

Доработка веб-приложения — интернет-магазина по продаже электронных книг

Введение

Вы — профессиональный разработчик веб-приложений на платформе Spring. Помимо основной работы java-программистом в одной из крупных IT-компаний в регионе (представьте себе компанию мечты), вы время от времени участвуете в разработке веб-проектов, как правило коммерческих. Это позволяет вам не только получать дополнительный доход, но и держать себя в профессиональном тонусе. Сейчас в свободное от основной работы время вы работаете над сайтом для книжного магазина, который недавно открылся в одном из центральных моллов вашего города.

Руководители магазина, ваши наниматели, хотят, чтобы магазин существовал не только физически, но и в виртуальном пространстве, и чтобы их покупатели могли приобретать и бумажные издания, и их электронные версии.

Дополнительный плюс для вас: на старте вы получили на 99.9% готовый фронтенд, разработку которого книжный магазин заказывал ранее у другого специалиста. Кроме этого, ваш заказчик подготовил техническое задание, описание структуры данных и даже API, что позволило вам сконцентрироваться в основном на бэкенд-составляющей проекта.

В данный момент бóльшая часть интернет-магазина уже реализована, вы почти готовы к финальной выкатке, остаются доработка и тестирование существующего функционала. Но в конце предыдущей рабочей недели руководители магазина инициировали внеочередной созвон-встречу с вами, где представили пакет комментариев и предложений по доработке и развитию интернет-магазина. По итогам встречи составлен список обязательных и дополнительных (необязательных, но рекомендуемых к выполнению ради вашего опыта) задач, над которыми вам предстоит поработать перед финальной сдачей проекта.

Задача 1

Цель задачи

Обработать данные о недавно просмотренном пользователем контенте.

Что нужно сделать

Спроектируйте и разработайте механизм сохранения данных о просмотрах книг пользователями и используйте эти данные:

- для дополнительной сортировки книг в ленте рекомендуемых на главной странице;
- в разделе «Популярное», что потребует доработки алгоритма расчёта популярности книги;
- в разделе «Недавно просмотренные», где будет выводиться список недавно просмотренных книг, этот раздел нужно создать.

Задача представляется комплексной, поэтому её стоит разбить на несколько этапов:

Этап 1

Определитесь со структурой данных, на которой будет работать процесс обработки информации о просматриваемых пользователями книгах. Если рассматривать задачу в плоскости имеющихся в нашем распоряжении абстракций, то нас интересует связь сущности, представляющей пользователя, и сущности, представляющей книгу. Мы хотим зафиксировать, для каких книг какие пользователи выполнили операцию «просмотр книги» — перешли на страницу `/books/slug.html`. Иными словами, нам необходимо вести журнал просмотров, который пользователи сами и будут заполнять, переходя по ссылкам на книги.

Этап 2

Теперь можно подняться на уровень выше и разобраться с логикой выборки популярных и рекомендуемых книг. При определении величин необходимых параметров для расчёта популярности книги в формуле вы должны учитывать влияние фактора недавно просмотренных книг. Кроме того, при проектировании необходимых алгоритмов следует заранее определить временной промежуток, в котором та или иная книга будет считаться недавно просмотренной пользователем.

Этап 3

После того как вы интегрируете аналитику по недавно просмотренным книгам в существующий комплекс данных, можно переходить на верхний уровень к разработке пользовательского интерфейса, представляющего список недавно просмотренных пользователем книг. На этом этапе во фронтенд будет необходимо добавить соответствующую страницу для вывода информации о недавно просмотренных книгах и провести рефакторинг безопасности, так как данный раздел должен быть доступен только авторизованным пользователям. В качестве шаблона страницы подойдёт `popular.html`.

Завершив этап, пришлите преподавателю ссылку на репозиторий или коммит для проверки.

Задача 2

Цель задачи

Привести функционал интернет-магазина в соответствие с техническим заданием.

Что нужно сделать

Один из приёмочных критериев при сдаче проекта в эксплуатацию — соответствие принимаемого ПО техническому заданию. В процессе разработки в этом документе многое может меняться как по инициативе заказчика, так и разработчика. Необходимо периодически актуализировать информацию о состоянии разработки по модулям системы через призму технического задания. В этом процессе мы тоже можем двигаться по уровням системы снизу вверх и разложить задачу на несколько этапов:

Этап 1

Проведите анализ структуры данных проекта и сверьте её с той, что представлена в техническом задании. На данном этапе необходимо понять, учитывает ли реализованная модель в полной мере все данные, представленные в техническом задании.

Для системного тестирования и демонстрации работы интернет-магазина, заполните базу данных книгами, пользователями,

просмотрами книг и т.д. Используйте для этого дополнительные миграции и [генератор mock-данных Mockaroo](#).

Отсутствующие и не имеющие аналогов в реализации модели необходимо реализовать и интегрировать в проект. После того как мы удостоверимся, что структура данных проекта по крайней мере покрывает необходимые модели по техническому заданию, можно подниматься на уровень выше для оценки и рефакторинга бизнес-логики проекта. Здесь пригодится SonarQube — см. цель задачи №4.

Этап 2

Проверьте, чтобы в разделе «Авторы» информация отображалась блоками. В разделе «Новинки» стоит обратить внимание на работу фильтра по датам, который позволяет выполнять поиск новинок по разным временным диапазонам. Также стоит обратить внимание на особенность вывода дерева жанров в разделе «Жанры», так как в техническом задании есть указание, что вложенность дерева может быть разных уровней, но на самых низких уровнях ссылки на жанры в интерфейсе пользователя должны располагаться в одной строке, а не списком-столбцом.

Этап 3

Уделите внимание облаку тегов и его алгоритму расчёта размера тега. Теги в облаке должны выводиться в виде блока ссылок разного размера, в котором размер шрифта ссылки должен соответствовать частоте использования того или иного тега. В личном профиле пользователя в разделе приобретённых книг (my.html) обратите внимание на переключатель «непрочитанное/архив». Удостоверьтесь в корректной работе механизма перемещения книг между корзиной и отложенными, а также удаления книг из этих контейнеров.

Этап 4

Проверьте работу системы рейтингов книг на соответствие техническому заданию. Не забудьте, что оценки на отзывы о книгах формируют рейтинг пользователя в системе интернет-магазина. Рейтинг пользователя представляет собой неотрицательное число типа double, которое можно представить на шкале из пяти звёзд в интерфейсе пользователя и рассчитать по следующей формуле:

$$P = \frac{\text{👍}}{\text{👍} + \text{👎}}$$

Этап 6

Завершающим этапом проверьте правильную работу всех навигационных ссылок, индикаторов и счётчиков в интерфейсе магазина. Обратите внимание на реализацию функционала формы обратной связи на странице «Контакты» (contacts.html) и в конце провести полный прогон разработанного ранее комплекса модульных и интеграционных тестов, чтобы убедиться в стабильной работе всех компонентов и модулей.

Формат сдачи материалов и оценивание

Готовый дипломный проект и его промежуточные итерации принимаются в формате git-репозитория с историей коммитов. Промежуточные итерации подлежат обсуждению с дипломным руководителем по мере появления вопросов и необходимости консультации. Готовый дипломный проект после утверждения руководителем подлежит защите перед комиссией преподавателей курса.

Рекомендации по выполнению

В работе над дипломным проектом руководствуйтесь [техническим заданием](#). Старайтесь формировать коммиты таким образом, чтобы их вклад в репозиторий был не слишком большим, когда историю составляют два коммита: initial и code_all_stuff. Но и слишком незначительный вклад коммита, вроде clean_old_comments, нежелателен и будет только захламлять общую картину. Также стоит придерживаться правила, что коммита в репозиторий достоин только проверенный на работоспособность код. Иными словами, если ваш новый commit ломает текущую стабильную версию, то лучше не делать такой коммит, сбросить внешний вид репозитория и сначала довести код до рабочего состояния на localhost. Старайтесь использовать в названии коммита ёмкие и не слишком длинные (в районе 30 символов) фразы. Детали можно указать в message-области коммита.