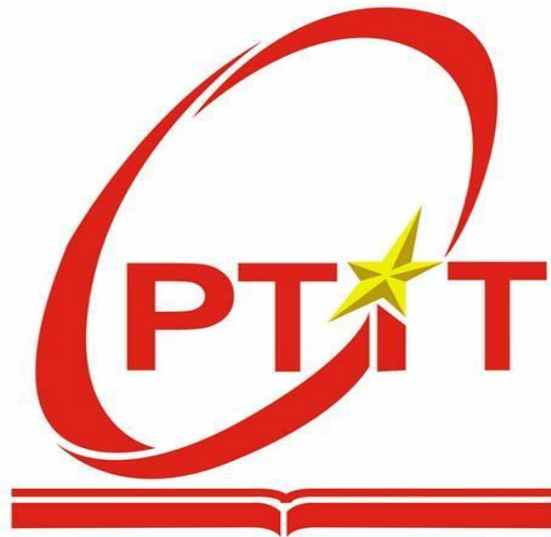


BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO HÀNG TUẦN
THỰC TẬP CƠ SỞ

THIẾT KẾ XÂY DỰNG WEBSITE TÍNH ĐIỂM GPA KẾT HỢP LƯU
TRỮ BÀI GIẢNG

Giảng viên hướng dẫn: TS. Kim Ngọc Bách

Họ và tên sinh viên: Phạm Trung Kiên

Lớp: E22CQCN02 – B

CHƯƠNG 1. TỔNG QUAN

Trong chương 1, đồ án trình bày cái nhìn tổng quan về đề tài, bao gồm:

- **Phát biểu bài toán**
- **Mục tiêu**
- **Phạm vi đề tài**
- **Đề xuất giải pháp**
- **Các công nghệ sử dụng**
- **Kết luận**

1.1. Phát biểu bài toán

Hiện nay số người dùng Internet trên toàn cầu đã đạt gần 4,66 tỷ người và con số này vẫn tiếp tục tăng theo từng năm. Sự phát triển vượt bậc của công nghệ đã mở ra nhiều cơ hội trong lĩnh vực giáo dục trực tuyến, giúp tối ưu hóa quy trình học tập và giảng dạy. Trong bối cảnh đó, nhu cầu quản lý thông tin học tập và đánh giá kết quả học tập trở nên cấp thiết hơn bao giờ hết.

Dựa trên thực tiễn đó, đề tài “Hệ thống tính điểm GPA kết hợp lưu trữ bài giảng” được xây dựng nhằm đáp ứng nhu cầu của sinh viên, giảng viên và quản trị viên trong môi trường giáo dục hiện đại. Hệ thống này không chỉ giúp sinh viên dễ dàng theo dõi và tính toán điểm GPA theo từng học kỳ mà còn cung cấp một kho lưu trữ bài giảng đa dạng, cho phép người dùng tra cứu, xem trước và tải về tài liệu học tập một cách nhanh chóng.

Hình thức bài toán:

Xây dựng một hệ thống quản lý học tập trực tuyến với các đối tượng sử dụng chính gồm:

- **Sinh viên (Student):**
 - Nhập và lưu trữ điểm số các môn học, từ đó tự động tính toán GPA.
 - Tra cứu bài giảng và tài liệu học tập theo môn học, chủ đề hoặc từ khóa.
 - Xem trước nội dung bài giảng trực tuyến mà không cần tải xuống.
- **Giảng viên (Lecturer):**

- Đăng tải, cập nhật và quản lý bài giảng, tài liệu tham khảo cho sinh viên.
- Theo dõi tiến trình học tập và kết quả GPA của sinh viên qua từng kỳ học.

- Quản trị viên (Admin):

- Quản lý người dùng, phân quyền truy cập và kiểm soát nội dung bài giảng trên hệ thống.
- Giám sát dữ liệu điểm số và điều chỉnh các thông số liên quan đến tính toán GPA.

Việc tích hợp tính năng tính điểm GPA cùng với kho lưu trữ bài giảng sẽ tạo ra một nền tảng học tập toàn diện, hỗ trợ tối đa cho quá trình đào tạo và học tập trực tuyến. Hệ thống không chỉ nâng cao hiệu quả quản lý học tập mà còn góp phần thúc đẩy sự chuyển đổi số trong lĩnh vực giáo dục.

1.2. Mục tiêu

- Hiểu về các giải pháp xây dựng hệ thống quản lý học tập trực tuyến, đặc biệt là tích hợp các chức năng tự động tính điểm GPA và lưu trữ bài giảng.
- Xây dựng một website với chức năng tính điểm GPA tự động cho sinh viên, đồng thời cung cấp kho lưu trữ bài giảng để giảng viên đăng tải và quản trị viên kiểm soát nội dung học tập.

1.3. Phạm vi đề tài

Trong giới hạn của đề tài, đồ án tập trung vào việc xây dựng chức năng tự động tính toán GPA cho sinh viên và tích hợp hệ thống lưu trữ bài giảng trực tuyến, phù hợp với nhu cầu quản lý học tập của các trường đại học và các cơ sở đào tạo. Dữ liệu bao gồm thông tin điểm số các môn học, bài giảng và tài liệu học tập, cùng với thông tin người dùng (sinh viên, giảng viên và quản trị viên). Dữ liệu sau khi được thu thập sẽ được xử lý và tinh chỉnh để phù hợp với cấu trúc cơ sở dữ liệu đã xây dựng.

1.4. Đề xuất giải pháp

Dựa trên việc tìm hiểu các hệ thống quản lý học tập trực tuyến hiện nay, hệ thống tính điểm GPA kết hợp lưu trữ bài giảng sẽ đáp ứng đầy đủ các nhu cầu thiết yếu của sinh viên, giảng viên và quản trị viên. Hệ thống này được thiết kế với các chức năng chính sau:

- **Sinh viên:**

- Tính điểm GPA tự động: Nhập điểm số và số tín chỉ của các môn học để hệ thống tự động tính toán GPA theo công thức chuẩn.
- Tra cứu lịch sử điểm: Theo dõi kết quả điểm số và tiến trình học tập qua từng học kỳ.
- Tìm kiếm bài giảng: Dễ dàng tìm kiếm và truy cập bài giảng, tài liệu học tập theo môn học, chủ đề hoặc từ khóa.
- Xem trước bài giảng trực tuyến: Hỗ trợ xem trước nội dung bài giảng mà không cần tải xuống.
- Quản lý hồ sơ cá nhân: Cập nhật và quản lý thông tin cá nhân cùng các dữ liệu học tập liên quan.

- **Giảng viên:**

- Đăng tải và quản lý bài giảng: Tạo, cập nhật và chia sẻ bài giảng, tài liệu học tập cho sinh viên.
- Theo dõi kết quả học tập: Giám sát tiến trình học tập của sinh viên và cung cấp phản hồi, tư vấn kịp thời.

- **Quản trị viên:**

- Quản lý người dùng: Phân quyền và bảo mật truy cập cho sinh viên và giảng viên, đảm bảo an toàn hệ thống.
- Quản lý nội dung: Kiểm soát, duyệt và điều chỉnh nội dung bài giảng cùng dữ liệu điểm số.
- Điều chỉnh cài đặt hệ thống: Giám sát toàn bộ hoạt động của hệ thống và thực hiện các điều chỉnh cần thiết nhằm tối ưu hóa hiệu suất.

Với giải pháp này, hệ thống không chỉ giúp sinh viên theo dõi kết quả học tập một cách chính xác mà còn tạo ra một kho lưu trữ bài giảng chất lượng, hỗ trợ việc học tập và giảng dạy một cách toàn diện và hiệu quả.

1.5. Các công nghệ sử dụng

1.5.1. Giới thiệu về SQL Server [2]

Giới thiệu tổng quan về SQL Server Database

SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft, sử dụng ngôn ngữ truy vấn có cấu trúc (SQL) để lưu trữ, quản lý và truy xuất dữ liệu. Với kiến trúc dựa trên các bảng dữ liệu có mối quan hệ chặt chẽ, SQL Server đảm bảo tính toàn vẹn và bảo mật của thông tin

thông qua các ràng buộc và giao dịch theo tiêu chuẩn ACID (Atomicity, Consistency, Isolation, Durability).

Ra đời từ cuối những năm 1980, SQL Server đã trải qua nhiều phiên bản cải tiến như SQL Server 2000, 2005, 2012, 2016, 2019 và phiên bản mới nhất. Hệ thống được thiết kế nhằm đáp ứng nhu cầu của các doanh nghiệp từ vừa đến lớn, với khả năng mở rộng linh hoạt và hiệu năng xử lý dữ liệu mạnh mẽ.

Các tính năng nổi bật của SQL Server bao gồm:

- **Hỗ trợ T-SQL:** Sử dụng Transact-SQL, một phiên bản mở rộng của SQL, giúp thực hiện các truy vấn phức tạp, xử lý giao dịch và quản lý dữ liệu một cách hiệu quả.
- **Quản lý giao dịch:** Đảm bảo tính nhất quán của dữ liệu thông qua cơ chế giao dịch và khóa, giúp bảo vệ dữ liệu trong quá trình xử lý.
- **Khả năng mở rộng:** Hỗ trợ mở rộng cả theo chiều dọc (nâng cấp phần cứng) và chiều ngang (phân phối tải), phù hợp với các hệ thống có lượng dữ liệu lớn.
- **Bảo mật cao:** Cung cấp các tính năng bảo mật tiên tiến như mã hóa dữ liệu, xác thực người dùng và kiểm soát truy cập.
- **Tích hợp và phân tích dữ liệu:** Tích hợp chặt chẽ với các công cụ của Microsoft như Power BI và Azure, hỗ trợ phân tích dữ liệu và báo cáo nâng cao.

Nhờ vào các ưu điểm trên, SQL Server đã trở thành một giải pháp quản lý dữ liệu hàng đầu, được ứng dụng rộng rãi trong các hệ thống doanh nghiệp, giúp tối ưu hóa quy trình xử lý và phân tích dữ liệu một cách hiệu quả và an toàn.

Tại sao lại cần SQL Server?

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ được phát triển bởi Microsoft, cung cấp các giải pháp toàn diện cho việc quản lý, lưu trữ và xử lý dữ liệu. Dưới đây là những lý do chính tại sao cần sử dụng SQL Server:

- **Tính toàn vẹn và nhất quán của dữ liệu:**
SQL Server hỗ trợ các giao dịch theo chuẩn ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo mọi giao dịch được thực hiện hoàn hảo và dữ liệu luôn ở trạng thái nhất quán, điều này rất quan trọng đối với các ứng dụng yêu cầu độ tin cậy cao.
- **Bảo mật mạnh mẽ:**
Hệ thống cung cấp các tính năng bảo mật tiên tiến như mã hóa dữ liệu, xác thực người dùng, phân quyền truy cập và kiểm soát truy cập chi tiết, giúp bảo vệ dữ liệu khỏi các truy cập trái phép và rủi ro bảo mật.

- **Khả năng mở rộng và hiệu năng cao:**
SQL Server được thiết kế để xử lý khối lượng dữ liệu lớn, có khả năng mở rộng theo cả chiều dọc (nâng cấp phần cứng) và chiều ngang (phân phối tải). Điều này cho phép hệ thống luôn duy trì hiệu suất tốt ngay cả khi lượng dữ liệu và số lượng giao dịch tăng mạnh.
- **Tích hợp với hệ sinh thái Microsoft:**
SQL Server tích hợp chặt chẽ với các công cụ và dịch vụ của Microsoft như Azure, Power BI, và Visual Studio, tạo nên một môi trường phát triển và vận hành toàn diện, hỗ trợ việc phân tích, báo cáo và trực quan hóa dữ liệu một cách hiệu quả.
- **Hỗ trợ ngôn ngữ truy vấn mạnh mẽ:**
Với ngôn ngữ T-SQL – phiên bản mở rộng của SQL, SQL Server cho phép thực hiện các truy vấn phức tạp, xử lý và phân tích dữ liệu một cách nhanh chóng và dễ dàng, từ đó giúp cải thiện hiệu suất làm việc của các nhà phát triển và quản trị viên.

Một số ưu điểm của SQL Server có thể kể đến như:

- **Tính toàn vẹn và nhất quán của dữ liệu:**
Hỗ trợ giao dịch theo tiêu chuẩn ACID, đảm bảo mọi giao dịch được thực hiện hoàn hảo và dữ liệu luôn được duy trì ở trạng thái nhất quán.
- **Bảo mật cao:**
Cung cấp các tính năng bảo mật tiên tiến như mã hóa dữ liệu, xác thực người dùng, phân quyền truy cập và kiểm soát truy cập chi tiết, giúp bảo vệ dữ liệu khỏi các truy cập trái phép.
- **Hiệu năng vượt trội:**
Tối ưu hóa truy vấn và xử lý giao dịch nhanh chóng, cho phép quản lý lượng dữ liệu lớn và đáp ứng nhu cầu xử lý dữ liệu phức tạp trong doanh nghiệp.
- **Khả năng mở rộng linh hoạt:**
Hỗ trợ mở rộng theo chiều dọc (nâng cấp phần cứng) và chiều ngang (phân phối tải), giúp hệ thống luôn hoạt động ổn định khi khối lượng dữ liệu và số lượng giao dịch tăng lên.
- **Tích hợp chặt chẽ với hệ sinh thái Microsoft:**
Dễ dàng tích hợp với các công cụ và dịch vụ như Azure, Power BI, và Visual Studio, tạo nên một môi trường phát triển và quản trị dữ liệu toàn diện.
- **Công cụ quản trị mạnh mẽ:**
Cung cấp các giao diện quản trị, công cụ giám sát, sao lưu và phục hồi dữ liệu hiệu quả, giúp quản trị viên dễ dàng theo dõi và duy trì hệ thống.
- **Hỗ trợ ngôn ngữ T-SQL:**
Với T-SQL (Transact-SQL), SQL Server cho phép viết các truy vấn phức tạp, xử lý dữ liệu và phân tích thông tin một cách dễ dàng và hiệu quả.

Các loại SQL Server Database chủ yếu được phân loại theo các phiên bản (edition) khác nhau để phù hợp với nhu cầu và quy mô sử dụng của doanh nghiệp. Một số loại chính bao gồm:

- **SQL Server Express:**
Phiên bản miễn phí, phù hợp với các ứng dụng nhỏ, cho mục đích học tập, phát triển hoặc các dự án quy mô nhỏ. Có giới hạn về dung lượng dữ liệu và tính năng.
- **SQL Server Developer:**
Cung cấp đầy đủ các tính năng của phiên bản Enterprise nhưng được cấp phép chỉ cho mục đích phát triển và thử nghiệm, không dùng cho môi trường sản xuất.
- **SQL Server Standard:**
Phù hợp với các doanh nghiệp nhỏ và vừa, cung cấp những tính năng cơ bản cần thiết để quản lý dữ liệu với hiệu năng ổn định và chi phí hợp lý.
- **SQL Server Enterprise:**
Dành cho các tổ chức lớn và các ứng dụng quan trọng, cung cấp các tính năng tiên tiến, khả năng mở rộng cao, hiệu năng tối ưu và bảo mật nâng cao.
- **SQL Server Web:**
Được thiết kế dành riêng cho các ứng dụng web, cung cấp các tính năng cần thiết để hỗ trợ hoạt động của các trang web và dịch vụ trực tuyến với chi phí thấp.
- **Azure SQL Database:**
Là dịch vụ cơ sở dữ liệu dựa trên đám mây của Microsoft trên nền tảng Azure, cung cấp khả năng mở rộng linh hoạt, bảo mật cao và quản lý tự động, phù hợp với các ứng dụng đòi hỏi tính linh hoạt và khả năng truy cập từ mọi nơi.

Mỗi loại phiên bản được thiết kế để đáp ứng các nhu cầu khác nhau, từ dự án cá nhân, ứng dụng nhỏ đến các hệ thống doanh nghiệp quy mô lớn và các giải pháp đám mây.

1.5.2. Giới thiệu NodeJS và ExpressJS [3]



Hình 1.1. NodeJS và ExpressJS

*Giới thiệu về Node.js

1. Node.js là gì?

Node.js là một nền tảng **JavaScript runtime** được xây dựng trên **V8 Engine** của Google Chrome. Nó giúp chạy mã JavaScript trên **server-side** thay vì chỉ trong trình duyệt như trước đây. Điều này giúp JavaScript có thể sử dụng để phát triển các ứng dụng backend mạnh mẽ, không chỉ giới hạn ở frontend.

2. Đặc điểm nổi bật của Node.js

- **Single-threaded & Non-blocking I/O (Mô hình đơn luồng, bất đồng bộ)**
Node.js sử dụng mô hình **event-driven (sự kiện)** và **non-blocking I/O (I/O không chặn)**, giúp xử lý nhiều yêu cầu cùng một lúc mà không cần tạo nhiều luồng như các hệ thống khác.
- **Hiệu suất cao với V8 Engine**
Node.js sử dụng **V8 JavaScript Engine** (cùng engine với Google Chrome), giúp thực thi mã JavaScript nhanh chóng và tối ưu hiệu năng.
- **Quản lý gói linh hoạt với npm (Node Package Manager)**
Node.js đi kèm với **npm**, giúp quản lý hàng triệu thư viện mã nguồn mở, giúp lập trình viên dễ dàng tích hợp các module cần thiết vào dự án.
- **Chạy trên nhiều nền tảng**
Node.js có thể chạy trên Windows, macOS, Linux mà không cần chỉnh sửa nhiều mã nguồn.
- **Hỗ trợ Microservices & API development**
Nhờ kiến trúc nhẹ, Node.js rất phù hợp để xây dựng **API RESTful**, **WebSocket**, **Microservices**, giúp tạo các ứng dụng hiện đại, có khả năng mở rộng cao.

3. Ứng dụng của Node.js

- **Phát triển Backend cho Web:** Node.js thường được dùng để xây dựng các API, hệ thống backend cho các ứng dụng web, thường kết hợp với Express.js.
- **Ứng dụng real-time (thời gian thực):** Node.js rất phổ biến trong các ứng dụng như **chat, game online, trading platform** nhờ khả năng xử lý bất đồng bộ nhanh chóng.
- **Ứng dụng IoT:** Node.js có thể sử dụng trong các hệ thống IoT để xử lý dữ liệu từ nhiều thiết bị cảm biến cùng lúc.
- **Xây dựng hệ thống streaming (phát trực tuyến):** Các ứng dụng như Netflix, YouTube sử dụng Node.js để tối ưu tốc độ truyền tải nội dung.

4. Vì sao nên dùng Node.js?

- **Hiệu suất cao và xử lý tốt số lượng request lớn.**
- **Dễ học, dễ sử dụng cho các lập trình viên JavaScript.**
- **Cộng đồng lớn, nhiều thư viện hỗ trợ.**
- **Thích hợp cho các ứng dụng thời gian thực, web API, microservices.**

Với những ưu điểm trên, Node.js đã trở thành một lựa chọn phổ biến trong việc phát triển ứng dụng web hiện đại.

*Giới thiệu về ExpressJS

ExpressJS là một framework web nhẹ, nhanh và linh hoạt dành cho Node.js. Được xây dựng nhằm đơn giản hóa quá trình phát triển các ứng dụng web và API, ExpressJS cung cấp một bộ công cụ mạnh mẽ giúp bạn dễ dàng xử lý các yêu cầu HTTP, định tuyến (routing) và quản lý middleware.

Đặc điểm nổi bật của ExpressJS:

- **Đơn giản và tối giản:**
ExpressJS mang lại một cấu trúc cơ bản nhưng mạnh mẽ, cho phép lập trình viên tạo ra các ứng dụng web chỉ với một số dòng code, giúp tiết kiệm thời gian phát triển.
- **Middleware linh hoạt:**
Express cho phép bạn sử dụng các middleware để xử lý các tác vụ như xác thực, ghi log, xử lý lỗi và các nhiệm vụ khác trước hoặc sau khi xử lý các yêu cầu từ client.
- **Hỗ trợ định tuyến mạnh mẽ:**
Framework này cung cấp cơ chế định tuyến dễ sử dụng, giúp quản lý các endpoint của API hoặc các trang web một cách rõ ràng và có cấu trúc.

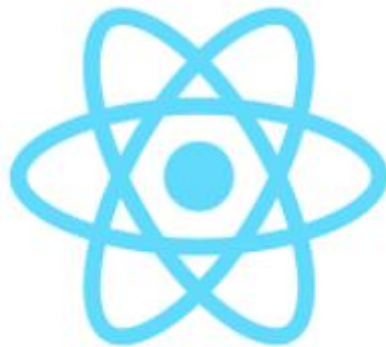
- **Tích hợp dễ dàng với các công cụ khác:**

ExpressJS có thể kết hợp linh hoạt với các công cụ template engines như EJS, Pug, hoặc Handlebars để xây dựng giao diện người dùng động, đồng thời dễ dàng tích hợp với các cơ sở dữ liệu, hệ thống xác thực và các module khác trong hệ sinh thái Node.js.

- **Hỗ trợ xây dựng API:**

Nhờ khả năng xử lý bất đồng bộ và hiệu suất cao của Node.js, ExpressJS rất phù hợp để xây dựng các API RESTful phục vụ cho các ứng dụng web, mobile, hay các hệ thống microservices.

1.5.3. Giới thiệu ReactJS [4]



Hình 1.2. ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook, nhằm mục đích xây dựng giao diện người dùng (UI) cho các ứng dụng web một cách hiệu quả và linh hoạt. ReactJS cho phép tạo ra các component độc lập, có thể tái sử dụng, giúp việc phát triển ứng dụng trở nên có cấu trúc rõ ràng và dễ bảo trì.

Một trong những điểm mạnh của ReactJS là cơ chế **Virtual DOM**. Thay vì thao tác trực tiếp trên DOM thật, React xây dựng một bản sao ảo của DOM để thực hiện các thay đổi, từ đó chỉ cập nhật những phần cần thiết khi có sự thay đổi trong dữ liệu. Điều này không chỉ giúp tăng tốc độ xử lý mà còn cải thiện hiệu suất của ứng dụng.

ReactJS cũng hỗ trợ lập trình theo kiểu **declarative**, giúp cho việc định nghĩa giao diện trở nên dễ hiểu hơn. Thay vì chỉ rõ cách thức cập nhật giao diện, lập trình viên chỉ cần mô tả trạng thái mong muốn, và React sẽ tự động quản lý quá trình render phù hợp với sự thay đổi của dữ liệu.

Ngoài ra, ReactJS có một hệ sinh thái phong phú với nhiều công cụ hỗ trợ như **Redux** để quản lý trạng thái, **React Router** để định tuyến, cũng như các công cụ build như **Create React App** giúp khởi tạo và phát triển dự án một cách nhanh chóng. Nhờ đó, ReactJS đã trở thành một lựa chọn hàng đầu cho việc xây dựng các ứng dụng web hiện đại với giao diện tương tác cao và khả năng mở rộng linh hoạt.

Ưu điểm của React

- ReactJS cực kì hiệu quả: ReactJS tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp ReactJS tránh những thao tác cần trên DOM mà nhiều chi phí.
- ReactJS giúp việc viết các đoạn code JS dễ dàng hơn: Nó dùng cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của ReactJS. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
- Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO (Tối ưu hóa công cụ tìm kiếm) và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về
- Các mẫu thành phần và dữ liệu cải thiện khả năng đọc, giúp duy trì các ứng dụng lớn hơn.

1.5.4. Giới thiệu MongoDB [5]



Hình 1.3. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, được thiết kế theo mô hình hướng đối tượng, cho phép lưu trữ dữ liệu theo cách linh hoạt mà không bắt buộc phải tuân theo một cấu trúc bảng cố định như trong các hệ thống RDBMS truyền thống. Nhờ vào khả năng lưu trữ dưới dạng tài liệu (document), MongoDB có thể dễ dàng xử lý các dữ liệu có cấu trúc phức tạp, đa dạng và không cố định – điều rất phù hợp với yêu cầu của các ứng dụng Big Data.

Các lợi thế của MongoDB so với các cơ sở dữ liệu quan hệ (RDBMS):

- **Ít phụ thuộc vào Schema:**
MongoDB lưu trữ dữ liệu dưới dạng tài liệu trong các Collection, trong đó mỗi tài liệu có thể có số lượng trường, nội dung và kích thước khác nhau. Điều này giúp giảm bớt sự phụ thuộc vào một Schema cứng nhắc như trong RDBMS.
- **Cấu trúc đối tượng rõ ràng:**
Dữ liệu được lưu dưới dạng JSON (hoặc BSON) cho phép mô tả cấu trúc của đối tượng một cách trực quan và rõ ràng.
- **Tránh các phép JOIN phức tạp:**
Không giống như các hệ quản trị dữ liệu quan hệ, MongoDB không yêu cầu sử dụng các phép JOIN giữa các bảng, giúp tối ưu hiệu suất xử lý các truy vấn phức tạp.
- **Khả năng truy vấn mạnh mẽ:**
MongoDB hỗ trợ một ngôn ngữ truy vấn linh hoạt dựa trên Document, cho phép thực hiện các truy vấn động và sâu trên dữ liệu tương đương với sức mạnh của SQL.
- **Dễ dàng mở rộng:**
Thiết kế của MongoDB cho phép mở rộng hệ thống (scaling) một cách linh hoạt, phù hợp với các ứng dụng cần xử lý khối lượng dữ liệu lớn.
- **Không cần chuyển đổi dữ liệu phức tạp:**
Việc ánh xạ dữ liệu từ đối tượng ứng dụng đến cấu trúc lưu trữ trong cơ sở dữ liệu trở nên đơn giản, do dữ liệu được lưu trữ ở dạng gần giống với cấu trúc trong ứng dụng.
- **Hiệu suất cao với bộ nhớ nội tại:**
MongoDB sử dụng bộ nhớ để lưu trữ tạm thời các phần công việc, giúp cải thiện tốc độ truy cập dữ liệu và tối ưu hiệu suất xử lý.

Nhờ những ưu điểm vượt trội này, MongoDB đã trở thành một lựa chọn hàng đầu cho việc xây dựng các ứng dụng cần lưu trữ và xử lý dữ liệu phức tạp, đa dạng và khối lượng lớn, đáp ứng hiệu quả các yêu cầu của các hệ thống hiện đại.

1.5.6. Giới thiệu Firebase



Hình 1.4. Firebase

Firebase là một nền tảng phát triển ứng dụng toàn diện được cung cấp bởi Google, giúp các nhà phát triển xây dựng và vận hành các ứng dụng web và di động một cách nhanh chóng và hiệu quả. Với Firebase, bạn không chỉ có được các dịch vụ backend mạnh mẽ mà còn được hỗ trợ các công cụ phân tích, giám sát và tối ưu hóa trải nghiệm người dùng.

Các tính năng nổi bật của Firebase bao gồm:

- **Realtime Database & Cloud Firestore:**
Cho phép lưu trữ và đồng bộ hóa dữ liệu theo thời gian thực, đảm bảo ứng dụng luôn cập nhật với dữ liệu mới nhất mà không cần phải thực hiện các thao tác refresh thủ công.
- **Authentication:**
Hỗ trợ nhiều phương thức đăng nhập như email/password, xác thực qua mạng xã hội (Google, Facebook, Twitter, v.v.), giúp quản lý và bảo mật người dùng một cách hiệu quả.
- **Cloud Storage:**
Cung cấp giải pháp lưu trữ file an toàn và có khả năng mở rộng, thích hợp cho việc lưu trữ hình ảnh, video và các tài liệu khác.
- **Hosting:**
Dịch vụ lưu trữ web tĩnh với hiệu suất cao và chứng chỉ SSL miễn phí, giúp triển khai ứng dụng nhanh chóng và bảo mật.

- **Cloud Functions:**

Cho phép chạy các đoạn mã backend trong môi trường serverless, xử lý logic nghiệp vụ mà không cần lo lắng về việc quản lý máy chủ.

- **Analytics & Crashlytics:**

Công cụ phân tích hành vi người dùng và giám sát hiệu năng ứng dụng, giúp phát hiện và khắc phục lỗi, cải thiện trải nghiệm người dùng.

Nhờ những tính năng vượt trội này, Firebase giúp giảm thiểu thời gian phát triển và gánh nặng quản lý hạ tầng backend, cho phép các nhà phát triển tập trung vào việc xây dựng tính năng cho ứng dụng và nâng cao trải nghiệm người dùng.

1.5. Kết luận

Chương 1 đã trình bày tổng quan về đề tài “Hệ thống tính điểm GPA kết hợp lưu trữ bài giảng” cùng với việc giới thiệu các công nghệ chính được sử dụng để xây dựng và phát triển hệ thống, bao gồm Node.js, ExpressJS, ReactJS, MongoDB và Firebase. Nội dung Chương 2 sẽ đi sâu vào phân tích và thiết kế hệ thống dựa trên các giải pháp đã được đề xuất, cụ thể:

- **Phân tích yêu cầu và xác định chức năng:**

Trình bày chi tiết các yêu cầu của hệ thống từ các đối tượng người dùng (Sinh viên, Giảng viên, Quản trị viên), từ đó xác định các chức năng chính như tính điểm GPA tự động, lưu trữ và truy cập bài giảng, quản lý hồ sơ người dùng, cũng như các tính năng quản trị.

- **Thiết kế kiến trúc hệ thống:**

Mô tả kiến trúc tổng thể của hệ thống, bao gồm sơ đồ luồng dữ liệu, mô hình các thành phần backend và frontend, cách thức giao tiếp giữa các module thông qua API và các giải pháp lưu trữ dữ liệu.

- **Thiết kế cơ sở dữ liệu và lưu trữ:**

Phân tích và thiết kế cấu trúc cơ sở dữ liệu MongoDB cho việc lưu trữ thông tin điểm số, bài giảng và dữ liệu người dùng, cũng như tích hợp Firebase Storage cho việc lưu trữ các tài liệu học tập.

Qua Chương 2, hệ thống sẽ được định hình một cách rõ ràng về mặt logic và kỹ thuật, tạo nền tảng vững chắc cho việc triển khai các chức năng theo yêu cầu của đề tài.

