**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦



**ĐỀ TÀI: XÂY DỰNG WEB SERVICE BẰNG SPRING BOOT CHO ECOMMERCE WEBSITE**

|  |  |  |  |
| --- | --- | --- | --- |
| Sinh viên thực hiện: | | | |
| STT | Họ tên | MSSV | Ngành |
| 1 | Trần Thành Thoại | 23540035 | Công nghệ thông tin |

**TP. HỒ CHÍ MINH – 04/2025**

# GIỚI THIỆU

Đề tài “Xây dựng web service bằng Spring Boot cho ecommerce website” nhằm phát triển một hệ thống backend API phục vụ cho website thương mại điện tử. Hệ thống cung cấp các chức năng cốt lõi bao gồm quản lý người dùng, xác thực phân quyền, quản lý đơn hàng và địa chỉ giao hàng.

* Công nghệ sử dụng:
* Ngôn ngữ lập trình: Java 21
* Framework: Spring Boot 3.4.5
* Cơ sở dữ liệu: MySQL
* Bảo mật: Spring Security với JWT (JSON Web Token)
* ORM (Object-Relational Mapping): JPA (Java Persistence API)/Hibernate
* Mapping: MapStruct
* Build tool: Maven

Kết quả đạt được, hệ thống đã triển khai đầy đủ chức năng: đăng ký/đăng nhập, sinh và xác thực JWT, phân quyền theo vai trò (RBAC), quản lý người dùng, đơn hàng, và địa chỉ. Mật khẩu người dùng được mã hóa bằng thuật toán BCrypt để tăng tính bảo mật. Token JWT có thời hạn 4 giờ, chứa thông tin vai trò và được kiểm tra hợp lệ qua endpoint introspection. API thiết kế chuẩn RESTful, phản hồi dạng JSON rõ ràng: code, message, result. Cơ sở dữ liệu được thiết kế chuẩn hóa 3NF, hỗ trợ mở rộng và đảm bảo tính toàn vẹn dữ liệu.

Cam kết: Đề tài do em tự phân tích thiết kế và không dựa trên đề tài nào khác. Em có tham khảo sách về thiết kế cơ sở dữ liệu (Em đề cập trong tài liệu tham khảo).

# MÔ TẢ CƠ SỞ DỮ LIỆU

Hình 1. Mô hình ERD của cơ sở dữ liệu ecommerce\_service

Cơ sở dữ liệu được thiết kế theo mô hình quan hệ nhằm lưu trữ và quản lý dữ liệu hiệu quả cho hệ thống thương mại điện tử. Thiết kế tuân thủ nguyên tắc ACID và chuẩn hóa theo Third Normal Form (3NF) để đảm bảo tính toàn vẹn và giảm thiểu dự thừa dữ liệu.

Dữ liệu được tổ chức thành các bảng. Mỗi bảng chứa các hàng và cột đại diện cho mỗi **thực thể** (Entity). Mỗi hàng đại diện cho một đối tượng cụ thể, chứa một tập các thuộc tính duy nhất. Mỗi cột là các **thuộc tính** (Attribute) chứa các điểm dữ liệu cụ thể để mô tả mỗi thực thể. **Mối quan hệ** (relationship) giữa các thực thể được thiết lập thông qua các khóa.

Ví dụ, Bảng nguoi\_dung đại diện cho thực thể người dùng, mỗi hàng trong bảng đại diện cho một người dùng, mỗi cột chứa các thuộc tính là các thông tin của người dùng như là mã người dùng, tên đăng nhập, mật khẩu, email, họ tên, số điện thoại, thời gian tạo thông tin. Mã người dùng là **khóa chính** để đảm bảo mỗi người dùng trong cơ sở dữ liệu là khác biệt, tuân thủ toàn vẹn dữ liệu và truy xuất được người dùng.

Cơ sở dữ liệu theo quy ước đặt tên **snake\_case** để đặt tên bảng, tên thuộc tính. Cơ sở dữ liệu thiết kế để phục vụ cho hệ thống thương mại điện tử, hỗ trợ quản lý người dùng, vai trò, quyền hạn, sản phẩm, đơn hàng, địa chỉ, đánh giá, thanh toán, v.v.

Phân quyền người dùng thiết kế theo mô hình **RBAC** (Role-Base Access Control). Trong đó quyền truy cập vào các chức năng, tài nguyên của hệ thống được gán cho các vai trò (Role). Người dùng (User) sẽ được gán một hoặc nhiều vai trò và thông qua vai trò đó thì sẽ có các quyền hạn (Permission) tương ứng.

## Thực thể người dùng

* Chức năng: Lưu trữ thông tin tài khoản của tất cả người dùng hệ thống**.**
* Các thuộc tính:
* ma\_nguoi\_dung (Khóa chính): Mã định danh duy nhất cho mỗi người dùng (UUID).
* ten\_dang\_nhap: Tên đăng nhập, duy nhất, dùng để đăng nhập hệ thống.
* mat\_khau: Mật khẩu đã được mã hóa.
* email: Địa chỉ email, duy nhất.
* ho\_ten: Họ tên đầy đủ của người dùng.
* so\_dien\_thoai: Số điện thoại liên hệ.
* thoi\_gian\_tao: Thời điểm tạo tài khoản.
* Mối quan hệ:
* 1-n với bảng dia\_chi: Một người dùng có thể có nhiều địa chỉ.
* 1-n với bảng don\_hang: Một người dùng có thể đặt nhiều đơn hàng.
* n-n với bảng vai\_tro qua bảng nguoi\_dung\_vai\_tro: Một người dùng có thể có nhiều vai trò.
* 1-n với bảng danh\_gia: Một người dùng có thể đánh giá nhiều sản phẩm.

Bảng 1. nguoi\_dung

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_nguoi\_dung | VARCHAR(36) | PRIMARY KEY |
|  | ten\_dang\_nhap | VARCHAR(50) | NOT NULL, UNIQUE |
|  | mat\_khau | VARCHAR(255) | NOT NULL |
|  | email | VARCHAR(255) | UNIQUE |
|  | ho\_ten | VARCHAR(255) |  |
|  | so\_dien\_thoai | VARCHAR(20) |  |
|  | thoi\_gian\_tao | DATETIME | DEFAULT CURRENT\_TIMESTAMP |

## Thực thể vai trò

* Chức năng: Lưu trữ thông tin vai trò của người dùng, bao gồm: ADMIN, SELLER, CUSTOMER.
* Các thuộc tính:
* ten\_vai\_tro (Khóa chính): Tên vai trò duy nhất.
* mo\_ta: Mô tả vai trò.
* Mối quan hệ:
* n-n với bảng nguoi\_dung qua bảng nguoi\_dung\_vai\_tro: Một vai trò có thể gán cho nhiều người dùng.
* n-n với bảng quyen\_han qua bảng vai\_tro\_quyen\_han: Một vai trò có thể có nhiều quyền hạn.

Bảng 2. vai\_tro

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ten\_vai\_tro | VARCHAR(50) | PRIMARY KEY |
|  | mo\_ta | TEXT |  |

## Thực thể quyền hạn

* Chức năng: Lưu thông tin quyền hạn của hệ thống**.**
* Các thuộc tính:
* ten\_quyen\_han (Khóa chính): Tên quyền hạn duy nhất.
* mo\_ta: Mô tả quyền hạn, có khả năng thao tác các tác vụ nào đó.
* Mối quan hệ:
* n-n với bảng vai\_tro qua bảng vai\_tro\_quyen\_han: Một quyền hạn có thể thuộc nhiều vai trò.

Bảng 3. quyen\_han

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ten\_quyen\_han | VARCHAR(50) | PRIMARY KEY |
|  | mo\_ta | TEXT |  |

## Bảng phụ giữa người dùng với vai trò và giữa vai trò với quyền hạn

* Bảng nguoi\_dung\_vai\_tro có chức năng tạo liên kết nhiều nhiều (n-n) giữa người dùng và vai trò. Khóa chính là (ma\_nguoi\_dung, ten\_vai\_tro).

Bảng 4. nguoi\_dung\_vai\_tro

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_nguoi\_dung | VARCHAR(36) | NOT NULL, FOREIGN KEY |
|  | ten\_vai\_tro | VARCHAR(50) | NOT NULL, FOREIGN KEY |

* Bảng vai\_tro\_quyen\_han có chức năng tạo liên kết nhiều nhiều (n-n) giữa vai trò và quyền hạn. Khóa chính là (ten\_vai\_tro, ten\_quyen\_han).

Bảng 5. vai\_tro\_quyen\_han

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ten\_vai\_tro | VARCHAR(50) | NOT NULL, FOREIGN KEY |
|  | ten\_quyen\_han | VARCHAR(50) | NOT NULL, FOREIGN KEY |

## Thực thể Token đã hủy

* Chức năng: Lưu trữ các JWT token đã bị vô hiệu hóa khi logout**.** Khi đăng xuất thì token của người dùng vẫn còn hiệu lực cho đến khi hết hạn, khi đăng nhập nếu token có nằm trong bảng dữ liệu này thì sẽ từ chối đăng nhập.
* Các thuộc tính:
* ma\_token (Khóa chính): Chứa mã token đã hủy.
* thoi\_gian\_het\_han: Thời gian hết hạn của Token.
* Mối quan hệ: Không có liên kết với các bảng khác

Bảng 6. quyen\_han

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ten\_quyen\_han | VARCHAR(50) | PRIMARY KEY |
|  | mo\_ta | TEXT |  |

## Thực thể địa chỉ

* Chức năng: Lưu trữ các địa chỉ giao hàng của người dùng.
* Các thuộc tính:
* ma\_dia\_chi (Khóa chính): Mã định danh địa chỉ.
* ma\_nguoi\_dung (Khóa ngoại): Tham chiếu đến người dùng sở hữu địa chỉ này.
* so\_nha: Số nhà, tên đường.
* phuong\_xa: Phường/xã.
* quan\_huyen: Quận/huyện.
* tinh\_thanh: Tỉnh/thành phố.
* Mối quan hệ:
* n-1 với bảng nguoi\_dung: Mỗi địa chỉ thuộc về một người dùng.
* 1-n với bảng don\_hang: Một địa chỉ có thể được dùng cho nhiều đơn hàng.

Bảng 7. dia\_chi

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_dia\_chi | VARCHAR(36) | PRIMARY KEY |
|  | ma\_nguoi\_dung | VARCHAR(36) | FOREIGN KEY |
|  | so\_nha | VARCHAR(50) | NOT NULL |
|  | phuong\_xa | VARCHAR(255) | NOT NULL |
|  | quan\_huyen | VARCHAR(255) | NOT NULL |
|  | tinh\_thanh | VARCHAR(255) | NOT NULL |

## Thực thể sản phẩm

* **Chức năng:** Lưu trữ thông tin các sản phẩm được bán trên website.
* Các thuộc tính:
* ma\_san\_pham (Khóa chính): Mã định danh sản phẩm.
* ten\_san\_pham: Tên sản phẩm.
* mo\_ta\_san\_pham: Mô tả chi tiết sản phẩm.
* gia\_san\_pham: Giá bán.
* so\_luong\_ton: Số lượng tồn kho.
* ma\_loai\_hang (Khóa ngoại): Tham chiếu đến loại hàng.
* ngay\_tao: Thời điểm thêm sản phẩm vào hệ thống.
* Mối quan hệ:
* n-1 với loai\_hang: Mỗi sản phẩm thuộc một loại hàng.
* 1-n với hinh\_anh\_san\_pham: Một sản phẩm có thể có nhiều hình ảnh.
* 1-n với chi\_tiet\_don\_hang: Một sản phẩm có thể xuất hiện trong nhiều chi tiết đơn hàng.
* 1-n với danh\_gia: Một sản phẩm có thể có nhiều đánh giá.

Bảng 8. san\_pham

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_san\_pham | VARCHAR(36) | PRIMARY KEY |
|  | ten\_san\_pham | VARCHAR(255) | NOT NULL |
|  | mo\_ta\_san\_pham | TEXT | NOT NULL |
|  | gia\_san\_pham | DECIMAL(10, 2) | NOT NULL |
|  | so\_luong\_ton | INT | NOT NULL |
|  | ma\_loai\_hang | VARCHAR(36) | FOREIGN KEY |
|  | ngay\_tao | DATETIME | DEFAULT CURRENT\_TIMESTAMP |

## Thực thể hình ảnh sản phẩm

* Chức năng: Lưu trữ các hình ảnh liên quan đến sản phẩm.
* Các thuộc tính:
* ma\_hinh\_anh (Khóa chính): Mã định danh hình ảnh.
* ma\_san\_pham (Khóa ngoại): Tham chiếu đến sản phẩm.
* Mối quan hệ:
* n-1 với san\_pham: Mỗi hình ảnh thuộc về một sản phẩm.

Bảng 9. hinh\_anh\_san\_pham

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_hinh\_anh | VARCHAR(36) | PRIMARY KEY |
|  | ma\_san\_pham | VARCHAR(36) | FOREIGN KEY |

## Thực thể loại hàng

* Chức năng: Lưu trữ các loại hàng hóa, phân loại sản phẩm.
* Các thuộc tính:
* ma\_loai\_hang (Khóa chính): Mã định danh loại hàng.
* ten\_loai\_hang: Tên loại hàng.
* Mối quan hệ:
* 1-n với san\_pham: Một loại hàng có thể có nhiều sản phẩm

Bảng 10. loai\_hang

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_loai\_hang | VARCHAR(36) | PRIMARY KEY |
|  | ten\_loai\_hang | VARCHAR(255) | NOT NULL |

## Thực thể đơn hàng

* Các thuộc tính:
* ma\_don\_hang (Khóa chính): Mã định danh đơn hàng.
* ma\_nguoi\_dung (Khóa ngoại): Người đặt hàng.
* ma\_dia\_chi (Khóa ngoại): Địa chỉ giao hàng.
* tong\_tien: Tổng số tiền của đơn hàng. Tính thông qua bảng chi tiết đơn hàng.
* trang\_thai: Trạng thái đơn hàng ('Chờ xử lý', 'Đang xử lý', 'Đã gửi', 'Đã giao', 'Đã hủy').
* thoi\_gian\_tao: Thời điểm tạo đơn hàng.
* Mối quan hệ:
* n-1 với nguoi\_dung: Mỗi đơn hàng thuộc về một người dùng.
* n-1 với dia\_chi: Mỗi đơn hàng sử dụng một địa chỉ giao hàng.
* 1-n với chi\_tiet\_don\_hang: Một đơn hàng có thể có nhiều chi tiết đơn hàng.
* 1-1 với thanh\_toan: Mỗi đơn hàng có một thông tin thanh toán.

Bảng 11. don\_hang

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_don\_hang | VARCHAR(36) | PRIMARY KEY |
|  | ma\_nguoi\_dung | VARCHAR(36) | FOREIGN KEY |
|  | ma\_dia\_chi | VARCHAR(36) | FOREIGN KEY |
|  | tong\_tien | DECIMAL(10,2) | NOT NULL |
|  | trang\_thai | ENUM('PENDING', 'PROCESSING', 'SHIPPED', 'DELIVERED', 'CANCELLED') | DEFAULT 'PENDING' |
|  | thoi\_gian\_tao | DATETIME | DEFAULT CURRENT\_TIMESTAMP |

## Thực thể chi tiết đơn hàng

* Chức năng: Lưu trữ chi tiết từng sản phẩm trong mỗi đơn hàng.
* Các thuộc tính:
* ma\_ct\_don\_hang (Khóa chính): Mã định danh chi tiết đơn hàng.
* ma\_don\_hang (Khóa ngoại): Tham chiếu đến đơn hàng.
* ma\_san\_pham (Khóa ngoại): Tham chiếu đến sản phẩm.
* so\_luong: Số lượng sản phẩm.
* gia: Giá tại thời điểm đặt hàng.
* Mối quan hệ:
* n-1 với don\_hang: Mỗi chi tiết đơn hàng thuộc về một đơn hàng.
* n-1 với san\_pham: Mỗi chi tiết đơn hàng liên kết với một sản phẩm.

Bảng 12. chi\_tiet\_don\_hang

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_ct\_don\_hang | VARCHAR(36) | PRIMARY KEY |
|  | ma\_don\_hang | VARCHAR(36) | FOREIGN KEY |
|  | ma\_san\_pham | VARCHAR(36) | FOREIGN KEY |
|  | so\_luong | INT | NOT NULL |
|  | gia | DECIMAL(10,2) | NOT NULL |

## Thực thể thanh toán

* Chức năng: Lưu trữ thông tin thanh toán cho các đơn hàng.
* Các thuộc tính:
* ma\_thanh\_toan (Khóa chính): Mã định danh thanh toán.
* ma\_don\_hang (Khóa ngoại): Tham chiếu đến đơn hàng.
* phuong\_thuc: Phương thức thanh toán (chuyển khoản, thẻ tín dụng, v.v.).
* trang\_thai: Trạng thái thanh toán: Đang xử lý, hoàn tất, thất bại (*Pending, completed, failed*)
* ngay\_thanh\_toan: Thời điểm thanh toán.
* Mối quan hệ:
* 1-1 với don\_hang: Mỗi thanh toán gắn với một đơn hàng.

Bảng 13. thanh\_toan

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_thanh\_toan | VARCHAR(36) | PRIMARY KEY |
|  | ma\_don\_hang | VARCHAR(36) | FOREIGN KEY |
|  | phuong\_thuc | VARCHAR(50) | NOT NULL |
|  | trang\_thai | ENUM('PENDING', 'COMPLETED', 'FAILED') | DEFAULT 'PENDING' |
|  | ngay\_thanh\_toan | DATETIME |  |

## Thực thể đánh giá

* Chức năng: Lưu trữ các đánh giá, nhận xét của người dùng về sản phẩm.
* Các thuộc tính:
* ma\_danh\_gia (Khóa chính): Mã định danh đánh giá.
* ma\_nguoi\_dung (Khóa ngoại): Người đánh giá.
* ma\_san\_pham (Khóa ngoại): Sản phẩm được đánh giá.
* diem\_danh\_gia: Điểm đánh giá (1-5).
* binh\_luan: Nhận xét, bình luận.
* ngay\_tao: Thời điểm đánh giá.
* Mối quan hệ:
* n-1 với nguoi\_dung: Mỗi đánh giá do một người dùng thực hiện.
* n-1 với san\_pham: Mỗi đánh giá dành cho một sản phẩm.

Bảng 14. danh\_gia

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** |
|  | ma\_danh\_gia | VARCHAR(36) | PRIMARY KEY |
|  | ma\_nguoi\_dung | VARCHAR(36) | FOREIGN KEY |
|  | ma\_san\_pham | VARCHAR(36) | FOREIGN KEY |
|  | diem\_danh\_gia | INT | (diem\_danh\_gia BETWEEN 1 AND 5) |
|  | binh\_luan | TEXT |  |
|  | ngay\_tao | DATETIME | DEFAULT CURRENT\_TIMESTAMP |

# THIẾT KẾ HỆ THỐNG

## Sơ đồ lớp

## Kiến trúc hệ thống

Hệ thống được thiết kế theo kiến trúc 3 lớp (Three Layer Design): Controller (API) ↔ Service (Business Logic) ↔ Repository (Data Access). Service là phần trung gian giữa Repository và Controller, chứa các business logic. Controller giao tiếp với Service và Service giao tiếp với Repository. Áp dụng Three layer design đảm bảo code sạch và dễ bảo trì, mở rộng, tách biệt rõ ràng các tầng, dễ kiểm thử.

# ỨNG DỤNG MÔ HÌNH MVC

Controller: Tiếp nhận request, trả response (Thiết kế API: code, message, result).

Service: Xử lý nghiệp vụ, validate, mã hóa mật khẩu, sinh JWT, v.v.

Repository: Truy xuất dữ liệu với JPA/Hibernate

Khi lỗi cần respond ra một thông báo có ý nghĩa.

A black screen with white text

AI-generated content may be incorrect.Sử dụng thư viện lombok để tự tạo ra getter, setter, contructor khi run, vì khi dự án phức tạp, đối tượng có nhiều thuộc tính thì code trông sẽ phức tạp hơn, khó bảo trì hơn. Hình sau là khi chạy sẽ tạo ra getter Username (Trong thư mục target)

Sử dụng thư viện mapstruct để ánh xạ các đối tượng và dto có cùng tên thuộc tính.

Xây dựng api xác thực tài khoản

Mã hóa mật khẩu bảng nguoi\_dung, sử dụng thuật toán hash bcrypt. sử dụng interface PasswordEncoder được cung cấp hông qua dependency spring-security-crypto

Ví dụ: mật khẩu “thoai123xyz” sẽ được mã hóa thành

“$2a$09$oLtkcfuozCndWoHbwzmW6OG8TnKP2.h/pdMKCiOLOyqcnwB0fUTtK”

|  |
| --- |
| A screenshot of a computer  AI-generated content may be incorrect. |
| A screenshot of a computer  AI-generated content may be incorrect. |

Khi triển khai thực tế thì không thể bắt người dùng đăng nhập lại mỗi khi chuyển trang, vì vậy JWT (JSON Web Token) sinh ra để đáp ứng việc này

|  |
| --- |
| A diagram of a computer code  AI-generated content may be incorrect. |

|  |
| --- |
| Dự án tiến hành triển khai JWT bằng thư viện nimbus-jose-jwt (phiên bản 10.3).  Thiết kế Token: Sử dụng thuật toán HS512, token có hiệu lực trong vòng 4 tiếng.  Tiến hành đăng nhập, thì kết quả được thiết kế trả về như hình sau: |
|  |
| Vào trang <https://jwt.io/> để giải mã token trên, hiển thị kết quả như hình sau, bao gồm thông tin thuật toán, trang web đã ban hành, username, thời gian hết han, thời gian ban hành, vai trò đúng như đã thiết kế trong mã nguồn. |
|  |

# KẾT LUẬN

Tóm tắt lại quá trình và kết quả hiện có.

Tóm tắt kết quả đạt được?

Kết quả/chức năng hài lòng nhất.

Viết 10 dòng, hoặc nửa trang.

Viết tốt được **2 điểm**

Đăng ký/đăng nhập: Kiểm tra trùng lặp, mã hóa mật khẩu, sinh JWT.

Quản lý người dùng: Phân quyền, cập nhật, xóa (chỉ admin).

Quản lý đơn hàng: Tạo đơn, cập nhật trạng thái,.

API trả về chuẩn: code, message, result, xử lý lỗi tập trung.

Mã hóa mật khẩu: BCrypt, không lưu plain text.

JWT: Token xác thực, lưu thông tin user, role, thời gian hết hạn.

Chuẩn hóa 3NF: Giảm dư thừa, tăng toàn vẹn dữ liệu.

ACID: Giao dịch đảm bảo nhất quán, rollback khi lỗi.

Tương lai sẽ triển khai các bảng còn lại theo cơ sở dữ liệu

Chức năng hài lòng nhất là xác thực JWT và phân quyền theo RBAC, giúp bảo vệ API hiệu quả.

**TÀI LIỆU THAM KHẢO  
(Không bắt buộc theo chuẩn quốc tế)**

Họ và tên tác giả 1, Họ và tên tác giả 2, …, Tên bài tham khảo, năm.

Tên bài viết trang web. Link: http:abc.com.vn (Ngày truy cập). Đối với tài liệu online hạn chế/ko tham khảo các blog công nghệ, wikipedia, facebook, youtube, mạng xã hội…

Trình bày không đúng style TLTK **– 1 điểm**.

**PHỤ LỤC PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành viên** | **Nhiệm vụ** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

***(Chú ý: Ghi rõ từng nhiệm vụ chi tiết của mỗi thành viên).***

Khi làm bài thu hoạch SV phải thực hiện đúng các quy định sau:

1. Sử đụng đúng Template này. Không đúng thì **– 2 điểm**.
2. Copy y chan trên Internet thì gọi là đạo văn và **nhận 0 điểm** cho bài thu hoạch.
3. Nộp đúng hạn theo thông báo của GV.
4. Không trình bày code trong nội dung bài thu hoạch. Nếu muốn thì trình bày tại phụ lục.

Sản phẩm:

1. File word (thêm pdf) báo cáo, ko in ra, nộp file mềm.
2. Slide thuyết trình.
3. Code

**PHỤ LỤC DEMO**

Chạy sản phẩm các bước sau

**Bước 1:**

* …
* …

**Bước 2:**

* …
* …

**Bước n:**

* …
* …

**PHỤ LỤC HÌNH**

Hình 1