

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра веб-технологий и компьютерного моделирования

КЕСС
Богдан Игоревич

РАЗРАБОТКА ИНФОРМАЦИОННОГО РЕСУРСА
«ЦЕНТР КАРЬЕРЫ БГУ»

Дипломная работа

Научный руководитель:
кандидат физ.-мат. наук,
доцент Ю.В. Позняк

Допущена к защите

« ___ » _____ 2018 г.

Зав. кафедрой веб-технологий и
компьютерного моделирования
кандидат физ.-мат. наук,
доцент _____ В.С Романчик

Минск, 2018

ОГЛАВЛЕНИЕ

Перечень условных обозначений	3
Реферат	4
Введение	7
1 Обзор и анализ методов и средств разработки информационного ресурса	9
1.1 Анализ предметной области информационного ресурса.....	9
1.2 Анализ существующих аналогов.....	11
1.3 Обзор и обоснование выбора инструментария разработки информационного ресурса	13
2 Алгоритмический анализ задачи. Разработка графического интерфейса и лингвистического обеспечения информационного ресурса.....	18
2.1 Функциональная модель информационного ресурса.....	18
2.2 Логическая модель и диаграмма прецедентов	20
2.3 Графический интерфейс информационного ресурса	23
2.4 Проектирование лингвистического обеспечения.....	26
3 Описание технологии реализации информационного ресурса	28
3.1 Подготовка к установке	28
3.2 Настройка сервера и дополнительных компонентов	29
3.3 Установка системы управления контентом «Joomla»	32
3.4 Разработка информационного ресурса	35
4 Апробация функционирования информационного ресурса	46
Заключение.....	51
Список использованной литературы	52
Приложение А Код информационного ресурса	53
Приложение Б Руководство пользователя	76

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

В настоящей пояснительной записке применяются следующие термины, обозначения и сокращения.

БД — база данных.

Клиент — это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.

Сервер — это программный компонент вычислительной системы, выполняющий сервисные функции по запросу клиента, предоставляя ему доступ к определённым ресурсам.

СУБД — система управления базами данных.

«DFD» — диаграмма потока данных.

«CMS» — информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым.

«IDEF0» — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

«Joomla» — система управления содержимым, написанная на языках «PHP» и «JavaScript», использующая в качестве хранилища базы данных «MySQL». Является свободным программным обеспечением, распространяемым под лицензией «GNU GPL».

«Apache HTTP-сервер» — свободный веб-сервер, поддерживающий операционные системы «Linux», «BSD», «Mac OS», «Microsoft Windows».

«AJAX» — «асинхронный JavaScript и XML» — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером.

РЕФЕРАТ

Дипломная работа выполнена в объеме: пояснительная записка на 84 страницах, 46 иллюстраций, 1 таблица, 10 источников, 2 приложения.

Ключевые слова: «Центр карьеры БГУ», информационный ресурс, онлайн ресурс, функциональное проектирование, информационное обеспечение.

Объектом разработки является универсальный информационный ресурс «Центр карьеры БГУ».

Цель разработки заключается в оптимизации процесса поиска работы выпускниками, а также повышении имиджа БГУ и его рейтинга.

В процессе проектирования выполнены следующие исследования:

- исследование специфики распределения и публикации базы вакансий;
- изучение требований предъявляемых при регистрации пользователей;
- разработка функциональной и логической модели ресурса;
- разработка верификации и апробации информационного ресурса.

Элементом практической значимости является, информационный ресурс, позволяющий управлять процессом поиска работы студентами, а также вести мониторинг за процессом публикации новых вакансий.

Областью практического применения является информационный ресурс Белорусского Государственного Университета.

В ходе дипломной работы прошли апробацию такие предложения, как механизм создания вакансии, механизм создание профиля компании, приглашения соискателя на собеседование.

Приведенный в дипломной работе расчетно-аналитический материал объективно отражает состояние разрабатываемого объекта, все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов.

РЭФЕРАТ

Дыпломная праца выканана ў аб'ёме: тлумачальная запіска на 84 старонках, 46 ілюстрацый, 1 табліца, 10 крыніц, 2 дадаткі.

Ключавыя словы: «Цэнтр кар'еры БДУ», інфармацыйны рэсурс, онлайн рэсурс, функцыянальнае праектаванне, інфармацыйнае забеспячэнне.

Аб'ектам распрацоўкі з'яўляецца універсальны інфармацыйны рэсурс «Цэнтр кар'еры БДУ».

Мэта распрацоўкі заключаецца ў аптымізацыі працэсу пошуку працы выпускнікамі, а таксама павышэнні іміджу БДУ і яго рэйтынгу.

У працэсе праектавання выкананы наступныя даследаванні:

- даследаванне спецыфікі размеркавання і публікацыі базы вакансій;
- вывучэнне патрабаванняў, якія прад'яўляюцца пры рэгістрацыі карыстальнікаў;
- распрацоўка функцыянальнай і лагічнай мадэлі рэсурсу;
- распрацоўка верыфікацыі і апрабавання інфармацыйнага рэсурсу.

Элементарам практычнай значнасці з'яўляецца, інфармацыйны рэсурс, які дазваляе кіраваць працэсам пошуку працы студэнтамі, а таксама весці маніторынг за працэсам публікацыі новых вакансій.

Вобласцю практычнага прымянення з'яўляецца інфармацыйны рэсурс Беларускага Дзяржаўнага Універсітэта.

У ходзе дыпломнай працы прайшлі апрабаванню такія прапановы, як механізм стварэння вакансіі, механізм стварэння профілю кампаніі, запрашэння суіскальніка на сумоўе.

Прыведзены ў дыпломнай працы разлікова-аналітычны матэрыял аб'ектыўна адлюстроўвае стан распрацоўванага аб'екта, усе запазычаныя з літаратурных і іншых крыніц тэарэтычныя і метадалагічныя палажэнні і канцэпцыі суправаджаюцца спасылкамі на іх аўтараў.

ABSTRACT

Diploma work is completed in the following state: explanatory note consisted of 84 pages, 46 pictures, 1 table, 10 information sources, 2 applications.

Key words: «BSU career center», information resource, online resource, functional design, information support.

The object of development is universal information resource “BSU career center”.

The design goal of the development is to optimize the process of job search for graduates, as well as to improve the image of BSU and its rating.

The following research are performed during design process:

- research of vacancies base distribution and publication;
- performing analysis of requirements provided for users registration and CV creation;
- development of resource functional model;
- development of information resource verification and approbation.

The ability to control fully and more faster the process of job seeking and monitor vacancies publishing process are practical elements of completed results.

The practical application field of this work is Belorussian State University.

During the process of diploma work design were approved such suggestions like vacancy creation process, company profile creation process, applicant invitation for the interview process.

The calculation and analytical material given in this diploma work objectively depicts the state of developing object, all the theoretical and methodological documents, taken from literature and other sources, and concepts are followed with links to their authors.

ВВЕДЕНИЕ

Любое современное учреждение высшего образования сталкивается с процессом распределения выпускников. Вопрос распределения является очень актуальным в связи с нестабильной ситуацией на рынке труда.

Выпускники Белорусского Государственного Университета пользуются спросом среди зарубежных и отечественных компаний.

С целью оптимизации процесса распределения и своевременного информирования студентов о проводимых мероприятиях и актуальных вакансиях было принято решение о разработке информационного ресурса «Центр карьеры БГУ» (План мероприятий по повышению рейтинга БГУ на 2017 год, «Разработать и ввести в эксплуатацию сайт виртуального центра по содействию трудоустройства и развитию карьеры» до 29.12.17, приказ ректора БГУ от 14.03.2017 № 114-ОД).

Целью данной дипломной работы является разработка и внедрение информационного онлайн ресурса «Центра карьеры БГУ».

«Центр карьеры БГУ» должен стать связующим звеном между нанимателями, студентами и высшим учебным заведением.

Следует отметить, что ресурс должен оптимизировать и упростить процесс поиска вакансий, а также повысить имидж университета и его рейтинг.

Для выполнения цели дипломной работы необходимо решить следующие задачи:

- исследовать ресурсы зоны «ВУ» на предмет наличия аналогов;
- проанализировать предметную область;
- исследовать специфику распределения и публикации базы вакансий;
- провести анализ требований, предъявляемых при регистрации пользователей и создании резюме;

- разработать структуру информационного ресурса с предъявленными требованиями;
- разработать функциональную модель ресурса;
- сверстать тему оформления;
- разработать информационные модули главной страницы;
- установить дополнительное расширение для управления вакансиями;
- осуществить апробацию и разработать руководство пользователя.

1 ОБЗОР И АНАЛИЗ МЕТОДОВ И СРЕДСТВ РАЗРАБОТКИ ИНФОРМАЦИОННОГО РЕСУРСА

1.1 Анализ предметной области информационного ресурса

Предметной областью называется часть реального мира, представляющая интерес для данного исследования.

Для оптимизации и совершенствования процедуры распределения выпускников необходим современный информационный ресурс, который обладает всеми необходимыми требованиями.

Информационный ресурс должен содержать информацию об актуальных вакансиях и резюме самих выпускников, а также иметь адаптивный интерфейс.

Необходимо организовать механизм регистрации пользователей с возможностью выбора роли. Система должна предусматривать 2 роли: «Наниматель» и «Соискатель». Пользователи роли «Наниматель» должны иметь возможность создать в системе свою компанию и указать следующие данные:

- название компании;
- контактное лицо;
- контактный телефон;
- контактный адрес электронной почты;
- веб-сайт;
- описание компании.

Наниматели должны иметь возможность создать вакансии с предварительной модерацией администрацией ресурса.

Следует отметить, что в системе должен быть организован поиск по вакансиям для соискателей и поиск резюме для нанимателей.

Раздел просмотра вакансий доступен для всех пользователей. Отправить сопроводительное письмо и откликнуться на вакансию должны иметь возможность только зарегистрированные пользователи.

В разделе «Вакансии» должны отображаться все созданные вакансии в форме списка. Рядом с каждой вакансией должна быть кнопка «Отклик на вакансию».

В разделе «Резюме» наниматели могут просмотреть все резюме соискателей по категориям. Данный раздел должен перенаправлять к списку резюме, созданному на базе факультета радиофизики БГУ.

Раздел «Распределение» предоставляет информацию необходимую для распределения. Следует отметить, что в этом разделе должен быть реализован сервис «Часто задаваемые вопросы» по организации обратной связи с представителями университета. Следует предусмотреть подраздел «Истории успеха», который должен содержать информацию о выпускниках и их истории трудоустройства в компании.

Раздел «Мероприятия» предназначен для информирования студентов, преподавателей, нанимателей о проводимых мероприятиях. Раздел необходимо разделить на две составляющие: актуальные мероприятия и прошедшие.

Раздел «Работодателям» должен в полной мере отражать информацию для работодателей, а также документы, необходимые для размещения вакансий и ссылка на вход в личный кабинет.

Раздел «Контакты» должен отражать информацию об актуальных контактах: телефоны, электронный адрес, юридический адрес, реквизиты.

В системе необходимо предусмотреть гибкую систему оповещений для нанимателей и соискателей. Также должен быть организован механизм сообщений между пользователями.

При отклике на вакансию наниматель должен получать сообщение на адрес электронной почты, указанный при регистрации. Администраторы ресурса должны получать сообщения при создании компании и добавлении вакансии.

Информационный ресурс должен позволять администраторам редактировать список категорий для размещения и группировки вакансий, а также всех полей, указанных в разделе «Вакансии» и «Резюме».

Детально проанализировав предметную область, становится видно, что необходимо связующее звено между студентами, выпускниками и нанимателями.

Таким образом, следует, что разработка информационного ресурса «Центр карьеры БГУ» очень актуальна.

1.2 Анализ существующих аналогов

«Работа ТУТ.БАЙ» — известный ресурс в Республике Беларусь по поиску сотрудников, созданный в 2008 году. Как и другие системы подобного рода, «Работа ТУТ.БАЙ» предоставляет возможность оценивать компании, что поднимает или понижает репутацию зарегистрированных нанимателей. Проект создан с использованием «РНР».

Состав ресурса «Работа. ТУТ.БАЙ»:

- база резюме сотрудников, структурированная по категориям;
- база нанимателей, ранжированных по рейтингу.

Преимущества:

- простой в работе и интуитивно понятный интерфейс.
- интеграция с популярными сервисами и социальными сетями;
- наличие средств, упрощающих идентификацию, поиск резюме и вакансий.

Недостатки:

- зависимость от ресурса «ТУТ.Бай»;
- высокая стоимость размещения вакансий;

- высокая стоимость доступа к базе резюме;
- громоздкий механизм поиска вакансий.

Интерфейс ресурса «Работа ТУТ.БАЙ» приведен на рисунке 1.1.

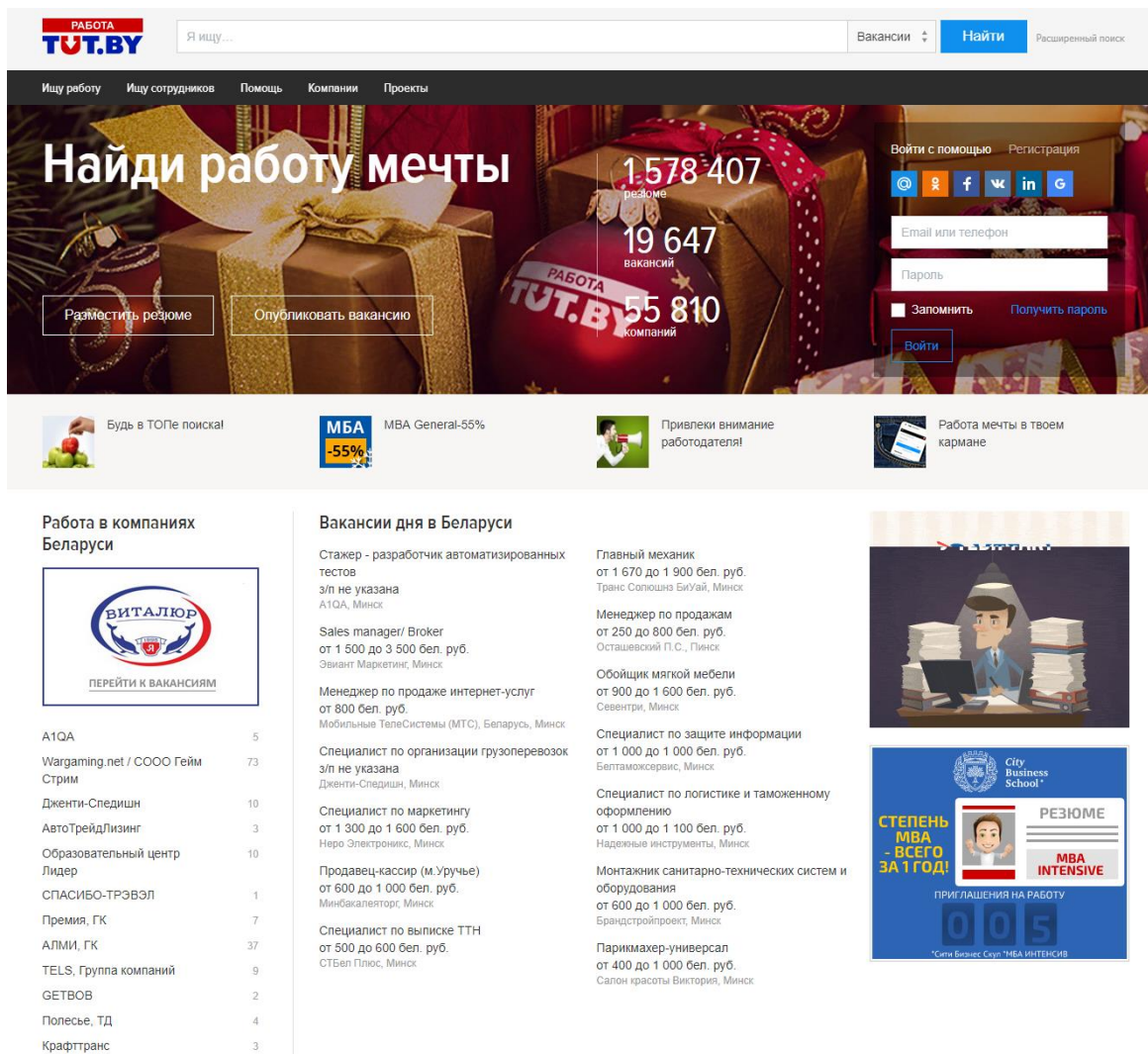


Рисунок 1.1 — Интерфейс ресурса «Работа ТУТ.БАЙ»

Разрабатываемый ресурс должен содержать базу резюме только студентов, выпускников, тем самым, предоставить конкурентные преимущества выпускникам Белорусского Государственного Университета перед другими соискателями.

Следует отметить, что данный ресурс должен содержать нормативно-правовую базу для выпускников. Исходя из этого, ни один из приведенных или аналогичных сервисов не решит возникшую задачу, а попытки адаптации могут вызвать дополнительные затраты.

1.3 Обзор и обоснование выбора инструментария разработки информационного ресурса

«CMS» — это система, которая позволяет в удобном виде создавать и управлять (редактировать, удалять) текстовыми материалами и мультимедийными документами (содержимое или контент) на сайте.

В «CMS» внутренняя структура и дизайн разделены от контента. Для администрирования веб-сайта, разработанного с использованием системы управления контентом, не нужно дополнительных знаний и технологий.

Все системы управления разделяются на платные и бесплатные. К бесплатным CMS относятся: «Joomla», «Drupal», «Wordpress». Из платных «CMS» наиболее известны: «DLE», «1С-Битрикс» которые больше подходят для развлекательных сайтов или каталогов.

На рисунке 1.2 приведен рейтинг систем управления сайтами.

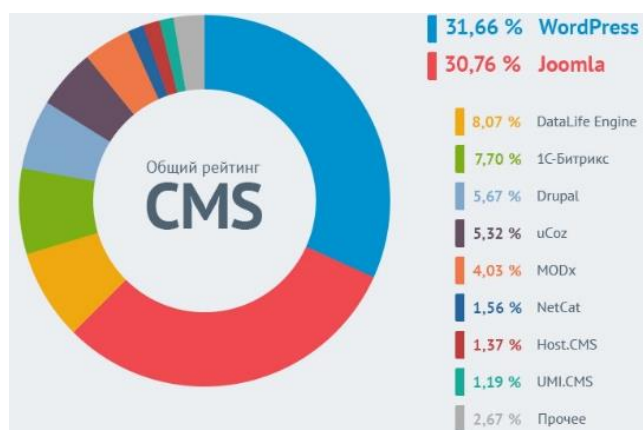


Рисунок 1.2 — Рейтинг систем управления сайтами

Для создания информационного ресурса «Центр Карьеры Белорусского Государственного Университета» использовалась «CMS Joomla», поскольку она обладает рядом преимуществ и уже внедрена на многих Интернет-ресурсах БГУ. Выбранная система управления позволит без затруднения производить администраторами актуализацию нового информационного ресурса.

Система представляет собой ядро, к которому присоединяются компоненты. Компоненты могут состоять по функциональному назначению из модуля, плагина и/или шаблона.

Модули — элементы, отображаемые на странице пользователей в виде отдельных блоков.

Плагины — элементы, отображаемые на административной панели инструментов и придающие дополнительную функциональность.

Шаблоны — элемент, отвечающий за унифицированное оформление всего информационного ресурса.

В ядро «Joomla» входят следующие составляющие:

– блок (функциональность определяется HTML-кодом внутри) и его разновидности с расширенными возможностями управления с административной панели: баннеры (модули для размещения рекламы), контакты (блок для размещения контактной информации и формы обратной связи);

– материал (статья) с метками (ключевые слова) и категориями (папками) для навигации пользователей, редактором для удобного оформления материала.

Система позволяет разработать собственное расширение посредством написания плагина на языке «PHP». Для подключения и взаимодействия расширений используется «API Joomla» [1].

Следует отметить, что «Joomla» — одна из самых популярных систем управления сайтом с открытым кодом. «CMS Joomla» позволяет добиться наилучшего качества при создании любого сайта.

Преимущества системы управления «Joomla» приведены в таблице 1.

Таблица 1 — Преимущества «Joomla»

Показатель	Описание
Стоимость	Бесплатно
Функционал	Система управления содержимым, написанная на языках «PHP» и «JavaScript», использующая в качестве хранилища базы данных «MySQL» или другие реляционные СУБД
Язык	Английский/Русский
Интерфейс пользователя	Компактный адаптивный интерфейс, не требующий специальных навыков
Системные требования	«PHP 5.x» или выше, «MySQL» 4.1.x, 5.x, «Apache 1.3.x» или выше

«Joomla» обладает следующими характеристиками:

- основана на базах данных с использованием языка «PHP» и системы управления «MySQL»;
- модуль безопасности для многоуровневой аутентификации пользователей/администраторов;
- секции новостей, продукции или услуг легко редактируемы и управляемы;
- настраиваемые схемы расположения элементов; включая левый, правый и центральный блоки меню;
- возможность загружать изображения в библиотеку для последующего использования;
- форум, опросы, голосования для эффективной обратной связи;
- работа под «Linux», «FreeBSD», «MacOSX», «AIX», «WinNT», «Win2K».

Возможности администрирования:

- возможность создавать неограниченное количество страниц;
- иерархическая организация структуры сайта;
- для каждой динамической страницы существует описание и ключевые слова в целях повышения рейтинга в поисковых системах;
- начало и окончание публикации материалов возможно запрограммировать по календарю;
- возможность ограничить доступ к определенным разделам сайта только для зарегистрированных пользователей;
- настраиваемые схемы расположения элементов, что позволяет сделать адаптивный и современный дизайн;
- дополнительные информационные модули: новости, счетчик посещений, подробная статистика посещений, гостевая книга, форум;
- изменение порядка объектов, включая новости, вопросы, статьи;
- генератор показов случайных новостей;
- модуль приёма от удалённых авторов новостей, статей и ссылок;
- иерархия объектов — количество секций, разделов, подразделов и страниц, зависит от поставленных задач;
- библиотека изображений позволяет хранить все медиа файлы для оперативного доступа;
- Возможность производить рассылку из более чем 360 служб рассылки новостей по всему миру;
- менеджер архива (перемещение старых статей в архив);
- возможность распечатать или отправить на «e-mail» любую статью;
- 3 визуальных редакторов, которые позволяют редактировать материалы в удобной для пользователя форме;
- предварительный просмотр перед окончательным размещением;

- возможность оперативной смены дизайна;
- экономное использование места на сервере за счет использования базы данных «MySQL»;
- возможность использования адресов страниц, адаптированных для лучшей индексации всеми поисковыми системами.

Таким образом, при установке данной системы на свой веб-сайт, разработчик получает полный контроль над обновлением информации и развитием сайта.

В качестве «Web-сервера» может использоваться так же «MS IIS» («Internet Information Server») версии 5.0 и выше.

«Joomla» может использоваться совместно с наиболее известными браузерами, такими как: «Google Chrome», «Microsoft Edge», «Opera» и «Mozilla Firefox», «Safari».

Для создания элементов интерфейса веб-сайта использовался «Adobe Photoshop» — многофункциональный графический редактор, работающий с растровыми изображениями.

«Adobe Photoshop» имеет большое количество автоматизированных функций, плагинов и инструментов, которые упрощают работу в программе.

Особенностью программы является то, что изображения, которые создаются или редактируются в ней состоят из слоев.

Детальный анализ предметной области и обзор существующих аналогов позволили сформировать теоретическую базу для построения функциональной модели и архитектуры разрабатываемого информационного ресурса.

Наиболее подходящим инструментом для разработки является система управления контентом «Joomla», а также СУБД «MySQL».

2 АЛГОРИТМИЧЕСКИЙ АНАЛИЗ ЗАДАЧИ. РАЗРАБОТКА ГРАФИЧЕСКОГО ИНТЕРФЕЙСА И ЛИНГВИСТИЧЕСКОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОГО РЕСУРСА

2.1 Функциональная модель информационного ресурса

Функциональная модель предназначена для описания процессов, в которых используется как естественный, так и графический языки. Для передачи информации о конкретной системе, источником языка является сама методология «IDEF0» [2].

Методология показывает, какие входные данные и каким образом влияют на формирование выходных данных, рассматривая порядок выполнения действий.

Первый уровень функциональной модели разрабатываемого ресурса является контекстная диаграмма, которая приведена на рисунке 2.1. Она представляет собой обобщенное описание работы информационного ресурса.

На общей функциональной схеме в качестве входящих данных выступают регистрационные данные пользователей. Управляющей информацией будет служить руководство пользователя, которое определяет способ работы с программой и положение о распределении выпускников.

Механизмы — администраторы, которые будут работать с ресурсом, а также программная логика, отвечающая за функционирование всего ресурса.

Выходной информацией будет являться «Отклик на вакансию», и «Приглашение на собеседование».

Декомпозиция общей функциональной схемы состоит из 4 блоков.

На первом этапе входными данными будут являться «Данные для регистрации». Итоговой информацией будет «Приглашение на собеседование».

Следует отметить, что управляющей информацией для всех блоков будет служить руководство пользователя и положение «О распределении».

После регистрации в системе наниматель должен создать компанию и тем самым, инициализировать себя в системе. Итоговой информацией на этом этапе будут являться аккаунт пользователя.

Следующим действием пользователя будет «Создание резюме». На данном этапе входящей информацией будет аккаунт, а итоговой информацией будет созданное в системе резюме.

Третьим этапом является «Отправка отклика на выбранную вакансию». Входящими данным для третьего блока является резюме сотрудника.

Этап «Пригласить на собеседование» позволяет пользователю в роле «Нанимателя» пригласить сотрудника на собеседование.

Схема декомпозиции показана на рисунке 2.2.



Рисунок 2.1 — Функциональная модель информационного ресурса



Рисунок 2.2 — Схема декомпозиции информационного ресурса

Функциональное моделирование позволило детально рассмотреть все процессы проектируемого ресурса и проанализировать все данные, которые учувствуют в работе ресурса.

2.2 Логическая модель и диаграмма прецедентов

Информационный ресурс должен осуществлять следующий функционал:

- вывод документов о распределении;
- публикация информации о вакансиях;
- публикация расширенной информации о выпускниках;
- публикация различных новостей, мероприятий, связанных с распределением и трудоустройством студентов.

Модель данных, в которой на логическом уровне описывается информационное содержание, называется логической моделью ресурса.

Логическая модель является основой для всех пользователей информационной системы [3].

Логическая модель описывает информационный ресурс, как единое целое. Логическое представление данных может существенно отличаться от общей модели данных.

Исходя из детального анализа предметной области, была разработана логическая модель информационного ресурса, изображенная на рисунке 2.3.

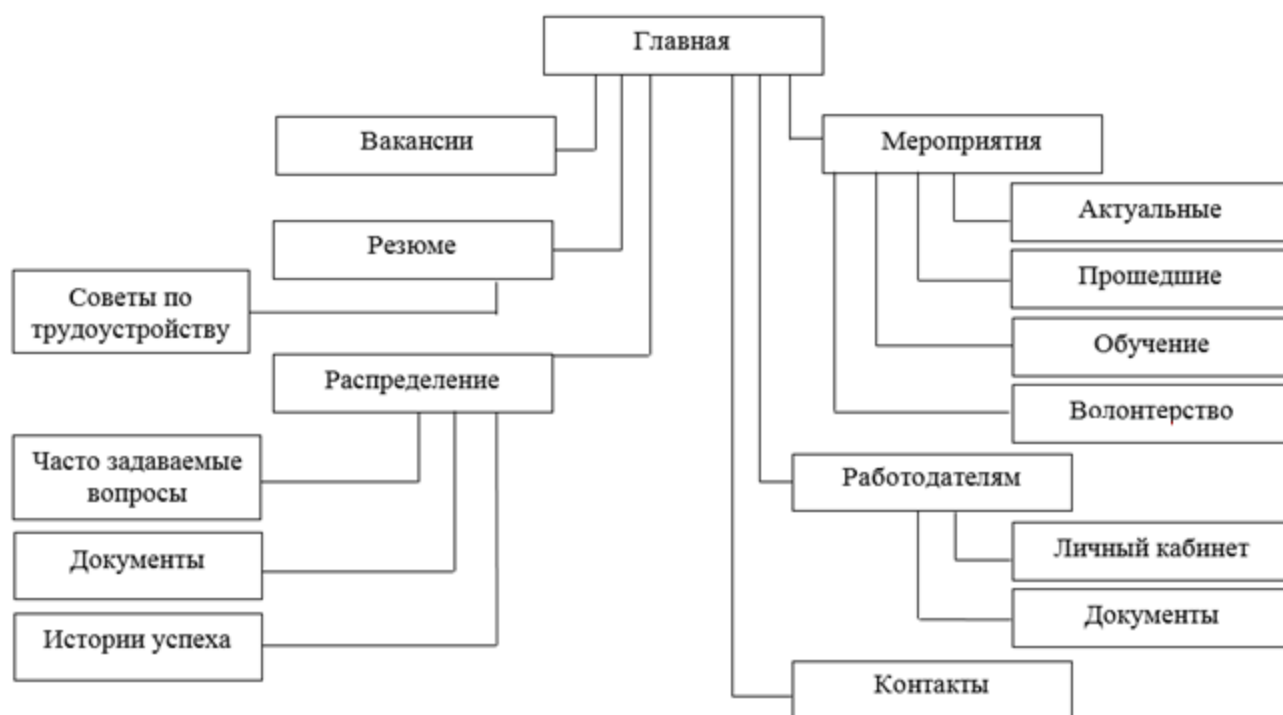


Рисунок 2.4 — Логическая модель ресурса

Одним из важных моментов является разграничение прав доступа у различных категорий пользователей. На сайте предусмотрено 4 группы пользователей: «Гость», «Соискатель», «Наниматель», «Администратор».

«Гость» при входе на сайт обладает наименьшими правами. Он может только просматривать всю информацию: статьи, фото, видео и другое.

Диаграмма прецедентов показана на рисунке 2.4.

«Соискать» наследует большинство прав от группы «Гость», но имеет возможность публиковать свои резюме, просматривать вакансии, редактировать созданные им публикации.

«Наниматель» наследует большинство прав от группы «Гость», но имеет возможность публиковать свои вакансии, просматривать резюме, редактировать созданные им публикации, которые ему будут делегированы администратором.

«Администратор» может добавлять новости, весь контент, редактировать шаблон, отключать добавлять модули, создавать дополнительное меню. Следует отметить, что администратор должен иметь возможность проверять созданные нанимателями вакансии.

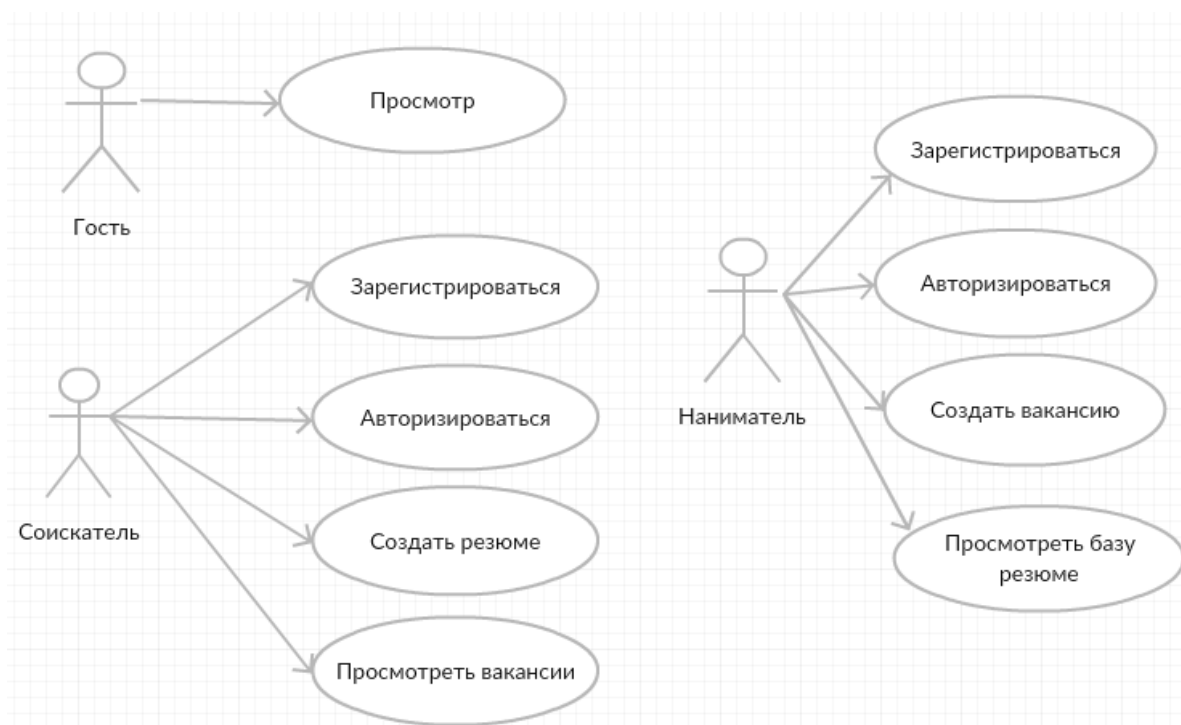


Рисунок 2.4 — Диаграмма прецедентов

В данном разделе была приведена и описана общая функциональная модель сайта, а также диаграмма прецедентов.

2.3 Графический интерфейс информационного ресурса

Графический интерфейс пользователя — разновидность пользовательского интерфейса, в котором элементы интерфейса (меню, кнопки, значки), представлены пользователю на экране в виде графических изображений.

Интерфейс пользователя — это элементы и компоненты веб-сайт, которые способны оказывать влияние на взаимодействие пользователя с программным обеспечением [4].

Исходя из вышеперечисленных определений, были выделены требования к составу графического интерфейса информационного ресурса «Центр карьеры Белорусского Государственного Университета».

Таким образом, требования к проектированию и построению интерфейса пользователя следующие:

- интерфейс должен быть интуитивно понятен простому пользователю;
- цвета интерфейса не должны раздражительно влиять на пользователя и соответствовать общей стилистике ресурсов БГУ.

Первым этапом проектирования дизайна сайта является разработка цветового и графического решения ресурса, расположения его основных элементов. На данном этапе были проанализированы цели создания ресурса, собственный стиль сайта, целевую аудиторию и на основе полученной информации разработано оформление страниц.

Работа с ресурсом начинается с посещения главной страницы, она является лицом ресурса. Дизайн сайта начинается с разработки концепции оформления главной страницы, т.к. все страницы ресурса оформляются в едином стиле, что обеспечивает удобство в навигации по сайту и легкое восприятие информации.

На рисунке 2.5 представлена схема графического интерфейса главной страницы информационного ресурса.

Для повышения уровня практичности, проектируемого графического пользовательского интерфейса, а также оптимизации временных затрат на разработку был применен набор навигационных стандартов проектирования графических интерфейсов.

Полученные решения позволили реализовать необходимые средства выполнения требуемых функций.

Информационный ресурс предназначен для информационного обеспечения, а также он не должен иметь сложную систему управления. Учитывая данную специфику следует, что сайт должен управляться с помощью меню. Каждый пункт главного меню содержит свою функцию.

Характерной особенностью интерфейса является то, что диалог с пользователем ведется не с помощью команд, а с помощью графических образов — меню, окон и других элементов.

Следующим этапом в разработке комплексного дизайна информационного ресурса является оформление внутренних страниц. Вспомогательные страницы должны быть интересны для посетителя, иметь оригинальный внешний вид, но сохранять визуальную целостность всего ресурса.

Внутренние страницы сайта несут основную информационную нагрузку. Удобный дизайн внутренних страниц ресурса должен помочь посетителю найти и усвоить нужную информацию.

Требования, которые устанавливают выбор необходимой цветовой схемы для проектируемого информационного ресурса:

- содержимое должно быть читаемым, а цвета контрастные;
- для привлечения посетителя следует применять интенсивные цвета.

На основании требований было решено использовать сочетание и оттенки синего цвета. В качестве дополнительного цвета подсветки элементов меню и других информационных блоков был выбран голубой цвет.

1			
2	3		
4			
5			
6			
7			
8			
9			
10			
11			
12			

Рисунок 2.5 — Схема главной страницы графического интерфейса:

1 — информационный блок; 2 — логотип; 3 — область меню;

4 — Слайд-шоу; 5 — быстрые ссылки; 6 — новости;

7 — информационный блок; 8 — блок «Лучшие выпускники»;

9-11 — информационные блоки;

12 — подвал сайта

Создаваемый интерфейс должен максимально реализовывать возможности информационного ресурса, но не перегружать пользователя обилием меню, кнопок, изображений, вкладок. Пользователь не должен искать тот или иной пункт меню, кнопку.

2.4 Проектирование лингвистического обеспечения

Лингвистическое обеспечение — совокупность языковых средств, используемых в системах, а также правил формализации естественного языка в целях повышения эффективности машинной обработки информации [4].

Одной из самых важных составляющих лингвистического обеспечения является интерфейс пользователя. В информационном ресурсе «Центр карьеры БГУ» он спроектирован таким образом, чтобы обеспечить пользователю наиболее простую схему работы.

Управление системой осуществляется через главное меню, которое расположено в верхней части сайта. Главное меню спроектировано таким образом, что содержит все необходимые ссылки для быстрого и беспрепятственного перехода к нужной странице.

Пользователи, которые не авторизованы в системе, но при этом пытаются открыть страницу, перенаправляются на форму авторизации, где происходит запрос логина и пароля. Механизм авторизации проверяет введенные пользователем данные на корректность. Если данные не соответствуют установленным требованиям, то срабатывает механизм валидации и пользователю выводятся соответствующие сообщения, а также подсвечиваются поля, в которых допущены ошибки. Если пользователь не зарегистрирован, то система автоматически предложит зарегистрироваться в ней.

Форма регистрации также оснащена механизмом валидации, который проверяет на корректность следующие поля:

- логин;
- пароль (не менее 6 символов);
- повторный ввод пароля;
- адрес электронной почты.

После прохождения процедуры проверки пользователь автоматически авторизуется в системе и перенаправляется на главную страницу.

Сообщение системы показано на рисунке 2.6.

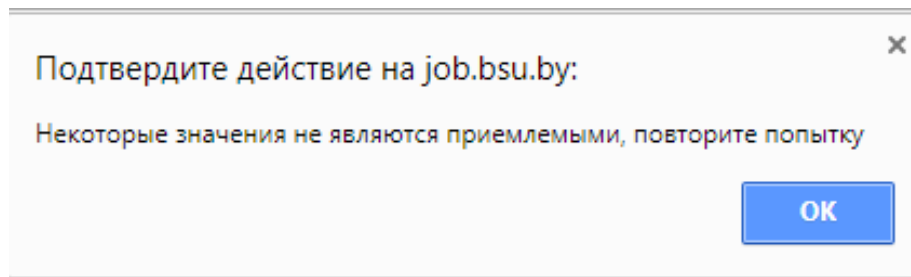


Рисунок 2.6 — Сообщение системы

Механизм валидации ввода приведен на рисунке 2.7.

A screenshot of a registration form. It contains several input fields: a name field with "eg", a password field with "lglg", a "Тема" (Topic) field, a CAPTCHA field with "3 + 4 = ?", and a "Сообщение" (Message) text area. A red border highlights the password field. A tooltip with an error icon and text is displayed over the password field: "Адрес электронной почты должен содержать символ '@'. В адресе 'jgjj' отсутствует символ '@'." At the bottom left is a dark button with a paper plane icon and the text "Отправить".

Рисунок 2.7 — Работа механизма валидации

В данном разделе были детально описаны и приведены функциональные схемы разрабатываемого приложения в нотации «IDEF0» и «DFD».

А также было описано лингвистическое обеспечение и механизмы взаимодействия интернет-приложения с пользователями.

3 ОПИСАНИЕ ТЕХНОЛОГИИ РЕАЛИЗАЦИИ ИНФОРМАЦИОННОГО РЕСУРСА

3.1 Подготовка к установке

«CMS Joomla» — это система управления сайтом общего назначения.

Требования, предъявляемые для установки «Joomla» к хостингу следующие:

- «Apache 1.3»;
- «MySQL 5.0.15»;
- «PHP 5.3.1».

Веб-хостинг — это услуга, суть которой заключается в размещении сайта на специальном сервере, который подключен к сети Интернет. Т.е. компания, предоставляющая эту услугу, сдает в аренду пространство и вычислительные мощности на сервере, который постоянно подключен к сети Интернет [5].

К веб-серверу предъявляются следующие требования:

- сервер должен работать в режиме высокой доступности, с установленной системой обработки отказов;
- сервер должен быть подключен к сети Интернет через канал с высокой пропускной способностью;
- сервер должен иметь резервный канал доступа в Интернет с возможностью балансировки нагрузки.

Для создания информационного ресурса «Центр карьеры БГУ» был использован выделенный сервер БГУ под управлением «ISPconfig». Конфигурация сервера следующая:

- операционная система «Linux Debian»;
- версия «PHP» 5.4.16;
- «MySQLi» версии «5.5.52-MariaDB».

3.2 Настройка сервера и дополнительных компонентов

Т.к. был использован хостинг Белорусского Государственного университета, то для начала установки в панели управления была создана база данных и пользователь, который обладает всеми необходимыми привилегиями.

Система управления хостингом «ISPConfig» позволяет создать базу данных «MySQL» и пользователя, а также настроить все необходимые привилегии для корректной установки системы [6].

На рисунке 3.1 показана схема создания базы данных средствами панели управления хостингом «ISPConfig».

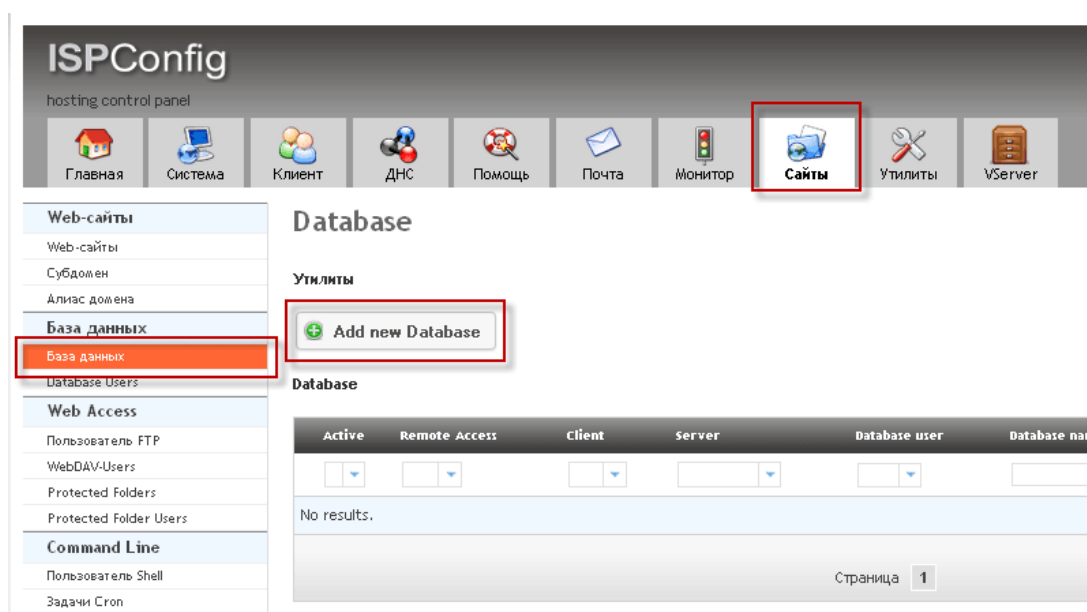


Рисунок 3.1 — Создание базы данных

Для корректной установки был создан владелец (пользователь) базы данных. Следует отметить, что при установке будут созданы основные таблицы, в которых будут храниться данные о системе, пользователях, материалах, категориях.

Создание пользователя показано на рисунке 3.2.

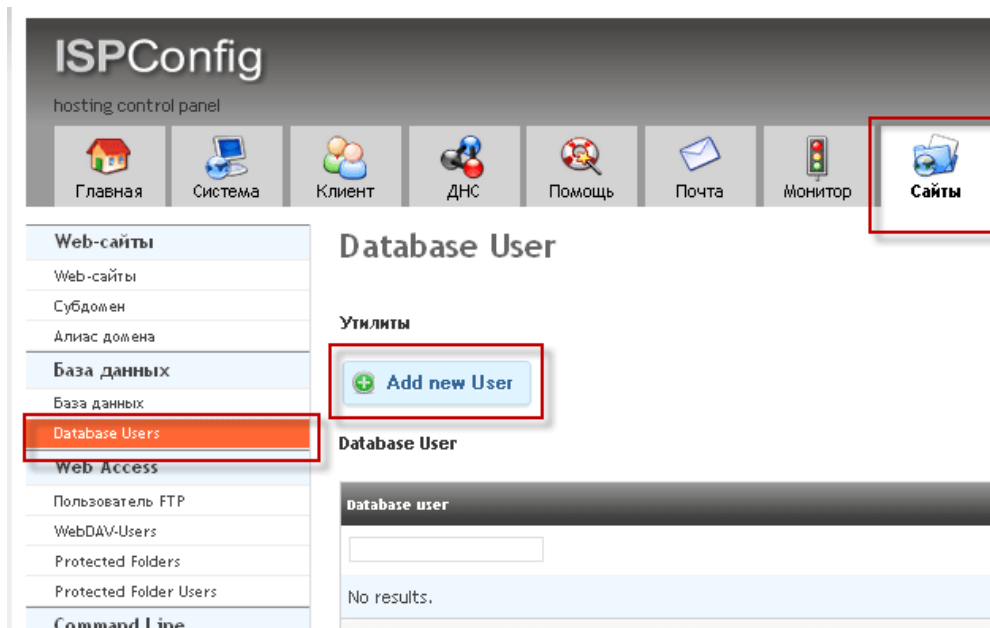


Рисунок 3.2 — Создание пользователя базы данных

Для загрузки файлов «CMS «Joomla» на сайт был создан пользователь «FTP» с правами на запись файлов.

Создание пользователя показано на рисунке 3.3.

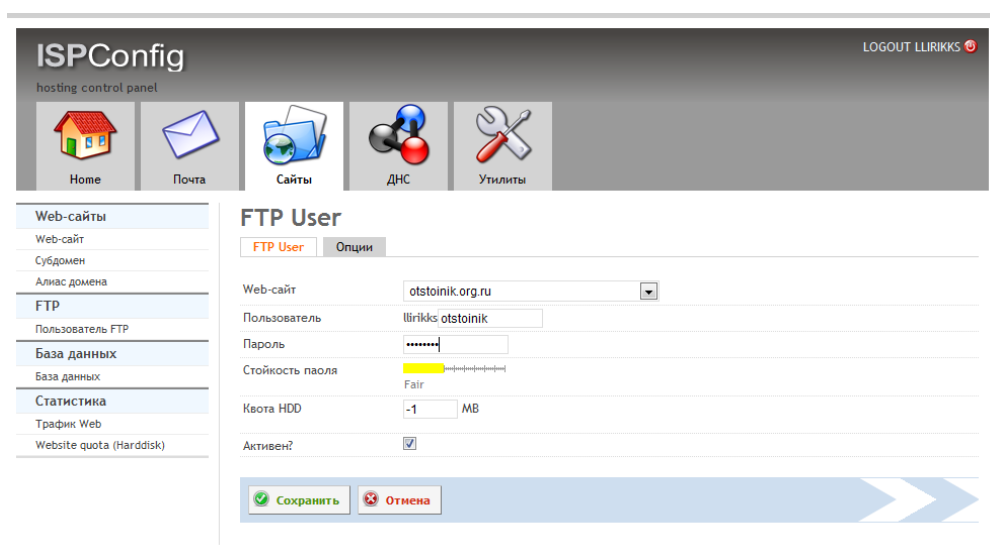


Рисунок 3.3 — Создание пользователя «FTP»

Загрузка файлов системы управления сайтом на сервер может осуществляться несколькими способами:

- загрузка файлов напрямую через «FTP-менеджер» или браузер;
- загрузка на сервер архива, а затем запуск специального скрипта на сервере, или распаковка через панель управления хостингом [7].

Т.к. хостинг управляется «ISPConfig», то для экономии времени был применён 2 способ. Загрузка архива с файлами показана на рисунке 3.4.

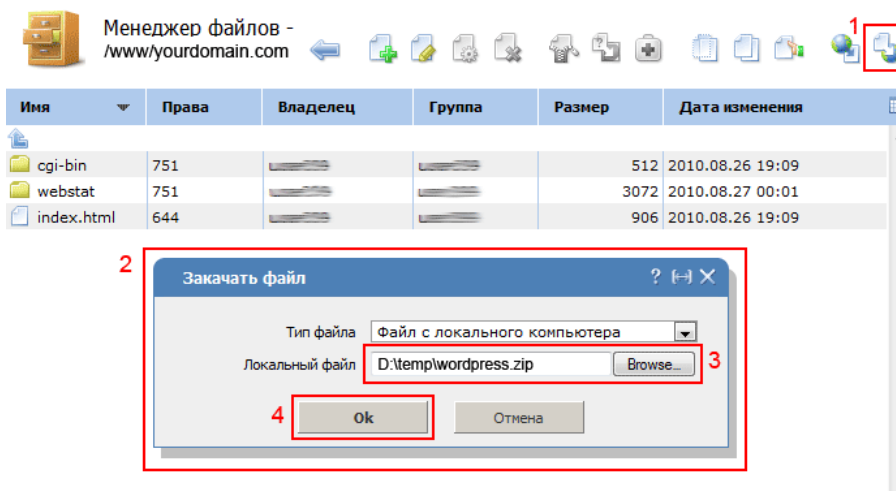


Рисунок 3.4 — Загрузка установочного архива

Для распаковки загруженного архива в «ISPConfig» необходимо выделить нужный архив и выбрать пункт «Извлечь». Процесс распаковки показан на рисунке 3.5.

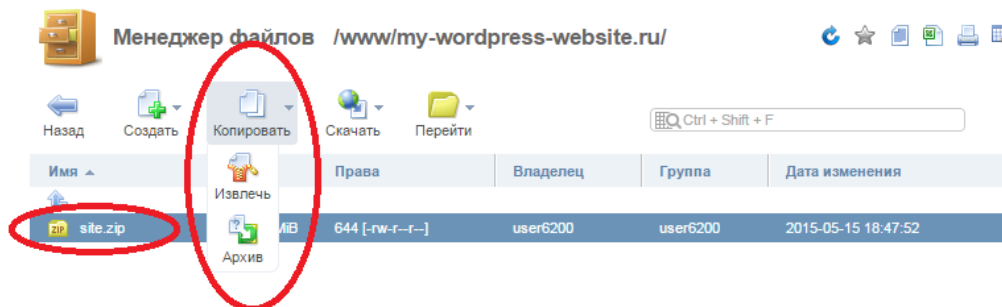
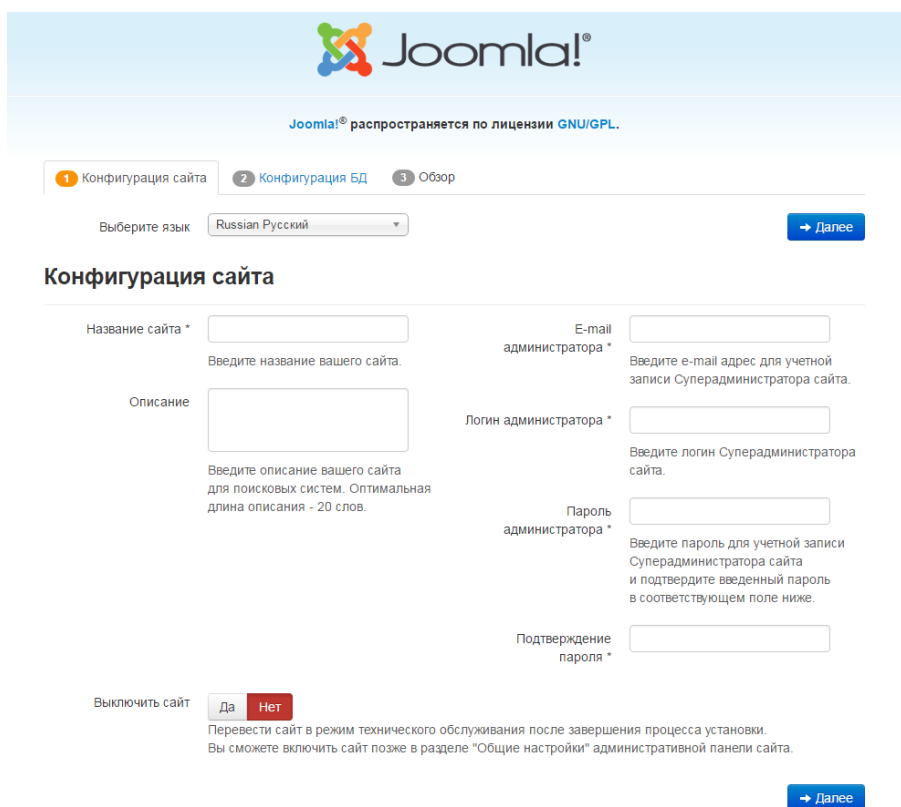


Рисунок 3.5 — Распаковка архива

3.3 Установка системы управления контентом «Joomla»

Для установки системы управления контентом «Joomla» необходимо запустить браузер и ввести в адресной строке название нашего сайта (job.bsu.by), после чего запуститься установка Joomla. На рисунке 3.6 показано диалоговое окно «Конфигурация сайта» для настройки «Joomla».



The screenshot shows the Joomla! installation configuration interface. At the top, the Joomla! logo and the text "Joomla!® распространяется по лицензии GNU/GPL." are visible. Below this, there are three steps: "1 Конфигурация сайта", "2 Конфигурация БД", and "3 Обзор". A language selection dropdown is set to "Russian Русский", and a "Далее" button is present. The main section is titled "Конфигурация сайта" and contains several input fields: "Название сайта *", "Описание", "E-mail администратора *", "Логин администратора *", "Пароль администратора *", and "Подтверждение пароля *". Each field has a corresponding instruction. At the bottom, there is a "Выключить сайт" section with "Да" and "Нет" radio buttons, and a "Далее" button.

Рисунок 3.6 — Окно для настройки «Joomla»

Для установки были введены следующие данные «Конфигурации сайта»:

- название сайта — Центр карьеры БГУ;
- описание — Центр карьеры БГУ;
- e-mail — указывается существующий рабочий электронный адрес;
- логин администратора — придумываем имя;
- пароль — придумываем пароль.

После ввода всех требуемых данных, необходимо нажать кнопку «Далее». После проверки корректности введенных данных откроется окно «Конфигурация базы данных» (Рисунок 3.7).

1 Конфигурация сайта 2 Конфигурация БД 3 Обзор

Конфигурация базы данных

← Назад → Далее

Тип базы данных *
Это обычно "MySQL".

Имя сервера базы данных *
Это обычно "localhost".

Имя пользователя *
Введите имя пользователя базы данных, выданное хостером. На локальном сервере обычно используется учетная запись "root" без пароля.

Пароль
Введите пароль пользователя базы данных. Не рекомендуется применять учетную запись без пароля.

Имя базы данных *
На некоторых хостингах присутствует ограничение по количеству используемых баз данных. Использование префиксов таблиц позволяет установить несколько сайтов на Joomla! в одну базу данных.

Префикс таблиц *
Укажите префикс таблиц или используйте **автоматически сгенерированный**. Рекомендуемая длина префикса: 3-4 символа (латинские буквы и цифры) и символ подчеркивания в конце. Убедитесь, что выбранный префикс не используется в именах существующих таблиц базы данных.

Действия с уже имеющимися таблицами *
Существующая резервная копия таблиц от предыдущей установки Joomla! будет заменена.

← Назад → Далее

Рисунок 3.7 — Конфигурация базы данных

Для конфигурации базы данных необходимо ввести следующие данные:

- имя сервера БД — localhost;
- имя пользователя – root;
- пароль;
- имя БД.

Все остальные поля необходимо оставить по умолчанию. Если данные были введены верно, то система установки перенаправит на страницу проверки введенных данных. Окно «Обзор введенных данных» показано на рисунке 3.8.

Следует отметить, что на данном шаге существует возможность возврата на предыдущие этапы установки и отредактировать необходимые поля. Если все данные введены корректно, то необходимо нажать кнопку «Установка».

Для корректной установки всех дополнительных компонентов необходимо установить «Демо-данные».

1 Конфигурация сайта **2** Конфигурация БД **3** Обзор

← Назад → Установка

Завершение установки

Установка демо-данных Нет

- Блог English (GB) демо-данные
- Визитка English (GB) демо-данные
- Стандартные English (GB) демо-данные
- Изучаем Joomla! English (GB) демо-данные
- Тестовые English (GB) демо-данные

Начинающим пользователям рекомендуется установить демо-данные, они помогут легче освоить основные функции системы.

Обзор

Отправить конфигурацию сайта на e-mail Да Нет

Позволяет после завершения установки отправить параметры конфигурации на e-mail: `dr.anastezia@yandex.by`

Название сайта	test.by
Описание	информационный сайт
Выключить сайт	Нет
E-mail администратора	dr.anastezia@yandex.by
Логин администратора	nasty
Пароль администратора	***

Тип базы данных	mysql
Имя сервера базы данных	localhost
Имя пользователя	root
Пароль	
Имя базы данных	test
Префикс таблиц	gr3ed_
Действия с уже имеющимися таблицами	Переименовать

Рисунок 3.8 — Окно проверки введенных данных перед установкой

После успешной установки «Joomla», для корректной работы всей системы, нужно удалить папку «installation». Процесс удаления директории установки показан на рисунке 3.9. В версии 3.8.3 существует возможность удалить инсталляционные файлы с веб-страницы.



Рисунок 3.9 — Удаление директории «Installation»

На данном этапе установка завершена. Для входа в административную панель необходимо в адресной строке браузера ввести: «*http://job.bsy.by/administrator*», затем ввести логин и пароль, который был указан на странице конфигурации.

3.4 Разработка информационного ресурса

Главными задачами при разработке любого веб-сайта являются: организация структуры сайта и определение его информационного наполнения. На первом этапе была создана информационная модель веб-сайта.

Реализация — это работа по созданию сайта. На этом этапе проводится подготовка текстового и графического материала (печать, сканирование).

Материал разбивается по файлам в соответствии со структурой. Организуются ссылки между файлами сайта.

Основное требование к содержимому веб-сайта — полнота и достоверность.

Структура сайта — это логическая разметка и физическая связка страниц сайта, а также расположение видимых элементов дизайна, обусловленная стандартами разработки сайтов. Разделяют внешнюю и внутреннюю структуру.

Внешняя структура включает в себя расположение видимых блоков на сайте (шапка, логотип, меню, контент). Разработка внешней структуры тесно связана с техническим созданием.

Внутренняя структура включает в себя принадлежность материалов к определенным категориям, а категорий к разделам, и ссылочную связку страниц. Такую схему называют логической структурой.

Логотип сайта был разработан новый. Это связано с разработкой новой цветной гаммы сайта. Логотип сайта показан на рисунке 3.10.

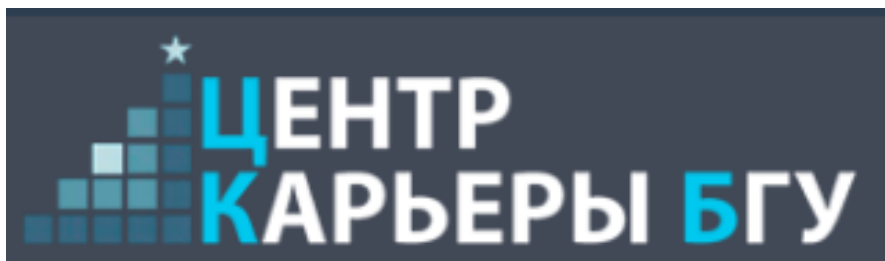


Рисунок 3.10 — Логотип сайта

Управлением и мониторингом сайта занимается администратор системы, который обладает всеми доступными правами на внесение изменений, как в содержимое, так и в структуру системы.

На рисунке 3.11 представлен инструмент системы управления контентом "Панель управления". Используя данное средство, администратор может

производить мониторинг основных событий, происходящих в системе, а также производить некоторые действия, минуя другие инструменты системы.

При помощи «Панели управления» администратор может выполнять следующие действия:

- производить мониторинг количества записей, страниц;
- изменять тему оформления системы;
- производить обновление системы управления контентом «Joomla»;
- отслеживать наличие новой технической документации о системе управления контентом «Joomla»;
- производить быструю публикацию новостей.

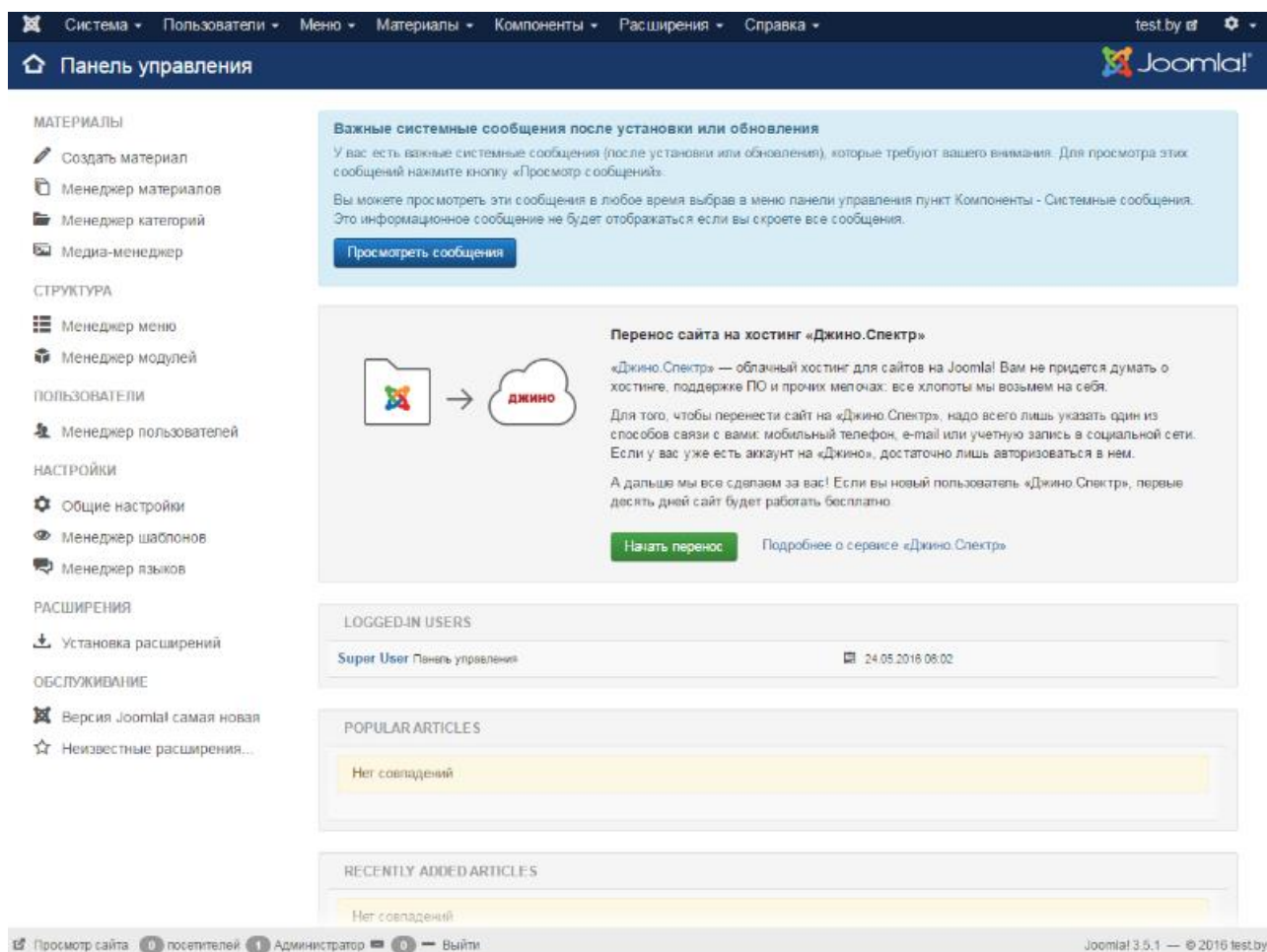


Рисунок 3.11 — Панель управления администратора

Для корректного отображения ресурса в поисковых системах и создания понятных ссылок в разделе «Настройки «SEO» были активированы следующие системные функции:

- включена функция SEF;
- включена функция перенаправление;
- включена функция добавления суффикса к ссылке.

Следует отметить, что для активации понятых ссылок на веб-сервере «Apache» был активирован режим «Mod_rewrite». Для активации режима необходимо переименовать файл «htaccess.txt» в «.htaccess» [8]. Настройки сайта для активации режима понятных ссылок показаны на рисунке 3.12.

Настройки SEO	
Включить SEF (ЧПУ)	<input checked="" type="checkbox"/> Да <input type="checkbox"/> Нет
Перенаправление URL	<input checked="" type="checkbox"/> Да <input type="checkbox"/> Нет
Добавлять суффикс к URL	<input checked="" type="checkbox"/> Да <input type="checkbox"/> Нет
Алиасы в Unicode	<input type="checkbox"/> Да <input checked="" type="checkbox"/> Нет
Включать название сайта	<input type="text" value="До"/> ▾

Рисунок 3.12 — Настройка ссылок

Для подключения созданного шаблона оформления был использован раздел системы управления «Управление шаблонами». Созданный шаблон работает под управлением системной платформы «Helix». Все необходимые настройки производились через панель управления данной платформы. Настройка темы оформления показана на рисунке 3.13. Код верстки приведен в приложении А.

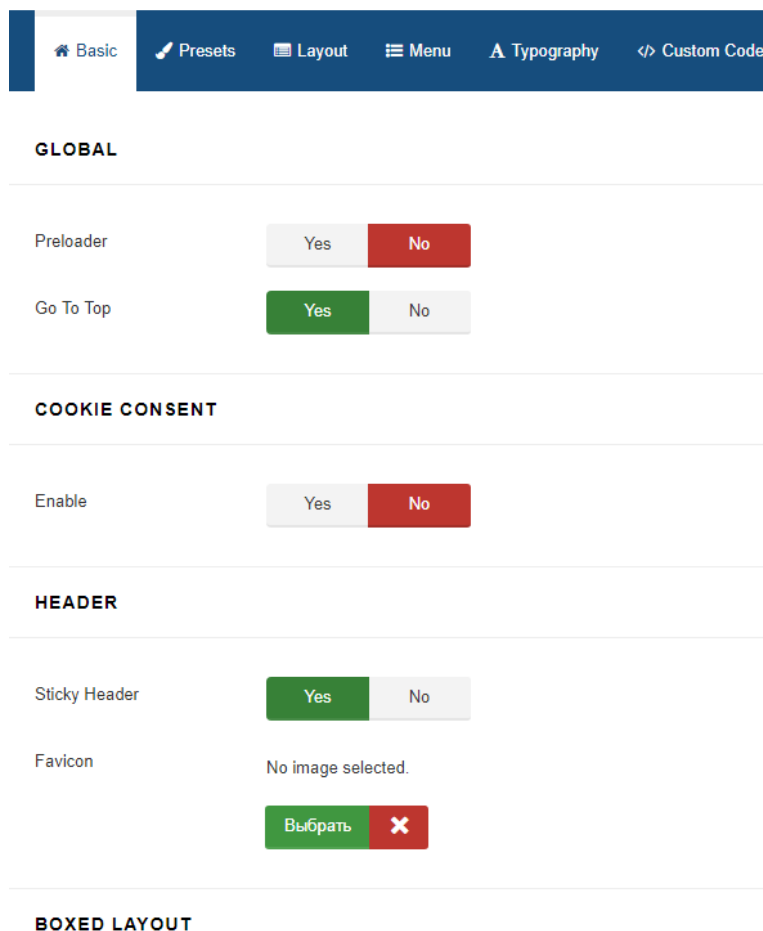


Рисунок 3.13 — Настройка темы оформления

На рисунке 3.14 представлен инструмент «Все меню», с помощью которого было создано главное меню информационного ресурса.

Для создания меню были выполнены следующие действия:

- было создано основное меню с системной меткой «Main menu»;
- был создан модуль для вывода главного меню;
- добавлены основные пункты меню.

Следует отметить, что используя данное средство, администратор может создавать новые страницы, удалять и редактировать существующие.

Редактирование страниц системы может проводиться, как в визуальном виде, так и в режиме редактирования «HTML/PHP» кода [9].

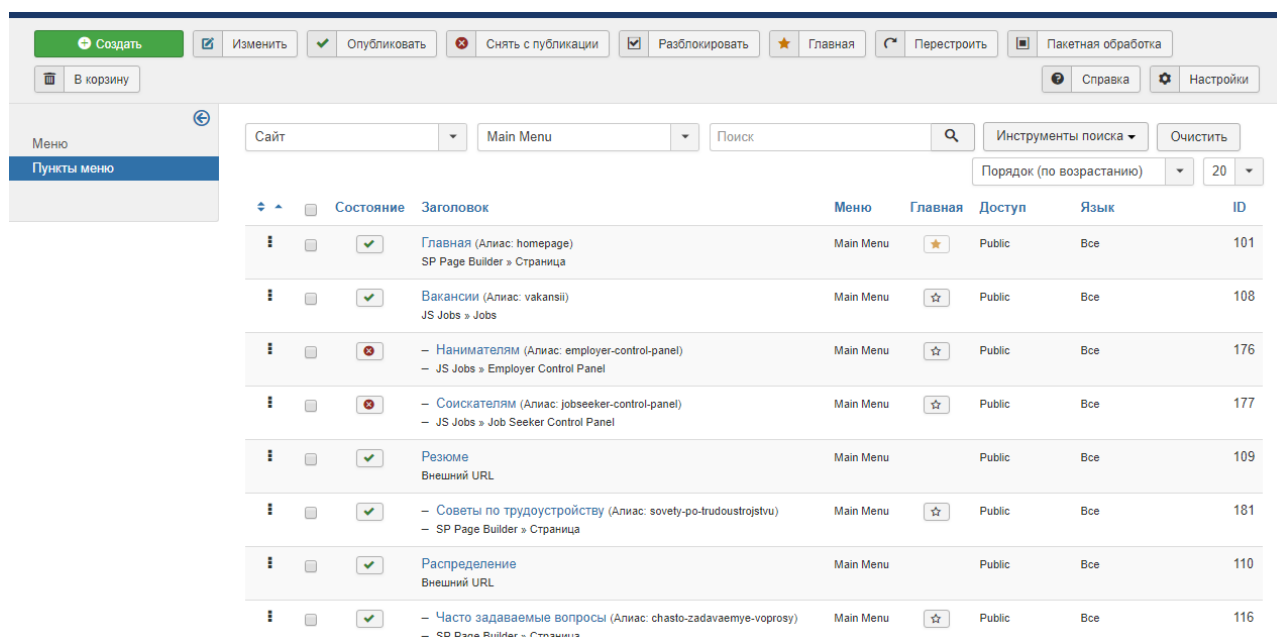


Рисунок 3.14 — Создание меню сайта

На рисунке 3.15 представлен «Менеджер материалов» — это раздел системы управления, который позволяет просмотреть все материалы, которые были загружены на сайт. В данном разделе существует возможность управления материалами, настройки порядка публикации и просмотра категории.

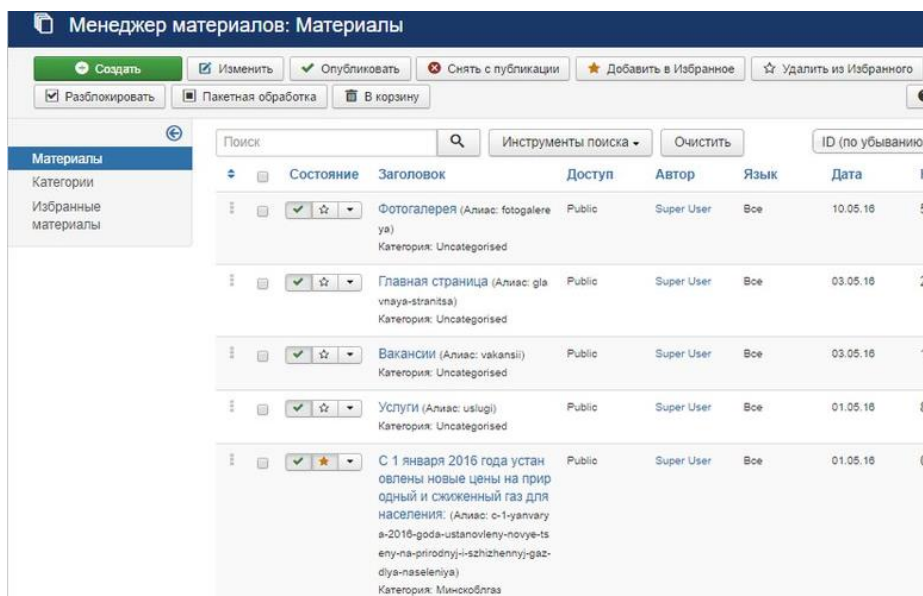


Рисунок 3.15 — Менеджер материалов

Все модели информационного ресурса были созданы с помощью инструмента «Менеджер модулей», который позволяет администратору редактировать различную информацию, касающуюся внешнего вида системы и системных компонентов страниц.

Использование «Менеджера модулей» предоставляет следующие функции:

- управление административной панелью;
- управление виджетами системы;
- управление меню системы.

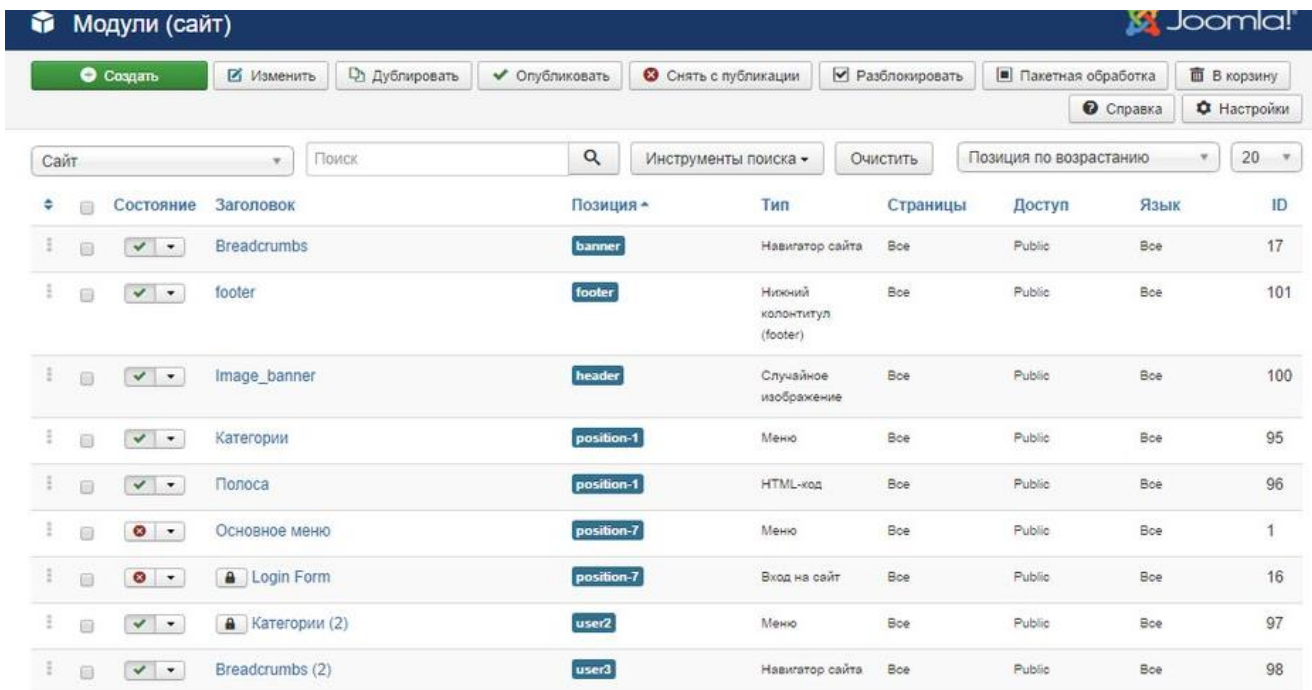


Рисунок 3.16 — Менеджер модулей

Для создания страниц и интерактивных модулей информационного ресурса было использовано системное приложение конструктора страниц «Pagebuilder».

«Pagebuilder» — это конструктор страниц, который позволяет из готовых блоков пользователю создать страницу, настраивая содержимое блоков, оформление и другие параметры, которые заложили разработчики расширения.

Компонент был установлен с помощью менеджера расширений. В корень сайта были скопированы установочные файлы и запущен процесс установки из папки временных файлов [10]. Процесс установки показан на рисунке 3.17.

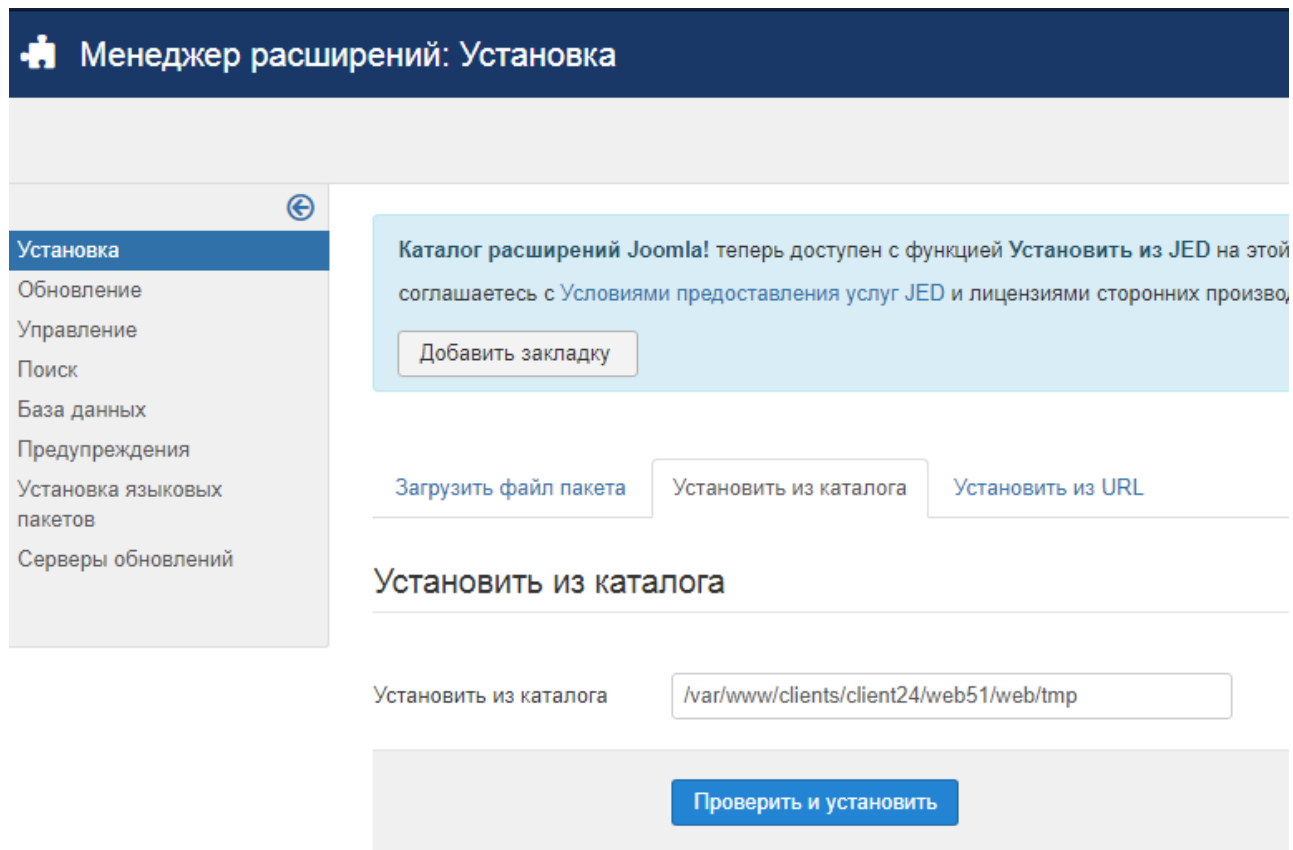


Рисунок 3.17 — Установка компонента

Для создания страницы средствами «Pagebuilder» необходимо выбрать пункт «Создать» в панели управления. После активации формы создания страницы необходимо добавить строки, указать их назначение и изменить параметры. Приложение предусматривает стандартные категории. Изменение параметров строки показано на рисунке 3.19.

Создание страницы с помощью расширения «Pagebuilder» показано на рисунке 3.18.

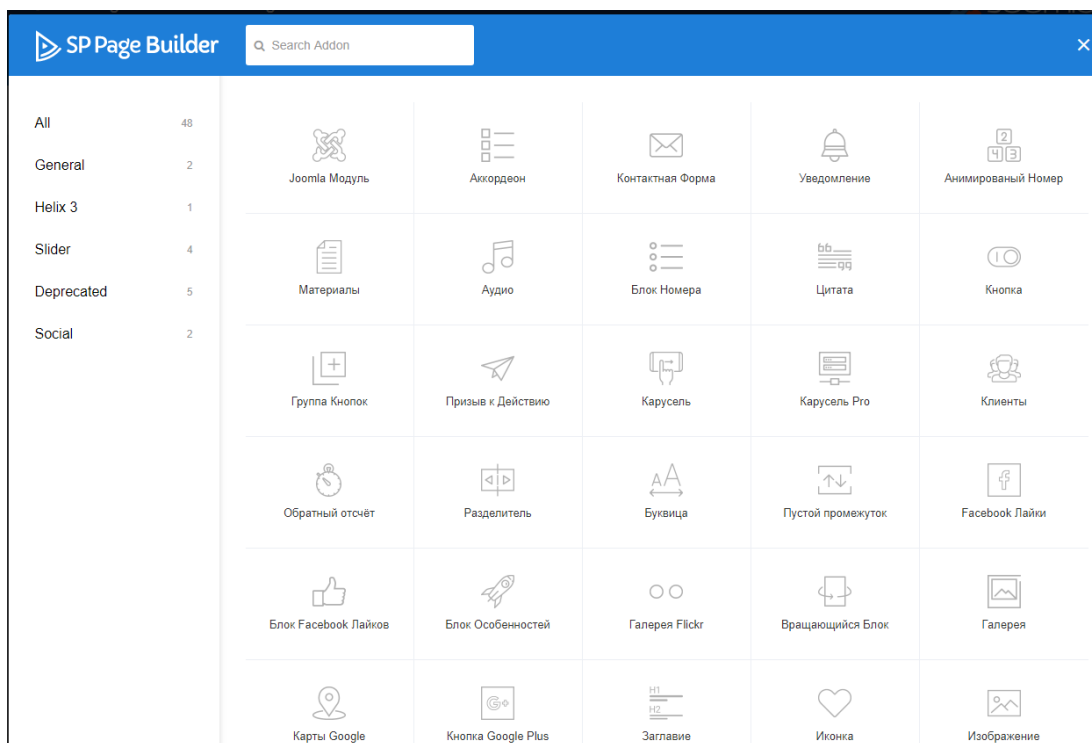


Рисунок 3.18 — Создание страницы

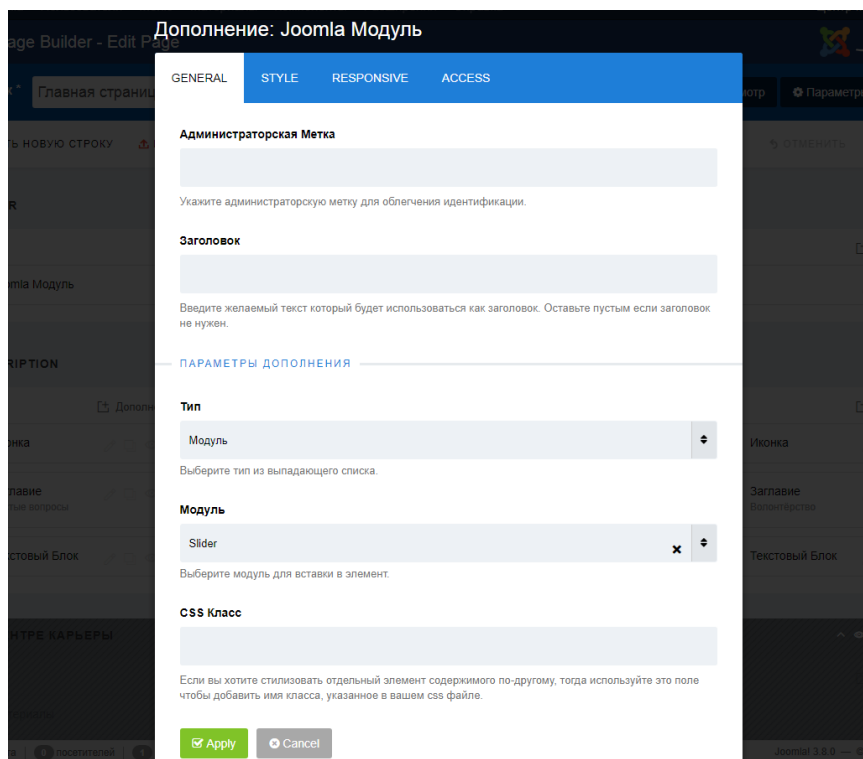


Рисунок 3.19 — Редактирование параметров строки

Для вывода созданных страниц необходимо создать новый пункт меню и указать в качестве типа меню компонент «Страница Pagebuilder». Настройка пункта меню показана на рисунке 3.20.

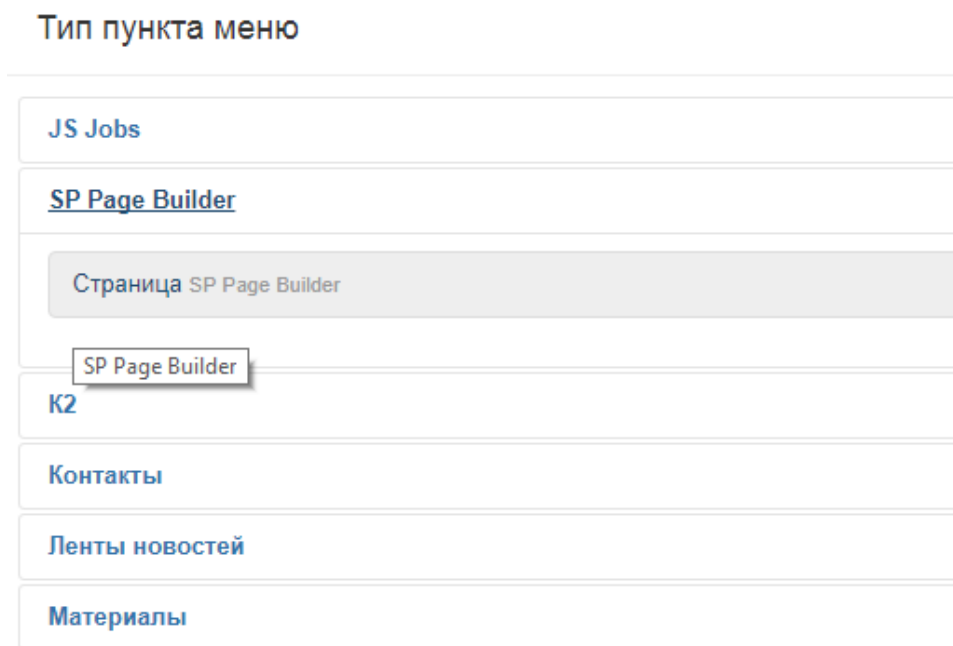


Рисунок 3.20 — Вывод страницы в меню

Для организации системы публикации вакансий и регистрации работодателей был использован и доработан системный компонент «JS JOBS».

Для публикации вакансии были созданы следующие поля:

- название вакансии;
- компания;
- тип занятости;
- уровень заработной платы;
- профессиональная область;
- уровень образования;
- опыт работы;
- описание.

Настройка полей показана на рисунке 3.21.

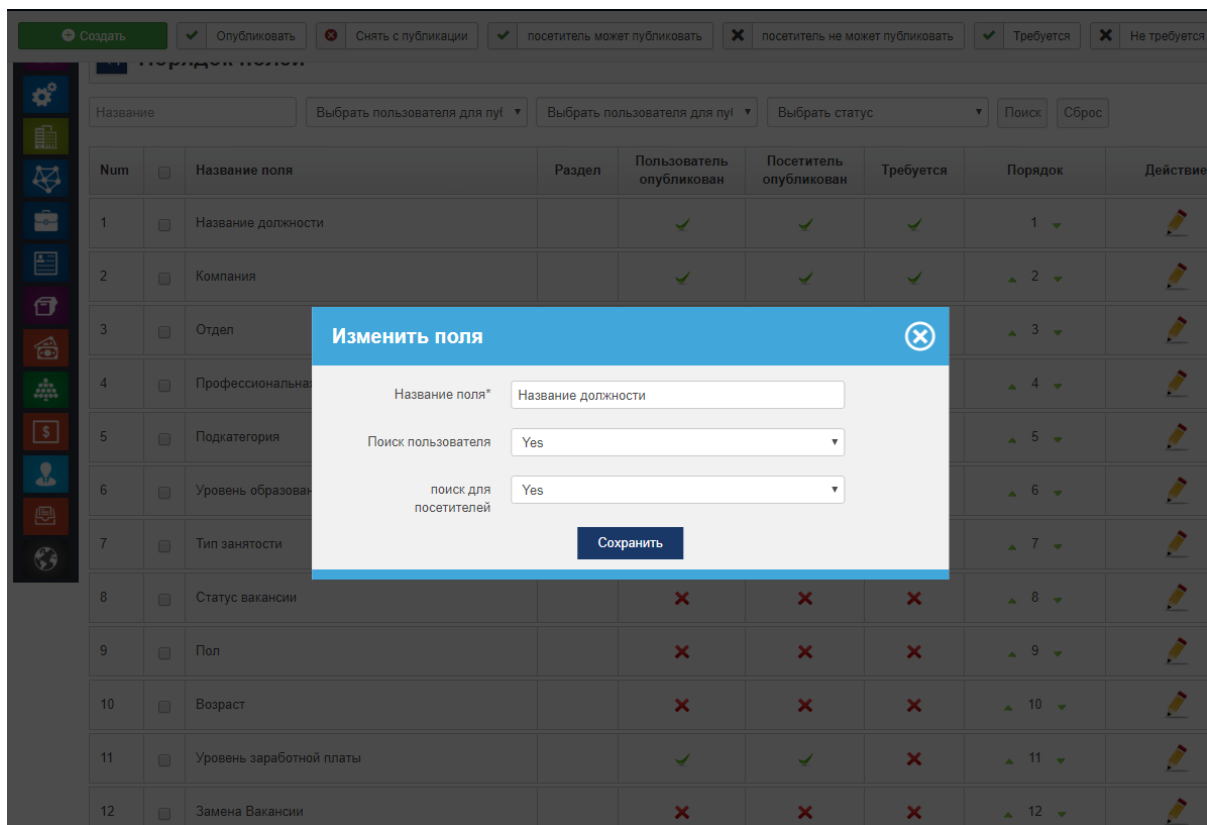


Рисунок 3.21 — Настройка полей вакансии

В данном подразделе была подробно описана программная реализация и структура веб-сайта. Для полноты описания были использованы рисунки программного кода с пояснениями. Программный код информационного ресурса приведен в приложении А.

4 АПРОБАЦИЯ ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННОГО РЕСУРСА

Апробация — это проверка на практике, в реальных условиях теоретически построенных методов.

В рамках апробации программного обеспечения был произведен вычислительный эксперимент, результатом которого должен являться реализованный информационный ресурс «Центр карьеры БГУ». Опытная эксплуатация состоит из следующих этапов:

- вход в систему;
- регистрация в системе;
- просмотр вакансий;
- создание вакансии;
- редактирование вакансии;
- отправка отклика нанимателю.

При вводе в адресной строке браузера адреса сайта «job.bsu.by», сразу же загружается главная страница сайта, которая демонстрирует работу ресурса. Главная страница приведена на рисунке 4.1.

При попытке открыть страницу сайта не авторизованному пользователю система предложит авторизоваться.

Если ранее пользователь не прошел процедуру регистрации, но при этом пытается войти — система автоматически проверит это и предложит перейти на страницу регистрации.

Следует отметить, что система также проверяет введенные данные и тут же сообщает об этом пользователю.

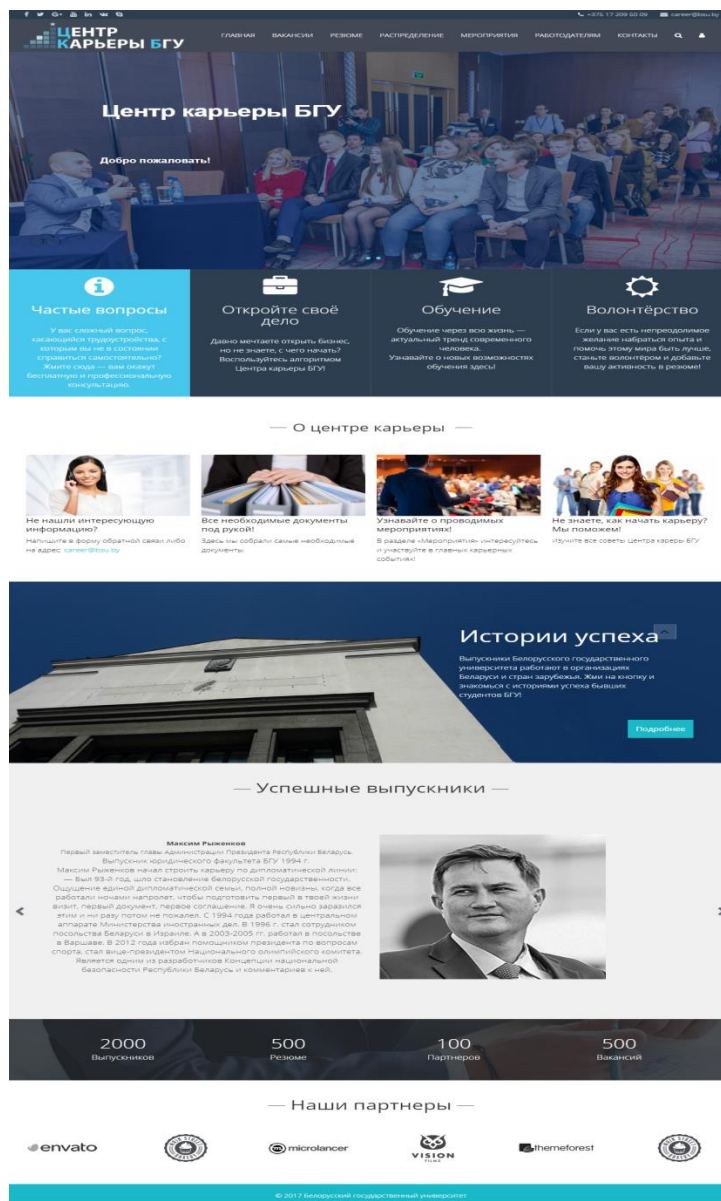


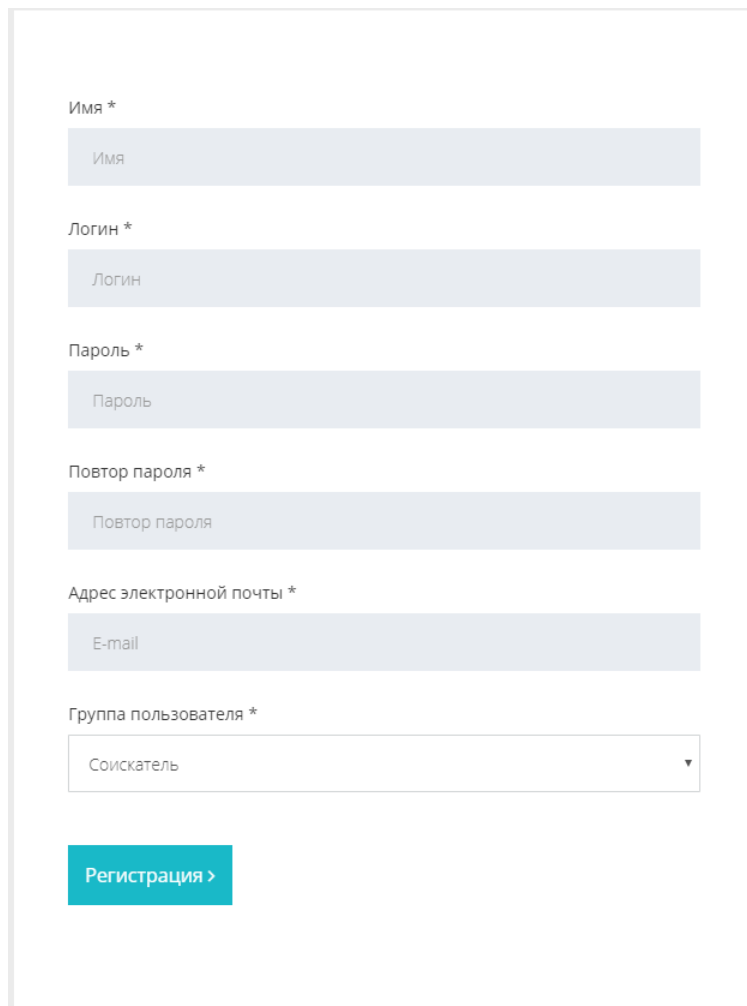
Рисунок 4.1 — Главная страница информационного ресурса «Центр карьеры БГУ»

При попытке открыть страницу сайта не авторизированному пользователю система предложить авторизоваться.

Если ранее пользователь не прошел процедуру регистрации, но при этом пытается войти — система автоматически проверит это и предложит перейти на страницу регистрации.

Следует отметить, что система также проверяет введенные данные и тут же сообщает об этом пользователю.

Страница регистрации показана на рисунке 4.2.



The image shows a registration form with the following fields and elements:

- Имя ***: Text input field with placeholder "Имя".
- Логин ***: Text input field with placeholder "Логин".
- Пароль ***: Text input field with placeholder "Пароль".
- Повтор пароля ***: Text input field with placeholder "Повтор пароля".
- Адрес электронной почты ***: Text input field with placeholder "E-mail".
- Группа пользователя ***: Dropdown menu with "Соискатель" selected.
- Регистрация >**: A teal button with a right-pointing arrow.

Рисунок 4.2 — Создание нового пользователя в системе

При загрузке страницы «Вакансии» пользователю выводится список возможных вакансий. Выбрав необходимую вакансию — пользователю отобразится расширенная информация. Перейдя по ссылке вакансии, пользователь может просмотреть ее описание, перейти к профилю работодателя или направить отклик.

Страница просмотра вакансий приведена на рисунке 4.3.

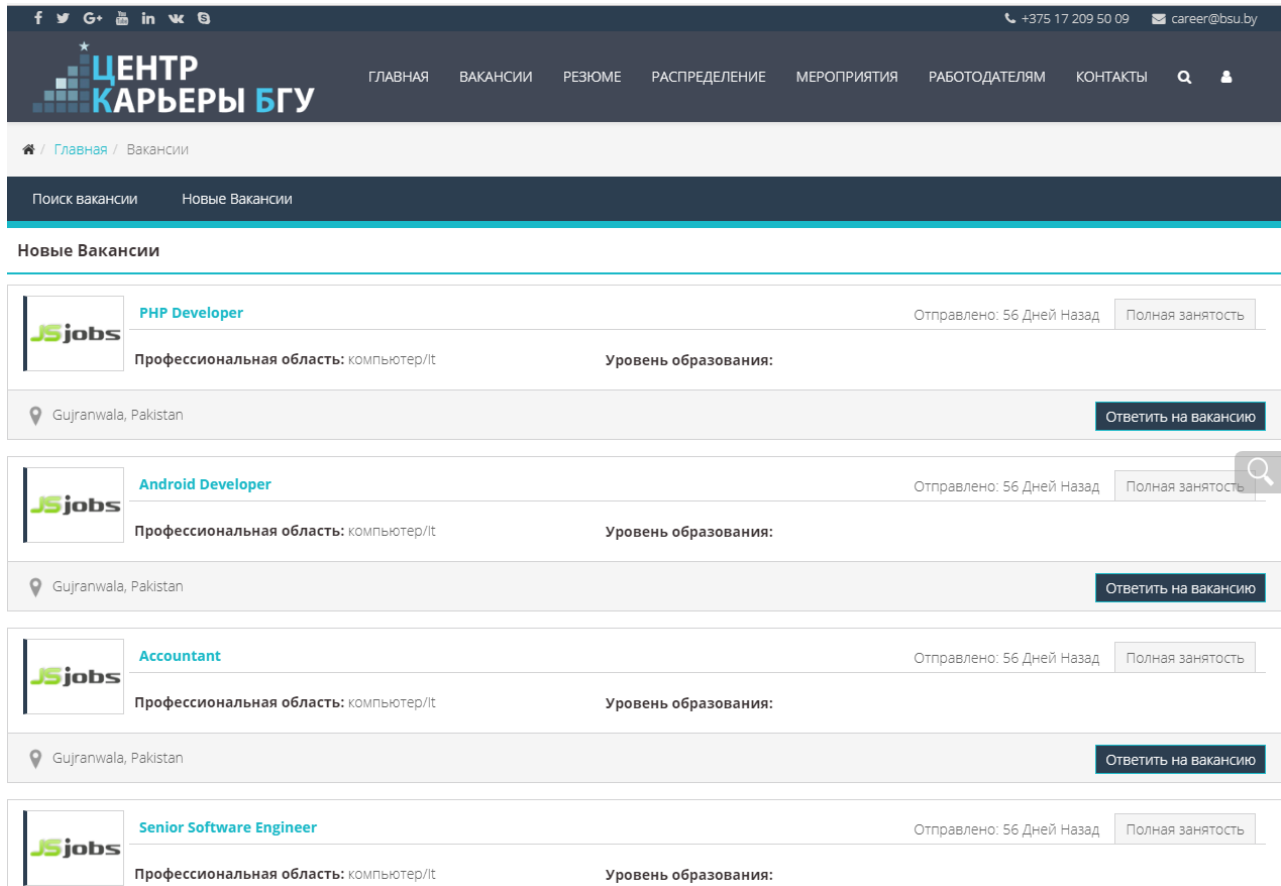


Рисунок 4.3 — Страница «Вакансии»

Создание вакансии состоит из следующих этапов:

- создание компании;
- создание вакансии и ее активация.

На этапе создания компании пользователь указывает информацию о компании и загружает в систему ее логотип.

На этапе создания вакансии пользователю необходимо указать название вакансии и выбрать категорию.

Процесс создания вакансии показано на рисунке 4.4.

Новая вакансия | Мои вакансии | Мои компании | Поиск резюме

Информация о вакансии

Название должности *

Компания *

Профессиональная область *

Тип занятости *

Уровень заработной платы

Уровень образования *

Рисунок 4.4 — Создание вакансии

Для редактирования созданной вакансии необходимо зайти в раздел «Мои вакансии», где нужно выбрать соответствующий пункт. Раздел «Мои вакансии» и форма редактирование показана на рисунке 4.5

Новая вакансия | Мои Вакансии | Мои Компании | Поиск резюме

Мои Вакансии

Разместить вакансию





Название	Категория	Занятость	Статус вакансии	Компания	Создано
 455				Компания: 01212	18-11-2017 Полная занятость
Профессиональная область : компьютер/it		Уровень образования:			
					   Резюме (1)

Рисунок 4.5 — Просмотр журналов

В данном разделе была детально описана процедура апробации информационного ресурса, а также произведена проверка работоспособности всех основных функций.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был представлен логически завершённый информационный ресурс.

В данной дипломной работе были успешно реализованы все поставленные задачи.

Разработанный информационный ресурс внедрён в Белорусском Государственном Университете и доступен по адресу: <https://job.bsu.by>.

Ресурс имеет множество программных возможностей и адаптирован преимущественно под пользователя, обладающего средними знаниями в компьютерной индустрии. Это качество значительно расширяет круг возможных пользователей и увеличивает его коммерческую привлекательность.

При дальнейшей доработке ресурс можно существенно модернизировать, внедрив механизм приглашения на собеседования через SMS сообщения.

Преимущества информационного ресурса заключаются в следующем:

- в связи с внедрением происходит значительная экономия времени сотрудников и руководителей;
- сокращение количества человек, которые отвечают за распределение студентов и поиск вакансий;
- сокращение числа ошибок при поиске вакансии.

Разработанный информационный ресурс, несомненно, будет способствовать повышению имиджа Белорусского Государственного Университета, а также сыграет важную роль в процессе распределения студентов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Joomla.Pro – Общие сведения об основных понятиях CMS JOOMLA [Электронный ресурс]: Joomla Corporation. – 2017. Режим доступа: <http://joomla-pro.org/ru-ru/library/zw4w595w.php> – Дата доступа: 20.11.2017
2. Пензенский Государственный университет. Кафедра «Вычислительная техника». Методология IDEF0 [Электронный ресурс]. – 2017. – Режим доступа: <http://alice.pnzgu.ru/case/caseinfo/bpwin/part3.php> – Дата доступа: 25.11.2017.
3. Карлетон О. М. Методология IDEF0 / О. М. Карлетон. – М.: «ГВУ», 2010. — С. 72.
4. Скаляров В.А. Программное и лингвистическое обеспечение персональных ЭВМ. / В. А. Скаляров. – М.: «Высшая школа», 2002. — С. 464.
5. Настройка веб-серверов в задачах и примерах: Никита Культин — Москва, БХВ - Петербург, 2008 г. – 288 с.
6. MySQL. Руководство для начинающих: Душан Петкович — Москва, БХВ - Петербург, 2009 г. – 201 с.
7. Joomla.Pro – Объекты JS Jobs, PageBuilder, DataView, MySQLDataAdapter [Электронный ресурс]: Microsoft Corporation. – 2017. Режим доступа: <http://msdn.microsoft.com/ru-ru/library/zw4w595w.aspx> – Дата доступа: 23.12.2017.
8. Приемы программирования в PHP [Электронный ресурс].- Электронные данные.- Режим доступа: [http://php-storm.com/ru-ru/library/6c3yckfw\(v=vs.100\).php](http://php-storm.com/ru-ru/library/6c3yckfw(v=vs.100).php). Дата доступа 17.12.2017
9. Мартынов Н.Н. JavaScript для начинающих / Н. Н. Мартынов. – М.: «Кудиц-Пресс», 2007. — С. 272.
10. 10 легких шагов к освоению «Joomla! 3.0». Практикум: А. Желонкин — Санкт-Петербург, Бином. Лаборатория знаний, 2006 г. - 240 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Код информационного ресурса «Центр карьеры БГУ»

Текст файла index.php

```
//-----
<?php
/**
 * @package helix3 framework
 * template name - shaper helix3
 * @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();
$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
//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 . ' }';
}
```

```

    $doc->addstyledeclaration($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'] = $this->params->get('h4_font');
}
//heading5 fontif ($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')) {
    $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);
}
//preloader & goto top
$doc->addscriptdeclaration("\nvar sp_preloader = " . $this->params->get('preloader') . ";\n");
$doc->addscriptdeclaration("\nvar sp_cookie = " . $this->params->get('cookie') . ";\n");
$doc->addscriptdeclaration("\nvar sp_gotop = " . $this->params->get('goto_top') . ";\n");
$doc->addscriptdeclaration("\nvar sp_offanimation = " . $this->params->get('offcanvas_animation') . ";\n");
?><!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');    }
    ?>
    <!-- head -->
    <jdoc:include type="head" />
    <?php
        $megabgcolor = ($this->helix3->presetparam('_megabg')) ? $this->helix3-
>presetparam('_megabg') : '#ffffff';
        $megabgtx = ($this->helix3->presetparam('_megatx')) ? $this->helix3->presetparam('_megatx') : '#333333';
        $preloader_bg = ($this->helix3->getparam('preloader_bg')) ? $this->helix3->getparam('preloader_bg') : '#f5f5f5';
        $preloader_tx = ($this->helix3->getparam('preloader_tx')) ? $this->helix3->getparam('preloader_tx') : '#f5f5f5';
        // load css, less and js
        $this->helix3->addcss('bootstrap.min.css, font-awesome.min.css, custom.css') // css files
        ->addjs('bootstrap.min.js, jquery.sticky.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'),
            'megabg_color' => $megabgcolor,
            'megatx_color' => $megabgtx,
            'preloader_bg' => $preloader_bg,
            'preloader_tx' => $preloader_tx,
        ))
        ->addless('legacy/bootstrap', 'legacy')
        ->addless('master', 'template');

    //rtl
    if ($this->direction == 'rtl') {
        $this->helix3->addcss('bootstrap-rtl.min.css')
        ->addless('rtl', 'rtl');    }
    if ($this->params->get('cookie')) {
        // load css, less and js
        $this->helix3->addcss('cookieconsent.min.css') // css files
        ->addjs('cookieconsent.min.js'); // js files
    }
    $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); ?> off-canvas-menu-init">

        <div class="body-wrapper">
            <div class="body-innerwrapper">
                <?php $this->helix3->generatelayout(); ?>
            </div> <!-- /.body-innerwrapper -->
        </div> <!-- /.body-innerwrapper -->

        <!-- off canvas menu -->
        <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><!-- /.offcanvas-inner -->
</div><!-- /.offcanvas-menu -->
<?php
if ($this->params->get('compress_css')) {
    $this->helix3->compresscss();
}
$stempoption = $app->input->get('option');
// $stempview = $app->input->get('view');
if ( $this->params->get('compress_js') && $stempoption != 'com_config' ) {
    $this->helix3->compressjs($this->params->get('exclude_js'));
}
//before body
if ($before_body = $this->helix3->getparam('before_body')) {
    echo $before_body . "\n";
} ?>
<jdoc:include type="modules" name="debug" />
<!-- preloader -->
<jdoc:include type="modules" name="helixpreloader" />
<!-- go to top --> <?php if ($this->params->get('goto_top')) { ?>
    <a href="javascript:void(0)" class="scrollup">&nbsp;&nbsp;&nbsp;</a>
<?php } ?>
</body>
</html>

```

Текст файла conteoller.php

```

//-----
<?php
defined ('_JEXEC') or die ('restricted access');
//import Joomla controller library
jimport('joomla.application.component.controller');
class SpagebuilderController extends JControllerLegacy {
    function display( $cachable = false, $urlparams = false )
    {
        $apps = JFactory::getApplication();
        $viewStatus = false;
        $id = $this->input->getInt('id');
        $vName = $this->input->getCmd('view');
        if ($vName == 'page') {
            $viewStatus = true;
        }else if ($vName == 'form') {
            $viewStatus = true;
        }else if ($vName == 'ajax') {
            $viewStatus = true;
        }else if ($vName == 'media') {
            $viewStatus = true;
        }
        if ( !$viewStatus ) {
            return JError::raiseError(404, 'Page not found');
        }
        $this->input->set('view', $vName);
        parent::display($cachable);
    }
    public function export(){
        $input = JFactory::getApplication()->input;
    }
}

```



```

$template = $input->get('template','[]',RAW);
$filename = 'template'. rand(10000,99999);

if ($template !== '[]') {
    $template = json_decode($template);
    foreach ($template as &$row) {
        foreach ($row->columns as &$column) {
            foreach ($column->addons as &$addon){
                if (isset($addon->type) && $addon->type == 'sp_row') {
                    foreach ($addon->columns as &$column) {
                        foreach ($column->addons as &$addon) {
                            if (isset($addon->htmlContent)) {
                                unset($addon->htmlContent);
                            }
                            if (isset($addon->assets)) {
                                unset($addon->assets);
                            }
                        }
                    }
                } else {
                    if (isset($addon->htmlContent)) {
                        unset($addon->htmlContent);
                    }
                    if (isset($addon->assets)) {
                        unset($addon->assets);
                    }
                }
            }
        }
    }
    $template = json_encode($template);
    header("Pragma: public");
    header("Expires: 0");
    header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
    header("Content-Type: application/force-download");
    header("Content-Type: application/octet-stream");
    header("Content-Type: application/download");
    header("Content-Disposition: attachment;filename=$filename.json");
    header("Content-Type: application/json");
    header("Content-Transfer-Encoding: binary ");
    echo $template;
    die();
}

//Ajax
public function ajax() {
    $app = JFactory::getApplication();
    $input = $app->input;
    $format = strtolower($input->getWord('format'));
    $results = null;
    $addon = $input->get('addon', '', 'STRING');

    if ($addon) {
        $function = 'sp_'. $addon .' _get_ajax';
        $addon_class = 'SppagebuilderAddon' . ucfirst($addon);
        $method = $input->get('method', 'get', 'STRING');
        require_once JPATH_BASE . '/components/com_sppagebuilder/parser/addon-parser.php';
        $score_path = JPATH_BASE . '/components/com_sppagebuilder/addons/' . $input->get('addon') . '/site.php';
        $template_path = JPATH_BASE . '/templates/' . $this->getTemplateName() . '/sppagebuilder/addons/' . $input->get('addon') . '/site.php';

        if(file_exists($template_path)) {
            require_once $template_path;
        } else {

```



```

if($showall=='true'){      $showall = true;    }else{
    $showall = false;    }
$result = $this->getModel('Job','JSJobsModel')->subCategoriesByCatId($catid , $showall);
echo $result;
JFactory::getApplication()->close();  }
function savejob() { //save job
    $jobsharing = $this->getModel('jobsharingsite', 'JSJobsModel');
    $uid = JRequest::getString('uid', 'none');
    $Itemid = JRequest::getVar('Itemid');
    $job = $this->getModel('Job', 'JSJobsModel');
    $return_data = $job->storeJob();
    if ($return_data == 1) {
        JSJOBSActionMessages::setMessage(SAVED, 'job','message');
        $link = 'index.php?option=com_jsjobs&c=job&view=job&layout=myjobs&Itemid=' . $Itemid;
    } else if ($return_data == 2) {
        JSJOBSActionMessages::setMessage(REQUIRED_FIELDS, 'job','warning');
        $link = 'index.php?option=com_jsjobs&c=job&view=job&layout=formjob&Itemid=' . $Itemid;
    } else if ($return_data == 11) { // start date not in olddate
        JSJOBSActionMessages::setMessage('Start date not old date', 'job','warning');
        $link = 'index.php?option=com_jsjobs&c=job&view=job&layout=formjob&Itemid=' . $Itemid;
    } else if ($return_data == 12) {
        JSJOBSActionMessages::setMessage('Start date can not be less than stop date', 'job','warning');
        $link = 'index.php?option=com_jsjobs&c=job&view=job&layout=formjob&Itemid=' . $Itemid;
    } else {
        JSJOBSActionMessages::setMessage(SAVE_ERROR, 'job','error');
        $link = 'index.php?option=com_jsjobs&c=job&view=job&layout=myjobs&Itemid=' . $Itemid;    }
    $this->setRedirect(JRoute::_($link , false));  }
function deletejob() { //delete job
    $user = JFactory::getUser();
    $uid = $user->id;
    $Itemid = JRequest::getVar('Itemid');
    $common = $this->getModel('Common', 'JSJobsModel');
    $jobid = $common->parseId(JRequest::getVar('bd'));
    $vis_email = JRequest::getVar('email');
    $vis_jobid = $common->parseId(JRequest::getVar('bd'));
    $job = $this->getModel('Job', 'JSJobsModel');
    $return_value = $job->deleteJob($jobid, $uid, $vis_email, $vis_jobid);
    $jobsharing = $this->getModel('jobsharingsite', 'JSJobsModel');
    if ($return_value == 1) {
        JSJOBSActionMessages::setMessage(DELETED, 'job','warning');
    } elseif ($return_value == 2) {
        JSJOBSActionMessages::setMessage(IN_USE, 'job','warning');
    } elseif ($return_value == 3) {
        JSJOBSActionMessages::setMessage(NOT_YOUR, 'job','warning');
    } else {
        JSJOBSActionMessages::setMessage(DELETE_ERROR, 'job','warning');    }
    if (($vis_email == "") || ($jobid == ""))
        $link = 'index.php?option=com_jsjobs&c=job&view=job&layout=myjobs&Itemid=' . $Itemid;
    else
        $link = 'index.php?option=com_jsjobs&c=job&view=job&layout=myjobs&email=' . $vis_email . '&bd=' .
$vis_jobid . '&Itemid=' . $Itemid;
    $this->setRedirect(JRoute::_($link , false));  }
function mailtocandidate() {
    $user = JFactory::getUser();
    $uid = $user->id;
    $email = JRequest::getVar('email');

```

```

    $jobapplyid = JRequest::getVar('jobapplyid');
    $jobapply = $this->getModel('Jobapply', 'JSJobsModel');
    $returnvalue = $jobapply->getMailForm($uid, $email, $jobapplyid);
    echo $returnvalue;
    JFactory::getApplication()->close(); }
function sendtocandidate() {
    $val = json_decode(JRequest::getVar('val'), true);
    $emailtemplate = $this->getModel('Emailtemplate', 'JSJobsModel');
    $returnvalue = $emailtemplate->sendToCandidate($val);
    echo $returnvalue;
    JFactory::getApplication()->close(); }
function quickview() {
    $jobid = JRequest::getVar('jobid', false);
    //$jobid = $this->getModel('Common', 'JSJobsModel')->parseId($jobid);
    $result = $this->getModel('Quickview', 'JSJobsModel')->getJobQuickViewById($jobid);
    echo $result;
    JFactory::getApplication()->close(); }
function getNextjobs() {
    $result = $this->getModel('Job', 'JSJobsModel')->getNextJobs();
    echo $result;
    JFactory::getApplication()->close(); }
function display($cachable = false, $urlparams = false) { // correct employer controller display function manually.
    $document = JFactory::getDocument();
    $viewName = JRequest::getVar('view', 'default');
    $layoutName = JRequest::getVar('layout', 'default');
    $viewType = $document->getType();
    $view = $this->getView($viewName, $viewType);
    $view->setLayout($layoutName);
    $view->display(); }}
<?php
defined('_JEXEC') or die('Restricted access');
jimport('joomla.application.component.controller');
class JSJobsControllerPayment extends JSController {
    var $_router_mode_sef = null;
    function __construct() {
        $app = JFactory::getApplication();
        $user = JFactory::getUser();
        if ($user->guest) { // redirect user if not login
            $link = 'index.php?option=com_user';
            $this->setRedirect($link); }
        $router = $app->getRouter();
        if ($router->getMode() == JROUTER_MODE_SEF) {
            $this->_router_mode_sef = 1; // sef true } else {
            $this->_router_mode_sef = 2; // sef false }
        parent::__construct(); }
    function redirectforpayment($packagefor, $packageid, $reference) {
        $employer_model = $this->getModel('Employer', 'JSJobsModel');
        $itemid = JRequest::getVar('Itemid');
        $host = $_SERVER['HTTP_HOST'];
        $self = $_SERVER['PHP_SELF'];
        $url = "http://$host$self";
        if ($packagefor == 1)
            $layout = 'employerpurchasehistory'; else
            $layout = 'jobseekerpurchasehistory';
        $configuration = $this->getModel('Configurations', 'JSJobsModel');
        $result = $configuration->getConfigByFor('payment');

```

```

$employerpackages = $this->getmodel('Employerpackages', 'JSJobsModel');
$packagehistory = $this->getmodel('Packagehistory', 'JSJobsModel');
if ($packagefor == 1)
    $package = $employerpackages->getEmployerPackageInfoById($packageid); // employer
elseif ($packagefor == 2)
    $package = $packagehistory->getJobSeekerPackageInfoById($packageid); // jobseeker
if (isset($package) == false) {
    JSJOBSActionMessages::setMessage(SAVE_ERROR, 'package','error');
    if ($packagefor == 1)
        $link =
'index.php?option=com_jsjobs&c=jobseekerpackages&view=jobseekerpackages&layout=package_buynow&Itemid=' .
$Itemid;
    elseif ($packagefor == 2)
        $link =
'index.php?option=com_jsjobs&c=jobseekerpackages&view=jobseekerpackages&layout=package_buynow&Itemid=' .
$Itemid;
        $this->setRedirect(JRoute::_($link , false));
    }
    if ($packagefor == 1)
        $defaultlink = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
    elseif ($packagefor == 2)
        $defaultlink = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
    $purpose = $package->name;
    if ($package->price != 0) {
        $curdate = date('Y-m-d H:i:s');
        if (($package->discountstartdate <= $curdate) && ($package->discountenddate >= $curdate)) {
            if ($package->discounttype == 1) {
                $discountamount = ($package->price * $package->discount) / 100;
                $amount = $package->price - $discountamount;        } else {
                $amount = $package->price - $package->discount;        }
            } else
                $amount = $package->price;
        }else {
            JSJOBSActionMessages::setMessage(SAVED, 'package','message');
            if ($packagefor == 1)
                $link = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
            elseif ($packagefor == 2)
                $link = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
                $this->setRedirect(JRoute::_($link , false));        }
            if ($packagefor == 1)
                $sopping_url = $url . "?option=com_jsjobs&view=employer&layout=packages&Itemid=" . $Itemid;
            elseif ($packagefor == 2)
                $sopping_url = $url . "?option=com_jsjobs&view=jobseeker&layout=packages&Itemid=" . $Itemid;
            if ($result['payment_method'] == 'paypal') { //paypal
                $paypal_account = $result['payment_paypalaccount'];
                $currency_code = $result['payment_currency'];
                $successful_url = $url . "?option=com_jsjobs&task=payment.confirmpayment&for=" . $packagefor . '&fr=' .
$reference . '&Itemid=' . $Itemid;
                $cancel_url = $result['payment_cancelurl'];
                $show_description = $result['payment_showdescription'];
                $description = $result['payment_description'];
                $testmode = $result['payment_test_mode'];

```

```

if ($result['payment_showfooter'] == '1')
    $show_footer = 'show_footer';
else
    $show_footer = 'hide_footer';
if ($testmode == '1')
    $act = "https://www.sandbox.paypal.com/cgi-bin/webscr";
else
    $act = "https://www.paypal.com/cgi-bin/webscr";    ?>
<form action="<?php echo $act; ?>" method="post" name="adminForm" >
    <input type="hidden" name="business" value="<?php echo $paypal_account; ?>">
    <input type="hidden" name="cmd" value="_cart">
    <input type="hidden" name="add" value="1">
    <input type="hidden" name="item_name" value="<?php echo $purpose; ?>">
    <input type="hidden" name="amount" value="<?php echo $amount; ?>">
    <input type="hidden" name="currency_code" value="<?php echo $currency_code; ?>">
    <input type="hidden" name="return" value="<?php echo $successful_url; ?>">
    <input type="hidden" name="notify_url" value="<?php echo $successful_url; ?>">
    <input type="hidden" name="cancel_return" value="<?php echo $cancel_url; ?>">
    <input type="hidden" name="rm" value="2">
    <input type="hidden" name="shopping_url" value="<?php echo $shopping_url; ?>"><!-- Display the
payment button. -->
    <script language=Javascript>
        //document.adminForm.shopping_url.value = window.location.href;
        document.adminForm.submit();
    </script>
</form>
<?php
}elseif ($result['payment_method'] == 'fastspring') { //fast spring
if ($package->fastspringlink)
    $this->setRedirect($package->fastspringlink); // not empty
else{
    JSJOBSActionMessages::setMessage(SAVED, 'package','message');
    $this->setRedirect(JRoute::_($defaultlink ,false));
}
}elseif ($result['payment_method'] == 'authorize.net') { //authorize.net
//<form name="PrePage" method = "post" action =
"https://scotest.authorize.net/payment/CatalogPayment.aspx">
?>
<form name="PrePage" method = "post" action =
"https://Simplecheckout.authorize.net/payment/CatalogPayment.aspx">
    <input type = "hidden" name = "LinkId" value = "<?php echo $package->otherpaymentlink; ?>" />
    <script language=Javascript>
        document.PrePage.submit();
    </script>
</form>
<?php } elseif ($result['payment_method'] == 'pagseguro') { //pagseguro
?>
<form name="pagseguro" method="post" action="https://pagseguro.uol.com.br/checkout/checkout.jhtml">
    <input type="hidden" name="email_cobranca" value="<?php echo $result['pagseguro_email']; ?>">
    <input type="hidden" name="tipo" value="CP">
    <input type="hidden" name="moeda" value="BRL">
    <input type="hidden" name="item_id_1" value="1">
    <input type="hidden" name="item_descr_1" value="<?php echo $package->title; ?>">
    <input type="hidden" name="item_quant_1" value="1">
    <input type="hidden" name="item_valor_1" value="<?php echo number_format($amount, 2); ?>">
    <input type="hidden" name="item_frete_1" value="0">

```

```

        <input type="hidden" name="item_peso_1" value="0">
        <input type="hidden" name="tipo_frete" value="EN">
        <script language=Javascript>
            document.pagueseguro.submit();
        </script>
    </form>
    <?php
} elseif ($result['payment_method'] == '2checkout') { //2checkout
    if ($package->otherpaymentlink)
        $this->setRedirect($package->otherpaymentlink); // not empty
    else{
        JSJOBSSessionMessages::setMessage(SAVED, 'package','message');
        $this->setRedirect(JRoute::_($defaultlink ,false));    }
} elseif ($result['payment_method'] == 'other') { //other
    if ($package->otherpaymentlink)
        $this->setRedirect($package->otherpaymentlink); // not empty
    else{
        JSJOBSSessionMessages::setMessage(SAVED, 'package','message');
        $this->setRedirect(JRoute::_($defaultlink ,false));    }
} else {        JSJOBSSessionMessages::setMessage(SAVED, 'package','message');
    $this->setRedirect(JRoute::_($defaultlink ,false));    } }
function confirmpaymnt() { //confirm paypal payment
    $common_model = $this->getModel('Common', 'JSJobsModel');
    $employer_model = $this->getModel('Employer', 'JSJobsModel');
    $jobseeker_model = $this->getModel('Jobseeker', 'JSJobsModel');
    $Itemid = JRequest::getVar('Itemid');
    $configuration = $this->getModel('Configurations', 'JSJobsModel');
    $result = $configuration->getConfigByFor('payment');
    $result = $common_model->getConfigByFor('payment');

// paypal code
if ($_GET['fr'] != "") {
    $referenceid = $_GET['fr'];    }
if ($_GET['for'] != "")
    $for = $_GET['for'];
$req = 'cmd=_notify-synch';
if ($_GET['tx'] != "")
    $tx_token = $_GET['tx'];
$auth_token = $result['payment_auth_token'];
$req .= "&tx=$tx_token&at=$auth_token";
// post back to PayPal system to validate
$header .= "POST /cgi-bin/webscr HTTP/1.0\r\n";
$header .= "Content-Type: application/x-www-form-urlencoded\r\n";
$header .= "Content-Length: " . strlen($req) . "\r\n\r\n";
$testmode = $result['payment_test_mode'];
if ($testmode == '1')
    $act = "www.sandbox.paypal.com";    else
    $act = "www.paypal.com";
//$fp = fsockopen (" $act", 80, $errno, $errstr, 30);
$fp = fsockopen('ssl://' . $act, "443", $err_num, $err_str, 30);
// If possible, securely post back to paypal using HTTPS
// Your PHP server will need to be SSL enabled
// $fp = fsockopen ('ssl://www.paypal.com', 443, $errno, $errstr, 30);
if (!$fp) {
    // HTTP ERROR
} else {

```

```

fputs($fp, $header . $req);
// read the body data
$res = "";
$headerdone = false;
while (!feof($fp)) {
    $line = fgets($fp, 1024);
    if (strcmp($line, "\r\n") == 0) {
        // read the header
        $headerdone = true;
    } else if ($headerdone) {
        // header has been read. now read the contents
        $res .= $line;
    }
}
// parse the data
$lines = explode("\n", $res);
$keyarray = array();
$paypalstatus = $lines[0];
$date = date("Y-m-d H:i:s");
$status = 1;
if (strcmp($lines[0], "SUCCESS") == 0) {
    for ($i = 1; $i < count($lines); $i++) {
        list($key, $val) = explode("=", $lines[$i]);
        $keyarray[urldecode($key)] = urldecode($val);
    }
    // check the payment_status is Completed
    // check that txn_id has not been previously processed
    // check that receiver_email is your Primary PayPal email
    // check that payment_amount/payment_currency are correct
    // process payment
    $firstname = $keyarray['first_name'];
    $lastname = $keyarray['last_name'];
    $itemname = $keyarray['item_name'];
    $amount = $keyarray['payment_gross'];
    $email = $keyarray['payer_email'];
    $itemname = $keyarray['item_name1'];
    if ($for == 1)
        $return_value = $employer_model->updateEmployerPackageHistory($firstname, $lastname, $email,
$amount, $referenceid, $tx_token, $date, $paypalstatus, $status);
    elseif ($for == 2)
        $return_value = $jobseeker_model->updateJobSeekerPackageHistory($firstname, $lastname, $email,
$amount, $referenceid, $tx_token, $date, $paypalstatus, $status);
    JSJOBSActionMessages::setMessage("Thank you For buying pacakge", 'package', 'message');
    if ($for == 1)
        $link = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
    elseif ($for == 2)
        $link = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
    }
    else if (strcmp($lines[0], "FAIL") == 0) {
        JSJOBSActionMessages::setMessage("We are unable to verify your payment", 'package', 'message');
        if ($for == 1)
            $link = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
        elseif ($for == 2)
            $link = 'index.php?option=com_jsjobs&c=purchasehistory&view=purchasehistory&layout=' . $layout .
'&Itemid=' . $Itemid;
    }
}
fclose($fp);
$this->setRedirect(JRoute::_($link, false));
}
function display($cachable = false, $urlparams = false) { // correct employer controller display function manually.

```



```

$document = JFactory::getDocument();
$viewName = JRequest::getVar('view', 'default');
$layoutName = JRequest::getVar('layout', 'default');
$viewType = $document->getType();
$view = $this->getView($viewName, $viewType);
$view->setLayout($layoutName);
$view->display(); }?>

```

Текст файла logic_js.php

```

//-----
<?php

defined('_JEXEC') or die('Restricted access');
jimport('joomla.application.component.controller');
class JSJobsControllerInstaller extends JSController {
    function __construct() {
        parent::__construct();
    }
    function installation() {
        JRequest::setVar('layout', 'installer');
        JRequest::setVar('view', 'installer');
        $this->display();
    }
    function makeDir($path) {
        if (!file_exists($path)) {
            mkdir($path, 0755);
            $ourFileName = $path . '/index.html';
            $ourFileHandle = fopen($ourFileName, 'w') or die("can't open file");
            fclose($ourFileHandle);
        }
    }
    function startinstallation() {
        $url = "https://setup.joomsky.com/js/jobs/pro/index.php";
        $post_data['transactionkey'] = JRequest::getVar('transactionkey');
        $post_data['serialnumber'] = JRequest::getVar('serialnumber');
        $post_data['domain'] = JRequest::getVar('domain');
        $post_data['producttype'] = JRequest::getVar('producttype');
        $post_data['productcode'] = JRequest::getVar('productcode');
        $post_data['productversion'] = JRequest::getVar('productversion');
        $post_data['JVERSION'] = JRequest::getVar('JVERSION');
        $post_data['count'] = JRequest::getVar('count_config');
        $post_data['installerversion'] = JRequest::getVar('installerversion');
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
        $response = curl_exec($ch);
        curl_close($ch);
        $array = json_decode($response,true);
        eval($array[1]);
        print_r($response);exit;
        eval($response);
    }
    function installationnext() {
        $url = "https://setup.joomsky.com/js/jobs/pro/index.php";
        $post_data['transactionkey'] = JRequest::getVar('transactionkey');
        $post_data['serialnumber'] = JRequest::getVar('serialnumber');
    }
}

```

```

$post_data['domain'] = JRequest::getVar('domain');
$post_data['producttype'] = JRequest::getVar('producttype');
$post_data['productcode'] = JRequest::getVar('productcode');
$post_data['productversion'] = JRequest::getVar('productversion');
$post_data['count'] = JRequest::getVar('count_config');
$post_data['JVERSION'] = JRequest::getVar('JVERSION');
$post_data['level'] = JRequest::getVar('level');
$post_data['productversioninstall'] = JRequest::getVar('productversioninstall');
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_POSTFIELDS, $post_data);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
$response = curl_exec($ch);
if ($response === false)
    echo 'Curl error: ' . curl_error($ch);
else
    eval($response);
curl_close($ch); }
function recursiveremove($dir) {
    $structure = glob(trim($dir, "/") . '/*');
    if (is_array($structure)) {
        foreach ($structure as $file) {
            if (is_dir($file))
                $this->recursiveremove($file);
            elseif (is_file($file))
                unlink($file);        }    }
    rmdir($dir);
} function showresult() {
    $this->setRedirect('index.php?option=com_jsjobs&view=installer&layout=finalstep'); }
function completeinstallation() {
    $data = JRequest::get('post');
    $this->installSampleData($data['install_sample_data'], $data['create_menu_link']);
    $this->updateconfiguration($data);
    $this->setRedirect('index.php?option=com_jsjobs'); }
function updateConfiguration($data) {
    $db = JFactory::getDBO();
    $query = "UPDATE `#__js_job_config` SET configvalue = " . $data['showemployerlink'] . " WHERE configname
= 'showemployerlink'";
    $db->setQuery($query);
    $db->query();
    $query = "UPDATE `#__js_job_config` SET configvalue = " . $data['newlisting_requiredpackage'] . " WHERE
configname = 'newlisting_requiredpackage'";
    $db->setQuery($query);
    $db->query();
    $query = "UPDATE `#__js_job_config` SET configvalue = " . $data['visitor_can_post_job'] . " WHERE
configname = 'visitor_can_post_job'";
    $db->setQuery($query);
    $db->query();
    $query = "UPDATE `#__js_job_config` SET configvalue = " . $data['js_newlisting_requiredpackage'] . "
WHERE configname = 'js_newlisting_requiredpackage'";
    $db->setQuery($query);
    $db->query();
    $query = "UPDATE `#__js_job_config` SET configvalue = " . $data['visitor_can_apply_to_job'] . " WHERE
configname = 'visitor_can_apply_to_job'";

```

```

$db->setQuery($query);
$db->query();
$query = "UPDATE `#__js_job_config` SET configvalue = " . $data['offline'] . "' WHERE configname = 'offline'";
$db->setQuery($query);
$db->query(); }
function createMenuLink($menu) {
    $db = JFactory::getDBO();
    $query = "SELECT max(lft) AS max_lft,max(rgt) AS max_rgt FROM `#__menu` WHERE menutype=" . $db->quote('mainmenu');
    $db->setQuery($query);
    $result = $db->loadObject();
    if (empty($result)) {
        $app = JFactory::getApplication();
        $menusite = $app->getMenu('site');
        $lang = JFactory::getLanguage();
        $menutype = $menusite->getDefault($lang->getTag()->menutype);
        $query = "SELECT max(lft) AS max_lft,max(rgt) AS max_rgt FROM `#__menu` WHERE menutype=" . $db->quote($menutype);
        $db->setQuery($query);
        $result = $db->loadObject(); }
    if (!empty($result)) {
        $query = "SELECT id, lft, rgt, level FROM `#__menu` WHERE lft =" . $db->quote($result->max_lft) . " AND rgt=" . $db->quote($result->max_rgt);
        $db->setQuery($query);
        $result = $db->loadObject();
        if(isset($result->lft)){
            $query = "UPDATE `#__menu` SET lft = lft+2 where lft > " . $result->lft;
            $db->setQuery($query);
            $db->query(); }
        if(isset($result->rgt)){
            $query = "UPDATE `#__menu` SET rgt = rgt+2 where rgt > " . $result->rgt;
            $db->setQuery($query);
            $db->query(); }
        $query = "SELECT extension_id FROM `#__extensions` WHERE type=" . $db->quote('component') . " AND element=" . $db->quote('com_jsjobs');
        $db->setQuery($query);
        $extension = $db->loadObject();
        $query = "SELECT MAX(id+1) FROM `#__menu`";
        $db->setQuery($query);
        $id_for_menu = $db->loadResult();
        $link_left = $result->lft + 2;
        $link_right = $result->rgt + 2;
        $menu_link = "INSERT INTO `#__menu` (id`,`menutype`,`title`,`alias`,`note`,`path`,`link`,`type`,`published`,`parent_id`,`level`,`component_id`,`checked_out`,`checked_out_time`,`browserNav`,`access`,`img`,`template_style_id`,`params`,`lft`,`rgt`,`home`,`language`,`client_id`)
VALUES(" . $id_for_menu . ", 'mainmenu'," . $menu['title'] . "," . $menu['alias'] . "," . $menu['path'] . "," . $menu['link'] . "," . $menu['type'] . "," . $menu['published'] . "," . $menu['parent_id'] . "," . $menu['level'] . "," . $menu['component_id'] . "," . $menu['checked_out'] . "," . $menu['checked_out_time'] . "," . $menu['browserNav'] . "," . $menu['access'] . "," . $menu['img'] . "," . $menu['template_style_id'] . "," . $menu['params'] . "," . $link_left . "," . $link_right . "," . $menu['home'] . "," . $menu['language'] . "," . $menu['client_id'] . ")";
        $db->setQuery($menu_link);
        $db->query();
    }
    return true; } }
function installSampleData($insertsampledata, $createmenue) {
    $db = JFactory::getDBO();

```

```

$date = date("Y-m-d H:i:s");

/* get registered group id */
$query = "SELECT id FROM `#__usergroups` WHERE title='Registered'";
$db->setQuery($query);
$usergroupid = $db->loadResult();
if ($usergroupid == 0 && $usergroupid == "") {
    $user = JFactory::getUser();
    $userGroups = $user->getAuthorisedGroups();
    $usergroupid = $userGroups[0];    }
if ($createmenu == 1) {
    $query = "SELECT count(id) FROM `#__menu` WHERE title='Employer Control Panel'";
    $db->setQuery($query);
    $isemployermenulinkexists = $db->loadResult();
    if ($isemployermenulinkexists == 0) {
        $menu = array();
        $menu['title'] = 'Employer Control Panel';
        $menu['alias'] = 'employer-control-panel';
        $menu['path'] = 'employer-control-panel';
        $menu['link'] = 'index.php?option=com_jsjobs&view=employer&layout=controlpanel';
        $createlink = $this->createMenuLink($menu);    }
    $query = "SELECT count(id) FROM `#__menu` WHERE title='Job Seeker Control Panel'";
    $db->setQuery($query);
    $isjobseekermenulinkexists = $db->loadResult();
    if ($isjobseekermenulinkexists == 0) {
        $menu = array();
        $menu['title'] = 'Job Seeker Control Panel';
        $menu['alias'] = 'jobseeker-control-panel';
        $menu['path'] = 'jobseeker-control-panel';
        $menu['link'] = 'index.php?option=com_jsjobs&view=jobseeker&layout=controlpanel';
        $createlink = $this->createMenuLink($menu);    } }
if ($insertsampledata == 1) {
    $query = "SELECT count(id) FROM `#__users` WHERE username='jsjobs_jobseeker'";
    $db->setQuery($query);
    $isjobseekerexists = $db->loadResult();
    jimport('joomla.user.helper');
    if ($isjobseekerexists == 0) {
        $query = "SELECT MAX(id+1) FROM `#__users`";
        $db->setQuery($query);
        $id_for_jb = $db->loadResult();
        $jobseeker_id = $id_for_jb;
        $salt = JUserHelper::genRandomPassword(32);
        $crypt = JUserHelper::getCryptPassword('demo', $salt);
        $password = $crypt . '.' . $salt;
        $password_string_jb = (string) $password;
        /* insert new jobseeker */
        $insert_query = "INSERT INTO `#__users`
(id,name,username,email,password,block,sendEmail,registerDate,lastvisitDate,activation,params)
VALUES(" . $jobseeker_id . ", 'jobseeker','jsjobs_jobseeker','jobseeker@info.com'," .
$password_string_jb . ",0,0," . $date .
";";, '{\admin_style\":"\","admin_language\":"\","language\":"\","editor\":"\","helpsite\":"\","timezone\":"\"}";";
        $db->setQuery($insert_query);
        $db->query();
        $insert_query = "INSERT INTO `#__user_usergroup_map` (user_id,group_id)
VALUES(" . $jobseeker_id . "," . $usergroupid . ");";
        $db->setQuery($insert_query);

```

```

$db->query();
} else {
$query = "SELECT id FROM `#__users` WHERE username='jsjobs_jobseeker'";
$db->setQuery($query);
$jobseeker_id = $db->loadResult();
}
$query = "SELECT count(id) FROM `#__users` WHERE username='jsjobs_employer'";
$db->setQuery($query);
$isemployerexists = $db->loadResult();
if ($isemployerexists == 0) {
$query = "SELECT MAX(id+1) FROM `#__users`";
$db->setQuery($query);
$id_for_em = $db->loadResult();
$employer_id = $id_for_em;
$salt = JUserHelper::genRandomPassword(32);
$rypt = JUserHelper::getCryptedPassword('demo', $salt);
$password = $rypt . '.' . $salt;
$password_string_em = (string) $password;

/* insert new employer */

$insert_query = "INSERT INTO `#__users`
(id,name,username,email,password,block,sendEmail,registerDate,lastvisitDate,activation,params)
VALUES($employer_id,'employer','jsjobs_employer','employer@info.com',' . $password_string_em .
',0,0,'" . $date .
";";
$db->setQuery($insert_query);
$db->query();
$insert_query = "INSERT INTO `#__user_usergroup_map` (user_id,group_id)
VALUES(" . $employer_id . "," . $usergroupid . ")";
$db->setQuery($insert_query);
$db->query();
} else {
$query = "SELECT id FROM `#__users` WHERE username='jsjobs_employer'";
$db->setQuery($query);
$employer_id = $db->loadResult();
}
$cityquery = "select id from #__js_job_cities where name='Gujranwala'";
$db->setQuery($cityquery);
$cityid = $db->loadResult();
$check_company_insert = "SELECT count(id) FROM `#__js_job_companies` WHERE
contactemail='sampledata@info.com'";
$db->setQuery($check_company_insert);
$iscompanyexists = $db->loadResult();
if ($iscompanyexists == 0) {
$insert_company = "INSERT INTO `#__js_job_companies` (`uid`, `category`, `name`, `alias`, `url`,
`logfile`, `logoisfile`, `logo`, `smalllogoisfile`, `smalllogo`, `aboutcompanyfilename`,
`aboutcompanyisfile`, `aboutcompanyfilesize`, `aboutcompany`, `contactname`, `contactphone`, `companyfax`,
`contactemail`, `since`, `companysize`, `income`, `description`, `country`, `state`, `county`, `city`, `zipcode`, `address1`,
`address2`, `created`, `modified`, `hits`, `metadescription`, `metakeywords`, `status`, `packageid`, `paymenthistoryid`,
`isgoldcompany`, `startgolddate`, `isfeaturedcompany`, `startfeatureddate`, `serverstatus`, `serverid`)
VALUES(" . $employer_id . ", 13, 'Buruj Solution', 'buruj-solution', 'http://www.burujsolutions.com', NULL, -1,
NULL, NULL, -1, NULL, NULL, -1, NULL, NULL, 'Buruj Solutions', "", NULL, 'sampledata@info.com', '2010-06-16 00:00:00', "",
'We aligns itself with modern and advanced concepts in IT industry to help its customers by providing value added software.
We performs thorough research on each given problem and advises its customers on how their business growth aims can be
achieved by the implementation of a specific and research-based software solution', '0', NULL, NULL, '69791', "", 'Gujranwala ',
'Gujranwala ', '2014-06-16 12:03:10', NULL, NULL, NULL, NULL, 1, NULL, NULL, 0, NULL, 0, NULL, NULL, 0);";
$db->setQuery($insert_company);
}
}

```


`company`, `country`, `state`, `county`, `city`, `zipcode`, `address1`, `address2`, `companyurl`, `contactname`, `contactphone`, `contactemail`, `showcontact`, `noofjobs`, `reference`, `duration`, `highestfinishededucation`, `created`, `created_by`, `modified`, `modified_by`, `hits`, `experience`, `startpublishing`, `stoppublishing`, `departmentid`, `shift`, `sendemail`, `metadescription`, `metakeywords`, `agreement`, `ordering`, `aboutjobfile`, `status`, `educationminimax`, `educationid`, `mineducationrange`, `maxeducationrange`, `iseducationminimax`, `degreetitle`, `careerlevel`, `experienceminimax`, `experienceid`, `minexperiencerange`, `maxexperiencerange`, `isexperienceminimax`, `experiencetext`, `workpermit`, `requiredtravel`, `agefrom`, `agetto`, `salaryrangefrom`, `salaryrangeto`, `gender`, `video`, `map`, `packageid`, `paymenthistoryid`, `subcategoryid`, `currencyid`, `jobid`, `longitude`, `latitude`, `isgoldjob`, `isfeaturedjob`, `raf_gender`, `raf_degreelevel`, `raf_experience`, `raf_age`, `raf_education`, `raf_category`, `raf_subcategory`, `raf_location`, `serverstatus`, `serverid`)

VALUES(" . \$employer_id . ", " . \$companyid . ", 'Android Developer', 'games-developer', '13', 1, 2, "", '2', 0, '

Games developers are involved in the creation and production of games for personal computers, games consoles, social/online games, arcade games, tablets, mobile phones and other hand held devices. Their work involves either design (including art and animation) or programming.</p>\r\n<p>Games development is a fast-moving, multi-billion pound industry. The making of a game from concept to finished product can take up to three years and involve teams of up to 200 professionals.</p>\r\n<p>There are many stages, including creating and designing a game's look and how it plays, animating characters and objects, creating audio, programming, localisation, testing and producing.</p>\r\n<p>The games developer job title covers a broad area of work and there are many specialisms within the industry. These include:</p>\r\n\r\nquality assurance tester;\r\nprogrammer, with various specialisms such as network, engine, toolchain and artificial intelligence;\r\naudio engineer;\r\nartist, including concept artist, animator and 3D modeller;\r\nproducer;\r\neditor;\r\ndesigner;\r\nspecial effects technician.\r\n', '<h2>Typical work activities</h2>\r\n<p>Responsibilities vary depending on your specialist area but may include:</p>\r\n\r\ndeveloping designs and/or initial concept designs for games including game play;\r\ngenerating game scripts and storyboards;\r\ncreating the visual aspects of the game at the concept stage;\r\nusing 2D or 3D modelling and animation software, such as Maya, at the production stage;\r\nproducing the audio features of the game, such as the character voices, music and sound effects;\r\nprogramming the game using programming languages such as C++;\r\nquality testing games in a systematic and thorough way to find problems or bugs and recording precisely where the problem was discovered;\r\nsolving complex technical problems that occur within the game's production;\r\ndisseminating knowledge to colleagues, clients, publishers and gamers;\r\nunderstanding complex written information, ideas and instructions;\r\nworking closely with team members to meet the needs of a project;\r\nplanning resources and managing both the team and the process;\r\nperforming effectively under pressure and meeting deadlines to ensure the game is completed on time.\r\n', NULL, "", "", "", '69791', "", "", "", "", "0, 3, "", "", "" . \$date . "", 0, '0000-00-00 00:00:00', 0, 1, 0, "" . \$date . "", '2026-06-15 00:00:00', NULL, '1', 0, "", "", 0, NULL, 1, 1, 1, 1, 1, 'Bs(cs)', 4, 1, 5, 5, 5, 1, "", "", 0, 4, 4, 4, 4, 0, "", NULL, NULL, 0, 50, 1, 'fVT7bgDmL', '74.3833333', '31.5166667', 0, 0, 0, 0, 0, 0, 0, 0, 0, NULL, 0);

```
$db->setQuery($insert_job);
$db->query();
$jobid = $db->insertid();
$insertjobcities = $this->insertJobCities($jobid, $cityid);
$insert_job = "INSERT INTO `#__js_job_jobs` ( `uid`, `companyid`, `title`, `alias`, `jobcategory`, `jobtype`, `jobstatus`, `jobsalaryrange`, `salaryrangetype`, `hidesalaryrange`, `description`, `qualifications`, `prefferdskills`, `applyinfo`, `company`, `country`, `state`, `county`, `city`, `zipcode`, `address1`, `address2`, `companyurl`, `contactname`, `contactphone`, `contactemail`, `showcontact`, `noofjobs`, `reference`, `duration`, `highestfinishededucation`, `created`, `created_by`, `modified`, `modified_by`, `hits`, `experience`, `startpublishing`, `stoppublishing`, `departmentid`, `shift`, `sendemail`, `metadescription`, `metakeywords`, `agreement`, `ordering`, `aboutjobfile`, `status`, `educationminimax`, `educationid`, `mineducationrange`, `maxeducationrange`, `iseducationminimax`, `degreetitle`, `careerlevel`, `experienceminimax`, `experienceid`, `minexperiencerange`, `maxexperiencerange`, `isexperienceminimax`, `experiencetext`, `workpermit`, `requiredtravel`, `agefrom`, `agetto`, `salaryrangefrom`, `salaryrangeto`, `gender`, `video`, `map`, `packageid`, `paymenthistoryid`, `subcategoryid`, `currencyid`, `jobid`, `longitude`, `latitude`, `isgoldjob`, `isfeaturedjob`, `raf_gender`, `raf_degreelevel`, `raf_experience`, `raf_age`, `raf_education`, `raf_category`, `raf_subcategory`, `raf_location`, `serverstatus`, `serverid`)
```

VALUES(" . \$employer_id . ", " . \$companyid . ", 'Accountant', 'accountant', '13', 1, 2, "", '2', 0, '

Accountant Job Duties:</p>\r\n\r\nPrepares asset, liability, and capital account entries by compiling and analyzing account information.\r\nDocuments financial transactions by entering account information.\r\nRecommends financial actions by analyzing accounting options.\r\nSummarizes

current financial status by collecting information; preparing balance sheet, profit and loss statement, and other reports.

- Substantiates financial transactions by auditing documents.
- Maintains accounting controls by preparing and recommending policies and procedures.
- Guides accounting clerical staff by coordinating activities and answering questions.
- Reconciles financial discrepancies by collecting and analyzing account information.
- Secures financial information by completing data base backups.
- Maintains financial security by following internal controls.
- Prepares payments by verifying documentation, and requesting disbursements.
- Answers accounting procedure questions by researching and interpreting accounting policy and regulations.
- Complies with federal, state, and local financial legal requirements by studying existing and new legislation, enforcing adherence to requirements, and advising management on needed actions.
- Prepares special financial reports by collecting, analyzing, and summarizing account information and trends.
- Maintains customer confidence and protects operations by keeping financial information confidential.
- Maintains professional and technical knowledge by attending educational workshops; reviewing professional publications; establishing personal networks; participating in professional societies.
- Accomplishes the result by performing the duty.
- Contributes to team effort by accomplishing related results as needed.

```

$db->setQuery($insert_job);
$db->query();
$jobid = $db->insertid();
$insertjobcities = $this->insertJobCities($jobid, $cityid);
$insert_job = "INSERT INTO `#__js_job_jobs` ( `uid`, `companyid`, `title`, `alias`, `jobcategory`, `jobtype`, `jobstatus`, `jobsalaryrange`, `salaryrangetype`, `hidesalaryrange`, `description`, `qualifications`, `prefferdskills`, `applyinfo`, `company`, `country`, `state`, `county`, `city`, `zipcode`, `address1`, `address2`, `companyurl`, `contactname`, `contactphone`, `contactemail`, `showcontact`, `noofjobs`, `reference`, `duration`, `heighestfinishededucation`, `created`, `created_by`, `modified`, `modified_by`, `hits`, `experience`, `startpublishing`, `stoppublishing`, `departmentid`, `shift`, `sendemail`, `metadescription`, `metakeywords`, `agreement`, `ordering`, `aboutjobfile`, `status`, `educationminimax`, `educationid`, `mineducationrange`, `maxeducationrange`, `iseducationminimax`, `degreetitle`, `careerlevel`, `experienceminimax`, `experienceid`, `minexperiencerange`, `maxexperiencerange`, `isexperienceminimax`, `experiencetext`, `workpermit`, `requiredtravel`, `agefrom`, `ageto`, `salaryrangefrom`, `salaryrangeto`, `gender`, `video`, `map`, `packageid`, `paymenthistoryid`, `subcategoryid`, `currencyid`, `jobid`, `longitude`, `latitude`, `isgoldjob`, `isfeaturedjob`, `raf_gender`, `raf_degreelevel`, `raf_experience`, `raf_age`, `raf_education`, `raf_category`, `raf_subcategory`, `raf_location`, `serversatus`, `serverid`)

```

VALUES(" . \$employer_id . ", " . \$companyid . ", 'Senior Software Engineer', 'senior-software-engineer', '13', 1, 2, " '2', 0, 'You might be responsible for the replacement of a whole system based on the specifications provided by an IT analyst but often you'll work with "off the shelf" software, modifying it and integrating it into the existing network. The skill in this is creating the code to link the systems together.'

You'll also be responsible for:

- Reviewing current systems
- Presenting ideas for system improvements, including cost proposals
- Working closely with analysts, designers and staff
- Producing detailed specifications and writing the programme codes
- Testing the product in controlled, real situations before going live
- Preparation of training manuals for users
- Maintaining the systems once they are up and running

Most employers will want you to have a BTEC HND at the very least to get a foot in the door, however some companies runthat will consider candidates with AS Levels.

If you've got a degree it will , especially if it's in an IT, science or maths based subject.

If you've got a non-IT degree you might still be able to apply to a graduate trainee scheme, or you can take a postgraduate conversion course to get your CV up to scratch.

It is possible to move into software development from another profession. If this is you, play-up your business and IT experience and be prepared to take some IT-based courses if necessary.

The courses you'll find open most doors are of course the programming qualifications such as:

- Java
- C++
- Smalltalk
- Visual Basic
- Oracle
- Linux
- NET

Keeping up with the rapid pace of change is vital in this profession, so you should benefit from a good solid training programme, especially if you work for a larger organisation.

You'll learn from more senior programmers and will go on external courses to keep your professional skills up to date. Your training should focus on programming, systems analysis and software from recognised providers including the British Computer Society, e-skills, the Institute of Analysts and Programmers and the Institute for the

Management of Information Systems.</p>\r\n<p>All the software vendors, including Microsoft and Sun run accredited training too.If you are self-employed then you should invest in training to keep your skills.</p>', NULL, "", "", "", '69791', "", "", "", "", "0, 1, "", "", "" . \$date . ""', 0, '0000-00-00 00:00:00', 0, 2, 0, "" . \$date . ""', '2026-06-16 00:00:00', NULL, '1', 0, "", "", 0, NULL, 1, 1, 1, 1, 1, 'Bs(cs)', 6, 1, 12, 5, 5, 1, "", '126', 0, 4, 4, 8, 8, 0, "", NULL, NULL, 0, 50, 1, 'JnDLpkZB8', '74.3833333', '31.5166667', 0, 0, 0, 0, 0, 0, 0, 0, 0, NULL, 0);";

```

$db->setQuery($insert_job);
$db->query();
$jobid = $db->insertid();
$insertjobcities = $this->insertJobCities($jobid, $cityid);
$insert_job = "INSERT INTO `#__js_job_jobs` ( `uid`, `companyid`, `title`, `alias`, `jobcategory`, `jobtype`, `jobstatus`, `jobsalaryrange`, `salaryrangetype`, `hidesalaryrange`, `description`, `qualifications`, `prefferdskills`, `applyinfo`, `company`, `country`, `state`, `county`, `city`, `zipcode`, `address1`, `address2`, `companyurl`, `contactname`, `contactphone`, `contactemail`, `showcontact`, `nooffjobs`, `reference`, `duration`, `heighestfinishededucation`, `created`, `created_by`, `modified`, `modified_by`, `hits`, `experience`, `startpublishing`, `stoppublishing`, `departmentid`, `shift`, `sendemail`, `metadescription`, `metakeywords`, `agreement`, `ordering`, `aboutjobfile`, `status`, `educationminimax`, `educationid`, `mineducationrange`, `maxeducationrange`, `iseducationminimax`, `degreetitle`, `careerlevel`, `experienceminimax`, `experienceid`, `minexperiencerange`, `maxexperiencerange`, `isexperienceminimax`, `experiencetext`, `workpermit`, `requiredtravel`, `agefrom`, `aget`, `salaryrangefrom`, `salaryrangeto`, `gender`, `video`, `map`, `packageid`, `paymenthistoryid`, `subcategoryid`, `currencyid`, `jobid`, `longitude`, `latitude`, `isgoldjob`, `isfeaturedjob`, `raf_gender`, `raf_degreelevel`, `raf_experience`, `raf_age`, `raf_education`, `raf_category`, `raf_subcategory`, `raf_location`, `serverstatus`, `serverid`)

```

VALUES(" . \$employer_id . ", " . \$companyid . ", 'Web Designer', 'web-designer', '13', 1, 2, "", '2', 0, <p>An associate"s degree program related to web design, such as an Associate of Applied Science in Web Graphic Design, provides a student with a foundation in the design and technical aspects of creating a website. Students learn web design skills and build professional portfolios that highlight their skills and abilities. Common topics include:</p>\r\n\r\nFundamentals of design imaging\r\nBasic web design\r\nAnimation\r\nMultimedia design\r\nContent management\r\nEditing for video and audio\r\nMultimedia programming and technology\r\n\r\n<p>A bachelor"s degree program in multimedia or web design allows students to learn advanced skills needed for professional web design. Students develop artistic and creative abilities in addition to technical skills. Degree programs, such as a Bachelor of Science in Web Design and Interactive Media, cover:</p>\r\n\r\nDatabases\r\nWebpage scripting\r\nProgramming\r\nDigital imaging\r\nMultimedia design\r\nWeb development\r\n', "", \r\nWriting and editing content\r\nDesigning webpage layout\r\nDetermining technical requirements\r\nUpdating websites\r\nCreating back up files\r\nSolving code problems\r\n', NULL, "", "", "", '69795', "", "", "", "", "0, 1, "", "", "" . \$date . ""', 0, '0000-00-00 00:00:00', 0, 0, 0, "" . \$date . ""', '2026-06-16 00:00:00', NULL, '1', 0, "", "", 0, NULL, 1, 1, 1, 1, 1, 1, "3, 1, 5, 5, 5, 1, "", "0, 4, 4, 1, 1, 0, "", NULL, NULL, 0, 50, 1, 'JZH6Nz2cm', '74.3833333', '31.5166667', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NULL, 0);";

```

$db->setQuery($insert_job);
$db->query();
$jobid = $db->insertid();
$insertjobcities = $this->insertJobCities($jobid, $cityid); }
$check_resume_insert = "SELECT count(id) FROM `#__js_job_resume` where email_address='sampledata@info.com'";
$db->setQuery($check_resume_insert);
$sisresumeexists = $db->loadResult();
if ($sisresumeexists == 0) {
    $resume_query = "INSERT INTO `#__js_job_resume` ( `uid`, `created`, `last_modified`, `published`, `hits`, `application_title`, `keywords`, `alias`, `first_name`, `last_name`, `middle_name`, `gender`, `email_address`, `home_phone`, `work_phone`, `cell`, `nationality`, `iamavailable`, `searchable`, `photo`, `job_category`, `jobsalaryrange`, `jobsalaryrangetype`, `jobtype`, `heighestfinishededucation`, `status`, `resume`, `date_start`, `desired_salary`, `djobsalaryrangetype`, `dcurrencyid`, `can_work`, `available`, `unavailable`, `total_experience`, `skills`, `driving_license`, `license_no`, `license_country`, `packageid`, `paymenthistoryid`, `currencyid`, `job_subcategory`, `date_of_birth`, `video`, `isgoldresume`, `isfeaturedresume`, `serverstatus`, `serverid`)
    VALUES (" . $jobseeker_id . ", "" . $date . "", NULL, 1, NULL, 'sample data', 'sample data', 'sample data', 'First', 'Last', "", '1', 'sampledata@info.com', '1234567', '1234567', '1234567', '126', 1, 1, NULL, 13, 1, 2, 1, '1', 1, "", '0000-00-00 00:00:00', 1, 2, 1, NULL, NULL, NULL, "", NULL, NULL, NULL, 1, 1.5, 1, 48, '0000-00-00 00:00:00', "", 0, 0, NULL, 0);";
}

```

```

$db->setQuery($resume_query);
$db->query();
$resumeid = $db->insertid();
$resume_query = "INSERT INTO `#__js_job_resumeaddresses` ( `resumeid`, `address`, `address_city`,
`address_zipcode`, `longitude`, `latitude`, `created`, `last_modified`, `serverstatus`, `serverid`)
VALUES (" . $resumeid . ", 'Gujranwala', 39882, 52250, 74.3833333, 31.5166667, "".$date.", '0000-00-00
00:00:00', NULL , NULL);";
$db->setQuery($resume_query);
$db->query();
$resume_query = "INSERT INTO `#__js_job_resumeemployers` ( `resumeid`, `employer`,
`employer_position`, `employer_resp`, `employer_pay_upon_leaving`, `employer_supervisor`, `employer_from_date`,
`employer_to_date`, `employer_leave_reason`,
`employer_city`, `employer_zip`, `employer_phone`, `employer_address`, `created`, `last_modified`, `serverstatus`, `serverid`)
VALUES (" . $resumeid . ", 'Sample data', 'Software engineer', 'Project management', 65000, 'Sample',
"".$date.", "".$date.", 'No reason', 39882, 52250, '123456789', 'Sample address', "".$date.", '0000-00-00 00:00:00', NULL, NULL);";
$db->setQuery($resume_query);
$db->query();
$resume_query = "INSERT INTO `#__js_job_resumeinstitutes` ( `resumeid`, `institute`, `institute_city`,
`institute_address`, `institute_certificate_name`, `institute_study_area`, `created`, `last_modified`, `serverstatus`, `serverid`)
VALUES (" . $resumeid . ", 'Sample data', 39882, 'Sample data', 'Sample', 'Sample', "".$date.", '0000-00-00
00:00:00', NULL, NULL);";
$db->setQuery($resume_query);
$db->query();
$resume_query = "INSERT INTO `#__js_job_resumelanguages` ( `resumeid`, `language`,
`language_reading`, `language_writing`, `language_understanding`, `language_where_learned`, `created`, `last_modified`,
`serverstatus`, `serverid`)
VALUES (" . $resumeid . ", 'Sample data', 'Good', 'Good', 'Good', 'Sample place', "".$date.", '0000-00-00
00:00:00', NULL, NULL);";
$db->setQuery($resume_query);
$db->query();
$resume_query = "INSERT INTO `#__js_job_resumereferences` ( `resumeid`, `reference`, `reference_name`,
`reference_city`, `reference_zipcode`, `reference_address`, `reference_phone`, `reference_relation`, `reference_years`,
`created`, `last_modified`, `serverstatus`, `serverid`)
VALUES (" . $resumeid . ", 'Sample data', 'Sample name', 39882, 52250, 'Sample address', '123456789',
'Sample relation', 'Sample years', "".$date.", '0000-00-00 00:00:00', NULL, NULL);";
$db->setQuery($resume_query);
$db->query();
}
$check_jobapply_insert = "SELECT count(id) FROM `#__js_job_jobapply` WHERE cvid=" . $db->
>quote($resumeid);
$db->setQuery($check_jobapply_insert);
$isjobapplyexists = $db->loadResult();
if ($isjobapplyexists == 0) {
if (!isset($resumeid)) {
$query = "SELECT id FROM `#__js_job_resume` WHERE email_address='sampledata@info.com'";
$db->setQuery($query);
$resumeid = $db->loadResult();
}
$jobs = "SELECT id FROM `#__js_job_jobs` WHERE title='Web Designer' OR title='senior software
engineer' OR title='Accountant' OR title='Games Developer' OR title='Php Developer';";
$db->setQuery($jobs);
$jobids = $db->loadObjectList();
foreach ($jobids AS $jobid) {
$appliedjobs = "INSERT INTO `#__js_job_jobapply` ( `jobid`, `uid`, `cvid`, `apply_date`, `resumeview`,
`comments`, `coverletterid`, `action_status`, `serverstatus`, `serverid`)
VALUES (" . $jobid->id . ", " . $jobseeker_id . ", " . $resumeid . ", " . $date .
", '0, NULL, NULL, 1, NULL, NULL);";
$db->setQuery($appliedjobs);
}
}
}

```


ПРИЛОЖЕНИЕ Б

(справочное)

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

1 Область применения

Информационный ресурс «Центр карьеры БГУ» разрабатывается с целью облегчения поиска вакансии студентами. Разработанное ПО предназначено для внедрения в качестве информационного ресурса Белорусского Государственного Университета.

1.2 Уровень подготовки пользователя

Для полноценного использования информационного ресурса уровень подготовки пользователя должен быть приравнен к уровню рядового пользователя операционной системы «Microsoft Windows».

1.3 Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Перед использованием информационного ресурса пользователю необходимо ознакомиться со следующей документацией:

- правила использования ПК;
- данное руководство пользователя.

2 Назначение и условия применения

Правила запуска информационного ресурса:

- включить ПК;
- подключиться к локальной сети или сети Интернет;

- запустить веб-браузер;
- ввести в адресной строке адрес сайта;

При правильном запуске загрузится главная страница информационного ресурса.

3 Описание операций

Информационный ресурс состоит из следующих операций:

- авторизация/регистрация пользователя;
- просмотр вакансий;
- отправка отклика;
- создание сопроводительного письма;
- поиск вакансий;
- создание компании;
- создание вакансии;
- просмотр резюме;
- подтверждение компании;
- подтверждении публикации вакансии.

3.1 Регистрация и авторизация в системе

Для регистрации в системе необходимо выбрать пункт меню с изображением человека. В открывшемся меню выбрать пункт «Регистрация».

После чего произойдёт перенаправление на страницу регистрации.

Для регистрации необходимо корректно заполнить следующие поля: имя, логин, пароль, адрес электронной почты и выбрать группу пользователя. Последовательность действия показана на рисунке Б.1.

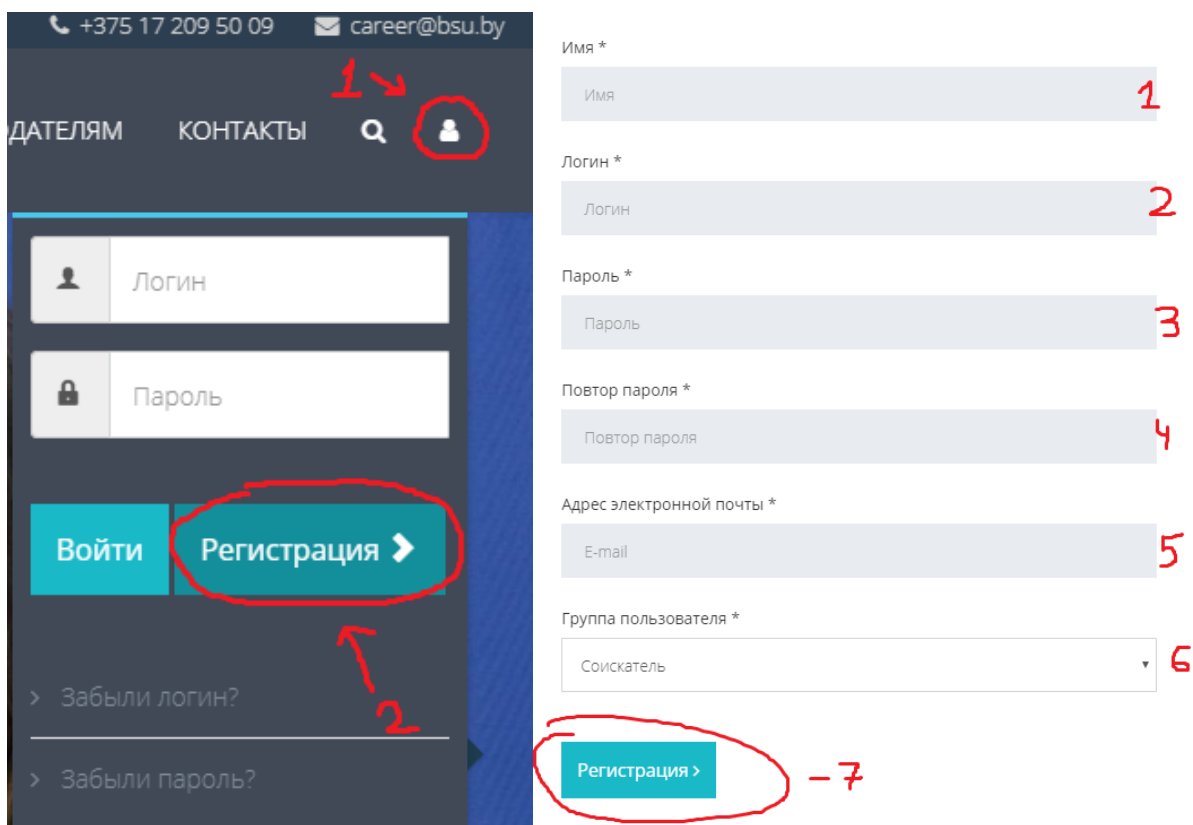


Рисунок Б.1 — Регистрация в системе

3.2 Просмотр и поиск вакансий

Для просмотра вакансий необходимо выбрать пункт главного меню «Вакансии». После нажатия отобразится страница со списком вакансий.

Для просмотра подробной информации о вакансии необходимо нажать на ее название. Последовательность просмотра вакансии показана на рисунке Б.2.

Вызов поиска возможен в 2-х вариантах:

- вызов из меню пользователя;
- нажатие по прозрачному значку справа.

Вызов поиска показан на рисунке Б.3.

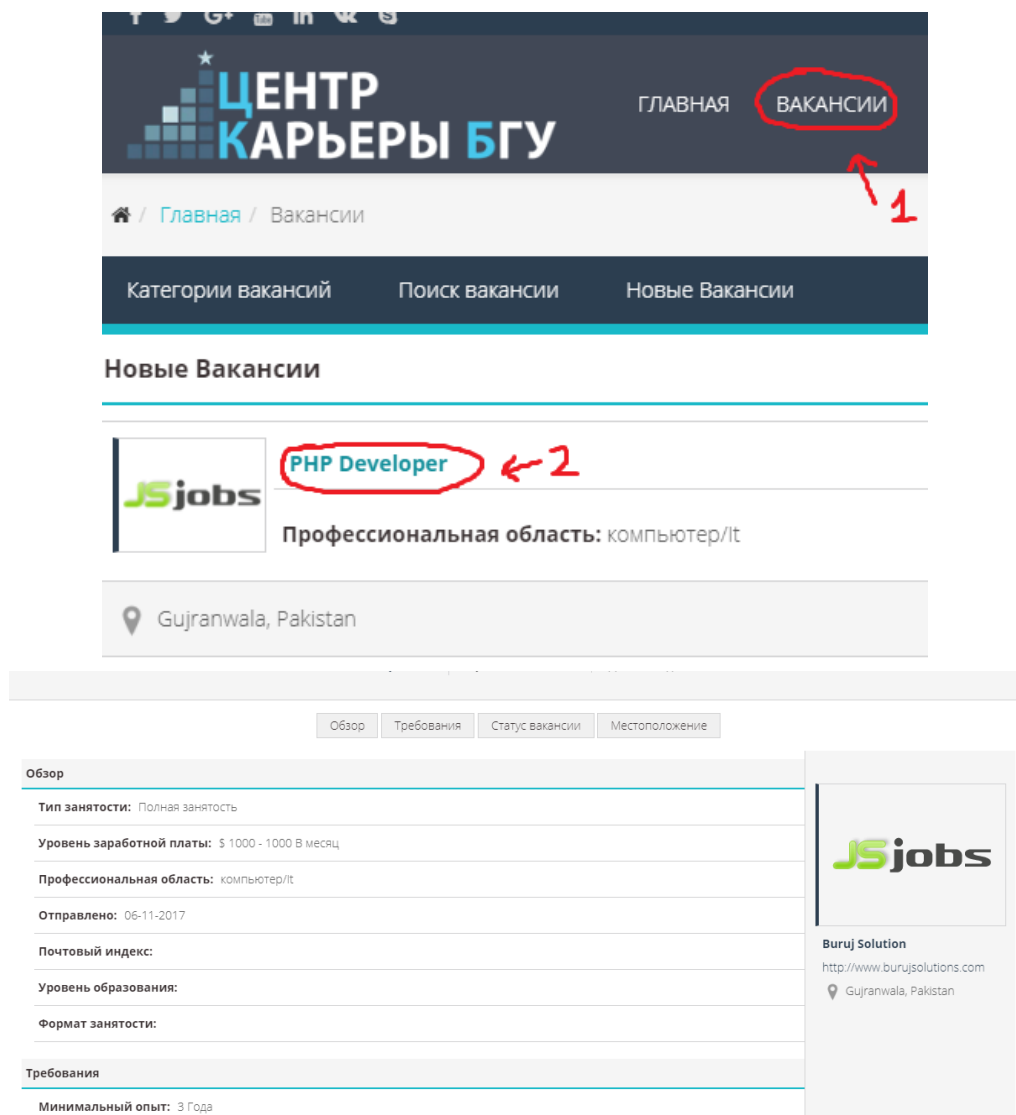
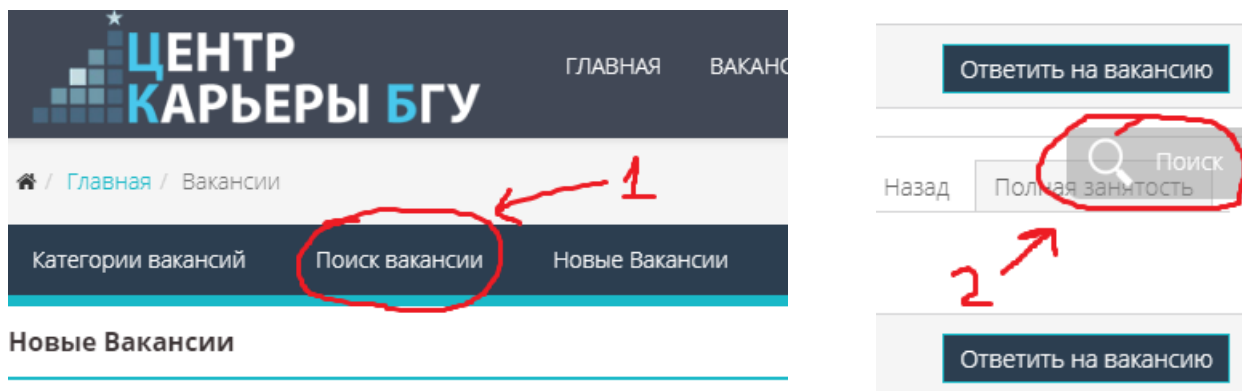


Рисунок Б.2 — Просмотр вакансии



Б.3 — Вызов окна поиска вакансии

3.3 Ответ на вакансию и создание сопроводительного письма

Для ответа работодателю достаточно нажать кнопку «Ответить на вакансию», расположенную рядом с описанием вакансии. Далее в открывшемся окне необходимо выбрать пункт «Ответить».

Для добавления сопроводительного письма необходимо в открывшемся ранее окне выбрать пункт «Отправить сопроводительное письмо». Последовательность действий показана на рисунке Б.4.

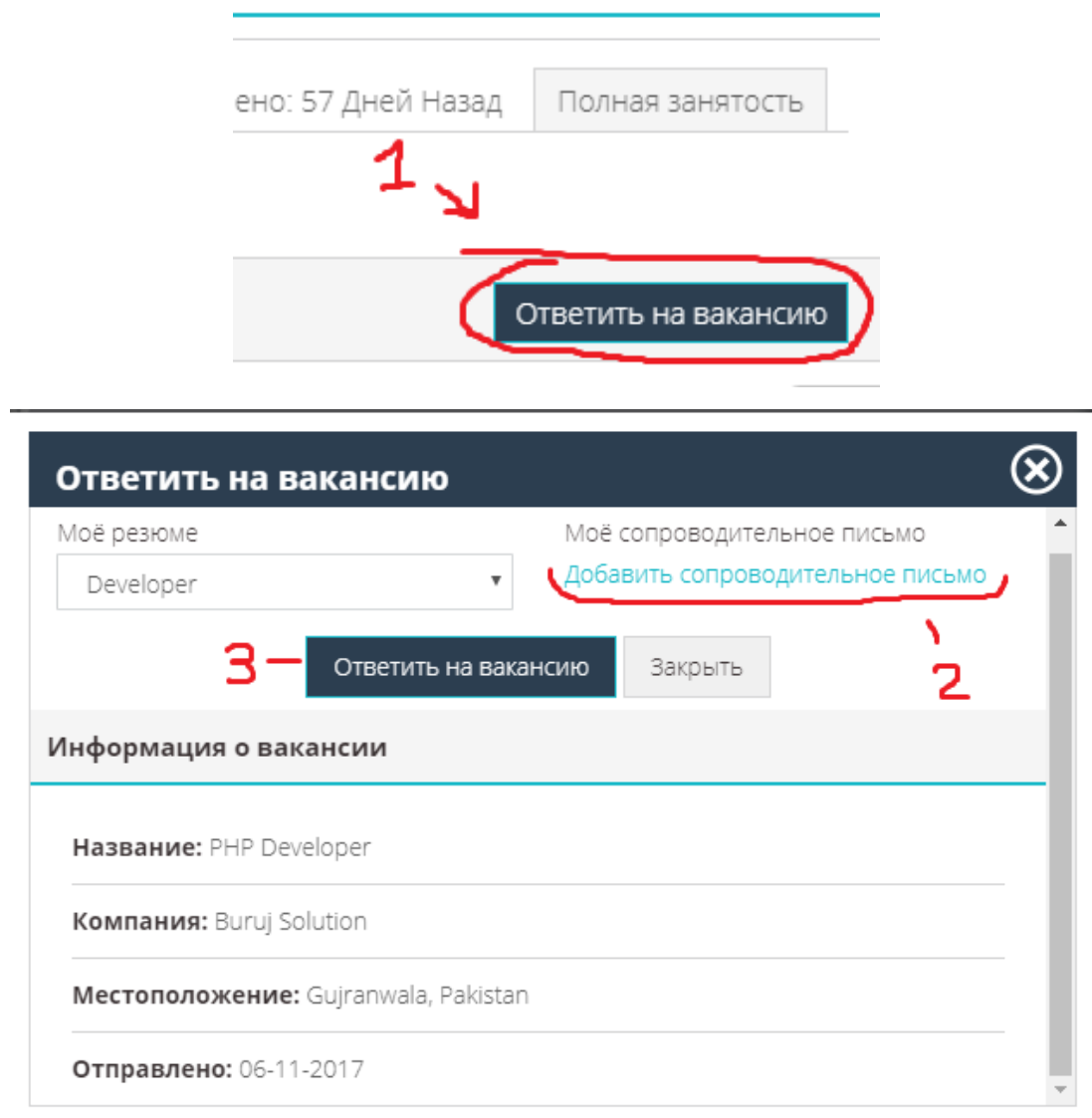


Рисунок Б.4 — Ответ на вакансии и создание сопроводительного письма

3.4 Создание компании

Для создания компании необходимо выбрать пункт меню «Работодателям» и в выпадающем списке выбрать «Личный кабинет». Находясь в личном кабинете необходимо выбрать пункт «Мои компании» и нажать кнопку «Добавить компанию». Вход в личный кабинет показан на рисунке Б.5.

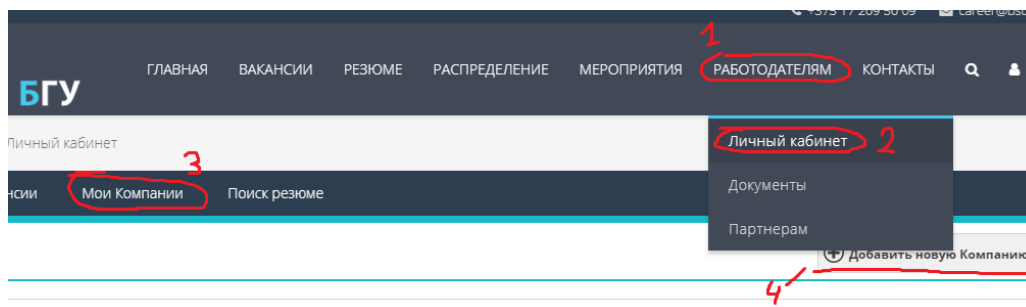


Рисунок Б.5 — Вход в личный кабинет

Далее необходимо заполнить форму и нажать кнопку «Сохранить», показанную на рисунке Б.6 и Б.7.

The image shows a form titled 'Новая вакансия' (New vacancy) with a sub-section 'Мои Компании' (My companies). The form is titled 'Информация о компании' (Company information). It contains several input fields: 'Название *' (Name), 'Контактное лицо' (Contact person), 'Контактный телефон' (Contact phone), 'Электронная почта' (Email), 'Категория *' (Category) with a dropdown menu showing 'компьютер/it', 'Веб-сайт' (Website), and 'Факс' (Fax). Each field has a corresponding input box.

Рисунок Б.6. — Создание компании

Почтовый индекс

Адрес

Логотип

Выберите файл | Файл не выбран

Разрешенные расширения: gif,jpg,jpeg,png
Максимальный размер файла (50КВ)

Сохранить Компанию

Рисунок Б.7 — Загрузка логотипа компании

3.5 Создание и редактирование вакансии

Для размещения вакансии необходимо выбрать пункт в личном кабинете «Мои вакансии». В открывшемся окне нужно нажать на кнопку «Разместить вакансию». На открывшейся странице нужно заполнить требуемые поля и нажать кнопку «Разместить». Последовательность действий показана на рисунке Б.8.

Следует отметить, что в разделе «Мои вакансии» пользователь может просматривать все ответы, направленные по конкретным вакансиям, а также при необходимости удалять их или редактировать (рисунок Б.9).

Новая вакансия | **Мои Вакансии** | Мои Компании | Поиск резюме

Мои Вакансии

Разместить вакансию

Название	Категория	Занятость	Статус вакансии	Компания	Создано	
	455	18-11-2017	Полная занятость	Компания: 01212	Профессиональная область : компьютер/it	Уровень образования:

Показано 1 - 1 из 1

20

Рисунок Б.8 — Создание вакансии

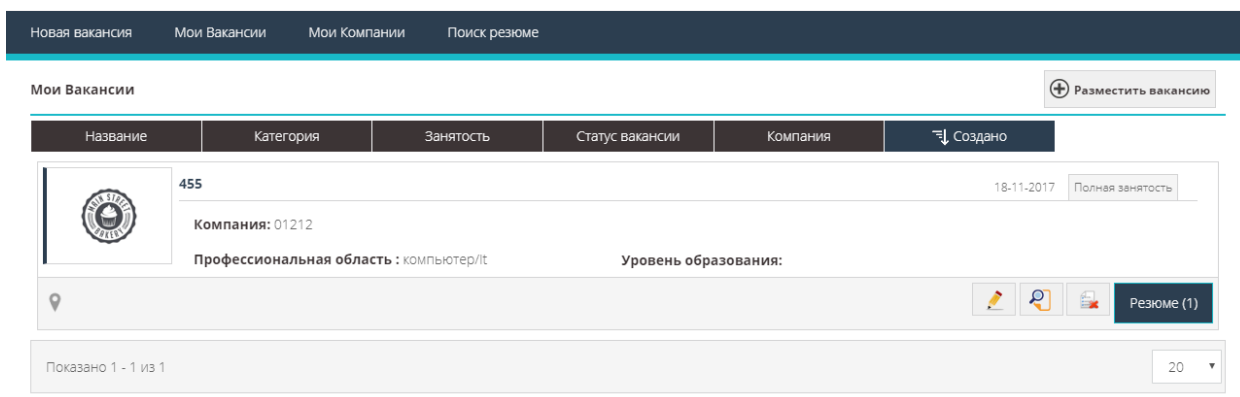


Рисунок Б.9 — Раздел «Мои Вакансии»

3.6 Подтверждение компании

Подтвердить регистрацию компании может только администратор.

Для подтверждения компании необходимо зайти в панель управления сайтом. Чтобы войти в панель управления нужно ввести в адресной строке браузера следующий адрес: <https://job.bsy.by/administrator>. После входа в панели управления необходимо выбрать компонент «JS Jobs». Последовательность действий показана на рисунке Б.10.

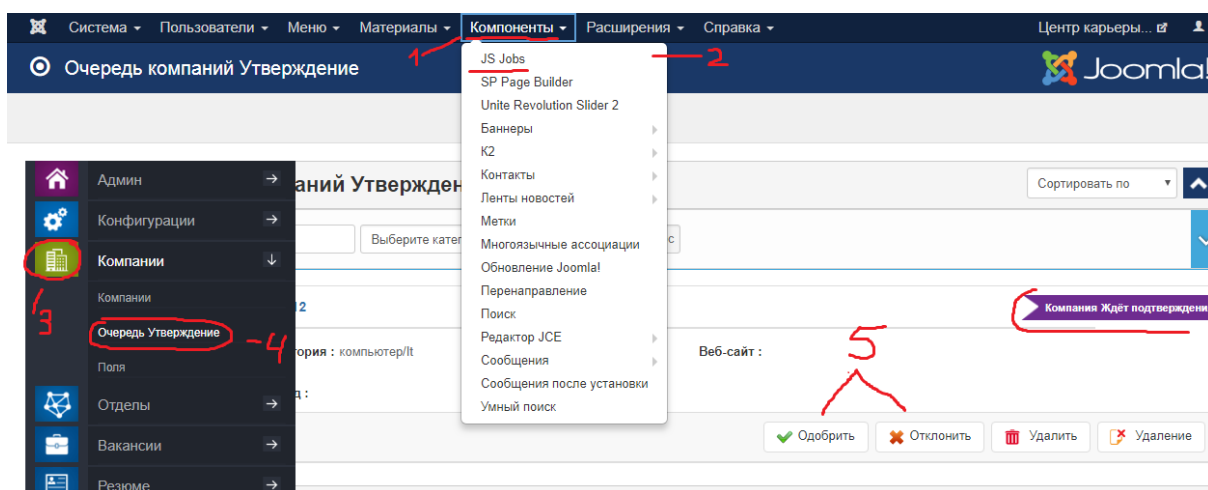


Рисунок Б.10 — Подтверждение компании

3.7 Подтверждение публикации вакансии

Механизм подтверждения созданной вакансии идентичен подтверждению компании. Пример подтверждения вакансии показан на рисунке Б.11.

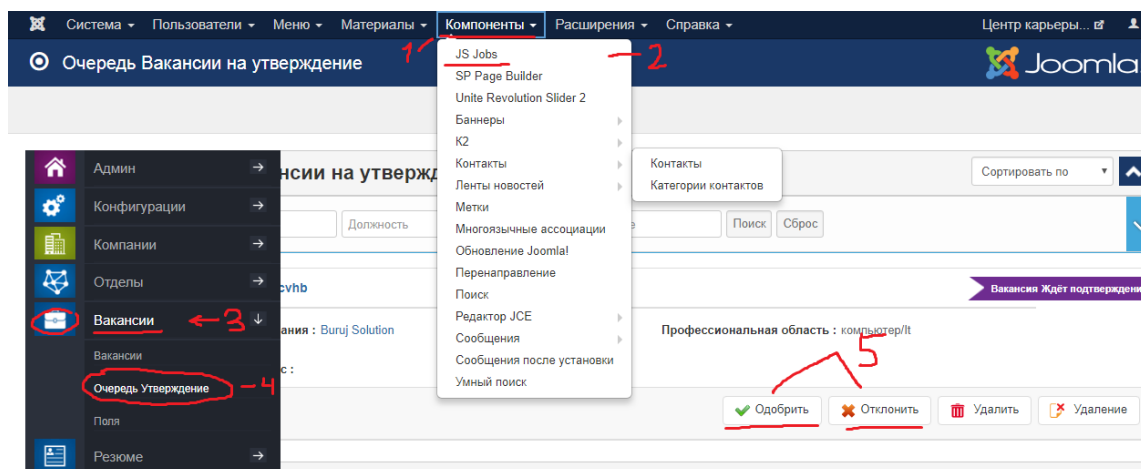


Рисунок Б.11 — Подтверждение вакансии

4 Аварийные ситуации

Некоторые аварийные ситуации:

- полный или частичный отказ технических средств приложения, включая сбои и отказы накопителей на жестких магнитных дисках;
- недостаточное количество встроенных системных библиотек.
- проблема отказа операционной системы от полноценной работы.
- отсутствие интегрированного пакета в браузер» JavaScript».

5 Рекомендации по освоению

На освоение информационного ресурса уйдёт минимальное количество времени, около 30 минут, если пользователь обладает всеми перечисленными выше требованиями.