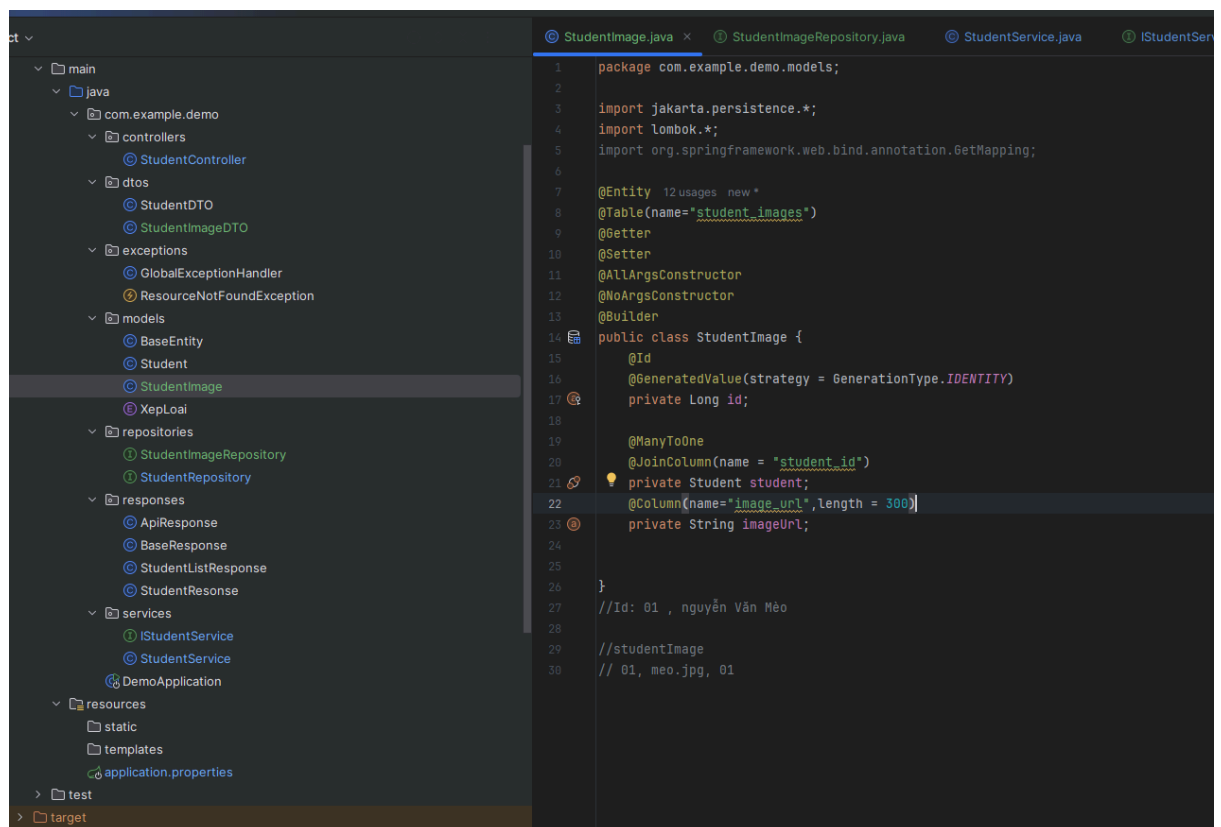


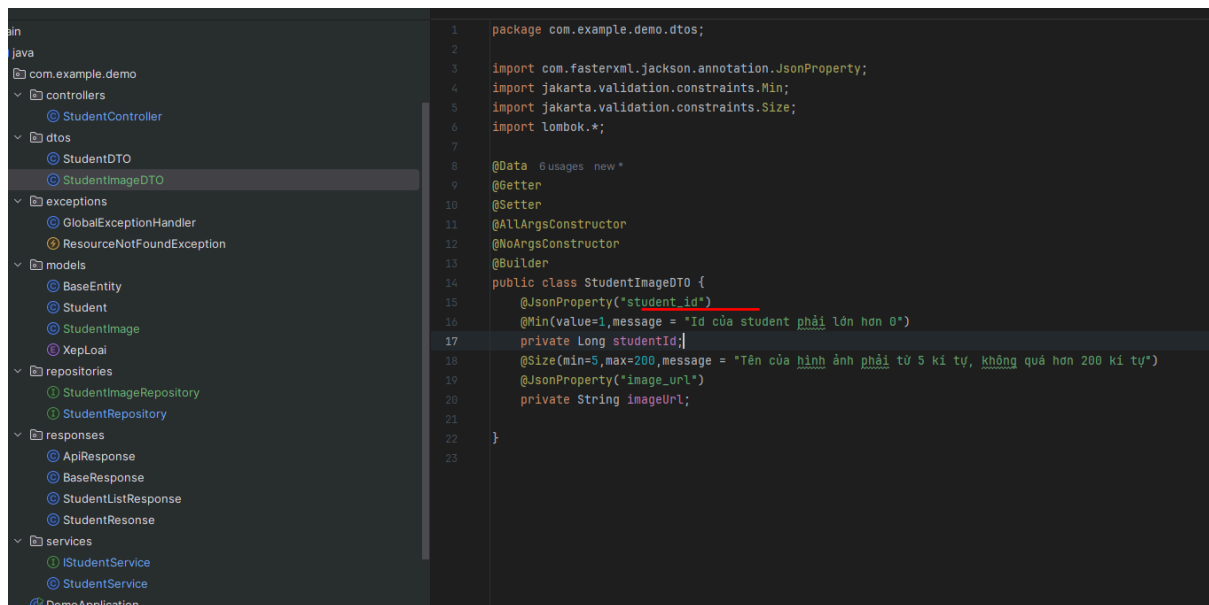
<https://docs.google.com/document/d/14Jrb-yZsZWobN2ySlqn7dUz2cCzAQBsj38d4YM6cg4o/edit?usp=sharing>

=+upload hình ảnh:

Model

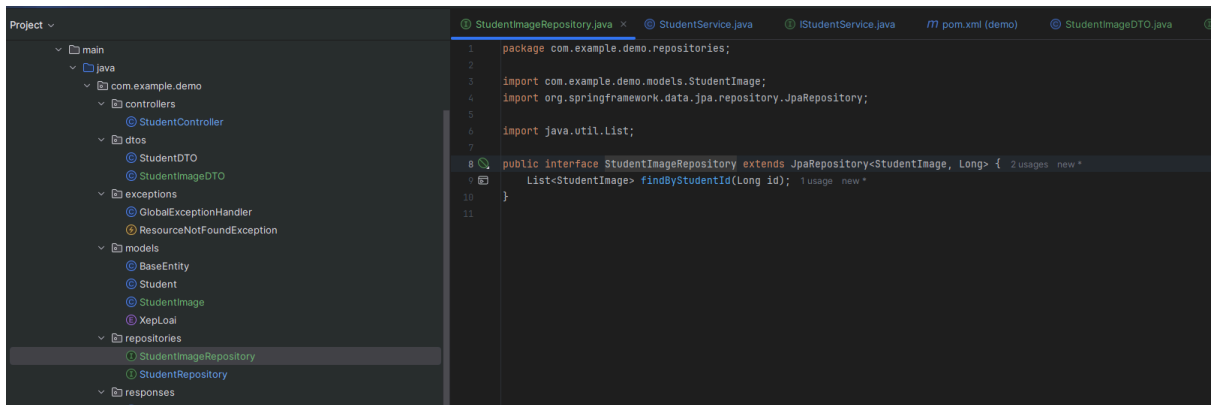


DTO

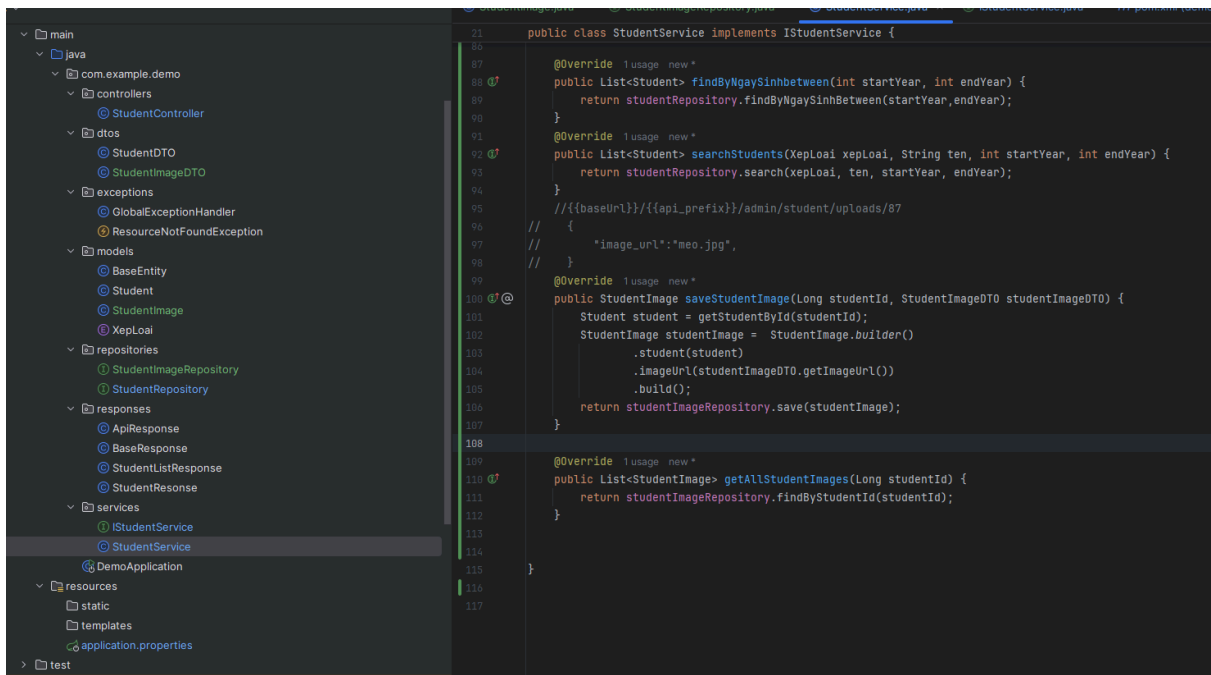


Result Grid				Filter Rows:	Edit:	Export/Import:	Wra
	id	image_url	student_id				
	1	meo.jpg	89				
	2	nai.jpg	89				
	3	nai.jpg	89				
	4	nai.jpg	89				
	5	nai.jpg	89				
	6	nai.jpg	83				
	7	nai.jpg	83				
	8	nai.jpg	83				
	9	96810f53-97ed-4aa5-b7f4-8ac669a8284a_hak...	87				
	10	d4062c3d-1a37-40b7-a308-5560bb1f0b98_hak...	87				
▶	11	a7b78454-c900-40cc-85ea-e767af7be009_nam...	87				
	12	nai.jpg	99				
	13	meo.jpg	99				
✱	NULL	NULL	NULL				

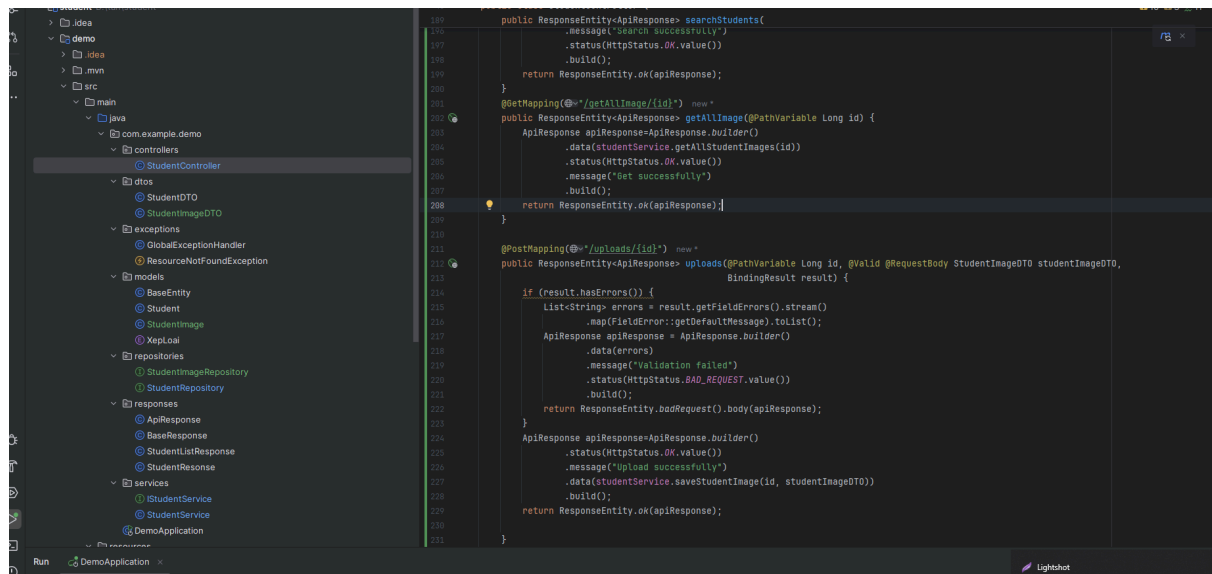
# repository



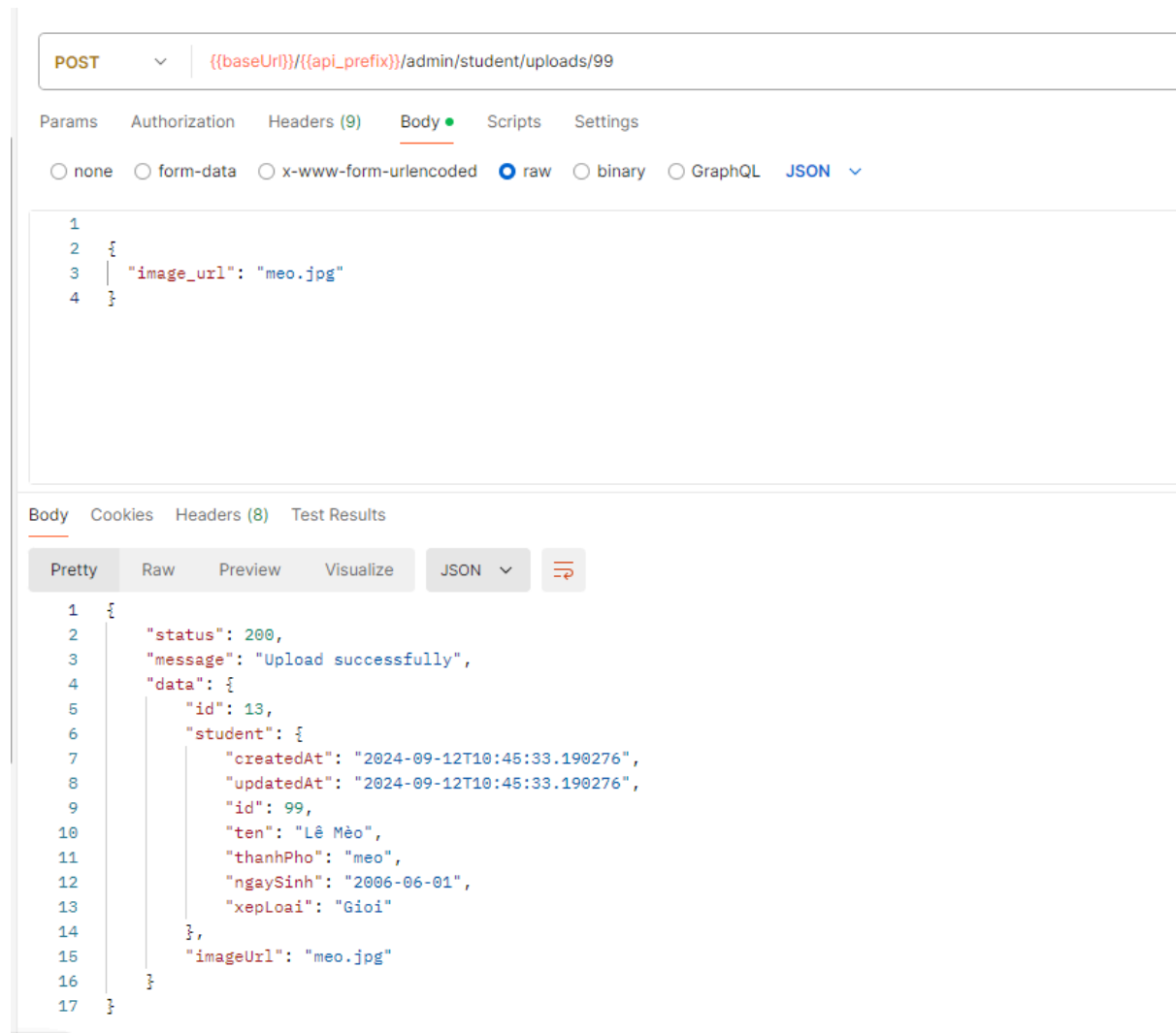
service



controller



Thầy ơi mình đang học gì vậy  
thầy, em đi đám cưới xong rồi =))  
=> Học uploads file nha => Dạ  
- Có khó hiểu không thầy?=> Khó  
chứ



Giải thích

`@JoinColumn(name = "student_id")` nghĩa là `student_id` trong bảng `student_images` là khóa ngoại trỏ đến khóa chính `id` của bảng `students`.

=> Điều này được chỉ định qua và Hibernate tự động thực hiện việc ánh xạ dựa trên cấu hình này.

Up hình

```

209     }
210
211     @PostMapping(value="@*/uploads/{id}") new *
212     public ResponseEntity<ApiResponse> uploads(@PathVariable Long id,@ModelAttribute("files") MultipartFile files) throws IOException {
213         String fileName=storeFile(files);
214         StudentImageDTO studentImageDTO=StudentImageDTO.builder()
215             .imageUrl(fileName)
216             .build();
217         ApiResponse apiResponse=ApiResponse.builder()
218             .status(HttpStatus.OK.value())
219             .message("Upload successfully")
220             .data(studentService.saveStudentImage(id, studentImageDTO))
221             .build();
222         return ResponseEntity.ok(apiResponse);
223     }
224
225     @ private String storeFile(MultipartFile file) throws IOException { 1 usage new *
226         String fileName = StringUtils.cleanPath(file.getOriginalFilename());
227         String uniqueFileName =UUID.randomUUID().toString()+" "+fileName;
228         java.nio.file.Path uploadDir=Paths.get( first: "upload");
229         if(!Files.exists(uploadDir)) {
230             Files.createDirectory(uploadDir);
231         }
232         java.nio.file.Path destination=Paths.get(uploadDir.toString(),uniqueFileName);
233         Files.copy(file.getInputStream(),destination,StandardCopyOption.REPLACE_EXISTING);
234         return uniqueFileName;
235     }
236
237 }
238
239

```

```

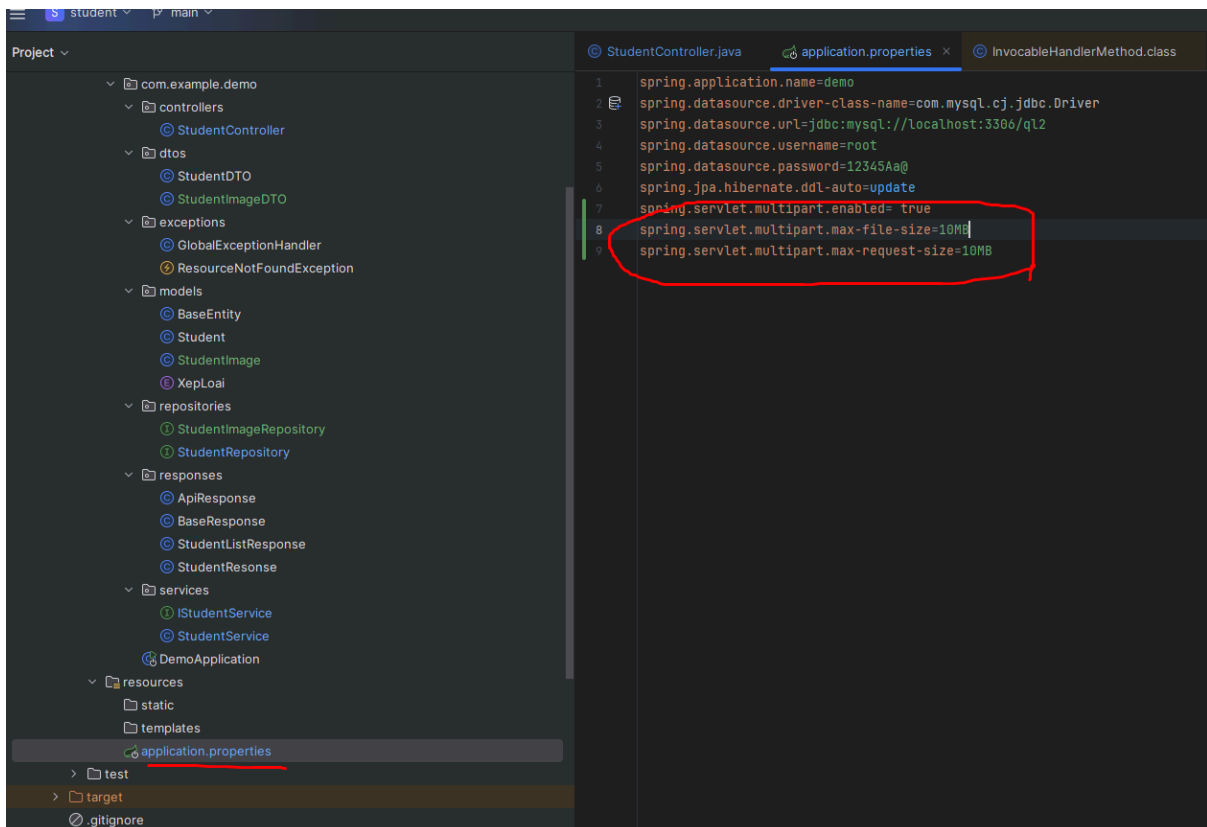
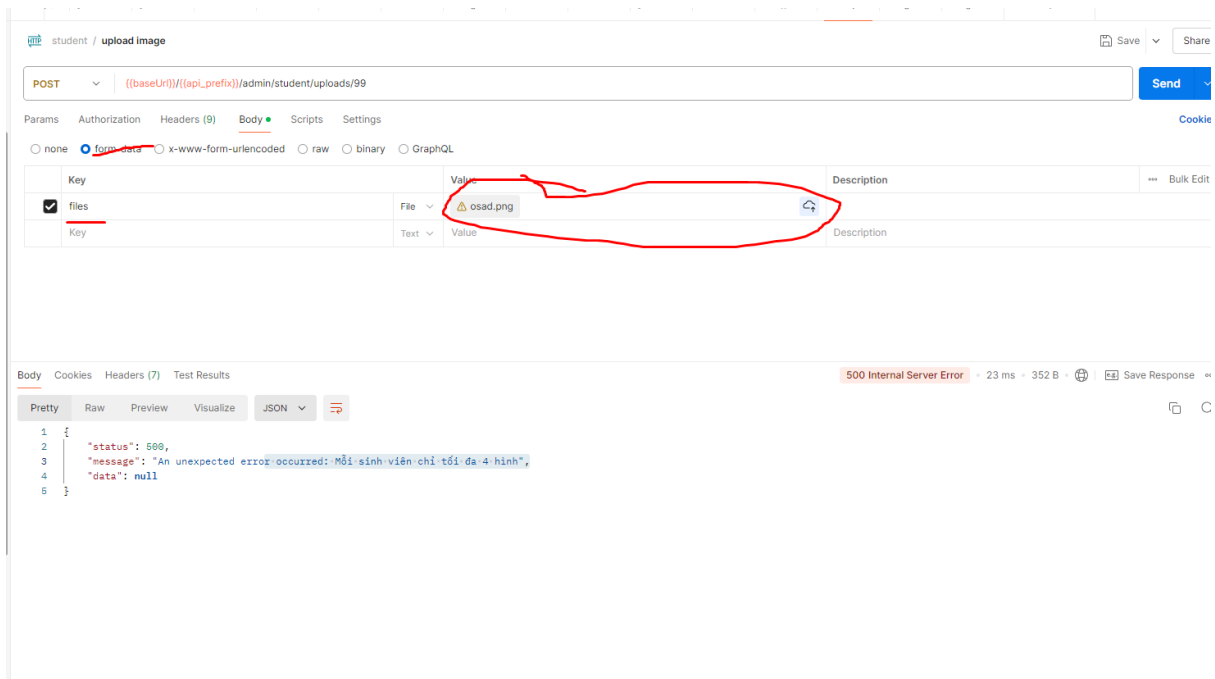
209     }
210
211     @PostMapping(value="@*/uploads/{id}") new *
212     public ResponseEntity<ApiResponse> uploads(@PathVariable Long id,@ModelAttribute("files") MultipartFile files) throws IOException {
213         String fileName=storeFile(files);
214         StudentImageDTO studentImageDTO=StudentImageDTO.builder()
215             .imageUrl(fileName)
216             .build();
217         ApiResponse apiResponse=ApiResponse.builder()
218             .status(HttpStatus.OK.value())
219             .message("Upload successfully")
220             .data(studentService.saveStudentImage(id, studentImageDTO))
221             .build();
222         return ResponseEntity.ok(apiResponse);
223     }
224
225     @ private String storeFile(MultipartFile file) throws IOException { 1 usage new *
226         String fileName = StringUtils.cleanPath(file.getOriginalFilename());
227         String uniqueFileName =UUID.randomUUID().toString()+" "+fileName;
228         java.nio.file.Path uploadDir=Paths.get( first: "upload");
229         if(!Files.exists(uploadDir)) {
230             Files.createDirectory(uploadDir);
231         }
232         java.nio.file.Path destination=Paths.get(uploadDir.toString(),uniqueFileName);
233         Files.copy(file.getInputStream(),destination,StandardCopyOption.REPLACE_EXISTING);
234         return uniqueFileName;
235     }
236
237 }
238
239

```

```

92 @
93 public List<Student> searchStudents(XepLoai xepLoai, String ten, int startYear, int endYear) {
94     return studentRepository.search(xepLoai, ten, startYear, endYear);
95 }
96 // {
97 //     "image_url":"meo.jpg",
98 // }
99
100 @Override 1 usage new *
101 public StudentImage saveStudentImage(Long studentId, StudentImageDTO studentImageDTO) {
102     Student student = getStudentById(studentId);
103     StudentImage studentImage = StudentImage.builder()
104         .student(student)
105         .imageUrl(studentImageDTO.getImageUrl())
106         .build();
107     int size=studentImageRepository.findById(studentId).size();
108     if(size>=4){
109         throw new InvalidParameterException("Mỗi sinh viên chỉ tối đa 4 hình");
110     }
111     return studentImageRepository.save(studentImage);
112 }
113
114 @Override 1 usage new *
115 public List<StudentImage> getAllStudentImages(Long studentId) {
116     return studentImageRepository.findById(studentId);
117 }
118
119
120
121

```



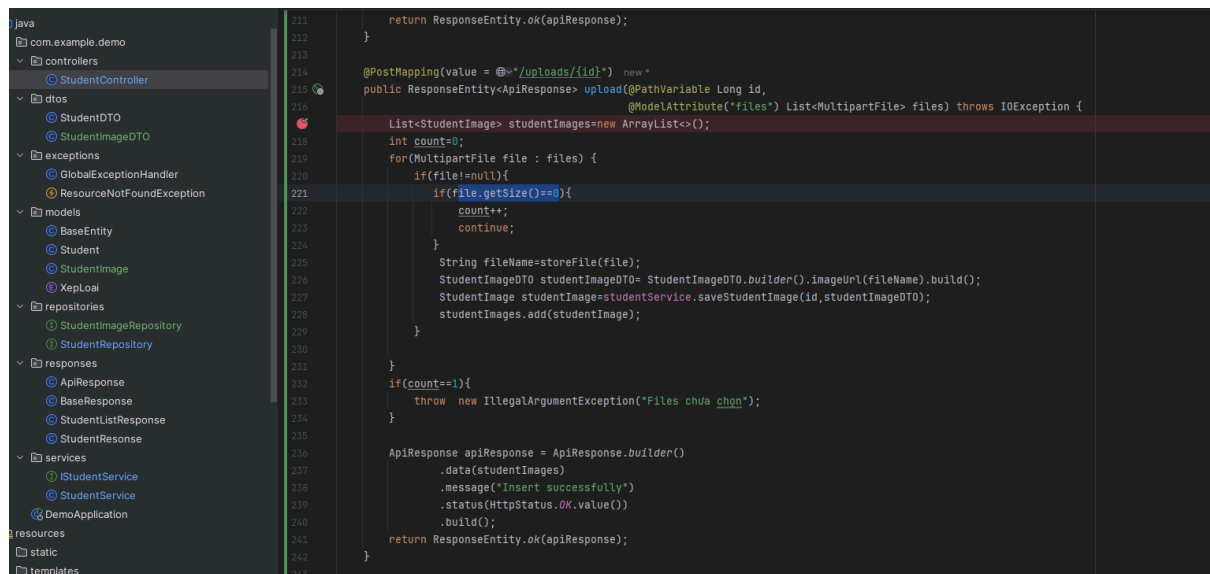
Giải thích:

- `file.getOriginalFilename()` lấy tên gốc của tệp tin tải lên.

- `StringUtils.cleanPath()` là một phương thức của thư viện `Spring` dùng để loại bỏ các ký tự không hợp lệ từ tên file và đảm bảo đường dẫn tệp tin là hợp lệ.
- Kiểm tra xem thư mục `upload` đã tồn tại chưa bằng cách sử dụng `Files.exists()`. Nếu thư mục chưa tồn tại, thì `Files.createDirectory()` được gọi để tạo thư mục.
- **Destination:** `java.nio.file.Path`  
`destination=Paths.get(uploadDdir.toString(),uniqueFileName);`  
=> đường dẫn đích của tệp tin, kết hợp thư mục `upload` và tên tệp tin duy nhất đã được tạo.
- `Files.copy(file.getInputStream(), destination, StandardCopyOption.REPLACE_EXISTING);`  
=>
  - + sao chép nội dung của tệp tin từ `InputStream` của `MultipartFile` vào đường dẫn đích `destination`. +
  - + `StandardCopyOption.REPLACE_EXISTING` có nghĩa là nếu một tệp tin với cùng tên đã tồn tại tại vị trí đích, nó sẽ bị thay thế.

```
import ch.qos.logback.core.util.StringUtil;
```





Chủ đề đồ án

Một trang blog với các tính năng tạo bài viết, bình luận, và quản lý người dùng.

Nền tảng học trực tuyến với các chức năng quản lý khoá học, bài giảng, và tương tác giữa giáo viên và học viên.

Quản lý kho bãi với tính năng theo dõi tồn kho, nhập kho, xuất kho và báo cáo.

Hệ thống giúp quản lý các dự án với các chức năng theo dõi tiến độ, phân chia nhiệm vụ và quản lý thành viên.

Hệ thống quản lý sách, theo dõi mượn trả và quản lý thành viên của một thư viện.

Một trang web cho thuê đồ cắm trại.

Xây dựng ứng dụng bán vé xem phim trực tuyến với chức năng chọn ghế, đặt vé, thanh toán và quản lý suất chiếu.

Xây dựng hệ thống quản lý dịch vụ khách sạn, bao gồm quản lý phòng, đặt phòng, và theo dõi dịch vụ bổ sung.

Phát triển hệ thống đấu giá cho phép người dùng đăng sản phẩm, tham gia đấu giá và quản lý lịch sử giao dịch.

Phát triển hệ thống quản lý điểm số cho học sinh, bao gồm tính năng nhập điểm, xem kết quả và phân tích thành tích.

Web bán hàng trực tuyến với chức năng giỏ hàng, thanh toán, quản lý sản phẩm và đơn hàng.

# Đồ án cuối khóa

## Đề Tài : Quản lý nhà hàng

### I. Giới Thiệu:

Tạo ra 1 nền tảng quản lý nhà hàng

### II. Tính Năng:

#### 1. Quản lý đặt bàn

- Cho phép khách hàng đặt bàn trước.
- Quản lý thời gian đặt bàn, số lượng khách, và trạng thái đặt chỗ.
- Hiển thị tình trạng bàn trống và đặt trước.

#### 2. Quản lý thực đơn

- Quản lý thông tin món ăn, bao gồm tên, giá cả, mô tả, và hình ảnh.
- Tạo và quản lý các danh mục món ăn
- Quản lý các chương trình khuyến mãi.

#### 3. Quản lý đơn hàng

- Quản lý đơn hàng từ khi khách gọi món đến khi hoàn tất thanh toán.
- Theo dõi trạng thái đơn hàng

#### • 4. Quản lý kho

- Theo dõi và quản lý nguyên liệu, bao gồm số lượng tồn kho, nhập hàng và xuất hàng.

#### 5. Quản lý nhân viên

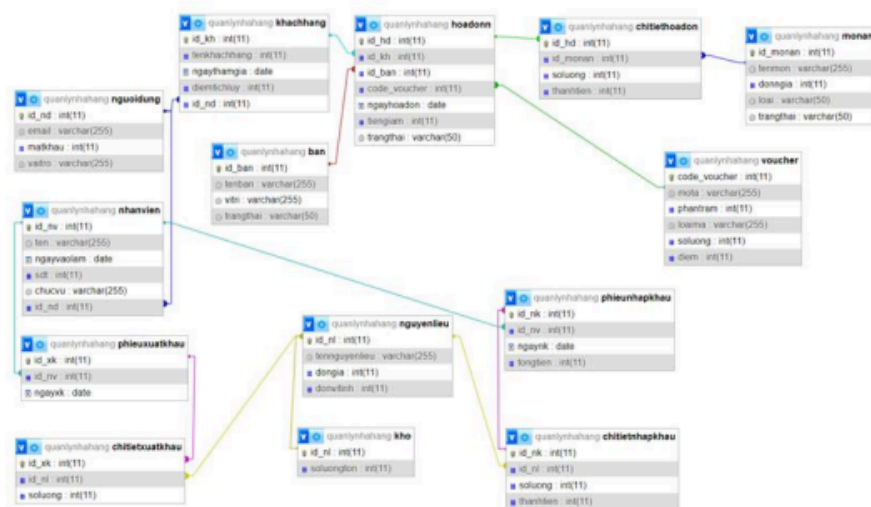
- Quản lý thông tin nhân viên, bao gồm ca làm việc, lương, và hiệu suất công việc.
- Phân công nhiệm vụ cho nhân viên theo ca và theo vị trí công việc.

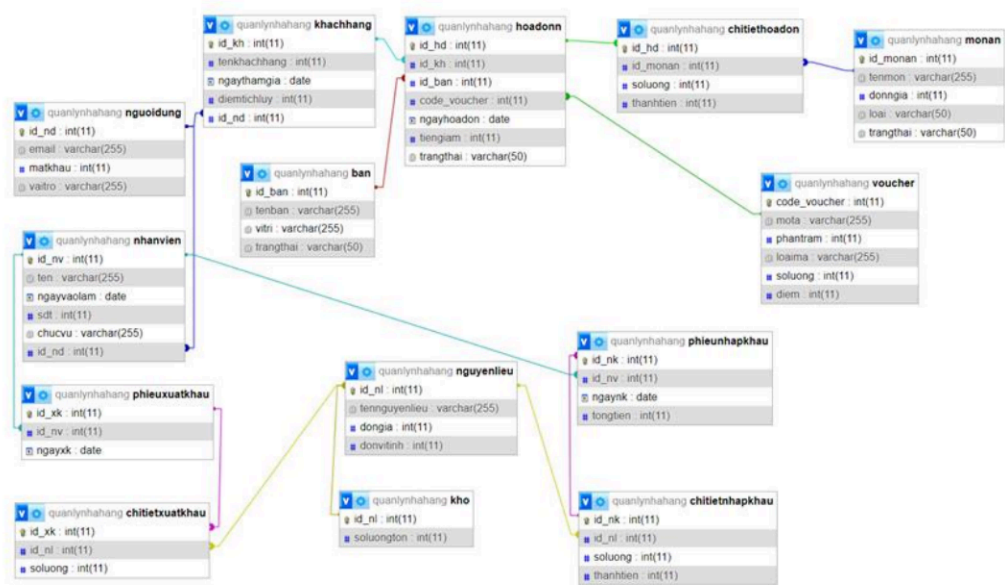
#### 6. Quản lý khách hàng

- Lưu trữ thông tin khách hàng, bao gồm lịch sử đặt chỗ, lịch sử đặt món, và phản hồi.
- Chương trình khách hàng thân thiết với các ưu đãi và phần thưởng dựa trên số lần ghé thăm.

Lý do chọn đề tài: Lúc trước em có làm thêm tại 1 nhà hàng và em biết cách vận hành 1 nhà hàng và em chọn đề tài này để rèn luyện và nâng cao kỹ năng về SQL, bao gồm việc thiết kế cơ sở dữ liệu, truy vấn dữ liệu, và tối ưu hóa hiệu suất

### III. Bảng thiết kế cơ sở dữ liệu





## ĐỒ ÁN TỐT NGHIỆP - HỒ TIẾN - CF-FIT-SB001

### I. GIỚI THIỆU ĐỒ ÁN

+ Tạo 1 nền tảng quản lý kho nguyên phụ liệu, bán thành phẩm và thành phẩm ngành hàng quần áo trẻ em.

+ Đối tượng: Cá nhân, xưởng sản xuất, startup local brand,...

### II. LÝ DO THỰC HIỆN ĐỒ ÁN

- Cá Nhân: Hiện đang xây dựng 1 Local Brand chuyên đồ ngủ cho trẻ sơ sinh, cần các công cụ có thể quản lý tồn kho từ xưởng, dự trữ nguyên liệu và có thể phác thảo chi phí sơ bộ cho 1 sản phẩm => tối ưu chi phí sản xuất.

- Cộng Đồng:

+ Hỗ trợ startup ngành thời trang hạn chế các thất thoát trong quá trình sản xuất mới, phác thảo sơ bộ chi phí sản xuất sản phẩm ( tối ưu sản xuất).

+ Phát triển thành 1 nơi kết nối các đơn vị cung cấp nguyên phụ liệu , xưởng gia công, chủ shop thời trang hoặc các startup ngành hàng thời trang.

+ Dẫn dắt chia sẻ kinh nghiệm.

### III. TÍNH NĂNG

- **Quản Lý Kho**

- + Thêm, sửa, xóa, phân loại các nguyên phụ liệu đầu vào.

- + Quản lý tồn kho theo dạng nguyên liệu, thành phẩm, bán thành phẩm,...

- + Truy xuất, kiểm soát nguyên liệu theo số lô đầu vào.

- **Tính Toán Chi Phí Sản Xuất**

- + Phân loại, tổng hợp các chi phí sản xuất để ra 1 sản phẩm (Nguyên phụ liệu, gia công, bao bì,...) => Tối ưu chi phí theo từng hạng mục.

- + Cho phép người dùng lên phác thảo sơ bộ, tính toán chi phí cần thiết cho 1 sản phẩm.

- **Quản Lý Sản Phẩm**

- + Cho phép truy xuất, kiểm soát lỗi sản phẩm theo từng công đoạn, từng nhà cung ứng.

- + Theo dõi tiến trình sản xuất sản phẩm.

- + Đánh giá năng lực nhà cung ứng.

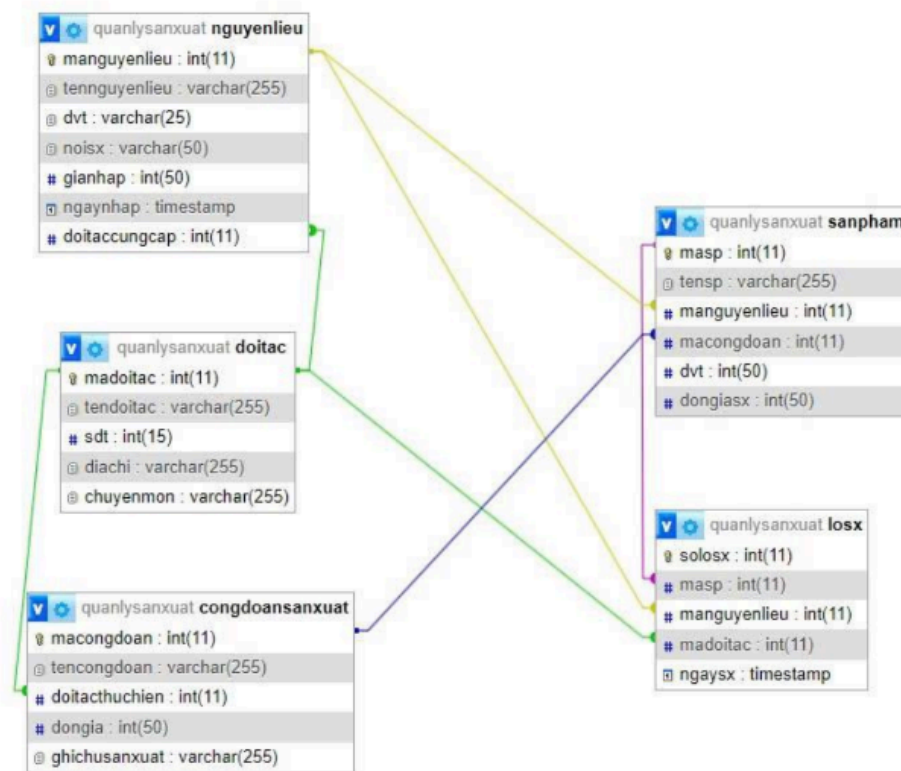
- + Đánh giá sản phẩm, rút ra những khoảng cần tối ưu.

### IV. ĐỊNH HƯỚNG PHÁT TRIỂN CỦA ĐỒ ÁN

+ Mở rộng sang lĩnh vực bán hàng để tạo thành 1 vòng khép kín từ khâu sản xuất, bán hàng, đánh giá sản phẩm => nhận định xu hướng thị trường qua các số liệu phân tích sau bán hàng.

+ Tinh chỉnh để có thể đáp ứng thêm cho việc sản xuất các ngành hàng khác ( không chỉ riêng thời trang).

+ Trở thành trang Marketing lớn nhất ngành thời trang, nơi các Startup có thể tìm nguồn nguyên liệu, xưởng có thể tìm đơn hàng gia công, và nhà cung cấp nguyên liệu có thể tìm đầu ra để phát triển ngành may mặc Việt Nam.



## Đồ án cuối khóa (Đề tài PHP)

### I. Giới thiệu

Tạo một nền tảng học lập trình tương tự như Coursera, Udemy, ...

### II. Tính năng chính

#### • Quản lý người dùng

- + Thêm, sửa, xóa người dùng
- + Đăng nhập, đăng xuất, tạo tài khoản
- + Quản lý thông tin các nhân học viên, cho phép học viên có thể cập nhật thông tin các nhân, thay đổi mật khẩu, và quản lý tài khoản.
- + Hiển thị thông tin về tiến độ học tập của học viên, các bài đã hoàn thành, tiến độ trong khoá học.

#### • Quản lý khóa học

- + Tạo xóa và cập nhật các khóa học.
- + Cho phép người quản trị khóa học đó thêm, xóa, sửa các bài giảng và tài liệu

#### • Phản hồi

- + Cho phép học viên bình luận đánh giá về khóa học, bài giảng
- + Phần comment dưới mọi bài giảng cho phép học viên có thể đặt câu hỏi với giáo viên trực tiếp (tương tự như phần comment trên facebook)

#### • Thanh toán

- + Tích hợp hệ thống thanh toán để thu học phí khóa học (Momo, chuyển khoản ngân hàng,...)

### III. Bảng thiết kế cơ sở dữ liệu

