**Технически университет – София**

**Факултет Приложна Математика и Информатика**

**Катедра:**

**Информатика и Софтуерни науки**



Учебна дисциплина „[Приложно програмиране с Java 2024/2025 (ИСН)](https://fpmi.bg/moodle/course/view.php?id=670) “

**Функционално описание и софтуерна архитектура на проект**

**„Система за онлайн магазин за обувки“**

**Изготвил: Любомир Славчев №471222077**

**Документация на проект**

**Описание на проекта**

„MyModelProject“ е уеб приложение, създадено с **Spring Boot** и използващо **JPA** за управление на базата данни. Проектът има за цел да моделира взаимоотношения между книги, автори и издатели чрез създаването на релационни таблици в базата данни. Той използва PostgreSQL като система за управление на базата данни и предоставя RESTful API за извършване на CRUD операции върху моделите.

**Основни функции:**

* Управление на книги, автори и издатели.
* Свързване на книги с автори (много към много връзка).
* Свързване на книги с издатели (много към едно връзка).

**Използвани технологии**

* **Java 17** – програмният език.
* **Spring Boot 3.1.0** – основен фреймуърк за разработка на уеб приложения.
* **Spring Data JPA** – за работа с релационни бази данни чрез JPA.
* **PostgreSQL** – система за управление на базата данни.
* **Maven** – инструмент за управление на зависимостите.
* **HQL (Hibernate Query Language)** – език за заявки към базата данни.

**Структура на базата данни**

Проектът използва три основни таблици:

**1. Author**

Таблицата author съдържа информация за авторите.

* **id** (bigserial, primary key) – уникален идентификатор на автора.
* **name** (varchar) – името на автора.

**2. Book**

Таблицата book съдържа информация за книгите.

* **id** (bigserial, primary key) – уникален идентификатор на книгата.
* **title** (varchar) – заглавие на книгата.
* **publisher\_id** (bigint, foreign key) – връзка с издателя.

**3. Publisher**

Таблицата publisher съдържа информация за издателите.

* **id** (bigserial, primary key) – уникален идентификатор на издателя.
* **name** (varchar) – името на издателя.

**4. author\_book (съединителна таблица)**

Таблицата author\_book съединява авторите с книгите (много към много връзка).

* **book\_id** (bigint, foreign key) – връзка към таблицата book.
* **author\_id** (bigint, foreign key) – връзка към таблицата author.

**Архитектура на проекта**

Проектът следва стандартната архитектура за Spring Boot приложения:

* **Model (Entities)** – класове, които представляват таблиците в базата данни. Всеки клас е анотиран с JPA анотации като @Entity, @ManyToMany, @ManyToOne, @JoinTable, за да се дефинират връзките между тях.
* **Repository** – интерфейси, които наследяват JpaRepository за извършване на CRUD операции с базата данни.
* **Controller** – класове за обработка на HTTP заявки и връщане на резултати, предоставяйки RESTful API.
* **Service** – класове за логика на бизнеса (по избор), които извършват обработка на данни между контролери и репозитории.

**Основни класове**

* **Book** – модел на книга, свързан с автори и издатели.
* **Author** – модел на автор, свързан с книги.
* **Publisher** – модел на издател, свързан с книги.
* **BookController** – контролер, отговорен за управлението на книгите през API.
* **AuthorController** – контролер, отговорен за управлението на авторите през API.
* **PublisherController** – контролер, отговорен за управлението на издателите през API.

**4. Достъп до API**

След като приложението е стартирано, може да използвате следните ендпойнти:

* **GET /books** – връща списък с всички книги.
* **GET /authors** – връща списък с всички автори.
* **GET /publishers** – връща списък с всички издатели.
* **POST /books** – създава нова книга.
* **POST /authors** – създава нов автор.
* **POST /publishers** – създава нов издател.

Можете да използвате Postman или друг HTTP клиент, за да тествате тези ендпойнти.

**Структура на директориите**

css

Копиране на код

MyModelProject/

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ ├── com/

│ │ │ │ ├── example/

│ │ │ │ │ └── mymodel/

│ │ │ │ │ ├── model/

│ │ │ │ │ │ ├── Author.java

│ │ │ │ │ │ ├── Book.java

│ │ │ │ │ │ ├── Publisher.java

│ │ │ │ │ │ └── AuthorBook.java

│ │ │ │ │ ├── controller/

│ │ │ │ │ │ ├── BookController.java

│ │ │ │ │ │ ├── AuthorController.java

│ │ │ │ │ │ └── PublisherController.java

│ │ │ │ │ └── MyModelApplication.java

│ │ └── resources/

│ │ ├── application.properties

├── pom.xml

**Заключение**

Този проект използва Spring Boot и JPA за изграждане на RESTful уеб приложение с релационна база данни. Той позволява създаване, четене, актуализиране и изтриване на книги, автори и издатели и предоставя лесен начин за взаимодействие чрез REST API.