3 Реализация

3.1 Логическая структура web-приложения

Предметной областью данного дипломного проекта является обучающий онлайн-ресурс.

Разрабатываемое в рамках дипломного проекта программное средство должно выполнять следующие функции:

* хранение и отображение информации о пользователях;
* возможность создания пользователем своих проектов;
* возможность просмотра и изменения созданных проектов;
* возможность организации обратной связи.

На основании всего вышеперечисленного формируется модель предметной области.

Логическая структура сайта представляет собой набор тематических разделов, связанных между собой и включает в себя тематические страницы, дающие возможность удобного использования и быстрого доступа к необходимой информации:

* «Главная»;
* «Регистрация»;
* «Профиль»;
* «Проекты»;
* «О проекте».

Каждая из страниц содержит ссылки на другие разделы сайта.

3.2 Физическая структура web-приложения

Для организации физической модели данных будут созданы следующие таблицы:

Таблица 1 - Пользователи

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | login | varchar(255) |
|  | password | varchar(255) |
|  | role | int(11) |

В таблице «Пользователи» будет содержаться регистрационная информация пользователей, а так же их роль.

Таблица 2 - Пользовательские данные

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | Страна | varchar(255) |
|  | Телефон | varchar(255) |
|  | E-mail | varchar(255) |
|  | Пол | varchar(255) |

В данной таблице будут содержаться контактные данные пользователя.

Таблица 3 – Роль

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | role | varchar(255) |

В таблице «Роль» содержится перечень ролей обозначающих доступ пользователей.

Таблица 4 – Проекты

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | Id\_users | int(11) |
|  | Name\_project | varchar(255) |

В таблице «Проекты» содержится перечень всех проектов созданных пользователями.

Таблица 5 – Html теги

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | Name\_tag | varchar(255) |

В таблице «Html теги» содержится перечень html тегов.

Таблица 6 – Атрибуты тегов

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | Id\_tag | int(11) |
|  | Attribute\_name | varchar(255) |

В таблице «Атрибуты тегов» содержится перечень атрибутов тегов.

Таблица 7 – Значения атрибутов

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | Id\_ attribute | int(11) |
|  | Attribute\_value | varchar(255) |

В таблице «Значения атрибутов» содержится перечень возможных значений атрибутов.

Таблица 8 – Css свойства

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | Property\_name | varchar(255) |

В таблице «Css свойства» содержится перечень возможных css свойств.

Таблица 9 – Значение свойства

|  |  |  |
| --- | --- | --- |
| Ключ | Имя | Тип |
| \* | id | int(11) |
|  | Id\_property | int(11) |
|  | Property\_value | varchar(255) |

В таблице «Значение свойства» содержится перечень возможных значений css свойств.

В данном дипломном проекте используется популярная в настоящее время и диктуемая Yii архитектура MVC. MVC направлен на отделение бизнес-логики от пользовательского интерфейса, чтобы разработчики могли легко изменять отдельные части приложения не затрагивая другие. В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс (например, текст, поля ввода), а контроллер обеспечивает взаимодействие между моделью и представлением.

При старте приложения первым делом запускается логгер который проверяет наполненность таблиц тегов и свойств и в случае их отсутствия выдает ошибку. Также при первом запуске приложения логгер может создать необходимые таблицы со всеми связями и необходимыми данными.

Если ошибок не обнаружено, пользователю возвращается главная страница сайта, в верхней части которой пользователю предлагается войти в систему или зарегистрироваться.

За регистрацию и создание нового пользователя отвечает RegisterServlet который перераспределяет данные по нужным сервисам. За вход пользователя в систему отвечает LoginServlet который проверяет существование пользователя на портале.

Класс UserService необходим для работы с DAO классами. К нему передаются данные для проверки и регистрации пользователей. Слой DAO классов необходим для непосредственного обращения к базе данных, записи, чтения и тд.

3.3 Описание структуры и формата страниц

Для простоты работы с приложением был создан шаблон:

<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" xmlns="http://www.w3.org/1999/xhtml" version="2.0">

<jsp:directive.attribute name="title"/>

<jsp:directive.attribute name="script" fragment="true"/>

<jsp:directive.attribute name="css" fragment="true"/>

<html>

<head>

<title>${title}</title>

<jsp:directive.include file="/resources/assets/fragments/stylesheet.jspf"/>

<jsp:invoke fragment="css"/>

<jsp:invoke fragment="script"/>

</head>

<body>

<section id="container">

<jsp:include page="/resources/assets/fragments/header.jspf"/>

<jsp:directive.include file="/resources/assets/fragments/menubar.jspf"/>

<section id="content">

<jsp:doBody/>

</section>

<jsp:directive.include file="/resources/assets/fragments/footer.jspf"/>

</section>

</body>

</html>

</jsp:root>

* И также были созданы fragments которые будут вставляться в тег <jsp:doBody/>.

Благодаря этому готовая страница имеет вид:

<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"

xmlns:ad="urn:jsptagdir:/WEB-INF/tags"

xmlns="http://www.w3.org/1999/xhtml" version="2.0">

<jsp:directive.page contentType="text/html; ISO-8859-1; charset=UTF-8"/>

<ad:template title="Main page">

<jsp:attribute name="script">

</jsp:attribute>

<jsp:directive.include file="/resources/assets/fragments/content/index.jspf"/>

</ad:template>

</jsp:root>

Данное программное средство содержит 4 вспомогательные папки: fragments, javascripts, images, stylesheets.

В папке javascripts находятся страницы .js, используемые для взаимодействия HTML и JavaScript .

В папке stylesheets лежат CSS-стили необходимые для корректной работы верстки страниц.

В папке images расположены графические элементы, используемые в разрабатываемом web приложении.

В папке fragments расположены фрагменты страниц используемые в шаблонной системе для создания страниц.

Каждая страница web-приложения снабжена элементами навигации (ссылками) для удобства перемещения и может работать независимо от других страниц.

Работа с программой начинается с загрузки главной страницы index.jspx, откуда пользователь может получить доступ к другим страницам интернет-сайта посредством ссылок, организующих переход на эти страницы.

3.4 Описание используемых библиотек и элементов управления

В данном приложении очень тесно реализована интеграция с библиотекой jQuery и производной от нее библиотекой jQuery UI.

jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API по работе с Ajax.

Точно так же, как CSS отделяет визуализацию от структуры HTML, JQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчик события нажатия кнопки, управление передаётся JQuery, идентифицирующей кнопки и затем преобразовывающий его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого JavaScript.

Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в jQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, бо́льшая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована.