|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №6**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Тема практической работы:**

**Студент группы** ИКБО-33-21 Май Ван Туан

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Волков М.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Москва 2023

СОДЕРЖАНИЕ

[**1. Цель работы** 3](#_Toc145190172)

[**2. Ход работы** 3](#_Toc145190173)

[**3. Ответы на вопросы** 5](#_Toc145190174)

## Постановка задачи

Студенту предлагается реализовать бизнес логику ко второй практике по Java. Нужно добавить к существующему приложению реализацию «корзины».

Пользователь должен иметь такие возможности:

1. Добавить товар в корзину.
2. Удалить товар из корзины.
3. Изменить количество товара в корзине.
4. Посмотреть всю корзину.
5. Оформить заказ и очистить корзину.

При добавлении товара в корзину должна быть проверка, что товар есть в наличии. Так же обработать случай, что товар добавили в корзину, после товар закончился на складе и оформить заказ невозможно.

Сервис должен с помощью docker-compose. Должно быть использовано паттерны проектирования: Чистая архитектура, MVC.

## Ход работы

Для начала создадим все нужные файлы для создания конфигурации серверного ПО (рис. 1).

A screenshot of a computer

Description automatically generated

1. Структура образа

Содержимое всех необходимых файлов показано на рисунках 2-8.

A computer screen shot of a black screen

Description automatically generated

1. Файл application.properties

A black rectangular object with a black border

Description automatically generated

1. Файл Dockerfile

A screenshot of a computer program

Description automatically generated

1. Файл docker-composer.yml

A screen shot of a computer program

Description automatically generated

1. Файл Book

A screen shot of a computer

Description automatically generated

1. Файл BoolRepository

A screen shot of a computer program

Description automatically generated

1. Сборка BookController

A screen shot of a computer program

Description automatically generated

1. Сборка BookView

Перейдём по адресу http://localhost:8080 (Рисунок 9 -11).

A screenshot of a computer

Description automatically generated

1. Страница Shopping App для клиентов

A screenshot of a computer

Description automatically generated

1. Страница Warehouse для администрации

A screenshot of a computer

Description automatically generated

1. Страница Books для администрации

## Ответы на вопросы:

* **Аннотация Service:**

Аннотация Service используется для пометки классов в Java как сервисов, которые могут предоставлять бизнес-логику, обработку данных и другие сервисные функции. Это помогает фреймворкам и инструментам легче управлять зависимостями и выполнением кода.

* **Паттерн MVC:**

MVC (Model-View-Controller) — это архитектурный паттерн, разделяющий компоненты приложения на три основные части: Model (модель), View (представление) и Controller (контроллер). Это улучшает поддерживаемость, расширяемость и переносимость кода.

* **Паттерн «Чистая архитектура»:**

Чистая архитектура – это методология проектирования программного обеспечения, предлагающая структурировать приложение так, чтобы его зависимости шли от внутренних слоев к внешним. Основные принципы включают разделение кода на уровни (Entities, Use Cases, Interface Adapters, Frameworks and Drivers) и независимость от внешних фреймворков.

* **Слои RestFull приложения:**
* Слой представления (Presentation): Отвечает за отображение данных пользователю. Включает в себя контроллеры, представления и шаблоны.
* Слой бизнес-логики (Business Logic): Содержит бизнес-правила и функциональность приложения. Здесь реализованы Use Cases.
* Слой данных (Data): Отвечает за доступ к данным и базе данных. Включает репозитории и источники данных.
* Слой инфраструктуры (Infrastructure): Обеспечивает взаимодействие с внешними системами и инфраструктурными компонентами. Включает в себя фреймворки, библиотеки и инструменты.

Каждый слой имеет свою ответственность, что обеспечивает легкость сопровождения, расширения и тестирования приложения.