

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**

# **LẬP TRÌNH TRỰC QUAN**

## **(IT008)**

**CHƯƠNG TRÌNH HỖ TRỢ DẠY VÀ MINH HỌA CHO CÁC THUẬT  
TOÁN SẮP XẾP**



**Giảng viên hướng dẫn: Nguyễn Thị Xuân Hương**

**Thành viên nhóm:**

- **Trần Nguyễn Thái Bình – 23520161**
- **Phan Lê Minh – 23520952**
- **Phạm Nguyên Cảnh - 23520167**

## NHIỆM VỤ ĐỒ ÁN MÔN HỌC

Họ và tên SV 1: **Trần Nguyễn Thái Bình**

MSSV: **23520161**

Họ và tên SV 2: **Phan Lê Minh**

MSSV: **23520952**

Họ và tên SV 3: **Phạm Nguyên Cảnh**

MSSV: **23520167**

Lớp: **IT008.P13**

Tên đề tài: **Xây dựng chương trình hỗ trợ dạy và minh họa cho các thuật toán sắp xếp**

Giảng viên giảng dạy: **Nguyễn Thị Xuân Hương**

Thời gian thực hiện: **Từ 1/10 đến 25/12**

Nhiệm vụ đồ án môn học(phụ thuộc vào từng chủ đề):

1. Xây dựng cơ sở dữ liệu MongoDB trên MongoDB Atlas.
2. Thiết kế giao diện phần mềm.
3. Lập trình xử lý phần mềm với các chức năng sau:
  - Mô phỏng các thuật toán sắp xếp dạng cột.
  - Ôn tập bằng trả lời câu hỏi và tính điểm.
  - Lưu kết quả ôn tập và xếp hạng.
4. Nộp file nén (\*.rar) lưu sản phẩm đề tài bao gồm:
  - File báo cáo word (\*.docx)
  - File thuyết trình (\*.pptx)
  - Thư mục chứa dự án (project), các class thư viện, CSDL, hình ảnh, ...)

*Tp.HCM, ngày ... tháng ... năm 2023*

**GIẢNG VIÊN GIẢNG DẠY**

*(Ký và ghi rõ họ tên)*

.....

## **LỜI CẢM ƠN**

Lời đầu tiên chúng em xin cảm ơn cô Nguyễn Thị Xuân Hương đã giúp đỡ, định hướng và hướng dẫn nhóm em trong thời gian tiến hành đồ án.

Vì là dự án đầu tay của chúng em đương nhiên sản phẩm sẽ không tránh khỏi những thiếu sót và hạn chế. Kính mong nhận được sự góp ý từ cô.

## **NHÓM SINH VIÊN THỰC HIỆN**

Trần Nguyễn Thái Bình – Phan Lê Minh – Phạm Nguyên Cảnh

[illegible]

**Người nhận xét**  
(Ký và ghi rõ họ tên)

.....

# MỤC LỤC

<b>1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>7</b>
1.1. Giới thiệu đề tài.....	7
1.2. Mô tả đề tài.....	7
1.3. Lý do chọn đề tài.....	7
1.4. Các chức năng chính .....	7
1.5. Công nghệ sử dụng .....	7
<b>2. GIỚI THIỆU CÔNG NGHỆ.....</b>	<b>8</b>
2.1. WPF .....	8
2.1.1. Giới thiệu.....	8
2.1.2. Lịch sử .....	8
2.1.3. Chức năng .....	9
2.1.4. Áp dụng.....	11
2.2. MongoDB.....	11
2.2.1. Giới thiệu.....	11
2.2.2. Lịch sử .....	11
2.2.3. Chức năng .....	12
2.2.4. Áp dụng.....	12
2.3. NodeJS .....	12
2.3.1. Giới thiệu.....	12
2.3.2. Lịch sử .....	12
2.3.3. Chức năng .....	12
2.3.4. Áp dụng.....	12
<b>3. THIẾT KẾ CHƯƠNG TRÌNH.....</b>	<b>13</b>
3.1. Mô tả .....	13
3.2. Chức năng .....	13
3.3. Cơ sở dữ liệu .....	13
3.3.1. Thiết kế cơ sở lưu trữ kết quả .....	13
3.3.2. Cơ sở dữ liệu mẫu.....	13
3.4. Thuật toán .....	14
<b>4. XÂY DỰNG ỨNG DỤNG .....</b>	<b>14</b>
4.1. Trang chủ .....	14
4.2. Trang trực quan thuật toán sắp xếp .....	15
4.2.1. Trang lựa chọn .....	15
4.2.2. Trang trực quan .....	15
4.3. Trang big O cheatsheet .....	15

4.4.	Trang tra cứu thông tin các các thuật toán .....	16
4.5.	Trang quiz .....	16
4.6.	Trang kết quả điểm số .....	16
4.7.	Trang hồ sơ người dùng .....	17
4.7.1.	Trang đăng nhập .....	17
4.7.2.	Trang quản lý hồ sơ .....	17
4.8.	Trang cài đặt .....	17
5.	TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN.....	18
5.1.	Ưu điểm .....	18
5.2.	Nhược điểm .....	18
5.3.	Hướng phát triển của đồ án .....	18
5.4.	Các công cụ hỗ trợ đã sử dụng .....	18
6.	TÀI LIỆU THAM KHẢO .....	18
7.	BẢNG PHÂN CÔNG.....	19

# 1. GIỚI THIỆU ĐỀ TÀI

## 1.1. Giới thiệu đề tài

- Đề án “Chương trình hỗ trợ dạy và minh họa cho các thuật toán sắp xếp” – **Visort**.

## 1.2. Mô tả đề tài

- **Visort** nhằm giúp chúng ta học tập, dễ dàng hình dung bằng cách minh họa cụ thể và giải thích cách thức hoạt động của thuật toán một cách trực quan. Đồng thời tích hợp câu hỏi để ôn tập và củng cố kiến thức.
- Minh họa thông qua các giá trị số hoặc hình ảnh.







## 1.3. Lý do chọn đề tài

Ngày nay, các ứng dụng tin học ngày càng phát triển và được áp dụng phổ biến trong nhiều lĩnh vực đời sống từ kinh tế đến khoa học xã hội, điển hình là ngành giáo dục. Việc sử dụng các ứng dụng tin học kết hợp với giảng dạy đang mang lại hiệu quả tích cực: nâng cao chất lượng giảng dạy và kiến thức cho học sinh, sinh viên. Dựa trên cơ sở đó, nhóm em tạo ra phần mềm Visort với hi vọng sẽ giúp ích cho các bạn học sinh, những người theo học lập trình và đồng thời bọn em cũng có cơ hội được áp dụng những kiến thức đã được học ở môn Lập trình trực quan.

## 1.4. Các chức năng chính

- Mô phỏng quá trình sắp xếp của các thuật toán.
- Câu hỏi lựa chọn thuật toán tối ưu để sắp xếp mảng cho trước, tính điểm số và lưu kết quả.

## 1.5. Công nghệ sử dụng

<ul style="list-style-type: none"><li>- Ngôn ngữ lập trình C#</li><li>- Dotnet Framework – WPF</li><li>- WPF UI</li></ul>			
<ul style="list-style-type: none"><li>- Git</li></ul>			
<ul style="list-style-type: none"><li>- Github</li></ul>			
<ul style="list-style-type: none"><li>- IDE – Visual Studio</li></ul>			
<ul style="list-style-type: none"><li>- MongoDB</li></ul>			

- NodeJS			
----------	-----------------------------------------------------------------------------------	--	--

## 2. GIỚI THIỆU CÔNG NGHỆ

### 2.1.WPF

#### 2.1.1. Giới thiệu

Windows Presentation Foundation (viết tắt là WPF) do Microsoft phát triển, là công nghệ kế tiếp Windows Form dùng để xây dựng các ứng dụng dành cho máy trạm chạy hệ điều hành Windows. WPF được giới thiệu từ năm 2006 trong .NET Framework 3.0 (dưới tên gọi Avalon), công nghệ này nhận được sự quan tâm của cộng đồng lập trình viên bởi nhiều điểm đổi mới trong lập trình ứng dụng và khả năng xây dựng giao diện thân thiện, sinh động.

#### 2.1.2. Lịch sử

Windows Presentation Foundation (WPF) ra đời vào ngày 21 tháng 11 năm 2006, là một phần của .NET Framework 3.0, nhằm cung cấp một nền tảng để xây dựng các ứng dụng đồ họa phong phú trên Windows.

Sau đây là các phiên bản WPF:

- **WPF 3.0 (11/2006):** Phiên bản khởi đầu với các tính năng cơ bản cho giao diện người dùng (UI) đồ họa, sử dụng DirectX cho việc render. Hỗ trợ XAML (eXtensible Application Markup Language), các tính năng như layout, styling, animation, và dữ liệu 2D/3D. Tập trung vào việc phát triển các ứng dụng có giao diện đồ họa phong phú.
- **WPF 3.5 (11/2007):** Tăng cường tính năng và hiệu suất. Cải thiện việc xử lý dữ liệu và các thành phần giao diện (controls), cùng với việc hỗ trợ tốt hơn cho các biểu mẫu (databinding) và các hiệu ứng hoạt hình (animations).
- **WPF 3.5 SP1 (08/2008):** Bản nâng cấp Service Pack với các sửa lỗi, cải thiện hiệu suất, hỗ trợ cho đồ họa pixel shader và khả năng load dữ liệu nhanh hơn. Đây là bản nâng cấp giúp cải thiện khả năng render của WPF trong các ứng dụng có đồ họa phức tạp.
- **WPF 4.0 (04/2010):** Phiên bản này tập trung vào việc cải thiện hiệu suất và các tính năng mới như hỗ trợ văn bản tốt hơn, layout và khả năng tương tác với hệ thống phông chữ (text rendering). Đưa vào các API mới và hỗ trợ tốt hơn cho đồ họa đa phương tiện (media).
- **WPF 4.5 (08/2012):** Cải tiến lớn về hiệu năng, khả năng xử lý bất đồng bộ (async/await), hỗ trợ tốt hơn cho ứng dụng có độ phân giải cao (High DPI), và cải thiện việc xử lý phông chữ. Bản này cũng hỗ trợ nhiều cách quản lý sự kiện và dễ dàng hơn trong việc phát triển giao diện người dùng phức tạp.

Sau WPF 4.5, có các thay đổi lớn trong WPF khi nó được chuyển sang nền tảng .NET Core:

Những thay đổi và cải tiến lớn trong thời gian gần đây:



- Chuyển đổi sang **.NET Core 3.0 (2019)**: Đây là bước đột phá lớn khi WPF được chuyển sang nền tảng .NET Core. Việc này giúp WPF trở nên hiện đại hơn với các cải tiến về hiệu năng, khả năng tương thích, và khả năng phát triển đa nền tảng (cross-platform) dễ dàng hơn.
- **.NET 5 (2020)**: Hợp nhất giữa .NET Framework và .NET Core. WPF tiếp tục hoạt động trên .NET 5 với hiệu suất tốt hơn và cải thiện về việc phát triển ứng dụng giao diện người dùng.
- **.NET 6 (2021)**: Tăng cường hơn nữa về hiệu suất và khả năng tối ưu hóa. Bản này hỗ trợ tốt hơn cho các máy có độ phân giải cao và các ứng dụng có quy mô lớn.
- **.NET 7 (2022)**: Cải thiện về tốc độ, tính ổn định, và khả năng mở rộng cho các ứng dụng WPF. Nhiều thay đổi về tính năng giúp việc phát triển ứng dụng đơn giản và linh hoạt hơn.

Như vậy, WPF đã tiến hóa không chỉ về tính năng mà còn về nền tảng. Nó được chuyển từ .NET Framework sang .NET Core và .NET 5/6/7, giúp cải thiện đáng kể hiệu suất, khả năng bảo trì và tính linh hoạt trong phát triển ứng dụng.

### 2.1.3. Chức năng

Dưới đây là các chức năng chính của WPF:

- **XAML (eXtensible Application Markup Language):**
  - XAML là ngôn ngữ đánh dấu được sử dụng để tạo giao diện người dùng trong WPF. Nó giúp định nghĩa giao diện dưới dạng cấu trúc phân cấp dễ đọc, dễ bảo trì.
  - Chúng ta có thể sử dụng XAML để định nghĩa các thành phần giao diện như nút, textbox, và các layout panels mà không cần viết mã C#.
- **Data Binding:**
  - WPF hỗ trợ mạnh mẽ cho tính năng data binding (ràng buộc dữ liệu), cho phép chúng ta kết nối giao diện người dùng với dữ liệu (dữ liệu có thể đến từ các nguồn như cơ sở dữ liệu, tệp, hoặc đối tượng).
  - Hỗ trợ binding một chiều, hai chiều, giúp đồng bộ hóa dữ liệu giữa giao diện và logic xử lý.
- **Styles và Templates:**
  - WPF cho phép chúng ta sử dụng styles để áp dụng một bộ thuộc tính giao diện đồng nhất cho nhiều thành phần UI, giúp cải thiện khả năng bảo trì và tùy biến giao diện.
  - Templates (control templates và data templates) cho phép chúng ta tùy chỉnh hoàn toàn cách hiển thị của các controls mà không cần phải viết lại logic của control đó.
- **Layout System:**
  - Hệ thống layout của WPF cho phép chúng ta xây dựng giao diện động có thể tự điều chỉnh kích thước và bố cục tùy theo kích thước cửa sổ, độ phân giải màn hình hoặc thay đổi dữ liệu.
  - Các thành phần layout phổ biến bao gồm: Grid, StackPanel, DockPanel, WrapPanel và Canvas.
- **2D và 3D Graphics:**
  - WPF hỗ trợ tích hợp cả đồ họa 2D và 3D thông qua DirectX, giúp tạo ra các giao diện đồ họa phong phú và phức tạp.

- Chúng ta có thể dễ dàng thêm hiệu ứng như xoay, phóng to, thu nhỏ và chiếu sáng cho các đối tượng đồ họa trong giao diện.
- **Multimedia:**
  - WPF có khả năng tích hợp âm thanh và video, giúp chúng ta có thể nhúng các phương tiện truyền thông vào giao diện.
  - Hỗ trợ các định dạng đa phương tiện khác nhau và tương thích với nhiều nguồn dữ liệu đa phương tiện (từ internet, tệp tin cục bộ).
- **Animation:**
  - WPF cung cấp hỗ trợ mạnh mẽ cho animation (hoạt hình), cho phép chúng ta tạo ra các chuyển động mượt mà và tinh vi cho các đối tượng giao diện người dùng.
  - Chúng ta có thể điều khiển các hiệu ứng chuyển động, làm mờ, và thay đổi màu sắc của các đối tượng theo thời gian.
- **Control và Custom Control:**
  - WPF cung cấp một tập hợp lớn các controls như Button, TextBox, ComboBox, ListBox, TreeView, và nhiều thành phần khác.
  - Chúng ta có thể dễ dàng tạo ra custom controls (controls tùy chỉnh) để đáp ứng nhu cầu cụ thể của ứng dụng.
- **Document Support:**
  - WPF hỗ trợ các tài liệu dạng Flow Documents và Fixed Documents, giúp chúng ta tạo và hiển thị các tài liệu với văn bản phong phú, hỗ trợ các bố cục văn bản động và tĩnh.
  - Hỗ trợ in ấn và khả năng hiển thị PDF-like documents.
- **Event Handling:**
  - Hệ thống sự kiện (events) trong WPF rất mạnh mẽ và linh hoạt. Chúng ta có thể sử dụng cơ chế routed events để xử lý sự kiện không chỉ tại đối tượng khởi nguồn, mà còn tại các đối tượng khác trong chuỗi route của sự kiện (bubbling và tunneling events).
- **Dependency Properties:**
  - WPF sử dụng dependency properties, một hệ thống thuộc tính đặc biệt giúp quản lý và theo dõi các thay đổi của thuộc tính. Hệ thống này tối ưu hóa việc thay đổi giá trị thuộc tính và giảm thiểu việc tiêu tốn tài nguyên.
- **Triggers và Behaviors:**
  - WPF cung cấp triggers để thay đổi giao diện khi một điều kiện cụ thể xảy ra (ví dụ: thay đổi màu của nút khi được nhấn).
  - Behaviors là cách tiếp cận để thêm logic cho các controls mà không cần viết mã C#, làm cho việc tạo hiệu ứng tương tác trở nên linh hoạt hơn.
- **High DPI Support:**

- WPF hỗ trợ tốt cho các màn hình có độ phân giải cao (High DPI), giúp các ứng dụng có giao diện sắc nét và điều chỉnh tự động khi chạy trên các màn hình có mật độ điểm ảnh khác nhau.

- **Globalization và Localization:**

- WPF hỗ trợ việc phát triển ứng dụng đa ngôn ngữ với tính năng globalization và localization, giúp chúng ta dễ dàng tùy chỉnh giao diện cho nhiều ngôn ngữ và văn hóa khác nhau.

Nhìn chung, WPF cung cấp một bộ công cụ toàn diện và linh hoạt để phát triển các ứng dụng có giao diện đồ họa phức tạp và hiện đại trên Windows.

#### 2.1.4. Áp dụng

- Dùng WPF để tạo ra giao diện đồ họa của Visort, trực quan hoá hình ảnh, từ đó nâng cao trải nghiệm người dùng.

## 2.2. MongoDB

### 2.2.1. Giới thiệu

- MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng, trong đó, các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).
- Đặc điểm chính:
  - Không có lược đồ cố định (schema-less): Các bản ghi trong MongoDB không cần phải tuân theo một cấu trúc lược đồ cứng nhắc.
  - Khả năng mở rộng ngang (horizontal scaling): MongoDB có thể dễ dàng mở rộng thông qua sharding, chia nhỏ dữ liệu ra nhiều server.
  - Hỗ trợ lưu trữ tài liệu phức tạp: MongoDB hỗ trợ lưu trữ các tài liệu nhúng (embedded documents) và mảng, giúp giảm bớt các thao tác join phức tạp như trong RDBMS.

### 2.2.2. Lịch sử

- MongoDB được phát triển bởi 10gen, một công ty phần mềm có trụ sở tại New York, vào năm 2007. Ban đầu, MongoDB được thiết kế để hỗ trợ một nền tảng dịch vụ khác, nhưng sau đó được phát triển thành một sản phẩm độc lập và phát hành lần đầu tiên vào năm 2009 dưới dạng mã nguồn mở.
- Các cột mốc quan trọng:
  - 2007: MongoDB bắt đầu được phát triển bởi 10gen (sau này đổi tên thành MongoDB Inc.).
  - 2009: Phiên bản đầu tiên của MongoDB được phát hành công khai.
  - 2013: MongoDB trở thành cơ sở dữ liệu NoSQL phổ biến nhất.
  - 2017: MongoDB phát hành nền tảng dịch vụ đám mây MongoDB Atlas.

### 2.2.3. Chức năng

- **CRUD:** MongoDB hỗ trợ đầy đủ các thao tác Create, Read, Update, Delete với dữ liệu.
- **Sharding:** Tính năng phân vùng (sharding) giúp chia nhỏ và phân phối dữ liệu trên nhiều máy chủ để tăng khả năng mở rộng.
- **Replication:** Tính năng sao lưu dữ liệu (replica set) giúp bảo đảm độ tin cậy và tính sẵn sàng cao của dữ liệu.
- **Aggregation:** MongoDB cung cấp pipeline aggregation để xử lý và tổng hợp dữ liệu từ nhiều tài liệu khác nhau.
- **Indexing:** Tính năng lập chỉ mục (index) trên các trường cụ thể giúp tăng tốc độ tìm kiếm và truy vấn dữ liệu.
- **Ad-hoc Queries:** Hỗ trợ các truy vấn linh hoạt, bao gồm các điều kiện phức tạp như so sánh, regex, và các phép toán trên mảng.

### 2.2.4. Áp dụng

- Lưu kết quả trả lời trên MongoDB, hỗ trợ đồng bộ kết quả trực tuyến, xếp hạng người dùng.

## 2.3. NodeJS

### 2.3.1. Giới thiệu

- Node.js là một runtime JavaScript dựa trên V8 JavaScript engine của Google Chrome. Nó được thiết kế để chạy JavaScript ngoài trình duyệt, chủ yếu dành cho các ứng dụng phía server.
- Node.js hỗ trợ mô hình event-driven (định hướng sự kiện) và non-blocking I/O (I/O không đồng bộ), giúp xây dựng các ứng dụng hiệu suất cao, đặc biệt trong lĩnh vực real-time như chat, API, hay ứng dụng streaming.

### 2.3.2. Lịch sử

- Năm 2009: Node.js được tạo ra bởi Ryan Dahl để giải quyết các vấn đề về hiệu suất của mô hình I/O blocking trong các ứng dụng web truyền thống.
- Phiên bản đầu tiên: Phát hành vào tháng 5 năm 2009.
- NPM (Node Package Manager): Giới thiệu vào năm 2010, giúp dễ dàng quản lý thư viện và module.
- Hiện nay: Node.js được phát triển bởi OpenJS Foundation và là một trong những công nghệ phổ biến nhất cho lập trình backend.

### 2.3.3. Chức năng

- **Chạy JavaScript trên server:** Biến JavaScript thành ngôn ngữ backend mạnh mẽ.
- **Xử lý non-blocking I/O:** Thích hợp cho các ứng dụng yêu cầu xử lý nhiều kết nối đồng thời.
- **Hỗ trợ module:** Hệ sinh thái module phong phú thông qua NPM.
- **Tích hợp real-time:** Dễ dàng xây dựng các ứng dụng real-time như WebSocket, chat hoặc hệ thống thông báo.
- **Cross-platform:** Chạy trên nhiều hệ điều hành (Windows, macOS, Linux).

### 2.3.4. Áp dụng

- Cài đặt các công cụ cần thiết, khôi phục các thư viện cần cho việc phát triển phần mềm ViSort một cách nhanh chóng và tiện lợi.

## 3. THIẾT KẾ CHƯƠNG TRÌNH

### 3.1. Mô tả

- Các thuật toán sắp xếp có một vai trò vô cùng thiết yếu và quan trọng trong đời sống kinh tế xã hội. Chúng được áp dụng trong nhiều ngành nghề và lĩnh vực khác nhau như quản lý dữ liệu, tìm kiếm và tối ưu hóa, đồng thời thuật toán sắp xếp là một trong những yếu tố cơ bản mà những người mới bắt đầu học lập trình được tiếp xúc. Việc hiểu rõ cách hoạt động của các thuật toán sắp xếp và áp dụng chúng hiệu quả trong từng trường hợp cụ thể sẽ mang lại lợi ích rất lớn.

### 3.2. Chức năng

Các chức năng trong ViSort bao gồm:

- Mô phỏng quá trình sắp xếp các giá trị số theo dạng cột.
- Hỗ trợ chỉnh tốc độ trực quan thuật toán.
- Câu hỏi xác định thuật toán tối ưu (5 câu hỏi định sẵn, có ngẫu nhiên thứ tự câu hỏi).
- Đăng ký, đăng nhập, xóa tài khoản lưu trữ kết quả các câu trả lời.
- Lưu kết quả người dùng lên cơ sở dữ liệu trên MongoDB.

### 3.3. Cơ sở dữ liệu

#### 3.3.1. Thiết kế cơ sở lưu trữ kết quả

username	password	score
----------	----------	-------

- Trường **score** có giá trị là số nguyên từ 0 đến 999.
- Trường **password** sẽ được băm.

#### 3.3.2. Cơ sở dữ liệu mẫu

USER NAME	PASSWORD	SCORE
kevinnitro	8DE3ED47276E2C5113D3BC4B3C8DC806C7104F2EF56A7CC8E0F1217B11D38832:8970AC6457D30F9F08A98FC4FE58F5F2:50000:SHA256	390
leminion	2EF411611F4D4E2E17AF0114FE351FE516BBF45DC9FC944CFF6559512B372C86:6800337A552261A70ACA5B5E0C47AE06:50000:SHA256	600
PNCanh	E509506C62DB89B694DECD123F0596E0B8258E6F37AF49FD29D5A54E884D4F93:3F67974C398A406C37C735EB7C5217EF:50000:SHA256	506

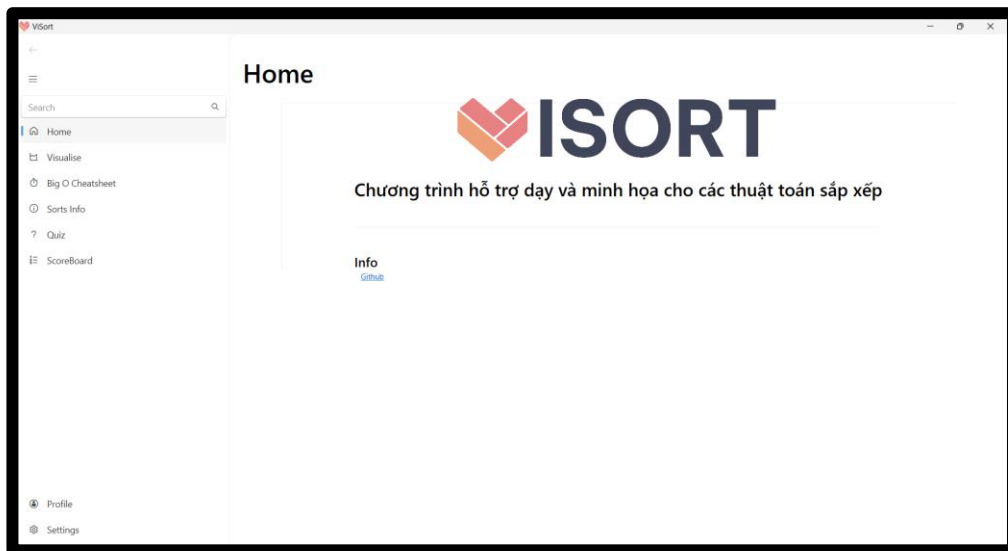
<pre> _id: ObjectId('67475f66c570453db54f1ee0') username : "lamin" password : "DA02AF9927289B5BCDE3CF7B8F758D8BD76084B60BA1220F870FEA79D9BD90DA:E2EEE..." score : 999 </pre>
<pre> _id: ObjectId('6747bb7abc27481c6c3b29ab') username : "user1" password : "67022E087F7A5ACEFB335CA6C818BB52E86B1B77A13AF73A6BC889CC6BB12AC8:E3778..." score : 0 </pre>
<pre> _id: ObjectId('6747be1c6368d7bb6dc508c8') username : "user2" password : "9870EB4B8DD887D76E1450A6622C83704E9B05A26289CAECA54C1B1638CED324:85375..." score : 456 </pre>
<pre> _id: ObjectId('6747be679d4a26cab19df032') username : "user3" password : "DEA27529CEAD2310C66066F1C3B5DF512E2EF1249E9EE69C1A956D035FBE8B74:BDD29..." score : 359 </pre>

### 3.4.Thuật toán

1. Thiết kế giao diện tổng quát, các nút lựa chọn, ô nhập liệu...
2. Tổng hợp và xử lý trực quan các loại thuật toán sắp xếp:
  - Selection Sort
  - Buble Sort
  - Insertion Sort
  - Heap Sort
  - Quick Sort
  - Merge Sort
  - Bucket Sort
  - Radix Sort
  - Count Sort
  - Shell Sort
  - Tim Sort
  - Tree Sort
3. Thiết kế câu hỏi ôn tập lựa chọn thuật toán ứng với từng câu hỏi mảng số nguyên.
4. Thuật toán để tính điểm số kết quả các câu hỏi trả lời.
5. Lưu trữ dữ liệu người chơi lên hệ thống cơ sở dữ liệu MongoDB. Cho phép đăng, xoá kết quả chơi thông qua username, password.

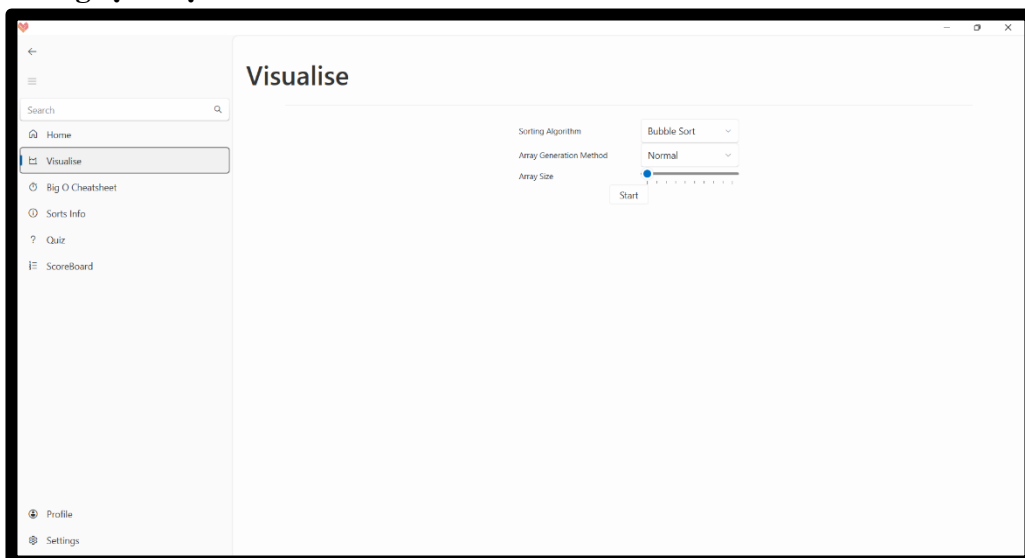
## 4. XÂY DỰNG ỨNG DỤNG

### 4.1.Trang chủ

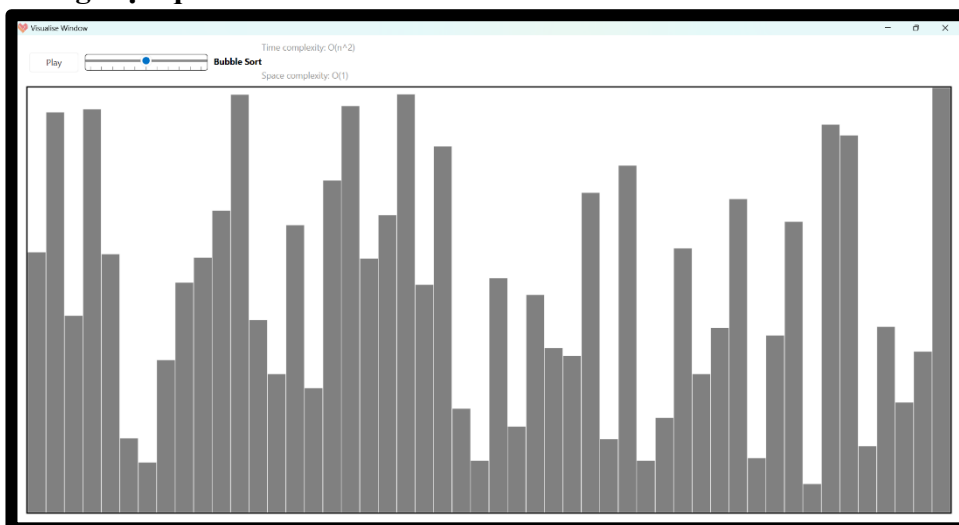


## 4.2. Trang trực quan thuật toán sắp xếp

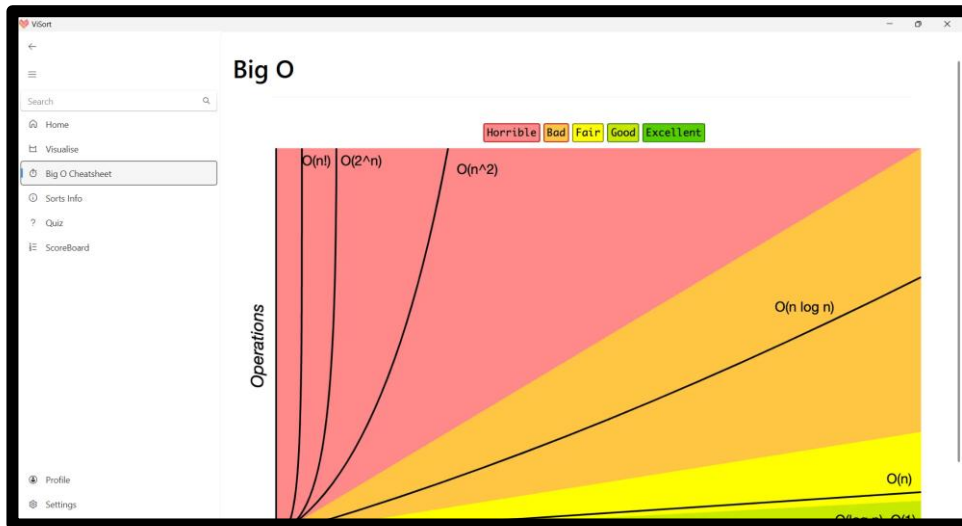
### 4.2.1. Trang lựa chọn



### 4.2.2. Trang trực quan



## 4.3. Trang big O cheatsheet



#### 4.4. Trang tra cứu thông tin các thuật toán

The 'Sorts Info' page shows details for 'Bubble Sort'. It includes a search bar with 'Bubble Sort' entered, a list of search results with 'Bubble Sort' selected, and a 'Bucket Sort' option. The main content area for 'Bubble Sort' provides the following information:

- Information: [Geeks for Geeks](#)
- Video: [YouTube](#)
- Time complexity:  $O(n^2)$
- Space complexity:  $O(1)$

#### 4.5. Trang quiz

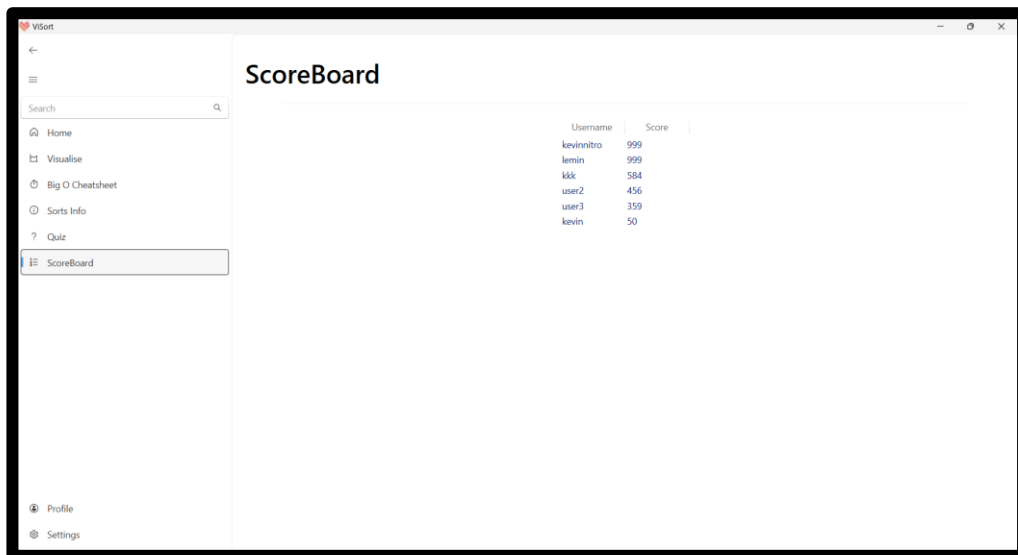
The 'Quiz' page displays a bar chart with 15 bars of increasing height. Above the chart, the numbers 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40, 50, 60, 70 are listed. Below the chart, there is a list of 12 sorting algorithms with radio buttons for selection:

- ☐ a. Bubble Sort
- ☐ b. Bucket Sort
- ☐ c. Counting Sort
- ☐ d. Heap Sort
- ☐ e. Insertion Sort
- ☐ f. Merge Sort
- ☐ g. Quick Sort
- ☐ h. Radix Sort
- ☐ i. Selection Sort
- ☐ j. Shell Sort
- ☐ k. Tim Sort
- ☐ l. Tree Sort

Navigation buttons include 'Previous', 'Kết thúc' (End), and 'Next'. A 'Hướng dẫn' (Guide) button is also present in the top right corner.

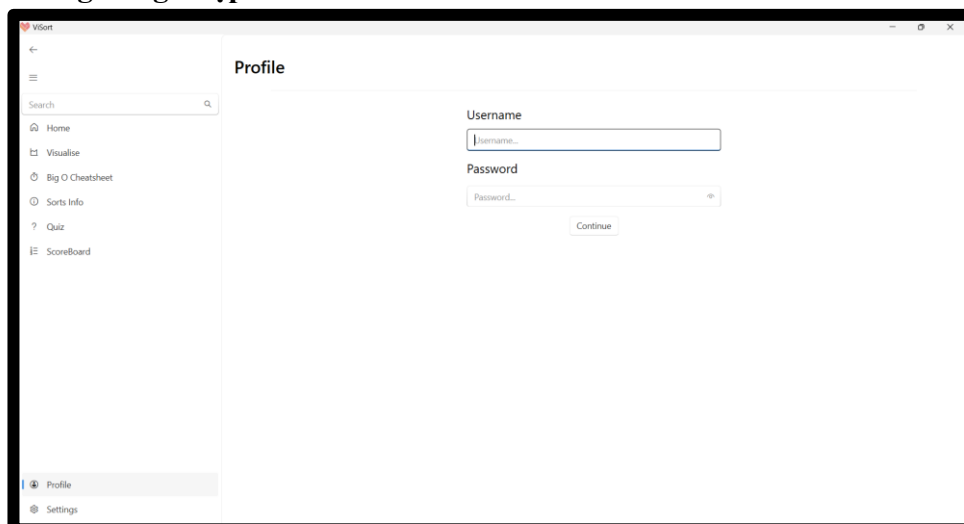
#### 4.6. Trang kết quả điểm số



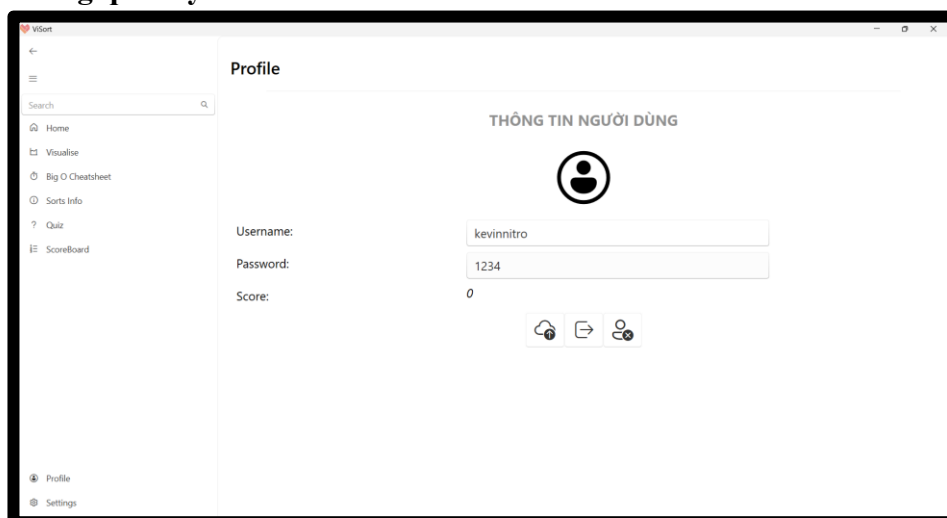


## 4.7. Trang hồ sơ người dùng

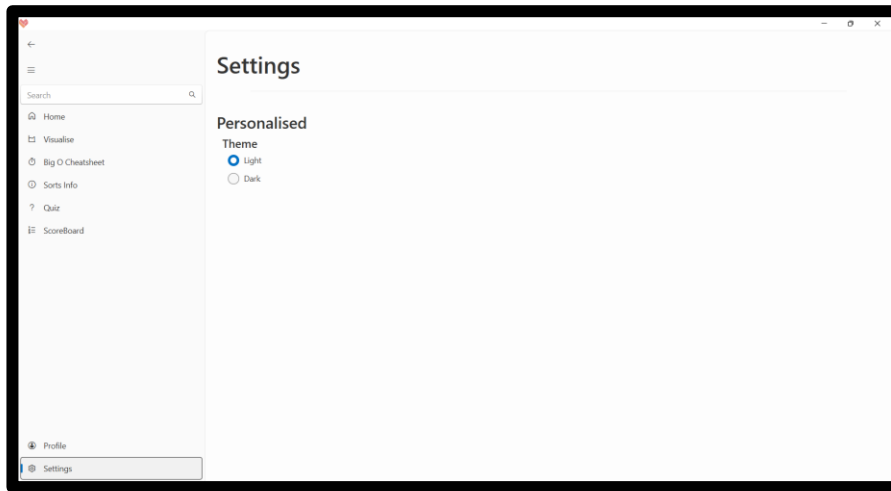
### 4.7.1. Trang đăng nhập



### 4.7.2. Trang quản lý hồ sơ



## 4.8. Trang cài đặt



## 5. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Ưu điểm

- Dễ sử dụng.
- Hỗ trợ ôn tập tính điểm.
- Cung cấp file thực thi.

### 5.2. Nhược điểm

- Bảo mật database chưa tốt, MongoDB URI trực tiếp trong source code.
- Hàm tính điểm số chưa được tối ưu.

### 5.3. Hướng phát triển của đồ án

- Tổng hợp thêm các thuật toán
- Tối ưu chế độ tối.
- Thêm câu hỏi và sắp xếp ngẫu nhiên mang tính công bằng và đổi mới.

### 5.4. Các công cụ hỗ trợ đã sử dụng

- [Conventional commit](#).
- [Commitlint](#).
- Pre-commit git hook bởi [husky.net](#) để format C# bằng [dotnet format](#) và format XAML bằng [XamlStyler](#).
- Bot [Deepsources](#), [CodeFactor](#) kiểm tra code convention, syntax, ...
- [Git-cliff](#) tạo changelog.
- Github action (CI/CD):
  - o Tự động phân phối file thực thi dựa trên các phiên bản chương trình.
  - o Kiểm tra format trên từng commit.

## 6. TÀI LIỆU THAM KHẢO

- [Lập trình WPF cơ bản - K team](#)
- [C# Guide - .NET managed language | Microsoft Learn](#)
- [MongoDB C# Driver - C#/.NET v3.0](#)
- [Complete MongoDB Tutorial - Net Ninja](#)

## 7. BẢNG PHÂN CÔNG

CÔNG VIỆC	THÁI BÌNH	LÊ MINH	NGUYỄN CẢNH
Viết báo cáo	X	X	X
Thuyết trình	X	X	X
Tổng hợp thuật toán	X	X	X
Thiết kế quiz, scoreboard		X	
Thiết kế userprofile		X	
Thiết kế trực quan hoá thuật toán			X
Thiết kế database	X		
Thiết kế giao diện chính	X		
CI/CD, Github action	X		