

на 0,03 %, при том же размере суммарной величины крупных рисков и суммарной величины рисков на инсайдеров – юридических лиц соответствовал меньший размер достаточности нормативного капитала, достаточности основного капитала I уровня и достаточности капитала I уровня на 5,27 %, 6,24 % и 3,11 % соответственно.

Таким образом, в результате использования оболочного анализа данных (DEA) дана оценка эффективности функционирования банков Республики Беларусь по показателям безопасного функционирования по входным и выходным параметрам. В качестве входных параметров рассматривались достаточность нормативного капитала, достаточность основного капитала I уровня и достаточность капитала I уровня, а в качестве выходных параметров – суммарная величина крупных рисков, суммарная величина рисков инсайдеров – юридических лиц и физических лиц. Так, ОАО «Белапропромбанк», ЗАО «Банк ВТБ», ОАО «Технобанк» и ОАО «БНБ-Банк» получили оценки их технической эффективности, равные 1, следовательно, перечисленные банки являются эффективными и образуют границу эффективности для других банков. Также были предложены значения входных и выходных параметров для не эффективных банков, при которой они будут иметь оценку технической эффективности, равную, 1, которые не превышают размер нормативов, установленные Национальным Банком Республики Беларусь.

Литература

1 Кошелюк, Ю. М. Граничный анализ эффективности функционирования российских банков в период 2004 – 2005 годов / Ю. М. Кошелюк. – Москва : Изд. дом ГУ-ВШЭ, 2007. – С. 113–121.

2 Using Data Envelopment Analysis (DEA) program for measuring efficiency [Electronic resource] : Project Gyru. – Mode of access: <https://www.projectguru.in/publications/data-envelopment-analysis-dea/>. – Date of access: 09.04.2019.

УДК 004.4'2:004.738.5:339.176:004.3

А. С. Козичев

РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА “PROBOOK” С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVA

Данная статья посвящена особенностям разработки веб-приложений на языке программирования Java. В статье рассмотрены технологии, использованные при разработке приложения, и особенности работы с ними. Помимо этого также представлена функциональная составляющая приложения и приведены примеры работы. Также рассмотрены особенности работы с базами данных с целью оптимизации времени выполнения запросов и их количества.

Даже сейчас, несмотря на повсеместное распространение интернета и электронных книг, остается множество людей, которые предпочитают читать обычные бумажные книги. Однако поиск необходимой книги в магазинах может занимать достаточно большое количество дорогого нам времени. Именно по этой причине было разработано веб-приложение, которое позволяет быстро найти и купить необходимые книги, не выходя из дома.

Данное веб-приложение разработано с использованием языка программирования Java [1], который обладает большим выбором фреймворков и библиотек для решения любых задач. В частности, в процессе создания использовалось семейство фреймворков Spring [2], которое позволило упростить создание веб-приложений путем уменьшения шаблонного кода, который необходимо писать, и добавления принципа внедрения зависимостей.

При разработке веб-приложения были использованы следующие технологии: система управления базами данных H2 для тестирования, MySQL для хранения данных о книгах, пользователях и их корзинах, списков избранных, заказов. Технология Hibernate позволила упростить работу с базами данных с помощью использования принципа ORM, который значительно сокращает количество необходимого для работы с базами данных кода. С помощью фреймворка Spring Security были реализованы регистрация пользователей и распределение ролей, а также шифрование их паролей, что сделало приложение более безопасным.

Чтобы оставить отзыв на товар, добавить его в список избранных или оформить заказ, пользователю необходимо зарегистрироваться на сайте, создав свой аккаунт, после чего войти в свой личный кабинет (рисунок 1).

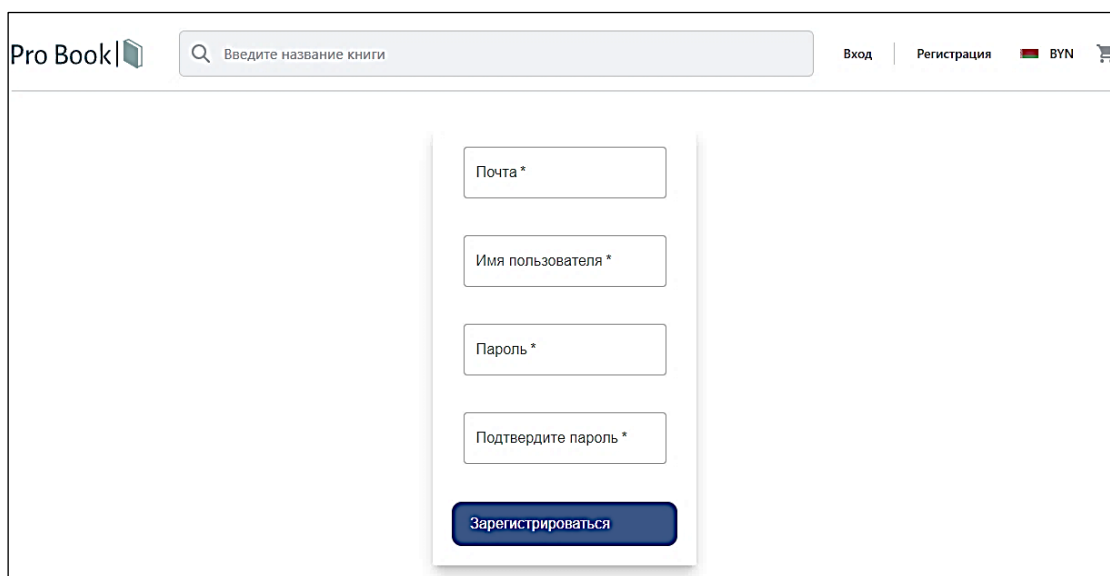


Рисунок 1 – Форма для регистрации

Далее пользователь переходит на главную страницу, с которой он может перейти к странице, содержащей каталог книг, а также меню, позволяющее произвести фильтрацию по множеству критериев и отсортировать книги желаемым образом (рисунок 2).

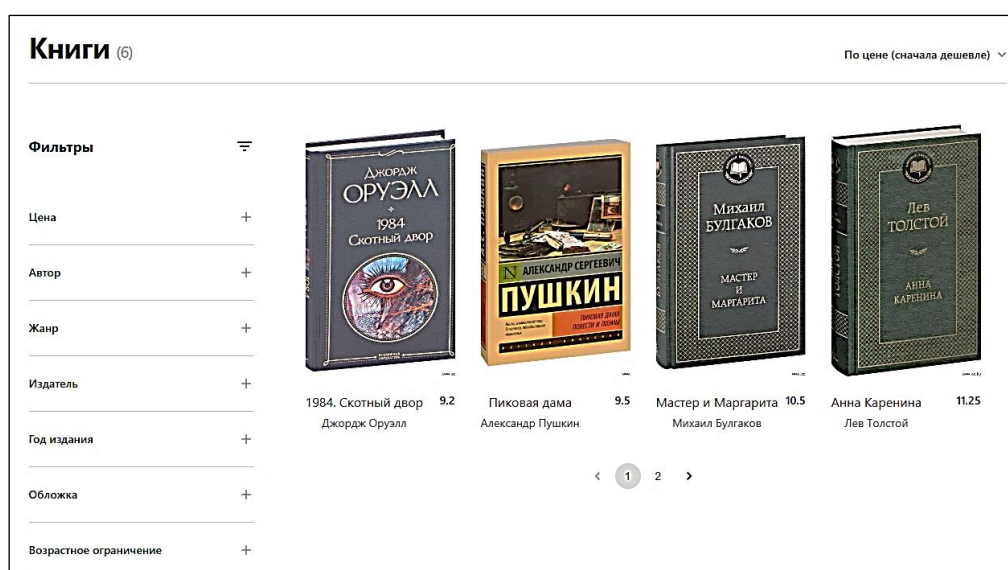


Рисунок 2 – Каталог товаров

Если пользователь проявит желание изучить товар более подробно, он может выбрать карточку товара, тем самым перейдя на его страницу, где он может изучить подробную информацию о книге, а также прочитать о ней отзывы (рисунок 3).



Рисунок 3 – Страница товара

Когда пользователь определился с выбором, то он может добавить товар в корзину выбором предназначенной для этого кнопки. Если же пользователь не уверен в выборе, то он может добавить товар в список избранных.

В корзине пользователь может изменить количество нужных ему книг и убрать ненужные, а также узнать общую стоимость покупок в корзине и перейти к оформлению заказа (рисунок 4).

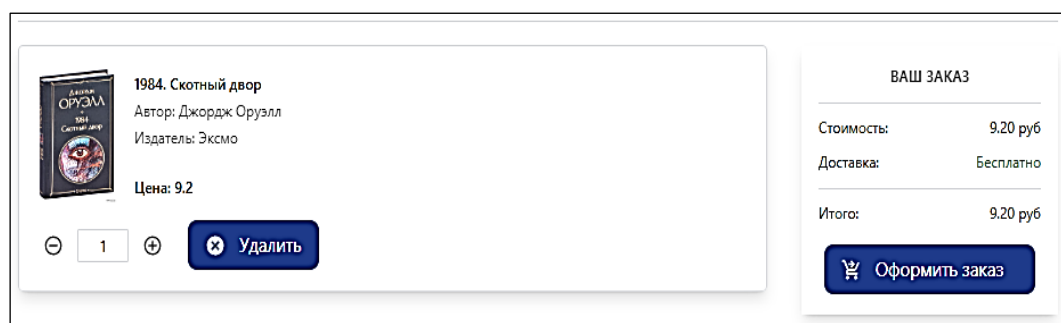


Рисунок 4 – Корзина

После этого пользователь может перейти к оформлению заказа или продолжить поиск необходимых товаров. Если же пользователь решил оформить заказ, то он переходит на страницу, где необходимо добавить адрес доставки или выбрать один из ранее добавленных (рисунок 5).

Рисунок 5 – Добавление адреса доставки

После выбора адреса доставки пользователь переходит на страницу, на которой он может проверить еще раз список товаров в заказе и проверить выбранный им адрес. В случае, если все верно, выбором кнопки «Оформить заказ» перед пользователем появляется сообщение об успешно оформленном заказе. Просмотреть информацию о заказе можно в профиле в разделе «Мои заказы», где отображается стоимость заказа, его статус, номер и дата оформления. Помимо этого предусмотрена фильтрация заказов по их статусу, что упрощает поиск нужного заказа среди ранее совершенных (рисунок 6).

Рисунок 6 – Просмотр списка заказов

Также предусмотрена админ-панель, которая позволяет администратору вести работу с БД непосредственно из приложения, изменяя статусы заказов пользователей.

Значительной частью данного приложения является работа над оптимизацией запросов и обращений к БД, что позволило значительно сократить время ожидания и загрузки страниц. Это достигается тем, что при разработке приложения для обмена данными между клиентом и сервером активно использовались DTO, которые позволили решить проблему избыточности возвращаемых данных и уменьшить объем передаваемых по сети данных. Также было использовано кэширование, предоставляемое ORM системой Hibernate, которое также значительно сократило время ответа от сервера.

Литература

- 1 Блинов, И. Н. Java from ERAM / И. Н. Блинов, В. С. Романчик. – Минск : Четыре четверти, 2020. – 561 с.
- 2 Крейг, Д. У. Spring в действии / Д. У. Крейг. – Москва : ДМК-Пресс, 2022. – 544 с.