package com.becoder.controller;  
  
import com.becoder.model.Document;  
import com.becoder.repository.DocumentRepository;  
import jakarta.servlet.http.HttpSession;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.multipart.MultipartFile;  
  
import java.io.File;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.nio.file.StandardCopyOption;  
import java.util.List;  
  
@Controller // Обозначает класс как контроллер Spring  
public class DocumentController {  
  
 @Autowired // Автоматическая инъекция DocumentRepository  
 private DocumentRepository documentRepo;  
  
 @GetMapping("/") // GET-запрос к корневому URL-адресу  
 public String index(Model m) {  
 // ... (код для отображения документов, если требуется)  
  
 // Получить список документов из хранилища  
 List<Document> documents = documentRepo.findAll();  
  
// Перебрать список и добавить элементы в модель  
 for (Document doc : documents) {  
 m.addAttribute("document", doc);  
 }  
  
// Возвратить шаблон, который отобразит список документов  
 return "index";  
  
 }  
  
 // POST-запрос к URL-адресу /documentUpload  
 @PostMapping("/documentUpload")  
 public String documentUpload(@RequestParam MultipartFile doc, HttpSession session) {  
  
 Document document = new Document(); // Создание нового объекта Document  
 document.setDocumentName(doc.getOriginalFilename());  
  
 Document savedDocument = documentRepo.save(document); // Сохранение Document в хранилище  
  
 if (savedDocument != null) { // Обработка загрузки файла  
 try {  
 File saveDir = new ClassPathResource("static/documents").getFile(); // Определение каталога сохранения  
 Path path = Paths.*get*(saveDir.getAbsolutePath() + File.*separator* + doc.getOriginalFilename()); // Обработка загрузки файла  
 Files.*copy*(doc.getInputStream(), path, StandardCopyOption.*REPLACE\_EXISTING*); // Копирование загруженного файла  
 } catch (Exception e) {  
 e.printStackTrace();  
 // Обработайте исключение должным образом, например, отобразите сообщение об ошибке  
 }  
 }  
  
 session.setAttribute("msg", "Документ загружен успешно");  
  
 return "redirect:/";  
 }  
}

package com.becoder;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class UploadDocumentsApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(UploadDocumentsApplication.class, args);  
 }  
  
}

spring.datasource.url=jdbc:mysql://localhost:3306/doc\_upload  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.username=root  
spring.datasource.password=Elchi2001#  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect  
spring.jpa.hibernate.ddl-auto=update  
  
spring.servlet.multipart.max-file-size=10MB  
spring.servlet.multipart.max-request-size=10MB

package com.becoder.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
  
@Entity  
public class Document {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private int id;  
 private String documentName;  
  
 // ... getters and setters  
  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getDocumentName() {  
 return documentName;  
 }  
  
 public void setDocumentName(String documentName) {  
 this.documentName = documentName;  
 }  
}

package com.becoder.repository;  
  
import com.becoder.model.Document;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface DocumentRepository extends JpaRepository<Document, Integer> {  
}

<!Doctype html>  
<html lang="ru" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Сайт для загрузки документов</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"  
 rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
</head>  
  
<body>  
<nav class="navbar navbar-expand-lg navbar-dark bg-success">  
 <div class="container-fluid">  
 <a class="navbar-brand" href="#">Сайт для загрузки документов</a>  
 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarSupportedContent">  
  
 </div>  
 </div>  
</nav>  
  
<div class="container">  
  
 <th:block th:if="${session.msg}">  
 <p class="text-center fs-4 text-success">[[${session.msg}]]</p>  
 </th:block>  
  
 <div class="row">  
 <div class="col-md-6 offset-md-3 mt-4">  
  
 <form action="documentUpload" enctype="multipart/form-data" method="post">  
 <div class="mb-3">  
 <label for="doc" class="form-label">Выберите документ:</label>  
 <input type="file" name="doc" id="doc" class="form-control">  
 </div>  
 <div class="text-center">  
 <button class="btn btn-primary">Загрузить</button>  
 </div>  
 </form>  
  
 </div>  
 </div>  
</div>  
  
<hr>  
  
<div class="container">  
 <div class="row">  
 </div>  
</div>  
  
</body>  
</html>

