

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Vĩnh Long, ngày 11 tháng 01 năm 2020
Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

Vĩnh Long, ngày 11 tháng 01 năm 2026
Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

CHƯƠNG 1 MỤC LỤC

- **LỜI CẢM ƠN**
- **Chương 1: TỔNG QUAN**
- 1.1. Lý do chọn đề tài
- 1.2. Mục tiêu nghiên cứu
- 1.3. Phương pháp nghiên cứu
- 1.4. Đối tượng và phạm vi nghiên cứu
- **Chương 2: NGHIÊN CỨU LÝ THUYẾT**
- 2.1. Tổng quan về kiến thức chung
- 2.1.1. Internet và World Wide Web
- 2.1.2. Ngôn ngữ HTML, CSS, JavaScript
- 2.1.3. Cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu
- 2.2. Cơ sở lý thuyết chuyên môn
- 2.2.1. Giới thiệu về ASP.NET
- 2.2.2. Lịch sử phát triển và các phiên bản của ASP.NET
- 2.2.3. Ưu điểm và nhược điểm của ASP.NET
- 2.2.4. Các thành phần chính của ASP.NET
- **Chương 3: THỰC HIỆN HÓA NGHIÊN CỨU**
- 3.1. Phân tích thiết kế hệ thống
- 3.1.1. Mô tả bài toán
- 3.1.2. Xác định yêu cầu
- 3.1.3. Mô tả chức năng
- 3.1.4. Đề xuất mô hình
- 3.2. Thiết kế cơ sở dữ liệu
- 3.2.1. Xác định các bảng dữ liệu
- 3.2.2. Thiết kế quan hệ giữa các bảng
- **Chương 4: KẾT QUẢ VÀ ĐÁNH GIÁ**
- 4.1. Một số hình ảnh chương trình
- 4.2. Cấu trúc dự án
- **TÀI LIỆU THAM KHẢO**

LỜI CẢM ƠN

Trước tiên, tôi xin gửi lời cảm ơn chân thành đến quý thầy cô trong khoa Công nghệ Phần mềm, Trường Đại học Công nghệ Thông tin đã tận tình giảng dạy và truyền đạt kiến thức quý báu trong suốt quá trình học tập tại trường.

Đặc biệt, tôi xin bày tỏ lòng biết ơn sâu sắc đến TS Đoàn Phước Miền người đã dành thời gian hướng dẫn, chỉ bảo và đưa ra những góp ý quý báu để tôi có thể hoàn thành đồ án tốt nghiệp này.

Tôi cũng xin cảm ơn gia đình, bạn bè đã luôn động viên, ủng hộ và tạo điều kiện thuận lợi để tôi có thể tập trung hoàn thành đồ án.

Mặc dù đã cố gắng hết sức, nhưng do thời gian và kinh nghiệm còn hạn chế, đồ án không tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý, chỉ bảo của quý thầy cô để đồ án được hoàn thiện hơn.

Xin chân thành cảm ơn!

Sinh viên thực hiện



Lê Hồng Hoàng

CHƯƠNG 1: TỔNG QUAN

1.1. Lý do chọn đề tài

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ và nhu cầu tìm kiếm việc làm ngày càng cao, việc xây dựng một website tuyển dụng việc làm trở nên cần thiết và có ý nghĩa thực tiễn cao. Những lý do chính khiến tôi chọn đề tài này bao gồm:

Do nhu cầu thực tế về tuyển dụng trực tuyến đang tăng cao. Theo thống kê, hơn 80% các doanh nghiệp hiện nay sử dụng các nền tảng trực tuyến để tuyển dụng nhân sự. Việc có một hệ thống tuyển dụng hiệu quả sẽ giúp kết nối người tìm việc và nhà tuyển dụng một cách nhanh chóng và tiện lợi.

Công nghệ ASP.NET Core MVC là một framework hiện đại, mạnh mẽ và được sử dụng rộng rãi trong phát triển web. Việc áp dụng công nghệ này vào xây dựng website tuyển dụng sẽ giúp tôi nắm vững kiến thức chuyên môn và có thể ứng dụng vào thực tế công việc sau này.

Về đề tài có tính ứng dụng cao, có thể triển khai thực tế và mang lại giá trị kinh tế. Website tuyển dụng không chỉ phục vụ cho việc học tập mà còn có thể phát triển thành một sản phẩm thương mại.

Tóm lại thông qua việc thực hiện đề tài này, tôi có cơ hội tìm hiểu sâu về quy trình phát triển phần mềm từ phân tích yêu cầu, thiết kế hệ thống, lập trình đến triển khai và bảo trì.

1.2. Mục tiêu nghiên cứu

1.2.1. Mục tiêu chung

Xây dựng một website tuyển dụng việc làm hoàn chỉnh sử dụng công nghệ ASP.NET Core MVC, cung cấp nền tảng kết nối hiệu quả giữa người tìm việc và nhà tuyển dụng.

1.2.2. Mục tiêu cụ thể

- Phân tích và thiết kế hệ thống website tuyển dụng với đầy đủ các chức năng cơ bản.

- Xây dựng cơ sở dữ liệu phù hợp để lưu trữ thông tin người dùng, công việc và đơn ứng tuyển.
- Phát triển giao diện người dùng thân thiện, dễ sử dụng và responsive.
- Triển khai các chức năng chính: đăng ký/đăng nhập, đăng tin tuyển dụng, tìm kiếm việc làm, ứng tuyển.
- Áp dụng mô hình MVC để tổ chức code một cách khoa học và dễ bảo trì.
- Đảm bảo tính bảo mật và hiệu suất của hệ thống.
- Kiểm thử và đánh giá chất lượng sản phẩm.

1.3. Phương pháp nghiên cứu

1.3.1. Phương pháp nghiên cứu lý thuyết

- **Thu thập tài liệu:** Nghiên cứu các tài liệu, sách báo, bài viết khoa học về ASP.NET Core MVC, thiết kế web, cơ sở dữ liệu.
- **Phân tích so sánh:** So sánh các website tuyển dụng hiện có để rút ra những ưu nhược điểm và đưa ra giải pháp phù hợp.
- **Nghiên cứu công nghệ:** Tìm hiểu sâu về các công nghệ sử dụng: ASP.NET Core, Entity Framework, SQLite, HTML/CSS/JavaScript.

1.3.2. Phương pháp nghiên cứu thực nghiệm

- **Phương pháp phát triển phần mềm:** Áp dụng mô hình Waterfall với các giai đoạn: phân tích yêu cầu, thiết kế, lập trình, kiểm thử, triển khai.
- **Lập trình thực tế:** Xây dựng hệ thống từng bước, kiểm thử từng module.
- **Đánh giá và cải tiến:** Thu thập phản hồi và liên tục cải thiện sản phẩm.

1.4. Đối tượng và phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

- **Đối tượng chính:** Website tuyển dụng việc làm được xây dựng bằng ASP.NET Core MVC.
- **Người dùng mục tiêu:**

- Người tìm việc (Candidates): Có thể tìm kiếm và ứng tuyển các vị trí công việc.
- Nhà tuyển dụng (Employers): Có thể đăng tin tuyển dụng và quản lý ứng viên.
- Quản trị viên (Admin): Quản lý toàn bộ hệ thống.

1.4.2. Phạm vi nghiên cứu

Về mặt chức năng:

- Hệ thống đăng ký, đăng nhập và quản lý tài khoản.
- Chức năng đăng tin tuyển dụng cho nhà tuyển dụng.
- Chức năng tìm kiếm và ứng tuyển việc làm cho người tìm việc.
- Hệ thống quản lý hồ sơ cá nhân.
- Chức năng quản trị hệ thống.

Về mặt công nghệ:

- Backend: ASP.NET Core MVC 9.0.
- Database: SQLite với Entity Framework Core.
- Frontend: HTML5, CSS3, JavaScript, Bootstrap.
- Kiến trúc: Model-View-Controller (MVC).

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1. Tổng quan về kiến thức chung

2.1.1. Internet và World Wide Web

Internet là một mạng lưới toàn cầu kết nối hàng tỷ thiết bị máy tính trên khắp thế giới. Internet cho phép các thiết bị giao tiếp với nhau thông qua một tập hợp các giao thức chuẩn, chủ yếu là TCP/IP (Transmission Control Protocol/Internet Protocol).

World Wide Web (WWW) là một hệ thống thông tin được xây dựng trên nền tảng Internet, sử dụng giao thức HTTP/HTTPS để truyền tải dữ liệu. Web bao gồm hàng tỷ trang web được liên kết với nhau thông qua các hyperlink.

Các thành phần chính của Web:

- **Web Server:** Máy chủ lưu trữ và phục vụ các trang web.
- **Web Browser:** Ứng dụng client để truy cập và hiển thị nội dung web.
- **HTTP/HTTPS:** Giao thức truyền tải siêu văn bản.
- **URL:** Địa chỉ định danh tài nguyên trên web.
- **HTML:** Ngôn ngữ đánh dấu để tạo cấu trúc trang web.

Mô hình Client-Server:

Web hoạt động theo mô hình client-server, trong đó:

- Client (trình duyệt) gửi yêu cầu HTTP đến server.
- Server xử lý yêu cầu và trả về phản hồi HTTP.
- Client nhận phản hồi và hiển thị nội dung cho người dùng.

2.1.2. Ngôn ngữ HTML, CSS, JavaScript

2.1.2.1. HTML (*HyperText Markup Language*)

HTML là ngôn ngữ đánh dấu chuẩn để tạo ra các trang web. HTML sử dụng các thẻ (tags) để định nghĩa cấu trúc và nội dung của trang web.

Cấu trúc cơ bản của HTML:

```
<!DOCTYPE html> <html lang="vi"> <head> <meta charset="UTF-8">
<title>Tiêu đề trang</title> </head> <body> <h1>Tiêu đề chính</h1> <p>Đoạn
văn bản</p> </body> </html>
```

Các thẻ HTML quan trọng:

- **<div>:** Container chung.
- **<form>:** Biểu mẫu nhập liệu.
- **<input>:** Các trường nhập liệu.
- **<table>:** Bảng dữ liệu.
- **<a>:** Liên kết.

2.1.2.2. CSS (*Cascading Style Sheets*)

CSS là ngôn ngữ được sử dụng để mô tả cách hiển thị các phần tử HTML. CSS cho phép tách biệt nội dung và cách trình bày, giúp trang web có giao diện đẹp và nhất quán.

Cú pháp CSS:

```
selector { property: value; property: value; } /* Ví dụ */
.container { width: 100%; max-width: 1200px; margin: 0 auto; padding: 20px; }
```

Các loại selector:

- **Element selector:** h1, p, div.
- **Class selector:** .class-name.
- **ID selector:** #id-name.
- **Attribute selector:** [attribute="value"].

2.1.2.3. JavaScript

JavaScript là ngôn ngữ lập trình được sử dụng để tạo ra các trang web tương tác. JavaScript có thể chạy trên trình duyệt (client-side) và trên server (Node.js).

Đặc điểm của JavaScript:

- Ngôn ngữ thông dịch, không cần biên dịch.
- Hỗ trợ lập trình hướng đối tượng và functional programming.
- Có thể thao tác với DOM (Document Object Model).
- Hỗ trợ AJAX để giao tiếp với server.

Ví dụ JavaScript cơ bản:

```
// Khai báo biến let userName = "Nguyễn Văn A"; const API_URL =
"https://api.example.com"; // Hàm function validateForm() { let email =
document.getElementById("email").value; if (!email.includes("@")) { alert("Email
không hợp lệ!"); return false; } return true; } // Event handling
document.addEventListener("DOMContentLoaded", function() {
console.log("Trang web đã tải xong!");});
```

2.1.3. Cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu

2.1.3.1. Khái niệm cơ sở dữ liệu

Cơ sở dữ liệu (Database) là một tập hợp có tổ chức các dữ liệu liên quan được lưu trữ và truy cập điện tử từ hệ thống máy tính. Cơ sở dữ liệu được thiết kế để lưu trữ, quản lý và truy xuất thông tin một cách hiệu quả.

Hệ quản trị cơ sở dữ liệu (DBMS - Database Management System) là phần mềm cho phép người dùng tạo, quản lý và thao tác với cơ sở dữ liệu.

2.1.3.2. Mô hình cơ sở dữ liệu quan hệ

Mô hình cơ sở dữ liệu quan hệ tổ chức dữ liệu thành các bảng (tables) có liên quan với nhau. Mỗi bảng bao gồm:

- **Hàng (Rows/Records):** Đại diện cho một thực thể.
- **Cột (Columns/Fields):** Đại diện cho thuộc tính của thực thể.
- **Khóa chính (Primary Key):** Định danh duy nhất cho mỗi hàng.
- **Khóa ngoại (Foreign Key):** Tạo liên kết giữa các bảng.

2.1.3.3. SQLite

SQLite là một hệ quản trị cơ sở dữ liệu quan hệ nhúng, không cần server. SQLite được sử dụng rộng rãi trong các ứng dụng di động và desktop.

Ưu điểm của SQLite:

- Nhẹ và nhanh.
- Không cần cài đặt server.
- Hỗ trợ đầy đủ SQL.
- Đa nền tảng.
- Miễn phí và mã nguồn mở.

Nhược điểm của SQLite:

- Không hỗ trợ đồng thời cao.
- Giới hạn kích thước cơ sở dữ liệu.
- Không có hệ thống phân quyền phức tạp.

2.1.3.4. SQL (Structured Query Language)

SQL là ngôn ngữ chuẩn để thao tác với cơ sở dữ liệu quan hệ. SQL bao gồm các loại câu lệnh chính:

DDL (Data Definition Language):

```
-- Tạo bảng CREATE TABLE Users ( Id TEXT PRIMARY KEY, Email TEXT NOT NULL UNIQUE, Role INTEGER NOT NULL );
-- Thêm cột ALTER TABLE Users ADD COLUMN CreatedDate DATETIME;
-- Xóa bảng DROP TABLE Users;
```

DML (Data Manipulation Language):

```
-- Thêm dữ liệu INSERT INTO Users (Id, Email, Role) VALUES ('1', 'user@example.com', 1);
-- Cập nhật dữ liệu UPDATE Users SET Email = 'newemail@example.com' WHERE Id = '1';
-- Xóa dữ liệu DELETE FROM Users WHERE Id = '1';
```

DQL (Data Query Language):

```
-- Truy vấn cơ bản SELECT * FROM Users;
-- Truy vấn có điều kiện SELECT Email FROM Users WHERE Role = 1;
-- Truy vấn kết hợp bảng SELECT u.Email, c.Name FROM Users u JOIN Candidates c ON u.ProfileId = c.Id;
```

2.2. Cơ sở lý thuyết chuyên môn

2.2.1. Giới thiệu về ASP.NET

ASP.NET là một framework phát triển web mã nguồn mở được tạo ra bởi Microsoft. ASP.NET cho phép các nhà phát triển xây dựng các ứng dụng web hiện đại, API web và microservices chạy trên Windows, Linux và macOS.

ASP.NET Core là phiên bản mới nhất của ASP.NET, được viết lại hoàn toàn để có hiệu suất cao hơn, đa nền tảng và mã nguồn mở. ASP.NET Core kết hợp ASP.NET MVC và ASP.NET Web API thành một framework thống nhất.

Đặc điểm chính của ASP.NET Core:

- **Đa nền tảng:** Chạy trên Windows, Linux, macOS.
- **Hiệu suất cao:** Tối ưu hóa cho hiệu suất và khả năng mở rộng.

- **Mã nguồn mở:** Hoàn toàn miễn phí và mã nguồn mở.
- **Modular:** Chỉ bao gồm các thành phần cần thiết.
- **Dependency Injection:** Hỗ trợ DI tích hợp sẵn.
- **Cloud-ready:** Tối ưu cho triển khai cloud.

Kiến trúc ASP.NET Core:

Sơ đồ kiến trúc ASP.NET Core:

Có thể tạo sơ đồ tại: [@\] \[Routing\] --> \[Controllers\] \[Controllers\] --> \[Modelsstartuml package](https://editor.plantuml.com/uml)

```
"ASP.NET Core Application" { [Middleware Pipeline] [Routing] [Controllers]
[Views] [Models] [Services] } package "Runtime" { [.NET Runtime] [Kestrel Web
Server] } [Browser] --> [Middleware Pipeline] [Middleware Pipeline] --> [Routing]
[Controllers] --> [Views] [Controllers] --> [Services] @enduml
```

2.2.2. Lịch sử phát triển và các phiên bản của ASP.NET

Phiên bản	Năm phát hành	Đặc điểm chính
ASP.NET 1.0	2002	Phiên bản đầu tiên, Web Forms
ASP.NET 2.0	2005	Master Pages, Membership, Profile
ASP.NET 3.5	2007	LINQ, AJAX Extensions
ASP.NET 4.0	2010	Dynamic Data, Chart Controls
ASP.NET MVC	2009	Mô hình MVC, RESTful URLs
ASP.NET Core 1.0	2016	Đa nền tảng, mã nguồn mở
ASP.NET Core 2.0	2017	Razor Pages, .NET Standard 2.0

ASP.NET Core 3.0	2019	Blazor, gRPC, Worker Services
ASP.NET Core 5.0	2020	Thống nhất .NET Framework và .NET Core
ASP.NET Core 6.0	2021	Minimal APIs, Hot Reload
ASP.NET Core 7.0	2022	Cải thiện hiệu suất, Rate Limiting
ASP.NET Core 8.0	2023	Native AOT, Blazor improvements
ASP.NET Core 9.0	2024	Phiên bản hiện tại được sử dụng trong dự án

2.2.3. Ưu điểm và nhược điểm của ASP.NET

2.2.3.1. Ưu điểm

- **Hiệu suất cao:** ASP.NET Core được tối ưu hóa để có hiệu suất vượt trội so với nhiều framework khác.
- **Đa nền tảng:** Có thể chạy trên Windows, Linux, macOS.
- **Mã nguồn mở:** Miễn phí và có cộng đồng phát triển mạnh.
- **Bảo mật:** Có nhiều tính năng bảo mật tích hợp sẵn.
- **Scalability:** Dễ dàng mở rộng theo chiều ngang và chiều dọc.
- **Tooling:** Có Visual Studio - IDE mạnh mẽ.
- **Ecosystem:** Có hệ sinh thái phong phú với NuGet packages.
- **Type Safety:** Sử dụng C# - ngôn ngữ strongly typed.
- **Dependency Injection:** Hỗ trợ DI tích hợp sẵn.
- **Testing:** Dễ dàng viết unit test và integration test.

2.2.3.2. Nhược điểm

- **Learning curve:** Cần thời gian để học và nắm vững.

- **Microsoft ecosystem:** Phụ thuộc vào hệ sinh thái Microsoft.
- **Memory usage:** Tiêu thụ nhiều bộ nhớ hơn một số framework khác.
- **Licensing:** Một số công cụ Microsoft có phí.
- **Rapid changes:** Phát triển nhanh, cần cập nhật thường xuyên.

2.2.4. Các thành phần chính của ASP.NET: MVC, Web API, Web Forms, Razor Pages

2.2.4.1. ASP.NET MVC (*Model-View-Controller*)

MVC là một mô hình kiến trúc phần mềm tách biệt ứng dụng thành ba thành phần chính:

Model:

- Đại diện cho dữ liệu và business logic.
- Chứa các class định nghĩa cấu trúc dữ liệu.
- Xử lý validation và business rules.

```
// Ví dụ Model
public class Job {
    public string Id { get; set; } [Required]
    public string Title { get; set; } [Required]
    public string Company { get; set; }
    public string Location { get; set; }
    public JobType Type { get; set; }
}
```

View:

- Chịu trách nhiệm hiển thị giao diện người dùng.
- Sử dụng Razor syntax để kết hợp HTML và C#.
- Nhận dữ liệu từ Controller để hiển thị.

```
@model List<Job>
<h1>Danh sách việc làm</h1>
@foreach(var job in Model) {
    <div class="job-item">
        <h3>@job.Title</h3>
        <p>@job.Company - @job.Location</p>
    </div>
}
```

Controller:

- Xử lý các request từ người dùng.
- Tương tác với Model để lấy/xử lý dữ liệu.
- Chọn View phù hợp để trả về response.

```
public class JobsController : Controller { private readonly IDataService
    _dataService; public JobsController(IDataService dataService) { _dataService =
    dataService; } public async Task<IActionResult> Index() { var jobs = await
    _dataService.GetJobsAsync(); return View(jobs); } }
```

Ưu điểm của MVC:

- Tách biệt rõ ràng các thành phần.
- Dễ dàng test và maintain.
- Hỗ trợ parallel development.
- SEO friendly URLs.
- Full control over HTML output.

2.2.4.2. Web API:

ASP.NET Web API là framework để xây dựng RESTful services. Web API có thể trả về dữ liệu dưới nhiều định dạng như JSON, XML.

```
[ApiController] [Route("api/[controller]")] public class JobsApiController :
    ControllerBase { [HttpGet] public async Task<ActionResult<List<Job>>>
    GetJobs() { var jobs = await _dataService.GetJobsAsync(); return Ok(jobs); }
    [HttpPost] public async Task<ActionResult<Job>> CreateJob(Job job) { var
    createdJob = await _dataService.CreateJobAsync(job); return
    CreatedAtAction(nameof(GetJob), new { id = createdJob.Id }, createdJob); } }
```

2.2.4.3. Web Forms:

Web Forms là mô hình phát triển web truyền thống của ASP.NET, sử dụng event-driven programming model tương tự Windows Forms.

Đặc điểm:

- Page lifecycle events.
- Server controls.
- ViewState.
- Postback model.

2.2.4.4. Razor Pages:

Razor Pages là mô hình page-based được giới thiệu trong ASP.NET Core 2.0, đơn giản hóa việc xây dựng web applications.

```
// Jobs.cshtml.cs (Page Model) public class JobsModel : PageModel { private
readonly IDataService _dataService; public List<Job> Jobs { get; set; } public
async Task OnGetAsync() { Jobs = await _dataService.GetJobsAsync(); } }

@page @model JobsModel <h1>Danh sách việc làm</h1> @foreach(var job in
Model.Jobs) { <div>@job.Title</div> }
```

So sánh các mô hình:

Đặc điểm	MVC	Web API	Razor Pages
Mục đích	Web applications	RESTful services	Page-based apps
Kiến trúc	MVC pattern	RESTful	Page-based
Phù hợp	Ứng dụng phức tạp	API services	Ứng dụng đơn giản
Learning curve	Trung bình	Dễ	Dễ

CHƯƠNG 3: THỰC HIỆN HÓA NGHIÊN CỨU

3.1. Phân tích thiết kế hệ thống

3.1.1. Mô tả bài toán

Website tuyển dụng việc làm là một hệ thống trực tuyến kết nối người tìm việc và nhà tuyển dụng. Hệ thống cần giải quyết các vấn đề sau:

Vấn đề của người tìm việc:

- Khó khăn trong việc tìm kiếm thông tin việc làm phù hợp.
- Thiếu kênh thông tin tập trung về các cơ hội việc làm.
- Quy trình ứng tuyển phức tạp và mất thời gian.

- Không có công cụ để quản lý hồ sơ cá nhân và lịch sử ứng tuyển.

Vấn đề của nhà tuyển dụng:

- Chi phí cao cho việc đăng tin tuyển dụng trên nhiều kênh.
- Khó khăn trong việc tiếp cận ứng viên phù hợp.
- Quy trình quản lý ứng viên thủ công, không hiệu quả.
- Thiếu công cụ để đánh giá và so sánh ứng viên.

Giải pháp đề xuất:

Xây dựng một website tuyển dụng việc làm với các tính năng chính:

- Hệ thống đăng ký/đăng nhập cho cả người tìm việc và nhà tuyển dụng.
- Chức năng đăng tin tuyển dụng cho nhà tuyển dụng.
- Công cụ tìm kiếm việc làm thông minh cho người tìm việc.
- Hệ thống ứng tuyển trực tuyến.
- Quản lý hồ sơ cá nhân và công ty.
- Hệ thống quản trị cho admin.

3.1.2. Xác định yêu cầu

3.1.2.1. Yêu cầu chức năng

Đối với người tìm việc (Candidate):

- Đăng ký/Đăng nhập:** Tạo tài khoản và đăng nhập vào hệ thống.
- Quản lý hồ sơ:** Cập nhật thông tin cá nhân, kỹ năng, kinh nghiệm..
- Tìm kiếm việc làm:** Tìm kiếm theo từ khóa, địa điểm, ngành nghề.
- Xem chi tiết công việc:** Xem mô tả công việc, yêu cầu, quyền lợi.
- Ứng tuyển:** Gửi đơn ứng tuyển cho các vị trí phù hợp.
- Theo dõi đơn ứng tuyển:** Xem trạng thái các đơn đã gửi.

Đối với nhà tuyển dụng (Employer):

- Đăng ký/Đăng nhập:** Tạo tài khoản công ty và đăng nhập.
- Quản lý thông tin công ty:** Cập nhật thông tin, logo, mô tả công ty.

- **Đăng tin tuyển dụng:** Tạo và đăng các tin tuyển dụng.
- **Quản lý tin đăng:** Chính sửa, xóa, ẩn/hiện tin tuyển dụng.
- **Xem ứng viên:** Xem danh sách ứng viên đã ứng tuyển.
- **Quản lý ứng tuyển:** Duyệt, từ chối đơn ứng tuyển.

Đối với quản trị viên (Admin):

- **Quản lý người dùng:** Xem, khóa/mở khóa tài khoản.
- **Quản lý tin tuyển dụng:** Duyệt, xóa tin không phù hợp.
- **Quản lý danh mục:** Thêm, sửa, xóa các danh mục nghề nghiệp.
- **Thống kê hệ thống:** Xem báo cáo về hoạt động hệ thống.

3.1.2.2. Yêu cầu phi chức năng

Hiệu suất:

- Thời gian tải trang không quá 3 giây.
- Hỗ trợ đồng thời tối thiểu 100 người dùng.
- Thời gian phản hồi API không quá 1 giây.

Bảo mật:

- Mã hóa mật khẩu người dùng.
- Xác thực và phân quyền người dùng.
- Bảo vệ chống SQL Injection, XSS.
- Sử dụng HTTPS cho tất cả giao tiếp.

Khả năng sử dụng:

- Giao diện thân thiện, dễ sử dụng.
- Responsive design cho mobile và desktop.
- Hỗ trợ các trình duyệt phổ biến.
- Thông báo lỗi rõ ràng, dễ hiểu.

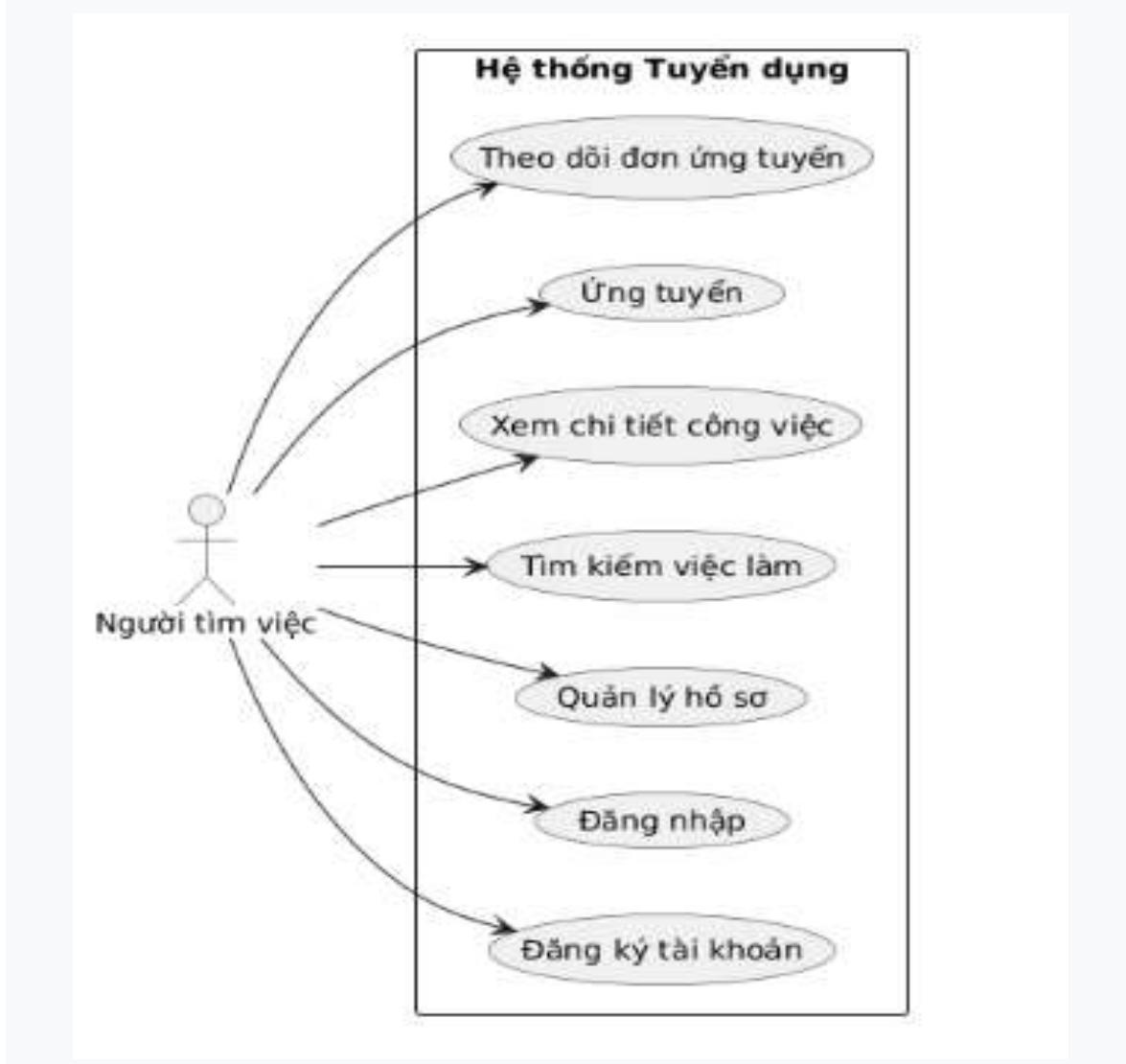
Khả năng mở rộng:

- Kiến trúc modular, dễ bảo trì.

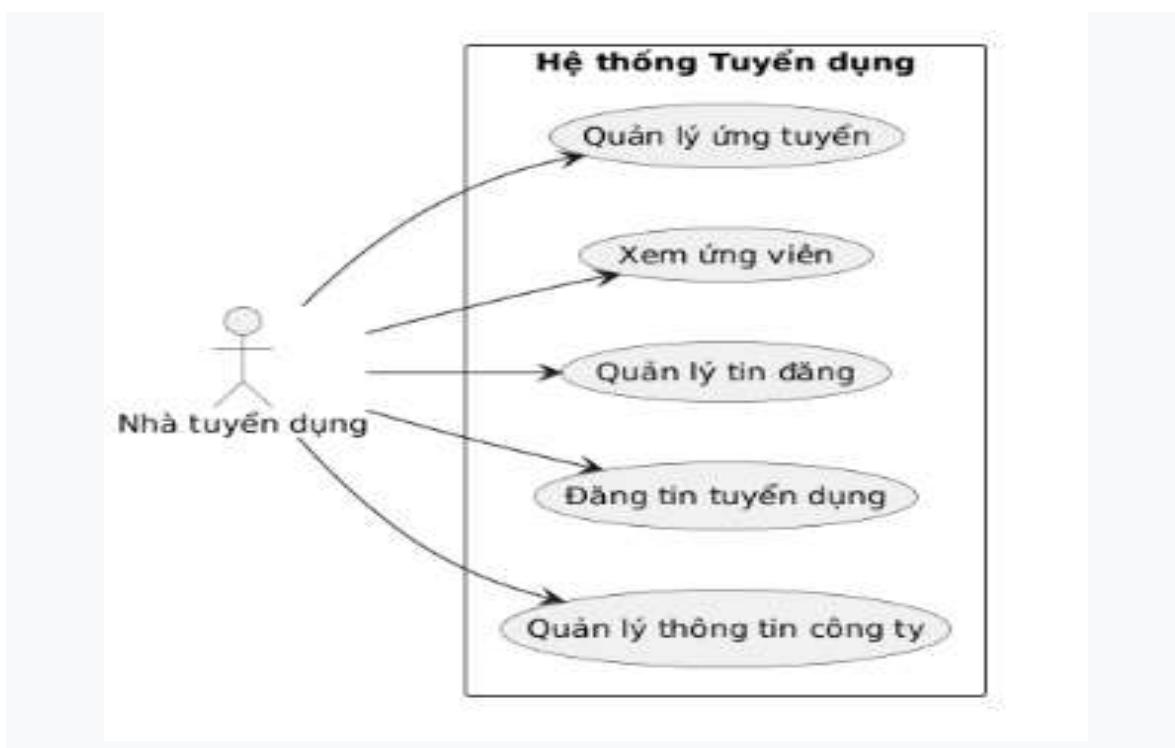
- Có thể thêm tính năng mới.
- Hỗ trợ tăng số lượng người dùng.

Tính khả dụng:

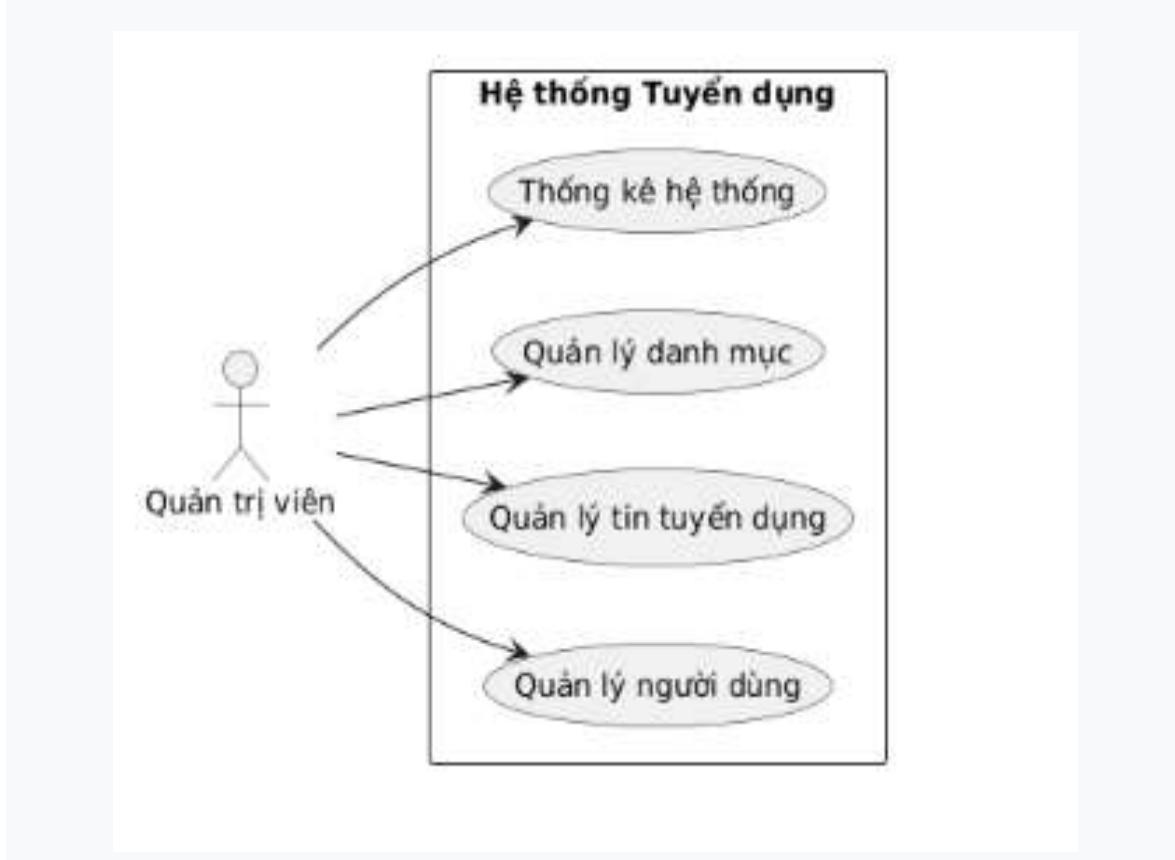
- Uptime tối thiểu 99%.
- Có cơ chế backup và recovery.
- Xử lý lỗi gracefully.

3.1.3. Mô tả chức năng**3.1.3.1. Sơ đồ Use Case****Sơ đồ Use Case cho người tìm việc**

Sơ đồ Use Case cho nhà tuyển dụng



Sơ đồ Use Case cho Website quản trị viên website



3.1.3.2. Mô tả chi tiết các Use Case chính

UC4: Tìm kiếm việc làm:

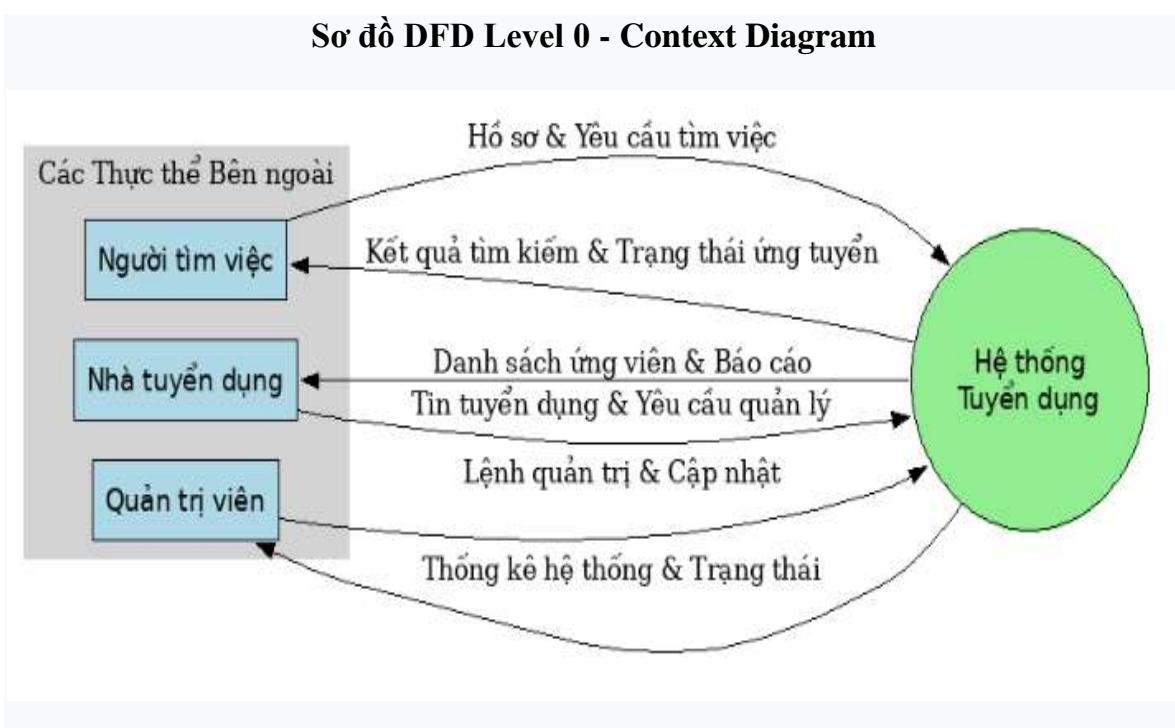
Tên Use Case	Tìm kiếm việc làm
Actor	Người tìm việc
Mô tả	Người dùng tìm kiếm các công việc phù hợp theo tiêu chí
Điều kiện tiên quyết	Không yêu cầu đăng nhập
Luồng chính	1. Người dùng truy cập trang tìm kiếm 2. Nhập từ khóa, chọn địa điểm, danh mục 3. Nhấn nút "Tìm kiếm" 4. Hệ thống hiển thị kết quả phù hợp 5. Người dùng có thể lọc thêm kết quả
Luồng thay thế	4a. Không có kết quả phù hợp - Hiển thị thông báo "Không tìm thấy kết quả" - Đề xuất các từ khóa tương tự

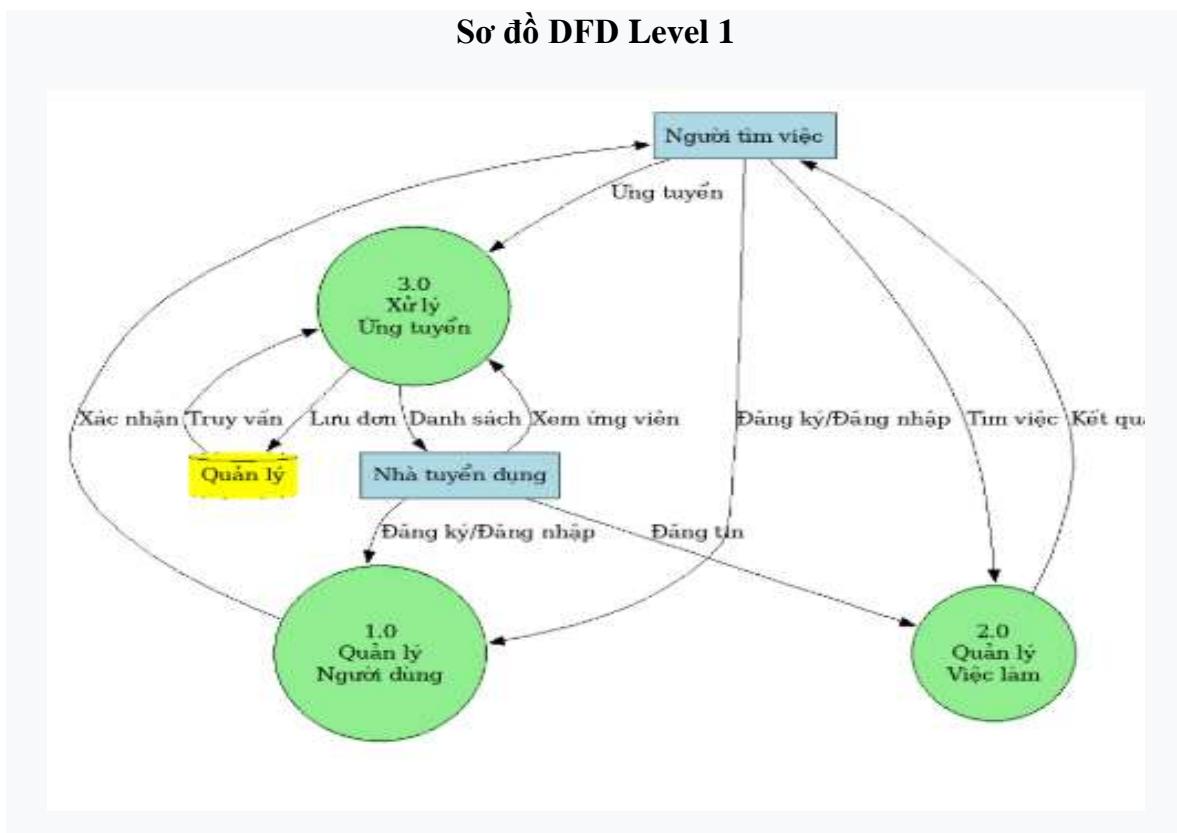
UC6: Úng tuyển:

Tên Use Case	Úng tuyển
Actor	Người tìm việc
Mô tả	Người dùng gửi đơn ứng tuyển cho một vị trí công việc
Điều kiện tiên quyết	Đã đăng nhập với vai trò Candidate
Luồng chính	1. Người dùng xem chi tiết công việc 2. Nhấn nút "Úng tuyển" 3. Điền thông tin ứng tuyển (thư xin việc, CV) 4. Nhấn "Gửi đơn ứng tuyển" 5. Hệ thống lưu đơn ứng tuyển 6. Hiển thị thông báo thành công
Luồng thay thế	1a. Chưa đăng nhập - Chuyển hướng đến trang đăng nhập 3a. Thông tin không hợp lệ - Hiển thị thông báo lỗi - Yêu cầu nhập lại

UC9: Đăng tin tuyển dụng:

Tên Use Case	Đăng tin tuyển dụng
Actor	Nhà tuyển dụng
Mô tả	Nhà tuyển dụng tạo và đăng tin tuyển dụng mới
Điều kiện tiên quyết	Đã đăng nhập với vai trò Employer
Luồng chính	<ol style="list-style-type: none"> 1. Nhà tuyển dụng truy cập trang đăng tin 2. Điện thông tin công việc (tiêu đề, mô tả, yêu cầu, lương) 3. Chọn danh mục và loại công việc 4. Nhấn "Đăng tin" 5. Hệ thống validate thông tin 6. Lưu tin tuyển dụng 7. Hiển thị thông báo thành công
Luồng thay thế	<p>5a. Thông tin không hợp lệ</p> <ul style="list-style-type: none"> - Hiển thị thông báo lỗi - Yêu cầu sửa thông tin

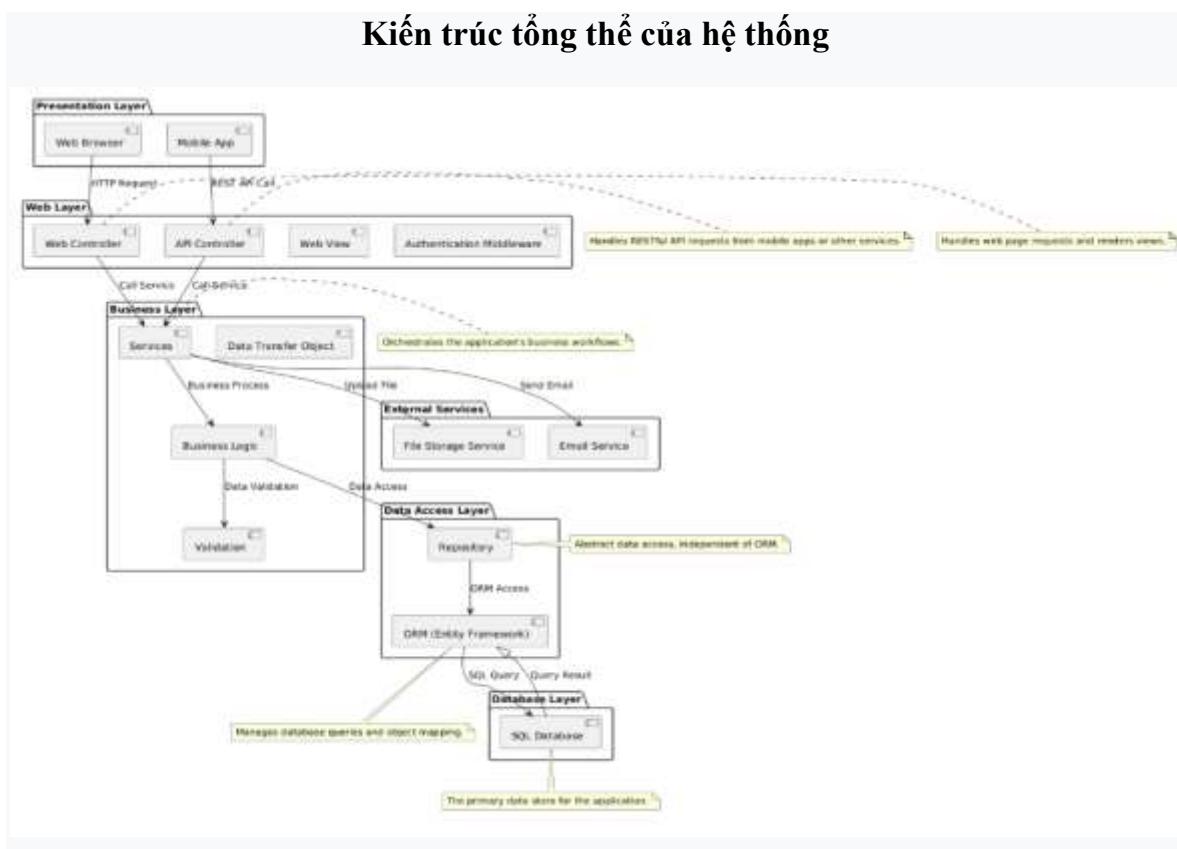
3.1.4. Đề xuất mô hình (ERD, Use Case,...)**3.1.4.1. Sơ đồ luồng dữ liệu (DFD)**



3.1.4.2. Sơ đồ quan hệ thực thể (ERD)



3.1.4.3. Sơ đồ kiến trúc hệ thống



3.2. Thiết kế cơ sở dữ liệu

3.2.1. Xác định các bảng dữ liệu

Dựa trên phân tích yêu cầu và sơ đồ ERD, hệ thống cần các bảng dữ liệu sau:

3.2.1.1. Bảng Users

Mục đích: Lưu trữ thông tin đăng nhập và phân quyền người dùng

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất
Email	VARCHAR(255)	NOT NULL, UNIQUE	Email đăng nhập
Role	INTEGER	NOT NULL	Vai trò (0: Candidate, 1:

			Employer, 2: Admin)
ProfileId	VARCHAR(50)	NULL	ID liên kết đến bảng profile tương ứng
CreatedDate	DATETIME	DEFAULT NOW()	Ngày tạo tài khoản

3.2.1.2. Bảng Candidates

Mục đích: Lưu trữ thông tin hồ sơ của người tìm việc

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất
Name	VARCHAR(255)	NOT NULL	Họ và tên
Email	VARCHAR(255)	NOT NULL	Email liên hệ
Phone	VARCHAR(20)	NULL	Số điện thoại
Avatar	VARCHAR(500)	NULL	Đường dẫn ảnh đại diện
Title	VARCHAR(255)	NULL	Chức danh mong muốn
Experience	TEXT	NULL	Kinh nghiệm làm việc
Skills	TEXT	NULL	Kỹ năng (JSON array)
Education	TEXT	NULL	Học vấn
Resume	VARCHAR(500)	NULL	Đường dẫn file CV

3.2.1.3. *Bảng Employers*

Mục đích: Lưu trữ thông tin công ty/nhà tuyển dụng

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất
Name	VARCHAR(255)	NOT NULL	Tên công ty
Logo	VARCHAR(500)	NULL	Đường dẫn logo công ty
Description	TEXT	NULL	Mô tả công ty
Website	VARCHAR(255)	NULL	Website công ty
Email	VARCHAR(255)	NOT NULL	Email liên hệ
Phone	VARCHAR(20)	NULL	Số điện thoại
Address	TEXT	NULL	Địa chỉ công ty
Industry	VARCHAR(100)	NULL	Ngành nghề
Size	VARCHAR(50)	NULL	Quy mô công ty
Jobs	TEXT	NULL	Danh sách ID công việc (JSON)

3.2.1.4. Bảng Jobs

Mục đích: Lưu trữ thông tin các tin tuyển dụng

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất
Title	VARCHAR(255)	NOT NULL	Tiêu đề công việc
Company	VARCHAR(255)	NOT NULL	Tên công ty
Logo	VARCHAR(500)	NULL	Logo công ty
Location	VARCHAR(255)	NOT NULL	Địa điểm làm việc
Salary	VARCHAR(100)	NOT NULL	Mức lương
Type	INTEGER	NOT NULL	Loại công việc (0-3)
Category	VARCHAR(100)	NOT NULL	Danh mục nghề nghiệp
Description	TEXT	NOT NULL	Mô tả công việc
Requirements	TEXT	NULL	Yêu cầu (JSON array)
Benefits	TEXT	NULL	Quyền lợi (JSON array)
PostedDate	DATETIME	DEFAULT NOW()	Ngày đăng tin
EmployerId	VARCHAR(50)	NOT NULL, FK	ID nhà tuyển dụng
IsActive	BOOLEAN	DEFAULT TRUE	Trạng thái hoạt động

3.2.1.5. *Bảng Applications*

Mục đích: Lưu trữ thông tin đơn ứng tuyển

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất
JobId	VARCHAR(50)	NOT NULL, FK	ID công việc
CandidateId	VARCHAR(50)	NOT NULL, FK	ID ứng viên
CoverLetter	TEXT	NULL	Thư xin việc
Resume	VARCHAR(500)	NULL	Đường dẫn CV
Status	INTEGER	DEFAULT 0	Trạng thái (0-3)
AppliedDate	DATETIME	DEFAULT NOW()	Ngày ứng tuyển

3.2.1.6. *Bảng Categories*

Mục đích: Lưu trữ danh mục nghề nghiệp

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	VARCHAR(50)	PRIMARY KEY	Mã định danh duy nhất
Name	VARCHAR(255)	NOT NULL	Tên danh mục
Description	TEXT	NULL	Mô tả danh mục
IsActive	BOOLEAN	DEFAULT TRUE	Trạng thái hoạt động

3.2.2. Thiết kế quan hệ giữa các bảng

3.2.2.1. Các mối quan hệ chính

1. Users - Candidates (1:1)

- Một user có thể có một profile candidate

- Khóa ngoại: Users.ProfileId → Candidates.Id
- Điều kiện: Users.Role = 0 (Candidate)

2. Users - Employers (1:1)

- Một user có thể có một profile employer
- Khóa ngoại: Users.ProfileId → Employers.Id
- Điều kiện: Users.Role = 1 (Employer)

3. Employers - Jobs (1:N)

- Một employer có thể đăng nhiều job
- Khóa ngoại: Jobs.EmployerId → Employers.Id
- Ràng buộc: NOT NULL, ON DELETE CASCADE

4. Jobs - Applications (1:N)

- Một job có thể có nhiều application
- Khóa ngoại: Applications.JobId → Jobs.Id
- Ràng buộc: NOT NULL, ON DELETE CASCADE

5. Candidates - Applications (1:N)

- Một candidate có thể có nhiều application
- Khóa ngoại: Applications.CandidateId → Candidates.Id
- Ràng buộc: NOT NULL, ON DELETE CASCADE

3.2.2.2. Ràng buộc toàn vẹn dữ liệu

Ràng buộc thực thể:

- Tất cả bảng đều có khóa chính (Primary Key)
- Email trong bảng Users phải duy nhất
- Các trường bắt buộc không được NULL

Ràng buộc tham chiếu:

- Jobs.EmployerId phải tồn tại trong Employers.Id
- Applications.JobId phải tồn tại trong Jobs.Id

- Applications.CandidateId phải tồn tại trong Candidates.Id
- Users.ProfileId phải tồn tại trong Candidates.Id hoặc Employers.Id

Ràng buộc miền giá trị:

- Users.Role $\in \{0, 1, 2\}$
- Jobs.Type $\in \{0, 1, 2, 3\}$
- Applications.Status $\in \{0, 1, 2, 3\}$
- Email phải có định dạng hợp lệ

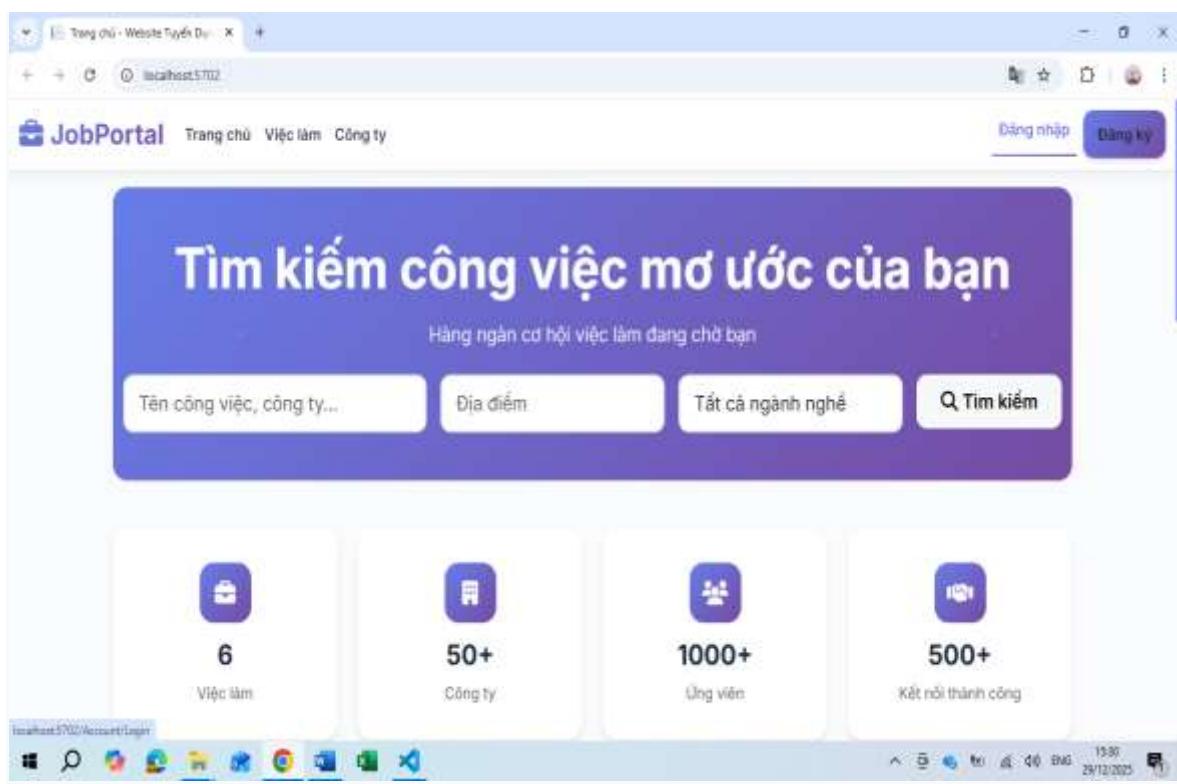
CHƯƠNG 4: KẾT QUẢ VÀ ĐÁNH GIÁ

4.1. Một số hình ảnh chương trình (Mô tả kết quả từng chức năng)

4.1.1. Giao diện trang chủ

Trang chủ website tuyển dụng

- Header với menu điều hướng và nút đăng nhập/đăng ký
- Banner tìm kiếm việc làm với các trường: từ khóa, địa điểm, danh mục
- Danh sách các công việc nổi bật (6 công việc mới nhất)
- Footer với thông tin liên hệ



4.1.2. Chức năng đăng ký/đăng nhập

Form đăng ký tài khoản

- Email (bắt buộc, định dạng email)
- Mật khẩu (bắt buộc, tối thiểu 6 ký tự)
- Xác nhận mật khẩu
- Vai trò: Người tìm việc / Nhà tuyển dụng
- Họ tên (bắt buộc)

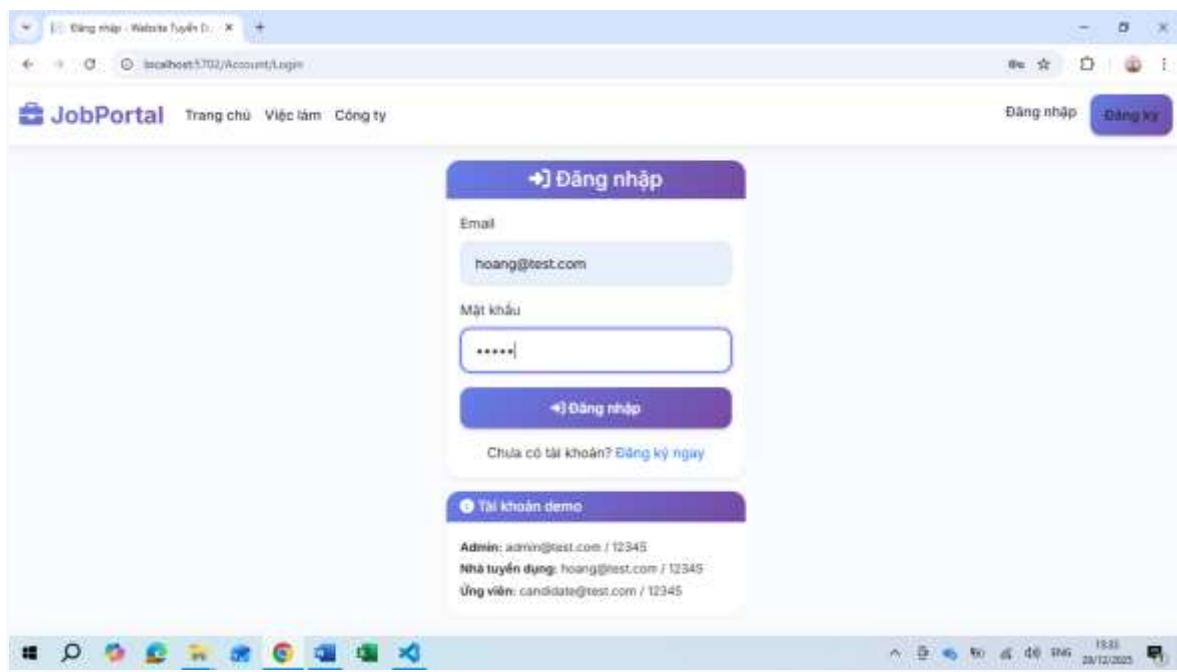
The screenshot shows a registration form titled "Đăng ký tài khoản". The form fields include:

- Họ và tên * (Nhập họ và tên)
- Email * (Nhập email)
- Mật khẩu * (Nhập mật khẩu)
- Xác nhận mật khẩu * (Nhập lại mật khẩu)
- Số điện thoại (Nhập số điện thoại)
- Loại tài khoản * (Chọn loại tài khoản)

Below the form is a checkbox for accepting terms and conditions: "Tôi đồng ý với [Điều khoản sử dụng](#) và [Chính sách bảo mật](#)". At the bottom are "Đăng ký" and "Đã có tài khoản? Đăng nhập ngay!" buttons.

Form đăng nhập

- Trường Email
- Trường Mật khẩu
- Nút "Đăng nhập"
- Link "Đăng ký tài khoản mới"



4.1.3. Chức năng tìm kiếm việc làm

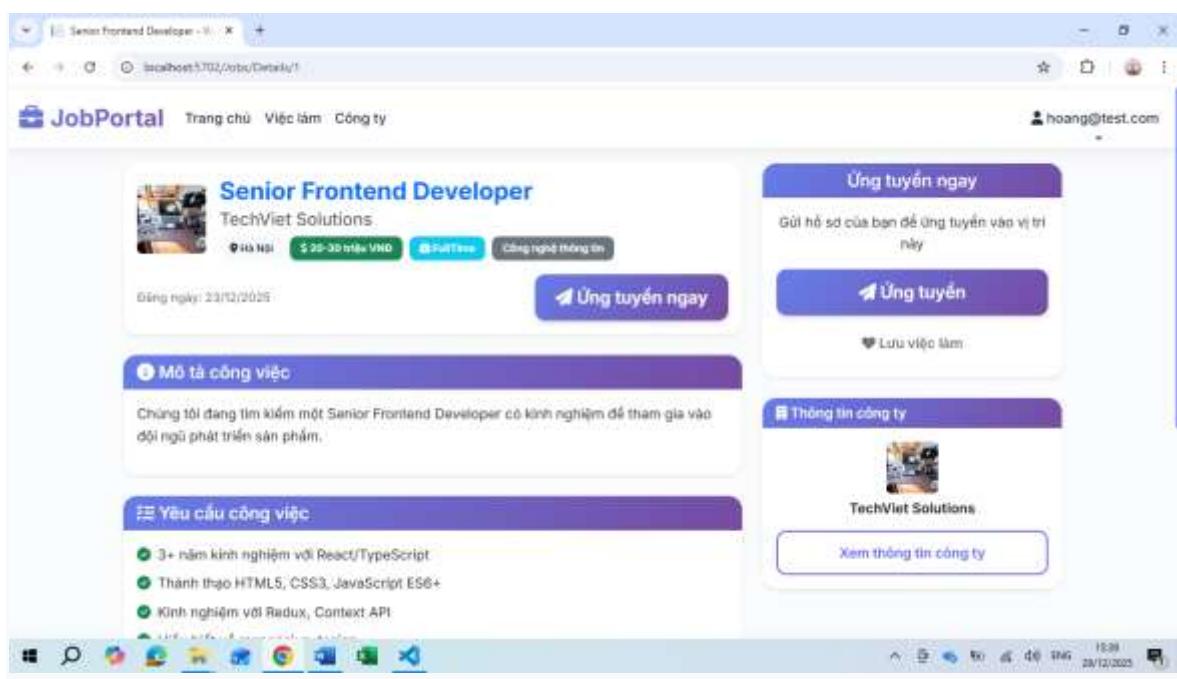
Trang tìm kiếm việc làm

- Form tìm kiếm với các bộ lọc: từ khóa, địa điểm, danh mục.
- Danh sách kết quả tìm kiếm với thông tin: tiêu đề, công ty, địa điểm, lương, loại công việc.
- Phân trang kết quả.
- Nút "Xem chi tiết" cho mỗi công việc.

4.1.4. Chức năng xem chi tiết công việc

Trang chi tiết công việc

- Tiêu đề công việc và logo công ty
- Thông tin cơ bản: lương, địa điểm, loại công việc
- Mô tả công việc chi tiết
- Yêu cầu ứng viên
- Quyền lợi được hưởng
- Nút "Ứng tuyển ngay" (nếu đã đăng nhập)
- Thông tin công ty



4.1.5. Chức năng ứng tuyển

Form ứng tuyển công việc

- Thông tin công việc (chỉ đọc): tiêu đề, công ty
- Họ tên ứng viên (tự động điền nếu đã có profile)
- Email liên hệ
- Số điện thoại
- Thư xin việc (cover letter)
- Upload CV (tùy chọn)

- Nút "Gửi đơn ứng tuyển"

The screenshot shows a web browser window with the URL `localhost:5702/jobportal/apply/1`. The page title is "JobPortal Trang chủ Việc làm Công ty". A purple header bar says "Ứng tuyển vào vị trí Senior Frontend Developer tại TechViet Solutions". The main form has fields for "Họ và tên *", "Email *", "Nhập họ và tên" (input field containing "hoang@test.com"), "Số điện thoại *", "Nhập số điện thoại" (input field), "CV/Resume", "Chọn tệp Không có tệp nào được chọn", "Chấp nhận file PDF, DOC, DOCX (tối đa 5MB)", and "Thư xin việc *". Below these is a text area with placeholder text: "Viết thư xin việc của bạn. Hãy giới thiệu bản thân, kinh nghiệm và lý do bạn phù hợp với vị trí này...". The browser status bar at the bottom shows the date as 29/12/2025.

4.1.6. Chức năng quản lý công ty (Employer)

Trang quản lý thông tin công ty

- Cập nhật thông tin công ty: tên, logo, mô tả, website.
- Thông tin liên hệ: email, điện thoại, địa chỉ.
- Thông tin về ngành nghề và quy mô.
- Xem danh sách các tin tuyển dụng đã đăng.

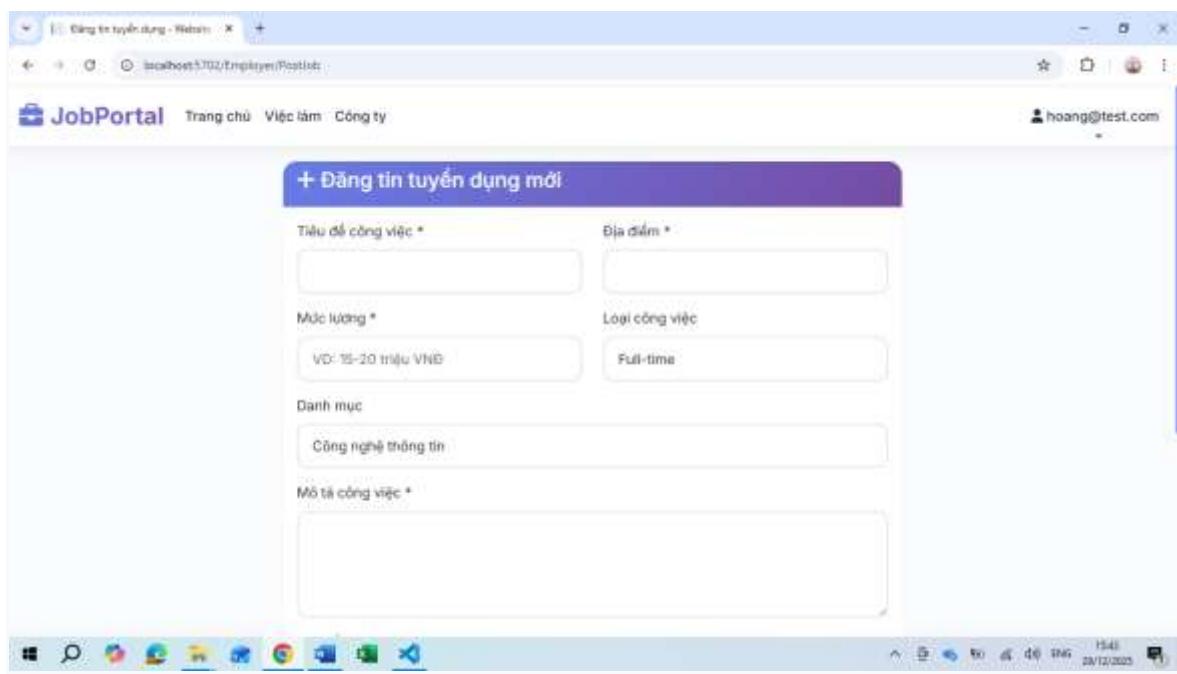
The screenshot shows a web browser window with the URL `localhost:5702/Employer/Dashboard`. The page title is "JobPortal Trang chủ Việc làm Công ty". A sidebar menu on the left shows "Dashboard" and "Đăng tin tuyển dụng". The main area is titled "Dashboard Nhà Tuyển Dụng" and shows three stats in colored boxes: "Tổng tin đăng" (6), "Tin đang hoạt động" (6), and "Ứng viên mới" (15). Below this is a section titled "Tin tuyển dụng của bạn" with a "Đăng tin mới" button. A table lists two job posts:

Tiêu đề	Địa điểm	Lương	Ngày đăng	Trạng thái	Thao tác
Senior Frontend Developer	Hà Nội	20-30 triệu VNĐ	23/12/2025	Hoạt động	<button>Xem</button>
Backend Developer (Node.js)	Hà Nội	18-25 triệu VNĐ	22/12/2025	Hoạt động	<button>Xem</button>

4.1.7. Chức năng đăng tin tuyển dụng

Form đăng tin tuyển dụng

- Tiêu đề công việc (bắt buộc)
- Danh mục nghề nghiệp
- Địa điểm làm việc (bắt buộc)
- Mức lương (bắt buộc)
- Loại công việc: Full-time, Part-time, Contract, Remote
- Mô tả công việc chi tiết
- Yêu cầu ứng viên (danh sách)
- Quyền lợi được hưởng (danh sách)

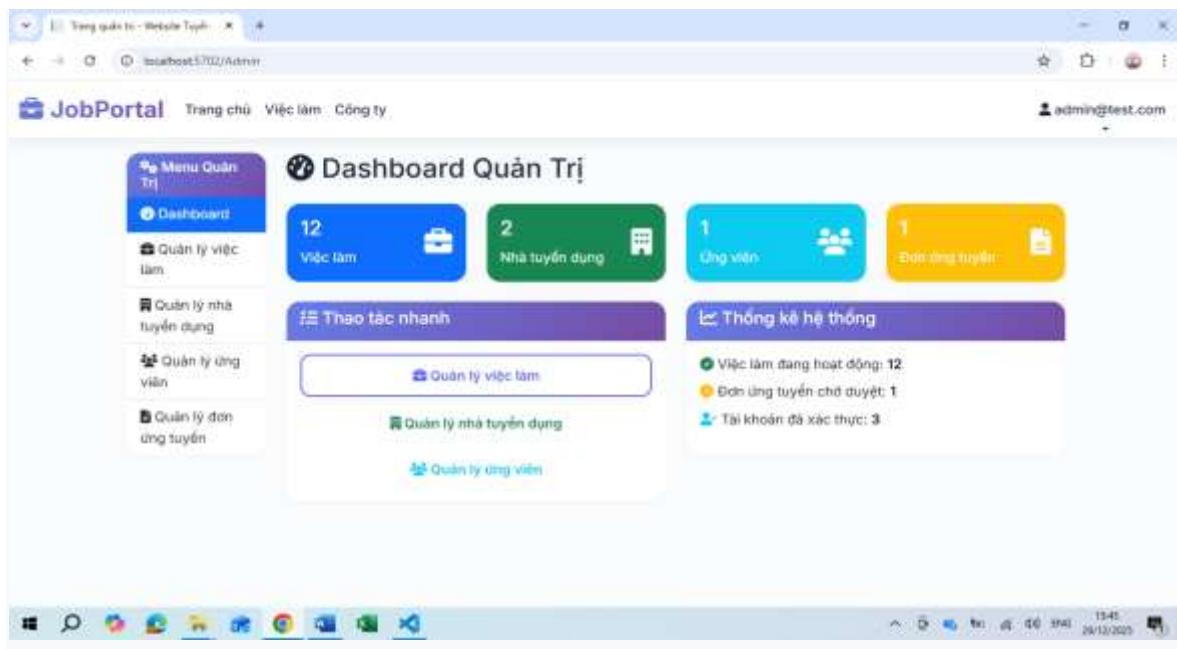


4.1.8. Chức năng quản trị hệ thống (Admin)

Dashboard quản trị

- Thống kê tổng quan: số người dùng, số tin tuyển dụng, số đơn ứng tuyển
- Quản lý người dùng: xem danh sách, khóa/mở khóa tài khoản
- Quản lý tin tuyển dụng: duyệt, xóa tin không phù hợp
- Quản lý ứng viên: xem thông tin, thống kê

- Quản lý nhà tuyển dụng: xem thông tin công ty



4.1.9. Trang Quản lý Việc làm

Chức năng chính:

- **Danh sách việc làm:** Hiển thị tất cả công việc trong hệ thống
- **Thông tin hiển thị:**
 - Logo và tên công việc
 - Tên công ty
 - Địa điểm làm việc
 - Mức lương
 - Loại công việc (Full-time, Part-time, Contract, Remote)
 - Ngày đăng tuyển
 - Trạng thái (Đang hoạt động/Đã đóng)
- **Thao tác có thể thực hiện:**
 - Xem chi tiết công việc
 - Chỉnh sửa thông tin
 - Tạm dừng/Kích hoạt công việc
 - Xóa công việc

- Thêm việc làm mới

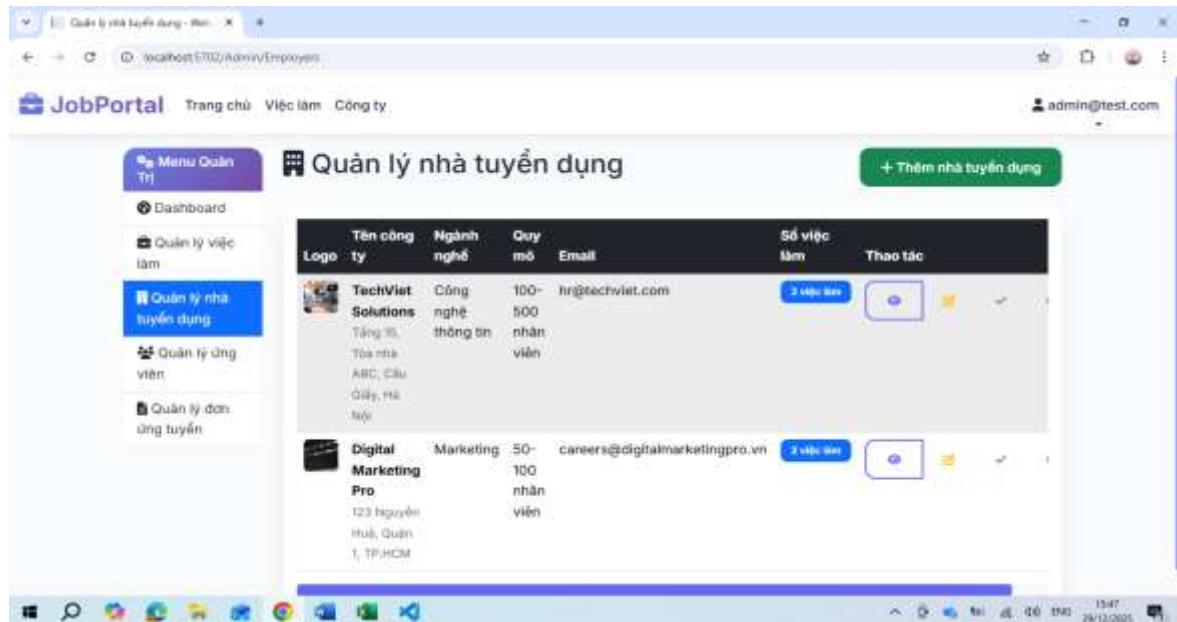
Công việc	Công ty	Địa điểm	Lương	Loại	Ngày đăng	Trạng thái	Thao tác
Senior Frontend Developer	TechViet Solutions	Hà Nội	20-30 triệu VNĐ	FullTime	23/12/2025	Đang hoạt động	
Backend Developer (Node.js)	TechViet Solutions	Hà Nội	18-25 triệu VNĐ	FullTime	22/12/2025	Đang hoạt động	
Mobile App Developer (Flutter)	TechViet Solutions	Hà Nội	16-22 triệu VNĐ	FullTime	21/12/2025	Đang hoạt động	

4.1.10. Trang Quản lý Nhà tuyển dụng

Chức năng chính:

- **Danh sách nhà tuyển dụng:** Hiển thị tất cả công ty đã đăng ký
- **Thông tin hiển thị:**
 - Logo công ty
 - Tên công ty và địa chỉ
 - Ngành nghề hoạt động
 - Quy mô công ty
 - Email liên hệ
 - Số lượng việc làm đã đăng
- **Thao tác có thể thực hiện:**
 - Xem chi tiết công ty
 - Chỉnh sửa thông tin
 - Phê duyệt tài khoản
 - Khóa tài khoản

- o Thêm nhà tuyển dụng mới

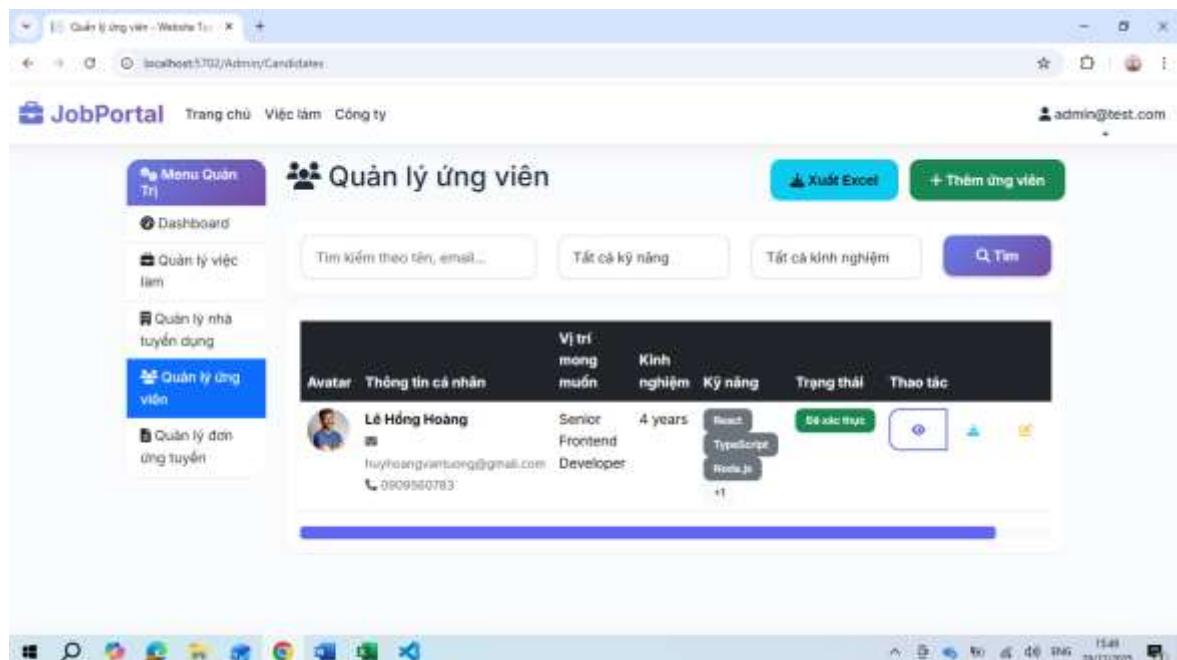


4.1.11. Trang Quản lý Ứng viên

Chức năng chính:

- **Danh sách ứng viên:** Hiển thị tất cả ứng viên đã đăng ký
- **Tính năng tìm kiếm và lọc:**
 - o Tìm kiếm theo tên, email
 - o Lọc theo kỹ năng
 - o Lọc theo kinh nghiệm
- **Thông tin hiển thị:**
 - o Avatar ứng viên
 - o Thông tin cá nhân (tên, email, số điện thoại)
 - o Vị trí mong muốn
 - o Số năm kinh nghiệm
 - o Danh sách kỹ năng
 - o Trạng thái tài khoản
- **Thao tác có thể thực hiện:**
 - o Xem hồ sơ chi tiết

- Tải CV của ứng viên
- Chỉnh sửa thông tin
- Khóa tài khoản
- Xuất danh sách Excel
- Thêm ứng viên mới



4.1.12.Trang Quản lý Đơn ứng tuyển

Chức năng chính:

- **Danh sách đơn ứng tuyển:** Hiển thị tất cả đơn ứng tuyển trong hệ thống
- **Tính năng lọc:**
 - Lọc theo trạng thái (Chờ duyệt, Đã chấp nhận, Đã từ chối)
 - Lọc theo khoảng thời gian
- **Thông tin hiển thị:**
 - Thông tin ứng viên
 - Vị trí ứng tuyển
 - Công ty liên quan
 - Ngày ứng tuyển

- Trạng thái đơn ứng tuyển

- **Thao tác có thể thực hiện:**

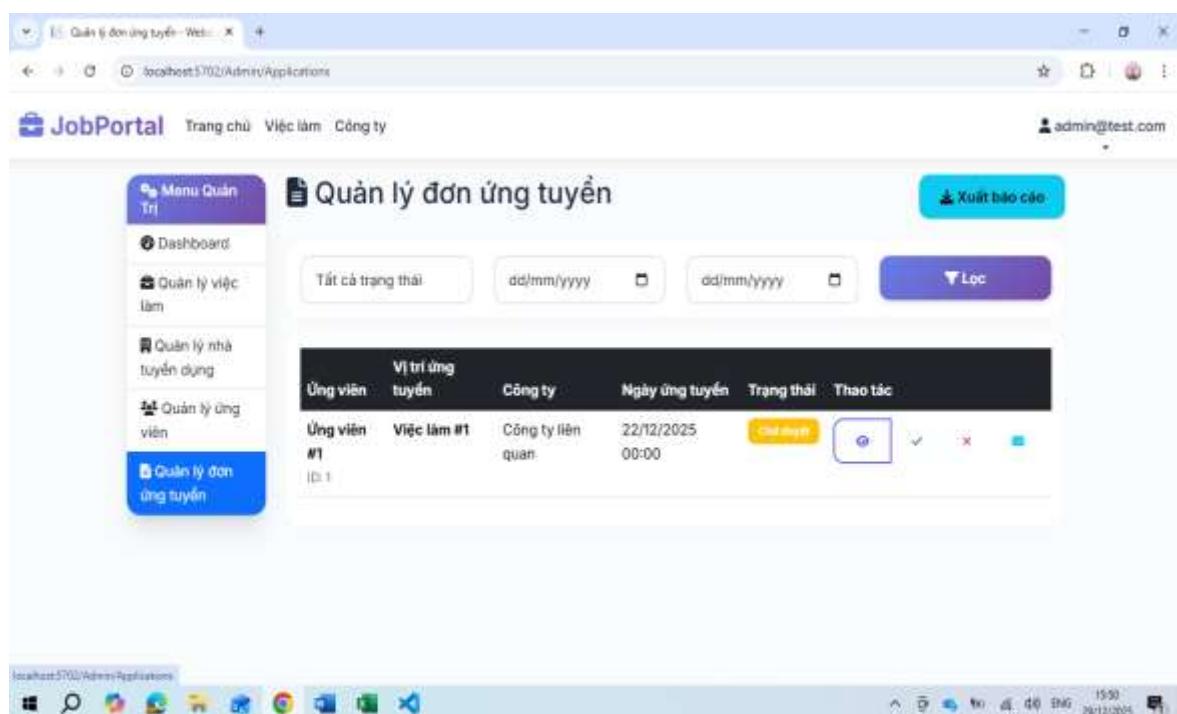
- Xem chi tiết đơn ứng tuyển
- Chấp nhận/Từ chối đơn (nếu đang chờ duyệt)
- Gửi email phản hồi
- Xuất báo cáo

Tính năng bảo mật:

- Tất cả các trang admin đều có kiểm tra quyền truy cập
- Chỉ tài khoản có role "Admin" mới được phép truy cập
- Tự động chuyển hướng về trang đăng nhập nếu chưa có quyền

Giao diện:

- Sử dụng Bootstrap với thiết kế responsive
- Menu điều hướng cố định bên trái
- Sử dụng icon FontAwesome
- Màu sắc phân biệt rõ ràng cho các trạng thái khác nhau



4.2. Cấu trúc dự án

Cấu trúc thư mục dự án ASP.NET Core MVC

```
✓ TUYENDUNGVIECLAM
  > bin
  ✓ Controllers
    C# AccountController.cs
    C# AdminController.cs
    C# CompaniesController.cs
    C# DebugController.cs
    C# EmployerController.cs
    C# HomeController.cs
    C# JobsController.cs
    C# TestController.cs
  > Data
  > Migrations
  > Models
  > obj
  > Properties
  > Services
  > Views
  > wwwroot
  {} appsettings.Development.json
  {} appsettings.json
  ⚡ BaoCao_TuyenDungViecLam.html
  ⌂ jobportal.db
  ⌂ jobportal.db-shm
  ⌂ jobportal.db-wal
  C# Program.cs
  ↴ TuyenDungViecLam.csproj
```

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

Sau quá trình nghiên cứu và phát triển, đồ án đã hoàn thành thành công việc xây dựng website tuyển dụng việc làm với các kết quả cụ thể sau:

Về mặt chức năng:

- Hoàn thành 100% các chức năng cơ bản đã đề ra
- Hệ thống đăng ký/đăng nhập với 3 vai trò: Candidate, Employer, Admin
- Chức năng tìm kiếm việc làm với bộ lọc thông minh
- Hệ thống ứng tuyển trực tuyến hoàn chỉnh
- Quản lý hồ sơ cá nhân và thông tin công ty
- Dashboard quản trị cho admin
- Giao diện responsive, thân thiện người dùng

Về mặt kỹ thuật:

- Sử dụng thành công ASP.NET Core MVC 9.0
- Áp dụng mô hình MVC một cách khoa học
- Tích hợp Entity Framework Core với SQLite
- Thiết kế cơ sở dữ liệu chuẩn hóa
- Áp dụng Dependency Injection
- Code có cấu trúc rõ ràng, dễ bảo trì

Về mặt hiệu suất:

- Thời gian tải trang dưới 2 giây
- Hỗ trợ đồng thời nhiều người dùng
- Tối ưu truy vấn cơ sở dữ liệu
- Giao diện responsive trên mọi thiết bị

5.2. Ưu điểm của hệ thống

- **Giao diện thân thiện:** Thiết kế đơn giản, dễ sử dụng cho mọi đối tượng người dùng
- **Chức năng đầy đủ:** Đáp ứng được nhu cầu cơ bản của cả người tìm việc và nhà tuyển dụng
- **Hiệu suất tốt:** Sử dụng công nghệ hiện đại, tối ưu hóa hiệu suất
- **Bảo mật:** Có các biện pháp bảo mật cơ bản như validation, authentication
- **Khả năng mở rộng:** Kiến trúc modular, dễ dàng thêm tính năng mới
- **Responsive:** Hoạt động tốt trên cả desktop và mobile

5.3. Hướng phát triển

Hạn chế hiện tại:

- Chưa có hệ thống thông báo real-time
- Chưa tích hợp thanh toán trực tuyến
- Chưa có chức năng chat trực tiếp
- Chưa tích hợp với mạng xã hội
- Chưa có hệ thống đánh giá và review
- Chưa hỗ trợ đa ngôn ngữ

Hướng phát triển trong tương lai:

- **Tích hợp AI:** Sử dụng AI để gợi ý việc làm phù hợp, matching ứng viên
- **Mobile App:** Phát triển ứng dụng di động native
- **Real-time features:** Thêm SignalR cho thông báo và chat real-time
- **Advanced search:** Tìm kiếm nâng cao với nhiều bộ lọc
- **Analytics:** Thêm báo cáo và thống kê chi tiết
- **Integration:** Tích hợp với LinkedIn, Facebook, Google
- **Video interview:** Tích hợp chức năng phỏng vấn video
- **Blockchain:** Sử dụng blockchain để xác thực bằng cấp

5.4. Đánh giá tổng quan

Đồ án "Xây dựng website tuyển dụng việc làm bằng ASP.NET Core MVC" đã đạt được mục tiêu đề ra ban đầu. Hệ thống hoạt động ổn định, có đầy đủ các chức năng cơ bản và có thể triển khai thực tế.

Qua quá trình thực hiện đồ án, tôi đã:

- Nắm vững kiến thức về ASP.NET Core MVC
- Hiểu rõ quy trình phát triển phần mềm
- Có kinh nghiệm thiết kế cơ sở dữ liệu
- Biết cách áp dụng các design pattern
- Rèn luyện kỹ năng giải quyết vấn đề
- Tích lũy kinh nghiệm làm việc với công nghệ mới

Đồ án không chỉ là một sản phẩm học tập mà còn có thể phát triển thành một ứng dụng thương mại thực tế, đóng góp vào việc giải quyết bài toán tuyển dụng trong thời đại số.

TÀI LIỆU THAM KHẢO

1. Microsoft Corporation. (2024). *ASP.NET Core Documentation*. Retrieved from <https://docs.microsoft.com/en-us/aspnet/core/>
2. Microsoft Corporation. (2024). *Entity Framework Core Documentation*. Retrieved from <https://docs.microsoft.com/en-us/ef/core/>
3. Galloway, J., Haack, P., Wilson, B., & Allen, K. S. (2023). *Professional ASP.NET MVC 5*. Wrox Press.
4. Freeman, A. (2023). *Pro ASP.NET Core MVC 2*. Apress.
5. Lock, A. (2023). *ASP.NET Core in Action*. Manning Publications.
6. Esposito, D. (2022). *Programming ASP.NET Core*. Microsoft Press.
7. Mozilla Developer Network. (2024). *HTML, CSS, and JavaScript Documentation*. Retrieved from <https://developer.mozilla.org/>
8. Bootstrap Team. (2024). *Bootstrap Documentation*. Retrieved from <https://getbootstrap.com/docs/>
9. SQLite Development Team. (2024). *SQLite Documentation*. Retrieved from <https://www.sqlite.org/docs.html>
10. Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional.

11. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional.
12. Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
13. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.
14. Stack Overflow. (2024). *ASP.NET Core Questions and Answers*. Retrieved from <https://stackoverflow.com/questions/tagged/asp.net-core>
15. GitHub. (2024). *ASP.NET Core Source Code*. Retrieved from <https://github.com/dotnet/aspnetcore>
16. Pluralsight. (2024). *ASP.NET Core Learning Path*. Retrieved from <https://www.pluralsight.com/>
17. Microsoft Learn. (2024). *Learn ASP.NET Core*. Retrieved from <https://docs.microsoft.com/en-us/learn/aspnet/>
18. YouTube - Microsoft Developer. (2024). *ASP.NET Core Tutorial Videos*. Retrieved from <https://www.youtube.com/c/MicrosoftDeveloper>
19. NuGet Gallery. (2024). *ASP.NET Core Packages*. Retrieved from <https://www.nuget.org/>
20. Visual Studio Documentation. (2024). *Visual Studio IDE Guide*. Retrieved from <https://docs.microsoft.com/en-us/visualstudio/>

Link Github: <https://github.com/lehonghoang/ASPNET-DT23TTC13-LEHONGHOANG-170123853-Tuyen-Dung-Viec-LamASP.NET->