**FPT UNIVERSITY**

**FPT POLYTECHNIC COLLEGE**

****

**MÔ TẢ DỰ ÁN**

**WEBSITE BÁN THỨC ĂN NHANH**

**SINH VIÊN: Nguyễn Thanh Phúc**

**MSSV: PS35206**

**LỚP: NET105 C# -SD18318**

**GIẢNG VIÊN : T. PHAN VIẾT THẾ**

***Ho Chi Minh City, June 2024***

Mục lục

[1. Giới thiệu dự án 3](#_Toc169098992)

[1.1. Mục Đích 3](#_Toc169098993)

[1.2. Phạm Vi 3](#_Toc169098994)

[2. Kiến trúc hệ thống 3](#_Toc169098995)

[3. Chi tiết hệ thống 3](#_Toc169098996)

[3.1. Project Api 3](#_Toc169098997)

[ Folder Data 3](#_Toc169098998)

[ Folder Context 3](#_Toc169098999)

[ Folder Models 3](#_Toc169099000)

[ Folder Mapper 4](#_Toc169099001)

[ Folder Services 4](#_Toc169099002)

[ Folder Controller 4](#_Toc169099003)

[ appsetting.json 7](#_Toc169099004)

[ Program.cs 7](#_Toc169099005)

[3.2. Project Mvc 7](#_Toc169099006)

[ Folder Areas 7](#_Toc169099007)

[ Folder Controller 7](#_Toc169099008)

[ Folder Extentions 8](#_Toc169099009)

[ Folder Models 8](#_Toc169099010)

[ Folder Services 8](#_Toc169099011)

[ Folder Views 8](#_Toc169099012)

[ appsetting.json 8](#_Toc169099013)

[4. Thông tin liên hệ 8](#_Toc169099014)

# 1. Giới thiệu dự án

## 1.1. Mục Đích

* Dự án này nhằm xây dựng một ứng dụng web sử dụng ASP.NET Core bao gồm hai phần chính:
* **API**: Cung cấp các dịch vụ RESTful cho các client.
* **MVC**: Giao diện người dùng quản lý và tương tác với hệ thống thông qua mô hình MVC (Model-View-Controller).

## 1.2. Phạm Vi

* **API**: Cung cấp các endpoints cho CRUD operations, xác thực và phân quyền, và các tính năng khác.
* **MVC**: Giao diện người dùng cho quản trị viên và khách hàng để tương tác với dữ liệu và các dịch vụ được cung cấp bởi API.

# 2. Kiến trúc hệ thống

* **ASP.NET Core MVC: Dùng cho front-end quản lý và hiển thị dữ liệu.**
* **Services dành cho thanh toán VnPay.**
* **ASP.NET Core Web API: Dùng để cung cấp các dịch vụ RESTful.**
* **Entity Framework Core: Dùng để tương tác với cơ sở dữ liệu.**
* **Identity: Dùng để quản lý người dùng và quyền hạn.**

# 3. Chi tiết hệ thống

## 3.1. Project Api

### Folder Data

* Chức các class đại diện các thực thể như categories, foods, users, orders, reviews, coupons…
* Thể hiện các ràng buộc Data validation, Model Relational Attribute**.**

### Folder Context

* Chứa lớp IdentityDbContext định nghĩa các DbSet, các fluent API giúp ràng buộc các mối quan hệ giữa các thực thể và thiết lập các giá trị mặc định như role, admin**.**

### Folder Models

* Chứa các View Models thông tin các thuộc tính phù hợp để CRUD**.**

### Folder Mapper

* Chứa profile để tạo ra các map hỗ trợ chuyển đổi các đối tượng nhanh chóng hơn.

### Folder Services

* Chứa các Interface và các Class implement các interface đó. Mô hình này giúp cho code được rõ ràng, dễ dàng sửa chữa hơn.
* Các interface cho category, food, order, review, account, email, coupon.

### Folder Controller

* Chứa các dịch vụ RestfulAPI
* Mô tả chi tiết:
* API Categories

*+HttpGet: api/categories*

Cho phép tất cả người dùng lấy thông tin tất cả danh mục**.**

*+HttpGet: api/categories/{id}*

Cho phép tất cả người dùng lấy thông tin chi tiết 1 danh mục.

*+ HttpPost: api/categories*

Thêm 1 danh mục mới. Chỉ admin mới được thực hiện.

*+ HttpPut: api/categories/{id}*

Cập nhật 1 danh mục theo id. Chỉ admin mới được thực hiện.

*+ HttpDelete: api/categories/{id}*

Xóa 1 danh mục theo id. Chỉ admin mới được thực hiện. Có thể không cho phép xóa danh mục.

* API Foods

*+HttpGet: api/foods*

Cho phép tất cả người dùng lấy thông tin tất cả sản phẩm.

*+HttpGet: api/foods/{id}*

Cho phép tất cả người dùng lấy thông tin chi tiết 1 sản phẩm.

*+HttpGet: api/foods/news*

Cho phép tất cả người dùng lấy thông tin 5 sản phẩm mới nhất.

*+HttpGet: api/foods/bestselling*

Cho phép tất cả người dùng lấy thông tin sản phẩm bán chạy nhất.

*+ HttpPost: api/foods*

Thêm 1 sản phẩm mới. Chỉ admin mới được thực hiện.

*+ HttpPut: api/foods/{id}*

Cập nhật 1 sản phẩm theo id. Chỉ admin mới được thực hiện.

*+ HttpDelete: api/categories/{id}*

Xóa 1 danh mục theo id. Chỉ admin mới được thực hiện. Có thể không cho phép xóa sản phẩm.

* API Coupons

*+HttpGet: api/coupons*

Cho phép admin lấy thông tin tất cả mã giảm giá.

*+HttpGet: api/coupons/{id}*

Cho phép tất cả người dùng lấy thông tin chi tiết 1 mã giảm giá theo id.

*+HttpGet: api/coupons/code/{code*}

Cho phép tất cả người dùng lấy thông tin chi tiết 1 mã giảm giá theo code.

*+ HttpPost: api/coupons*

Thêm 1 mã giảm giá mới. Chỉ admin mới được thực hiện.

*+ HttpPut: api/coupons/{id}*

Cập nhật 1 mã giảm giá theo id. Chỉ admin mới được thực hiện.

*+ HttpDelete: api/coupons/{id}*

Xóa 1 mã giảm giá theo id. Chỉ admin mới được thực hiện.

* API Orders

*+HttpGet: api/Orders*

Cho phép tất cả người dùng (đăng nhập) lấy thông tin tất cả Order (người dùng sẽ chỉ lấy thông tin hóa đơn của mình).

*+HttpGet: api/Orders/{id}*

Cho phép tất cả người dùng (đăng nhập) lấy thông tin chi tiết 1 hóa đơn.

*+ HttpPost: api/Order*

Cho phép tất cả người dùng (đăng nhập) đặt hàng.

*+ HttpPut: api/Orders/update-status*

Cập nhật trạng thái của 1 đơn hàng. Người dùng phải đăng nhập

API Reviews

*+HttpGet: api/Reviews/{foodID}*

Cho phép tất cả người dùng lấy thông tin tất cả Review của 1 sản phẩm theo Id sản phẩm.

*+ HttpPost: api/Reviews*

Cho phép tất cả người dùng (đăng nhập) đánh giá.

* API Auth

*+HttpPost: api/auth/login*

Cho phép tất cả người dùng đăng nhập vào website. Sau đó sẽ trả về LoginReponse chứa các thông tin như token, userI…

*+HttpPost: api/auth/register*

Cho phép tất cả người dùng đăng kí tài khoản của mình.

*+HttpGet: api/auth/getalluser*

Cho phép admin xem thông tin tất cả người dùng.

*+ HttpGet: api/auth/{id}*

Cho phép tất cả người dùng (đăng nhập) xem thông tin người dùng (người dùng sẽ chỉ lấy thông tin của mình).

*+ HttpPut: api/auth/setrole/{id}*

Admin được phép chỉnh sửa role của người dùng.

*+ HttpPut: api/auth/updateuser/{id}*

Tất cả người dùng đăng nhập được phép đổi thông tin tài khoản của mình.

* API Emails

*+HttpPost: api/email*

Cho phép tất cả người dùng gửi nhận mail cần thiết

*+HttpPost: api/email/requestforgotpassword*

Yêu cầu gửi mã xác nhận khi người dùng quên mật khẩu

*+HttpPost: api/email/resetpassword*

Đổi mật khẩu cho người dùng

*+ HttpPost: api/auth/resetpassword*

Tất cả người dùng được phép đổi mật khẩu tài khoản.

### appsetting.json

* + Cấu hình chuối kết nối cơ sở dữ liệu và đối tượng Smtp để gửi mail và các thuộc tính của Jwt.

### Program.cs

* Chứa các đăng kí Service như các repository, DbContext, AutoMapper, Identity, các cơ chế cho Jwt.
* Chứa các Middleware cần thiết cho ứng dụng.

## 3.2. Project Mvc

### Folder Areas

* Nơi chứa template mvc dành cho Admin gồm controller, models, views, data.
* Các controller dành cho Admin quản lí các category, food, order, account sử lí dữ liệu trước khi gọi api và truyền dữ liệu cho view hiển thị các thông tin.

### Folder Controller

* Các controller dành cho người dùng sử lí, gửi các dữ liệu cần thiết cho view hiển thị.

### Folder Extentions

* Các tiện ích cho ứng dụng như kiểm tra file có phải là imgae không, hỗ trợ chuyển đổi json sang object và ngược lại, thư viện VnPay.

### Folder Models

* Chứa các models bổ sung như CartVM, PagedResponse, ResetPassword, VnPaymentResponseModel.

### Folder Services

* Chứa interface và class VnPayService hỗ trợ thanh toán qua VnPay.

### Folder Views

* Chứa các file razor view cho việc hiển thị dữ liệu và gửi dữ liệu qua controller.

### appsetting.json

* Cấu hình chuỗi kết nối và các thuộc tính VnPay

# 4. Thông tin liên hệ

Nếu bạn gặp vấn đề hoặc có câu hỏi, vui lòng liên hệ với tôi qua email nguyenphuc14112003@gmail.com hoặc zalo: 0898827656.