|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| 1. **СПЕЦИАЛЬНЫЙ РАЗДЕЛ**    1. **Проектирование базового алгоритма решения задачи**   Цель курсового проекта – разработка web-приложения «Сервис для создания полиграфической наградной и рекламной продукции». Для достижения поставленной цели необходимо разработать приложение, состоящее из   * 4 веб-страниц: главная, страница разметки, личный кабинет пользователя и страница администрирования; * базы данных для информации о зарегистрированных на сайте пользователей; * файловой структуры для хранения изображений, загруженных пользователем, и изображений, используемых пользователем на сайте при разметке полиграфического изделия.   Для управления базой данных и осуществления операций над данными в ней, используется СУБД MySQL. MySQL – свободное кроссплатформенное ПО для управления реляционными базами данных. База данных содержит единственную таблицу «*users*», которая в свою очередь состоит из следующих полей:   * id – уникальный идентификатор пользователя; * login – логин пользователя, состоящий из цифр, букв латинского и русского алфавитов; * hash – результат двойного хеширования пароля пользователя методом «*md5*»; * name – имя пользователя, состоящее из букв латинского и русского алфавитов; * surname – фамилия пользователя, состоящая из букв латинского и русского алфавитов; * role – роль пользователя в приложении (администратор либо пользователь).   На основе вышеизложенного алгоритма была разработана схема работы системы, представленная в приложении А. | | | | | | |
|  | | | | | | |
|  |  |  |  |  | КП.09.02.03.ПКИПТ.297С.216.18 ПЗ | Лист |
|  |  |  |  |  | 12 |
| Изм | Лист | № документа | Подпись | Дата |
|  | | | | | | |
| * 1. **Описание программного продукта**   Данный программный продукт представляет собой web-приложение, взаимодействующее с базой данных. Клиентская часть приложения разработана с использованием следующий технологий:   * язык разметки HTML – стандарт для разметки документов во всемирной паутине. Подавляющее большинство веб-страниц содержат описание разметки на языке HTML. Язык интерпретируется браузером, а его 5 версия, используемая при разработке данного продукта поддерживается 97% браузеров; * язык описания внешнего вида документа CSS3 для поддержания специфического дизайна на высоком уровне. Методы каскадных талиц стилей находятся в классовой иерархии, что позволяет удобно манипулировать стилями объектов DOM; * скриптовый язык JavaScript – язык, позволяющий реализовать интерактивность web-приложений. Используя этот язык можно вносить изменения в структуру и содержимое web-страницы, изменять стили элементов, добавлять и удалять теги; * Библиотека jQuery, написанная на языке JavaScript, предоставляет возможность быстро, просто и удобно производить операции над элементами DOM. С ее помощью можно задавать элементам анимацию, скрывать и показывать диалоговые окна, отправлять на сервер асинхронные запросы; * AJAX - Asynchronous JavaScript and XML – технология, позволяющая реализовать динамическое обновление элементов страницы без перезагрузки всего содержимого. Позволяет производить операции в «фоновом» режиме, не влияя на работу пользователя на сайте. Также позволяет экономить время загрузки страницы, используя на ней минимум элементов, а остальное содержимое загружать при необходимости.   Серверная часть реализована на языке PHP. Работа с СУБД MySQL осуществляется средствами встроенного расширения mySQLi. Работа с графической частью изделия (отрисовка элементов) производится средствами расширения GD.  Разработанное приложение состоит из следующих модулей и файлов: | | | | | | |
|  | | | | | | |
|  |  |  |  |  | КП.09.02.03.ПКИПТ.297С.216.18 ПЗ | Лист |
|  |  |  |  |  | 13 |
| Изм | Лист | № документа | Подпись | Дата |
|  | | | | | | |
| * Папка *php* содержит серверную часть приложения: * *auth.php* – модуль, реализующий методы входа и регистрации, выхода, валидации (проверки на корректность) входных данных, методы получения текущего ID и роли пользователя; * *config.php* – модуль, реализующий методы загрузки, сохранения, восстановления файлов конфигурации. Эти файлы содержат в себе информацию о разметке изделия, созданной пользователем: положения элементов, их ID или содержимое и другие настройки; * *const.php* – файл, содержащий все используемые в web-приложении константы, в т.ч. различные стандартные значения, пути к файлам, шаблоны валидации и др. * *draw.php* – модуль, содержащий методы для отрисовки конфигурации на изображении указанного размера; * *files.php* – скрипт, содержащий реализацию класса File для упрощения работы с файлами и папками; * *image.php* – скрипт, содержащий реализацию класса Image для упрощения работы с изображениями; * *mysql.php* – скрипт, содержащий методы для создания и восстановления базы данных, безопасного открытия и закрытия соединения; * *mysql\_config.php* – файл, содержащий константы для подключения к базе данных: имя пользователя и пароль, имя базы данных, таблицы, имя хоста; * *rfc.php* – скрипт, обрабатывающий входящие HTTP-запросы и вызывающий в соответствии с запросом необходимый метод, реализованный в одном из модулей серверной части; * *segments.php* – модуль, реализующий методы для работы с элементами разметки: загрузка, добавление и удаление изображений; * *tools.php* – модуль, содержащий утилитарные методы для генерации UUID, получения информации о файловой системе сервера; * *user.php* – модуль, реализующий методы взаимодействия с данными пользователя: получение личной информации, списка конфигураций, получение | | | | | | |
|  | | | | | | |
|  |  |  |  |  | КП.09.02.03.ПКИПТ.297С.216.18 ПЗ | Лист |
|  |  |  |  |  | 14 |
| Изм | Лист | № документа | Подпись | Дата |
|  | | | | | | |
| Списка загруженных файлов, загрузка и удаление файлов с сервера;   * папка *CSS* содержит таблицы стилей для web-страниц: * *common.css* – общие для всех страниц стили; * *index.css* – стили для главной страницы; * *workbench.css* – стили для страницы разметки изделия; * *user.css* – стили для личного кабинета пользователя; * *admin.css* – стили для страницы администрирования; * папка *JS* содержит скрипты клиентской части: * *admin.js* – скрипт, отвечающий за адаптивность верстки страницы администрирования и динамическую загрузку данных с сервера; * *auth.js* – скрипт, реализующий методы запроса к серверу на вход и регистрацию, выход из аккаунта; * *color\_picker.js* – реализует объект «цветовой круг», используемый для определения цвета элемента; * *common.js* – содержит общий для всех страниц функционал, такой как: анимация появления/скрытия окон входа и регистрации, анимации иконок «вход», «регистрация», «личный кабинет», «выход»; * *config.js* – содержит методы для отправки запросов к серверу на получение и сохранение конфигурации. Также реализует механизм загрузки конфигурации на страницу (расположение элементов из конфигурации в видимой части страницы); * *index.js* – скрипт, отвечающий за адаптивность верстки главной страницы и динамическую загрузку данных с сервера; * *jquery.js –* библиотека *jQuery*; * *nice\_input.js* – реализует надстройку над стандартным тегом «input» для замены полей ввода на анимированные, со всплывающей подсказкой; * *type\_list.js* – реализует элемент «список типов» для представления информации о доступных элементах разметки по типам. Размещает доступные изображения по группам в скрываемые блоки. Открыт может быть только один блок; * *user.js* – скрипт, отвечающий за адаптивность верстки страницы личного | | | | | | |
|  | | | | | | |
|  |  |  |  |  | КП.09.02.03.ПКИПТ.297С.216.18 ПЗ | Лист |
|  |  |  |  |  | 15 |
| Изм | Лист | № документа | Подпись | Дата |
|  | | | | | | |
| кабинета и динамическую загрузку данных с сервера;   * *workbench.js –* скрипт, отвечающий за адаптивность верстки страницы разметки изделия и динамическую загрузку данных с сервера; * *empty\_configuration.json* – файл-шаблон для файла конфигурации; * *index.html –* главная страница; * *workbench.php* – страница разметки; * *user.php* – страница личного кабинета; * *admin.php* – страница администрирования.   1. **Отладка и тестирование**   Первичная отладка производилась в IDE Eclipse Neon.3. Тестирование и отладка производились в ручном режиме с использованием веб-сервера OpenServer в наиболее популярных браузерах:   * Opera; * Mozilla Firefox; * Yandex Browser; * Google Chrome; * Vivaldi.   В ходе тестирования были проверены функциональные требования. Тестирование производилось по следующим критериям:   * + реакция приложения на ввод некорректных данных;   + корректность отображения web-страниц в различных браузерах;   + проверка скорости отклика и загрузки данных.   Тестирование показало полное соответствие функциональным т нефункциональным требованиям. В результате тестирования был выявлен ряд недочетов, которые впоследствии были исправлены. Приложение работает корректно. | | | | | | |
|  | | | | | | |
|  |  |  |  |  | КП.09.02.03.ПКИПТ.297С.216.18 ПЗ | Лист |
|  |  |  |  |  | 16 |
| Изм | Лист | № документа | Подпись | Дата |
|  | | | | | | |
| **2.4 Руководство пользователя** | | | | | | |
|  | | | | | | |
|  |  |  |  |  | КП.09.02.03.ПКИПТ.297С.216.18 ПЗ | Лист |
|  |  |  |  |  | 17 |
| Изм | Лист | № документа | Подпись | Дата |