

РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА RUBY ON RAILS

Катышева Т.М., Костюченко К.С.
Научный руководитель: Е.А. Мыцко
Томский политехнический университет
e-mail: tmk3@tpu.ru, ksk12@tpu.ru

Введение

В наше время роль интернет-магазинов сложно переоценить: удобство их использования по достоинству было оценено и закрепило свои позиции на рынке. Для обеспечения такого результата разработчиками было создано множество методов и инструментов проектирования, позволяющих технически правильно и с минимальными затратами времени создавать и тестировать веб-проекты. Фреймворки (frameworks) – сборники таких инструментов – активно используются разработчиками в качестве платформ для реализуемых веб-проектов.

Одним из популярных в наши дни фреймворком является Ruby on Rails (сокр. RoR). RoR представляет архитектурный образец Model – View – Controller (модель – представление – контроллер) для веб-приложений, а также обеспечивает их интеграцию с веб-сервером и сервером базы данных [1]. Так же в составе RoR есть отличные средства автоматизированного тестирования, что ускоряет переход проекта от стадии «программа написана» к стадии «программа работает без ошибок». Данные отличительные особенности RoR уже на начальных этапах проектирования значительно облегчают и увеличивают эффективность разработки веб-приложений.

Дополнительные средства разработки

При разработке веб-сайта также необходимы дополнительные средства разработки. Для описания разметки выбран CSS-фреймворк Susy, который полностью поддерживает RoR. В данном фреймворке используется язык SASS. В качестве дополнительных средств описания разметки используются язык HTML5 и каскадные таблицы стилей CSS3.

В качестве системы управления базами данных (СУБД) была выбрана SQLite, встроенная в RoR и не требующая дополнительных модулей для синхронизации. Преимуществами SQLite являются: высокая надежность, скорость и эффективность, полностью свободная лицензия.

В качестве дополнительных языков программирования для манипуляций со страницами используются языки PHP и JavaScript. Скриптовый язык PHP обладает большим набором встроенных средств для разработки веб-приложений, например, он взаимодействует с СУБД SQLite. Язык JavaScript имеет полную интеграцию с языками HTML и CSS. Также вместе с JavaScript для разработки веб-сайта

используется библиотека jQuery, в которую включена полезная для разработки технология AJAX. Данная технология заключается в «фоновом» обмене данными браузера с веб-сервером.

Структура сайта

Для определения структуры веб-сайта был проведен аналитический обзор сайтов сходной тематики. В результате на данном этапе проектирования была выделена структура интернет-магазина, состоящая из следующих блоков: каталог товаров, корзина, вход и регистрация, личный кабинет, типовые внутренние страницы и отзывы.

Из них особого внимания требуют: каталог товаров, корзина и обеспечение входа и регистрации пользователей и модераторов.

Блок «Каталог товаров»

Список товаров и их характеристики хранятся в базе данных.

Для работы с базами данных в RoR имеются такие инструменты как: миграции и валидация данных.

При помощи миграций записи об изменениях в базе данных производятся не с применением обычного SQL, а с помощью Ruby DSL. В результате миграции хранятся как файлы проекта, что позволяет в любое время совершить откат до желаемой миграции.

Под валидацией данных в RoR подразумевается проверка на приемлемость заносимых в базу данных элементов. При помощи валидатора могут быть введены требуемые условия для каждого из полей модели.

Для внедрения каталога товаров в структуру сайта необходима страница с формой для добавления нового товара, которой смогут пользоваться модераторы. Так же необходима страница с перечнем товаров в каталоге, где пользователям будет доступна краткая информация о каждом товаре, а модератором также будет доступна возможность удаления товаров из каталога и изменения их статуса (есть в наличии/нет в наличии). Кроме этого требуется страница для просмотра отдельного товара, где будет представлено его подробное описание и отзывы пользователей.

Блок «Корзина»

Для работы с корзиной необходим учет сессий. Сессия Rails представляет собой структуру, похожую на хэш, которая продолжает существовать от запроса к запросу [2]. В отличие от простых cookie-файлов, сессии могут содержать любые объекты. Таким образом, для хранения объектов корзины покупателя между запросами используются сессии.

Чтобы пользователи могли просматривать содержимое корзины, в структуре сайта предусмотрена соответствующая страница для отображения перечня товаров, их количества и стоимости. Так же имеется страница для оформления заказа, где расположены формы для указания реквизитов покупателя и предоставлен выбор способа оплаты. Последняя необходимая страница – это страница, на которой осуществляется оплата.

Функциональная возможность взимания платы с покупателя не встроена в RoR, но данный фреймворк включает в себя поддержку дополнительных gem-пакетов. Так, для осуществления возможности работы с кредитными картами и электронными кошельками был рассмотрен gem-пакет Active Merchant. Данное расширение является частью системы электронной коммерции Shopify.

Блок «Регистрации и авторизации»

Для эффективного построения решений задач данного блока, были рассмотрены следующие gem-пакеты: Devise и OmniAuth.

OmniAuth является библиотекой, которая стандартизует многопользовательскую аутентификацию для веб-приложений. Чтобы сделать возможной авторизацию через какую-либо социальную сеть, необходимо так же использование соответствующей версии OmniAuth (OmniAuth-Facebook, OmniAuth-Vkontakte и т.п.)

Devise представляет собой гибкий инструмент настройки аутентификации пользователей в системе Rails. Данный пакет включает в себя несколько независимых блоков, в числе задач которых были рассмотрены следующие:

- Защита данных: хэширование и хранение паролей в базе данных для проверки подлинности пользователя, пока он находится в системе.
- Поддержка OmniAuth.
- Поддержка системы отправки сообщений на электронный адрес пользователя для подтверждения регистрации/восстановления пароля.
- Обработка запросов о регистрации пользователя и предоставление возможности редактирования и удаления аккаунта.
- Напоминание пользователя путем сохранения cookie-файлов.

- Удаление сессий, неактивных в течение заданного промежутка времени.
- Блокировка аккаунта при достижении заданного количества неудачных попыток входа в систему.

Для разделения прав доступа пользователей и модераторов необходимо обеспечить поддержку входа в административную область, доступ к которой должен быть ограничен.

Осуществление данного разделение возможно путем создания двух контроллеров: контроллер сессии для поддержки входа в административную область и выхода из нее; контроллер для работы администраторов. Для ограничения доступа к области администрирования тем, кто не является администратором, используются имеющиеся в Rails обратные вызовы (callback). Они позволяют перехватить вызовы методов действий, добавляя свою собственную обработку перед тем, как методы будут реально вызваны [2].

В области администрирования предусмотрено размещение таких методов как: создание, обновление и удаление корзин, создание и удаление товарных позиций, создание заказов, просмотр списка пользователей, удаление пользователей.

Заключение

На данном этапе была разработана структура интернет-магазина и рассмотрены возможности ее программной реализации. В рамках решаемых задач были изучены возможности фреймворка Ruby on Rails, а также дополнительных модулей, интегрируемых в данный фреймворк. Кроме этого, были выбраны вспомогательные инструменты разработки.

В результате был завершен этап проектирования интернет-магазина и проведена подготовка для его реализации.

Список использованных источников

1. Веб-разработка с удовольствием [Электронный ресурс] / Ruby on Rails. – URL: <http://rubyonrails.ru> (дата обращения: 06.06.2016).
2. Руби С., Томас Д., Хэнсон Д. Rails 4. Гибкая разработка веб-приложений. – СПб.: Питер, 2014. – 448с.
3. Изучение Rails на примерах [Электронный ресурс] / Ruby on Rails Tutorial. – URL: <http://railstutorial.ru> (дата обращения: 06.06.2016).
4. Active Merchant. *Active Merchant*. Available at: <http://activemerchant.org> (accessed: 06.06.2016).
5. Devise. *GitHub*. Available at: <https://github.com/plataformatec/devise> (accessed: 06.06.2016).
6. OmniAuth. *GitHub*. Available at: <https://github.com/intridea/omniauth> (accessed: 06.06.2016).