**Слайд «Введение» – Женёк**

Добрый день! Разрешите представить вам наш веб-каталог, предназначенный для ведения книжного фонда и упрощения работы персонала. На слайде вы видите основные цели проекта и краткую сводку возможностей. Я же хочу обратить внимание на то, что наше приложение разрабатывалось в первую очередь для автоматизации процессов, связанных с учётом и редактированием сведений о книгах, авторах и издательствах. Это даёт более удобный доступ к данным и, при необходимости, быстрый поиск, массовые операции и расширенное редактирование.

**Слайд «Анализ предметной области» – Дыма**

Здесь перечислены проблемы, которые наш проект решает, а также примеры уже существующих крупных систем (Koha, VuFind). Наша команда отметила, что эти комплексные решения хоть и мощные, но зачастую избыточны или сложны для внедрения в небольшом магазине. Поэтому мы сосредоточились на создании лёгкого в освоении каталога, который охватывает узкоспециализированные задачи, но при этом остаётся гибким и дружелюбным к пользователям без глубоких IT-навыков.

**Слайд «Методы и инструменты управления проектом» – Дыма**

Чтобы управлять разработкой, мы организовали процесс в GitHub и разделили задачи с помощью Kanban‑досок. Это позволило разбить функционал на мелкие шаги, видеть прогресс, планировать время, а также прозрачно контролировать статус каждой задачи. Такой подход упрощает взаимодействие и помогает вовремя находить узкие места.

**Слайд «Работа с GitHub» – Дыма**

На данном слайде описано, как мы использовали GitHub: у нас есть основная ветка, ветка разработки и ветки по функциональным направлениям (фронтенд, бекэнд, БД). При таком подходе каждый разработчик работал над своим куском кода в изолированной ветке. Затем через pull-request мы сливаем изменения обратно и проводим ревью, чтобы не нарушить существующий функционал. Это здорово помогло поддерживать чистоту кода и избежать конфликтов.

**104 КОММИТА. МЫ ДЕЙСТВОВАЛИ НАВЕРНЯКА**

**Слайд «Работа с Kanban доской» – Женёк**

Помимо GitHub, мы завели Kanban-доски, отражающие основные этапы: от постановки задач в очереди до тестирования и релиза. На слайде вы видите наши ключевые столбцы — например, «В работе» или «Ожидает тестирования». Это делает прогресс понятным как для разработчиков, так и для всей команды. Мы ставили приоритеты, распределяли исполнителей, а в итоге добивались чёткого понимания, на каком этапе сейчас находится каждая часть проекта.

**Слайд «Определение стека разработки» – Егориум**

Для реализации каталога мы выбрали Java и Spring Boot в качестве основы сервера. Также вы видите на слайде упоминание PostgreSQL как СУБД и Spring Data JPA для упрощённого доступа к данным. На фронтенде мы применили HTML, CSS, JavaScript и Bootstrap. Всё это — проверенные временем технологии, которые удобны в сопровождении и позволяют легко добавлять новые фичи.

**Слайд «Архитектура приложения» – Женёк**

Наша архитектура разделена на несколько слоёв, о чём кратко упомянуто на слайде. Но отмечу чуть подробнее:

1. **Слой данных (DAO)**: классы Book, Author, PublishingCompany и т. д. со связями OneToMany или ManyToMany.
2. **Слой доступа к данным (репозитории)**: для каждой сущности — свой интерфейс, наследующийся от JpaRepository, что упрощает CRUD-операции.
3. **Слой бизнес-логики (сервисы)**: тут мы инкапсулируем всю «умную» логику, например, логику массовых удалений или поиск с фильтрацией.
4. **Веб-слой (контроллеры)**: тут REST-эндпоинты, принимающие запросы и отвечающие JSON-данными.
5. **Слой представления**: наши HTML-страницы и скрипты, которые обращаются к контроллерам через AJAX.

Подобная многоуровневая структура упрощает поддержку и масштабирование.

**Слайд «Проектирование базы данных» – Никита + от себя (+ возможно Максим)**

На этом слайде мы перечислили ключевые таблицы: authors, publishing\_companies, books, authorships, styles, book\_styles и administrators. Каждая таблица имеет собственную цель. Например, authorships нужна, чтобы реализовать связь многие-ко-многим между автором и книгой, а book\_styles — между книгой и стилями. Важно, что все ссылки логически выстроены, и есть внешние ключи, чтобы поддерживать целостность данных.

**Слайд «Разработка пользовательского интерфейса (Frontend)» – Егориум + от себя**

Фронтенд ориентируется на Bootstrap: это позволяет быстро строить адаптивные формы и таблицы, а также придаёт единый стиль всему приложению. На слайде показаны основные элементы интерфейса: верхнее меню, формы поиска, таблицы для отображения списков. Мы добавили сортировку, фильтрацию, массовое удаление — всё это становится доступно буквально в пару кликов. Кодовая часть на JavaScript реализует отправку запросов к бекэнду, обрабатывает ответы и обновляет содержимое таблицы без перезагрузки страницы.

**Слайд «Разработка Backend» – Дыма**

Серверная часть написана на Spring Boot, где каждый тип сущности получил контроллер. Мы видим на слайде упоминание CRUD-операций. Расскажу чуть глубже: для книг это, например, GET /api/books (вернуть все книги), POST /api/books (создать новую книгу), PUT /api/books/{isbn} (обновить существующую) и DELETE /api/books/{isbn} (удалить книгу). Также есть специальные эндпоинты: search, bulk-delete и т. д. Всё это упростило работу с массивными операциями: например, массовая выгрузка/загрузка данных из CSV, которая тоже отражена на слайде. Эта часть позволяет хранить бэкапы или быстро импортировать данные.

**Слайд «Тестирование приложения» – Женёк**

Наконец, тестирование. Для нашего проекта были важны модули тестов: в репозитории есть тесты на уровне контроллеров и сервисов. Мы применяем JUnit5, который отображает в отчётах, всё ли корректно работает. Это помогает быстро отлавливать ошибки и проверять, что логика, например, при удалении автора, корректно чистит или сохраняет связанные записи. Фактически, мы сумели покрыть основные сценарии тестами и быть уверенными, что после каждого обновления функционал остаётся работоспособным.

**Завершение – Максим)**

Таким образом, представленный веб-каталог должен значительно упростить жизнь сотрудникам магазина: поиск, добавление, редактирование и удаление книг, авторов, издательств и стилей становятся максимально прозрачными и понятными. Если у вас возникнут вопросы, я буду рад на них ответить. Спасибо за внимание!