

ИСПОЛНЕНИЕ И МОДИФИКАЦИЯ ИНТЕРАКТИВНОЙ СОЦИАЛЬНОЙ СЕТИ

Акиншева И.В., Овсянникова И.П., Пантюхов В.А.
Могилевский государственный университет продовольствия
г. Могилев, Беларусь

Программирование сетевых приложений в среде JavaScript – это этап создания web-приложений, которые включают в себя html, CSS, JavaScript и один из серверных языков программирования (PHP, Java, Ruby, JavaScript, Go и другие). Актуальность исполнения интерактивной социальной сети с элементами JavaScript заключается в том, что большинство населения в настоящее время обмениваются данными и мыслями через сеть Интернет, а также развивается понятие о свободных средствах массовой информации в Интернет. Таким образом, все больше людей используют Интернет-площадку в качестве не только способа общения, но и решения определенных задач, в том числе и в сфере бизнеса.

Цель данной работы – создание социальной сети для обмена сообщениями и оценки суждений других людей («лайки»/«дизлайки»). Разработка программного обеспечения осуществляется с использованием языка гипертекстовой разметки HTML, каскадной таблицы стилей CSS, скриптового языка JavaScript, серверного скриптового языка программирования PHP, базы данных MySQL и Framework-a Materialize со стороны «клиента». Для разработки и агрегации используется среда WebStorm [1].

Поставленная цель достигается посредством решения следующих задач: создание понятного и информативного интерактивного интерфейса для пользователя; создание механизма «добавление в друзья» и обмена сообщениями; создание понятной структуры базы данных и грамотного управления данными в ней благодаря Workbench MySQL; создание дополнительных web-страниц, интерактивных элементов, которые могут помочь пользователю интуитивно работать в социальной сети.

Работа программы начинается с момента перехода по ссылке socialnet.loc. При этом браузер отправляет запрос на сервер, после которого возвращается «клиенту» файл, описывающий начальную страницу с заранее подключенными файлами стилизации. Далее для авторизации со стороны «клиента» необходимо ввести e-mail, пароль и нажать кнопку входа в приложение. При этом сервер обрабатывает запрос через специальные скрипты, написанные на языке PHP, берет данные из базы данных и генерирует ответ. В зависимости от валидности введенных данных (пароль должен быть введен латинскими буквами в количестве не менее 8 символов, e-mail должен иметь общепринятую структуру) ответ может быть либо в виде перехода на защищенную ссылку (перейти по ней на страницу приложения в браузере можно только авторизовавшись), либо в виде перезагрузки данной страницы с выводом информации об ошибке. Если введенная информация валидна, то сервер возвращает файл описывающий страницу личного кабинета авторизовавшегося пользователя с подключенными скриптами и файлами стилизации. Информация об авторизации данного пользователя записывается в локальную память со стороны «сервера» посредством sockie. По такому принципу работают и другие защищенные запросы от «клиента» к «серверу» [2].

Для работы с запросами от «клиента» к «серверу» используется архитектурный паттерн MVC («модель»-«представление»-«контроллер») – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных

компонента, таким образом, что модификация каждого компонента может осуществляться независимо.

Модель предоставляет данные и реагирует на команды контроллера, изменяя свое состояние; представление отвечает за отображение данных модели пользователю, реагируя на изменение модели; контроллер интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Чтобы зайти в приложение нужно запустить программу OpenServer для создания локального сервера и ввести в строку браузера `socialnet.loc`. Такие действия позволят отправить запрос на локальный сервер. В первую очередь запрос попадет в файл `routes/web.php`, который является точкой входа для любого запроса в данном приложении. Файл `routes/web.php` содержит имена всех роутеров и контроллеров, которые за них отвечают.

Разработанный файл `welcome.blade.php` является так называемым «view engine» – файл, который интерпретатор превращает на выходе в файл с расширением `.php`, предварительно вставив в него динамическую информацию и сгенерировав динамическую html-структуру. Так, например, для пользователя, написавшего два сообщения, сгенерируется структура, которая содержит два html-блока с различным контентом. После формирования окончательной html-структуры, интерпретатор возвращает файл с расширением `.html` и сервер возвращает данный файл пользователю.

В данном приложении реализован один из контроллеров в следующем формате: файл `UserController.php` представляет собой класс, наследующий класс `Framework-a Materialize` и реализующий различные интерфейсы; класс `UserController` инкапсулирует в себе публичные методы, задающие логику обработки запросов, привязанных конкретно к используемому контроллеру.

Модификация разработанной интерактивной социальной сети предусматривает изменение/улучшение следующих аспектов:

- 1) валидация данных со стороны «сервера»;
- 2) валидация данных со стороны «клиента»;
- 3) применение современных UI-стандартов со стороны «клиента»;
- 4) адаптивный интерфейс (адаптация происходит под ширину экрана используемого устройства: смартфон, планшет и пр.);
- 5) организация асинхронных запросов со стороны «сервера» при редактировании записей: оценивание постов (проставление «лайков»/«дизлайков») без перезагрузки страницы браузера;
- 6) защищенное соединение с административной панелью;
- 7) использование современного `Framework-a Materialize` со стороны «клиента»;
- 8) устранение несоответствий (bag) с целостностью данных в базе данных пользователей;
- 9) добавление анимационных эффектов со стороны «клиента»;
- 10) разработка системы модификации («уведомления» пользователя о получении новой информации на своей странице в социальной сети).

Полученный после модификации программный продукт является широкопрофильным и консолидирующим различные возможности современных программных средств на используемом паттерне MVC.

Литература

1. Стивенс У.Р. UNIX: разработка сетевых приложений / У.Р. Стивенс, Б. Феннер, Э.М. Рудофф. – СПб.: Питер, 2007. – 1039 с.
2. Бибо Б. jQuery. Подробное руководство по JavaScript / Б. Бибо, И. Кац; пер. с англ. – СПб.: Символ-Плюс, 2011. – 624 с.