**ĐỀ THI THỬ SỐ 1**

**MÔN: PHÁT TRIỂN ỨNG DỤNG WEB NÂNG CAO**

**LỚP CTK44 – THỜI GIAN: 90 PHÚT**

**CÂU 1: (2 điểm)**

1. Học phần Phát triển ứng dụng Web nâng cao đã trải qua bao nhiêu buổi thực hành?
2. Có bao nhiêu bài thực hành (Lab) chính thức? Trong số đó, bạn thấy thích nhất là bài thực hành nào? Vì sao?
3. Những nội dung chính của bài thực hành (Lab) số 3 là gì?

***Trả lời:***

1. Học phần Phát triển ứng dụng Web nâng cao đã trải qua 8 buổi
2. Có 6 bài thực hành chính thức, thích nhất là lab 6 bởi bài thực hành đó em đã hoàn thiện giao diện và xây dựng được gần hết các chức năng đã đề ra ở đề bài
3. Bài thực hành số 3 với nội dung chính là chức năng cho quản trị viên

* Xây dựng từ đầu đến cuối các chức năng cơ bản dành cho người quản trị của một trang blog đơn giản. Giao diện dành cho người quản lý blog được phác thảo
* Nội dung chính gồm có:

+ Tạo Area co phân hệ dành cho người quản trị viên

+ Xây dựng chức năng xem danh sách và tìm bài viết

+ Sử dụng gói thư viện Mapster

+ Xây dựng chức năng thêm/cập nhật bài viết

+ Bổ sung tính năng upload hình ảnh

+ Sử dụng FluentValidation để kiểm tra tính hợp lệ của dữ liệu

+ Ghi nhật ký hệ thống sử dụng NLog

**CÂU 2: (2 điểm)**

Cho ba lớp Category (danh mục), Candy (kẹo) và CandyContext như sau:

public class Category

{

// Mã số danh mục

public int Id { get; set; }

// Tên danh mục

public string Name { get; set; }

// Cho biết hiển thị lên menu hay không

public [……] ShowOnMenu { get; set; }

}

public class Candy

{

// Mã số

public int Id { get; set; }

// Tên mặt hàng kẹo

public string Name { get; set; }

// Giá bán (đơn vị là USD, luôn lớn hơn 0 và

// có phần lẻ thập phân. Ví dụ: 5.99).

public [……] Price { get; set; }

// Ngày hết hạn sử dụng

public [……] ExpirationDate { get; set; }

// Mã số danh mục (Mỗi mặt hàng kẹo phải thuộc

// đúng một danh mục)

public [……] CategoryId { get; set; }

public Category Category { get; set; }

}

public class CandyContext : DbContext

{

public DbSet<Category> Categories { get; set; }

public DbSet<Candy> Candies { get; set; }

}

1. Hãy xác định kiểu dữ liệu cho các thuộc tính ShowOnMenu, Price, ExpirationDate, và CategoryId. (Sinh viên sửa trực tiếp mã nguồn ở trên bằng cách thay tên kiểu dữ liệu vào chỗ [……])
2. Giả sử, ta có lớp CandyService thực thi interface ICandyService chứa phương thức GetCandiesAsync. Hãy định nghĩa phương thức này để tìm các mặt hàng kẹo chưa hết hạn sử dụng theo tên mặt hàng (nếu giá trị truyền vào khác rỗng), đơn giá (miền giá min-max có giá trị hợp lệ), danh mục (nếu có truyền giá trị) hoặc tên danh mục (tìm chính xác theo tên danh mục nếu truyền giá trị khác rỗng) và sắp xếp kết quả tăng dần theo tên mặt hàng kẹo.

public async Task<IList<Candy>> GetCandiesAsync(

string name = null,

int? categoryId = null,

string categoryName = null,

decimal? minPrice = null,

decimal? maxPrice = null,

CancellationToken cancellationToken = default)

{

// Sinh viên viết mã lệnh ở đây

}

1. Price: float

ShowOnMenu: bool

ExpirationDate: dateTime

CategoryId: int

**CÂU 3: (3 điểm)**

Bạn An sử dụng kết quả ở câu 2 để định nghĩa một API Endpoint như sau:

app.MapGet("/candies", (ICandyService service, CandyFilterModel model) =>

{

var candies = await service.GetCandiesAsync(

model.Name, model.CategoryId, model.CategoryName,

model.MinPrice, model.MaxPrice);

return candies;

})

.WithName("GetCandies")

.Produces<Candy>()

.WithOpenApi();

1. Hãy giúp bạn An định nghĩa lớp CandyFilterModel.
2. Hãy giúp bạn An tìm những lỗi sai trong đoạn mã trên và sửa lại cho đúng.
3. Hãy giải thích ý nghĩa của toàn bộ đoạn mã trên (sau khi đã sửa).
4. Hãy nêu một đề xuất để cải thiện hiệu suất cho API Endpoint trên.

***Trả lời:***

**CÂU 4: (3 điểm)**

Sau khi xây dựng API Endpoint (ở câu 3) hoàn chỉnh, bạn An tạo một React component – tên là CandyList – để hiển thị danh sách mặt hàng kẹo lên ứng dụng Web. Bạn An định nghĩa mã nguồn cho component này như sau:

import React, { useState } from 'react'

const CandyList = () => {

    const [candies, setCandies] = useState([]);

    fetch(`https://localhost:7076/candies`)

        .then(response => response.json())

        .then(data => setCandies(data));

    return (

        <div class="product-list">

            {candies.map((item) => (

               <div className="product-item">

                    <h1>item.name</h1>

                    <p>$ { item.price }</p>

                </div>

            ))}

        </div>

    )

}

export default CandyList;

1. Hãy cho biết đoạn mã trên chứa những lỗi sai hoặc cảnh báo nào? Giải thích nguyên nhân gây ra lỗi.
2. Hãy giúp bạn An cách sửa lỗi và viết lại mã nguồn đúng.
3. Hãy bổ sung mã lệnh để hiển thị dòng thông báo “***Đang tải dữ liệu …***” trong khi chờ kết quả trả về từ API hoặc thông báo lỗi “***Không thể tải dữ liệu***” nếu gặp lỗi khi gọi API.

***Trả lời:***

**--- HẾT ---**