**Документация на Проект "Онлайн Магазин"**

Описание

Проектът „Онлайн магазин“ е уеб базирани приложение , което позволява на потребителите да разглеждат и купуват различни продукти онлайн. Целта на проекта е да предостави удобна и сигурна платформа за онлайн пазаруване.

Функционалности

* Регистрация и управление и вход на потребителски и админски профили
* Добавяне, редактиране и изтриване на продукти, които се предлагат в магазина
* Разглеждане на всички налични продукти
* Разглеждане на всички продукти част от кампания с промоции и разпродажби
* Възможност за закупуване на продукт
* Реализация като клиент-сървър приложение
* Следене за наличност на продукт
* Възможност за намаления на продукти
* Добавяне на продукти в количката за покупки

Технологии

* Front-end – Java, HTML
* Back-End – Java (Spring Boot)
* Thymeleaf
* N-Layer Design
* Model View Controler (MVC)
* Object**-**RelationalMapping **(**ORM**)**

Разяснения

* **Spring boot** - Spring Boot е иновативна технология в областта на Java, която цели да опрости процеса на създаване на приложения. Spring Boot позволява лесно създаване на самостоятелни, готови за изпълнение приложения. Това означава, че приложението се съдържа в едно JAR (Java Archive) файл, който може да бъде стартиран директно с Java команда, без необходимост от развитие на сложни конфигурации за сървъри. Spring Boot включва вградени зависимости за общи библиотеки и компоненти, необходими за създаването на различни видове приложения (например, уеб приложения, бази данни и др.). Това позволява лесно добавяне на функционалности, като просто добавяте нужните зависимости в конфигурационния файл. Spring Boot използва автоматична конфигурация, която се опитва да предвиди нуждите на вашето приложение. Това включва предварително конфигурирани настройки за много често използвани библиотеки и технологии, които могат да бъдат променяни или заменяни според нуждите на разработчика. Spring Boot включва вграден уеб сървър (например Tomcat, Jetty или Undertow), което означава, че не се изисква отделна инсталация на сървър за изпълнение на уеб приложения. Spring Boot е мощен инструмент за създаване на Java приложения, който значително опростява процеса на разработка, като предоставя гъвкавост и производителност в едно.
* **Thymeleaf** - Thymeleaf е модерен шаблон за уеб приложения, който се използва често в Java и Java-based приложения, особено в контекста на уеб разработката със Spring Framework. Thymeleaf използва HTML-подобен синтаксис, който го прави лесен за усвояване и използване от уеб разработчиците. Той е проектиран да бъде интуитивен и да се интегрира добре с HTML и CSS. Thymeleaf се интегрира перфектно със Spring Framework чрез модул, наречен "Spring Boot Starter Thymeleaf". Това позволява лесното му използване във Spring Boot приложения, като предоставя вградена поддръжка за много от функционалностите му. Thymeleaf позволява вграждането на данни от моделите (Model) на приложението директно в HTML шаблоните. Той поддържа изрази (Expressions), които могат да бъдат използвани за показване на данни, логика и променливи. Thymeleaf предлага богат набор от вградени функции и операции, които могат да бъдат използвани за манипулиране и форматиране на данни в шаблоните. Това включва операции за итерация през списъци, условни оператори, форматиране на дати и други. Thymeleaf поддържа концепцията за фрагменти, което позволява разделянето на шаблоните на повторно използваеми части. Това е полезно за постигане на компонентен подход в уеб разработката. Thymeleaf е популярен избор за уеб разработчиците в света на Java, особено в средата на Spring приложенията, поради своята лесна интеграция, мощни възможности и приятен за използване синтаксис.
* **N-Layer Design** - N-Layer Design (също известен като многослойна архитектура) е подход за проектиране на софтуерни приложения, при който приложението е разделено на различни слоеве или нива, като всяко ниво има определена функционалност и отговорности. Съдържа следните слоеве:

1. Presentation Layer (слой за представяне):

- Това е нивото, което се грижи за визуалното представяне и комуникацията с потребителите или други системи.

- Тук се намират интерфейсите като уеб страници, API ендпойнти или графични потребителски интерфейси (GUI).

2. Business Layer (бизнес слой):

- Този слой съдържа бизнес логиката на приложението.

- Тук се извършват обработки, правила, валидации и операции, свързани с бизнес дейността.

3. Data Access Layer (слой за достъп до данни):

- Този слой е отговорен за комуникацията със съхранението на данни (например база данни, файлове и други източници).

- Тук се използват операции за четене, запис и манипулиране на данни.

4. Infrastructure Layer (слой за инфраструктура):

- Този слой предоставя общи служби и компоненти, използвани от другите слоеве.

- Тук се намират конфигурации, сервизи за сигурност, логове, управление на сесии и други.

N-Layer Design се стреми да раздели приложението на добре дефинирани и отделни слоеве, които взаимодействат помежду си чрез добре дефинирани интерфейси. Този подход подобрява разширяемостта, поддържането и тестването на приложението, като позволява разработчиците да се концентрират върху специфичната функционалност на всеки слой.

* **Model-View-Controller (MVC)** - е архитектурен шаблон за проектиране на софтуерни приложения, който се използва широко за уеб разработка и не само. Той разделя апликацията на три основни компонента:

1. Модел (Model):

- Моделът представлява данните и бизнес логиката на приложението.

- Тук се обработват и съхраняват данните, които се използват от приложението.

- Моделът не е пряко свързан с изгледа или потребителския интерфейс.

2. Изглед (View):

- Изгледът представлява потребителския интерфейс на приложението.

- Той е отговорен за визуалното представяне на данните, предоставяне на интерактивност и взаимодействие с потребителя.

- Изгледът не съдържа бизнес логика, а просто отразява данните от модела.

3. Контролер (Controller):

- Контролерът поема ролята на посредник между модела и изгледа.

- Той приема заявки от потребителя чрез изгледа, извиква съответните операции в модела за обработка на данните и определя какво трябва да бъде показано в изгледа.

- Контролерът се грижи за маршрутирането на заявките и за осигуряването на правилното взаимодействие между модела и изгледа.

MVC разделя отговорностите в приложението така, че всяка част да има ясно дефинирани функции и да е лесно поддръжаема и разширяема. Този подход позволява разделение на кода за бизнес логика, потребителски интерфейс и логика за управление, което е основа за много модерни уеб и приложения.

* **Object-Relational Mapping (ORM)** е техника в програмирането, която позволява свързването на обекти от програмния код с данни от релационна база данни. Основната идея на ORM е да се предостави абстракция, която позволява работата с обекти в програмния код, като тези обекти автоматично се превръщат в записи в базата данни и обратно.

Инсталация

A screenshot of a computer

Description automatically generated

Използване

1. Регистрация в системата
2. Вход като админ или обикновен потребител и използването на съответното меню
3. Разглеждане на каталога с продукти
4. Добавяне на продукт в продуктовата количка
5. Финализиране на пазаруването

Изготвили:

Любомир Рачев,

Александър Николов,

Доан Али,

Валери Мирчевски,

Иван Андонов,

Румен Калчев.