

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

Толыбекова А.М.

Платформа маркетплейса для образовательной сферы

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное
обеспечение

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой ПИ
Доктор ФНЦ,
 М. Турдалиулы
"6.10.2021" г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

На тему: " Платформа маркетплейса для образовательной сферы"
по специальности 5В070400 – Вычислительная техника и программное
обеспечение

Выполнила

Толыбекова А.М.

Научный руководитель



А. Жумабаев

Институт кибернетики и информационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

УТВЕРЖДАЮ

Доктор Р.И.Д.



М. Турдалулы

июня 2021 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Толыбековой Арайлы Маликовне

Тема: "Платформа маркетплейса для образовательной сферы"

Утверждена приказом проректора по академической работе № 1162 - 6

Исходные данные к дипломному проекту: Паспорт проекта

Перечень подлежащих разработке в дипломном проекте вопросов:

- а) сделать анализ актуальных систем создания маркетплейсов;
- б) понять принцип и значение маркетплейсов в современных рыночных условиях, рассмотреть разновидности электронной услуги;
- в) изучить основные понятия, которые используются при проектировании информационных систем типа «интернет-магазин»;
- г) определить понятие интернет-услуги, сущность, функции, разновидности;
- д) разобраться с моделями организации интернет-бизнеса в Казахстане экономическую специфику задач, которые они помогают решить
- е) проанализировать электронные продажи и их системы разработки;
- ж) выполнить проектирование платформы маркетплейса;
- з) разработать платформу маркетплейса для образовательной сферы;
- и) оценить, насколько эффективен разработанный программный продукт.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 31 рисунок.

Рекомендуемая основная литература: из 34 наименований.

ГРАФИК
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Обзор современных технологий и программных средств для разработки маркетплейса	31 января	
2. Понятие маркетплейса и современные технологии разработки	11 февраля	
3. Программные средства для разработки маркетплейса	11 февраля	
4. Серверные технологии для разработки маркетплейса	20 февраля	
5. Платформа маркетплейса для образовательной сферы	20 марта	
6. Структура и элементы дизайна маркетплейса	7 апреля	
7. Разработка базы данных для маркетплейса	9 апреля	
8. Применение CMS системы в разработке маркетплейса для образовательной сферы	9 апреля	
9. Написание пояснительной записки к дипломному проекту	15 апреля	

Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	Рамазан А. Б.- ассистент	31.05.2021	
Нормоконтролер	Маргулан К.- лектор	6.06.2021	

Научный руководитель  А.А Жумабаев

Задание принял к исполнению обучающийся  А.М. Толыбекова
Дата "5" июня 2021 г.

Аңдатпа

Соңғы біліктілік жұмысының мақсаты – веб-сайт түрінде нарық платформасын құру және құру.

Дипломдық жұмыстың жаңалығы - білім беру саласы үшін нарық алаңын құру. Нарықта әртүрлі қызмет көрсетушілер, яғни білім беру қызметтерін ұсынатын әр түрлі компаниялар орналасады. Клиент бағасы мен орналасқан жері бойынша өзіне сәйкес компанияны тандайды. Біздің елімізде тауарлар мен қызметтерді сататын әртүрлі интернет-платформалар бар. Бірақ бүгінде білім беру нарығында платформа жоқ. Бұл ерекшелік дипломдық жұмыстың басты жаңалығы болып табылады.

Бұл жұмыста нарық алаңын дамытудың теориялық мәселелері, оның ішінде білім беру саласы үшін нарықтың ерекшеліктері зерттелген және алынған материал негізінде заманауи даму технологиялары қолданылатын электронды сайт жасалды.

АННОТАЦИЯ

Целью выпускной квалификационной работы является разработка и создание платформы маркетплейса в виде сайта.

Новизна дипломной работы – создание платформы маркетплейса для образовательной сферы. В маркетплейсе будут располагаться разные поставщики услуг, то есть разные компании, которые представляют образовательные услуги. Клиент будет выбирать подходящую для себя компанию по цене и локации. В нашей стране существуют разные интернет платформы, которые продают товары и услуги. Но на сегодняшний день не существует платформы маркетплейса для образовательной сферы. И эта особенность является главной новизной дипломной работы.

В данной работе изучены теоретические вопросы разработки платформы маркетплейса, в том числе, рассмотрены особенности маркетплейса для образовательной сферы и на основании усвоенного материала разработан электронный сайт, где применены современные технологии разработки.

ANNOTATION

The purpose of the final qualifying work is the development and creation of a marketplace platform in the form of a website.

The novelty of the thesis is the creation of a marketplace platform for the educational sphere. The marketplace will house different service providers, that is, different companies that provide educational services. The client will choose a suitable company for himself by price and location. In our country, there are various Internet platforms that sell goods and services. But today there is no educational marketplace platform. And this feature is the main novelty of the thesis.

In this paper, the theoretical issues of developing the marketplace platform are studied, including the features of the marketplace for the educational sphere, and on the basis of the acquired material, an electronic site has been developed, where modern development technologies are applied.

СОДЕРЖАНИЕ

Введение	9
1 Обзор современных технологий и программных средств для разработки маркетплейса	12
1.1 Понятие маркетплейса и современные технологии разработки	12
1.2 Программные средства для разработки маркетплейса	16
1.3 Серверные технологии для разработки маркетплейса	23
2 Платформа маркетплейса для образовательной сферы	27
2.1 Структура и элементы дизайна маркетплейса	27
2.2 Разработка базы данных для маркетплейса	38
2.3 Применение CMS системы в разработке маркетплейса для образовательной сферы	42
Заключение	52
Список использованной литературы	54

ВВЕДЕНИЕ

Интернет, как всемирное средство взаимодействий, где отсутствуют территориальные и национальные границы, позволяет обмениваться различной информацией. С каждым годом интернет все сильнее проникает в работу Казахстанских компаний, внося существенные перемены в формат ведения бизнеса, а также, изменяя его вид, знакомя с новейшими возможностями и предоставляя не используемые прежде технологии для продвижения. Интернет в бизнесе играет стратегически важную роль не только для иностранных предприятий, но и для Казахстанских. Причиной является непрерывный рост числа пользователей сети, делая его главным источником привлечения потенциальных партнеров и каналом продаж во многих отраслях. Все более очевидна роль привлечения интернет-технологий в продвижение не только масштабных компаний, но и небольших предприятий.

В современном мире, интернет может использоваться не только как дополнительный канал продажи продукции, но и как инструмент, с помощью которого можно решить отдельные маркетинговые задачи. С увеличением числа пользователей, интернет становится главным маркетинговым каналом. Благодаря ему у предприятий появляется возможность выхода на мировой рынок, существенно уменьшать затраты в цепочке спроса и предложения, выводить обслуживание заказчиков на новый уровень, оказаться на ранее недоступных рынках, наладить новые потоки дохода, и, наконец, пересмотреть свою деятельность. Компания должна поделиться информацией о себе: деятельность, вид услуги, расценки, и т.д. Это можно сделать разными путями. Наиболее эффективный способ – зарегистрировать предприятие на информационной платформе маркетплейса.

Интернет-услуги – процесс по осуществлению физических либо нефизических услуг на специально для этого предназначенных электронных площадках, позволяющих дистанционно оформлять заказы. Интернет-услуги в Казахстане развиваются весьма удачно. Согласно анализу рынка интернет-услуги в Казахстане объем реализуемых товаров постоянно растет, как и прибавляется число предприятий, приходящих в онлайн-торговлю.

В 2000 году уровень проникновения интернета в Казахстане находился на отметке 0,5%, в 2006-м повысился до 3%, в 2008-м, во время кризиса – 15%, а в 2011-м вырос до 41%. В 2019 году 82,5% жителей регулярно пользовались интернетом. Рынок e-commerce увеличивается также, как и рост проникновения интернета. Например, за 2019 год число онлайн-платежей выросло в 2,3 раза, и составило около 14,4 трлн тенге (что занимает 3,7% всей розничной торговли). Прирост активных потребителей составил 72% их стало 3,2 млн человек.

В 2018 году 11,9% всех торговых сделок по всему миру были сделаны в онлайн-режиме, а к 2021 году эта цифра выросла до 17,5%. Эксперты отмечают, что развитие и мультипликативное влияние электронной торговли на

экономические показатели страны позволяет ожидать ежегодный вклад в ВВП Казахстана на уровне 1,1%.

Согласно статистическим данным World Bank, Ovum (World Cellular Information Service), 76,4% жителей Казахстана пользовались интернетом в 2017 году. В данном рейтинге Казахстан занял второе место, уступив Великобритании (94,8% пользователей), он оставил за собой США (76,2%), Польшу (76%) и Россию (76%). Позитивным моментом стало и увеличение числа соединений с помощью мобильных устройств в республике: за 2018 год их было 18,2 млн, а к 2022 году ожидается 25,6 млн.

Несмотря на то, что в области интернет торговли можно наблюдать кризис, не сдающий обороты наравне с этим можно отследить медленное, но стабильное развитие. Кризис в экономике стимулирует продавцов и поставщиков услуг к онлайн-развитию. На это оказывают влияние большие затраты на привычные продажи с прилавка из-за повышенных ставок на аренду, трат на работников и т.д., а также изменяющееся потребительское поведение: все больше казахстанцев перед тем, как приобрести товар или услугу пытаются найти в интернете более выгодное предложение. Не зря именно кризис 2014–2016 годов ускорил развитие торговли в онлайн-режиме в Казахстане: для сокращения издержек торговля направились в интернет. Свой вклад в это внес рост развития интернета в Казахстане – в частности мобильного и, в частности, в регионах.

Онлайн-магазины не заменяют привычные, таким образом, увеличивается рынок сбыта. В некоторых отраслях отсутствующий онлайн-магазин является стратегическим промахом.

В дипломной работе не будут затрагиваться достоинства и минусы платформы маркетплейса по сравнению с традиционной – на данный момент это намного сложнее, чем это описано в трудах, которые представлены в сети интернет.

Актуальность темы дипломной работы. Электронная коммерция – особая форма торговли, которая способна дополнять привычную форму торговли, а может функционировать без посторонней помощи. Торговая бизнес-модель специализированной узконаправленной категории товаров на растущих интернет-рынках, включая Казахстан, отличается усиленным развитием, в этом состоит отличие от торговых площадок, где предлагается широкий выбор различных услуг и товаров. В данной работе будут изучены теоретические вопросы разработки платформы маркетплейса, в том числе, рассмотрены особенности маркетплейса для образовательной сферы и на основании усвоенного материала разработан электронный сайт, где будут применены современные технологии разработки.

Объект дипломной работы. Объектом дипломной работы рассматривается платформа маркетплейса для образовательной сферы.

Предмет исследования дипломной работы. При исследовании и разработке дипломной работы были использованы технологии php, css и CSM Joomla.

Цель дипломной работы. Целью выпускной квалификационной работы является разработка и создание платформы маркетплейса в виде сайта.

Новизна дипломной работы. Новизна моей дипломной работы – создание платформы маркетплейса для образовательной сферы. В маркетплейсе будут располагаться разные поставщики услуг, то есть разные компании, которые представляют образовательные услуги. Клиент будет выбирать подходящую для себя компанию по цене и локации. В нашей стране существуют разные интернет платформы, которые продают товары и услуги. Но на сегодняшний день не существует платформы маркетплейса для образовательной сферы. И эта особенность является главной новизной дипломной работы.

Основные задачи дипломной работы:

- сделать анализ актуальных систем создания маркетплейсов;
- понять принцип и значение маркетплейсов в современных рыночных условиях, рассмотреть разновидности электронной услуги;
- изучить основные понятия, которые используются при проектировании информационных систем типа «интернет-магазин»;
- определить понятие интернет-услуги, сущность, функции, разновидности;
- разобраться с моделями организации интернет-бизнеса в Казахстане, экономическую специфику задач, которые они помогают решить;
- проанализировать электронные продажи и их системы разработки;
- выполнить проектирование платформы маркетплейса.
- разработать платформу маркетплейса для образовательной сферы;
- оценить, насколько эффективен разработанный программный продукт.

Структура дипломной работы. Работа изложена на 50 страницах компьютерного набора, состоит из двух глав, иллюстрируется 31 рисунком и содержит список использованных источников.

1 Обзор современных технологий и программных средств для разработки маркетплейса

1.1 Понятие маркетплейса и современные технологии разработки

Маркетплейс (online e-commerce маркетплейс) выступает в качестве платформы для электронной торговли, информирующей об услугах или товарах заинтересованных лиц.

Маркетплейс – это оптимизированная интернет-платформа, на которой предлагаются услуги и продукты. Одну и ту же вещь часто можно у нескольких продавцов, причем у каждого товар может стоить по-разному.

Так как маркетплейсы предоставляют товары и услуги от большого числа поставщиков, выбор значительно больше, чем в узконаправленных розничных онлайн-магазинах. С 2014 года в интернете наблюдается значительный рост маркетплейсов, это стало возможно благодаря их востребованности. Если брать заграничные решения маркетплейсов, то можно привести в качестве примера, Shopify, eBay, Amazon, TripAdvisor, Alibaba, Aliexpress, Lyft, Airbnb и многие другие. В принципе, любое из этих приложений является торговой площадкой, хотя бы потенциально. В нашей стране хорошо развивается платформа маркетплейсов. Можем привести такие примеры, как kasperski.kz – интернет-магазин от Каспибанка, ЯндексТакси – можно вызвать такси через мобильное приложения, Glovo – курьерская служба и.т.д.

В 1995 году Пьером Омидьяром была запущена площадка eBay, именно он считается основателем маркетплейса. На сайте собрались коллекционеры и те, кто хотел платить за эксклюзивные вещи.

Основной фактор, который определяет маркетплейс, — огромное число вендоров в пределах одного сайта. Вендор (vendor, provider) — компания, занимающаяся поставкой товаров, изготавливающая и продающая продукцию.

Благодаря многочисленности вендоров маркетплейс может предоставить пользователям больше выгодных предложений. Растущее количество потребителей, проявляющих интерес, делает место более интересным для новых вендоров. Чтобы схема не давала сбоев в работе, маркетплейс отслеживает выполнение трех функций:

- знакомить покупателей и поставщиков;
- сделать проще общение, обмен продукцией и платежами;
- создать нормативно-правовую базу, находящуюся на основе функционирования рынка.

С ростом своей популярности, маркетплейсы постепенно вытесняют интернет-магазины и становятся основным источником торговли в интернете. Чтобы не потеряться в большом количестве имеющихся площадок, ознакомьтесь с примерами из разных рынков.

Суть маркетплейса – в интернете, на базарной площади города – везде, маркетплейс – это то точка, где большое количество потребителей делают

покупки, выбирают из огромного числа предложений от продавцов. Marketplace – это изначально не про «приложение» или «сайт».

В общем, маркетплейс – место, подходящее для того, чтобы можно было осуществлять куплю/продажу товаров и услуг. Marketplace – это в простом понимании, место торговли. В отличие от предложения товаров в виде прайс-листа одного продавца, маркетплейс является местом, где представлены предложения большого количества продавцов.

Главная задача маркетплейса заключается в увеличении числа продаж за счет упрощения диалога между продавцами и потребителями, а также в осуществлении имеющегося потенциала места торговли, которое знакомо и удобно для всех участников рыночных отношений.

Значение маркетплейса:

- все продавцы и их продукция собрана в одном месте;
- предлагается больший ассортимент, чем в офлайн магазине;
- большее число подготовленных потребителей;
- есть возможность просто и быстро отфильтровать товары по цене и характеристикам;
- имеется база с отзывами и рейтингом продавцов, что упрощает взаимодействие;
- уменьшаются транзакционные расходы;
- обеспечивается синергия лояльности: к продавцу, продукции и точке реализации;
- облегчение процесса приобретения: структурирование, категорирование и таргетирование.

Маркетплейс выступает в роли мульти-торговой площадки. Мультиторговля – это множество сделок, которые совершаются в одно время. Под моно-торговой площадкой понимается место, один продавец обслуживает одного покупателя.

Маркетплейс, по отношению к товару, которые находятся на его площадке (поле), может быть:

- мульти-товарной площадкой (к примеру, 100 000 SKU – торговых наименований, которые включают в себя товары народного потребления от бытовой химии до техники);
- специализированной тематической площадкой для реализации определенного вида товаров (это значит, что она нацелена на определенную целевую аудиторию – товары для шитья, автозапчасти, товары для офисных работников, потребители программ для мобильных устройств).
- По отношению к потребителям, маркетплейс может быть:
- площадкой продукции для покупателей (бытовые товары);
- площадкой для специалистов (продукция для профессиональных строителей);
- площадкой для решения бизнес-вопросов (товары для водителей: от запчастей до прицепов).

В привычном смысле маркетплейс – это гипермаркеты, то есть, магазины, у которых на реализации находится большое количество различных наименований, упорядоченных по категориям. Благодаря развитию электронной коммерции, маркетплейс перешел онлайн-формат. На сегодняшний день маркетплейс является агрегаторами, торговыми площадками, дающими возможность размещать разнообразные предложения по оказанию услуг и продаже товаров, в структурированном и удобном для потребителей формате.

Маркетплейс (в e-commerce) представляет собой онлайн-площадку, занимающуюся сбором, систематизацией характеристик товаров и услуг различных предприятий, которые прошли регистрацию в системе и дающая эту информацию, если она потребуется потребителю в структурированном виде, чтобы было удобно сравнивать, выбирать и покупать приглянувшиеся товары.

Если речь идет об онлайн-маркетплейсе, то его смысл заключается в уделенной базе данных держателей товаров и самой продукции: стоимости, их числа, условий реализации, к которой могут без посторонней помощи подключиться продавцы и потребители, первые – создавая предложения, другие – проявляя намерение для их приобретения.

Подключиться к маркетплейс можно:

- с помощью веб-браузера ПК;
- с помощью веб-браузера смартфона;
- с помощью мобильного приложения;

Рассмотрим подробно разновидности маркетплейсов.

Супермаркет – точка реализации разноплановых товаров, которые разделены по категориям;

Прайсовый агрегатор занимается сбором, группировкой товаров и их стоимости из разных мест, предназначена для объединения данных, собранных в разнообразных источниках в один;

Классифайдс является одной из разновидностей маркетплейса, площадка по оказанию услуг, позволяющая делиться объявлениями продавца о продаже, преимущественно подержанных вещей.

Бизнес Маркетплейс, в отличие от бизнеса Продавца состоит в том, что продавец занимается реализацией своей продукции, а владелец Маркетплейс занимается предоставлением «полок» для этой продукции, на них продавцы выставляют предметы, а потребители могут изучать содержимое этих «полок» и покупать то, что им необходимо. «Полки» могут быть и физическими и в виртуальном формате.

Платформа маркетплейса (виртуальная торговая площадка, онлайн биржа труда, онлайн-магазин) может быть рассмотрена как прикладная система, представляющая собой элемент технологии системы электронной торговли. Наравне с физическим магазином, маркетплейс занимается осуществлением функций: предоставляет услуги и товары потребителю, принимает заказы, продает и доставляет покупки.

Довольно распространенным является еще одно определение понятия, более простое, – сайт, который торгует товарами в интернет-пространстве.

Платформа маркетплейса дает возможность покупателям оформить заявку на приобретение, определиться каким способом расплатиться и как будет доставлен заказ в онлайн-режиме.

Основное различие маркетплейса и традиционного магазина заключается в расположении и осуществлении коммуникации с посетителем, применения сети для того, чтобы осуществить все доступные операции.

В интернет-магазинах присутствуют специализированные понятия, в том числе провайдер электронной коммерции – предприятие, которое занимается предоставлением услуги по реализации коммерческой деятельности в интернете. Основным способом оплаты в сети – пластиковая карточка и электронные денежные средства. Провайдерами систем электронной коммерции являются предприятия, имеющие непосредственную связь с банковскими структурами и с интернет-провайдерами. Число провайдеров электронной коммерции регулярно увеличивается.

Наравне с любой системой коммерции электронная, имеет непосредственную связь с расходами на создание и содержание онлайн-магазина. Траты на разработку электронной коммерции или определенного интернет-магазина имеют зависимость от некоторого количества факторов: характер торговли, размеры и финансовые возможности предприятия, наличие у предприятия вычислительной техники и доступа к интернету, имеются ли в штате программисты.

В числе наиболее важных затрат на разработку и последующее развитие платформу маркетплейса выделяются такие компоненты, как:

- аппаратное обеспечение (потребуется надежный сервер с бесперебойными каналами связи);

- ПО (могут быть задействованы бесплатные системы, однако, создание структуры и ее сопровождение отнимает крупную сумму денег);

- затраты на маркетинг (рекламирование маркетплейса, только при задействовании платных систем получится поместить макетплейс на первые страницы поисковиков);

- траты, связанные с сопровождением системы (выплаты дополнительному персоналу, обучение новому, поддержание функционирования сервера);

- у крупных маркетплейсов имеются собственные представительства (офисы, склады) в каждом крупном городе Казахстана, поэтому к расходам стоит добавить и их обслуживание.

В целом интернет-магазин может быть рассмотрен с двух подходов, как к части торговой деятельности офлайн магазина в интернете и как к отдельной самостоятельной коммерческой единице, это позволяет увеличить охват клиентов, добавляя интернет-покупателей.

Стоит упомянуть, что на данный момент можно наблюдать скрещивание систем обычной и онлайн торговли, где преобладает одна сторона, в зависимости от вида деятельности.

Так как при онлайн продажах у клиента отсутствует возможность хорошо изучить товар или услугу при подписании договора, то виртуальные продажи принадлежат относится именно к удаленному способу реализации товаров.

Реализация услуг и товаров в онлайн-магазинах осуществляется после заключения розничного договора купли-продажи, который является основой отношений между продавцами или поставщиками и потребителями.

Информация о товаре или услуге, написанная на сайте продавца, в которой содержатся все имеющие важное значение пункты договора розничной купли-продажи, считается публичной офертой. Соответственно, у продавца есть обязательство заключать договор с любым лицом, которое желает купить его товар, который предложен в описании. Подписывать договор с каждым потребителем не требуется.

1.2 Программные средства для разработки маркетплейса

Существует большое число разнообразных вариантов как сделать интернет-сайт, все эти варианты можно разделить в зависимости от степени автоматизации на два вида: визуальный и программируемый. Первый из них позволяет разработку web-сайта с высоким уровнем автоматизации. Данный метод появился благодаря необходимости сделать менее трудоёмким процесс разработки вебсайта и уменьшить время, затраченное для выполнения задачи. При применении данного способа, разработчиком создается страница на экране визуального редактора, онлайн-овый или программный, после на каркасе программа генерирует необходимый HTML код для всех страниц. Главным принципом визуальных редакторов является следующая формулировка WYSIWYG, которое расшифровывается как «What You See Is What You Get», в переводе означает «То, что видно при разработке, то и получается при просмотре странички». При разработке веб сайта, где применяется визуальный метод, в качестве инструментов обязательно выбираются редакторы визуального конструирования. Создание веб-сайтов с помощью данных редакторов пользователю дается шанс работы без доступа в интернет. После создания готовый веб-сайт размещается в интернете. Как описывалось ранее, визуальные редакторы делятся на 2 большие группы, имеющие колоссальные отличия. Первая группа – онлайн-овые редакторы. Под ними понимаются специализированные механизмы, применяемые при разработке вебсайтов (преимущественно несложных), которые находятся в Интернете, в основном на сервисах, предоставляющих хостинг. К примеру, ucoz.ru, boom.ru, wix.ru, tilda.cc. Второй группой являются программы клиенты (офлайновые), которые размещаются на ПК. С помощью второго вида появляется возможность работать с масштабными проектами. Самые широко распространенные редакторы: Adobe GoLive, NamoWEbEditor, DreamWeaver (Macromedia), Microsoft FrontPage и пр. Однако, даже при создании веб-сайта с визуальным редактором, нужно иметь

знания основ языка HTML, так как у любой редактор может допустить ошибку в коде документа. Разработчик, имеющий знания основ языка, сможет исправить все допущенные ошибки [9].

К программируемым способам можно отнести написание веб-сайтов на Фреймворках и языках веб-программирования, как Ruby, Perl, ASP.NET, PHP, Java и пр., но нужно иметь ввиду что языке гипертекстовой разметки Html является базой любого кода веб-сайта. Именно поэтому его знаниями должны обладать разработчики. Часто знания, которыми обладают программисты, трудящиеся над проектом становятся решающими факторами при выборе Фреймворка и языка программирования. Большая часть задач успешно решается при любом современном популярном языке программирования.

Важно понимать, чем отличается Фреймворк от языка программирования. Язык программирования подразумевает определенный базовый синтаксис с наличием стандартных библиотек, на основе которого разрабатываются определенные программы. Фреймворк используются для получения доступа к необходимым библиотекам, с помощью которых процесс создания приложений и веб-сайтов становится проще. Языки и Фреймворки, такие как ASP.NET и JSP, выступают как единое целое. Языки PHP и Perl могут применяться, без помощи Фреймворка. Все языки веб программирования можно поделить на клиентские и серверные. Как понятно по самому названию клиентские языки применяются при написании программы, которые функционируют на стороне клиента, например, веб-браузер; а серверные языки используются для функционирующих на компьютере программ. При упоминании клиентских типов языков веб программирования обязательно нужно упомянуть JavaScript, так как он наравне с HTML, является основой большинства web-технологий (ярким примером может выступить распространенная технология AJAX), поэтому возможность использовать его при программировании – это базовый навык веб-разработчика [10].

Главный минус – меньше, если сравнивать с *nix, это количество недорогих хостингов, а также то, что в ситуации с выделенным хостингом потребуется покупать серверную лицензию. Но при сравнении с тратами на разработку сложных веб сайтов, а, также количеством необходимого трафика, победа за PHP, который безоговорочно можно назвать распространенным языком для веб программирования. Его главные достоинства это: простота синтаксиса, высокая скорость, а также его поддержание большим числом хостингов. Основным достоинством является, использование для написания большинства CMS (Content Management System), таких как – WordPress, Drupal и Joomla, которые являются самыми популярными. Один из достаточно широко используемых языков веб-программирования на платформе Unix это – Perl. Язык программирования Perl имеет сложный и запутанный синтаксис, потому что не был предназначен для веб программирования [11].

JSP (Java Server Pages) одна из технологий J2EE, который используют при разработке веб сайтов на Java. JSP очень похож ASP.NET, поэтому при выборе между этими технологиями, в основном отталкиваются от личных вкусов, не

обращая внимания на их минусы и плюсы. В настоящий момент все большую популярность получил Фреймворк Ruby on Rails и язык Ruby. На его основе возможно в короткие сроки разработать веб-сайт с необходимыми функциональными особенностями. Самый большой недостаток Ruby – невысокое быстродействие [12].

Отдельно стоит отметить разработку веб сайта при помощи CMS, перевести можно как «система управления содержимым», в конкретной ситуации подразумевается содержимое веб страницы. CMS – это достаточно обширное понятие, туда входят простые системы, состоящие из пары скриптов, так и сложные универсальные системы, которые используются при решении возникающих задач разработки веб-сайтов. CMS является настраиваемой системой и используется для простоты и надежности управления информацией с web сайта. После стандартизации главных составляющих веб-сайтов появились CMS. Рубрикации, структуры, методы подачи материалов, на базе которых были созданы первые системы для создания и развития web-сайтов. Системы CMS стали популярными после того, как для создания начала применяется модульная структура, благодаря которой, CMS трансформируется в зависимости направлений web-сайтов. Так же системы стали популярными по причине невысокой стоимости, за которую клиенты получали достаточно мощную функциональность. Новейшая система управления контентом ассоциируется с конструктором, который помогает при создании и сопровождении личного web сайта. Можно создавать и развивать новые рубрики, проводить обновление и загружать файлы. Любая CMS основана на принципе, что большая часть web сайтов обладает схожей структурой и рубрикой, но отличается внешним видом и различным содержимым. Во многих CMS есть определенные пункты, которые обычно включены в web-сайт: контакты для связи, цены, сводка новостей, информационные статьи, описание фирмы и пр., ими можно управлять через раздел администрирования web сайта. Для создания структуры и навигации на web сайте в CMS не нужно знаний html, ведь эти системы созданы как конструктор, чтобы данная разработка делалась визуально, чтобы неопытный пользователь смог справиться. Любая CMS имеет разнообразные варианты доступа. Как обычно администратор имеет доступ ко всем без исключения разделам, то есть менять структуру web сайта, также любой раздел. Редактор же имеет возможность менять текст на сайте, но не имеет доступа к структуре рубрик, добавлять новые, удалять старые. Пользователь может вносить корректировки в небольшой части рубрик, либо в определенные материалы, находящиеся в них. Так скажем, CMS является многопользовательской системой, которая позволяет предоставлять различный доступ к web сайту различным группам. Некоторые CMS имеют разные шаблоны дизайнов. Также с помощью встроенных визуальных средств CMS можно поработать над дизайном: поработать с графическими компонентами, шрифтом, цветом и пр. [13].

Первое мнение о веб-сайте у посетителей складывается после того, как они знакомятся с его отдельными веб-страницами. Во многом то, как будет оценен

веб-сайт зависит от того, как будет оценена каждая отдельная страница. Поэтому, следует особое внимание анализу потребностей аудитории, в особенности стоит хорошо обдумать оформление страниц на веб-сайте. Так как веб-сайт выступает в роли интерфейса, при помощи которого предприятие будет поддерживать связь с потребителями, то необходимо сделать анализ эффективности всех страниц веб-сайта.

При создании web сайта не стоит игнорировать важные детали. Проводя анализ внешнего вида веб-страниц, должны быть рассмотрены некоторые важные моменты. Первое, обязательно учитывается этикет веб. Какие элементы дизайна могут выдать непрофессионала? В веб существуют некоторые традиции, наполнение отпугнет посетителя, если оно выглядит раздражающе. Помимо этого, просмотр веб-сайтов конкурентных компаний подскажет, что является хорошим инструментом, а что нет [14].

Дизайнеру, стоит хорошенько над стилем пользовательского интерфейса веб-сайта и определить наиболее удачный вариант для того, чтобы преподнести информацию. Важно понимать, каким деталям стиля пользовательского интерфейса необходимо уделить больше внимания и то, какая информация должна быть прописана на каждой странице.

Графика веб сайта, картинки всегда хорошо воспринимаются пользователями. Они разбавляют длинные страницы с текстом. Но графика забирает существенную часть полосы пропускания линии связи, которая является соединяющим звеном между посетителями веб-сайта и сетью. Если у картинки большой размер, то на ее полную загрузку может уйти слишком много времени, возрастает вероятность потери заказчиков, которые ждут, когда прогрузится картинка. Поэтому заполняя сайт графикой необходимо следовать правилам, стоит придерживаться правилам, которые будут рассмотрены дальше. Использование графики должно быть минимальным. Кроме проблем с полосой пропускания, имеются и другие причины не игнорировать этот момент. Картинки могут отвлечь внимание, если с ними переборщить, или они достаточно большие, или анимированные. Они значительно больше, чем текст, поэтому на загрузку уходит больше времени. Не исключено, что пока будут загружаться изображения, посетители могут ознакомиться с информацией на веб-сайте. Однако, если у иллюстраций важная роль или на них расположены важные данные, то ожидание будет раздражать [15].

Для создания маркетплейса потребуются такие инструментальные средства разработки, как:

- набор веб-разработчика «Денвер», в который входит набор дистрибутивов (Apache, PHP, MySQL, Perl и т.д.)

- программная оболочка, применяемая при создании сайтов на ПК: Adobe Photoshop CS6 (для создания логотипа), Notepad++ (для быстрого редактирования).

- браузер Google Chrome, Internet Explorer, Mozilla Firefox с плагином FireBag – для того, чтобы протестировать продукт в различных условиях.

MySQL – это наиболее популярная и распространенная СУБД в интернете. Ее использование отлично подходит как для маленьких сайтов, так и для достаточно больших. Она быстро стала популярной, ведь зарекомендовала себя, как удобную, надежную, высокоскоростную. Даже начинающие программисты, как правило, не испытывают трудностей при работе с ней. Еще одним несомненным достоинством является то, что она бесплатная [16].

На эффективность некоторых страниц web-сайта влияют различные элементы: ссылки, цвет, структура, графики, шрифт. Конечно, страницы могут и не основываться на принципах, описанные в этой главе, но нужно знать что есть потенциальные ловушки, связанные при применении дизайна страницы некоторых элементов.

В ситуации нашего маркетплейса для образовательной сферы при маленьком количестве ассортимента не запланированы большие объемы продаж, именно поэтому применение универсальной CMS имеющий плагин онлайн магазина полностью целесообразно.

Разберем методы выбора программного средства. Для начала выделяем критерии, которые ищем в CMS маркетплейса. При создании помним, что имеется большое количество подходов и методик для создания маркетплейса, рассчитанных на реализацию главных функций новейшего маркетплейса и нужно выбрать самый подходящий вариант.

Итоги исследований, которые проводились разными изданиями говорят, что большей части интернет-потребителей нравится смотреть продукцию и выбирать продукцию, покупать предварительно не проходя регистрацию. Поэтому одной из функций является возможность того, чтобы контент был доступен при гостевом входе, CMS должна давать незарегистрированным пользователям возможность выбирать и совершать покупки.

Безопасность – это второй среди важных, но первый по значительности критерий, которые предъявляются при отборе системы в условиях проведения разнообразных операций с финансами. Безопасность подлежит рассмотрению с двух позиций – защита операций с финансами и защита личной информации клиента. Для определения безопасности программного продукта выделяют несколько подходов, начиная от соотношения коэффициента востребованности с интенсивностью нахождения критических уязвимостей до сертификатов безопасности, систем шифрования информации и применения защищенных протоколов. Основываясь на статистических данных онлайн-исследований очевидные решения считаются более безопасными, потому что разработчики больше внимания уделяют защите информации и тестированию кода, вовремя исправляют найденные ошибки [17].

Для написания многих систем управления онлайн-магазином использовался PHP, как язык программирования с добавлением баз данных MySQL и современных фреймворков. Данный вид сайта поддерживается почти всеми существующими хостингами. Несмотря на то, что языковая основа одинакова, для корректного функционирования все приложения нуждаются в разном количестве ресурсов. Встречаются случаи, когда необходимы большее

число ресурсов для менее функциональных решений, чем для универсальной системы управления контентом. О производительности необходимо узнавать на начальном этапе, для этого можно использовать различные системы тестирования ресурса и сервисы.

При успешном функционировании маркетплейса, а в этом заключается главная цель создания сайта, будет появляться задача увеличить количество товара, увеличить пропускную способность, модернизировать интерфейс, повысить функциональность. В этих ситуациях систему нужно будет расширить или модернизировать. Для этого, в большинстве случаев, нанимаются сторонние разработчики. Популярные решения смогут сделать проще процесс по подбору специалиста-разработчика чтобы он выполнил все необходимые доработки. Чаще всего услуги программиста для распространенных решений стоят дешевле. Поэтому нового CMS-кандидата нужно рассматривать с точки зрения возможности расширения. Для принятия решения по выбору системы есть плюсы: детальные примеры, предоставленная техническая документация, видео уроки для разработчиков.

Ответственность за ценообразование, учет остатков и хранение иной вспомогательной информации, создание каталога продукции лежит на плечах корпоративной информационной системы. Обычно в этой роли приходит одно решение на платформе 1С: Предприятие. Для запланированного взаимодействия, большое внимание уделяется поставляемому с CMS модулю интеграции, который. Не многие CMS предлагают полную интеграцию с КИС, нужно уточнять заранее. Нужно помнить, что разработка такого модуля для некоторых CMS является дорогостоящим мероприятием, вероятно легче выбрать другое решение, чем страдать над разработкой.

Базируясь на критериях, описанных выше, нужно выбирать популярную CMS систему. CMS систем много, поэтому при отборе стоит добавить еще несколько критериев.

В интернете представлена 4 типа CMS, которые могут применяться для создания электронного магазина:

- коробочные (коммерческие продукты, они дают возможность сделать сайт своими силами);
- студийные (системы, на них процесс разработки сайта может быть осуществлен только из разработчиком)
- open-source системы магазинов (программы, находящиеся в открытом доступе, имеющие открытый код);
- open-source системы регулирования контента с модулями онлайн-магазинов.

Чтобы сделать выбор, стоит более подробно разобрать каждую систему. Коммерческую систему и коробочную версию не берем в расчет из-за серьезных финансовых вливаний, которые не могут выплатить маленькие предприятия.

Разберем пользующиеся популярностью CMS (Content Management System) системы, имеющие плагины для платформы маркетплейса.

WordPress CMS главным образом позиционирует себя как движок для блога. Основные достоинства Wordpress:

- простота первичной настройки;
- несложно установить;
- в большом количестве представлены дополнения и шаблоны;
- можно найти большое число сайтов фанатов и форумов;
- у сайта несложное управление;
- во время доработки на основе этого движка возможно создать сайты какого угодно типа;
- есть шанс делать публикации, прибегая к помощи сторонних скриптов.
- Главными минусами Wordpress являются:
- даже когда на сайте маленькая посещаемость, сервер испытывает повышенную нагрузку;
- плагины конфликтуют между собой;
- для нормального функционирования потребуется установить огромное количество различных дополнений.

Joomla – эта CMS немного тяжеловата для сайта, но в то же время отличается большими функциональными возможностями. Joomla обладает только одним минусом – весьма странный интерфейс панели управления администратора, к которому не так просто привыкнуть. Там нет ничего удобного. Грузить статьи неудобно, работать с обновлениями неудобно и т.д. однако, со временем ко всему можно привыкнуть.

Joomla также наделена множеством шаблонов и дополнений, за которые не требуют оплаты. безусловно, чтобы разработать полноценный веб сайт, на начальном этапе лучше обзавестись некоторыми плагинами.

Среди плюсов некоторые отмечают стабильное обновление движка joomla, но если посмотреть под другим углом, то это больше минус, ведь постоянно нужно будет их отслеживать и устанавливая обновленные версии. Также это может свидетельствовать о том, что создатели регулярно ошибаются и таким образом правят недочеты. Еще один минус, который скорее не касается новичков – избыточность и сложность программного кода, которая имеет отрицательное влияние на скорость, с которой функционирует сайт, а также затрудняется поиск и исправление недоработок. Joomla доступен для скачивания на joomlaportal.ru [18].

Drupal предоставляется бесплатно, также находится в топе самых распространенных cms для сайта. В 2010 году drupal заняла первое место как лучшая cms. Ее используют в своей работе многие влиятельные компании, обладает огромным активным сообществом, в котором состоят пользователи и разработчики, это облегчает знакомство с системой.

Drupal подходит для применения при построении различного рода веб страниц: блоги, новостные порталы, информационно-аналитические сайты, форумы, социальные сети и др. и без того огромную функциональность, можно сделать еще больше, просто скачав дополнительные модули на официальном

сайте. В нее по умолчанию установлено несколько функций: лента с новостями, голосование, сбор новостей, форум, движок для блога, поиск и загрузка файлов. Основной минус drupal, которые первым бросается в глаза – со стороны сервера предъявляется завышенное требование к оперативной памяти, минимум 128 МБ. Загрузить drupal можно на сайте drupal.ru

Анализируя возможности разработки маркетплейса для образовательной сферы, выбор падает на CMS Joomla. Эта CMS относится к гибкой системе управления сайтом, так как для нее пишется множество число модулей. В официальном каталоге наравне с платными расширениями, имеются те, которые не требуют оплаты. Данная система наделена рядом плагинов для разработки платформы маркетплейса, более подробно на них остановимся немного позже. Платформа маркетплейса на Joomla, может быть рассмотрена для крупных компаний и не будет серьезно уступать платформам, которые были разработаны, как раз для электронной коммерции.

Подведем итог. Для крупных маркетплейсов применяем движки специально разработанные для интернет коммерции, так же можно использовать Joomla вместе с плагином интернет-магазина [19].

1.3 Серверные технологии для разработки маркетплейса

Как уже было отмечено ранее, на серверной части сайта лежат все функциональные возможности, это влияет на то, что и число языков программирования данного раздела сайта существенно больше, чем клиентской и, помимо этого, не существует единой технологии, которая бы выступала как стандарт разработки. Среди этого типа языков выделяются Ruby, ASP, JSP, Python, Perl, но большее распространение (свыше 5 млн серверов) у языка PHP версии 5 и 5.6. Особенность этого языка, конкретно его 5 версии, заключается в том что, полноценной поддержке объектно-ориентированной технологии программирования, это облегчает работу повторного применения кода, делает лучше его читаемости и дает возможность команде разработчиков трудиться над разработкой сайта. На языке PHP существует большое количество каркасов разработки структурных и функциональных элементов сайта – фреймворков. Самые распространенные: Doctrine, Symphony, Zend, Drupal, Yii, Smarty, Wordpress и пр. Среди них существуют специализированные и универсальные, которые служат для того, чтобы реализовать задачи веб разработки и определенные функции. Используя фреймворки, облегчается написание более стандартного и читаемого кода, можно обеспечить существенное ускорение процесса разработки. На основе фреймворков созданы CMS-системы управления контентом сайта, специальные решения, позволяющие развернуть структуру сайта и добавить в нее весь необходимый функционал, который зависит от сайта.

На современных динамических сайтах в обязательном порядке используется база данных для того, для хранения разнообразного контента, а

также еще для того, чтобы реализовывать свои функции. Самые распространенные СУБД: MySQL, PostgreSQL, ORACLE (см. рисунок 1). Это облегченные системы, это значит, что наделены недостаточным функционалом, которого бы хватало, чтобы решить все задачи веб разработки. Среди них более популярна MySQL. Она проста в эксплуатации, но обрабатывает запросы на высокой скорости, это играет важную роль, когда имеется большое число обращения пользователей к серверу, где находится сайт.[20]

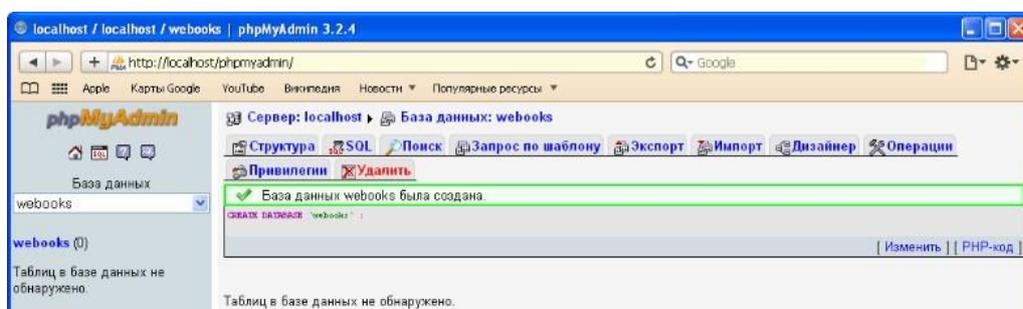


Рисунок 1 – Веб-интерфейс phpMyAdmin для администрирования СУБД MySQL

В back-end самым распространенным является пара PHP и MySQL. PHP имеет большое число средств, для организации масштабных работ с базой данных. Не будем углубляться в описание языка программирования, приведены и описаны фрагменты работы с программным кодом во второй части дипломной работы.

При создании сайта, неважно какая CMS лежит в основе выделяют 2 главных подхода:

I) Система запускается на ПК, после чего разрабатывается веб сайт, заполняется все необходимой информацией, а затем устанавливается на сервере. Этот вариант обладает одним достоинством – если что-нибудь сделать неправильно, то это будет находиться на компьютере, где это делалось и очень много недостатков, система может неправильно отображаться на сервере, различные базы с данными и версии языка, перенастройка файлов с доступом к ресурсам. Описанный способ применяется при получении базовых принципов взаимодействия с web системами.

Установить CMS на ПК довольно легко, но, несмотря на это потребуется соблюдать определенную последовательность действий. Нет смысла расписывать каждый шаг установки, но некоторые важные моменты будут коротко рассмотрены. На ПК должен быть установлен база данных MySQL, язык PHP, Сервер Apache. Необходимо установить на ПК программную платформу XAMPP и настроить.

Когда платформа будет установлена, используя компонент phpMyAdmin необходимо сделать базу, универсальную для Joomla и VirtueMart.

II) Систему переносится на сервер, система разворачивается на сервере, при ее разработке можно следить за итогом создания. Если в процессе не

забывать про резервное копирование, то у этого способа нет недостатков. Он более сложный и ответственный и именно он был применен для создания маркетплейса образовательной сферы.

Чтобы сайт можно было разместить на хостинге стоит определиться с незанятым доменным именем, после определиться с хостом провайдера.

В сети находятся хостинги, которые требуют оплаты и за которые платить не нужно. Аккаунт, который не требует оплаты весьма ограниченный, запуск интернет-магазина, установка плагинов может спровоцировать блокировку.

Имеется различные подходы к созданию сайтов. В зависимости от места создания, выделяются следующие:

- создание сайта осуществляется на локальном устройстве, как было описано ранее, размещается локальный сервер на устройстве программиста, преимущественно Apache и сервер, на котором отсутствуют данные MySQL (как правило, это уже созданный пакет программистом XAMPP или WAMP)

- создание сайта осуществляется на сервере. Мы прибегнем к помощи второго способа создания, в соответствии с Agile технологией ПО и уже на стартовых итерациях создания сайт станет доступен в интернете. Он сразу готов к использованию, а в это время его функциональные возможности будет потихоньку увеличиваться и становиться шире, пока создание не будет завершено. У такого способа больше перспектив еще и потому, что SEO оптимизация сайта стартует на начальных этапах его создания, т.к. он уже в самом начале должен быть зарегистрирован в поисковиках, а к завершению создания сайта, поисковики уже хорошо его знают.

В обоих случаях при переносе сайта подразумевается применение специально разработанных программ-клиентов. Среди них внимания заслуживают следующие: FileZilla, FireFTP, c-panel.[21]

FileZilla — многоязычный свободный FTP-клиент с исходным кодом, открытым в доступе для Linux, macOS и Microsoft Windows. Программа поддерживает SFTP, FTP, и FTPS (FTP через SSL/TLS) и имеет настраиваемый интерфейс, то есть пользователь может сменить тем оформления. Имеет возможность синхронизацией каталогов, перетаскивания объектов и еще поиском на удалённом сервере. Они являются не чем другим, как FTP-клиентами, они наделены стандартным двухпанельным интерфейсом размещения на хостинге. Чтобы перенести систему на хостинг был задействован FTP- клиент FileZilla (см. Рисунок 2).

Данный процесс не нуждается в подробных характеристиках, за исключением одного. Так же, как и на локальном сервере, средствами клиента phpMyAdmin разрабатывается база данных на сервере. Она применится в момент установки (развертывании системы).

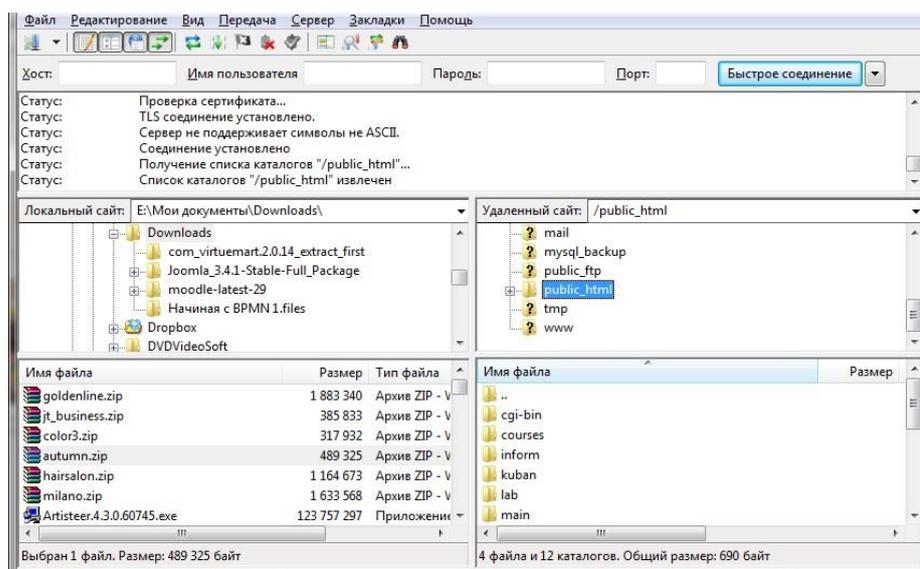


Рисунок 2 – FTP-клиент – FileZilla

Для того, чтобы отредактировать код элементов системы применялся популярный кодовый редактор Sublime Text 2 (см. Рисунок 3). Sublime Text – удобный текстовый редактор. Поддерживает разные плагины на языке Python. Разработчик редактора позволяет бесплатно и без ограничений использовать программу, однако сама программа извещает о необходимости приобретения лицензии [22].

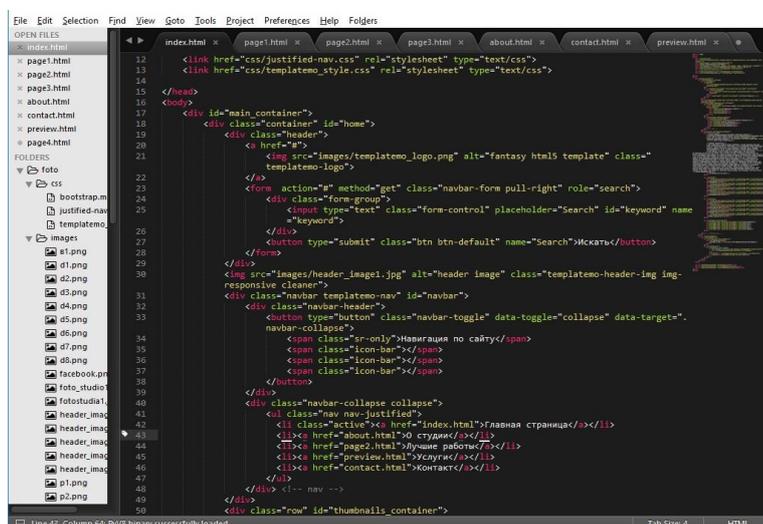


Рисунок 3 – Кодовый редактор Sublime Text

2 Платформа маркетплейса для образовательной сферы

2.1 Структура и элементы дизайна маркетплейса

Структурой сайта называется деление портала на логические блоки, которые расположены в определенной последовательности. Другими словами, это система распределения разделов для услуг компании, страниц с информацией, карточек услуг, служебных страниц. Изобразив структуру на листе или экране компьютера, вы с первого взгляда сможете оценить полную картину маркетплейса (см. Рисунок 4).



Рисунок 4 – Структура маркетплейса

Рассмотрим подробнее на примере маркетплейса для образовательной сферы. Выберем по одному пункту из всех иерархических блоков. Главная – это начальная страница, Категории – направление курса (пример: электрика и электроника, бизнес, медицина, иностранные языки, сельское хозяйство, фильм и медиа, музыка, IT: для IT специалистов), Компании и репетиторы – учителя, спикеры, названия компании, свойство – программирование на Java, курсы юридической грамотности и т.д. Свойством может быть вид, цена услуги, и другое. После нажатия на какое-нибудь свойство системой будет выдано, к примеру, курсы английского языка от всех компаний.

Возможно ли при создании структуры сайта пропустить этот шаг? Можно ли остановиться на примерной структуре? Сделать это не получится. Причины заключаются в следующем:

- из-за плохой клиентоориентированности у портала будет небольшое количество посещений;
- сократится время, которое один пользователь будет проводить на сайте (пользователь заход – сталкивается с проблемой – закрывает страницу);
- снижение информационной ценности портала (посетителю будет тяжело отыскать необходимую информацию);
- снижение лояльности посетителей (наш маркетплейс будет главным местом покупок, а не запасным вариантом);
- обнаружатся аналогичные страницы;
- возникнут сложности с тем, чтобы добавить новые услуги, страницы.
- Сайт с плохой структурой неудобно использовать посетителям, а также самим работникам компании [23].

Теперь хотела рассказать о служебных страницах. Служебными страницами называются подразделы на сайте, где представлена информация для посетителей, которая не имеет непосредственного отношения к услугам. Независимо от того, что все это кажется неважным, это значимые части маркетплейса. К примеру, потенциальным покупателем был найден интересующий его образовательный курс, он собирается купить его, но не может отыскать на сайте информацию об условиях курса. Ему легче посетить сайт конкурента, где все просто и понятно и записаться на курсы, чем пытаться получить необходимую информацию через общение со службой поддержки (в особенности, если заявки принимаются только с помощью электронной почты).

Список обязательных служебных страниц:

- FAQ (вопрос и ответы);
- условия гарантии;
- способы оплаты;
- курсы;
- учителя;
- контактная информация учителя;
- сертификаты;
- контакты для связи;
- адрес компании.

Примеры хорошей структуры маркетплейса

Отличное решение – в одном разделе располагается одна услуга. Если избежать повторов не получается никаким способом, то лучше не использовать больше двух. Данная информация не относится к свойствам.

Маленький уровень вложенности. Не стоит пытаться уместить несколько десятков услуг в один раздел маркетплейса. Лучше подумать над тем, каким образом сегментировать услуги. Эта рекомендация также относится к количеству разделов. Если пользователь кликнул на какую-нибудь категорию, то не должно сразу вылезти несколько десятков подразделов. Достаточно создания

структуры: Категории – Профессии и навыки – Бухгалтерский учет или Категории – Иностранные языки – Английский язык. Лучше дать пользователю самостоятельно выбрать, воспользовавшись поисковым фильтром.

Необходимо оперировать понятиями, которые будут понятны даже детям (см. рисунок 5). Ниже представлен пример хорошей структуры.

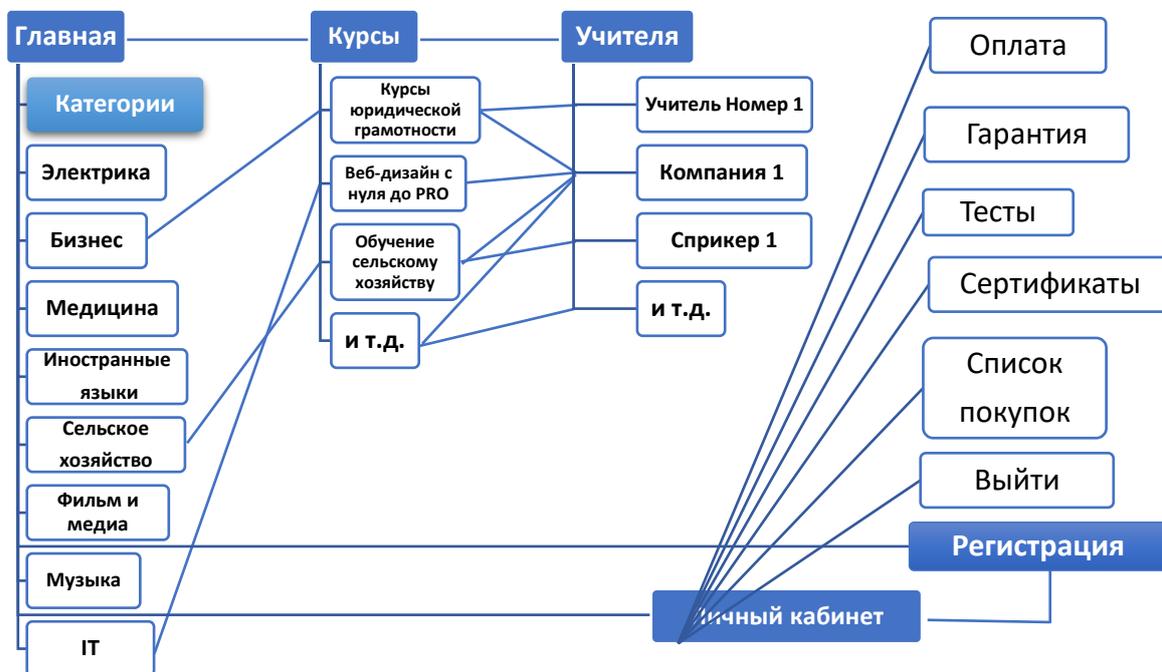


Рисунок 5 – Структура маркетплейса и связь учителями

На этой схеме видно, что страницы с авторизацией и регистрацией находятся в разных разделах структуры. Присутствуют также и служебные страницы, в том числе присутствует переход в социальные сети.

Из этого можно сделать определенные итоги:

- Структура маркетплейса является проводником посетителя. Без нее покупатель растеряется и найдет иной структурированный маркетплейс. На сетевом рынке наблюдается огромная конкуренция, поэтому даже из-за небольшой недоработки все дело может провалиться.

- Структура маркетплейса начинает разрабатываться еще во время проектирования сайта. Перед тем, как запускать портал в работу, желательно протестировать: к примеру, можно прибегнуть к помощи фокус-группы, которая оценит работу сайта и укажет на недостатки.

- Разработать структуру маркетплейса относительно легко, особенно если этим занимается человек, который точно знает, что нужно делать. Поэтому не стоит заниматься экономией на этом этапе, так как это может отразиться в будущем.

Теперь можем приступить к дизайну нашего маркетплейса. Для того, чтобы решить эту проблему создадим шаблон (template) для нашего маркетплейса, в нем можно поработать с настройками оформления, выбирать разную цетовую гамму, дописывать необходимые данные. С настройками довольно просто работать, используя административную панель Joomla.

Для решения данного вопроса, хватит знаний верстальщика, потребуется некоторое время, чтобы понять структуру и логику работы. Таким способм интегрируются любые виды сайтов – от визитки до маркетплейса.[24]

Немного хочу обсудить о структуре нашего шаблона. На 6 рисунке показано все файлы нашего шаблона.

Имя	Дата изменения	Тип	Размер
css	08.05.2020 18:47	Папка с файлами	
fonts	08.05.2020 16:15	Папка с файлами	
html	08.05.2020 16:15	Папка с файлами	
images	08.05.2020 16:15	Папка с файлами	
js	08.05.2020 16:15	Папка с файлами	
language	08.05.2020 16:15	Папка с файлами	
error.php	07.05.2020 17:39	Файл "PHP"	2 КБ
index.html	07.05.2020 17:01	Chrome HTML Do...	1 КБ
index.php	08.05.2020 18:57	Файл "PHP"	6 КБ
template_preview.png	07.05.2020 16:58	Файл "PNG"	29 КБ
template_thumbnail.png	07.05.2020 17:00	Файл "PNG"	22 КБ
templateDetails.xml	08.05.2020 18:56	Документ XML	4 КБ

Рисунок 6 – Структура шаблона

Первым делом нужно создать папку и файлы, со следующим наполнением:
 – файл «templateDetails.xml» – здесь хранятся все данные, касающиеся разработчика, версии, здесь же прописываются данные, касающиеся наполнения, модулей, файлов с языками и др.

– картинки «template_preview.png, template_thumbnail.png» – образ шаблона, в любом графическом редакторе можно сделать в формате иконок.

– файл «index.php» – основной файл, в нем расположится подключение скриптов, параметров, модули. Также именно в данном файле находится верстка.

– файл «index.html» – файл html, в котором ничего нет. Его следует добавлять в каждую папку. Это поможет защититься от злоумышленников, они не смогут ознакомиться с наполнением.

– файл «error.php» – страница 404 ее создание происходит по своему желанию, совсем не обязательно использовать стандартную страницу 404 Joomla, имеющую не очень презентабельное оформление.

– папка «language» – данные Файлы с языками стоит создавать в том случае, если планируется мультиязычный шаблон.

– папка «js» – здесь расположены скрипты. Постепенно будет подключено все то, что требуется для функционирования портала.

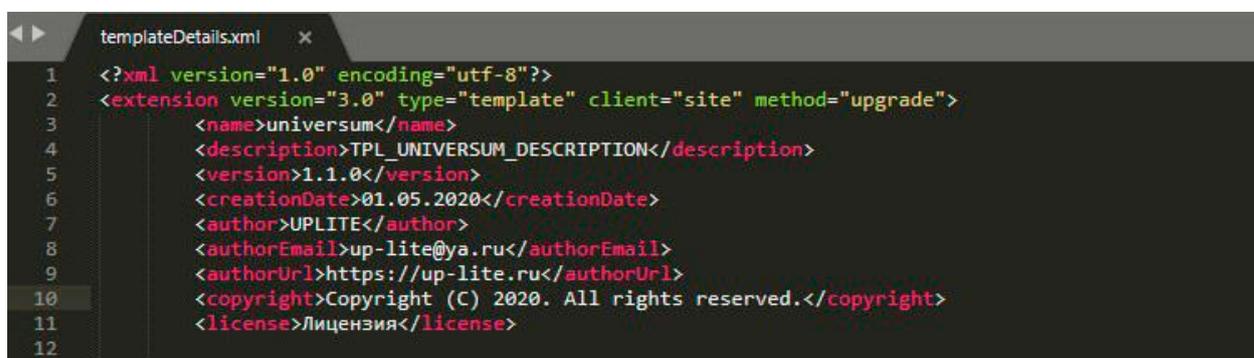
– папка «images» – как понятно из названия, в этой папке расположены изображения шаблона. В ней лучше всего держать лишь те изображения, которые имеют прямое отношение к оформлению, потому что загружать изображения для информации к шаблонам – не самый удобный вариант.

– папка «html» – предназначена для переопределения модулей и частей Joomla. В ней находится та информация, которую не нужно перезаписывать после обновления.

– папка «fonts» – сюда подключаются шрифты для портала, которых нет на Google Fonts. Их желательно подключать с помощью главного шаблона. Это необходимо для более высокого оценивания в Google Page Speed.

– папка «css» – здесь находятся стили шаблонов, в будущем они подключатся к имеющемуся шаблону [25] .

Внутри файла «templateDetails.xml» идет заполнение данного файла данными, характеризующими позиции модулей, настройки, данные о версии и разработчике, языковые файлы (см. рисунок 7).



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <extension version="3.0" type="template" client="site" method="upgrade">
3   <name>universum</name>
4   <description>TPL_UNIVERSUM_DESCRIPTION</description>
5   <version>1.1.0</version>
6   <creationDate>01.05.2020</creationDate>
7   <author>UPLITE</author>
8   <authorEmail>up-lite@ya.ru</authorEmail>
9   <authorUrl>https://up-lite.ru</authorUrl>
10  <copyright>Copyright (C) 2020. All rights reserved.</copyright>
11  <license>Лицензия</license>
12
```

Рисунок 7 – Описание файла «templateDetails.xml»

1. Первая строчка обязательная, содержит версию xml и кодировку:
`<?xml version="1.0" encoding="utf-8"?>`
2. Потом начинаем описывать наше приложение открывая тег extension:
`<extension version="3.0" type="template" client="site" method="upgrade">`
Указываем клиент установки и метод установки, версию Joomla, тип расширения.
3. Рассмотрим следующие строчки.
Имя шаблона:
`<name>MarketPlace</name>`
Описание шаблона:
`<description>Описание нашего шаблона</description>`
Ниже будем рассматривать, как это используется на практике. В данном виде содержится языковая переменная, как например:
Версия приложения:
`<version>2.0.1</version>`
Дата создания:
`<creationDate>01.02.2021</creationDate>`

Автор (разработчик):

`<author> Толыбекова Арайлы</author>`

Наш электронный почтовый ящик :)

`<authorEmail>marketplace@test.ru</authorEmail>`

Думая в пояснении, не нуждается:

`<authorUrl>http://mp.nelson.kz</authorUrl>`

Копирайт:

`<copyright> Copyright © 2021 - Автор сайта: Толыбекова Арайлы Маликовна</copyright>`

Информация о лицензии:

`<license>Лицензия</license>`

4. Вся информацию в этом блоке описываете на свое усмотрение. Все материалы здесь описываются, основываясь на личных предпочтениях. Затем происходит описание файлового наполнения, это значит, что в теге `files` требуется перечислить все файлы и папки, подлежащие установке. Важный момент: часто в процессе загрузки можно обнаружить то, что пропал файл `index.html` в каждой из папок, поэтому не стоит забывать проверять соответствие файлов и папок тому, что заявлено в описании и что есть на самом деле (см. рисунок 8).

```
13
14
15 <files>
16   <folder>css</folder>
17   <folder>fonts</folder>
18   <folder>html</folder>
19   <folder>images</folder>
20   <folder>js</folder>
21   <folder>404</folder>
22   <folder>language</folder>
23
24   <filename>index.html</filename>
25   <filename>index.php</filename>
26   <filename>error.php</filename>
27   <filename>templateDetails.xml</filename>
28   <filename>template_thumbnail.png</filename>
29   <filename>template_preview.png</filename>
30
31 </files>
```

Рисунок 8 – Тег `files` внутри файла `<templateDetails.xml>`

5. Языковые файлы. При разработке шаблона, рассчитанного на русскоговорящих посетителей – в создании данного файла нет необходимости, добавление этого раздела пропускается, а в тег `description` вписывается русское описание. Если приложение рассчитано на пользователей, не знающих русский язык, то необходимо добавить английский вариант по аналогии с нижеприведенным примером (см. рисунок 9).

```

32 <languages folder="language">
33 <language tag="en-GB">en-GB/en-GB.tpl_universum.ini</language>
34 <language tag="en-GB">en-GB/en-GB.tpl_universum.sys.ini</language>
35 <language tag="ru-RU">ru-RU/ru-RU.tpl_universum.ini</language>
36 <language tag="ru-RU">ru-RU/ru-RU.tpl_universum.sys.ini</language>
37 </languages>
38

```

Рисунок 9 – Тег language внутри файла «templateDetails.xml»

После установки данные файлы будут размещены в глобальной папке с языками, в зависимости от размещения в панели администратора, Joomla будет их автоматически подхватывать.[26]

Важно проследить, чтобы теги и имена файлов были указаны корректно. Действия аналогичны для всех локализаций, рассмотрим пример с русской локализацией.

– «ru-RU.tpl_marketplace.ini» – для фронтальной части шаблона, которая будет показана на сайте.

– «ru-RU.tpl_marketplace.sys.ini» – для административной части. Здесь стоит описать позиции модулей, переменные, описать настройки.

Выше мы добавили:

```
<description>TPL_UNIVERSUM_DESCRIPTION</description>
```

TPL_UNIVERSUM_DESCRIPTION – это и есть переменная, которую мы запишем в языковой файл, и Joomla 3.0 будет «подтягивать» значение из него.

Открываем файл с названием ru-RU.tpl_marketplace.ini и записываем значение переменной: TPL_UNIVERSUM_DESCRIPTION="<div style='background: #EDF4FA; border-radius: 5px; padding: 15px; margin: 0 0 15px; '><p>Спасибо за установку шаблона</p></div>"

Здесь вбивается любое значение, важный момент – нельзя переходить на новую строчку, если это сделать, то ничего не заработает. Чтобы сообщения были красиво оформлены, можно прибегнуть к помощи html.

Важный момент: описание всех языковых переменных начинается с новой строки.

Итоговый вариант <description> виден при загрузке шаблона:

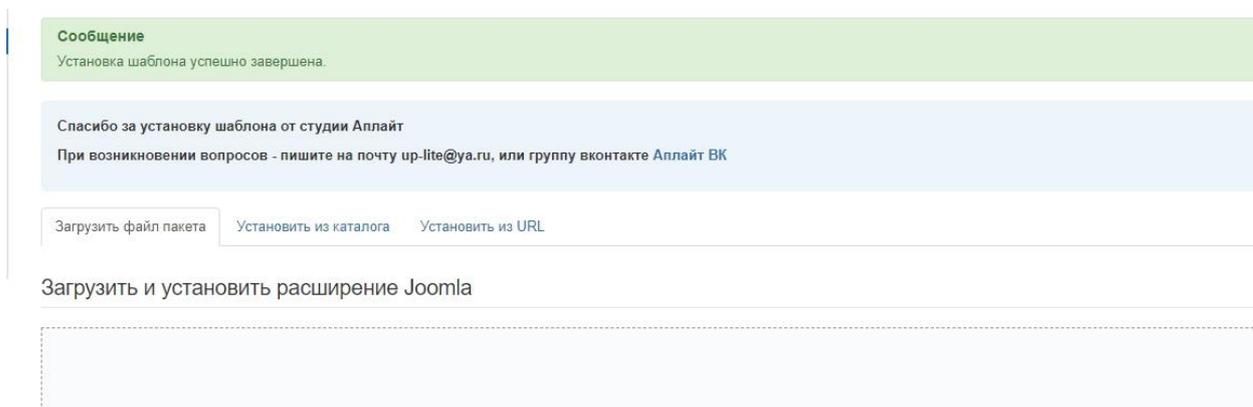


Рисунок 10 – Сообщение после установки шаблона

Чтобы эта информация отображалась в настройках, эта переменная должна также находиться в файле «ru-RU.tpl_marketplace.sys.ini»:

6. Описание позиций шаблона (что и в каком месте будет располагаться), с использованием раздела <positions>:

```
38
39     <positions>
40         <position>breadcrumbs</position>
41         <position>cart</position>
42         <position>side-menu</position>
43         <position>banner-block</position>
44         <position>main-menu</position>
45         <position>slider</position>
46         <position>bottom-1</position>
47         <position>bottom-2</position>
48         <position>bottom-3</position>
49         <position>mobile-menu</position>
50         <position>bottom-menu-column-2</position>
51         <position>bottom-menu-column-3</position>
52     </positions>
53
```

Рисунок 11 – Тег positions внутри файла «templateDetails.xml»

При наличии html-верстки, стоит сразу определиться с тем, какие необходимы блоки и позиции, где и какие моменты будут дублироваться. И уже отталкиваясь от этих данных, стоит определить позиции модулей. Если эти позиции будут прописаны только в шаблоне, но не будут отмечены в данном файле, то это скажется на отображении перечня позиций при публикации модуля, это все можно прописать вручную, и модуль будет опубликован (см. рисунок 11).

```
54     <config>
55         <fields name="params">
56             <fieldset name="basic">
57                 <field
58                     name="sitetitle"
59                     type="text"
60                     label="JGLOBAL_TITLE"
61                     description="JFIELD_ALT_PAGE_TITLE_LABEL"
62                     default=""
63                     filter="string"
64                 />
65
66                 <field
67                     name="sitedescription"
68                     type="text"
69                     label="JGLOBAL_DESCRIPTION"
70                     description="JGLOBAL_SUBHEADING_DESC"
71                     default=""
72                     filter="string"
73                 />
74                 <field
75                     name="tplbdcolor"
76                     type="color"
77                     label="Цвет кнопок"
78                     description="Цвет кнопок"
79                     class=""
80                     default="#ffab0a"
81                 />
82                 <field
83                     name="tplbgcolor"

```

Рисунок 12 – Параметры шаблона, внешний вид настроек

7. Параметры шаблона. Здесь выбираются настройки, подлежащие изменению в административной панели Joomla. Если это сделать, то это облегчит использование посетителями и администраторами. У них отпадет необходимость в открытии кодовых файлов, данные можно изменить в несколько нажатий. Предлагаются различные настройки шаблонов. Здесь будут затронуты главные, с их помощью можно добавить нужные материалы и поработать с цветовой гаммой (см. рисунок 12). После чего аналогичным образом можно поработать с более гибкими настройками шаблона, также можно сделать любое количество вкладок, чтобы можно было сгруппировать настройки [27].

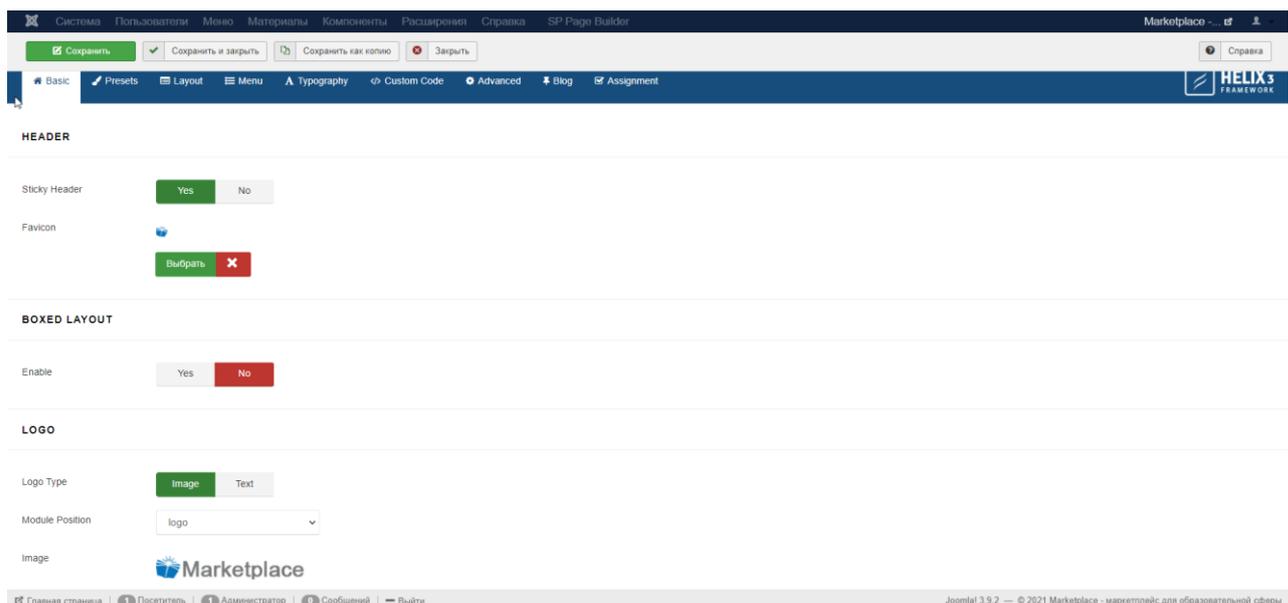


Рисунок 13 – Главная страница шаблона

С помощью тега `<fieldset name="basic">` создаются вкладки, в `<fieldset>` находятся поля с настройками. Разберемся, что включают в себя поля.

Название поля: `name="sitetitle"`.

Тип поля: `type="text"`

Метка: `label="TPL_BODY_FONT_FAMILY_LABEL"`

Описание: `description="TPL_BODY_FONT_FAMILY_DESC"` Значение по умолчанию (может отличаться в зависимости от типа поля):

`default=""`

Все разновидности полей представлены на официальной странице Joomla. Чтобы языковые переменные, упомянутые ранее, были отображены в админке, их следует продублировать в файл `«ru-RU.tpl_marketplace.sys.ini»`:

`TPL_BODY_FONT_FAMILY_LABEL`

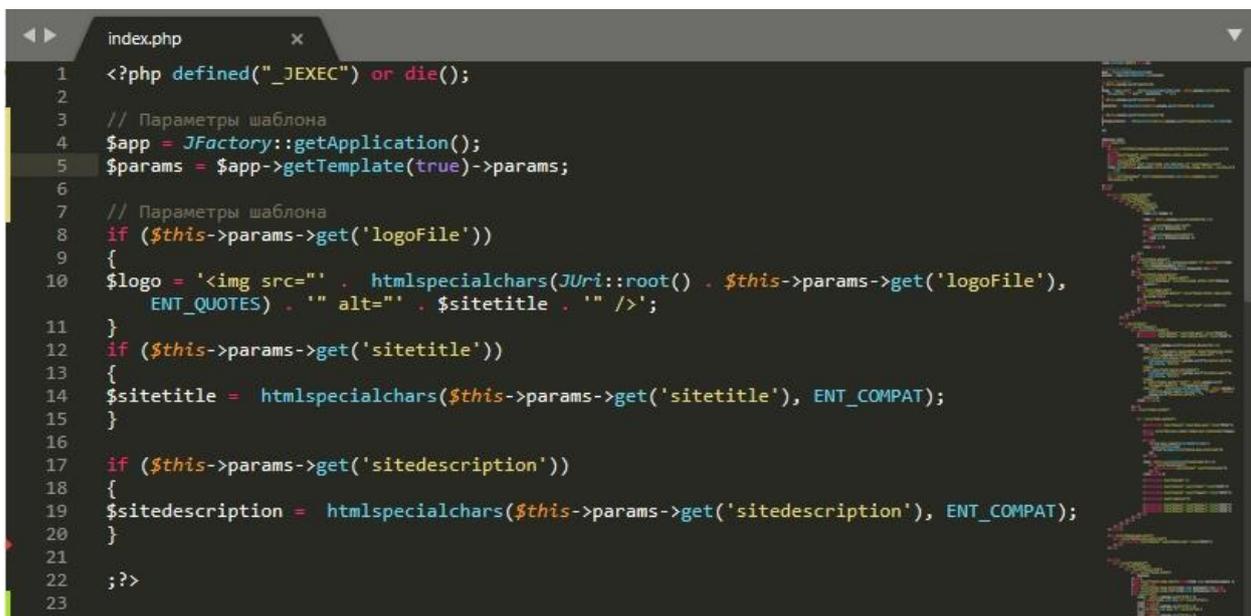
`TPL_BODY_FONT_FAMILY_DESC`

Когда шаблон будет использован только для русскоязычных пользователей, то значения полей сразу прописываются на нужном языке.

Внутри файла `«index.php»` расположится верстка создающегося маркетплейса, в котором также будут отображены выводы настроек и

необходимых параметров Joomla для правильного функционирования сайта (см. рисунок 14).

Первая обязательная строка php файла выглядит как следующее выражение:



```
1 <?php defined("_JEXEC") or die();
2
3 // Параметры шаблона
4 $app = JFactory::getApplication();
5 $params = $app->getTemplate(true)->params;
6
7 // Параметры шаблона
8 if ($this->params->get('logoFile'))
9 {
10 $logo = '';
12 }
13 if ($this->params->get('sitetitle'))
14 {
15 $sitetitle = htmlspecialchars($this->params->get('sitetitle'), ENT_COMPAT);
16 }
17 if ($this->params->get('sitedescription'))
18 {
19 $sitedescription = htmlspecialchars($this->params->get('sitedescription'), ENT_COMPAT);
20 }
21
22 ;?>
```

Рисунок 14 – Файл «index.php» – основной файл

Продолжаем дальше наполнять шаблон, после этого идут типовые разделы html. Код, которых прописан ниже:

```
<!doctype html>
<html lang="ru">
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<meta name="viewport" content="width=device-width, initial-scale=1">
<jdoc:include type="head"/>
<meta charset="UTF-8">
<link rel="shortcut icon" href="<?php echo $favicon; ?>" type="image/x-icon">
<!--style-->
<link rel="stylesheet" href="/templates/<?php echo $this->template; ?>/css/tpl-
style.css" />
<script src="/templates/<?php echo $this->template; ?>/js/main.js"
type="text/javascript"></script>
</head>
```

Здесь:

`<jdoc:include type="head"/>` – используется для загрузки компонентов и скриптов модулей. Также для загрузки скриптов на Joomla.

`<link rel="stylesheet" href="/templates/<?php echo $this->template; ?>/css/tpl-style.css" />` – для загрузки файлов стилей нашего шаблона.

`<script src="/templates/<?php echo $this->template; ?>/js/main.js" type="text/javascript"></script>` – загружаем скрипты шаблона, для повышения скорости загрузки на сайте, стоит загружать в футере.

Затем в тег `<body>` необходимо вписать верстку, а в места, предназначенные для материалов – определенные переменные, они помогут раскидывать содержимое из админки по разделам. Они более подробно расписаны ранее в файле `xml`.

`<body>`

`<?php echo $logo; ?>` – выводит логотип, заданный в параметрах шаблона.

`<?php echo $sitetitle; ?>` – выводит `<title>`, заданный в параметрах шаблона.

Таким образом выводим все параметры, заданные в настройках.

`<jdoc:include type="modules" name="slider" />` – выводит модуль в позицию `slider`.

`<jdoc:include type="message" />` – для вывода системных сообщений.

`<jdoc:include type="component" />` – для вывода компонентов.

`</body>`

Оставшиеся папки и файлы содержат верстку, она персональная для всех.

Стоит также разобраться с файлом `«error.php»` здесь отображается то, что собой представляет страница 404 (см. рисунок 15), здесь она имеет такой вид (он изменяется в настройках шаблона):

```
11
12 if (!isset($this->error))
13 {
14     $this->error = JError::raiseWarning(404, JText::_('JERROR_ALERTNOAUTHOR'));
15     $this->debug = false;
16 }
17
18 // Get language and direction
19 $doc = JFactory::getDocument();
20 $app = JFactory::getApplication();
21 $this->language = $doc->language;
22 $this->direction = $doc->direction;
23
24 $params = $app->getTemplate(true)->params;
25
26 ?>
27 <!DOCTYPE html>
28 <html lang="ru">
29 <head>
30     <meta charset="utf-8"/>
31     <title>404 - Страница не найдена</title>
32     <meta name="author" content=""/>
33     <meta name="keywords" content=""/>
34     <meta name="description" content=""/>
35
36     <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"/>
37     <link rel="stylesheet" href="/templates/<?php echo $this->template; ?>/css/tp1-style.css" />
38
39 </head>
40 <?php if ($params->get('tplbdcolor')) : ?>
41     <style>
42         :root {
43             --main-color: <?php echo $params->get('tplbdcolor'); ?>;
44             --main-color-bg: <?php echo $params->get('tplbgcolor'); ?>;
45         }
46     </style>
47 <?php endif; ?>
48 <body>
49     <div class="container">
50         <div class="info">
51             <h2>Что то пошло не так</h2>
52             <p>Страница, которую вы ищите удалена или не существует.</p>
```

Рисунок 15 – Файл «error.php» – ошибка 404

Теперь хотела рассказать об установке шаблона. Для начала шаблон должен быть архивирован. После стоит перейти на административную панель Joomla и поместить созданный шаблон (см. рисунок 16):

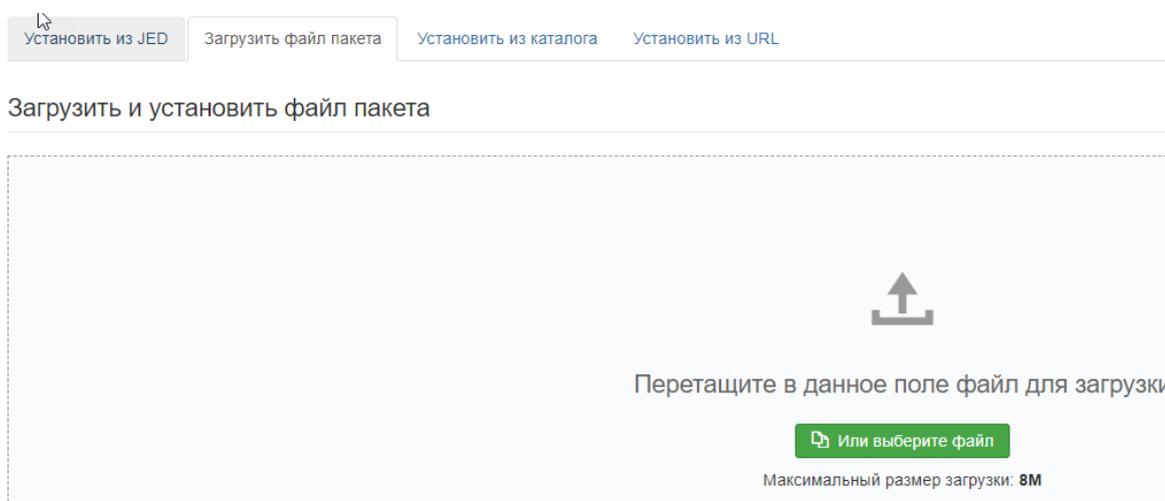


Рисунок 16 – Загрузка шаблона

Выбирать загруженный шаблон по умолчанию и перейти на сайт, чтобы посмотреть все ли правильно функционирует (см. рисунок 17).

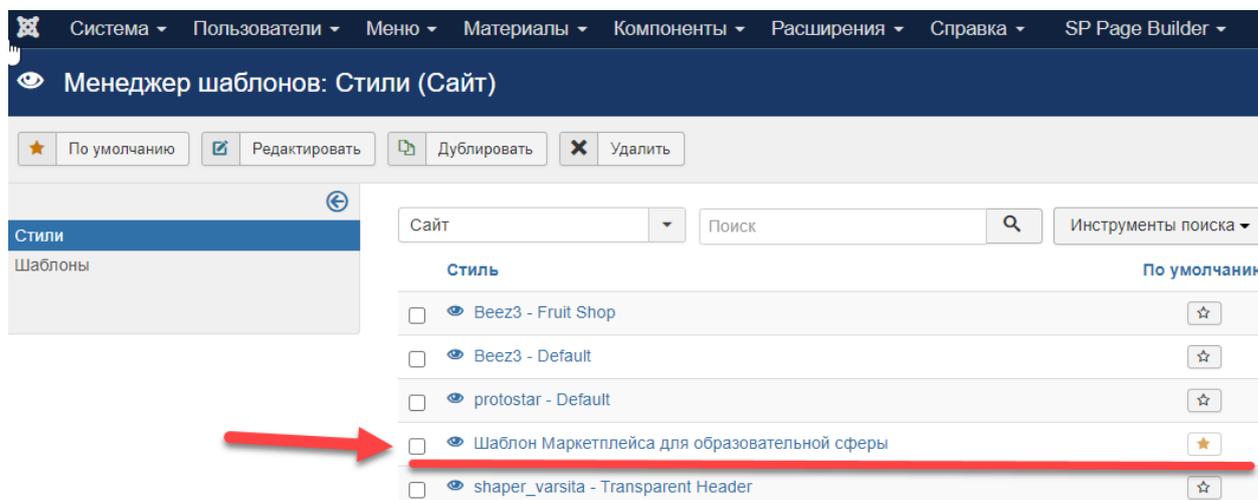


Рисунок 17 – Шаблон маркетплейса

2.2 Разработка базы данных для маркетплейса

Перед тем как приступить к Joomla, стоит разобраться с понятием база данных, так как не все начинающие веб-разработчики знают, что это такое.

Под базой данных принято понимать представленное в определенном виде объединение независимых материалов (статьи, настройки, информация о

закрепленных файлах, данных посетителей, без которых невозможно авторизоваться). При этом имеющаяся информация группируется так, чтобы при помощи определенного интерфейса доступ к ней был быстр и удобен. Поэтому ПО, применимое при взаимодействии с базами данных, называют Системой управления базами данных (СУБД). Достаточно популярный MySQL является такой системой. Помимо нее существует множество разных СУБД, каждая из них отличается по способу хранения материалов.[28]

Установить базу данных Joomla следует до того, как произойдет ее загрузка, сделать это можно с помощью любого подходящего варианта. Например, воспользовавшись phpMyAdmin, он расположен на каждом хостинге и имеется в сборках ПО: Denwer, OpenServer, Xampp и т.д.

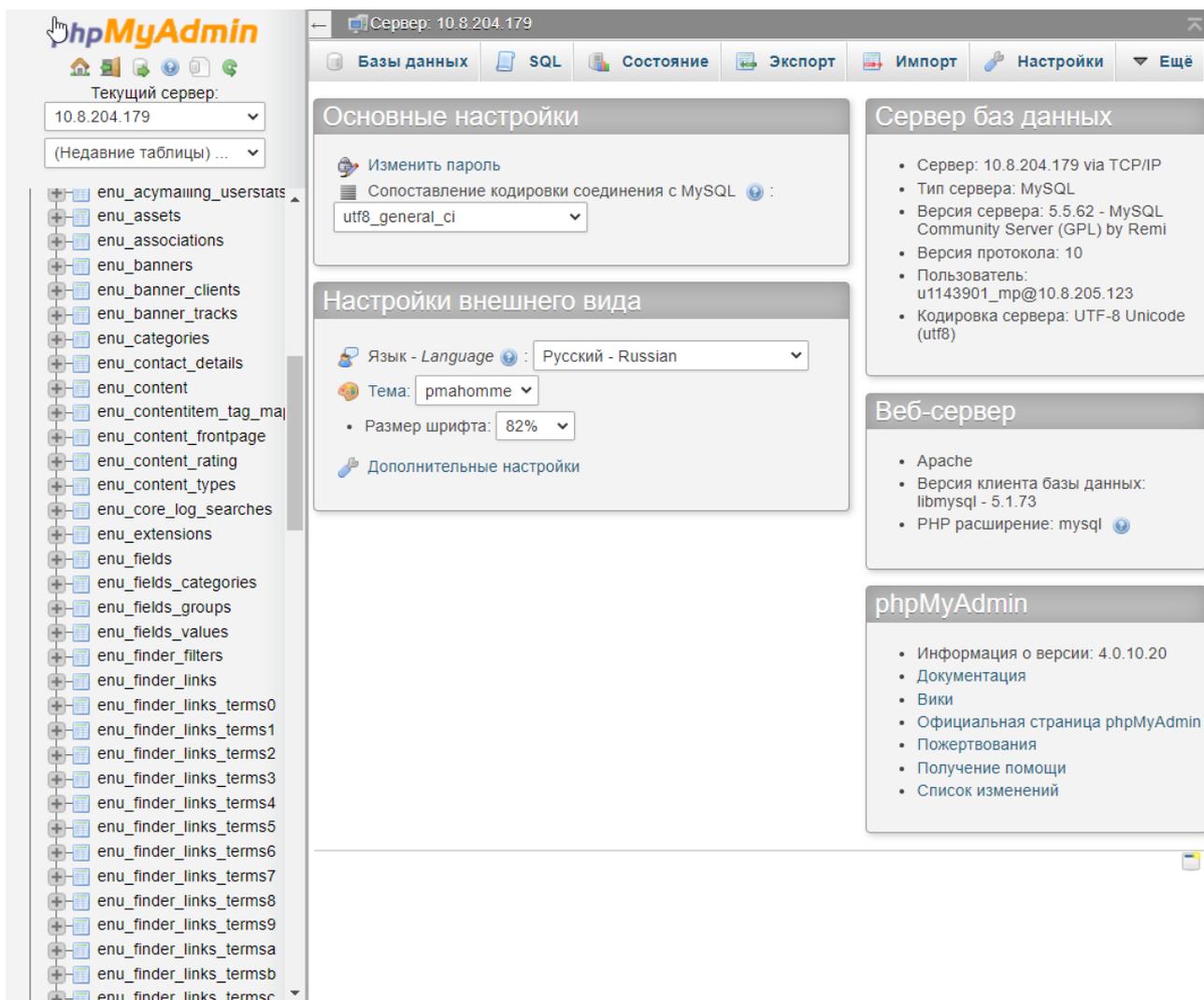


Рисунок 18 – Основная страница phpMyAdmin

На основной странице скрипта phpMyAdmin имеется вкладка «Базы данных», на ней и происходит создание новых баз данных. То есть стоит написать наименование и нажать на кнопку создания. Затем устанавливается CMS (см. рисунок 19).

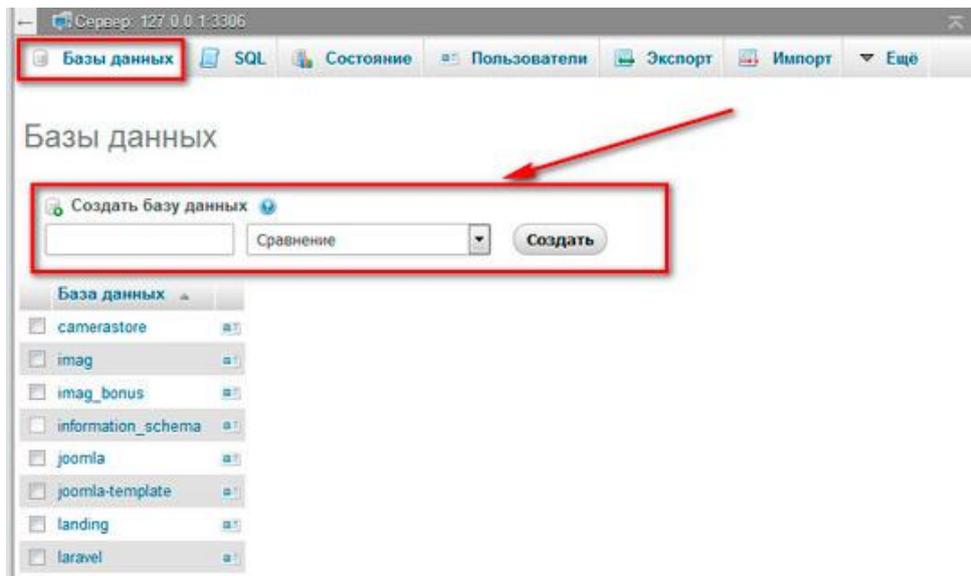


Рисунок 19 – Создание базы данных для маркетплейса

Таблицы базы данных Joomla появятся сразу при загрузке, в них будет расположена некоторая базовая информация, без которой невозможно функционирование CMS.

Теперь не возникнет проблем с тем, чтобы перейти в базу данных на Joomla. Стоит загрузить phpMyAdmin. Потом надо нажать на название нужной базы, после чего откроется страница, где можно просмотреть ее наполнение. Остановимся подробнее на наполнении базы данных Joomla (см. рисунок 20).

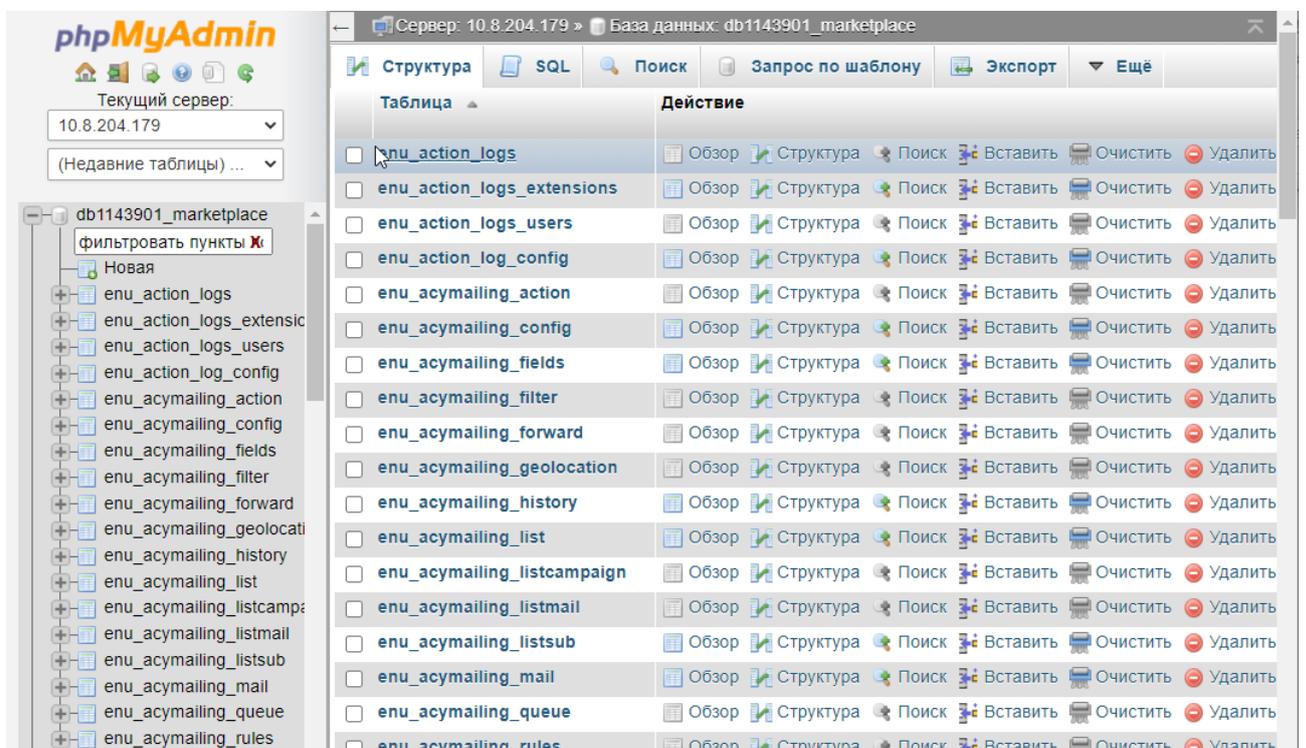


Рисунок 20. Структура базы данных маркетплейса

Как видно, здесь расположено огромное число разных таблиц, но это только в обычной версии, при установке различных дополнений, каждым из них добавится еще несколько таблиц, без которых невозможно дальнейшее функционирование.

Префикса не вызывает сложностей – необходимо придумать уникальное название для всех таблиц. Ведь в большинстве своем названия таблиц, без учета префикса – классические, с ними знакомы все, кто хотя бы раз имел дело с Joomla. Поэтому любой мошенник может отыскать кукую угодно таблицу по названию. Однако, если во время установки сгенерировать случайный префикс и добавить его в название, то название таблицы можно считать неизвестным параметром. Также префиксы помогают избавиться от повторения названий, когда в одной базе находятся табличная информация для нескольких копий [29].

Кратко остановимся на ключевых таблицах из Joomla:

- префикс *_banners* – данные о загруженных баннерах (часть *com_banners*).
- префикс *_categories* – информация о загруженных категориях (часть *com_content*).
- префикс *_content* – информация или материалы с персональной страницы (часть *com_content*).
- префикс *_extensions* – данные о загруженных дополнениях
- префикс *_languages* – загруженные языки.
- префикс *_menu_types* – разработанные меню (часть *com_menu*)
- префикс *_menu* – разработанные категории меню (часть *com_menu*)
- префикс *_users* – авторизированные в системе посетители (часть *com_users*)

Это лишь малая часть той табличной информации, которая появляются после загрузки Joomla.

Во время создания портала и при добавлении разнообразных дополнений в базе может образоваться множество различной информации, которую система не использует. В таком случае стоит провести чистку. Стоит воспользоваться помощью инструмента phpMyAdmin, стоит выбрать необходимую таблицу и убрать из нее неиспользуемые записи.

Бесполезные материалы в виде таблиц баз данных могут копиться при применении огромного числа разнообразных дополнений, которые со временем удаляют из системы. После удаления, некоторая информация может оставаться. Поэтому, когда решено очистить таблицы, стоит изучить таблицы, находящиеся в базе данных. Практически всегда название скачанного расширения аналогично названию таблицы. Поэтому если расширение убрано, а таблицы с таким названием находятся в базе, то их также стоит убрать.

В табличке префикс *_extensions* можно увидеть данные скачанных дополнений, поэтому если какие-то из них удалены, но они по-прежнему отображаются в таблице, значит они также подлежат удалению.

Чтобы удалить таблицу из базы данных. Надо перейти в панель управления вашего хостинга и войти в инструмент phpMyAdmin. Потом надо выбрать нашу

базу данных. Для того чтобы удалить таблицы базы данных, нажмите «Выделить все», затем в фильтре выберите «Удалить» (см. рисунок 21).

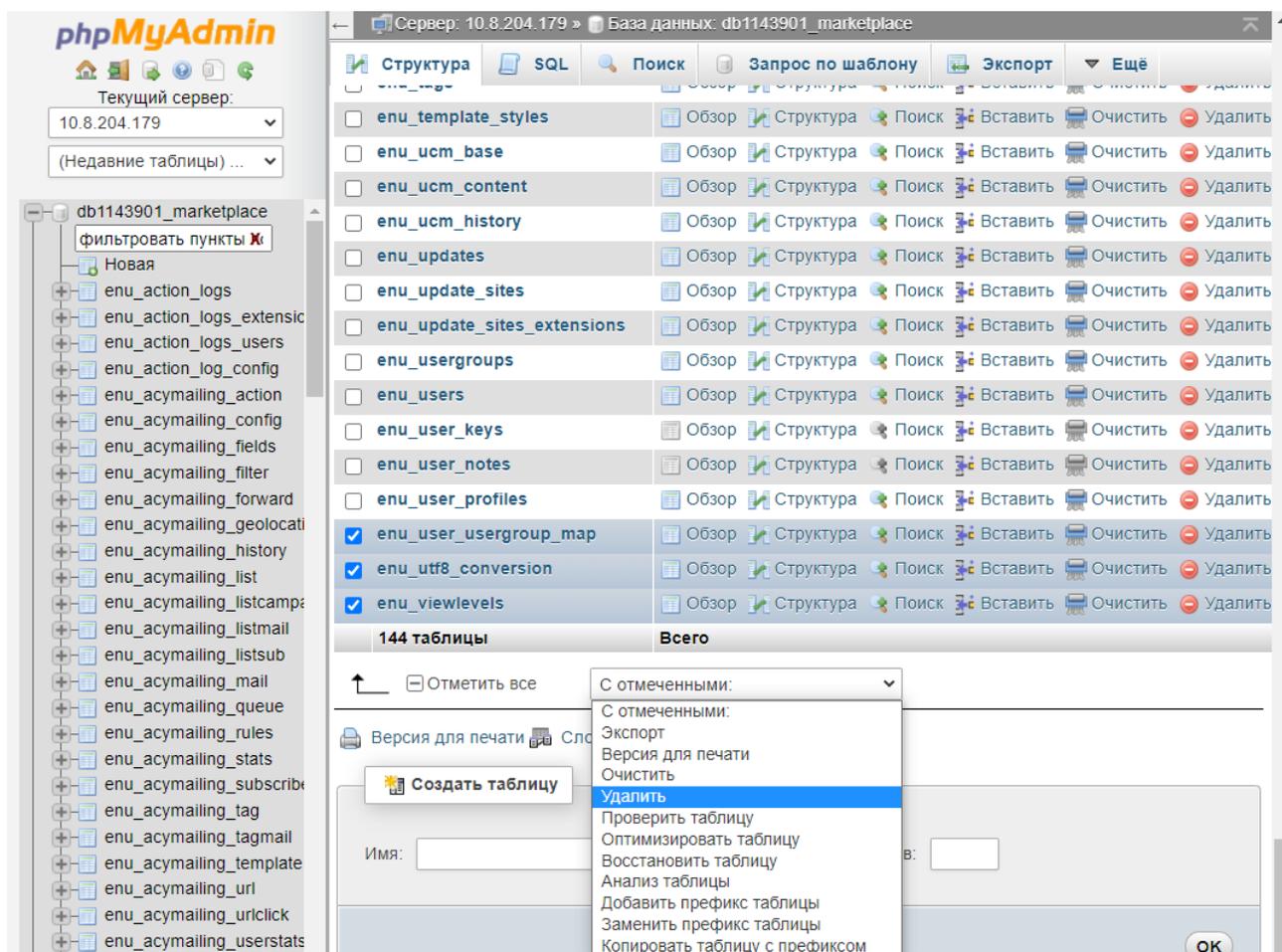


Рисунок 21 – Удаление таблицы из базы данных

Этого вполне достаточно для того, чтобы избавить базу данных от лишних материалов. Для более глубокой очистки потребуется хорошо разбираться в CMS Joomla.

2.3 Применение CMS системы в разработке маркетплейса для образовательной сферы

Joomla — одна из лучших платформ CMS, которую вы можете использовать для создания сайта. Как система управления контентом, она предлагает всё необходимое для управления публикациями и создания новых страниц с лёгкостью. На сегодняшний день он считается одной из лучших альтернатив WordPress. И установка Joomla не займёт много времени.

Теперь я хочу рассказать об установке CMS Joomla версии 3.0. Сначала мы рассмотрим ручную установку Joomla. Для этого нам нужно приложить руки и

сделать немного больше работы. В целом, нам нужно выполнить 5 простых шагов:

1. Создаем базу данных MySQL и пользователя.

Для начала перейдите на панель управления хостингом и откройте раздел «Базы данных -> Базы данных MySQL». Укажите имя базы данных, пользователя и пароль, как показано в примере ниже:

Subscriber
Subscription

Home Hosted Domains Пользователи службы Mail Websites Личный кабинет

Odin Panel > More Services > Базы данных >

Add New Database

New Database Info

Тип базы данных MySQL

Имя базы данных db1143901_

Имя базы данных can not be longer than 64 symbols, where first 10 sym entered manually.

New Database's user Info

Имя пользователя базы данных u1143901_ mp

Имя пользователя базы данных can not be longer than 16 symbols, wh can be entered manually.

Password * Средний [Создать](#) [Пок:](#)

*Required fields [Next >>](#) [Cancel](#)

Рисунок 22 – Создание базу данных

Обязательно запишите где-нибудь подробности, так как позже вам понадобится их использовать.

2. Загрузить последнюю версию Joomla.

Затем бесплатно скачайте официальный установочный пакет Joomla с их сайта. Чтобы получить последнюю версию, нажмите «Скачать Joomla!» и ваш запрос будет завершён в ближайшее время.

3. Загрузить архив Joomla и распаковать его на нашем хостинге.

Теперь хочу подробнее рассказать об использовании FTP-клиента. На этом этапе вам нужно загрузить сжатый файл в корневой каталог вашего домена и распаковать его.[30]

Кроме того, вы можете сделать это с помощью FTP-клиента. Чтобы использовать его, вам понадобится ваш IP-адрес хоста, имя пользователя FTP, пароль и порт. Вы можете найти их, перейдя в «Файлы->Учётные записи FTP» на панели управления (см. рисунок 23).

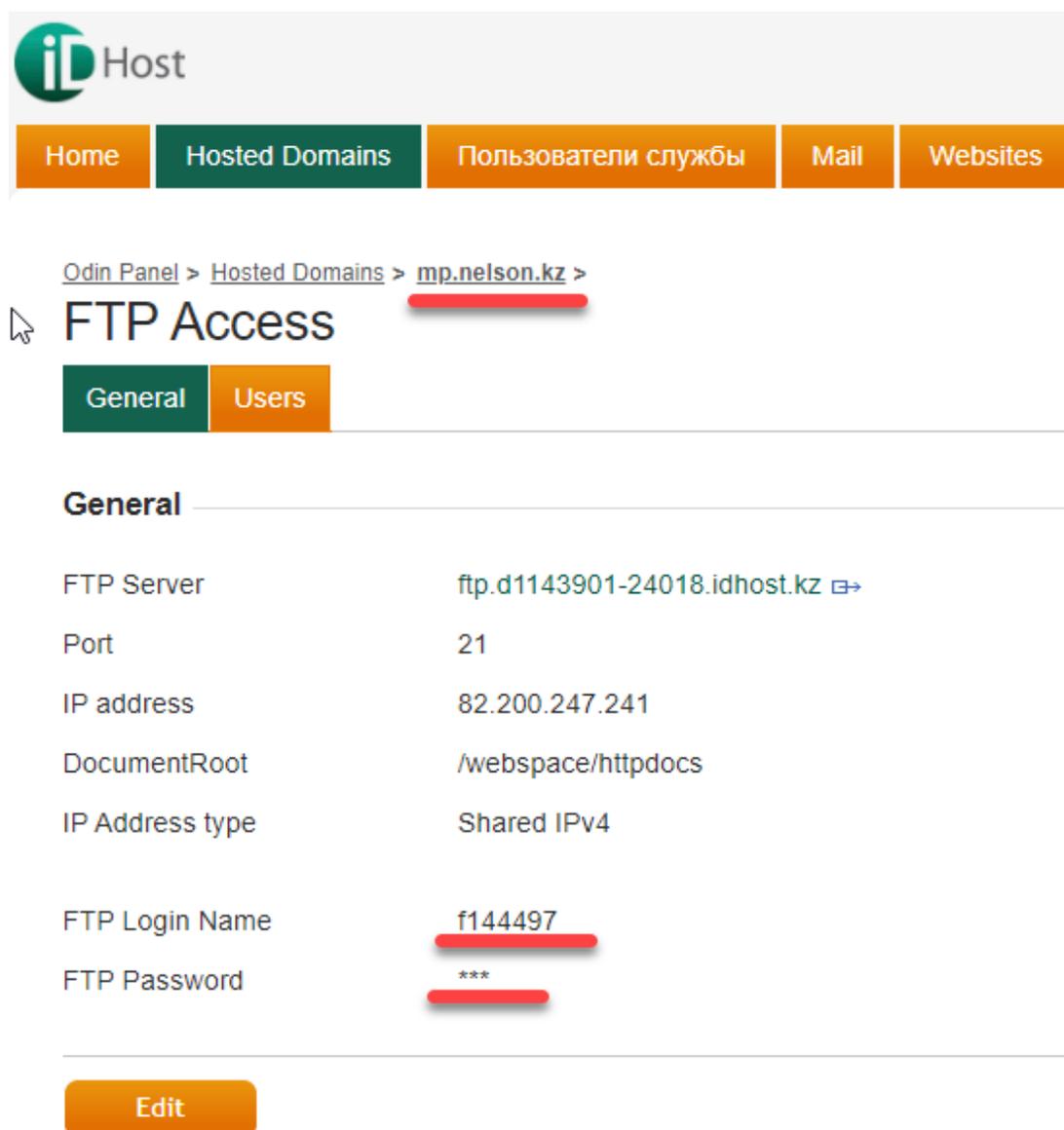


Рисунок 23 – FTP доступ

Настройте FTP-клиент FileZilla и нажмите кнопку «Quickconnect». В идеале вы должны сначала извлечь содержимое на вашем компьютере, так как FileZilla не имеет возможности сделать это. Затем перетащите файлы из локального хранилища в каталог public_html, как показано ниже:

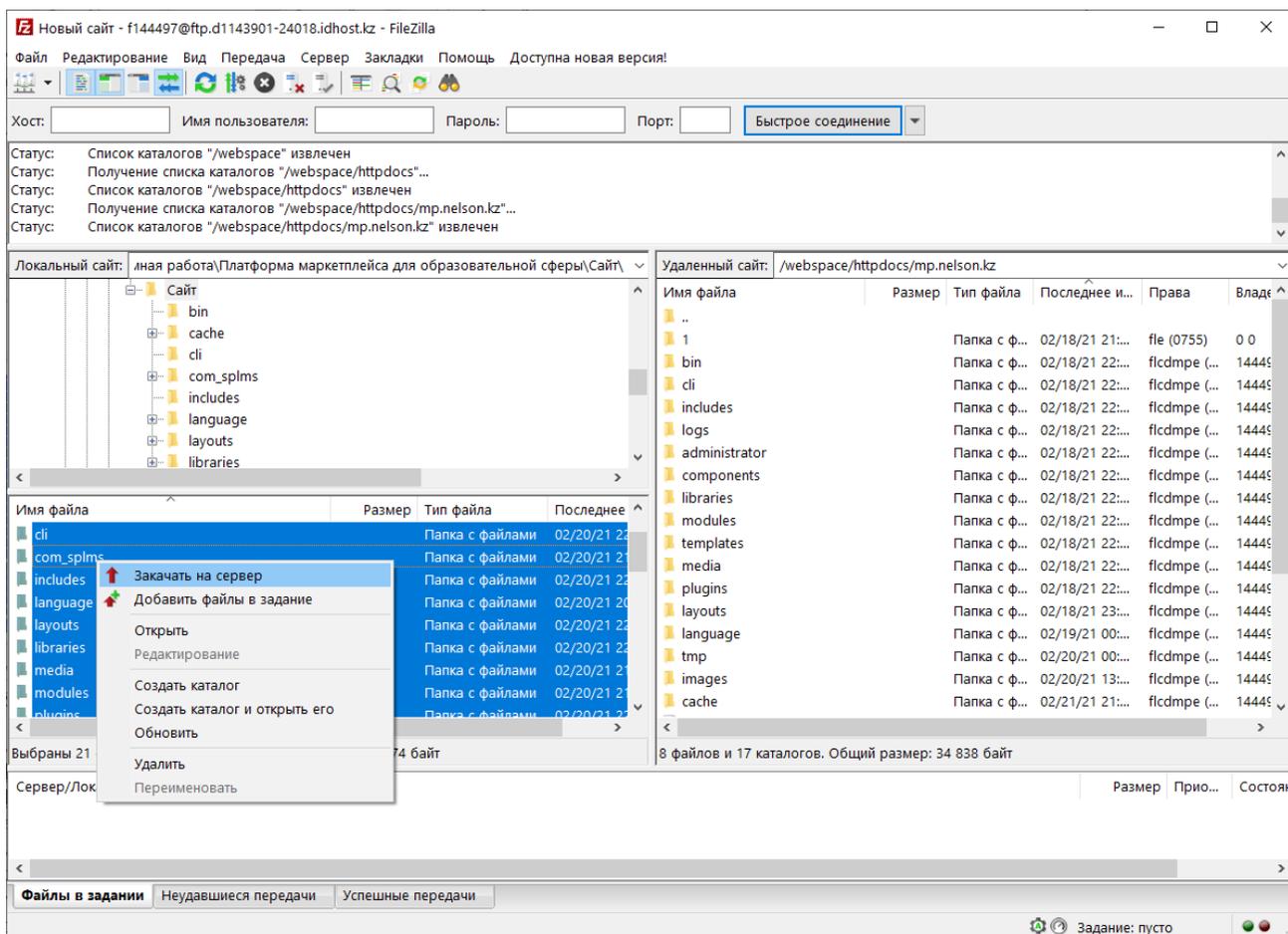


Рисунок 24 – Загрузка файлов через FTP клиент – FileZilla

Поскольку существует много отдельных файлов, это может занять от 10 до 15 минут (см. рисунок 24).

4. Убедитесь, что файлы Joomla находятся в корневом каталоге.

Чтобы Joomla работала через наш сайт **mp.nelson.kz** все файлы должны быть размещены в `public_html`. Если вы не набрали точку при извлечении или загрузке всего каталога, появится дополнительная папка. Вам нужно получить к ней доступ и переместить всё в `public_html`.

Если вы использовали файловый менеджер IDHost, вам необходимо:

1. Создать дополнительный каталог.
2. Выбрать все файлы.
3. Щёлкнуть правой кнопкой мыши и выбрать функцию «Переместить».
4. Указать место назначения как `/public_html`.
5. Нажать «Переместить» ещё раз, чтобы продолжить.

Если вы использовали FTP-клиент FileZilla, вам следует:

1. Создать дополнительную папку.
2. Выбрать все файлы в ней.
3. Перетащить их на один каталог вверх (представлен папкой с двумя точками).
4. Дождаться окончания процесса.

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
administrator		Directory	01/08/2019 1...	flcdmpe (...	u574398...
bin		Directory	01/08/2019 1...	flcdmpe (...	u574398...
cache		Directory	01/08/2019 1...	flcdmpe (...	u574398...
cli		Directory	01/08/2019 1...	flcdmpe (...	u574398...
components		Directory	01/08/2019 1...	flcdmpe (...	u574398...
images		Directory	01/08/2019 1...	flcdmpe (...	u574398...
includes		Directory	01/08/2019 1...	flcdmpe (...	u574398...

Selected 7 files and 16 directories. Total size: 13,851,468 bytes

Рисунок 25 – FTP клиент – FilleZilla

5. Последний этап установка Joomla.

На этом этапе вы можете начать установку Joomla. Начните с посещения своего домена через любой браузер и убедитесь, что ваш хост MySQL, пользователь, база данных и значения паролей находятся под рукой.

Во-первых, вам нужно будет ввести имя вашего сайта Joomla, описание и настроить пользователя с правами администратора (см. рисунок 25).

Рисунок 26 – 1 этап при установке Joomla

Затем вам нужно будет заполнить детали базы данных MySQL, которые мы получили в шаге «Создаем базу данных MySQL и пользователя». В нашем случае имя хоста — <https://d1143901-24018.idhost.kz>, но это может варьироваться в зависимости от вашего провайдера.

1 Configuration 2 Database 3 Overview

Database Configuration ← Previous Next →

Database Type *
This is probably "MySQL".

Host Name *
This is usually "localhost" or a name provided by your host.

Username *
Either a username you created or a username provided by your host.

Password
For site security using a password for the database account is mandatory.

Database Name *
Some hosts allow only a certain DB name per site. Use table prefix in this case for distinct Joomla! sites.

Table Prefix *
Create a table prefix or use the randomly generated one. Ideally four or five characters long, it may only have alphanumeric characters and MUST end in an underscore. Make sure that the prefix chosen is not already used by other tables.

Old Database Process *
"Backup" or "Remove" any existing tables from former Joomla! installations with the same "Table Prefix".

Рисунок 27 – Подключение базы данных при установке Joomla

На последнем шаге вам нужно будет просмотреть свои данные и выбрать дополнительные модули (см. рисунок 27).

Это оно! Вы успешно установили Joomla CMS в свою учётную запись веб-хостинга. Чтобы начать создание своего сайта, перейдите в административную зону, расположенную по адресу *mp.nelson.kz/administrator*.

Для того чтобы зайти в панель управления движком, нужно перейти по адресу вида «адрес_сайта/administrator/» (например, у в нашем случае на локальном хостинге этот URL такой: «mp.nelson.kz/administrator/»), на странице, которая открылась и авторизоваться (ввести указанные во время установки движка логин и пароль), дальше нажать кнопку «Войти» (см. рисунок 28).



Рисунок 28 – Вход в админ панель

Далее в браузере отобразится на панели управления главная страница, где основная часть занята блоками с предупреждением о наличии важных сообщений. Для того чтобы его убрать, нужно прочесть их и для каждого нажать на кнопку «Скрыть это сообщение», о переносе сайта на хостинг, списком авторизаций и популярных статей, размещённых на сайте. В этой версии их список не пуст, так как при установке был выбран режим загрузки демо-контента [31].

В верхней части админ-панели показывается главное меню, которое от страницы к странице остаётся неизменным. Во время установки расширений оно пополняется новыми пунктами. В левой части находятся так называемые быстрые ссылки, которые дублируют пункты основного меню, но набор их меняется в зависимости от того, на какой странице админ панели сейчас находитесь. Боковое меню показывается исключительно для ускорения работы с панелью управления движком и удобства. В нижней части страниц можно увидеть дополнительную информацию, где расположена ссылка для перехода к просмотру сайта.

– *Панель управления.* Кнопка дает возможность перехода на главную страницу. Однако, доступен он не всегда, и когда он недоступен новички испытывают дискомфорт. Если пункт Панель управления недоступен, то вероятно, открыта страница редактирования, чтобы выйти из него нужно нажать кнопку «Закреть».

– *Общие настройки.* Заметьте, они разбиты на вкладки. На вкладке (Сайт) можно закрыть сайт на время выполнения технических работ, изменить заданное во время инсталляции Joomla имя ресурса, (при этом настраивается видимое пользователям сообщение и картинка), указать метаданные, а именно описание и ключевые слова для поисковиков, убрать показ версии Joomla, и настроить другие параметры SEO.

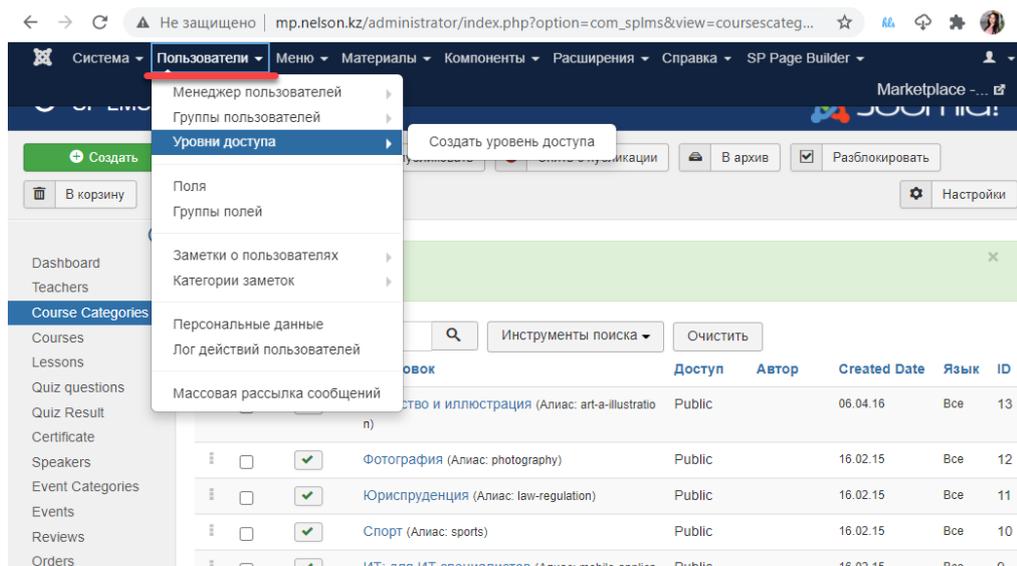


Рисунок 29 – С учётными записями можно работать с предназначенной вкладкой «Пользователи»

– *Менеджер пользователей.* Дает возможность управлять отдельно взятыми учётными записями — удалять, блокировать, активировать и пр. Существует функции пакетной обработки для случаев когда регистраций было слишком много. Подраздел «Создать пользователя» позволяет без регистрации на сайте стандартным способом, добавить учётную запись сразу из админ-панели.

– *Группы.* Менеджер пользователя дает индивидуальный подход, что обеспечивает работу с отдельно взятым пользователем, а в этом пункте работа идет с группами учётных записей, таких в Joomla существует около десятка. Категории напрямую связаны с привилегиями. Самыми большими правами обладает группа пользователей «Super Users», которые могут менять на сайте всё. Мелкие права у гостей (Guest), которые только просматривают сайт. Всегда можно создавать собственные группы, если существующих недостаточно [32].

– *Уровни доступа.* Дает возможность управлять привилегиями пользователей. Разрешать какие-то действия одним и запрещать их же другим позволяет разбиение на группы. Уровни доступа обычно используются на больших сайтах, в штате которых работает множество людей: есть контент-менеджеры, администраторы, копирайтеры (см. рисунок 29).

– *Меню.* О создании пунктов меню расскажем в следующем разделе, пока что остановимся знанием, что данный раздел содержит все инструменты управления кнопками на сайте.

– *Материалы.* Здесь можно управлять медиа-контентом, а также статьями, их категориями (видео, звук, а также картинки).

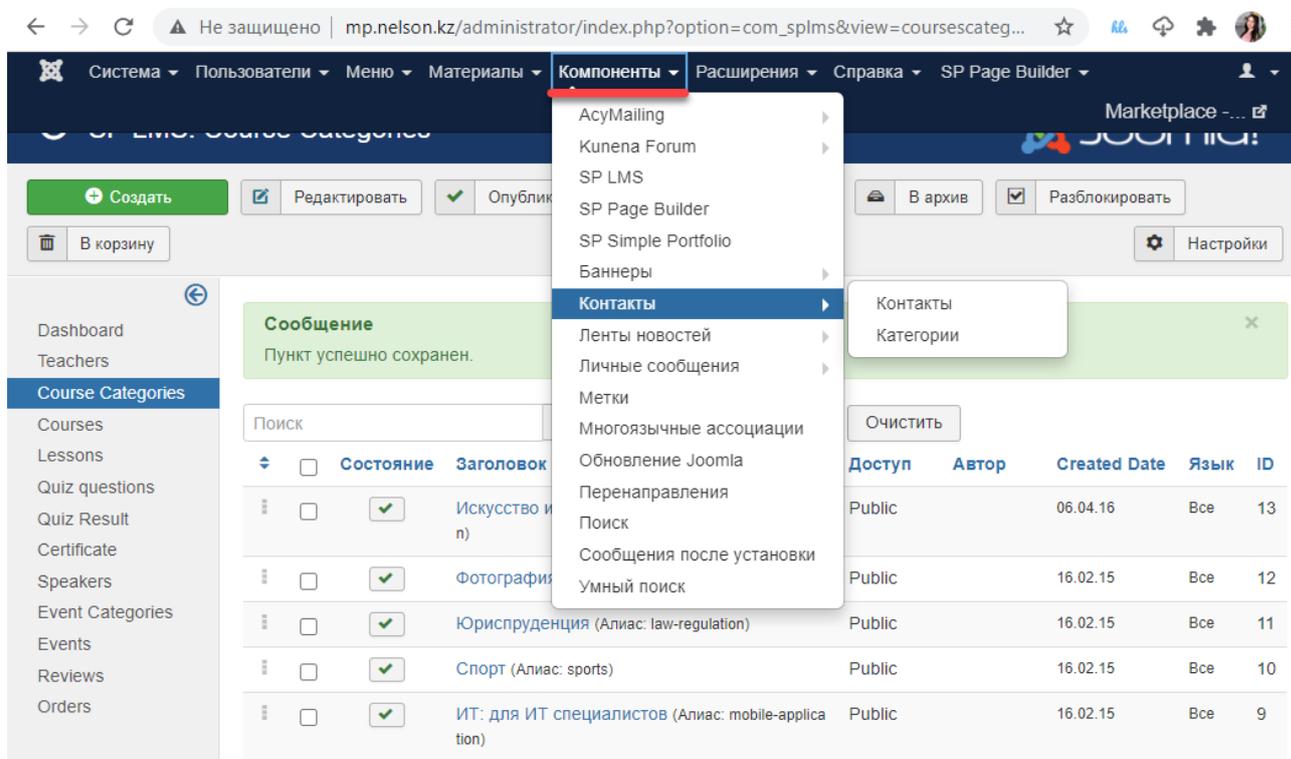


Рисунок 30 – Вкладка «Компоненты»

– *Компоненты*. Расширения в Joomla делятся на три типа: компоненты, модули и плагины. Подробнее расскажем позже, но, если вкратце, компонент является самым функциональным видом расширения движка, он имеет свои плагины и даже сам представляет собой небольшую CMS. Точно такие же расширения имеет подменю Компоненты. Вместе с Joomla устанавливаются расширения (см. рисунок 30).

– *Баннеры*. Добавление рекламных баннеров на страницы ресурса и получение по ним исчерпывающей статистики в дальнейшем: просмотры, клики — всё, как полагается.

– *Контакты*. Используются при создании страницы с формой обратной связи, а именно с контактными данными владельца сайта.

– *Ленты новостей*. С помощью этого компонента можно легко наполнить свой сайт чужим контентом, выводя новости из сторонних RSS-лент.

– *Обновление Joomla*. С помощью этой функции устанавливаются свежие версии движка. Этой функцией нужно пользоваться достаточно часто, потому что сразу после обновления релиза CMS будет доступна на главной странице.

– *Метки*. Дают возможность управлять тегами. Меток можно ставить сколько угодно, с помощью них пользователь будет легко ориентироваться на сайте и искать данные на нужную тему.

– *Перенаправления*. Переадресация с одного сайта на другой (например, 205-й текст).

– *Поиск*. Расскажет о анализе поисковых запросов.

– *Системные сообщения*. Встречает на главной странице панели управления, донося о непрочитанных сообщениях от системы.

– *Сообщения*. Используются при чтении и отправке личных сообщений.

– *Умный поиск*. Он помогает ввести запрос, дополняя незаконченное слово так же, как на Google или Яндекс. Использование компонента увеличивает нагрузку на сервер, так как без индексации страниц интеллектуальный поиск не работает.

– *Расширения*. Здесь собраны шаблоны, модули, плагины и языки — почти все, кроме компонентов дополнения (см. рисунок 31).

– *Менеджер расширений*. Дает возможность установить: шаблон, плагин или модуль, язык, компонент, а после установки управление тоже идет отсюда.

– *Менеджер модулей*. Дает возможность управлять видом расширений, они обычно даются в специально отведённых темах оформления, внутри материалов и также как виджеты в WordPress.

– *Менеджер плагинов*. С их помощью функционал Joomla расширяется, а здесь можно ими управлять.

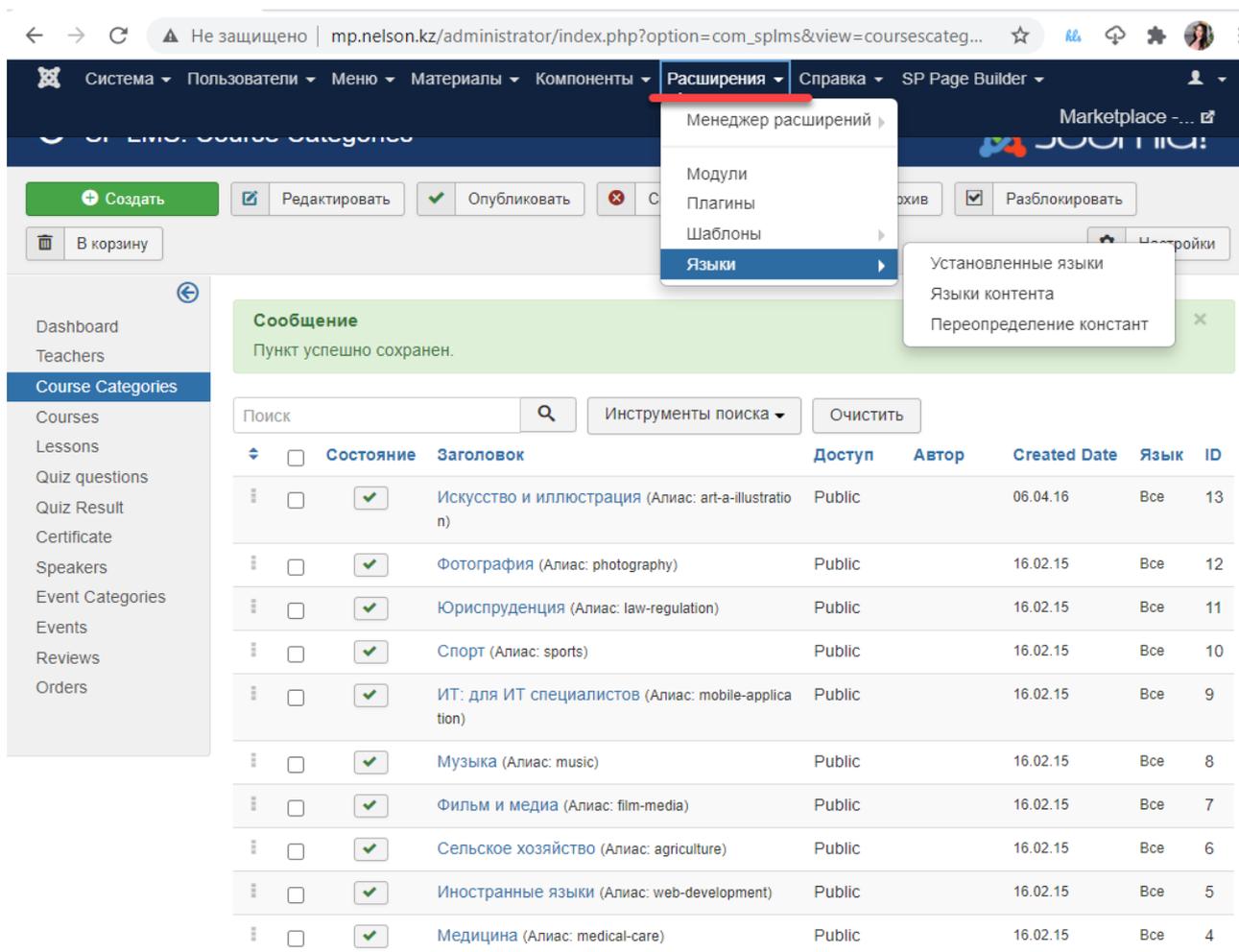


Рисунок 31 – Вкладка «Расширения»

– *Менеджер шаблонов.* Отсюда можно управлять темами оформления движка.

– *Менеджер языков.* С помощью этого менеджера можно управлять языками: видеть установленные пакеты, сделать настройку языков, загрузить новые при необходимости [33].

ЗАКЛЮЧЕНИЕ

В ходе работы над квалификационной дипломной работой был создан полностью функциональный маркетплейс для образовательной сферы, который целиком готов к использованию. Созданный мною маркетплейс уже запущен и доступен по адресу «mr.nelson.kz». Этот маркетплейс будет использоваться абитуриентами, школьниками и студентами. При его использовании у пользователей появится доступ к необходимым знаниям, они смогут получить ответы на интересующие вопросы, воспользовавшись форумом. При его загрузке в глобальную сеть, география его распространения достигнет мирового масштаба.

Наш маркетплейс – это оптимизированная интернет-платформа, на которой предлагаются услуги для образовательной сферы. Один и тот же курс можно купить у нескольких компаний или учителей, причем каждый курс может стоить по-разному.

Основной фактор, который определяет маркетплейс — огромное число вендоров в пределах одного сайта. Вендор (vendor, provider) — компания либо репититор, предлагающая услуги для образовательной сферы.

При создании маркетплейс был проведен анализ современных web-технологий, с помощью которых можно разрабатывать интерактивные маркетплейсы. Наилучшими вариантами для достижения этой задачи стали Macromedia Dreamweaver MX 2019 и Joomla 3.0.

При создании маркетплейса решили следующие задачи:

- сделали анализ актуальных систем создания маркетплейсов;
- поняли принцип и значение маркетплейсов в современных рыночных условиях, рассмотрели разновидности электронной услуги;
- определили понятие функции, интернет-услуги, разновидности, сущность;
- разобрались с моделями организации онлайн-бизнеса в Казахстане, которые решают экономическую специфику задач;
- проанализировали электронные продажи и их системы разработки;
- изучили общие понятия проектирования информационных систем типа «интернет-магазин»;
- выполнили проектирование платформы маркетплейса.
- разработали платформу маркетплейса для образовательной сферы;
- оценили, насколько эффективен разработанный программный продукт.

Созданный маркетплейс полностью отвечает каждому требованию, которое было выдвинуто при определении задач. При создании маркетплейса были применены готовые модули аутентификации, форума и осуществления поиска по сайту. Эти модули дорабатывались, учитывая специфику web-сайта, после чего удачно внедрены в его структуру.

В качестве дальнейшего улучшения маркетплейса, возможно создание модулей доступа к системам удаленного образования. Так же можно доработать

интерфейс сайта для того, чтобы в будущем повысить его информативность, привлекательность и сделать более удобным.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Старова Т.С., Могилев А.В. Типология образовательных веб-сайтов / М: Вопросы Интернет-образования, № 2 // – 2008 // – №3 // – С. 41-44.
- 2 Маркетплейсы — локомотив роста e-commerce // Электронная версия на сайте <https://www.cossa.ru/trends/224910/>.
- 3 Маркетплейсы — материал из Википедии // Электронная версия материала на сайте <https://ru.wikipedia.org/wiki/Маркетплейс>.
- 4 Marketplace: нишевые торговые площадки в интернете // Электронная версия на сайте <https://ru.wiki.rademade.com/what-is-marketplace>.
- 5 Записки маркетолога // Электронная версия на сайте http://www.marketch.ru/marketing_dictionary/marketing_terms_m/marketplace/.
- 6 Маркетплейс: что это такое, самые лучшие и популярные платформы электронной коммерции в Казахстане // Электронная версия на сайте <https://www.cleverence.ru/articles/elektronnaya-kommertsiya/marketpleys-chto-eto-takoe-samye-luchshie-i-populyarnye-platformy-elektronnoy-kommertsii-v-rossii/>.
- 7 Что такое маркетплейс? // Электронная версия на сайте <https://kz.horoshop.eu/blog/chto-takoe-marketpleys/>.
- 8 Как запустить маркетплейс. Проблема курицы и яйца // Электронная версия на сайте <https://gopractice.ru/build-grow-marketplace-chicken-and-egg-problem-part1/>.
- 9 Колиснеченко, Д.Н. Движок для вашего сайта. CMSJoomla // М.: ООО И.Д. Вильяме, 2013. – 256 с.
- 10 Гендина, Н.И. Использование формализованных методов при подготовке текста для веб-сайтов учреждений культуры / Н.И. Гендина, Н.И. Колкова, О.И. Алдохина // Научные и технические библиотеки. – 2008. – №3. – С. 29-35.
- 11 Гендина, Н.И. Лингвистические средства проектирования контента веб-сайтов / Н.И. Гендина // Научные и технические библиотеки. – 2008. – №3. – С. 5-14.
- 12 Иванов. А. Подготовка сайта для индексирования в поисковых системах // Электронная версия на сайте <http://www.ashmanov.com/pap/ivsprep.phtm>.
- 13 Инькова, Н. А. Создание Web-сайтов: Учебно-методическое пособие // Электронная версия на сайте <http://club-edu.tambov.ru/methodic/fio/p5.doc>.
- 14 Кречетников, К.Г. Рекомендации по проектированию мотивационной и содержательной составляющих образовательной среды вуза на основе информационных технологий // Электронная версия на сайте <http://www.auditorium.ru/aud>.
- 15 Мэтьюс, М. FrontPage 2000: полное руководство / М. Мэтьюс, Э.Полсен; пер. с англ. – К.: Издательская группа ВHV, 2000. – 752 с.
- 16 Селин, Г. Интернет-пособие по созданию Web-сайта, его публикации и способах раскрутки // Электронная версия на сайте <http://web-book.narod.ru/>.

17 Панкова, Е.В. Интернет-сайт среднего профессионального учебного заведения: эффективность поиска в Интернете / Панкова Е.В., Косинова С. А. // Сорников Я.А. Использование стилей при создании Web- страниц с помощью программы DreamWeaver. М: Вопросы Интернет-образования, № 39.

18 Чебан О.О. Интернет-технологии в школе. Создание административных web-страниц с помощью PHP и MySQL // М: Вопросы Интернет-образования, №34.

19 Чебан О.О. Интернет - технологии в школе. Сервер данных MySQL // М: Вопросы Интернет-образования, № 32-33. е библиотеки. // 2008. – №3. – С. 41-44.

20 Порублев Илья Николаевич, Ставровский Андрей Борисович Алгоритмы и программы. Решение олимпиадных задач. – М.: «Вильямс», 2007. – С. 480.

21 Дональд Кнут Искусство программирования, том 1. Основные алгоритмы // The Art of Computer Programming, vol.1. Fundamental Algorithms. – 3-е изд. // М.: «Вильямс», 2006. – С. 720.

22 Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн Алгоритмы: построение и анализ // INTRODUCTION TO ALGORITHMS. – 2-е изд. // М.: «Вильямс», 2006. – С. 1296.

23 Как создать сайт на Joomla!: пошаговое руководство для начинающих // Электронная версия на сайте <https://texterra.ru/blog/kak-sozdat-sayt-na-joomla-poshagovoe-rukovodstvo-dlya-nachinayushchikh.html>.

24 Создание сайта - алгоритм разработки // Электронная версия на сайте <http://joomla.ru/docs/articles/site-development/1918-creation-of-site>.

25 Создание сайта портфолио - Joomla! Может! // Электронная версия на сайте <https://joomla.ru/docs/articles/cms-joomla/2587-joomla-portfolio>.

26 Создание сайта на CMS Joomla // Электронная версия на сайте <https://www.seostop.ru/sozдание-saita/joomla.html>.

27 Управление пользователями внутри базы данных MySQL С помощью phpMyAdmin // Электронная версия на сайте <https://www.webhostingsecretrevealed.net/ru/blog/web-hosting-guides/a-short-guide-to-user-management-within-mysql-database-with-phpmyadmin/>.

28 MySQL и phpMyAdmin: включение сервера и доступ к его администрированию // Электронная версия на сайте <https://support.qnap.ru/hc/ru/articles/360000721274>.

29 Как пользоваться PhpMyAdmin // Электронная версия на сайте <https://losst.ru/kak-polzovatsya-phpmyadmin>.

30 Хаген Граф, Баскинов А. 10 легких шагов к освоению Joomla! 3.0 Хаген Граф // М. – 2012. – 250 с.

31 АК SHENU Building website with Joomla! 1.5 in 60 minutes // 2012. – 160 с.

32 Jennifer Marriott, Elin Waring The Official Joomla! Book // 2010. – 220 с.

33 Технология Flash В Создании Сайтов // Электронная версия на сайте <http://www.rusarticles.com/veb-dizajn-statya/tehnologiya-flashv-sozdanii-sajtov-3408023.html>.

34 PHP против ASP.Net - что лучше? // Электронная версия на сайте <http://nevlabs.ru/articles/web/php-vs-aspnet/>.

Приложение А

Техническое задание

А.1.1 Техническое задание на разработку мобильного приложения

Техническое задание распространяется на разработку онлайн маркетплейса для образовательной сферы, предназначенной для просмотра, подачи и продвижения объявлений. Потенциальными пользователями являются частные лица, покупающие или продающие образовательные услуги и продукты. Онлайн платформа избавит продавцов и покупателей от посредников и рекламных действий на других сайтах, а также позволит увеличить количество пользователей сервисом.

А.1.1.1 Основание для разработки

Система разрабатывается на основании дипломной работы.

А.1.1.2 Назначение

Платформа предназначена для просмотра, поиска, подачи и продвижения образовательных объявлений со всего Казахстана.

А.1.1.3 Требования к функциональным характеристикам

Список функций, которые приложение должно обеспечивать:

- отображение списка объявлений;
- отображение подробной информации об объявлении (с возможностью увеличения фотографий);
- регистрации в системе;
- авторизации в системе;
- хранение данных авторизованного пользователя;
- поиск объявлений (по тексту, городу или категории);
- фильтрация объявлений;
- сортировка объявлений (по времени или стоимости);
- подача объявления;

Продолжение приложения А

- восстановление пароля;
- получение данных из внутреннего хранилища.

А.1.1.4 Требования к надежности

Предусмотреть валидацию вводимых данных. Предусмотреть сбои и другие исключительные ситуации при работе пользователя с приложением. Обеспечить целостность данных.

А.1.1.5 Требования к составу и параметрам технических средств

Минимальные системные требования: 48 Мб свободной памяти; объем ОЗУ – 256 Мб и выше, Поддержка 4G – рекомендуется.

А.1.1.6 Требования к информационной и программной совместимости

Система предназначена для работы на любых устройствах с доступом в интернет.

А.1.1.7 Требования к программной документации

Тексты программ должны быть комментированы.

Приложение Б

Текст программы

```
<?php
/**
 * @package Joomla.Site
 *
 * @copyright Copyright (C) 2005 - 2019 Open Source Matters, Inc. All rights
reserved.
 * @license GNU General Public License version 2 or later; see LICENSE.txt
 */

/**
 * Define the application's minimum supported PHP version as a constant so it can be
referenced within the application.
 */
define('Joomla_MINIMUM_PHP', '5.3.10');

if (version_compare(PHP_VERSION, Joomla_MINIMUM_PHP, '<'))
{
    die('Your host needs to use PHP ' . Joomla_MINIMUM_PHP . ' or higher
to run this version of Joomla!');
}

// Saves the start time and memory usage.
$startTime = microtime(1);
$startMem = memory_get_usage();

/**
 * Constant that is checked in included files to prevent direct access.
 * define() is used in the installation folder rather than "const" to not error for PHP
5.2 and lower
 */
define('_JEXEC', 1);

if (file_exists(__DIR__ . '/defines.php'))
{
    include_once __DIR__ . '/defines.php';
}

if (!defined('_JDEFINES'))
{
```

Продолжение приложения Б

```
define('JPATH_BASE', __DIR__);
require_once JPATH_BASE . '/includes/defines.php';
}

require_once JPATH_BASE . '/includes/framework.php';

// Set profiler start time and memory usage and mark afterLoad in the profiler.
JDEBUG ? JProfiler::getInstance('Application')->setStart($startTime, $startMem)-
>mark('afterLoad') : null;

// Instantiate the application.
$app = JFactory::getApplication('site');

// Execute the application.
$app->execute();
<?php
class JConfig {
    public $offline = '0';
    public $offline_message = 'Сайт закрыт на техническое обслуживание.<br
/>Пожалуйста, зайдите позже.';
    public $display_offline_message = '1';
    public $offline_image = "";
    public $sitename = 'Marketplace - маркетплейс для образовательной
сферы';
    public $editor = 'tinymce';
    public $captcha = '0';
    public $list_limit = '20';
    public $access = '1';
    public $debug = '0';
    public $debug_lang = '0';
    public $debug_lang_const = '1';
    public $dbtype = 'mysqli';
    public $host = 'mysql370.cp.idhost.kz';
    public $user = 'u1143901_mp';
    public $password = 'kaz94zhelide';
    public $db = 'db1143901_marketplace';
    public $dbprefix = 'enu_';
    public $live_site = "";
    public $secret = '14sD1R9APjFdadY2';
    public $gzip = '0';
    public $error_reporting = 'default';
```

Продолжение приложения Б

```
public $helpurl =  
'https://help.joomla.org/proxy?keyref=Help{major}{minor}:{keyref}&lang={langcode}';  
public $ftp_host = "";  
public $ftp_port = "";  
public $ftp_user = "";  
public $ftp_pass = "";  
public $ftp_root = "";  
public $ftp_enable = '0';  
public $offset = 'UTC';  
public $mailonline = '1';  
public $mailer = 'mail';  
public $mailfrom = 'n.nysan@bk.ru';  
public $fromname = 'Marketplace - маркетплейс для образовательной  
сферы';  
public $sendmail = '/usr/sbin/sendmail';  
public $smtpauth = '0';  
public $smtpuser = "";  
public $smtp_pass = "";  
public $smtp_host = 'localhost';  
public $smtp_secure = 'none';  
public $smtp_port = '25';  
public $caching = '0';  
public $cache_handler = 'file';  
public $cachetime = '15';  
public $cache_platformprefix = '0';  
public $MetaDesc = 'Marketplace - маркетплейс образовательных  
программ и учебных курсов, дает возможность размещать образовательные  
события такие как: краткосрочные семинары, тренинги, мастер-классы и другие  
обучающие курсы по всем направлениям экономики Казахстана в одной базе.';  
public $MetaKeys = "";  
public $MetaTitle = '1';  
public $MetaAuthor = '1';  
public $MetaVersion = '0';  
public $robots = "";  
public $sef = '1';  
public $sef_rewrite = '0';  
public $sef_suffix = '1';  
public $unicodeslugs = '0';  
public $feed_limit = '10';  
public $feed_email = 'none';
```

Продолжение приложения Б

```
public $log_path =
'/var/www/vhosts/12/144497/webpace/httpdocs/mp.nelson.kz/administrator/logs';
    public $tmp_path =
'/var/www/vhosts/12/144497/webpace/httpdocs/mp.nelson.kz/tmp';
        public $lifetime = '15';
        public $session_handler = 'database';
        public $shared_session = '0';
        public $memcache_persist = '1';
        public $memcache_compress = '0';
        public $memcache_server_host = 'localhost';
        public $memcache_server_port = '11211';
        public $memcached_persist = '1';
        public $memcached_compress = '0';
        public $memcached_server_host = 'localhost';
        public $memcached_server_port = '11211';
        public $redis_persist = '1';
        public $redis_server_host = 'localhost';
        public $redis_server_port = '6379';
        public $redis_server_auth = "";
        public $redis_server_db = '0';
        public $proxy_enable = '0';
        public $proxy_host = "";
        public $proxy_port = "";
        public $proxy_user = "";
        public $proxy_pass = "";
        public $massmailoff = '0';
        public $replyto = "";
        public $replytoname = "";
        public $MetaRights = "";
        public $sitename_pagetitles = '2';
        public $force_ssl = '0';
        public $session_memcache_server_host = 'localhost';
        public $session_memcache_server_port = '11211';
        public $session_memcached_server_host = 'localhost';
        public $session_memcached_server_port = '11211';
        public $session_redis_persist = '1';
        public $session_redis_server_host = 'localhost';
        public $session_redis_server_port = '6379';
        public $session_redis_server_auth = "";
        public $session_redis_server_db = '0';
        public $frontediting = '1';
        public $cookie_domain = "";
```

Продолжение приложения Б

```
        public $cookie_path = "";
        public $asset_id = '1';
    }
<?php
/**
 * @package Helix3 Framework
 * @author JoomShaper http://www.joomshaper.com
 * @copyright Copyright (c) 2010 - 2015 JoomShaper
 * @license http://www.gnu.org/licenses/gpl-2.0.html GNU/GPLv2 or later
 */

defined('_JEXEC') or die;
$doc = JFactory::getDocument();
$app = JFactory::getApplication();

//Load Helix
$helix3_path = JPATH_PLUGINS.'/system/helix3/core/helix3.php';

if (file_exists($helix3_path)) {
    require_once($helix3_path);
    $this->helix3 = Helix3::getInstance();
} else {
    die('Please install and activate helix plugin');
}

$comingsoon_title = $this->params->get('comingsoon_title');
if( $comingsoon_title ) {
    $doc->setTitle( $comingsoon_title . ' | ' . $app->get('sitename') );
}

$comingsoon_date = explode('-', $this->params->get("comingsoon_date"));

//Load jQuery
JHtml::_('jquery.framework');
?>
<!DOCTYPE html>
<html class="sp-comingssoon" xmlns="http://www.w3.org/1999/xhtml"
xml:lang="<?php echo $this->language; ?>" lang="<?php echo $this->language; ?>"
dir="<?php echo $this->direction; ?>">
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

Продолжение приложения Б

```
<?php
if($favicon = $this->helix3->getParam('favicon')) {
    $doc->addFavicon( JURI::base(true) . '/' . $favicon);
} else {
    $doc->addFavicon( $this->helix3->getTemplateUri() . '/images/favicon.ico' );
}
?>
<jdoc:include type="head" />
<?php
$this->helix3->addCSS('bootstrap.min.css, font-awesome.min.css')
->lessInit()->setLessVariables(array(
    'preset'=>$this->helix3->Preset(),
    'bg_color'=> $this->helix3->PresetParam('_bg'),
    'text_color'=> $this->helix3->PresetParam('_text'),
    'major_color'=> $this->helix3->PresetParam('_major')
))
->addLess('master', 'template')
->addLess('presets', 'presets/'.$this->helix3->Preset())
->addJS('jquery.countdown.min.js');
?>
</head>
<body>
<div class="sp-comingsoon-wrap">
    <div class="container">
        <div class="text-center">
            <div id="sp-comingsoon">
                <?php if( $comingsoon_title ) { ?>
                    <h1 class="sp-comingsoon-title">
                        <?php echo $comingsoon_title; ?>
                    </h1>
                <?php } ?>

                <?php if( $this->params->get('comingsoon_content')
) { ?>
                    <div class="sp-comingsoon-content">
                        <?php echo $this->params-
>get('comingsoon_content'); ?>
                    </div>
                <?php } ?>

                <div id="sp-comingsoon-countdown" class="sp-
comingsoon-countdown">
```

Продолжение приложения Б

```
</div>

<?php if($this->countModules('comingssoon')) { ?>
<div class="sp-position-comingssoon">
    <jdoc:include type="modules"
name="comingssoon" style="sp_xhtml" />
</div>
<?php } ?>

<?php
//Social Icons
$facebook      = $this->params->get('facebook');
$twitter        = $this->params->get('twitter');
$googleplus     = $this->params->get('googleplus');
$Pinterest      = $this->params->get('pinterest');
$youtube        = $this->params->get('youtube');
$linkedin       = $this->params->get('linkedin');
$dribbble       = $this->params->get('dribbble');
$behance        = $this->params->get('behance');
$skype          = $this->params->get('skype');
$flickr         = $this->params->get('flickr');
$vk             = $this->params->get('vk');

if( $this->params->get('show_social_icons') && (
$facebook || $twitter || $googleplus || $Pinterest || $youtube || $linkedin || $dribbble ||
$behance || $skype || $flickr || $vk ) ) {
    $html = '<ul class="social-icons">';

    if( $facebook ) {
        $html .= '<li><a target="_blank"
href="'. $facebook .'"><i class="fa fa-facebook"></i></a></li>';
    }
    if( $twitter ) {
        $html .= '<li><a target="_blank"
href="'. $twitter .'"><i class="fa fa-twitter"></i></a></li>';
    }
    if( $googleplus ) {
        $html .= '<li><a target="_blank"
href="'. $googleplus .'"><i class="fa fa-google-plus"></i></a></li>';
    }
    if( $Pinterest ) {
```

Продолжение приложения Б

```

                                $html .= '<li><a target="_blank"
href="'. $pinterest ."'><i class="fa fa-pinterest"></i></a></li>';
                                }
                                if( $youtube ) {
                                $html .= '<li><a target="_blank"
href="'. $youtube ."'><i class="fa fa-youtube"></i></a></li>';
                                }
                                if( $linkedin ) {
                                $html .= '<li><a target="_blank"
href="'. $linkedin ."'><i class="fa fa-linkedin"></i></a></li>';
                                }
                                if( $dribbble ) {
                                $html .= '<li><a target="_blank"
href="'. $dribbble ."'><i class="fa fa-dribbble"></i></a></li>';
                                }
                                if( $behance ) {
                                $html .= '<li><a target="_blank"
href="'. $behance ."'><i class="fa fa-behance"></i></a></li>';
                                }
                                if( $flickr ) {
                                $html .= '<li><a target="_blank"
href="'. $flickr ."'><i class="fa fa-flickr"></i></a></li>';
                                }
                                if( $vk ) {
                                $html .= '<li><a target="_blank"
href="'. $vk ."'><i class="fa fa-vk"></i></a></li>';
                                }
                                if( $skype ) {
                                $html .= '<li><a href="skype:'.
$skype .'?chat"><i class="fa fa-skype"></i></a></li>';
                                }
                                }
                                $html .= '<ul>';

                                echo $html;
                                }

                                ?>

                                </div>
                                </div>
                                </div>
```

```
</div>

<script type="text/javascript">

    jQuery(function($) {
        $('#sp-comingsoon-countdown').countdown('<?php echo
trim($comingsoon_date[2]); ?><?php echo trim($comingsoon_date[1]); ?><?php
echo trim($comingsoon_date[0]); ?>', function(event) {
            $(this).html(event.strftime('<div class="days"><span
class="number">%-D</span><span class="string">% !D:<?php echo
JText::_("HELIX_DAY"); ?>,<?php echo JText::_("HELIX_DAYS");
?></span></div><div class="hours"><span class="number">%H</span><span
class="string">% !H:<?php echo JText::_("HELIX_HOUR"); ?>,<?php echo
JText::_("HELIX_HOURS"); ?></span></div><div class="minutes"><span
class="number">%M</span><span class="string">% !M:<?php echo
JText::_("HELIX_MINUTE"); ?>,<?php echo JText::_("HELIX_MINUTES");
?></span></div><div class="seconds"><span class="number">%S</span><span
class="string">% !S:<?php echo JText::_("HELIX_SECOND"); ?>,<?php echo
JText::_("HELIX_SECONDS"); ?></span></div>'));
        });
    });

</script>

</body>
</html>
<?php
/**
 * @package Helix3 Framework
 * @author JoomShaper http://www.joomshaper.com
 * @copyright Copyright (c) 2010 - 2015 JoomShaper
 * @license http://www.gnu.org/licenses/gpl-2.0.html GNU/GPLv2 or later
 */
//no direct acces
defined ('_JEXEC') or die ('resticted access');
?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this-
>language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this-
>direction; ?>">
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
```

Продолжение приложения Б

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<?php
$doc = JFactory::getDocument();

if($favicon = $this->params->get('favicon')) {
    $doc->addFavicon( JURI::base(true) . '/' . $favicon);
} else {
    $doc->addFavicon( $this->baseurl . '/templates/'. $this->template
./images/favicon.ico' );
}
?>

<jdoc:include type="head" />
    <link rel="stylesheet" href="<?php echo $this->baseurl;
?>/media/jui/css/bootstrap.min.css" type="text/css" />
    <link rel="stylesheet" href="<?php echo $this->baseurl; ?>/templates/<?php
echo $this->template; ?>/css/frontend-edit.css" type="text/css" />
    <link rel="stylesheet" href="<?php echo $this->baseurl; ?>/templates/<?php echo
$this->template; ?>/css/template.css" type="text/css" />
</head>
<body class="contentpane">
    <jdoc:include type="message" />
    <jdoc:include type="component" />
</body>
</html>
<?php
/**
 * @package Helix3 Framework
 * @author JoomShaper http://www.joomshaper.com
 * @copyright Copyright (c) 2010 - 2015 JoomShaper
 * @license http://www.gnu.org/licenses/gpl-2.0.html GNU/GPLv2 or later
 */
//no direct acces
defined ( '_JEXEC' ) or die ( 'resticted access' );

$doc = JFactory::getDocument();
$params = JFactory::getApplication()->getTemplate('true')->params;

//Favicon
if($favicon = $params->get('favicon')) {
    $doc->addFavicon( JURI::base(true) . '/' . $favicon);
} else {
```

Продолжение приложения Б

```
$doc->addFavicon( $this->baseurl . '/templates/' . $this->template .
'/images/favicon.ico' );
}

//Logo
if($logo_image = $params->get('logo_image')) {
    $logo = JURI::base(true) . '/' . $logo_image;
} else {
    $logo = $this->baseurl . '/templates/' . $this->template .
'/images/presets/preset1/logo@2x.png';
}

//Stylesheets
$doc->addStylesheet( $this->baseurl . '/templates/' . $this->template .
'/css/bootstrap.min.css' );
$doc->addStylesheet( $this->baseurl . '/templates/' . $this->template . '/css/font-
awesome.min.css' );
$doc->addStylesheet( $this->baseurl . '/templates/' . $this->template .
'/css/template.css' );

$doc->setTitle($this->error->getCode() . ' - ' . $this->title);
$header_contents = "";
if(!class_exists('JDocumentRendererHead')) {
    $head = JPATH_LIBRARIES . '/joomla/document/html/renderer/head.php';
    if(file_exists($head)) {
        require_once($head);
    }
}
$header_renderer = new JDocumentRendererHead($doc);
$header_contents = $header_renderer->render(null);

?>
<!DOCTYPE html>
<html class="error-page" xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php
echo $this->language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo
$this->direction; ?>">
    <head>
        <meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <?php echo $header_contents; ?>
    </head>
```

Продолжение приложения Б

```
<body>
  <div class="error-page-inner">
    <div>
      <div class="container">
        <div class="error-page-logo">
          
        </div>
        <h1 class="error-code"><?php echo $this-
>error->getCode(); ?></h1>
        <p class="error-message"><?php echo $this-
>error->getMessage(); ?></p>
        <a class="btn btn-primary btn-lg error-
button" href="<?php echo $this->baseurl; ?>/" title="<?php echo JText::_('HOME');
?>"><i class="fa fa-chevron-left"></i> <?php echo JText::_('HELIX_GO_BACK');
?></a>
        <?php echo $doc->getBuffer('modules',
'404', array('style' => 'sp_xhtml')); ?>
      </div>
    </div>
  </div>
</body>
</html>
<?php
/**
 * @package Helix3 Framework
 * Template Name - Shaper Varsita
 * @author JoomShaper http://www.joomshaper.com
 * @copyright Copyright (c) 2010 - 2015 JoomShaper
 * @license http://www.gnu.org/licenses/gpl-2.0.html GNU/GPLv2 or later
 */
//no direct acces
defined ('_JEXEC') or die ('resticted access');

$doc = JFactory::getDocument();
$app = JFactory::getApplication();
$menu = $app->getMenu()->getActive();

JHtml::_('jquery.framework');
JHtml::_('bootstrap.framework'); //Force load Bootstrap
unset($doc->_scripts[$this->baseurl . '/media/jui/js/bootstrap.min.js']); // Remove
joomla core bootstrap
```

Продолжение приложения Б

```
//Load Helix
$helix3_path = JPATH_PLUGINS.'/system/helix3/core/helix3.php';

if (file_exists($helix3_path)) {
    require_once($helix3_path);
    $this->helix3 = helix3::getInstance();
} else {
    die('Please install and activate helix plugin');
}

//Coming Soon
if($this->helix3->getParam('comingsoon_mode')) header("Location: ".$this->baseUrl."&?tmpl=comingsoon");

//Class Classes
$body_classes = "";

if($this->helix3->getParam('sticky_header')) {
    $body_classes .= ' sticky-header';
}

$body_classes .= ($this->helix3->getParam('boxed_layout', 0)) ? ' layout-boxed' : ' layout-fluid';

if (isset($menu) && $menu) {
    if ($menu->params->get('pageclass_sfx')) {
        $body_classes .= ' ' . $menu->params->get('pageclass_sfx');
    }
}

//Body Background Image
if($bg_image = $this->helix3->getParam('body_bg_image')) {

    $body_style = 'background-image: url(' . JURI::base(true) . '/' . $bg_image . ');';
    $body_style .= 'background-repeat: ' . $this->helix3->getParam('body_bg_repeat') . ' ';
    $body_style .= 'background-size: ' . $this->helix3->getParam('body_bg_size') . ' ';
    $body_style .= 'background-attachment: ' . $this->helix3->getParam('body_bg_attachment') . ' ';
    $body_style .= 'background-position: ' . $this->helix3->getParam('body_bg_position') . ' ';
    $body_style = 'body.site { ' . $body_style . ' }';
}
```

```
$doc->addStydeclaration( $body_style );
}

//Body Font
$webfonts = array();

if( $this->params->get('enable_body_font') ) {
    $webfonts['body'] = $this->params->get('body_font');
}

//Heading1 Font
if( $this->params->get('enable_h1_font') ) {
    $webfonts['h1'] = $this->params->get('h1_font');
}

//Heading2 Font
if( $this->params->get('enable_h2_font') ) {
    $webfonts['h2'] = $this->params->get('h2_font');
}

//Heading3 Font
if( $this->params->get('enable_h3_font') ) {
    $webfonts['h3'] = $this->params->get('h3_font');
}

//Heading4 Font
if( $this->params->get('enable_h4_font') ) {
    $webfonts['h4'] = $this->params->get('h4_font');
}

//Heading5 Font
if( $this->params->get('enable_h5_font') ) {
    $webfonts['h5'] = $this->params->get('h5_font');
}

//Heading6 Font
if( $this->params->get('enable_h6_font') ) {
    $webfonts['h6'] = $this->params->get('h6_font');
}

//Navigation Font
if( $this->params->get('enable_navigation_font') ) {
```

Продолжение приложения Б

```
$webfonts['.sp-megamenu-parent'] = $this->params->get('navigation_font');
}

//Custom Font
if( $this->params->get('enable_custom_font') && $this->params-
>get('custom_font_selectors') ) {
    $webfonts[ $this->params->get('custom_font_selectors') ] = $this->params-
>get('custom_font');
}

$this->helix3->addGoogleFont($webfonts);

//Custom CSS
if($custom_css = $this->helix3->getParam('custom_css')) {
    $doc->addStyledeclaration( $custom_css );
}

//Custom JS
if($custom_js = $this->helix3->getParam('custom_js')) {
    $doc->addScriptdeclaration( $custom_js );
}

?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this-
>language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this-
>direction; ?>">
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <?php
if($favicon = $this->helix3->getParam('favicon')) {
    $doc->addFavicon( JURI::base(true) . '/' . $favicon);
} else {
    $doc->addFavicon( $this->helix3->getTemplateUri() . '/images/favicon.ico' );
}
    ?>

<jdoc:include type="head" />

<?php
```

Продолжение приложения Б

```
$this->helix3->addCSS('bootstrap.min.css, font-awesome.min.css') // CSS Files
->addJS('bootstrap.min.js, jquery.sticky.js, smoothscroll.js, main.js') // JS Files
->lessInit()->setLessVariables(array(
    'preset'=>$this->helix3->Preset(),
    'bg_color'=> $this->helix3->PresetParam('_bg'),
    'text_color'=> $this->helix3->PresetParam('_text'),
    'major_color'=> $this->helix3->PresetParam('_major')
))
->addLess('legacy/bootstrap', 'legacy')
->addLess('master', 'template');

//RTL
if($this->direction=='rtl') {
    $this->helix3->addCSS('bootstrap-rtl.min.css')
    ->addLess('rtl', 'rtl');
}

$this->helix3->addLess('presets', 'presets/'.$this->helix3->Preset(),
array('class'=>'preset'));

//Before Head
if($before_head = $this->helix3->getParam('before_head')) {
    echo $before_head . "\n";
}
?>
</head>
<body class="<?php echo $this->helix3->bodyClass( $body_classes ); ?>">
<div class="body-innerwrapper">
    <?php $this->helix3->generatelayout(); ?>

    <div class="offcanvas-menu">
        <a href="#" class="close-offcanvas"><i class="fa fa-remove"></i></a>
        <div class="offcanvas-inner">
            <?php if ($this->helix3->countModules('offcanvas')) { ?>
                <jdoc:include type="modules" name="offcanvas" style="sp_xhtml" />
            <?php } else { ?>
                <p class="alert alert-warning"><?php echo
JText::_('HELIX_NO_MODULE_OFFCANVAS'); ?></p>
                <?php } ?>
            </div>
        </div>
    </div>
</div>
```

Продолжение приложения Б

```
<?php

if($this->params->get('compress_css')) {
    $this->helix3->compressCSS();
}

if($this->params->get('compress_js')) {
    $this->helix3->compressJS( $this->params->get('exclude_js') );
}

if($before_body = $this->helix3->getParam('before_body')) {
    echo $before_body . "\n";
}

?>
<jdoc:include type="modules" name="debug" />
<link rel="SHORTCUT ICON"
href="/templates/shaper_varsita/favicon.ico" type="image/x-icon" />

</body>
</html>
<?php
/**
 * @package Helix3 Framework
 * @author JoomShaper http://www.joomshaper.com
 * @copyright Copyright (c) 2010 - 2015 JoomShaper
 * @license http://www.gnu.org/licenses/gpl-2.0.html GNU/GPLv2 or later
 */
//no direct acces
defined ( '_JEXEC' ) or die ( 'resticted acces' );

$doc = JFactory::getDocument();
$app = JFactory::getApplication();

require_once JPATH_ADMINISTRATOR .
'/components/com_users/helpers/users.php';

$twofactormethods = UsersHelper::getTwoFactorMethods();
?>
<!DOCTYPE html>
```

Продолжение приложения Б

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo $this->direction; ?>">
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <?php
  if($favicon = $this->params->get('favicon')) {
    $doc->addFavicon( JURI::base(true) . '/' . $favicon);
  } else {
    $doc->addFavicon( $this->baseurl . '/templates/'. $this->template
./images/favicon.ico' );
  }
  ?>

  <jdoc:include type="head" />
  <link rel="stylesheet" href="<?php echo $this->baseurl; ?>/templates/<?php
echo $this->template; ?>/css/bootstrap.min.css" type="text/css" />
  <link rel="stylesheet" href="<?php echo $this->baseurl; ?>/templates/<?php
echo $this->template; ?>/css/template.css" type="text/css" />
<body>
  <div class="container">
    <div class="row">
      <div class="col-sm-6 col-sm-offset-3">
        <div class="offline-inner">
          <jdoc:include type="message" />

          <div id="frame" class="outline">
            <?php if ($app->get('offline_image')
&& file_exists($app->get('offline_image'))): ?>
              get('sitename')); ?>" />
            <?php endif; ?>
            <h1>
              <?php echo
htmlspecialchars($app->get('sitename')); ?>
            </h1>
            <?php if ($app-
>get('display_offline_message', 1) == 1 && str_replace(' ', '', $app-
>get('offline_message')) != "") : ?>
              <p>
```

Продолжение приложения Б

```
<?php echo $app->get('offline_message'); ?>
</p>
<?php elseif ($app-
>get('display_offline_message', 1) == 2 && str_replace(' ', ",
JText::_('JOFFLINE_MESSAGE')) != ") : ?>
<p>
<?php echo
JText::_('JOFFLINE_MESSAGE'); ?>
</p>
<?php endif; ?>
<form action="<?php echo
JRoute::_('index.php', true); ?>" method="post" id="form-login">
<div class="form-group"
id="form-login-username">
<input
name="username" id="username" type="text" class="form-control"
placeholder="<?php echo JText::_('JGLOBAL_USERNAME'); ?>" size="18" />
</div>
<div class="form-group"
id="form-login-password">
<input
type="password" name="password" class="form-control" size="18"
placeholder="<?php echo JText::_('JGLOBAL_PASSWORD'); ?>" id="passwd" />
</div>
<?php if
(count($twofactormethods) > 1) : ?>
<div class="form-group"
id="form-login-secretkey">
<input type="text"
name="secretkey" class="form-control" size="18" placeholder="<?php echo
JText::_('JGLOBAL_SECRETKEY'); ?>" id="secretkey" />
</div>
<?php endif; ?>
<div class="form-group"
id="submit-buton">
<input
type="submit" name="Submit" class="btn btn-success login" value="<?php echo
JText::_('JLOGIN'); ?>" />
</div>
```

Продолжение приложения Б

```
name="option" value="com_users" />
name="task" value="user.login" />
name="return" value="<?php echo base64_encode(JUri::base()); ?>" />
JHtml::_('form.token'); ?>
</form>
</div>
</div>
</div>
</div>
</body>
</html>
```

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

ОТЗЫВ
НАУЧНОГО РУКОВОДИТЕЛЯ
на дипломный проект

Толыбековой Арайлы Маликовны

5B070400 – Вычислительная Техника и Программное Обеспечение

На тему: " Платформа маркетплейса для образовательной сферы"

Оценка работы

Тема дипломной работы является актуальной. Современное развитие интернета, развитие онлайн-курсов, трансформация целей образования и влияние эпидемиологической обстановки способствует новым формам развития образовательных процессов и взаимодействия его участников. Маркетплейс является примером площадки, где возможна быстрая и объемная монетизация знаний.

Дипломница выполнила исследования в данной области и реализовала рабочую версию платформы с использованием современных инструментов программирования. Все работы выполнены строго по графику и с высокой ответственностью.

В процессе работы над дипломом автор проявила трудолюбие, самостоятельность в принятии проектных решений, профессионализм в подготовке текстовых и графических материалов.

Дипломный проект выполнен с учетом всех требований по данной специальности. Студент Толыбекова А.М. рекомендована к прохождению защиты диплома.

Научный руководитель

Сениор-лектор



Жумабаев А.А.

"5" июня 2021 г.

Метаданные

Название

Измененный - Платформа маркетплейса для образовательной сферы.docx

Автор

Арайлы Толыбекова

Научный руководитель

Жибек Алибиева

Подразделение

ИКИИТ

Список возможных попыток манипуляций с текстом

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв		14
Интервалы		0
Микропробелы		3
Белые знаки		0
Парафразы (SmartMarks)		68

Объем найденных подобиий

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.


25

Длина фразы для коэффициента подобия 2


11079

Количество слов


85160

Количество символов

Подобия по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз

Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	https://www.seostop.ru/sozdanie-saita/joomla/adminka.html	51	0.46 %
2	https://www.seostop.ru/sozdanie-saita/joomla/adminka.html	47	0.42 %
3	https://www.seostop.ru/sozdanie-saita/joomla/adminka.html	40	0.36 %
4	https://timeweb.com/ru/community/articles/sozdaem-svoy-shablon-dlya-joomla-poshagovoe-rukovodstvo	36	0.32 %
5	https://timeweb.com/ru/community/articles/sozdaem-svoy-shablon-dlya-joomla-poshagovoe-rukovodstvo	34	0.31 %

6	https://www.woldi.ru/raznoe/joomla-ne-zahodit-v-adminku-joomla-ne-puskaet-v-adminku-stranicza-obnovlyaetsya-bez-rezultata.html	31	0.28 %
7	https://timeweb.com/ru/community/articles/sozdaem-svoy-shablon-dlya-joomla-poshagovoe-rukovodstvo	28	0.25 %
8	https://www.seostop.ru/sozдание-saita/joomla/adminka.html	20	0.18 %
9	https://timeweb.com/ru/community/articles/sozdaem-svoy-shablon-dlya-joomla-poshagovoe-rukovodstvo	20	0.18 %
10	https://www.seostop.ru/sozдание-saita/joomla/adminka.html	19	0.17 %

из базы данных RefBooks (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из домашней базы данных (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

из программы обмена базами данных (0.27 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	KazNAU/9015_e93bdd265dfe9c5926ea7819f816a4a6.docx KazNAU 5/19/2021 Kazakh National Agrarian University (KazNAU)	30 (3) 0.27 %

из интернета (5.81 %)

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	https://www.seostop.ru/sozдание-saita/joomla/adminka.html	309 (18) 2.79 %
2	https://timeweb.com/ru/community/articles/sozdaem-svoy-shablon-dlya-joomla-poshagovoe-rukovodstvo	200 (14) 1.81 %
3	https://www.joomlafr.fr/en/joomla-documentations-tutorials/joomla-3-x-templates-cheatsheet/doc_download/68-joomla-3-x-templates-cheatsheet	58 (7) 0.52 %
4	https://www.woldi.ru/raznoe/joomla-ne-zahodit-v-adminku-joomla-ne-puskaet-v-adminku-stranicza-obnovlyaetsya-bez-rezultata.html	36 (2) 0.32 %
5	https://www.docsity.com/ru/internet-magazin-wordpress/5223891/	30 (3) 0.27 %
6	https://ofisrecycle.weebly.com/blog/uchebnik-po-front-page2003	11 (1) 0.10 %

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	------------	---

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы :

Автор: Толыбекова Арайлы Маликовна

Координатор: Жибек Алибиева

Название: Измененный - Платформа маркетплейса для образовательной сферы.docx

Коэффициент подобия 1: 6.08%

Коэффициент подобия 2: 2.41%

Коэффициент цитирования: 2.36%

Замена букв: 14

Белые знаки: 0

Микропробелы: 3

Интервалы: 0

После анализа Отчета подобия констатирую следующее:

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите. Обоснование:

.....
.....
.....
.....
.....

.....
.....
6 июня 2021

Дата

A small, rectangular image showing a handwritten signature in blue ink on a light-colored background. The signature is somewhat stylized and difficult to read.

Подпись Научного руководителя