

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC KINH TẾ THÀNH PHỐ HỒ CHÍ MINH (UEH)
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ



ĐỒ ÁN MÔN HỌC

ĐỀ TÀI

**Ứng dụng LinkedList trong quản lý bài đăng
trên mạng xã hội**

Học phần: Cấu Trúc Dữ Liệu & Giải Thuật

Danh sách nhóm:

1. Sinh viên: Nguyễn Cáp Bảo Huyền
2. Sinh viên: Tô Gia Linh
3. Sinh viên: Võ Ngọc Cẩm Tâm
4. Sinh viên: Cù Thị Thùy Trang

Chuyên ngành: AN TOÀN THÔNG TIN

Khóa: K49

Giảng viên hướng dẫn: TS. Đặng Ngọc Hoàng Thành

Tp.Hồ Chí Minh, Ngày 02 Tháng 05 năm 2024

MỤC LỤC

CHƯƠNG 1: PHÂN TÍCH VÀ THIẾT KẾ LỚP SINGLY LINKED LIST	3
1.1. Các Khái Niệm Liên Quan	3
1.2. Cài Đặt Và Cấu Trúc Lớp Post	5
1.3. Các Chức Năng Trên Lớp Post	7
CHƯƠNG 2: ỨNG DỤNG CỦA SINGLY LINKED LIST TRONG QUẢN LÝ BÀI ĐĂNG TRÊN MẠNG XÃ HỘI	9
2.1. Giới Thiệu Ứng Dụng PostHub	9
2.2. Các Chức Năng Của Ứng Dụng PostHub	9
2.3. Hướng Dẫn Sử Dụng Ứng Dụng PostHub	12
CHƯƠNG 3: THIẾT KẾ GIAO DIỆN	14
3.1. Giao Diện Chính	14
3.2. Chi Tiết Chức Năng	15
CHƯƠNG 4: THẢO LUẬN & ĐÁNH GIÁ	17
4.1. Các Kết Quả Nhận Được	17
4.2. Một Số Thử Thách	17
4.3. Hướng Phát Triển	17
PHỤ LỤC	19
Link Mã Nguồn	19
Hướng Dẫn Cách Cài Đặt Để Chạy	19
Nhiệm Vụ Của Các Thành Viên	22
TÀI LIỆU THAM KHẢO	23

CHƯƠNG 1: PHÂN TÍCH VÀ THIẾT KẾ LỚP SINGLY LINKED LIST

1.1. Các Khái Niệm Liên Quan

A. Khái niệm danh sách liên kết

- Mỗi danh sách liên kết được cấu tạo từ các phần tử được gọi là các nút.
- Mỗi một nút gồm có 2 phần: phần dữ liệu và phần thông tin liên kết.
- Mỗi danh sách liên kết luôn có một nút đầu (header) để chỉ định vị trí đầu tiên của danh sách.

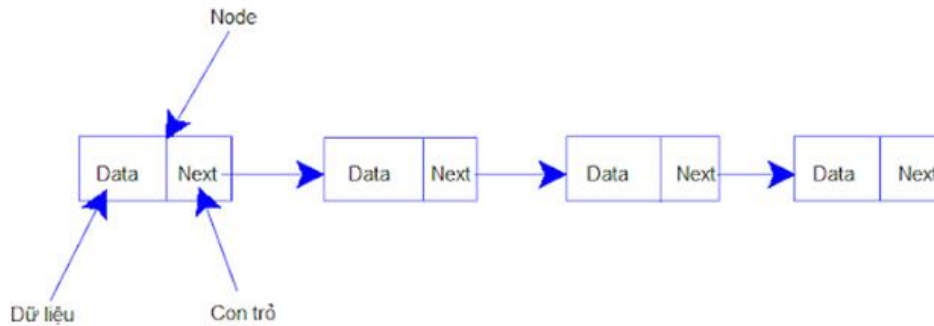
B. Phân loại danh sách liên kết:

- Có các loại danh sách liên kết như: danh sách liên kết đơn, danh sách liên kết đôi và danh sách liên kết vòng.
- Danh sách liên kết đơn: Mỗi nút chỉ có một liên kết (link) trỏ đến nút tiếp theo.
- Danh sách liên kết đôi: Mỗi nút có hai liên kết, một trỏ đến nút tiếp theo (Flink) và một trỏ đến nút trước đó (Blink).
- Danh sách liên kết vòng: Mỗi nút trong danh sách liên kết vòng có hai phần là dữ liệu lưu trữ giá trị thực tế của nút và liên kết (link) trỏ đến nút tiếp theo trong danh sách.

C. Khái niệm danh sách liên kết đơn

- Danh sách liên kết đơn (Singly Linked List) là một cấu trúc dữ liệu động, nó là một danh sách mà mỗi phần tử đều liên kết với phần tử đứng sau nó trong danh sách. Mỗi phần tử (được gọi là một node hay nút) trong danh sách liên kết đơn là một cấu trúc có hai thành phần:
- Thành phần dữ liệu: lưu thông tin về bản thân phần tử đó.
- Thành phần liên kết: lưu địa chỉ phần tử đứng sau trong danh sách, nếu phần tử đó là phần tử cuối cùng thì thành phần này bằng NULL.

Mô hình của danh sách liên kết đơn:



D. Các thuật toán cơ bản của danh sách liên kết đơn

a) Thêm phần tử đầu vào danh sách liên kết đơn

Ta xác định xem danh sách liên kết đơn có rỗng hay không. Nếu danh sách đó rỗng, ta gán luôn Head vào Node cần thêm. Nếu danh sách không rỗng, ta trở liên kết từ Head vào Node mới. Sau đó mới gán lại Head vào Node này.

b) Thêm phần tử đuôi vào danh sách liên kết đơn

Tương tự như cách thêm phần tử đầu, ta sẽ xác định xem danh sách liên kết đơn có rỗng hay không. Nếu rỗng thì cho Node mới làm Tail. Nếu danh sách không rỗng, thì trở Tail sẵn có đến Node này rồi gán lại Tail vào Node mới được trở.

c) Thêm phần tử vào một điểm bất kỳ vào danh sách liên kết đơn

Ta đặt p là Node cần thêm, còn q là Node đứng trước vị trí cần thêm. Đầu tiên, ta sẽ kiểm tra xem Node q có gán với NULL hay không. Nếu có gán tức là danh sách rỗng. Khi đó chỉ cần gán p lên đầu là được. Nếu không, ta sẽ thực hiện theo các bước sau: trở $p \rightarrow \text{Next} = q \rightarrow \text{Next}$, sau đó $q \rightarrow \text{Next} = p$. Khi hoàn thành, phải kiểm tra tiếp q có phải nút cuối hay không. Nếu phải thì cần tiếp tục gán p vào Tail.

d) Xóa phần tử đầu khỏi danh sách liên kết đơn

Ta kiểm tra xem danh sách đó có rỗng không. Nếu có thì trực tiếp xóa đi và để giá trị bằng 0 là được. Còn nếu danh sách không rỗng thì thực hiện theo các bước sau: đầu tiên là gán lại Head vào vị trí đứng sau phần tử cần xóa, nhớ phải lưu Head lại. Sau đó mới tiến hành xóa.

e) Xóa phần tử ở điểm bất kỳ khỏi danh sách liên kết đơn

- Nếu cần xóa Node p sau một Node q bất kỳ, ta sẽ có 3 trường hợp cần xét:

- Nếu q là NULL suy ra danh sách rỗng, không cần xóa mà chỉ cần chỉnh về 0
- Nếu Next của q là NULL, chứng tỏ p là NULL, suy ra p không tồn tại để xóa
- Nếu p có tồn tại, kiểm tra xem p có phải Tail không, nếu có thì chỉ cần gán Tail lại vào q là được.

f) Duyệt danh sách liên kết đơn và in

Ta kiểm tra xem danh sách đã hoàn chỉnh hay chưa, ta sẽ gán một Node bằng Head. Sau đó kiểm tra xem Node đó NULL hay không. Nếu đã đạt tức là ta đã có dữ liệu của Node này. Tiếp tục thực hiện thao tác đó cho đến Node NULL, đó chính Tail của danh sách.

1.2. Cài Đặt Và Cấu Trúc Lớp Post

A. Cài đặt lớp Post

Lớp Post bao gồm một số thành phần cơ bản sau:

```

public class Node
{
    3 references
    public string Data { get; set; }
    10 references
    public Node Next { get; set; }

    1 reference
    public Node(string data)
    {
        Data = data;
        Next = null;
    }
}

0 references
public class LinkedList
{
    private Node head;

    0 references
    public void AddPost(string data)
    {
        Node newNode = new Node(data);
        if (head == null)
        {
            head = newNode;
        }
        else
        {
            Node current = head;
            while (current.Next != null)
            {
                current = current.Next;
            }
        }
    }
}

```

```

        current.Next = newNode;
    }
}
0 references
public void EditPost(int index, string newData)
{
    Node current = head;
    int currentIndex = 0;
    while (current != null && currentIndex < index)
    {
        current = current.Next;
        currentIndex++;
    }

    if (current != null)
    {
        current.Data = newData;
    }
}

0 references
public void DeletePost(int index)
{
    if (head == null) return;

    if (index == 0)
    {
        head = head.Next;
        return;
    }

    Node current = head;
    Node previous = null;
    int currentIndex = 0;

```

```

    while (current != null && currentIndex < index)
    {
        previous = current;
        current = current.Next;
        currentIndex++;
    }

    if (current != null)
    {
        previous.Next = current.Next;
    }
}

```

B. Cấu trúc lớp Post

Post được xây dựng dựa trên cấu trúc của danh sách liên kết đơn. Lớp có thể lựa chọn hiển thị một bài đăng trong một thư mục với các chức năng cơ bản: thêm bài đăng mới, sửa bài đăng, xóa bài đăng và được lưu trong liên kết của Node trong danh sách liên kết đơn.

1.3. Các Chức Năng Trên Lớp Post

Lớp Post sử dụng danh sách liên kết đơn để lưu trữ và quản lý các bài đăng một cách có hiệu quả, bao gồm các chức năng cơ bản của một danh sách liên kết đơn: thoát, thêm, sửa và xóa phần tử.

- Thêm Bài Đăng

- Thêm một bài đăng mới vào cuối danh sách liên kết.
- Tham số: data (string): Nội dung của bài đăng mới.
- Chức năng:
 - + Tạo một node mới với dữ liệu được cung cấp.
 - + Nếu danh sách đang trống (head là null), node mới này sẽ trở thành đầu danh sách.
 - + Nếu danh sách không trống, đi tới cuối danh sách và liên kết node mới này vào node cuối cùng.
 - + Trả về: Không có (void).

- Sửa Bài Đăng

- Sửa nội dung của một bài đăng tại vị trí chỉ định trong danh sách.
- Tham số:
 - + index (int): Chỉ số của bài đăng trong danh sách, bắt đầu từ 0.
 - + newData (string): Nội dung mới cho bài đăng.
- Chức năng:
 - + Duyệt qua danh sách bắt đầu từ đầu cho tới khi tìm thấy vị trí cần sửa.
 - + Nếu tìm thấy node tại chỉ số đó, cập nhật dữ liệu của node đó với nội dung mới.

+ Nếu không tìm thấy (index vượt quá kích thước danh sách), không làm gì cả.

+ Trả về: Không có (void).

- Xóa Bài Đăng

- Xóa bài đăng tại vị trí chỉ định trong danh sách.

- Tham số: index (int): Chỉ số của bài đăng cần xóa, bắt đầu từ 0.

- Chức năng:

+ Nếu danh sách trống hoặc index không hợp lệ, không làm gì cả.

+ Nếu xóa node đầu tiên (index = 0), cập nhật head thành node tiếp theo.

+ Nếu xóa node ở vị trí khác, tìm node trước đó của node cần xóa và cập nhật liên kết của nó để bỏ qua node cần xóa.

+ Trả về: Không có (void).

CHƯƠNG 2: ỨNG DỤNG CỦA SINGLY LINKED LIST TRONG QUẢN LÝ BÀI ĐĂNG TRÊN MẠNG XÃ HỘI

2.1. Giới Thiệu Ứng Dụng PostHub

Ứng dụng PostHub là một ứng dụng quản lý bài đăng trên mạng xã hội được xây dựng dựa trên cấu trúc dữ liệu Singly Linked List. Ứng dụng được thiết kế nhằm đưa ra một giao diện đơn giản và dễ dùng giúp người dùng có thể dễ dàng quản lý và đăng các bài viết trên mạng xã hội, Singly Linked List được áp dụng vào việc quản lý danh sách các bài đăng, mỗi Node của list sẽ chứa thông tin về một bài đăng cùng với việc đối chiếu tới các bài đăng trước và sau. Việc đó sẽ giúp ích nhiều và ngoài ra còn cho phép chúng ta thêm, sửa, xóa giữa các bài đăng một cách thuận tiện và nhanh chóng.

2.2. Các Chức Năng Của Ứng Dụng PostHub

Ứng dụng PostHub bao gồm một số chức năng sau:

A. Hiện thị danh sách bài đăng

Người dùng có thể xem danh sách các bài đăng.

```
4 references
private void HienThi()
{
    string query = "SELECT * FROM baiDang";
    SqlCommand cmd = new SqlCommand(query, conn);
    SqlDataReader dr = cmd.ExecuteReader();
    DataTable dt = new DataTable();
    dt.Load(dr);
    dgvbaiDang.DataSource = dt;
}
```

B. Hiện thị thông tin bài đăng

Người dùng có thể xem thông tin của bài đăng như mã bài đăng, nội dung bài đăng, tác giả, hình ảnh và ngày đăng.

```
1 reference
private void dgvbaiDang_CellClick(object sender, DataGridViewCellEventArgs e)
{
    int index = e.RowIndex;
    if (index >= 0)
    {
        txtPostID.Text = dgvbaiDang.Rows[index].Cells["postID"].Value.ToString();
        txtNoiDungBaiDang.Text = dgvbaiDang.Rows[index].Cells["noiDungBaiDang"].Value.ToString();
        txtTacGia.Text = dgvbaiDang.Rows[index].Cells["tacGia"].Value.ToString();
        dtptngayDang.Text = dgvbaiDang.Rows[index].Cells["ngayDang"].Value.ToString();
    }
}
```

C. Thêm bài đăng mới

Người dùng có thể thêm các bài đăng mới trên mạng xã hội và bài đăng sẽ được hiển thị vào danh sách các bài đăng.

```
1 reference
private void btnThem_Click(object sender, EventArgs e)
{
    if (txtPostID.Text == "" || txtNoiDungBaiDang.Text == "" || txtTacGia.Text == "" || dtpngayDang.Text == "")
    {
        MessageBox.Show("Vui lòng nhập đầy đủ thông tin");
        return;
    }
    else
    {
        string query = "INSERT INTO baiDang VALUES(@postID, @noiDungBaiDang, @tacGia, @ngayDang)";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@postID", txtPostID.Text);
        cmd.Parameters.AddWithValue("@noiDungBaiDang", txtNoiDungBaiDang.Text);
        cmd.Parameters.AddWithValue("@tacGia", txtTacGia.Text);
        cmd.Parameters.AddWithValue("@ngayDang", dtpngayDang.Value);
        cmd.ExecuteNonQuery();

        Post newPost = new Post
        {
            PostID = txtPostID.Text,
            noiDungBaiDang = txtNoiDungBaiDang.Text,
            tacGia = txtTacGia.Text,
            ngayDang = dtpngayDang.Value.ToString()
        };
        posts.AddLast(newPost);
        HienThi();
    }
}
```

D. Xóa bài đăng

Người dùng cũng có thể xóa bỏ các bài đăng cũ không cần thiết khỏi danh sách quản lý bài đăng trên mạng xã hội.

```

1 reference
private void btnXoaBaiDang_Click(object sender, EventArgs e)
{
    if (txtPostID.Text == "")
    {
        MessageBox.Show("Vui lòng nhập mã bài đăng cần xóa");
        return;
    }
    else
    {
        string query = "DELETE FROM baiDang WHERE postID = @postID";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@postID", txtPostID.Text);
        cmd.ExecuteNonQuery();
        // Xóa trong linkedList
        foreach (Post post in posts)
        {
            if (post.PostID == txtPostID.Text)
            {
                posts.Remove(post);
                break;
            }
        }
        HienThi();
    }
}

```

E. Sửa bài đăng

Ứng dụng cho phép người dùng có thể chỉnh sửa các thông tin cho bài đăng của mình bao gồm: nội dung bài đăng, tác giả và ngày đăng.

```

1 reference
private void btnSuaBaiDang_Click(object sender, EventArgs e)
{
    if (txtPostID.Text == "" || txtNoiDungBaiDang.Text == "" || txtTacGia.Text == "" || dtpngayDang.Text == "")
    {
        MessageBox.Show("Vui lòng nhập đầy đủ thông tin");
        return;
    }
    else
    {
        string query = "UPDATE baiDang SET noiDungBaiDang = @noiDungBaiDang, tacGia = @tacGia, ngayDang = @ngayDang WHERE postID = @postID";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@postID", txtPostID.Text);
        cmd.Parameters.AddWithValue("@noiDungBaiDang", txtNoiDungBaiDang.Text);
        cmd.Parameters.AddWithValue("@tacGia", txtTacGia.Text);
        cmd.Parameters.AddWithValue("@ngayDang", dtpngayDang.Value);
        cmd.ExecuteNonQuery();
        // Sửa trong linkedList
        foreach (Post post in posts)
        {
            if (post.PostID == txtPostID.Text)
            {
                post.noiDungBaiDang = txtNoiDungBaiDang.Text;
                post.tacGia = txtTacGia.Text;
                post.ngayDang = dtpngayDang.Value.ToString();
            }
        }
        HienThi();
    }
}

```

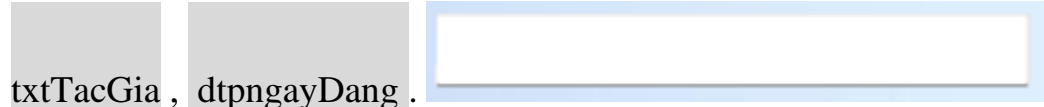
2.3. Hướng Dẫn Sử Dụng Ứng Dụng PostHub

A. Hiển thị bài đăng

- Danh sách bài đăng hiện tại được hiển thị trên DataGridView `dgvbaiDang`.
- Mỗi dòng trong DataGridView đại diện cho một bài đăng.
- Các cột hiển thị thông tin của bài đăng như ID, nội dung, tác giả, ngày đăng.

B. Thêm bài đăng mới

- Nhập đầy đủ thông tin vào các textbox: `txtPostID`, `txtNoiDungBaiDang`, `txtTacGia`, `dtpngayDang`.




- Click vào nút `btnThem`.
- Ứng dụng sẽ thêm bài đăng mới vào database và linked list.
- Danh sách bài đăng trên DataGridView sẽ được cập nhật.

C. Sửa bài đăng

- Chọn bài đăng cần sửa trong DataGridView.
- Sửa đổi thông tin bài đăng trong các textbox.
- Click vào nút `btnSuaBaiDang`.
- Ứng dụng sẽ cập nhật thông tin bài đăng trong database và linked list.
- Danh sách bài đăng trên DataGridView sẽ được cập nhật.

D. Xóa bài đăng

- Nhập ID bài đăng cần xóa vào textbox `txtPostID`. 
- Click vào nút `btnXoaBaiDang`.
- Ứng dụng sẽ xóa bài đăng khỏi database và linked list.
- Danh sách bài đăng trên DataGridView sẽ được cập nhật.

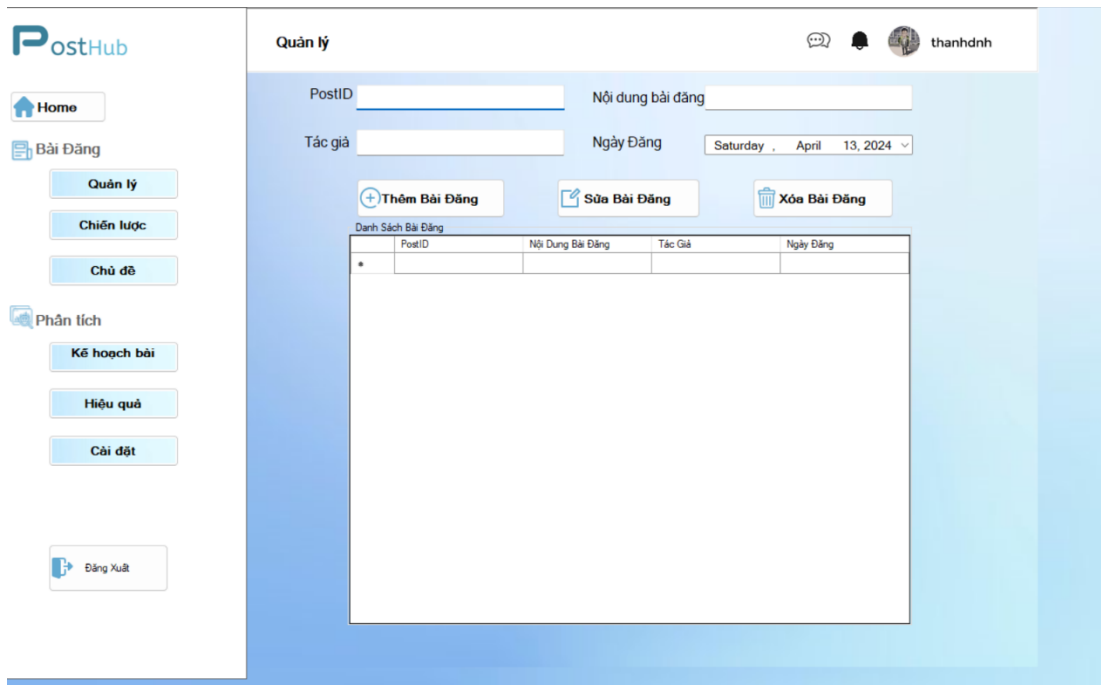
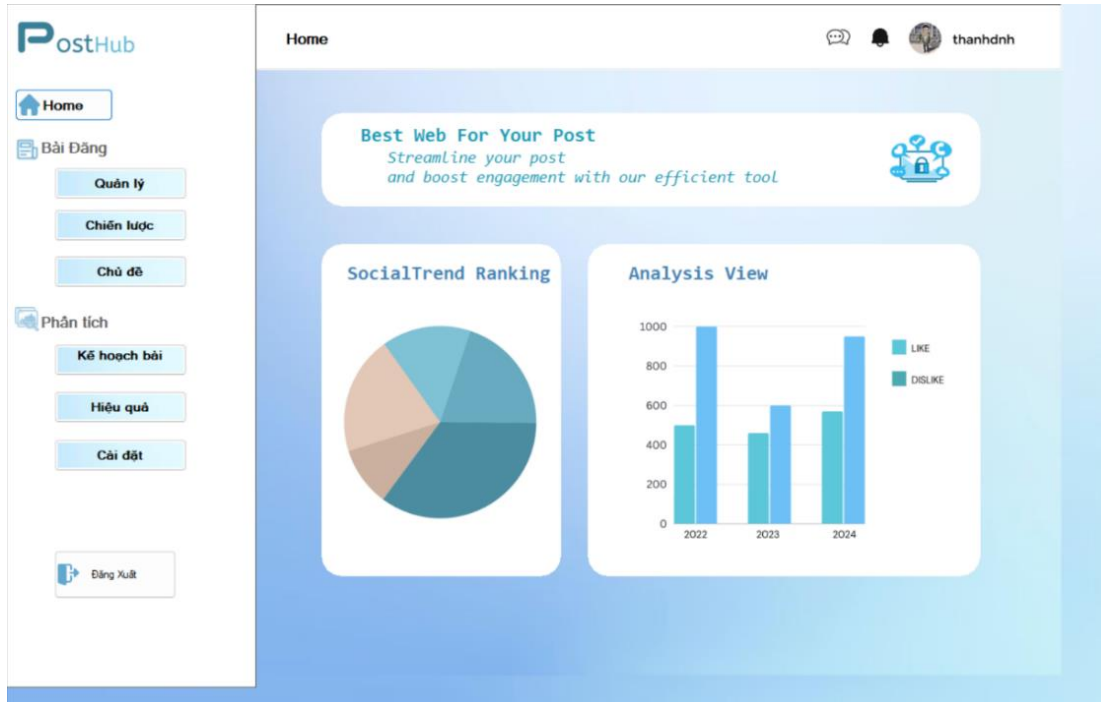
E. Xem thông tin bài đăng

- Click vào một dòng trong DataGridView `dgvbaiDang`.
- Thông tin của bài đăng được chọn sẽ được hiển thị trong các textbox `txtPostID`, `txtNoiDungBaiDang`, `txtTacGia`, `dtpngayDang`.

CHƯƠNG 3: THIẾT KẾ GIAO DIỆN

3.1. Giao Diện Chính

Giao diện mà chúng ta sẽ thấy khi chương trình khởi chạy như sau:



3.2. Chi Tiết Chức Năng

A. Hiển thị các bài đăng

Màn hình trống chính giữa sẽ là nơi để hiển thị danh sách các bài đăng được thêm vào từ máy tính. Khi đưa bài đăng lên từ nút thêm bài đăng, bài đăng sẽ được đưa vào danh sách và được đánh số theo số thứ tự tăng dần.

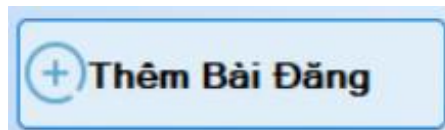
Mỗi bài đăng đều có các thông tin như ID, nội dung bài đăng, tác giả và ngày đăng. Các thông tin sẽ thuận tiện cho người dùng theo dõi.

The screenshot shows the PostHub management interface. On the left is a sidebar with the PostHub logo and navigation buttons: Home, Bài Đăng (with a sub-menu containing Quản lý, Chiến lược, and Chủ đề), Phân tích (with sub-menu items Kế hoạch bài, Hiệu quả, and Cài đặt), and Đăng Xuất. The main area is titled 'Quản lý' and features a form for adding or editing posts. The form includes fields for PostID (555), Nội dung bài đăng (Một ngày của tôi), Tác giả (Huyền), and Ngày Đăng (Wednesday, April 17, 2024). Below the form are three buttons: '+ Thêm Bài Đăng', 'Sửa Bài Đăng', and 'Xóa Bài Đăng'. A table titled 'Danh Sách Bài Đăng' displays a list of posts with columns for PostID, Nội Dung Bài Đăng, Tác Giả, and Ngày Đăng. The table contains two rows: one for PostID 333 (Chiến dịch mùa xuân, Trang, Apr 15 2024 2:48AM) and one for PostID 555 (Một ngày của tôi, Huyền, Apr 17 2024 2:48AM). A plus icon is visible in the first row of the table.

PostID	Nội Dung Bài Đăng	Tác Giả	Ngày Đăng
333	Chiến dịch mùa xuân	Trang	Apr 15 2024 2:48AM
555	Một ngày của tôi	Huyền	Apr 17 2024 2:48AM

B. Các nút điều khiển đối với bài đăng

- Nút thêm bài đăng:



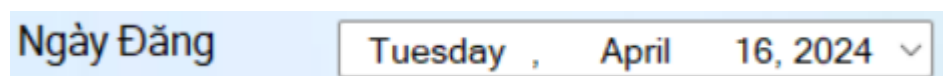
- Nút sửa bài đăng:



- Nút xóa bài đăng:



- Thanh điều chỉnh ngày đăng:



CHƯƠNG 4: THẢO LUẬN & ĐÁNH GIÁ

4.1. Các Kết Quả Nhận Được

- Tạo và quản lý các bài đăng thông qua Linked List.
- Chức năng tổng hợp dữ liệu khi người dùng thêm, sửa, xóa bài đăng thì chương trình cập nhật cả database và linked list để đảm bảo dữ liệu được đồng bộ.
- Ngoài các chức năng thêm, sửa, xóa bài đăng thì còn có chức năng xử lý sự kiện click trên DataGridView: Sự kiện `dgvbaiDang` được kích hoạt khi người dùng click vào một ô trong DataGridView `dgvbaiDang`.

4.2. Một Số Thử Thách

- Hiệu suất: Khi sử dụng linked list để lưu trữ dữ liệu bài đăng có thể làm ảnh hưởng đến hiệu suất khi số lượng bài đăng lớn.
- Bảo mật: Chương trình chưa có biện pháp bảo mật để ngăn chặn truy cập trái phép vào dữ liệu.
- Lỗi ngoại lệ: Chương trình có thể gặp lỗi ngoại lệ khi xảy ra các trường hợp không mong muốn.
- Xung đột dữ liệu: Có thể xảy ra xung đột dữ liệu khi nhiều người dùng truy cập và sửa đổi dữ liệu cùng lúc.

4.3. Hướng Phát Triển

- Hỗ trợ đa ngôn ngữ: Cung cấp nhiều ngôn ngữ khác nhau cho giao diện và chức năng, nâng cấp nền tảng và tiếp cận đến nhiều người dùng khác.
- Quản lý và bảo mật tài khoản: Cung cấp các ID riêng biệt cho từng cá nhân, mật khẩu có tính bảo mật cao, và bảo vệ an toàn thông tin người dùng.

- Phân tích hiệu quả bài viết: Thống kê về lượt tương tác của các cá nhân đối với từng bài viết của mình.
- Lưu trữ bài viết: Cung cấp cho người dùng một kho lưu trữ để có thể lưu trữ các bài viết đã đăng trước đó hoặc là các bài viết nháp.
- Gợi ý nội dung: Cập nhật các xu hướng, sở thích, và hiệu quả của các bài viết trước đây để đưa ra các gợi ý chủ đề phù hợp.
- Phân tích xu hướng: Giúp người dùng nắm bắt xu hướng nội dung và hành vi hiện tại trên mạng xã hội.
- Giao diện thân thiện: Phát triển và tối ưu hóa giao diện đảm bảo cho người dùng có trải nghiệm dễ dàng và tiện lợi.


PHỤ LỤC

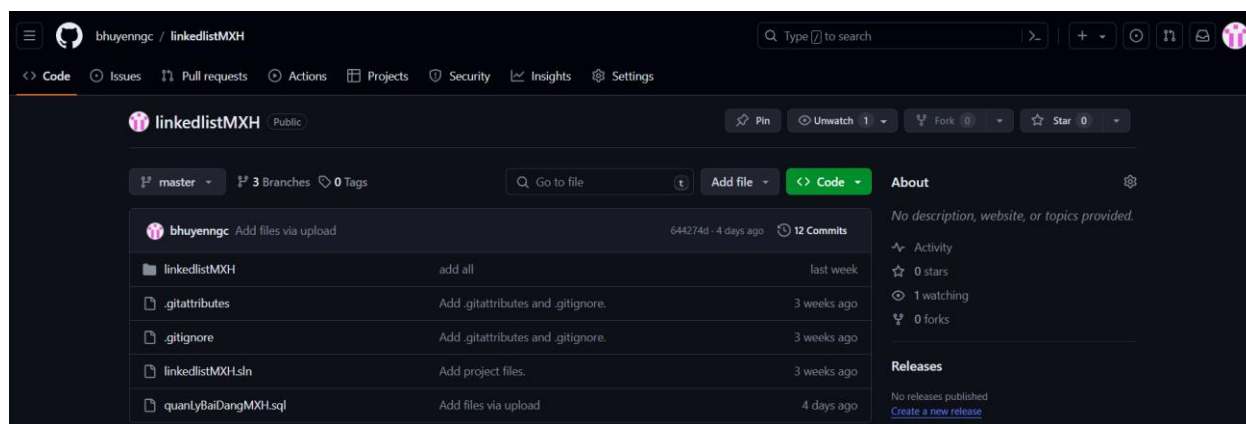
Link Mã Nguồn

<https://github.com/bhuyenngc/linkedListMXH>

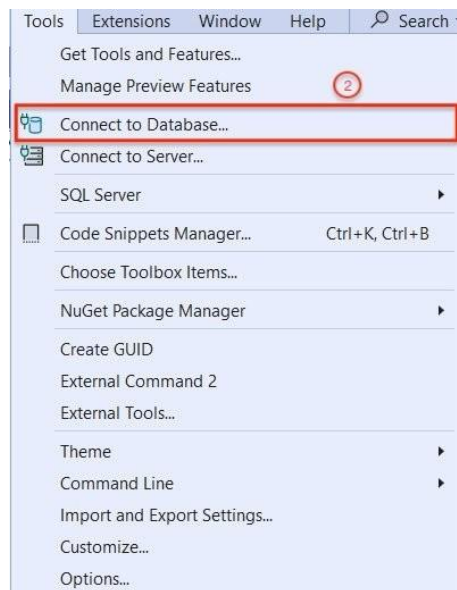
Hướng Dẫn Cách Cài Đặt Để Chạy

B1: Truy cập vào đường link GitHub đã được nhóm đăng tải, sau đó tải file `quanLyBaiDangMXH.sql` về máy tính và tiếp theo nhấn vào chỗ Code (như ảnh)

 và bấm COPY.



B2: Vào project của bạn, chọn Tools → chọn Connect to Database.



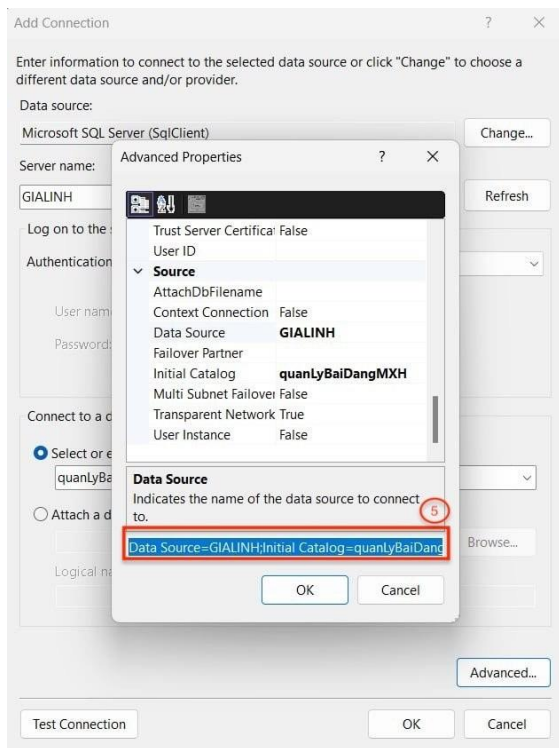
B3:

- Server Name: Là tên máy chủ SQL Server hoặc địa chỉ IP của máy tính.
- Authentication: Chọn “Windows Authentication” .
- Select or enter a database name: Chọn quanLyBaiDangMXH.

B4: Click “Test Connection” để kiểm tra kết nối → click OK.

The screenshot shows the 'Add Connection' dialog box. The 'Data source' is set to 'Microsoft SQL Server (SqlClient)'. The 'Server name' is 'GIALINH'. The 'Authentication' is 'Windows Authentication'. The 'Connect to a database' section has 'Select or enter a database name:' selected, with 'quanLyBaiDangMXH' entered. The 'Test Connection' button is highlighted with a red box and a circled number 1. The 'Server name' dropdown is highlighted with a red box and a circled number 2. The 'Authentication' dropdown is highlighted with a red box and a circled number 3. The 'Advanced...' button is highlighted with a red box and a circled number 4. The 'Connect to a database' section is highlighted with a red box and a circled number 5.

B5: Click Advanced → copy phần Data Source (như ảnh) → click OK.



B6: Thay thế Data Source trong dòng lệnh này bằng Data Source vừa copy được

```
string strConn = @"Data Source=LAPTOP-02MQR8KD;Initial Catalog=quanLyBaiDangMXH;Integrated Security=True";
```

B7: Đã hoàn thành kết nối SQL Server với Visual Studio.

Nhiệm Vụ Của Các Thành Viên

Thành viên	Nhiệm vụ	Hoàn thành
Nguyễn Cáo Bảo Huyền	Tổng hợp source code, thiết kế giao diện menu chính và chi tiết các chức năng	100%
Tô Gia Linh	Base code, thảo luận, đánh giá về các kết quả nhận được và hướng phát triển, làm slide	100%
Võ Ngọc Cẩm Tâm	Base code, khái niệm, trình bày thuật toán và cài đặt thuật toán, làm slide	100%
Cù Thị Thùy Trang	Lý thuyết mô tả ứng dụng PostHub, phân tích và thiết kế lớp, tổng hợp nội dung, làm đồ án	100%

TÀI LIỆU THAM KHẢO

<https://teky.edu.vn/blog/danh-sach-lien-ket-don/>

<https://topdev.vn/blog/danh-sach-lien-ket-don-trong-c/?amp>

<https://howkteam.vn/q/ket-noi-giua-sql-2019-voi-visual-2019-52206>

<https://www.youtube.com/watch?v=mgM-rxOrabI&t=68s>