка защиты	22.04.19 - 26.04.19	
Создание защиты данных	29.04.19 - 03.05.19	
Создание матрицы реализации	06.05.19 - 10.05.19	
Использование фреймворка	13.05.19 - 17.05.19	
Настройка базового функционала	20.05.19 - 24.05.19	
Безопасность использования системы	27.05.19 - 31.05.19	
Валидация условий труда	03.06.19 - 05.06.19	
Расчет тепловых нагрузок	05.06.19 - 08.06.19	
Расчет тепловых балансов	08.06.19 - 09.06.19	

Дата выдачи задания «14» января 2019 г.

Заведующий кафедрой _____ (Подпись) _____ (Ф.И.О)

Научный руководитель проекта _____ (Подпись) _____ (Ф.И.О)

Задание принял к исполнению студент _____ (Подпись) _____ (Ф.И.О)

АННОТАЦИЯ

Дипломный проект посвящен проектированию защиты пользовательских данных на уровне электронного торгового портала. Была спроектирована электронная торговая площадка, благодаря которой пользователи могли бы использовать портал для своих целей. Рассмотрены различные параметры конфигурации программного обеспечения Laravel, на которой был создан электронный торговый портал. В целях улучшения защиты пользовательских данных был выполнен backup и шифрование данных пользователей.

Также был произведен анализ условий труда при проектировании системы защиты и эксплуатации. В части технико-экономических обосновании были произведены расчеты экономической эффективности проектируемой системы защиты.

АҢДАТПА

Дипломдық жоба операциялық жүйе деңгейінде пайдаланушылық деректерді қорғауды жобалауға арналған. Ұйымның атау электрондық сауда порталы жобаланды. Электрондық сауда порталы құрыстарылған Laravel бағдарламалық жасақтамасының түрлі конфигурация параметрлері қарастырылады. Пайдаланушы деректерін қорғауды жақсарту мақсатында сақтық көшірме және пайдаланушылардың файлдарды шифрлеу орындалды.

Сонымен қатар, қорғау және пайдалану жүйесін жобалау кезінде еңбек жағдайларына талдау жүргізілді. Техникалық-экономикалық негіздемелер бөлігінде жобаланатын қорғау жүйесінің экономикалық тиімділігінің есептеулері жүргізілді.

ANNOTATION

The graduation project is dedicated to data user data protection at the level of an electronic trading portal. An electronic marketplace was designed, thanks to which user could use the portal for their own purposes. The various configuration parameters of the Laravel software, on which the electronic trading portal was created, are considered. In order to improve data protection, backup and encryption of user data was performed.

Also, an analysis of working conditions in the design of protection and operation systems was made. In terms of feasibility studies, calculations were made of the economic efficiency of the designed protection system.

Содержание

Введение.....	3
1. ЭЛЕКТРОННЫЙ ТОРГОВЫЙ ПОРТАЛ.....	4
1.1 Электронные торговые площадки. Виды и их функции.....	4
1.2 ИЗУЧЕНИЕ И АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LARAVEL.....	7
1.3 СОПРОВОЖДЕНИЕ ПРОГРАММНОГО ПРОЕКТА.....	21
2 РЕАЛИЗАЦИЯ ЭЛЕКТРОННОЙ ТОРГОВОЙ ПЛОЩАДКИ.....	23
2.1 ПОСТАНОВКА ЗАДАЧИ.....	23
2.2 ТРЕБОВАНИЯ К СОЗДАНИЮ ЭЛЕКТРОННО-ТОРГОВОЙ ПЛОЩАДКИ И ПРОЕКТИРОВАНИЮ ЗАЩИТЫ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ.....	23
2.3 ОБЪЕКТ ЗАЩИТЫ.....	24
2.4 ПРОЕКТИРОВАНИЕ ПЛОЩАДКИ.....	24
3 СОЗДАНИЕ И НАСТРОЙКА ЗАЩИТЫ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ.....	30
3.1 СОЗДАНИЕ И НАСТРОЙКА ПЛАТФОРМЫ РЕАЛИЗАЦИИ ЭЛЕКТРОННОЙ ТОРГОВОЙ ПЛОЩАДКИ С СИСТЕМОЙ ЗАЩИТЫ.....	32
3.1.1 Используемый фреймворк.....	32
3.1.2 Возможности веб-фреймворка – Laravel.....	32
3.1.3 Расширение базового функционала веб-фреймворка – Laravel.....	34
4 Безопасность жизнедеятельности.....	37
4.1 Анализ условий труда.....	37
4.2 Расчет тепловых нагрузок в помещении.....	39
4.3 РАСЧЕТ ТЕПЛОВОГО БАЛАНСА ПОМЕЩЕНИЯ.....	41
4.4 ВЫВОД ПО ЧАСТИ БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....	44
5 Технико-экономическое обоснование.....	46
5.1 Определение трудоемкости разработки.....	47
5.2 Расчет затрат на разработку.....	47
5.3 Расчет затрат на электроэнергию.....	49
5.4 Расчет затрат на оплату труда.....	50
5.5 Расчет затрат по социальному налогу.....	51
5.6 Амортизация основных фондов и прочие затраты.....	52
5.7 Определение возможной (договорной) цены ПО.....	53
ЗАКЛЮЧЕНИЕ.....	54
Список литературы.....	55
Список литературы.....	57

Введение

У каждого фреймворка есть огромное количество поклонников и противников. При выборе фреймворка необходимо учитывать несколько факторов. Во-первых, фреймворк должен иметь ровно столько возможностей, сколько необходимо проекту. Излишний функционал не нужен, но при этом никогда не должно возникать ограничений при доработке проекта. Во-вторых, время разработки сайта не должно увеличиваться за счет выбора того или иного фреймворка. В-третьих, фреймворк должен быть достаточно распространен, для того чтобы проект можно было передать другому разработчику. Наконец, безопасность фреймворка - один из важнейших показателей. Все фреймворки хороши, если находятся в правильных руках.

Ключевые слова: информационная безопасность, центр оборудования, техники и технологий.

Проанализировав рынок СНГ в сфере оборудования, техники и технологий мне удалось узнать, что нет единого центра, который мог бы сосредоточить всех физических и юридических лиц, а также экспертов в области оборудования, техники и информационных технологий. После длительных размышлений, анализов и взвешивания рисков я понял, что рынок востребован в проекте под названием «TechnoMarkets», который я реализовывал в течение шести месяцев.

Создавать заново велосипед мне не хотелось и потому я решил использовать «движок» для этих целей. Мое внимание привлек Laravel 5.5, который был выпущен 30 августа 2017 года.

Исследуя возможности Laravel, я понял, что данный веб-фреймворк подходит как нельзя лучше для создания моего портала «TechnoMarkets». Laravel включает поддержку планирования периодически выполняемых задач через пакет Scheduler, слой абстракции Flysystem, который позволяет использовать удаленное хранилище также, как и локальные файловые системы, улучшенную обработку активов пакета через Elixir и упрощенная аутентификация с внешней стороны через дополнительный пакет Socialite.

В выпуске 5.5 есть много функций, таких как Laravel Dusk, Laravel Mix, Blade Components и Slots, Markdown Emails, автоматические фасады, улучшения маршрута, которые мне отлично подходили для разработки. Есть пакет Whoops для отладки ошибок, что очень удобно в корректировке тех или иных кодов. Существуют метод обработки запроса, новые директивы для шаблонизатора Blade, команда для миграции баз данных - migrate:fresh. Все данные возможности позволили понять саму работу по модернизации и структуру «движка» в целом.

1. ЭЛЕКТРОННЫЙ ТОРГОВЫЙ ПОРТАЛ.

1.1 Электронные торговые площадки. Виды и их функции

Электронная торговая площадка (ЭТП, e-market, c-marketplace) — это информационная торговая система, позволяющая дистанционно осуществлять поиск, покупки и продажи товаров и услуг, совершать сделки и другие торговые и финансовые операции с использованием информационно-телекоммуникационных сетей и информационных технологий. Они также называются виртуальными торговыми площадками. Сделки купли-продажи товаров и услуг на ЭТП совершаются в режиме реального времени практически с любыми видами товаров и услуг. Продавцы и покупатели могут находиться на больших расстояниях друг от друга. Работа на ЭТП, как правило, является платной. На некоторых площадках плата взимается с каждой транзакции, на других устанавливается фиксированная цена независимо от проводимых транзакций. Типы ЭТП и их краткая характеристика. Электронные торговые площадки подразделяются на следующие типы:

- государственные и коммерческие;
- оптовой, розничной и оптово-розничной торговли;
- открытые и закрытые;
- многопрофильные и узкоспециализированные;
- создаваемые продавцами, покупателями, третьими лицами;
- вертикальной и горизонтальной структуры.

Государственные ЭТП предназначены для осуществления госзакупок и госзаказов в электронном виде. Одной из крупных коммерческих ЭТП является B2B-Center. В 2013 г. 42,8% торгов в корпоративном секторе осуществлялось с помощью этой ЭТП. Система располагает развитыми системами статистики и аналитики для выявления трендов и проведения собственного анализа цен на товары и услуги, определения результативности торгов по отраслям и компаниям. Электронные торговые площадки используются в розничной, оптовой и оптово-розничной торговле. Используемые при этом электронные методы торговли позволяют оптимизировать стоимость товаров и услуг, сроки их поставки, увеличить число потенциальных клиентов, сократить финансовые издержки, оптимизировать поиск деловой информации, необходимой для совершения сделки. Электронные торговые площадки могут быть закрытыми, предназначенными для работы ограниченного, строго определенного контингента (например, "производитель — поставщик — потребитель"), и открытыми, предназначенными для неограниченного числа покупателей и продавцов. Электронные торговые площадки могут быть многопрофильными и узкоспециализированными. Многопрофильные характеризуются большей посещаемостью, но менее эффективны и более сложны с организационной

точки зрения, чем специализированные площадки. Электронные торговые площадки подразделяются на:

- создаваемые покупателем;
- создаваемые продавцом;
- создаваемые третьими лицами, посредниками.

Площадки, создаваемые покупателями (buyer-driven), принадлежат одной или нескольким крупным фирмам. Создаются для привлечения компаний-поставщиков с целью оптимизации процесса закупок, расширения деловых контактов и сети поставок. Площадки, создаваемые продавцами (seller-driven), принадлежат одной или нескольким крупным фирмам-продавцам. Создаются с целью увеличения числа покупателей, снижения затрат по продаже товаров или услуг. Площадки, создаваемые посредниками, предназначены для сведения покупателей и продавцов, и удовлетворения их взаимных деловых интересов. Некоторые ЭТП, например, биржи, в зависимости от их структуры подразделяются на вертикальные и горизонтальные. Вертикальные ЭТП функционируют в определенных отраслях — сельском хозяйстве, электроэнергетике, нефтехимии, металлургии и т.д. Горизонтальные ЭТП специализируются на отдельных видах продукции, характерных для разных отраслей. Работая на Электронной Торговой Площадке, заказчик или поставщик может успешно решать различные вопросы, возникающие в повседневной деловой практике, потому что данные системы выполняют следующие важные функции:

1. Информационная функция позволяет ознакомиться с перечнем организаций, работающих на ЭТП, получить информацию по интересующей организации.
2. Функция маркетинга позволяет осуществлять поиск покупателей и потребителей интересующих работ и услуг, а также получать информацию о потребностях и предложениях работ и услуг, которые размещают на площадке другие организации.
3. Рекламная функция. Разместив информацию о своей организации на ЭТП, фирма сразу попадает в единое информационное пространство.
4. Торговая функция - позволяет в качестве организатора торгов осуществлять полный комплекс разнообразных торгово-закупочных мероприятий по приобретению товаров и услуг. В качестве участника торгов - осуществлять комплекс действий для эффективной продажи деятельности организаций, правильно выбрать контрагентов для выполнения поставок, работ и услуг по интересующей тематике.
6. Функция защиты информации делает возможным безопасный электронный документооборот, построенный с использованием сертифицированных средств криптографической защиты информации (ЭЦП).

Модели работы ЭТП:

Возможны различные принципы взаимодействия участников сделки на ЭТП. Это могут быть либо модели с фиксированными ценами, типичные для продаж по каталогам, либо системы динамического ценообразования, характерные для аукционной или биржевой торговли. В современных ЭТП используются три основные модели организации взаимодействия участников: онлайн-каталог, аукцион и биржа. В случае если ЭТП предоставляет сразу несколько типов взаимодействия, ее клиенты могут выбрать наиболее приемлемую для них модель.

Онлайн-каталог (ONLINE CATALOG) обеспечивает быстрый выход покупателей на нужных продавцов. Вместо того, чтобы просматривать горы зачастую устаревших каталогов поставщика, покупатели могут использовать поисковые возможности интернета, сравнивая товары сразу по нескольким параметрам, включая цену, даты поставки, гарантии, информацию по обслуживанию и т. д. Каталогная модель оптимальна для отраслей, характеризующихся большим количеством мелких предприятий и частым проведением сделок с относительно недорогими товарами, поскольку позволяет покупателю ознакомиться с предложениями большого числа поставщиков.

Аукцион (Auction) в основном отличается от онлайн-каталога тем, что цены не фиксированы, а устанавливаются во время торгов. Данная модель используется для проведения электронных закупок на конкурсной (тендерной) основе.

Биржа (Exchange) применяется на рынках, где цена может быстро меняться в зависимости от соотношений спроса и предложения. Эта модель позволяет согласовывать спрос/предложение и определять рыночные цены, а также регистрировать и проводить сделки в режиме реального времени. Некоторые аналитики в качестве отдельного вида ЭТП выделяют "сообщество" (COMMUNITY). Это информационные порталы, представляющие посетителям экономические новости, отраслевую и корпоративную аналитику, списки вакансий, а также позволяющие участникам общаться через чат или с помощью доски объявлений. Хотя площадки такого рода и могут сводить в одном виртуальном пространстве потенциальных покупателей и продавцов, отнести их к ЭТП можно с некоторыми оговорками. Далеко не всегда информационные порталы предназначены их создателями для проведения электронных сделок и обладают сервисами их поддержки. Строго говоря, к ЭТП относятся лишь те из информационных порталов, которые обладают также функциями электронного каталога, аукциона или биржи.

1.2 ИЗУЧЕНИЕ И АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ LARAVEL

Laravel стремится преобразить процесс разработки PHP, это относится и к локальной среде разработки. Vagrant обеспечивает простой, элегантный способ настройки управления виртуальными машинами.

Laravel Homestead — официальная подготовленная Vagrant-"коробка", которая предоставляет вам замечательную среду проектирования без обязательной установки PHP, HHVM, веб-сервера и любого другого программного серверного обеспечения на вашем компьютере. Можно больше не беспокоиться о том, что ваша операционная система засоряется! Vagrant-коробки очень удобны. Если что-то пошло не так, вы можете уничтожить и пересоздать коробку в считанные минуты.

Homestead запускается на любой ОС Windows, Mac или Linux, и включает веб-сервер Nginx, PHP 5.6, MySQL, Postgres, Redis, Memcached и все другие полезные штуки, которые вам понадобятся для разработки удивительных Laravel-приложений. Установка VirtualBox и Vagrant. Прежде чем запустить Homestead-среду, вы должны установить VirtualBox и Vagrant. Эти программные пакеты предоставляют простые в использовании визуальные инсталляторы для всех популярных операционных систем. Добавление Vagrant-коробки. Как только VirtualBox и Vagrant будут установлены, вы должны добавить коробку laravel/homestead в установленный пакет Vagrant, используя следующую команду в вашем терминале: `vagrant box add laravel/homestead`. Потребуется несколько минут, чтобы загрузить коробку. Все зависит от вашей скорости подключения к Интернету. Установка Homestead с помощью Composer + PHP Tool.

Преимущества Laravel:

- удобно читаемые Laravel коллекции и хелперы по работе с ними;
- storage - отдельное место в структуре для хранилища;
- eloquent - ORM для работы с базой, реализация шаблона ActiveRecord. Помимо стандартных CRUD-операций, в ней есть: методы доступа и мутаторы, безопасное удаление, области запросов, отношения, наблюдатели модели;
- artisan - позволяет управлять командной строкой;
- простая маршрутизация, валидация входящих параметров;
- интегрированный сборщик scss и скриптов (elixir);
- интегрированный шаблонизатор Blade;
- очереди заданий, планировщик задач;
- миграции и сиды;
- интегрированная система модульного тестирования phpunit;
- поддержка WebSockets для создания настоящих интерактивных приложений;
- ACL плагины;

-высокая популярность на западе;

-частые релизы новых версий.

Недостатки Laravel

-отсутствуют интегрированные шаблоны интерфейсов.

Ниже приведу сравнение фреймворков Yii и Laravel

Характеристика	Yii	Laravel
Скорость обучения	Один из самых высокопроизводительных фреймворков PHP.	Слабое место Laravel. Фреймворк сравнительно медленно дает ответ и собирает данные в режиме реального времени.
Валидация форм	Привязана к классу формы или модели.	Валидатор существует как отдельный класс-помощник.
Безопасность	ACF, RBAC плюс расширения для авторизации по OpenID, OAuth или OAuth2.	RBAC, ACL и фильтры.
Работа с шаблонами	Twig и Smarty, Asset, плагины	Blade, Asset, плагины. Кэширование шаблонов настроено по дефолту.
Кэширование	APC, Database, File Memcached, Redis WinCache, XCache Zend Data Cache.	APC, Database, File Memcached, Redis.
Локализация приложений	Да	Да
Простота освоения*	Легко	Легко
Документация на русском	Есть	Есть
Сообщество GitHub (отслеживают/поставили звезду)	487 / 4 862	4629 / 47 303
Количество расширений	2 170	14 672

Рисунок 1.1.2 – сравнение 1

Характеристика	Yii	Laravel
Последняя версия	2.0.15	5.7
Дата последнего релиза	март 2018	сентябрь 2018
Версия PHP	≥ 5.4.0	≥ 7.1.3
Дополнительные требования	-	PHP JSON, MCrypt PHP
Установка	Нужно загрузить приложение. С установкой справится новичок.	Для установки потребуется Composer PHP. Кроме того, в комплекте предоставляется специальный установщик.
MVC	Да	Да
ORM	DAO, ActiveRecord, Doctrine 2	DAO, Eloquent ORM, Doctrine 2
Встроенные решения для интерфейсов	Bootstrap и встроенные модули.	-
Тестирование	Встроенные инструменты: PHPUnit и Codeception.	BrowserKit, DomCrawler и HttpKernel и встроенный PHPUnit.
Отладка	Консоль с несколькими уровнями протоколирования для регистрации и дебаггинга запросов к базам данных.	Протоколирование и Imho. Плюс можно дополнительно установить расширение.
Миграция	Встроенный инструмент (класс Migration).	Встроенный инструмент (Migration) и инструмент наполнения данными (Seeder).

Рисунок 1.1.3 – сравнение 2

Рассмотрение пакета Vagrant

Как только коробка будет добавлена к пакету Vagrant, можно устанавливать консоль Homestead, используя команду `Composer global: composer global require "laravel/homestead=~2.0"`. Не забудьте поместить директорию `~/.composer/vendor/bin` в переменную `PATH`, чтобы был найден исполняемый файл `homestead` при запуске команды `homestead` в терминале. После установки консоли Homestead выполните команду `init`, чтобы создать конфигурационный файл `Homestead.yaml`: `homestead init`.

Файл `Homestead.yaml` будет помещён в директорию `~/.homestead`. Если вы используете систему Mac или Linux, вы можете отредактировать файл `Homestead.yaml`, выполнив команду `homestead edit` в терминале: `homestead edit`. Ручная установка с помощью Git (без локального PHP). Если вы не хотите устанавливать PHP на своей локальной машине, вы можете установить Homestead вручную, просто клонировав репозиторий. Клонировать репозиторий в центральный каталог Homestead, где вы храните все свои проекты Laravel, так, чтобы коробка Homestead стала хостом всех ваших Laravel (и PHP) проектов: `git clone https://github.com/laravel/homestead.git Homestead`. После установки консоли Homestead выполните команду `bash init.sh`, чтобы создать конфигурационный файл `Homestead.yaml`: `bash init.sh`. Файл `Homestead.yaml` будет помещён в директорию `~/.homestead`.

Настройка ключа SSH

Затем отредактируйте файл `Homestead.yaml`. В этом файле вы можете настроить путь к своему публичному ключу SSH, а также настроить папки, которые вы хотите расшарить между вашей основной машиной и виртуальной машиной Homestead. На Mac и Linux вы можете создать пару SSH-ключей, используя следующую команду: `ssh-keygen -t rsa -C "you@homestead"`. На Windows вы можете установить `.com/Git` и использовать Git Bash, встроенную в оболочку Git, чтобы выполнить команду, указанную выше. Также вы можете использовать `PuTTY` и `PuTTYgen`. Как только вы создали SSH-ключ, задайте путь к ключу в параметре `authorize` в вашем файле `Homestead.yaml`.

Настройка общих папок

Параметр `Folders` в файле `Homestead.yaml` перечисляет все папки, которые вы хотите расшарить для вашей среды Homestead. Поскольку файлы в этих папках будут меняться, они будут синхронизироваться с вашей локальной машиной и средой Homestead. Вы можете настроить столько папок, сколько вам необходимо.

Настройка сайтов Nginx

Не знакомы с Nginx? Нет проблем. Параметр `sites` позволяет легко связать «домен» с папкой в среде Homestead. Типовая конфигурация сайта включена в файл `Homestead.yaml`. И снова вы можете добавить столько сайтов

к своей среде Homestead, сколько необходимо. Homestead может служить удобной виртуальной средой для каждого проекта Laravel, над которым вы работаете. Вы можете настроить использование HHVM для любого сайта Homestead, установив параметр `hhvm` в значение `true`:

sites:

- `map: homestead.app. to: /home/vagrant/Code/Laravel/public;`

- `hhvm: true;`

- Bash Aliases.

Чтобы добавить Bash aliases к вашей коробке Homestead, просто добавьте их в файл `aliases` в корне директории `~/homestead`.

Запуск Vagrant Box

Как только вы отредактировали `Homestead.yaml`, выполните команду `homestead up` в вашем терминале. Если вы установили Homestead вручную и не используете инструмент PHP `homestead`, выполните команду `vagrant up` из директории, которая содержит ваш клонированный репозиторий Homestead Git. Vagrant будет загружать виртуальную машину и формировать ваши общие папки и сайты Nginx автоматически! Чтобы уничтожить машину, вы можете использовать команду `homestead destroy`. Для просмотра полного списка доступных команд Homestead выполните `homestead list`. Не забывайте добавлять «домены» для своих сайтов Nginx в файл `hosts` на вашей машине. Файл `hosts` перенаправит ваши запросы к локальным доменам в вашу среду Homestead. На Mac и Linux этот файл расположен в `/etc/hosts`. На Windows он расположен в `C:\Windows\System32\drivers\etc\hosts`. Необходимо удостовериться, что IP-адрес тот же, что вы установили в своём файле `Homestead.yaml`. Как только вы добавили домен в свой файл `hosts`, вы можете получить доступ к сайту через свой веб-браузер. `http://homestead.app`

Подключение к базам данных

База `homestead` сразу настроена на использование MySQL, и Postgres. Для ещё большего удобства конфигурация БД Laravel local будет использовать эту базу данных по умолчанию. Чтобы подключиться к вашей базе данных MySQL или Postgres с вашей главной машины через Navicat или Sequel Pro, вы должны подключиться к 127.0.0.1 через порт 33060 (MySQL) или 54320 (Postgres). Имя пользователя и пароль для обеих баз данных — `homestead / secret`.

Добавление дополнительных сайтов

После настройки и запуска вашей среды Homestead вы можете захотеть добавить дополнительные сайты Nginx для своих Laravel-приложений. Вы можете запустить в одной среде Homestead столько установок Laravel, сколько захотите. Есть два способа сделать это. Вы можете просто добавить сайты в свой файл `Homestead.yaml` и затем выполнить `vagrant provision`. А ещё вы можете использовать скрипт `serve`, который доступен в среде Homestead.

Чтобы использовать скрипт `serve`, подключитесь по SSH к вашей среде Homestead и запустите следующую команду: `serve domain.app /home/vagrant/Code/path/to/public/directory`.

Все файлы настроек Laravel хранятся в папке `config`. Действия после установки – имя вашего приложения. Первая вещь, которую вы должны сделать после установки фреймворка - придумать имя своему будущему приложению. Это имя будет фигурировать в качестве корневого имени в неймспейсе классов, которые будут использоваться в вашем приложении. По умолчанию ваше приложение находится в папке `app`, имеет имя `App` и автозагружается при помощи Composer согласно стандарту PSR-4. Вы можете изменить его при помощи артизанкоманды `app:name`. Например, чтобы изменить имя приложения на `Horsefly`, выполните эту команду в корне установленного фреймворка: `php artisan app:name Horsefly`

Дальнейшее конфигурирование приложения

Laravel "из коробки" практически не требует дополнительного конфигурирования - вы можете сразу начать писать код. Максимум, вам может быть нужно изменить настройки доступа к базе данных в `config/database.php` и, возможно, изменить параметры `timezone` и `locale` в `config/app.php`. Далее, вам нужно определиться с названием среды выполнения, в которой будет работать ваше приложение на данной машине. Например, когда вы разрабатываете приложение, вам нужно видеть подробный текст возникающих ошибок. По умолчанию фреймворк считает, что выполняется в среде `production` и в ней текст ошибки не выводится. Поэтому вы определяете, что на данной машине у вас среда выполнения `local` и в `config/local/app.php` ставите параметр `debug` в `true`. Примечание – никогда не ставьте `app.debug` в `true` в продакшне, т.е. на хостинге.

Права на запись

Папки внутри `storage` должны быть доступны веб-серверу для записи. Если вы устанавливаете фреймворк на Linux или MacOS - открыть папки на запись можно командой `chmod -R 777 storage`

Пути

Расположение некоторых папок, используемых фреймворком, можно изменить. Пути до таких папок задаются в `bootstrap/paths.php`.

Настройка среды выполнения (the environment)

Часто необходимо иметь разные значения для разных настроек в зависимости от среды, в которой выполняется приложение. Например, вы можете захотеть использовать разные драйвера кэша на локальном и производственном (продакшн) серверах. Это легко достигается использованием настроек, зависящих от среды. Просто создайте новую папку

внутри папки config, название которой совпадает с именем вашей среды, таким как local. Затем создайте файлы настроек, которые вы хотите переопределить, и укажите в них значения для этой среды. Например, вы можете переопределить драйвер кэша для локальной среды, создав файл cache.php внутри config/local с таким содержимым:

```
<?php;
return [;
'driver' => 'file',;
].
```

Не используйте слово 'testing' для названия среды - оно зарезервировано для юнит-тестов. Заметьте, что вам не нужно указывать каждую настройку, которая есть в исходном конфигурационном файле, кроме настроек, которые вы хотите переопределить. Настройки среды будут наложены на эти базовые файлы. Теперь нам нужно сообщить Laravel как определить, в какой среде он работает. Начальная среда всегда production. Однако, вы можете настроить и другие среды в файле bootstrap/environment.php который находится в корне установки Laravel. В этом файле вы найдёте вызов метода \$app->detectEnvironment. Функция, которая ему передаётся, используется для определения текущей среды. В этой функции вы можете определять среду выполнения по имени машины, вызываемому домену или переменной окружения.

```
<?php;
$env = $app->detectEnvironment(function();
return getenv('APP_ENV');
}).
```

В этом примере название среды выполнения у нас находится в переменной окружения APP_ENV. Занести туда значения вы можете в файле .htaccess, правилах nginx или при помощи файла .env, который должен находиться в корне фреймворка. По поводу файла См. соответствующий раздел документации.

Получение текущей среды мы можете получить текущую среду с помощью метода environment объекта Application: \$environment = \$app->environment(); Вы также можете передать аргументы в этот метод чтобы проверить, совпадает ли среда с переданным значением:

```
if ($app->environment('local'))
{
// Среда - local
}
if ($app->environment('local', 'staging'))
{
// Среда - local ИЛИ staging
}
```

\$app можно получить из сервис-контейнера по ключу 'Illuminate\Contracts\Foundation\Application'. Если вы используете вышеприведенный код в сервис-провайдере, то вместо \$app используйте \$this->app.

Настройка сервис-провайдеров

При использовании среды, вы возможно захотите "добавить" дополнительные сервис-провайдеры к исходным в файле app. Однако, если вы это сделаете, вы заметите, что сервис-провайдеры среды переопределяют сервиспровайдеры в вашем исходном файле настроек app. Чтобы заставить сервис-провайдеры не переопределять, а добавляться к существующим, используйте вспомогательный метод `append_config` в вашем файле настроек среды:

```
app:  
'providers' => append_config(array(  
    'LocalOnlyServiceProvider',  
))
```

Защита конфиденциальных данных

Для "настоящих" приложений предпочтительно хранить конфиденциальные данные вне ваших обычных файлов настроек. Такие данные как пароли к базам данных, ключи доступа к различным API-сервисам, ключи шифрования должны храниться вне ваших обычных файлов настроек всегда, когда это только возможно. И где же мы должны их хранить в таком случае? К счастью, Laravel предоставляет очень простое решение для защиты таких настроек - хранение их в "dot" файлах(имена этих файлов начинаются с точки). Во-первых, настройте ваше приложение таким образом, чтобы ваш компьютер определялся в среде local. Далее, создайте файл `.env.php` в корневой папке вашего проекта (в ней обычно находится файл `composer.json`). `.env`-файл содержит список переменных окружения:

```
APP_ENV=local  
DB_USERNAME=homestead  
DB_PASSWORD=homestead
```

Все эти значения будут автоматически доступны в вашем приложении в суперглобальных переменных `$_ENV` и `$_SERVER`. Теперь вы можете обратиться к этим глобальным переменным внутри ваших файлов настроек: `'password' => $_ENV['DB_PASSWORD']`. Убедитесь, что вы добавили файл `.env.php` в ваш файл `.gitignore`. Это позволит другим разработчикам в вашей команде создавать свои собственные локальные файлы настроек, также как и спрятать ваши конфиденциальные данные от системы котроля. Теперь, на вашем производственном (продакшн) сервере, создайте файл `.env.php` в корне вашего приложения, который будет содержать настройки для этого конкретного сервера. Как и файл локальный `.env.php`, этот файл никогда не должен включаться в системуконтроля версий (`git`, `mercurial`, `svn` и т.п.). Вы

можете создавать файлы для каждой среды вашего приложения. Например, среда `development` будет всегда загружать файл `.env.development.php`, если он существует. Однако, среда `production` (среда производства) всегда использует файл `.env.php`.

Режим обслуживания

Когда ваше приложение находится в режиме обслуживания (`maintenance mode`), специальный шаблон будет отображаться вместо всех ваших маршрутов. Это позволяет "отключать" ваше приложение во время обновления или обслуживания. Проверка на режим обслуживания уже включена в стандартный фильтр `App::before` в файле `app/Http/Filters/MaintenanceFilter.php`. Ответ от этой проверки будет возвращен пользователю когда приложение находится в режиме обслуживания. Для включения этого режима просто выполните команду `Artisan-a down: php artisan down`. Чтобы выйти из режима обслуживания выполните команду `up:php artisan up`

Режим обслуживания и очереди

Пока ваше приложение находится в режиме обслуживания, очереди не будут обрабатываться. Работа очередей будет возобновлена когда приложение выйдет из режима обслуживания.

-Красивые URL;

-Apache;

Laravel поставляется вместе с файлом `public/.htaccess`, который настроен для обработки URL без указания `index.php`. Если вы используете Apache в качестве веб-сервера обязательно включите модуль `mod_rewrite`. Если стандартный `.htaccess` не работает для вашего Apache, попробуйте следующий:

-Options +FollowSymLinks;

-RewriteEngine On;

-RewriteCond %{REQUEST_FILENAME} !-d;

-RewriteCond %{REQUEST_FILENAME} !-f;

-RewriteRule ^ index.php [L];

-Nginx.

Если вы используете в качестве веб-сервера Nginx, то используйте для ЧПУ следующую конструкцию: `location / { try_files $uri $uri/ /index.php?$query_string;`

Если вы используете Homestead, то вам ничего делать не нужно, там всё это уже настроено.

Использование MVC

Вид - это шаблон. Он не содержит логики. Контроллер - блок, который получает данные от пользователя. Обрабатывает, нормализует, проверяет правильность ввода. И передает данные в модель. Контроллер не должен содержать никакой информации о внешнем виде приложения. Модель - основа логики приложения. Отвечает за расчеты, выборку информации из БД, изменение информации в БД и т. д. Моделью можно представить библиотеку функций.

Схема работы

Все запросы пользователей перенаправляются сервером в глобальную точку входа по адресу `public/index.php`. Файл `index.php` загружает созданный Композером автозагрузчик классов при помощи файла `bootstrap/autoload.php`. В `index.php` создается объект глобального класса фреймворка Ларавель с использованием файла `bootstrap/app.php`. Запрос поступает в ядро обработки `http`-запросов - класс с именем `Kernel`. `app/Http/Kernel.php`. Класс `Kernel` наследует свой функционал от родительского класса `Kernel`. Чтобы не было конфликтов для родительского ядра назначен псевдоним `HttpKernel`. Путь к нему: `\vendor\laravel\framework\src\Illuminate\Foundation\Http\Kernel.php`. В этом классе определяется массив `bootstrappers`, где указан список классов, которые должны запускаться перед выполнением запроса.

Ядро `app/Http/Kernel.php` принимает в качестве входного параметра запрос и отдает ответ. Это видно в точке входа `index.php $response->send()`; Как только все сервис-провайдеры зарегистрированы и приложение загружено, запрос пользователя поступает в Роутер. Фреймворк решает какой маршрут обработает тот или иной запрос. В качестве обработчика маршрута может быть назначен определенный контроллер или функция. Если обработчик - контроллер, то запрос передается контроллеру. Он выполняет необходимые преобразования данных и обращается к модели с запросом данных, которые должны быть отображены на экране. Если нужны данные из БД, то модель связывается с БД и выбирает данные, приводит их к определенному виду и возвращает контроллеру. Контроллер передает их виду, который отображает данные на экране. Если обработчиком выступает функция, то и данные из функции роутера передаются в вид. Приложение на Laravel можно разрабатывать на любой операционной системе. Можно использовать IDE (интегрированную среду разработки) на свой выбор, например PhpStorm (любители текстовых редакторов могут выбрать Atom или Sublime Text). Здесь еще может пригодиться Laravel IDE Helper Generator – пакет, который генерирует файл-хелпер, в котором содержатся статические классы фасадов. Вторым полезным инструментом станет Composer, который позволяет обновлять фреймворк и загружать в проект дополнительные пакеты. Его можно скачать по этой ссылке. Большинство PHP-пакетов имеют минимум зависимостей и поэтому могут быть легко добавлены в приложение. При написании любого проекта его нужно тестировать – в Laravel представлены функциональные тесты (Feature-тесты), проверяющие

функционал проекта (с точки зрения пользователя), и модульные тесты (Unit-тесты), которые проверяют саму логику проекта. Еще один полезный инструмент - Laravel Debugbar, пакет, который позволяет контролировать и отлаживать код (интегрируется в PHP Debug Bar). Он отслеживает запросы, например, позволяет отследить SQL-запросы для их оптимизации. Для сборки проекта понадобится Laravel Mix. Этот API используется для определения инструкций сборки Webpack для Laravel приложения. Laravel идет в ногу со временем, он изменяется и дорабатывается, это современный фреймворк, подходящий для широкого круга задач. Laravel позволяет использовать сервис-провайдер (service provider), благодаря которому можно централизованно подключать необходимые компоненты приложения. В Laravel можно легко расширять любые компоненты. Также отдельно можно отметить удобную маршрутизацию и валидацию входящих параметров. Laravel дает возможность работать с различными базами данных, изменять их структуру, откатывать изменения и т.д.

Сайты на Laravel отличаются:

- широким функционалом (можно сделать проект с практически любым необходимым функционалом);
- удобной административной панелью (можно сделать панель конкретно под определенный проект и его задачи);
- высоким уровнем безопасности баз данных (сайты надежно защищены от SQL-инъекций);
- масштабируемостью (функционал проекта можно легко расширить). На базе Laravel было разработано несколько CMS.

Перед началом разворачивания фреймворка убедитесь, что на компьютере установлены утилиты Composer и NodeJS. У второго использую версию 6 ветки, хотя это не принципиально.

Перед установкой фреймворка, рекомендую воспользоваться пакетом hirak/prestissimo, позволяющим загружать несколько пакетов при установке/обновлении фреймворка одновременно. Без него процесс установки/обновления линейный, то есть запрашивает один пакет, ждет ответ, после скачивает (если ответ от сервера успешный), далее переходит к следующему. Пакет `hirak/prestissimo` же позволяет одновременно скачивать все необходимые пакеты и уже после этого начинает их ставить.

Для установки пакета выполните в консоли:

```
composer global require "hirak/prestissimo:^0.3".
```

Итак, открываем командную консоль и вводим: `composer global require "laravel/installer"`. Это нужно делать всего один раз при самой первой установке. При разворачивании последующих приложений, данную команду пропускаем. После того, как скачали установщик фреймворка в глобальное хранилище композера, нам нужно скачать сам фреймворк Laravel 5.3. Для этого у нас есть две команды на выбор: `laravel new blog` `composer create-project --prefer-dist laravel/laravel`

blog. Они обе выполняют одно и то же действие, разница лишь... И так видно в чем разница). Единственный нюанс, при первой установке путь к исполняемому файлу композера прописывается автоматически при его установке, а путь к команде «laravel» — нет. Лично мне удобно использовать команду «laravel», поэтому идем в «Этот компьютер» (пользуюсь Windows 10, поэтому все шаги буду описывать под нее). Открываем «Параметры» и в поисковой строке вводим «Система», далее «Изменить параметры». В окне «Свойства системы» переходим во вкладку «Дополнительно» и жмем кнопку «Переменные среды». В открывшемся окне, в переменных средах для пользователя, открываем изменение `PATH`, где добавляем путь к файлу исполняемому `laravel`. У меня он находится в папке `c:\Users\Developer\AppData\Roaming\Composer\vendor\bin\`, где «Developer» — имя учетной записи (пользователя). Скриншоты с подробностями. Итак, выполняем одну из команд:

- laravel new blog
- composer create-project --prefer-dist laravel/laravel blog

Фреймворк скачается в папку «blog». Вы можете сразу прописать адрес Вашего сайта. Для этого видоизменим команду: laravel new mysite.dev composer create-project --prefer-dist laravel/laravel mysite.dev. После этого необходимо перезапустить `OpenServer`, чтобы он «увидел» новый сайт. Все. На установка завершена и переходим к следующему этапу. Так как многие пользуются репозиториями, сразу идем в файл `.gitignore`, приводя его к виду:

```
-.env;  
-/vendor;  
-/node_modules;  
-/public/storage;  
-/.idea;  
-Homestead.yaml;  
-Homestead.json;  
-*.npm-debug.log*
```

То есть, мы включаем в запрет выгрузки информации из node_modules и автогенерируемых файлов из `public/build` и другие «системные» файлы. Содержимое папок `vendor` и `node_modules` подгружаются автоматически. Environment (.env). Здесь все достаточно просто:

```
-APP_ENV=local;  
-APP_DEBUG=true  
-DB_CONNECTION=pgsql;  
-DB_HOST=127.0.0.1;  
-DB_PORT=5432;  
-DB_DATABASE=my_database;  
-DB_USERNAME=root;  
-DB_PASSWORD=;
```


-`DB_CONNECTION` — имя подключения соответствующей базы данных из конфигурации `config/databases.php >> connections`; В случае с PostgreSQL указываем `pgsql`, а в случае с MySQL/MariaDB — `mysql`. Также не забываем указывать порт, используемый для базы данных. Номер порта можно посмотреть в `OpenServer` в на вкладке `Сервер`. Перед этим, воспользовавшись приложением `PgAdmin III` из состава `OpenServer`, необходимо добавить базу данных и прописать параметры в файле `.env`.

Для работы нам понадобятся следующие пакеты:

```
-graham-campbell/exceptions;  
-barryvdh/laravel-debugbar;  
-barryvdh/laravel-ide-helper;  
-laracasts/generators.
```

Laravel Exceptions

Пакет нужен для улучшенного вывода информации о возникающих ошибках при разработке в режиме дебага. Установка проста: поочередно выполняем в консоли команды:

```
-composer require graham-campbell/exceptions;  
-composer require filp/whoops -dev;
```

Далее, в файле `config/app.php` добавляем сервис-провайдер `GrahamCampbell\Exceptions\ExceptionsServiceProvider::class` в блок `providers`. После этого, в файле `App\Exceptions/Handler.php` в блоке `use` меняем `use Illuminate\Foundation\Exceptions\Handler as ExceptionHandler;` на `use GrahamCampbell\Exceptions\NewExceptionHandler as ExceptionHandler;`

Далее, в файле `config/app.php` добавляем сервис-провайдер `Barryvdh\Debugbar\ServiceProvider::class,` в блок `providers`.

Установка Elixir

В консоли, находясь в папке проекта, поочередно выполняем команды:

```
-npm install --global gulp-cli;  
-npm install --no-bin-links.
```

Так как, еще со времен Laravel 5.2, пользуюсь VueJs, который в 5.3 идет «из коробки», подключим и его со всеми необходимыми пакетами:

```
-npm rebuild node-sass  
-npm install hammerjs vue-async-data materialize-css
```

Обратите внимание: такие компоненты, как `vue` и `vue-resource` не вписал, так как они идут «из коробки». Далее, в файле `resources/assets/js/bootstrap.js` после строки `window.\$ = window.jQuery = require('jquery');` найдете подключение CSS-фреймворка Twitter Bootstrap (`require('bootstrap-sass');`).

Так как в этом примере используем не его, а MaterializeCSS — заменяем на `require('./vendor/materialize-css/js/bin/materialize.min');` Далее, находим:

```
-window.Vue = require('vue');  
-require('vue-resource');
```

И приводим к виду:

```
-window.Vue = require('vue');
-var VueResource = require('vue-resource');
-var VueAsyncData = require('vue-async-data');
-Vue.use(VueResource);
-Vue.use(VueAsyncData);
```

MaterializeCSS

Давно перешел с CSS-фреймворка Twitter Bootstrap в пользу «материального» дизайна MaterializeCSS, так как лично для меня он выглядит более современным. Для его установки необходимо выполнить команду:

`-npm install materialize-css`. Далее, необходимо подключить используемый MaterializeCSS шрифт `Material Icon` в шаблоне:

```
<!DOCTYPE html>
<html>
<head>
  <!--Import Google Icon Font-->
  <link href="http://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
</head>
<body>
  //
</body>
</html>
```

Настройка консоли

При разработке очень удобный инструмент браузера — консоль. Чтобы организовать «быстрый поиск» проблемной строки, в файл `gulpfile.js` добавляем строку `elixir.config.sourcemaps = true;`, отвечающую за генерацию «сурс-мапа». При работе с MaterializeCSS и SASS, у меня он имеет вид:

- Рабочий gulpfile.js.

Resources: SASS

Так как не все компоненты из состава фреймворка MaterializeCSS нужны для разработки того или иного проекта, добавляем ссылки на них в наш файл `resources/assets/sass/app.scss`:

Содержимое app.scss

Итак, в ходе предыдущих действий, выполнив команду `gulp` в консоли, на выходе мы получим 2 файла: `app.js` и `app.css`. Так как была использована система контроля версий (не стоит же разглагольствовать на тему кэширования ресурсов браузером, да?), в шаблоне `resources/views/layouts/app.blade.php` необходимо прописать:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
```

```

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
{{!--<!-- CSRF Token -->--}}
<meta name="csrf-token" content="{{ csrf_token() }}">
{{!--<!-- Import Google Icon Font-->--}}
<link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
{{!--<!-- Styles -->--}}
<link href="{{ elixir('css/app.css') }}" rel="stylesheet">
</head>
<body>
  // Content
  {{!--<!-- JavaScripts -->--}}
  <script src="{{ elixir('js/app.js') }}"></script>
</body>
</html>

```

Очень часто бывает так, что переименовали какой-нибудь класс и у нас сразу отказывает работать команда `composer update`.

При большинстве ошибок, возникающих при запуске консольных команд, помогает команда:

```

-composer dump-autoload;
-composer dumpautoload.

```

Также, полезным будет посмотреть список всех `artisan`-команд:

```
php artisan list
```

И, конечно же, список активных роутов:

```
-php artisan route:list
```

Репозиторий

Клонируем репозиторий:

```
git clone https://github.com/andrey-helldar/laravel-example-to-juniors.git
```

И поочередно выполняем команды:

```

-composer install;
-npm install;
-gulp

```

1.3 СОПРОВОЖДЕНИЕ ПРОГРАММНОГО ПРОЕКТА

Фреймворк, будучи "каркасом" для создания и сопровождения программного проекта, облегчает задачу разработчика. Существует множество фреймворков для создания сайтов и для различных языков программирования, обладающих как плюсами, так и минусами. Хорошо известные CSS-фреймворки: Foundation и Bootstrap. Из современных PHP-фреймворков можно выделить Yii, Symfony и Laravel. Они дружелюбны не только к профессионалам, но и новичкам. Популярность и широкий функционал позволяет без особого труда найти полезную информацию по этим фреймворкам. В наше время IT-технологии не стоят на месте, каждую минуту создаются новые сайты, приложения и разного рода программные продукты. Тысячи строк кода помогают компьютеру понять человека. Зачастую, разработчик применяет одни и те же алгоритмы в разных проектах. И было бы не целесообразно тратить уйму времени на то, чтобы раз за разом писать одно и то же. Что же может облегчить эту задачу. Существует множество определений термина «фреймворк». Рассмотрим некоторые из них и попытаемся найти то, что их объединяет.

Framework – это программная платформа, определяющая структуру программной системы, программное обеспечение, которое облегчает поставленную перед разработчиком задачу и объединяет разные компоненты большого программного проекта. Это такая структура, включающая в себя библиотеки кода, различное программное обеспечение, которые облегчают жизнь разработчика и способствуют объединению разных составляющих большого программного проекта. Фреймворк содержит базовые программные модули. Все специфичные компоненты реализуются разработчиком на их основе. В переводе с английского, framework означает «каркас». Действительно, если сравнить создание программного проекта со строительством дома, то фреймворк – каркас и разработчику остается лишь навесить окна и стены. Фреймворк предлагает разработчику уже встроенные классы:

- Для работы с базой данных;
- Для создания функциональных форм;
- Для описания логики и др.

Все это может использоваться в большинстве ваших проектов, при этом использование и подключение будет простым благодаря таким «каркасам».

Используя фреймворки не нужно писать с нуля десятки, а то и сотни тысяч строк, тем самым достигается большая производительность. Становится проще сопровождать проект, вносить изменения и исправлять ошибки, «склеивать» компоненты в более сложные системы. Фреймворк помогает организовать структурированную архитектуру проекта, т.е. все файлы будут находиться в порядке и на нужных местах.

Плюсы фреймворков

- Разработка на фреймворке позволяет добиться простоты сопровождаемости проекта.
- Проекты легко масштабируются и модернизируются.
- Решения на фреймворках работают гораздо быстрее.
- Решения на фреймворках превосходят самописные системы по безопасности.
- Минусы фреймворков
- Многие функции реализовываются индивидуально, т.к. фреймворки содержат только базовые компоненты бизнес-логики.
- Требуется понимание бизнес-процессов.
- Web-разработка и фреймворки

Одна из самых популярных областей применения фреймворков – создание сайтов, web-приложений и сервисов и это естественно. Главным преимуществом фреймворка является тот факт, что зачастую веб-сайты и приложения используют стандартизированную структуру организации компонентов. И создание такой структуры становится проще, если использовать фреймворки. Практически у любого сайта есть форма для авторизации. И каждый раз «изобретать велосипед» было бы не разумным решением. В интернете мы можем найти огромное количество фреймворков для создания сайтов и для разных языков программирования. Одни предназначены для упрощения в разработке интерфейсов, другие помогают в написании логики программного проекта.

В пример хотелось бы привести наиболее популярные фреймворки, с которыми сможет разобраться как профессиональный разработчик, так и новичок. Практически невозможно не найти какую-либо информацию по этим фреймворкам и это их огромное преимущество. Одним из самых знаменитых HTML/CSS/JS-фреймворков, помогающих в разработке интерфейса сайта является Foundation, который состоит из CSS файлов и нескольких плагинов JQuery(JS-фреймворк). Главным его конкурент - Bootstrap, на котором так же написано огромное количество проектов.

Стоит отметить основные плюсы данных фреймворков:

- Удобство;
- Простота для новичков;
- Популярность, а значит развитое сообщество;
- Функционал;

Другие фреймворки облегчают написание логики сайта и web-приложения или сервиса. Например, одними из самых популярных PHP-фреймворков являются:

-Yii : прост в освоении и использовании, высокая производительность относительно других php-фреймворков и пр. возможности.

Symfony: мощная функциональность, развитое сообщество, большое преимущества перед другими php-фреймворками в разработке сложных проектов.

-Laravel: доступность, мощьность, хороший функционал.

В заключение данного раздела, стоит отметить, что каждый из описанных фреймворков имеет как сильные, так и слабые стороны. Все они обладают большими возможностями, которые реализованы по-своему. У каждого разработчика свои предпочтения, поэтому выбор будет субъективен. Прежде чем решить, какой же из них лучше, разработчику стоит самому попробовать каждый из этих фреймворков.

2 РЕАЛИЗАЦИЯ ЭЛЕКТРОННОЙ ТОРГОВОЙ ПЛОЩАДКИ

2.1 ПОСТАНОВКА ЗАДАЧИ

Создавая Электронно-Торговую площадку и проектируя систему защиты пользовательских данных, важно поставить правильно перед собой следующие задачи:

- ознакомиться с технологиями защиты пользовательских данных;
- выявить недостатки созданных электронных торговых площадок;
- изучить внутренние возможности реализации проекта;
- изучить внешние возможности программного обеспечения, на базе которой создается портал;
- сформулировать требования по безопасности;
- предоставить информационную безопасность для пользователей электронного торгового портала;
- внедрить и изучить методы создания системы защиты данных пользователей.

2.2 ТРЕБОВАНИЯ К СОЗДАНИЮ ЭЛЕКТРОННО-ТОРГОВОЙ ПЛОЩАДКИ И ПРОЕКТИРОВАНИЮ ЗАЩИТЫ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ

Проектируемая система защиты пользовательских данных должна обеспечить защиту пользователя в системе, система которой будет установлена на виртуальной сети. Также должна обеспечить конфиденциальность, целостность и доступность.

Система защиты должна выполнять следующие функции:

- обеспечить безопасность конфиденциальных пользовательских данных на уровне операционной системы;
- обеспечить защиту несанкционированного доступа;
- защитить систему от атак из Интернета;
- мониторинг операционной системы.

Как правило большинство организации при обеспечении безопасности информационных систем затраты на безопасность не могут превышать экономические потери, т.е. ущерб от реализации возможных угроз

безопасности. В связи с этим, я использовал бесплатные возможности для реализации практической части дипломной работы.

В качестве рекомендаций по дальнейшей оптимизации системы, я предлагаю рассмотреть приобретение UTM-решения – универсальный шлюз безопасности, объединяющий в себя функции межсетевого экрана, антивируса, системы обнаружения и предотвращения вторжений, системы глубокой проверки пакетов.

2.3 ОБЪЕКТ ЗАЩИТЫ

Электронно-торговая площадка - единый центр оборудования, техники и технологий, который на основе представительства оказывает ряд услуг, одной из которых является предоставление электронно-торговой площадки. В данной дипломной работе демонстрируется реализация электронно-торговой площадки, мер безопасности в нем.

Ключевые слова: информационная безопасность, центр оборудования, техники и технологий.

Электронно-торговая площадка – центр, который на основе представительских и дистрибьюторских прав обеспечивает сопровождение контрактов, доставки, процессов пуско-наладки, сервисного и гарантийного обслуживания оборудования со всего мира.

Портал состоит из основных обновляемых 4 разделов:

1. главная страница (рисунок 1);
2. торговая площадка;
3. новости (рисунок 2);
4. контакты.

2.4 ПРОЕКТИРОВАНИЕ ПЛОЩАДКИ

Предусмотрены ролевые ограничения, которые связаны с предотвращением загрузки пользователями запрещенных материалов, предусмотренных законодательством Республики Казахстан. Из разделов, относящихся к модулю авторизации существуют:

- вход;
- регистрация.

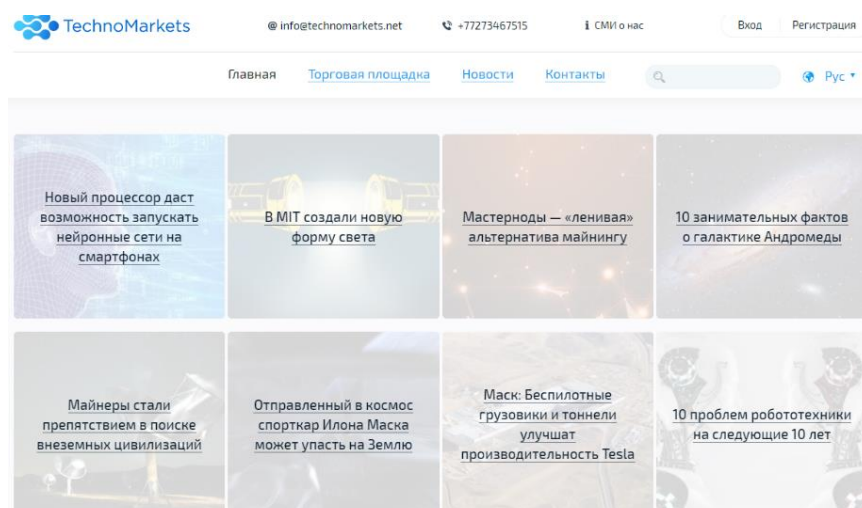


Рисунок 1 – Главная страница Technomarkets

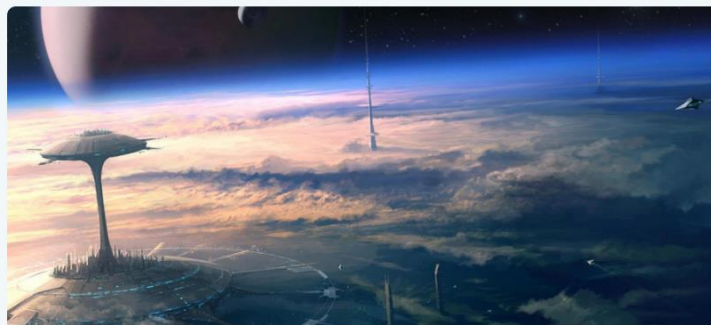
В модуле «Пользователи» проходят регистрацию с указанием имени, наименовании электронной почты, контактного телефона. После правильного заполнения анкеты пользователю выдается личный аккаунт. Далее успешно пройдя аутентификацию, пользователь может использовать личный кабинет для своих целей с возможностью ее редактирования.

Главная страница состоит из сегментов, в которых расположены логотип, почта, ссылка и модули авторизации (вход/регистрация пользователей), инструкция с подробными шагами по использованию Technomarkets.net. Также есть новостной блок, в котором находятся 8-разделов с режимом слайд-шоу. Для каждого раздела проводится загрузка информации через административную часть портала.

Раздел электронно-торговой площадки является одной из самых значимых частей. В нем сосредоточены физические и юридические лица, которые могут добавлять и просматривать загружаемые объявления пользователей. На данной странице можно увидеть количество зарегистрированных пользователей и объявлений с указанием контактных данных, адресов и всей необходимой информации. Форма поиска работает по заданным параметрам пользователя. Есть возможность выбора необходимой страны, города, а также заданного текста, которого можно найти по всем объявлениям, начальной и конечной цены в выбранной валюте (тенге, рубль, доллар, евро). Не ограничиваясь обычным поиском, я добавил расширенный поиск, благодаря которому можно найти более детальную информацию. В этом же разделе находится блок для баннеров (их загрузка проводится через административную часть).

Новости

[Главная](#) → [Новости](#)



10 технологий будущего, которые обязательно изменят этот мир

С развитием технологий мы начинаем замечать, как отвечает и начинает изменяться вместе с этим развитием окружающий нас мир. Несмотря на то, что многие технологии в этом списке уже начинают потихоньку применяться, некоторые по-прежнему остаются лишь в своем зачаточном состоянии. Мы находимся на пороге...

Компании и их рейтинги

1. TOO Арти-строй
2. TOO Трубы-23
3. TOO Электрон-Снаб
4. TOO MediaEquipment
5. TOO MedicalFirst
6. TOO FirstPlace
7. TOO Qlasterkz
8. TOO Arystanov
9. TOO AziyaPromCo
10. TOO ЦентрТехно

19.02.2018

19.02.2018

19.02.2018

Рисунок 2 – Новости

Раздел «Новости» создан с инициацией ознакомления пользователей со всеми трендами в области технологий. Кроме загрузки новостей, также отдаю предпочтение загрузки научным статьям. В правой части расположен рейтинг зарегистрированных пользователей, который работает на основе оценивания другими пользователями портала Technomarkets.net. В новостном разделе загрузка новостей и статей происходит на русском и английском языках. В силу того, что в области оборудования, техники и технологий большое количество отраслей и категорий, было решено создать раздел “Категории” (рисунок 3). Администратор имеет возможность добавлять необходимую категорию с привязкой к родительской отрасли. Действия, предусмотренные в разделе: удаление и изменение данных.

id	Родительская категория	Название	Ссылка	Действия
128	Оборудование	Страхование	strahovanie	Удалить Изменить
127	Оборудование	Прочее оборудование	prochee-oborudovanie	Удалить Изменить
126	Оборудование	Сертификации	sertifikacii	Удалить Изменить

Рисунок 3 – Раздел Категории

Раздел «Контакты» создан с инициацией ознакомления пользователей с технической поддержкой портала. Данный портал будет модернизирован таким образом, чтобы любой пользователь мог получить ответ на свои вопросы в кратчайшие сроки. Каждому администратору будет присвоен номер и цель вопроса, благодаря чему можно ускорить решение к различным вопросам (рисунок 3.1)

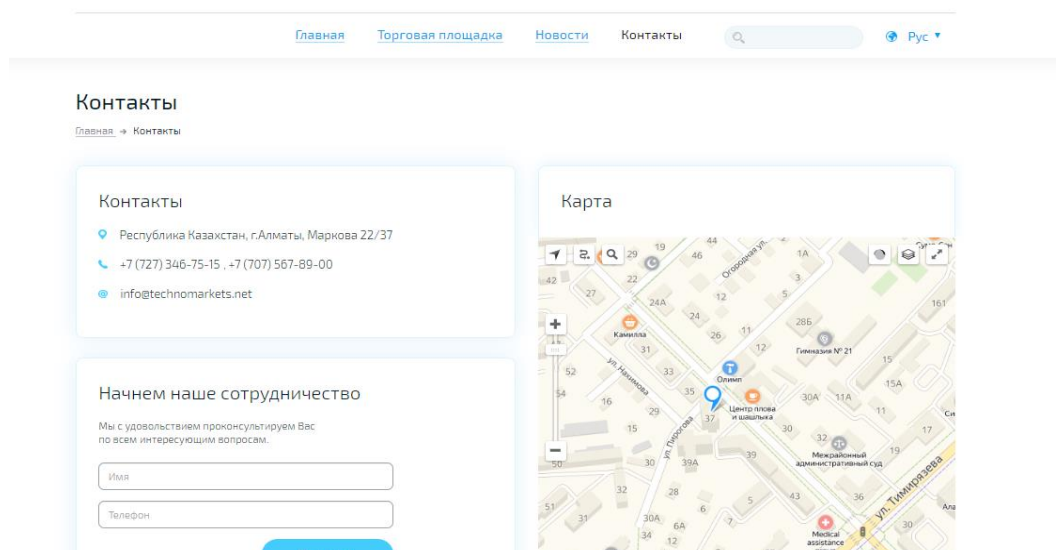


Рисунок 3.1 – раздел Контакты

Так как Technomarkets работает не только для городов Казахстана, но и СНГ и Европы, было решено создать отдельный раздел стран и городов в административной части для более детального изучения спроса и потребления услуг регионами (рисунок 4).

Панель управления > Locations

Страны и города Добавить

Show 10 entries Search:

id	Название	Родительская категория	Действия
2702	Нерлес	Финляндия	Изменить Удалить
2701	Наантали	Финляндия	Изменить Удалить
2700	Миккели	Финляндия	Изменить Удалить
2699	Мариехамн	Финляндия	Изменить Удалить
2698	Лоймаа	Финляндия	Изменить Удалить

Рисунок 4 – Раздел регионов

На главной странице пользователи (зарегистрированные и не зарегистрированные) могут оставлять заявки на необходимые запросы. После регистрации заявки автоматически загружаются на форму в админ-панели (рисунок 5). Данная форма позволяет ускорить процессы обработки заявок. При модернизации я предусматриваю внедрение автоматизации поиска запросов, которая будет привязана к поисковикам масштабного уровня (google, yandex, mail и т.п.).

Заказы с формы Добавить

Show 10 entries Search:

id	Название	Имя	Email	Телефон	Город	Категория	Статус	Дата создание	Действия
61	Компоненты для напыления пенополиуретана	Арман			Астана	Строительные технологии	обрабатывается	2018-02-28 03:55:14	Удалить Изменить
60	Дизельный генератор 250квт	Ерлан			Актау	Энергетическое оборудование	обрабатывается	2018-02-23 12:38:18	Удалить Изменить
55	Робот-каменьщик, Модель: Hadrian X, Производитель: FastBrick Robotics, Контакты: +61 (08) 9380 0240, Адрес:122 Sultana Rd West, High Wycombe, Western Australia 6057. Хочу узнать стоимость, сроки поставки и возможно ли купить в марте 2018 г.	Роман			Астана	Другие виды техники	обрабатывается	2018-01-08 10:06:38	Удалить Изменить

Рисунок 5 – Заказы с формы

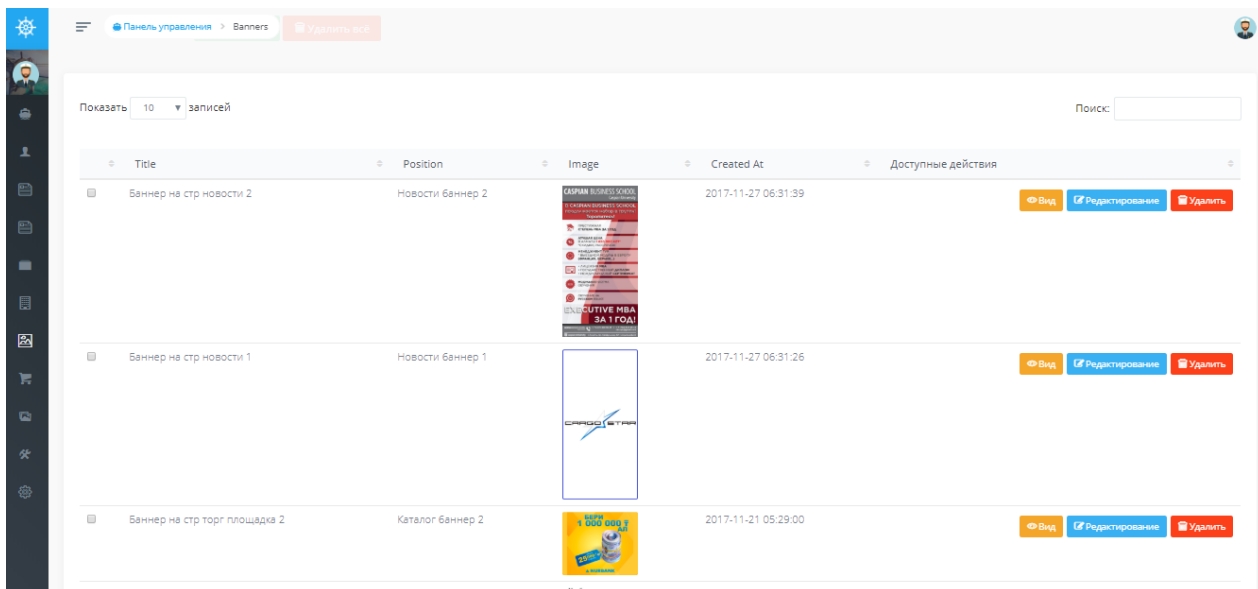


Рисунок 5.1 – Раздел баннеров панель управления

В административной панели есть возможность добавления баннеров, которые в последующем будут располагаться в различных разделах портала. (Рисунок 5.1). Функции вида, редактирования и удаления позволят администраторам правильно расположить баннера на электронно-торговой площадке.

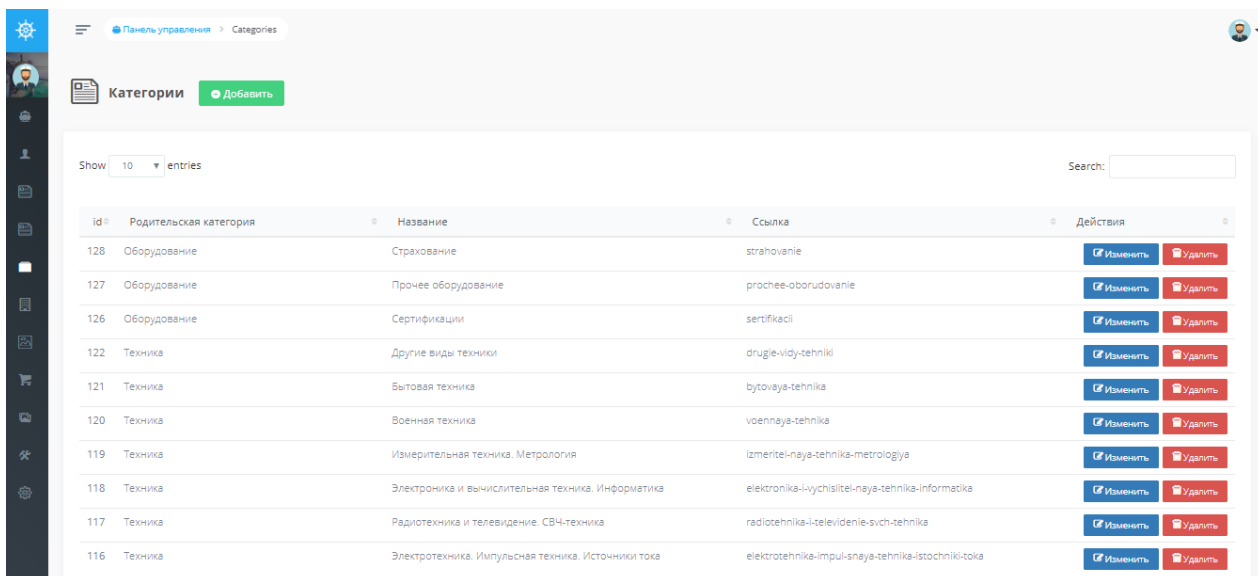


Рисунок 5.2 – Раздел категорий в панели управления

Также в административной панели есть возможность добавления категорий. У каждой категории с наименованием есть родительская категория. Раздел категорий состоит из области оборудования, техники и технологий. (Рисунок 5.2). Функции в панели управления позволят администраторам правильно произвести сортировку категорий на торговой площадке.

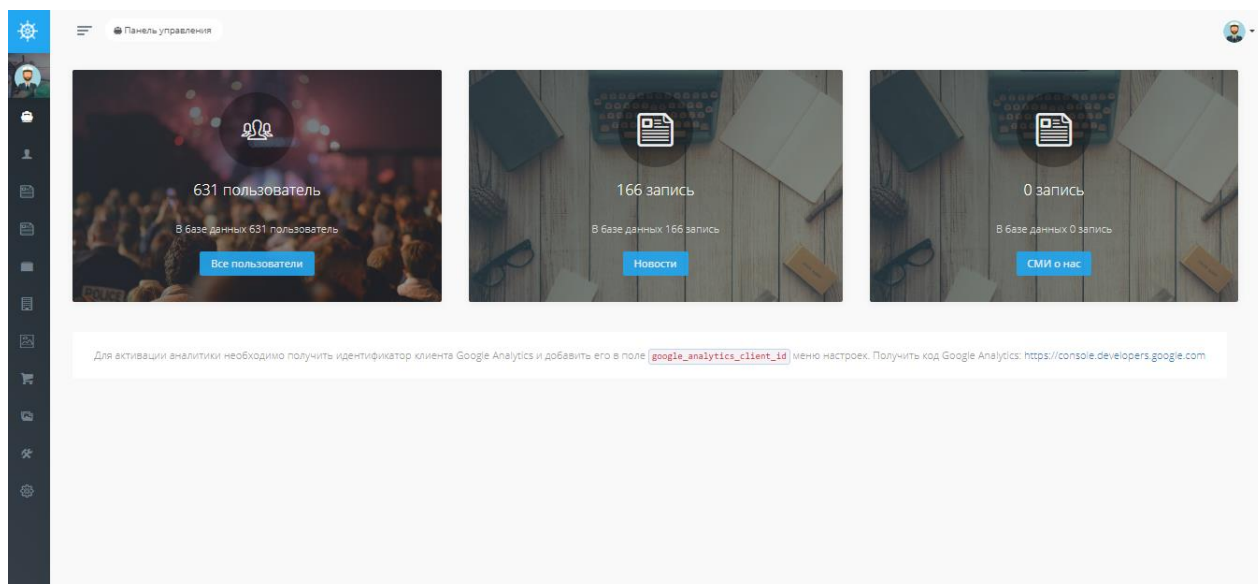


Рисунок 5.3 – Раздел dashboard в панели управления

В данном разделе dashboard можно увидеть количество зарегистрированных пользователей, записей в разделе «новости» и «СМИ о нас». Для активации полной аналитики необходимо получить идентификатор клиента Google Analytics и добавить его в поле `google_analytics_client_id` меню настроек. Получить код Google Analytics возможно на официальном сайте console.developers.google.com (рисунок 5.3).

3 СОЗДАНИЕ И НАСТРОЙКА ЗАЩИТЫ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ

В целях обеспечения информационной безопасности Technomarkets была внедрена политика разграничения ролей администрирования (рисунок 6). Администратор-1 отвечает за загрузку контента. Администратор-2 отвечает за проверку загружаемого контента, Администратор-3 за принятие заказов с формы главной страницы, Администратор-4 за управление и контроль загружаемых данных пользователями, Администратор-5 за редактирование информации на торговой площадке. Контент должен быть сформирован согласно правилам Technomarkets и не противоречить нормам законодательства Республики Казахстан. Таким образом настроено разделение полномочий администраторов, отвечающих за автоматизацию (использование средств информационно-коммуникационных технологий для оптимизации создания, поиска, сбора, накопления, хранения, обработки, получения, использования, преобразования, отображения, распространения и предоставления информации) и информатизацию (для автоматизации деятельности субъектов информатизации). В дальнейшем я предусматриваю

аудит информационной системы в целях повышения эффективности ее использования.

Для редактирования возможностей администраторов имеется своя админ-панель, которая позволит ограничить действия тех или иных администраторов (рисунок 7). Также присутствует функционал создания ролей. Пример: добавление “Администратора-6” с “ролью-6”.

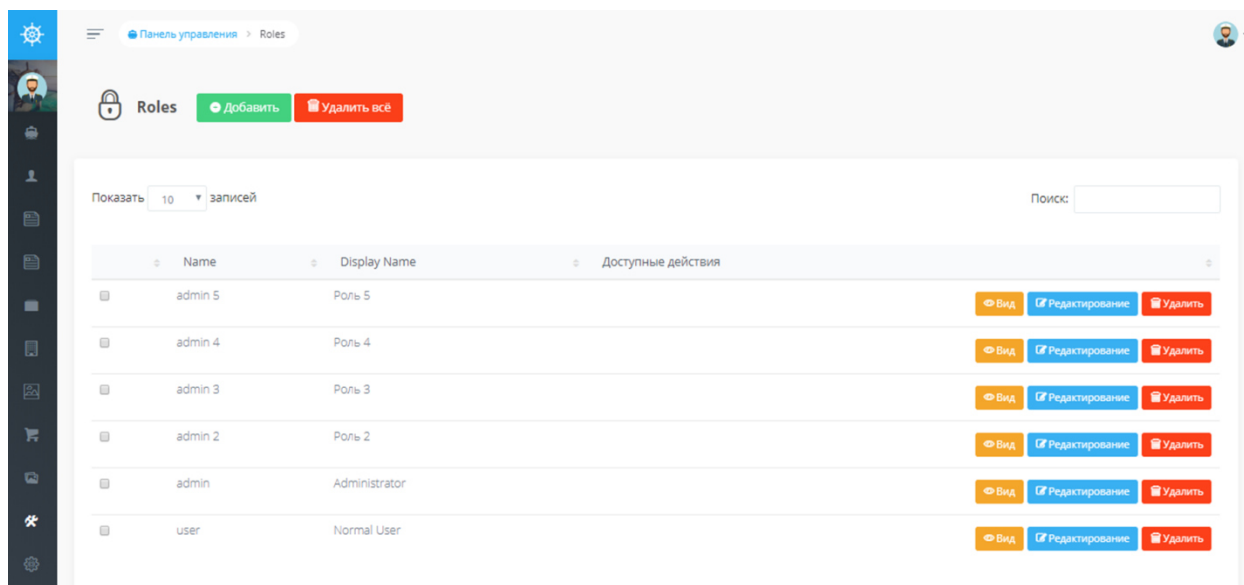


Рисунок 6 – Администрирование

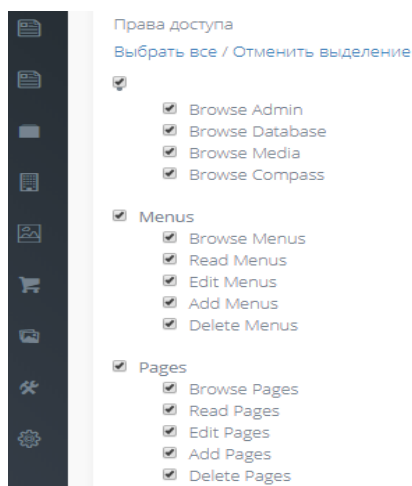


Рисунок 7 – Права доступа

3.1 СОЗДАНИЕ И НАСТРОЙКА ПЛАТФОРМЫ РЕАЛИЗАЦИИ ЭЛЕКТРОННОЙ ТОРГОВОЙ ПЛОЩАДКИ С СИСТЕМОЙ ЗАЩИТЫ

3.1.1 Используемый фреймворк

У каждого фреймворка есть огромное количество поклонников и противников. При выборе фреймворка необходимо учитывать несколько факторов. Во-первых, фреймворк должен иметь ровно столько возможностей, сколько необходимо проекту. Излишний функционал не нужен, но при этом никогда не должно возникать ограничений при доработке проекта. Во-вторых, время разработки сайта не должно увеличиваться за счет выбора того или иного фреймворка. В-третьих, фреймворк должен быть достаточно распространен, для того чтобы проект можно было передать другому разработчику. Наконец, безопасность фреймворка - один из важнейших показателей. Все фреймворки хороши, если находятся в правильных руках.

Ключевые слова: информационная безопасность, центр оборудования, техники и технологий.

Проанализировав рынок СНГ в сфере оборудования, техники и технологий мне удалось узнать, что нет единого центра, который мог бы сосредоточить всех физических и юридических лиц, а также экспертов в области оборудования, техники и информационных технологий. После длительных размышлений, анализов и взвешивания рисков я понял, что рынок востребован в проекте под названием «TechnoMarkets», который я реализовывал в течение шести месяцев.

Создавать заново велосипед мне не хотелось и потому я решил использовать «движок» для этих целей. Мое внимание привлек Laravel 5.5, который был выпущен 30 августа 2017 года.

3.1.2 Возможности веб-фреймворка – Laravel

Исследуя возможности Laravel, я понял, что данный веб-фреймворк подходит как нельзя лучше для создания моего портала «TechnoMarkets». Laravel включает поддержку планирования периодически выполняемых задач через пакет Scheduler, слой абстракции Flysystem, который позволяет использовать удаленное хранилище также, как и локальные файловые системы, улучшенную обработку активов пакета через Elixir и упрощенная аутентификация с внешней стороны через дополнительный пакет Socialite.

В выпуске 5.5 есть много функций, таких как Laravel Dusk, Laravel Mix, Blade Components и Slots, Markdown Emails, автоматические фасады, улучшения маршрута, которые мне отлично подходили для разработки. Есть пакет Whoops для отладки ошибок, что очень удобно в корректировке тех или иных кодов. Существуют метод обработки запроса, новые директивы для шаблонизатора Blade, команда для миграции баз данных - migrate:fresh. Все

данные возможности позволили понять саму работу по модернизации и структуру «движка» в целом.

Ниже опишу возможности «движка» и каким образом я их использовал.

Информативная документация. Начать я решил с особенности Laravel, с которой неизбежно сталкиваются все разработчики при освоении новой технологии. Это документация Laravel, которая, по моему мнению, является очень хорошей и структурированной. Думаю, что это также добавляет популярности данному движку среди разработчиков. По крайней мере, в моём случае это правда.

MVC структура кода. Данный шаблон проектирования зарекомендовал себя как проверенное временем решение эффективной структуры приложений. MVC позволяет делать код более читабельным, а процесс разработки комфортным, разграничивая работу frontend- и backend-разработчиков.

Artisan - консоль Laravel, в арсенале команд которой есть работа с миграциями, контроллерами и моделями, авторизацией и прочими базовыми компонентами фреймворка.

Миграции. Своего рода, контроль версий для структуры таблиц базы данных (БД). Каждый файл содержит либо структуру таблиц, либо изменения её структуры. Пример создания таблицы пользователей сайта в соответствующей миграции приводится на рисунке 8. В движке есть множество методов для манипуляций миграциями: откат отдельных, сброс всех и т.д.

```
1 Schema::create('users', function (Blueprint $table) {
2
3     $table->increments('id');
4
5     $table->string('name');
6
7     $table->string('email')->unique();
8
9     $table->string('password');
10
11    $table->rememberToken();
12
13    $table->timestamps();
14
15 });
```

Рисунок 8 – Создание таблицы пользователей сайта

Blade – шаблонизатор с набором своих директив. Принцип его работы похож на большинство своих собратьев – Smarty, Twig и т.д.

Twitter Bootstrap и jQuery из коробки. После установки Laravel фреймворка в распоряжении разработчика находятся файлы app.js и app.css, которые представляют собой скомпонованные и минимизированные jQuery и Bootstrap самых последних версий на момент выхода релиза Laravel.

Laravel Mix. Пакет представляет собой надстройку, позволяющую разделять css и js код на отдельные модули, конфигурировать их использование, настраивать минификацию и использование css-препроцессоров (sass, less, stylus и т.д.).

Регистрация и аутентификация. Из коробки Laravel предоставляет механизм регистрации и авторизации пользователей, что упрощает жизнь разработчикам, позволяя не изобретать очередные велосипеды.

Валидаторы. На PHP можно увидеть при проверке на существование в БД значения, вводимого с формы, путём ручного запроса в базу для избегания его дублирования. По правде говоря, поначалу, когда я только начинал работать с Laravel, я поступал точно так же, т.к. не знал о существовании и назначении валидаторов.

Механизм очередей. В Laravel из коробки доступны инструменты организации очередей процессов. Данная функция незаменима для HighLoad-проектов, так как позволяет разгрузить сервер от постоянной работы.

Интерфейс для Cron задач. Laravel предоставляет набор методов для создания и управления задачами, выполняемыми с помощью планировщика задач Cron. Пример задачи с выполнением каждый час в промежутке между 7 и 22 часами дня представлен на рисунке 9.

```
1 | $schedule->command('reminders:send')->hourly()->between('7:00', '22:00');
```

Рисунок 9 – Задача планировщика Cron

Удобный дебаггинг кода и тестирование реализуется наличием debug panel, специальной функции dd() для вывода данных на экран (аналог PHP-конструкции `echo '<pre>'; print_r($var); die();`) и подробным логом приложения. Кроме того, в процессе тестирования работоспособности ресурсов есть возможности эмуляции посещения страниц сайта и различных действий (нажатие на ссылки, кнопки, ввод текста и т.д.) благодаря использованию компонентов Symfony.

3.1.3 Расширение базового функционала веб-фреймворка – Laravel

Возможность расширения базового функционала. Разработчики Laravel ввели возможность расширения функционала базового приложения за счёт установки пакетов, которые являются аналогами модулей для CMS.

Кэширование из коробки. По умолчанию доступен драйвер кэширования file, благодаря которому закэшированная информация будет храниться в файловой системе.

Удобный механизм роутинга. Манипуляции с URL, доступными на сайте, в Laravel невероятно просты и удобны. Всё, что нужно сделать, для добавления Laravel 5.5 routes – это отредактировать файл routes/web.php. Простейшее добавление нового роута выглядит как показано на рисунке 10.

```
1 Route::get('/', function () {
2
3     return view('welcome');
4
5 });
```

Рисунок 10 – Добавление роута

Работа с сессиями различными способами. Как известно, объекты сессий хранятся на сервере, следовательно, ими можно легко и удобно манипулировать серверными языками программирования, а также задавать им различные хранилища.

Работа на страницах производится через панель управления системы Laravel. Пройдя идентификацию, становится доступной административная часть портала (рисунок 11).

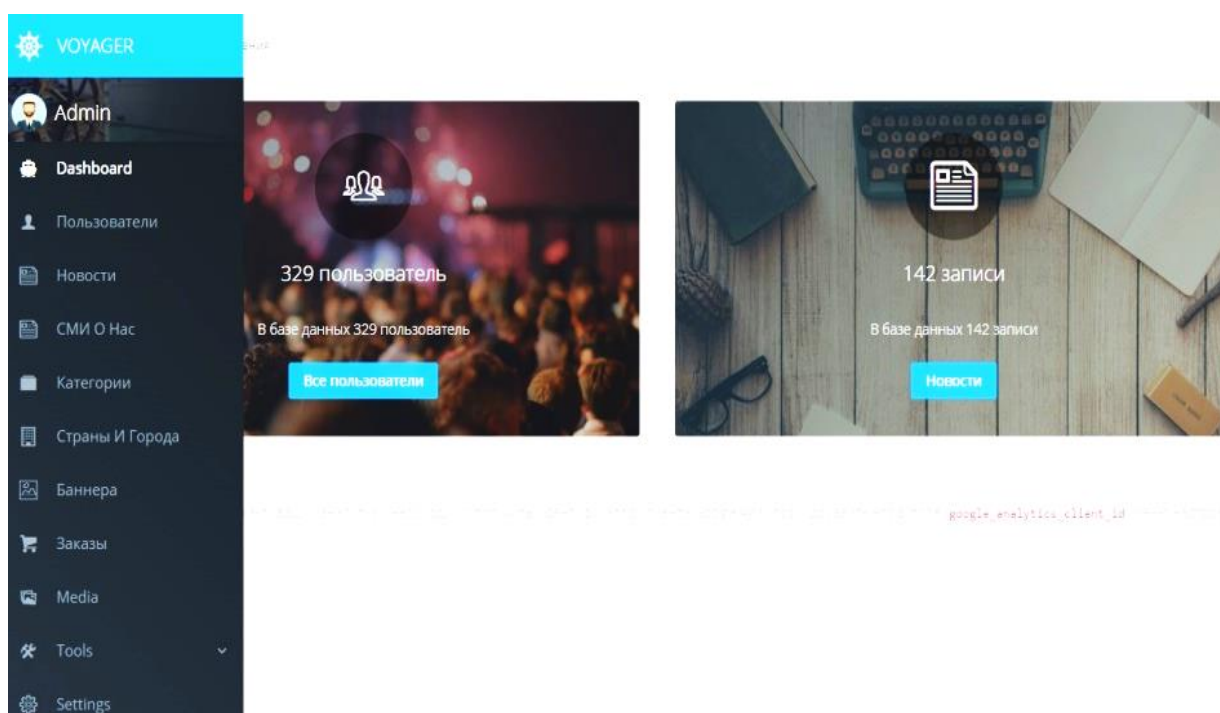


Рисунок 11 – Административная часть портала

Voyager устанавливается через Composer (установка и настройка Composer Laravel). Чтобы установить Voyager, в корневой папке текущего портала запускается команда: `composer require tcg/voyager`.

В панели управления Voyager видны разделы, в которых есть возможность просмотра необходимой информации, а также загрузка контента.

Существует таблица, в которой есть информация обо всех зарегистрированных пользователях (рисунок 11). Также присутствует функционал входа администратора и личные кабинеты. Это создано для того,

чтобы предотвратить действия пользователей, нарушающие действия, связанные с законодательством Республики Казахстан, а также внутренними правилами Technomarkets.

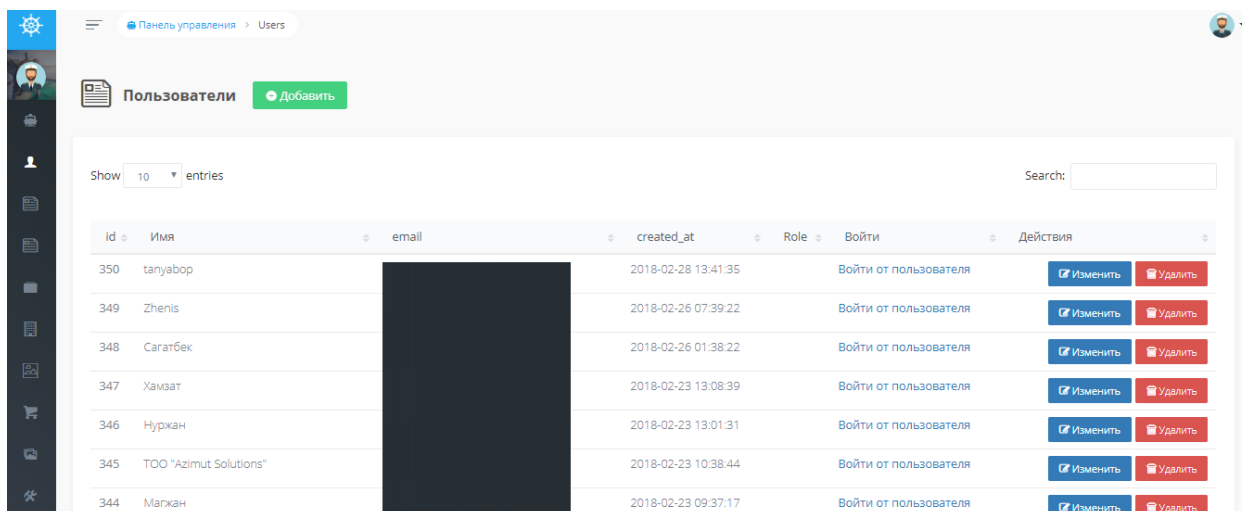


Рисунок 12 – Раздел «Пользователи»

Laravel предоставляет функции устойчивого шифрования по алгоритму AES с помощью расширения MyScip для PHP. Примеры использования методов шифрования/дешифрования строк показаны на рисунках 13, 14.

```
$encrypted = Crypt::encrypt('секрет');
```

Рисунок 13 – Использование шифрования при передаче параметров

```
$decrypted = Crypt::decrypt($encryptedValue);
```

Рисунок 14 – Использование дешифрования при передаче параметров

4 Безопасность жизнедеятельности

4.1 Анализ условий труда

В выполнении дипломной работы рассматривается проектирование и создание электронной торговой площадки с защитой пользовательских данных. Проект был разработан с целью создания на рынке услуг центра оборудования, техники и технологий для стран СНГ и Европы. Были рассмотрены различные способы реализации проекта. Изначальное позиционирование проекта направлено на создание площадки с информационной безопасностью. Факт того что была проведена полноценная аналитика и мониторинг, говорит о том что, подобранная тема была востребована в направлении информационной безопасности в электронно-торговой площадке. В сравнении с ранней модернизацией портала было выявлено ряд улучшенных мер в направлении информационной безопасности. Качество безопасности пользовательских данных в системах было развито. Пользователи различных организаций были проинформированы данным проектом, что послужило приятным бонусом в корпоративной культуре на рынке оказываемых услуг. Рассматривая проект, непосредственно был реализован портал с системой защиты данных, благодаря выбранному движку и программному обеспечению “Laravel”. Политики применимые в безопасности являются базовыми мерами защиты. Основные технические моменты параллельно были учтены во время создания.

Компания, в которой внедрялся проект, называется «TechnoMarkets». На рынке информационного сервиса, компания имеет значительное место. Род деятельности компании, это обслуживание различных организаций также продажи антивирусного программного обеспечения. Данная компания имеет офис в нежилом помещении. Подсчитывая количество сотрудников, можно сказать, что количество работающих в данной организации составляет 30 человек. Непосредственно в офисе имеются свои отделы, каждая их которых занимается своим делом. К примеру, отдел администрирования занимается управлением всего оборудования, в частности своих клиентов. В организации насчитывается 6 отделов. В процессе работы сотрудники организаций тесно взаимодействуют с другими отделами. Так как целью компании является клиентская удовлетворённость и продажи продуктов организаций. Также присутствует отдел программистов. В отделе программистов разрабатываются программные обеспечения, как на компьютер, так и на телефон.

Программное обеспечение применимое для защиты пользовательских данных требует минимальных затрат и усилий. Компания, в которой проектировалась защита пользовательских данных, дала возможность реализации запланированных действий. Технические средства организации были достаточными для улучшения качества безопасности пользовательских данных организации. Главные сотрудники отдела администрирования

предоставляют возможность выполнения проекта, предлагая все средства организации.

В процессе работы над проектом на позиции руководителя проекта, выявил вредный производственный недостаток. Недостатком было: плохая вентиляция производственного помещения. Вентиляция производственного помещения – это совокупность мероприятий и устройств, необходимых для обеспечения заданного качества воздушной среды в рабочем помещении. Вентиляция принадлежит главенствующую роль в нормализации воздушной среды на рабочих местах и в производственных помещениях. Естественная вентиляция помещений может быть неорганизованной и организованной. При неорганизованной вентиляции поступление и удаление воздуха происходит через окна, форточки, специальные проемы, а также через не плотности наружных ограждений. Организованная естественная вентиляция производственных помещений называется аэрацией. Она осуществляется с помощью специально создаваемых конструктивных элементов промышленных зданий - аэрационных фонарей или с помощью специальных каналов или шахт, функционирующих под действием теплового напора.

Исследуя организацию, а в частности местность нахождения здания, то офис находится в благоприятных условиях для сотрудников организаций. Если брать в счет род деятельности руководителя проекта, то это ежедневная работа на персональном компьютере, конструкция рабочего места является оптимально размещенным на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей (размер монитора, клавиатуры и других). Конструкция рабочей мебели (столы и кресла) обеспечивает возможность индивидуальной регулировки. Рабочие места, высотой 0,8 м, размещены боковой стороной к окну. При эксплуатации электрооборудования существует опасность поражения электрическим током. В связи с этим вилки и розетки имеют контакты зануления, а все кабели спрятаны в кабель-каналы. Оборудование является практически бесшумным.

В помещении, где располагаются сотрудники, применяется естественное освещение, осуществляемое через боковое окно, ориентированное на запад. Для защиты от избыточного света и ярких лучей используются регулируемые жалюзи с вертикальными ламелями.

Внутри помещения офиса у рабочего места руководителя проекта уровень опасных и вредных факторов не превышает установленных нормативов и каждое рабочее место сотрудников работающих в помещении, где работодатели выполняют свои должностные обязанности максимально приспособлено для характера выполняемых работ. Помещение является светлым, сухим и чистым, соответствующее санитарно-гигиеническим нормам.

4.2 Расчет тепловых нагрузок в помещении

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние)

НАРУЖНЫЕ ТЕПЛОВЫЕ НАГРУЗКИ

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными. Теплопоступления и тепло потери в результате разности температур определяются по формуле 0.1:

$$Q_{\text{огр}} = V_{\text{пом}} * X_o * (t_{\text{Нрасч}} - t_{\text{Врасч}}), \text{ Вт (0.1), где}$$

$V_{\text{пом}}$ – объем помещения, м^3 ;

$$V_{\text{пом}} = 3.5 * 3.5 * 3 = 36.75 \text{ м}^3;$$

X_o – удельная тепловая характеристика, $\text{Вт/м}^3 * ^\circ\text{C}$;

$$X_o = 0,42 \text{ Вт/м}^3 * ^\circ\text{C};$$

$t_{\text{Нрасч}}$ – наружная температура (параметр А). Для холодного периода – средняя температура самого холодного месяца в 14 часов, для теплого периода – средней температуре самого жаркого месяца в 14 часов.

$t_{\text{Врасч}}$ – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года:

$$t_{\text{Нрасч}} = 28,4 \text{ } ^\circ\text{C}$$

$$t_{\text{Врасч}} = 26 \text{ } ^\circ\text{C}$$

$$Q_{\text{огр}} = 36,75 * 0,42 * 2,4 = 37,04 \text{ Вт}$$

Для холодного времени года

$$t_{\text{Нрасч}} = -8 \text{ } ^\circ\text{C}$$

$$t_{\text{Врасч}} = 19 \text{ } ^\circ\text{C}$$

$$Q_{\text{огр}} = 36,75 * 0,42 * 27 = 416,74 \text{ Вт}$$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле 0.2:

$$Q_p = (q^I F_o^I + q^{II} F_o^{II}) * \beta_{\text{с.з}} \text{ (0.2), где}$$

q^I, q^{II} – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м²;

F_0^I, F_0^{II} – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м²;

$\beta_{с.з.}$ – коэффициент теплопропускания. Для штор-жалюзи с металлическими пластинами:

$$\beta_{с.з.} = 0,15$$

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F_0^I = F_0^{II} = F_0 = 0$:

$$Q_p = q^{II} F_0 * \beta_{с.з.} = q_{вр} * K_1^T * K_2 * \beta_{с.з.} * n * S_0 \quad (0.3), \text{ где}$$

$q_{вр}$; $q_{вр}$ – тепловые потоки от рассеянной радиации, Вт/м². Для широты в 44° СШ после полудня в 14-15 ч. при расположении ЮВ:

$$q_{вр} = 63 \text{ Вт/м}^2;$$

$F_0 = nS_0 = 2*2 = 4 \text{ м}^2$ – площадь светового проема (n – число окон; S_0 – площадь 1 окна);

K_1 – коэффициент затемнения остекления переплетами (K_1^T – для проемов в тени).

$$K_1^T = 1,28;$$

K_2 – коэффициент загрязнения остекления:

$$K_2 = 0,95.$$

Тогда:

$$Q_p = 63 * 1,28 * 0,95 * 0,15 * 4 = 45,96 \text{ Вт}$$

Для широты в 44° СШ после полудня в 14-15 ч. при расположении ЮЗ:

$$q_{вр} = 101 \text{ Вт/м}^2;$$

$$F_0 = nS_0 = 2*2 = 4 \text{ м}^2$$

Тогда:

$$Q_p = 101 * 1,28 * 0,95 * 0,15 * 4 = 73,69 \text{ Вт}$$

Тогда общее тепlopоступление солнечного излучения с обеих сторон равно:

$$Q_p = 45,96 + 73,69 = 119,65 \text{ Вт}$$

ВНУТРЕННИЕ ТЕПЛОВЫЕ НАГРУЗКИ

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами фотокопировальными машинами пр.;

Летом при 24 °С один мужчина выделяет явного тепла 68 Вт, а общего – 104 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$$Q_{л}^я = 68*2 + 68*1*0,85 = 193,8 \text{ Вт}$$

А выделение общего тепла:

$$Q_{л}^{\circ} = 104 \cdot 2 + 104 \cdot 1 \cdot 0,85 = 296,4 \text{ Вт}$$

Зимой при 18 °С один мужчина выделяет явного тепла 89 Вт, а общего – 104 Вт. Тогда выделение явного тепла в помещении составит:

$$Q_{з}^{\text{я}} = 89 \cdot 2 + 89 \cdot 1 \cdot 0,85 = 253,65 \text{ Вт.}$$

А выделение общего тепла:

$$Q_{з}^{\circ} = 104 \cdot 2 + 104 \cdot 1 \cdot 0,85 = 296,4 \text{ Вт}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле:

$$Q_{осв} = \eta \cdot N_{осв} \cdot F_{пол}, \text{ Вт} \quad (0.4)$$

где η – коэффициент перехода электрической энергии в тепловую (для лампы накаливания $\eta=0,92-0,97$);

$N_{осв}$ – установленная мощность ламп ($N=60 \text{ Вт/м}^2$);

$F_{пол}$ – площадь пола:

$$F_{пол} = 3,5 \cdot 3,5 = 12,25 \text{ м}^2$$

Тогда:

$$Q_{осв} = 0,92 \cdot 60 \cdot 12,25 = 676,2$$

Тепло, выделяемое производственным оборудованием, определяется по формуле:

$$Q_{об} = N_{уст} \cdot K \quad (0.5)$$

$$Q_{об} = 0,3 \cdot 3 \cdot 0,75 \cdot 10^3 = 675 \text{ кВт.}$$

Теплопритоки, возникающие за счёт находящейся оргтехники – это 30% мощности оборудования:

$$Q_{орг} = 3 \cdot 0,3 \cdot 0,3 \cdot 10^3 = 270 \text{ кВт}$$

4.3 РАСЧЕТ ТЕПЛОВОГО БАЛАНСА ПОМЕЩЕНИЯ

На основании выполненных расчетов составим баланс теплопоступлений в помещении:

$$Q_{изб} = Q_p + Q^{\text{я}} + Q_{осв} + Q_{об} + Q_{орг} + Q_{огр}$$

$$\text{Лето: } Q_{изб}^{\text{л}} = 119,65 + 193,8 + 676,2 + 675 + 270 + 37,04 = 1,97 \text{ кВт}$$

$$\text{Зима: } Q_{изб}^{\text{з}} = 119,65 + 253,65 + 676,2 + 675 + 270 + 416,74 = 2,41 \text{ кВт}$$

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле:

$$Q_{н} = \frac{Q_{изб.лето} \cdot 860}{V_{пом}}$$

$$Q_{н} = \frac{2,41 \cdot 860}{60,75} = 34,11 \text{ ккал/м}^3$$

При $Q_{н} > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^{\circ}\text{C}$,

при $Q_{н} < 20 \text{ ккал/м}^3$, $\Delta t = 6 \text{ }^{\circ}\text{C}$.

Определение количества воздуха, необходимое для поступления в помещение:

$$L = \frac{Q_{\text{изб}} \times 860}{C \times \Delta t \times \gamma}$$

$$L = \frac{2,41 \cdot 860}{0,24 \cdot 8 \cdot 1,206} = 895,01 \text{ м}^3/\text{час}$$

где $C=0,24$ ккал/(кг · °С) – теплоемкость воздуха,

$\gamma=1,206$ кг/м³ – удельная масса приточного воздуха.

ВЫБОР КОНДИЦИОНЕРА И МЕСТО РАСПОЛОЖЕНИЯ

Исходя из полученных результатов, для удаления лишнего тепла и очистки воздуха нужно использовать вентиляционную систему, которая способна обеспечить требуемую подачу воздуха $L=895,01$ (м³/ч). В данном случае подойдет Кондиционер MIDEA MDSA-09HRFN1 INVERTER . Данный кондиционер способен обеспечить подачу воздуха до 1200 м³ /ч.

Технические характеристики:

- Мощность (охлаждение): 2.93 кВт;
- Мощность (обогрев): 2.93 кВт;
- Потребляемая мощность при охлаждении: 2200 Вт;
- Потребляемая мощность при обогреве: 2240 Вт;
- Подача воздуха более: 895.01 м³/ч;
- Обслуживаемая площадь: 28 м²;
- Уровень шума внутреннего блока: 37-41 дБ;
- Уровень шума внешнего блока: 48 дБ;
- Цвет: серый.

Характеристики подключения:

- Вентиляция: 1200 м³/час;
- Класс энергоэффективности при охлаждение/обогреве: A++/A+ ;
- Электропитание, В/Гц/Ф:220 Вт;
- Энергопотребление в режиме ожидания не более 1 Вт;

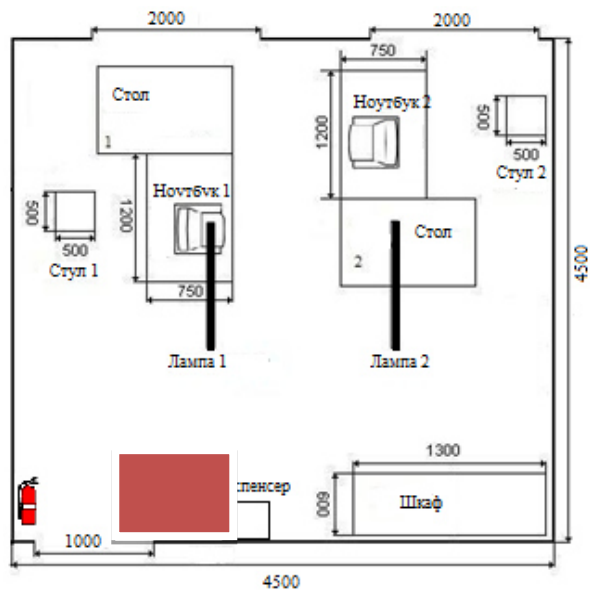


Рисунок 4.1 – Расположение кондиционера в помещении



Рисунок 4.2 – Выбранный кондиционер



Рисунок 4.3 – Внутренность кондиционера

Таблица 4.1 - Нормы микроклимата в помещении

Период года	Категории работ по уровню энергозатрат, Вт	Температура воздуха, град. С	Температура поверхности, град. С	Относительная влажность воздуха, %	Скорость движения воздуха
Холодный	Ia (до 139)	22-24	21-25	60-40	0,1
	Iб (140-174)	21-23	20-24	60-40	0,1
	IIa (175-232)	19-21	18-22	60-40	0,2
	IIб (233-290)	17-19	16-20	60-40	0,2
	III (более 290)	16-18	15-19	60-40	0,3
Теплый	Ia (до 139)	23-25	22-26	60-40	0,1
	Iб (140-174)	22-24	21-25	60-40	0,1
	IIa (175-232)	20-22	19-23	60-40	0,2
	IIб (233-290)	19-21	18-22	60-40	0,2
	III (более 290)	18-20	17-21	60-40	0,3

4.4 ВЫВОД ПО ЧАСТИ БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

Здание – это совокупность помещений, представляющих собой ограниченный объем, в пределах которого протекает жизнедеятельность человека. Процесс жизнедеятельности сопровождается взаимодействием человека с окружающей его средой помещения. Правильная организация

помещений и здания в целом открывает возможность обеспечения в них безопасных и эффективных условий пребывания человека. Внутренняя среда помещения, проявляющаяся в большом числе факторов воздействия на человека, называется микроклиматом помещения. Среди факторов внутренней среды выделим комплекс микроклиматических условий, оказывающих наиболее осязаемое физиологическое воздействие на человека. К ним относят тепловые условия в помещении и состав внутреннего воздуха. Человек познает мир частично через ощущения, частично сознанием. При этом непосредственно поступающая информация об окружающей среде соотносится в мозгу с информацией, накопленной в памяти на базе предыдущего опыта. Это обстоятельство свидетельствует об индивидуальности восприятия человеком внутреннего микроклимата помещения. Окружающая среда, которая не содержит раздражающих и возбуждающих факторов, препятствующих физической и умственной работе, а также отдыху, называется комфортной. Приведенное определение распространяется также на тепловые условия и состав воздуха помещения. Тепловые условия в настоящее время принято оценивать температурой воздуха, радиационной температурой помещения, относительной влажностью и подвижностью воздуха.

Состав воздуха характеризуется концентрацией углекислоты, концентрацией вредных газов, паров, пыли. Восприятие воздуха характеризуется также озono-ионным составом и запахами. Перечисленные параметры являются исходными при проектировании зданий и систем обеспечения микроклимата и нормируются. При этом определение нормативных параметров исходит из стремления к достижению оптимальных значений, т.е. таких, при которых как можно меньшее число людей было бы им и недоволено. Использование оптимальных параметров микроклимата не во всех зданиях бывает целесообразным и экономически оправданным. Поэтому в отечественных нормах широко используется понятие допустимых параметров, представляющих собой разумные граничные значения, при которых не наблюдается отрицательного воздействия на организм человека.

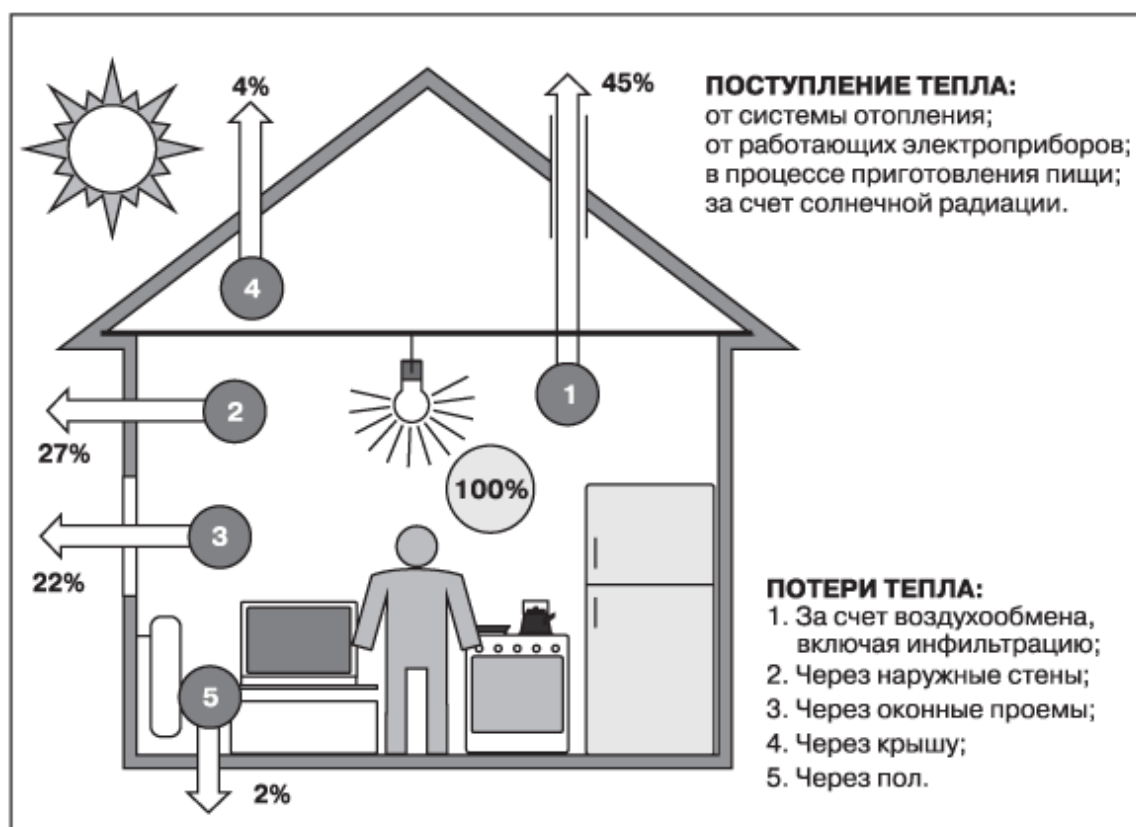


Рисунок 4.2 – Поступление и потери тепла в помещении

5 Технико-экономическое обоснование

Цель данного дипломного проекта заключается в разработке электронно-торговой площадки с защитой пользовательских данных, которая позволит понять работу портала изнутри.

В разработке ЭТП будет участвовать группа специалистов, которая включает в себя: технический руководитель, программист-разработчик. В обязанности технического руководителя входит соблюдение и разработка рабочих графиков, их контроль и оптимизация. В обязанности программиста-разработчика входит разработка технического обоснования, разработка программного обеспечения, его тестирование и сопровождение. Следовательно, основная работа ложится на плечи программиста-разработчика, в то время как технический руководитель занимается организационными вопросами. Технико-экономическое обоснование содержит следующие пункты:

- определение трудоемкости разработки;

- расчет затрат на разработку;
- определение ценности готового продукта;
- оценка результатов работы ЭТП.

5.1 Определение трудоемкости разработки

Для того, чтобы точно определить сложность разработки электронного торгового портала, необходимо произвести деление всей задачи на более простые этапы. Это позволит эффективно следить за прогрессом разработки, за счет деления сложной задачи на более легкие подзадачи. Такой подход, с моей точки зрения, считается более эффективным и позволяет результативно и быстро обрабатывать подзадачи. Модель распределения сложности разработки и стадии разработки представлены в таблице 4.1.

Таблица 4.1 – Этапы разработки ПО

Этапы разработки	Вид работы	Трудоемкость, чел. час.
Этап 1	Постановка задач	20
Этап 2	Разработка технического задания	40
Этап 3	Анализ созданных ЭТП	20
Этап 4	Подготовка ПО	15
Этап 5	Изучение рынка оказываемых услуг	20
Этап 6	Реализация проекта	45
Этап 7	Модернизация ЭТП	35
Этап 8	Создание защиты пользовательских данных	30
Итого:		225

Продолжительность рабочего дня равна 8 часам. В результате для реализации ЭТП необходимо $225:8=28$ рабочих дней.

5.2 Расчет затрат на разработку

Определение затрат необходимых для разработки ЭТП производится на основе имеющейся сметы, которая включает следующие элементы:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

Материальные затраты делятся на основные и вспомогательные затраты на материалы, энергию и другие затраты необходимые для разработки. Расчет материальных затрат происходит по форме, предоставленной в таблице 4.2.

Таблица 5.2 – Затраты на материальные ресурсы

Наименован	Марка	Ед.	Количес	Цена за	Сумма
------------	-------	-----	---------	---------	-------

ие материала		измерения	тво	ед. в тенге	в тенге
Бумага для офиса	International Paper	Упаковка	1	1 000	1 000,00
Тетрадь (96 листов)	Маяк Канц	Штук	2	200	400,00
Блокнот	КТС-ПРО	Штук	2	400	800,00
Ручки	Parker Jotter	Штук	2	90	180,00
Компьютерная мышь	TECH	Штук	1	3 000	3 000,00
Итого:					5 380,00

Для разработки программного обеспечения будет использоваться ноутбук HP ProBook G430-g1, мощности ноутбука достаточно для выполнению поставленных задач. В связи с тем, что ноутбук имеет уже установленную операционную систему и необходимое программное обеспечение для разработки, нет необходимости приобретать дополнительные ПО.

Общую сумму, необходимую на материальные средства (Z_m) можно рассчитать по следующей формуле:

$$Z_m = \sum P_i * C_i, \quad (4.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;
 C_i - цена за единицу i -го вида материального ресурса, тг;
 i - вид материального ресурса;
 n - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 4.3.

Таблица 5.3 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	Lenovo Ideapad 330 81FK00CXR К	Штук	1	390 000	390 000,00
Принтер	Samsung SCX-3400	Штук	1	52 874	52 874,00
Хостинг	PS.kz	Штук	2	2 350	4 700,00
Модем	Ericsson T073G	Штук	1	14 000	14 000,00

ОС	Windows 10	Штук	1	-	-
Домен	PS.kz	Штук	1	3 338	3 338,00
Итого:					464 912,00

$$З_m = 5\,380 + 464\,912 = 470\,292,00 \text{ тг.}$$

Для реализации программного обеспечения необходимы материалы на сумму 470 292,00 тенге.

5.3 Расчет затрат на электроэнергию

Так как при разработке программного обеспечения не обойтись без потребления электроэнергии, имеет смысл произвести расчет затрат на электроэнергию.

Согласно таблице 4.1 для разработке программного обеспечения необходимо порядка 225 часов, теперь необходимо рассчитать стоимость электроэнергии, которая будет потрачена в течении 225 часов. Для принтера расчет будет проводиться для периода в 24 часа, так как нет необходимости постоянно использовать принтер.

$$Э = З_{\text{эл.эн.обор.}} + З_{\text{доп.нужды.}} \quad (4.2)$$

где $З_{\text{эл.эн.обор.}}$ – затраты на электроэнергию оборудования;

$З_{\text{доп.нужды.}}$ – затраты электроэнергии на дополнительные нужды.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$З_{\text{эл.эн.обор.}} = \sum W * K_{\text{исц}} * S * T, \quad (4.3)$$

где W – потребляемая мощность, Вт;

$K_{\text{исц}}$ – коэффициент использования ($K_{\text{исц}} = 0,7..0,9$);

T – время работы;

S – тариф (1кВт/ч = 23,85 тг).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 4.4.

Таблица 4.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/кВтч	Сумма, тг.
Ноутбук	0,6	0,7	225	23,85	2 254,90
Модем	0,08	0,9	225	23,85	386,40

Принтер	0,5	0,9	24	23,85	257,60
Кондиционер	0,8	0,9	180	23,85	3 091,00
Освещение	0,3	0,7	225	23,85	1 127,00
Итого:					7 117,00

$$Z_{\text{эл.эн.обор.}} = 7\,117,00 \text{ (тенге)}$$

На дополнительные потребности расходы подсчитываются на основе повышенного показателя в объеме 5% от расходов на электроэнергию:

$$Z_{\text{доп.нужды}} = 5\% * Z_{\text{эл.эн.обор.}} \quad (4.4)$$

Определим затраты на дополнительные потребности согласно формуле (4.4):

$$Z_{\text{доп.нужды}} = 0.05 * 7\,117 = 355,85 \text{ (тенге)}$$

Исходя из всех расчетов, полные расходы на электроэнергию составляют:

$$Z = 355,85 + 7\,117,00 = 7\,472,85 \text{ (тенге)}$$

5.4 Расчет затрат на оплату труда

Для разработки электронного торгового портала, как указывалось ранее, необходимо два работника:

- руководитель проекта – управление рабочим временем, корректировка рабочих процессов, координация, изучение предметной области;
- разработчик – разработка ЭТП, тестирование и сопровождение.

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$Z_{\text{тр}} = \sum ЧС_i * T_i \quad (4.5)$$

где $ЧС_i$ - часовая ставка i -го работника, тг;

T_i - трудоемкость разработки модели, чел.×ч; i - категория работника;

n - количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (4.6)$$

где $ЗП_i$ - месячная заработная плата i -го работника, тг;
 $ФРВ_i$ - месячный фонд рабочего времени i -го работника, час.

Месячная заработная плата руководителя равняется 200 000 тенге и месячная заработная плата разработчика равняется 150 000 тенге. Рассчитаем часовую ставку каждого работника согласно формуле (4.6):

$$\begin{aligned} ЧС_{\text{руководитель}} &= \frac{275\,000}{22 * 8} = 1\,562,50 \text{ тг/ч} \\ ЧС_{\text{разработчик}} &= \frac{210\,500}{22 * 8} = 1\,196,02 \text{ тг/ч} \end{aligned}$$

Часовая ставка руководителя составляет 1 136,36 (тг/ч), трудоемкость разработки равняется 100 часам. Часовая ставка разработчика составляет 852,3 (тг/ч), трудоемкость разработки равняется 225 часам. Согласно формуле (4.5) можно рассчитать сумму расходов на заработную плату работников:

$$З_{\text{тр}} = 1\,562,50 * 100 + 1\,196,02 * 225 = 156\,250 + 269\,104,5 = 425\,354,5$$

Расчеты затрат по оплате труда показаны в таблице (4.5).

Таблица 4.5. – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	Инженер-проектировщик	100	1136,36	156 250
Разработчик	Инженер	225	852,3	269 104,5
Итого:				425 354,5

5.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонда оплаты труда. Социальный налог можно рассчитать по следующей формуле:

$$С_{\text{н}} = (ФОТ - ПО) * 0,095 \quad (4.7)$$

где ПО - отчисления в пенсионный фонд, они составляют 10% от ФОТ.

$$\begin{aligned} ПО &= 293\,767,8 * 0,1 = 29\,376,8 \text{ тенге} \\ С_{\text{н}} &= (293\,767,8 - 29\,376,8) * 0,095 = 25\,117,2 \text{ тенге} \end{aligned}$$

Результаты расчетов представлены в таблице (4.6):

Таблица 4.6 – Начисление социального налога

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления, тг	Социальный налог, тг
Руководитель	1	113 636	11 364	9715
Разработчик	1	191 767,5	19 177	16 396,1
Итого:				25 117,2

5.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_r = \frac{C_{об} * H_a}{100} \quad (4.8)$$

где, $C_{об}$ – стоимость оборудования;

H_a – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_r = \frac{390\,000 * 28}{100} = 109\,200 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_r = \frac{97\,500 * 28}{365} = 7479,45 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (4.7).

Таблица 4.7 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	390 000	25	97 500	9 082,2
Принтер	52 874	25	13 218	108,64
Модем	14 000	20	2 800	172,6

Хостинг	4 700	20	940	105,98
Домен	3 338	15	500,7	52,28
Итого:			114 958,7	9 521,67

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице (4.8). Продемонстрирована диаграмма рабочих расходов.

Таблица 4.8 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг	Проценты
Затраты на оборудование	470 292,00	54 %
Затраты на оплату труда	293 767,8	37 %
Социальные налоги	25 117,2	4 %
Затраты на электроэнергию	7 472,85	4 %
Амортизация основных фондов	9 521,67	1 %
Итого по смете:	800 435,67	100 %

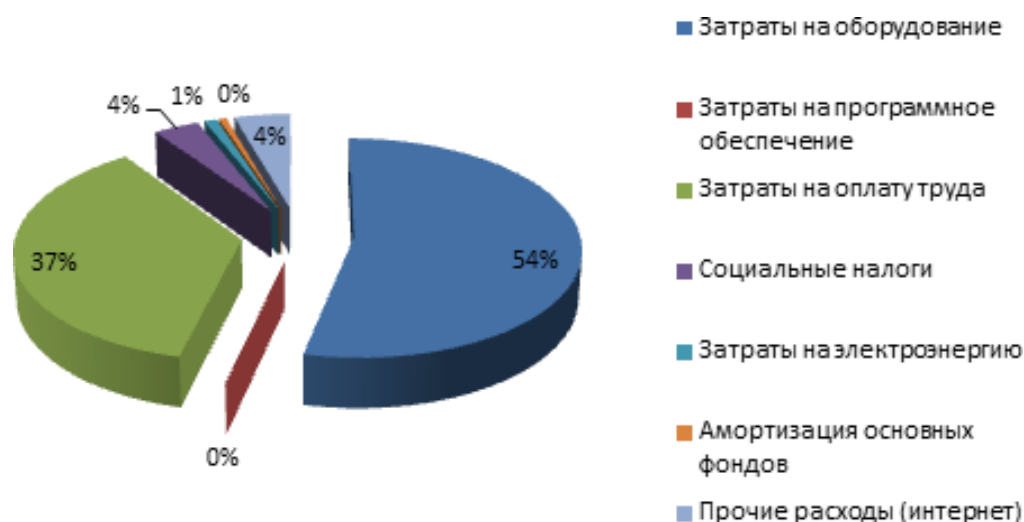


Рисунок – Диаграмма затрат

5.7 Определение возможной (договорной) цены ПО

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость $Ц_d$ для программного обеспечения можно рассчитать по следующей формуле:

$$Ц_d = Z_{\text{нир}} \left(1 + \frac{P}{100} \right), \quad (4.9)$$

где $Z_{\text{нир}}$ – затраты на разработку программного обеспечения, тг;

P – средний уровень рентабельности ЭТП, (%). Данный параметр принят равным 25%.

$$C_d = 800\,435,67 \left(1 + \frac{25}{100}\right) = 800\,435,67 + 207\,008,94 = 1\,007\,444,7 \text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле:

$$C_p = C_d + C_d * \text{НДС}, \quad (4.10)$$

$$C_p = 1\,007\,444,7 + 1\,007\,444,7 * 0,12 = 1\,128\,338,06 \text{ тенге}$$

Возможная договорная цена с учетом НДС - 1 128 338,06 тенге

Затраты на разработку (себестоимость) - 800 435,67 тенге

Прибыль 207 008,91 тенге

ЗАКЛЮЧЕНИЕ

Изучение Laravel фреймворка - перспективное занятие со всех точек зрения: как с позиций профессионального роста, так и карьерного. Про себя скажу, что я стал замотивирован с первых дней работы с ним – настолько мне данный фреймворк пришёлся по душе. Наверное, потому, что «его логика совпадает с моей собственной», как сказал один из программистов (при личной встрече с ним) на конференции “KazHackStan”, которую я посетил совместно с обучающимися, преподавателями и экспертами «Алматинского университета энергетики и связи». Таким образом, в данной работе я рассмотрел наиболее интересные и значимые плюсы Laravel 5.5.

В дипломной работе описан процесс создания портала Technomarkets и меры организации его защиты при помощи разных инструментов. Эти ограничения в администрировании существенно могут повлиять на предотвращение загрузки пользователями запрещённых материалов, непредусмотренных законодательством Республики Казахстан и позволит пресекать попытки экстремистских лидеров вербовать пользователей, ограничить доступ к потенциально вредоносной информации, позволит выявлять экстремистские организации на стадии формирования, что является угрозой для рынка услуг информационной безопасности РК. Это важно, так как площадка задумана для работы с пользователями стран Европы и Азии.

Список литературы

1. Laravel по-русски URL: <https://laravel.ru/> (дата обращения: 28.02.2018)
2. Электронная торговая площадка URL: <http://technomarkets.kz/> (дата обращения: 28.02.2018)
3. Электронная торговая площадка URL: <http://technomarkets.kz/admin> (дата обращения: 29.02.2018)
4. Jason Todd. Deploying a Vyatta Core Firewall – USA: Sans Institute, 2010.
5. Tim Boyles. CCNA Security Study Guide – Indianapolis USA: SyBEX, 2012.
6. Белов С.В. Безопасность жизнедеятельности. М: Высшая школа 1999.
7. Баклашов Н.И., Китаева Н.Ж., Терехов Б.Д. Охрана труда на предприятиях связи и охрана окружающей среды. – М.: Радио и связь, 1989.
8. Кошулько Л.П., Суляева Н.Г., Генбач А.А. Производственное освещение. Методические указания. – Алматы:АИЭС, 1989.
9. Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. – Алматы, 2009.
10. Корольченко А.В. Естественное и искусственное освещение. - М.: Издательство Москва, 2004.
11. Дюсебаев М.К. Безопасность жизнедеятельности: методические указания к выполнению раздела дипломных проектов. – Алматы.: АИЭС, 2003.
12. Голубицкая Е. А., Жигульская Г. М. Экономика связи. – М.: Радио и связь, 2000.

13. Аманжолова К. Б., Алибаева С. А. Экономика предприятия телекоммуникации: Учебное пособие. - Алматы: АИЭС, 2003.
14. Облачные технологии для бизнеса. URL: <http://profitday.kz/cloud> (дата обращения: 17.03.2019)
15. Cbtnuggets: GNS3 1.x Fundamentals. URL: www.cbtnuggets.com/it-training/gns3-1-x-fundamentals (дата обращения: 26.02.2019)

Список литературы

16. Справочник системного администратора, И.В.Коробко. Операционные системы и сети, 2009.
17. Microsoft Windows 7. Руководство администратора, Алексей Чермарев. Операционные системы и сети, 2010.
18. Windows IT Pro/RE, Открытые системы. Операционные системы и сети, 2013.
19. Jason Todd. Deploying a Vyatta Core Firewall – USA: Sans Institute, 2010.
20. Tim Boyles. CCNA Security Study Guide – Indianapolis USA: SyBEX, 2012.
21. Белов С.В. Безопасность жизнедеятельности. М: Высшая школа 1999.
22. Баклашов Н.И., Китаева Н.Ж., Терехов Б.Д. Охрана труда на предприятиях связи и охрана окружающей среды. – М.: Радио и связь, 1989.
23. Кошулько Л.П., Суляева Н.Г., Генбач А.А. Производственное освещение. Методические указания. – Алматы:АИЭС, 1989.
24. Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. – Алматы, 2009.
25. Корольченко А.В. Естественное и искусственное освещение. - М.: Издательство Москва, 2004.
26. Дюсебаев М.К. Безопасность жизнедеятельности: методические указания к выполнению раздела дипломных проектов. – Алматы.: АИЭС, 2003.
27. Голубицкая Е. А., Жигульская Г. М. Экономика связи. – М.: Радио и связь, 2000.
28. Аманжолова К. Б., Алибаева С. А. Экономика предприятия телекоммуникации: Учебное пособие. - Алматы: АИЭС, 2003.
29. Облачные технологии для бизнеса. URL: <http://profitday.kz/cloud> (дата обращения: 17.03.2016)
30. Cbtnuggets: GNS3 1.x Fundamentals. URL: www.cbtnuggets.com/it-training/gns3-1-x-fundamentals (дата обращения: 26.02.2016)