**import org.springframework.stereotype.Controller;**

**import org.springframework.ui.Model;**

**import org.springframework.web.bind.annotation.\*;**

**import org.springframework.web.multipart.MultipartFile;**

**import org.springframework.web.servlet.mvc.method.annotation.MvcUriComponentsBuilder;**

**import org.springframework.web.servlet.mvc.support.RedirectAttributes;**

**import java.io.IOException;**

**import java.util.List;**

**import java.util.stream.Collectors;**

**@Controller**

**public class FileController {**

**private final StorageService storageService;**

**public FileController(StorageService storageService) {**

**this.storageService = storageService;**

**}**

**@GetMapping("/")**

**public String listUploadedFiles(Model model) throws IOException {**

**List<String> fileNames = storageService.loadAll()**

**.map(path ->**

**MvcUriComponentsBuilder.fromMethodName(FileController.class,**

**"serveFile", path.getFileName().toString()).build().toString())**

**.collect(Collectors.toList());**

**model.addAttribute("files", fileNames);**

**return "uploadForm";**

**}**

**@GetMapping("/files/{filename:.+}")**

**@ResponseBody**

**public ResponseEntity<Resource> serveFile(@PathVariable String filename) {**

**Resource file = storageService.loadAsResource(filename);**

**return ResponseEntity.ok().header(HttpHeaders.CONTENT\_DISPOSITION,**

**"attachment; filename=\"" + file.getFilename() + "\"").body(file);**

**}**

**@PostMapping("/")**

**public String handleFileUpload(@RequestParam("file") MultipartFile file,**

**RedirectAttributes redirectAttributes) {**

**storageService.store(file);**

**redirectAttributes.addFlashAttribute("message",**

**"You successfully uploaded " + file.getOriginalFilename() + "!");**

**return "redirect:/";**

**}**

**@ExceptionHandler(StorageFileNotFoundException.class)**

**public ResponseEntity<?> handleStorageFileNotFound(StorageFileNotFoundException exc) {**

**return ResponseEntity.notFound().build();**

**}**

**}**

**import org.springframework.core.io.Resource;**

**import org.springframework.web.multipart.MultipartFile;**

**import java.nio.file.Path;**

**import java.util.stream.Stream;**

**public interface StorageService {**

**void init();**

**void store(MultipartFile file);**

**Stream<Path> loadAll();**

**Path load(String filename);**

**Resource loadAsResource(String filename);**

**void deleteAll();**

**}**

**import org.springframework.core.io.Resource;**

**import org.springframework.core.io.UrlResource;**

**import org.springframework.stereotype.Service;**

**import org.springframework.web.multipart.MultipartFile;**

**import java.io.IOException;**

**import java.net.MalformedURLException;**

**import java.nio.file.\*;**

**import java.util.stream.Stream;**

**@Service**

**public class FileSystemStorageService implements StorageService {**

**private final Path rootLocation = Paths.get("upload-dir");**

**@Override**

**public void init() {**

**try {**

**Files.createDirectory(rootLocation);**

**} catch (IOException e) {**

**throw new StorageException("Could not initialize storage", e);**

**}**

**}**

**@Override**

**public void store(MultipartFile file) {**

**try {**

**if (file.isEmpty()) {**

**throw new StorageException("Failed to store empty file " + file.getOriginalFilename());**

**}**

**Files.copy(file.getInputStream(), this.rootLocation.resolve(file.getOriginalFilename()),**

**StandardCopyOption.REPLACE\_EXISTING);**

**} catch (IOException e) {**

**throw new StorageException("Failed to store file " + file.getOriginalFilename(), e);**

**}**

**}**

**@Override**

**public Stream<Path> loadAll() {**

**try {**

**return Files.walk(this.rootLocation, 1)**

**.filter(path -> !path.equals(this.rootLocation))**

**.map(this.rootLocation::relativize);**

**} catch (IOException e) {**

**throw new StorageException("Failed to read stored files", e);**

**}**

**}**

**@Override**

**public Path load(String filename) {**

**return rootLocation.resolve(filename);**

**}**

**@Override**

**public Resource loadAsResource(String filename) {**

**try {**

**Path file = load(filename);**

**Resource resource = new UrlResource(file.toUri());**

**if (resource.exists() || resource.isReadable()) {**

**return resource;**

**} else {**

**throw new StorageFileNotFoundException(**

**"Could not read file: " + filename);**

**}**

**} catch (MalformedURLException e) {**

**throw new StorageFileNotFoundException("Could not read file: " + filename, e);**

**}**

**}**

**@Override**

**public void deleteAll() {**

**FileSystemUtils.deleteRecursively(rootLocation.toFile());**

**}**

**}**

**public class StorageFileNotFoundException extends StorageException {**

**public StorageFileNotFoundException(String message) {**

**super(message);**

**}**

**public StorageFileNotFoundException(String message, Throwable cause) {**

**super(message, cause);**

**}**

**}**

**import org.springframework.boot.CommandLineRunner;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**import org.springframework.context.annotation.Bean;**

**@SpringBootApplication**

**public class FileUploadDownloadApplication {**

**public static void main(String[] args) {**

**SpringApplication.run(FileUploadDownloadApplication.class, args);**

**}**

**@Bean**

**CommandLineRunner init(StorageService storageService) {**

**return (args) -> {**

**storageService.deleteAll();**

**storageService.init();**

**};**

**}**

**}**

**<!DOCTYPE html>**

**<html xmlns:th="http://www.thymeleaf.org">**

**<head>**

**<meta charset="UTF-8">**

**<title>File Upload and Download</title>**

**</head>**

**<body>**

**<h**