

Bài: [Live Stream] Các công việc của ngành lập trình và hành trang cần thiết khi xin việc IT

Xem bài học trên website để ủng hộ Kteam: [\[Live Stream\] Các công việc của ngành lập trình và hành trang cần thiết khi xin việc IT](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Là một trong những buổi tâm huyết được chia sẻ bởi anh **Kim Long** - Founder dự án [Howkteam.vn](#) và là tác giả nhiều khóa học về lập trình.

Trong buổi này, Kteam và bạn sẽ cùng nhau tìm hiểu về các ngành nghề trong lập trình cũng như những hướng phát triển mà các bạn có thể định hướng trong tương lai.

Bạn sẽ được giải đáp những câu hỏi:

- Ngành lập trình là làm gì? Có những công việc nào trong ngành lập trình?
- Các hướng phát triển trong ngành.
- Lộ trình học lập trình từ chưa biết gì?
- Các kỹ năng cần có để tìm kiếm cơ hội việc làm trong ngành
- Cùng rất nhiều chia sẻ thực tế từ trải nghiệm của người trong ngành.

Thời gian diễn ra

Thời gian: 20h00 – 08/10/2023

Channel Youtube: [Kteam](#)

LiveStream diễn ra vào Chủ nhật hàng tuần. Nhớ đón xem nội dung qua bài biên tập tại [howkteam.com](#) > [Lập trình](#) > [Những điều trường học không dạy](#).

Trong phần đầu tiên, chúng ta sẽ cùng nhau làm quen với một số ngành nghề nổi bật trong lập trình

Phát triển phần mềm ứng dụng:

Xác định mục tiêu và yêu cầu: Xác định mục tiêu của trang web, đối tượng mục tiêu, và yêu cầu chức năng cần có.

Lập kế hoạch (Planning):

- Xây dựng kế hoạch tổ chức, xác định cấu trúc trang web, lựa chọn nền tảng phù hợp.
- Tạo sơ đồ trang web (sitemaps) để xác định cấu trúc và luồng dẫn hướng của trang web.

Thiết kế (Design):

- Thiết kế giao diện người dùng (UI/UX design) với sự tập trung vào trải nghiệm người dùng.
- Xây dựng wireframes, thiết kế mô hình và lựa chọn màu sắc, font chữ, hình ảnh, biểu tượng, ...

Phát triển (Development):

- Lựa chọn công nghệ và ngôn ngữ lập trình phù hợp với yêu cầu của dự án.
- Xây dựng và lập trình tính năng, giao diện, tối ưu mã nguồn, và tương tác với cơ sở dữ liệu.

Kiểm thử (Testing):

- Thử nghiệm và kiểm tra trang web trên các trình duyệt và thiết bị khác nhau.

- Kiểm tra tính năng, giao diện, hiệu suất, bảo mật và tương thích.

Triển khai (Deployment):

- Đưa trang web vào môi trường sản xuất, cấu hình máy chủ và tối ưu hoá hiệu suất.
- Xác định và xử lý lỗi, chuẩn bị cho phép truy cập công chúng.

Bảo trì và cập nhật (Maintenance and Updates):

- Duy trì và quản lý trang web sau khi triển khai.
- Cập nhật nội dung, bảo mật, hiệu suất, và tính năng mới.

Phát triển ứng dụng di động (Mobile App Development):

Phát triển ứng dụng di động (Mobile App Development) là quá trình tạo ra ứng dụng phần mềm dành cho thiết bị di động như điện thoại di động, máy tính bảng và các thiết bị thông minh khác. Đây là một lĩnh vực công nghệ ngày càng phát triển nhanh chóng, đáng chú ý với sự gia tăng sử dụng thiết bị di động trên toàn cầu.

Ngôn ngữ lập trình:

- Java và Kotlin: Được sử dụng cho phát triển ứng dụng Android.
- Swift và Objective-C: Được sử dụng cho phát triển ứng dụng iOS.

Công cụ phát triển (Development Tools):

- Android Studio: Phát triển ứng dụng Android.
- Xcode: Phát triển ứng dụng iOS.

Công nghệ đa nền tảng (Cross-Platform Technologies):

- React Native: Cho phát triển ứng dụng đa nền tảng với JavaScript.
- Flutter: Khung làm việc của Google cho phát triển ứng dụng đa nền tảng.
- Cơ sở dữ liệu (Database):
 - SQLite, Firebase: Dùng cho cơ sở dữ liệu nhỏ và dễ triển khai.
 - MySQL, PostgreSQL, MongoDB: Dùng cho ứng dụng yêu cầu quy mô lớn.

Kỹ thuật responsive design và UI frameworks:

- Bootstrap, Foundation: Frameworks hỗ trợ xây dựng giao diện đáp ứng.
- CSS, HTML5: Các kỹ thuật phổ biến để tạo giao diện người dùng.

Tổng kết:

Phát triển ứng dụng di động đòi hỏi kỹ thuật cao cũng như hiểu biết sâu về thiết bị di động và nền tảng phát triển. Các công nghệ đa nền tảng ngày càng phổ biến giúp giảm thời gian và tối ưu hóa quá trình phát triển. Quan trọng nhất, chất lượng và trải nghiệm người dùng cuối cùng là mục tiêu hàng đầu trong phát triển ứng dụng di động.

Phát triển web (Web Development):

Phân thành hai lĩnh vực chính: **Front-end** (người phát triển giao diện người dùng) và **Back-end** (người phát triển hệ thống và dữ liệu).

Các ngôn ngữ và framework phổ biến cho Front-end:

Ngôn ngữ (Languages):

- HTML (HyperText Markup Language): Ngôn ngữ cơ bản để tạo cấu trúc và định dạng nội dung trên trang web.

- CSS (Cascading Style Sheets): Ngôn ngữ dùng để thiết kế và trang trí giao diện người dùng, điều chỉnh kiểu, màu sắc, kích thước của các phần tử HTML.
- JavaScript: Ngôn ngữ lập trình phía client mạnh mẽ, được sử dụng để tạo các tương tác động và xử lý sự kiện trên trình duyệt.

Framework và thư viện (Frameworks and Libraries):

- React: Thư viện JavaScript phổ biến để xây dựng giao diện người dùng động và hiệu suất cao. Phát triển bởi Facebook.
- Angular: Framework JavaScript mạnh mẽ để xây dựng ứng dụng web phức tạp và đa tính năng. Phát triển bởi Google.
- Vue.js: Framework JavaScript linh hoạt và dễ sử dụng để xây dựng giao diện người dùng tương tác. Thiết kế tối giản và tập trung vào việc làm việc với các thành phần.
- Bootstrap: Framework CSS phổ biến để xây dựng giao diện người dùng responsive và hấp dẫn. Cung cấp các lớp và thành phần sẵn có để giúp tối ưu hóa quá trình thiết kế.
- Sass (Syntactically Awesome Stylesheets): Ngôn ngữ mở rộng CSS với nhiều tính năng hữu ích như biến, lồng, mixin. Giúp quản lý mã CSS hiệu quả hơn.
- LESS: Ngôn ngữ mở rộng CSS, tương tự Sass, giúp tăng cường khả năng phát triển và quản lý mã CSS.
- jQuery: Thư viện JavaScript giúp tương tác với DOM dễ dàng và giảm bớt mức độ phức tạp của mã JavaScript.
- D3.js: Thư viện JavaScript để tạo đồ thị, biểu đồ và visualizations phức tạp dựa trên dữ liệu.
- Material-UI: Thư viện React cung cấp các thành phần tuân thủ nguyên tắc thiết kế Material Design của Google.
- Foundation: Framework CSS cho thiết kế giao diện người dùng responsive và hỗ trợ nhanh chóng.

Các ngôn ngữ phổ biến cho Back-end:

- JavaScript (Node.js): Node.js là một môi trường chạy mã JavaScript phía máy chủ, cho phép xây dựng ứng dụng back-end hiệu suất cao và không đồng bộ.
Các framework phổ biến: Express.js, Koa.js, Nest.js.
- Python: Python là một ngôn ngữ linh hoạt và phổ biến được sử dụng rộng rãi cho phát triển back-end.
Các framework phổ biến: Django, Flask, Pyramid.
- Java: Java là một ngôn ngữ ổn định và mạnh mẽ, thường được sử dụng trong các ứng dụng doanh nghiệp và lớn.
Các framework phổ biến: Spring, Hibernate, Apache Struts.
- C# (ASP.NET Core): ASP.NET Core là một framework phát triển ứng dụng web mã nguồn mở được xây dựng trên nền tảng .NET.
Các framework phổ biến: ASP.NET Core.
- Ruby: Ruby là một ngôn ngữ dễ đọc, dễ viết và mạnh mẽ, thường được sử dụng cho phát triển web.
Các framework phổ biến: Ruby on Rails, Sinatra.
- PHP: PHP là một ngôn ngữ phổ biến cho phát triển web động, được tích hợp trực tiếp vào mã HTML.
Các framework phổ biến: Laravel, Symfony, CodeIgniter.
- Go (Golang): Go là một ngôn ngữ mới, được thiết kế cho hiệu suất cao và phát triển dự án lớn.
Các framework phổ biến: Gin, Echo.
- Rust: Rust là một ngôn ngữ an toàn và hiệu suất cao, được sử dụng ngày càng phổ biến cho phát triển back-end.
Các framework phổ biến: Actix, Rocket.
- Scala: Scala là một ngôn ngữ kết hợp các tính năng của lập trình hướng đối tượng và hàm.
Các framework phổ biến: Play Framework, Akka.

Phát triển phần mềm (Software Development):

Xây dựng các ứng dụng và chương trình máy tính đa dạng, từ ứng dụng desktop đến hệ thống quản lý dự án phức tạp.

Ngôn ngữ lập trình:

- C#: Thường được sử dụng cho phát triển trên nền tảng .NET.
- Java: Được sử dụng rộng rãi và chủ yếu trên nền tảng Java.
- C++: Sử dụng cho ứng dụng yêu cầu hiệu suất cao.

Công cụ phát triển (Development Tools):

- Visual Studio: Phổ biến cho phát triển ứng dụng trên nền tảng .NET.
- Eclipse, IntelliJ IDEA: Được sử dụng cho phát triển ứng dụng Java.

Cơ sở dữ liệu (Database):

- Microsoft SQL Server, MySQL, PostgreSQL: Các hệ quản trị cơ sở dữ liệu phổ biến.
- SQLite: Thích hợp cho các ứng dụng nhỏ và đơn giản.

Công cụ quản lý mã nguồn (Version Control):

- Git: Công cụ phổ biến để quản lý phiên bản mã nguồn.

Công cụ kiểm thử (Testing Tools):

- Selenium: Được sử dụng cho kiểm thử tự động.
- JUnit, NUnit: Các framework kiểm thử cho Java và .NET.

Tổng kết:

Phát triển phần mềm PC là một quy trình đa bước đòi hỏi kiến thức vững về lập trình và quy trình phát triển. Điều quan trọng nhất là tạo ra sản phẩm phần mềm có chất lượng, hiệu suất và đáp ứng được yêu cầu và mong đợi của người dùng cuối.

Phân tích dữ liệu và khoa học dữ liệu (Data Analysis and Data Science):

Phân tích dữ liệu (Data Analysis):

Phân tích dữ liệu là quá trình tách biệt, kiểm tra, và xem xét dữ liệu để tìm hiểu thông tin, tìm ra mô hình, và đưa ra những kết luận từ dữ liệu đã cho. Điều này thường bao gồm việc sử dụng các phương pháp và kỹ thuật thống kê.

Mục tiêu:

- Tìm hiểu đặc điểm, xu hướng, và mẫu trong dữ liệu.
- Xác định các thông tin quan trọng và học hỏi từ dữ liệu.
- Hỗ trợ quyết định và đưa ra các đề xuất dựa trên dữ liệu.

Phương pháp:

- Thống kê: Sử dụng các phương pháp thống kê để phân tích mối quan hệ và tính toán các chỉ số mô tả.
- Trực quan hóa dữ liệu: Sử dụng biểu đồ, đồ thị và các hình vẽ khác để hiển thị dữ liệu một cách rõ ràng và dễ hiểu.
- Mô hình hóa: Xây dựng mô hình dự đoán và phân loại dữ liệu.

Công cụ:

- Phần mềm phổ biến: Excel, Python (pandas, NumPy), R, MATLAB.

Khoa học dữ liệu (Data Science):

Khoa học dữ liệu là quá trình sử dụng các kỹ thuật và phương pháp phức tạp để tìm hiểu, khai phá, và rút ra thông tin từ dữ liệu lớn và phức tạp. Điều này đặc biệt chú trọng đến việc áp dụng machine learning và deep learning.

Mục tiêu:

- Dự đoán xu hướng và kỹ thuật dự đoán.
- Tìm hiểu cấu trúc ẩn trong dữ liệu.
- Tìm ra thông tin giá trị từ dữ liệu để hỗ trợ quyết định kinh doanh.

Phương pháp:

- Machine Learning: Sử dụng các thuật toán máy học để xây dựng mô hình dự đoán và phân loại.
- Deep Learning: Sử dụng mạng neural sâu để xử lý dữ liệu phức tạp và trích xuất thông tin.
- Xử lý Big Data: Xử lý và phân tích dữ liệu lớn và phức tạp sử dụng các công nghệ và kỹ thuật đặc biệt.

Công cụ:

- Ngôn ngữ lập trình: Python, R, SQL.
- Công cụ và framework: TensorFlow, PyTorch, scikit-learn, Hadoop, Spark.

Tổng kết:

Phân tích dữ liệu tập trung vào việc tìm hiểu và rút ra thông tin từ dữ liệu hiện có bằng cách sử dụng các phương pháp thống kê và trực quan hóa. Trong khi đó, khoa học dữ liệu tập trung vào việc áp dụng các kỹ thuật phức tạp, bao gồm machine learning và deep learning, để khám phá dữ liệu, xây dựng mô hình, và đưa ra dự đoán.

An ninh mạng và bảo mật (Cybersecurity):

An ninh mạng và bảo mật (Cybersecurity) là lĩnh vực chuyên về bảo vệ thông tin, hệ thống máy tính, dữ liệu, và các tài sản kỹ thuật số khỏi các mối đe dọa và tấn công mạng. Đây là một lĩnh vực ngày càng quan trọng vì sự phụ thuộc ngày càng lớn vào công nghệ thông tin và internet trong cuộc sống hàng ngày.

Mục tiêu của An ninh mạng và bảo mật:

Bảo vệ tài sản kỹ thuật số: Đảm bảo an toàn cho thông tin, dữ liệu quan trọng, ứng dụng và các tài nguyên kỹ thuật số khác.

Bảo vệ quyền riêng tư và quyền sở hữu: Bảo đảm quyền riêng tư của người dùng và sở hữu trí tuệ.

Bảo đảm tính sẵn sàng và khả dụng (availability): Đảm bảo rằng các hệ thống và dịch vụ có thể sử dụng một cách bình thường và hiệu quả mà không bị ngừng hoạt động.

Phòng ngừa và phản ứng: Ngăn ngừa các tấn công và xử lý hiệu quả khi các tấn công xảy ra.

Các lĩnh vực quan trọng trong An ninh mạng và bảo mật:

Bảo mật hệ thống và mạng: Bao gồm việc bảo vệ hệ điều hành, ứng dụng, dịch vụ mạng, và cơ sở hạ tầng mạng.

Bảo mật ứng dụng và phần mềm: Bảo vệ ứng dụng và phần mềm khỏi các lỗ hổng bảo mật và tấn công.

Quản lý danh tính và quyền truy cập: Kiểm soát quyền truy cập vào hệ thống và dữ liệu để đảm bảo chỉ những người được ủy quyền mới có thể truy cập.

Bảo mật dữ liệu và mã hóa: Bảo vệ dữ liệu quan trọng bằng cách sử dụng các thuật toán mã hóa mạnh mẽ.

Phòng ngừa và phản ứng tấn công (Incident Response): Xây dựng kế hoạch và quy trình để xử lý các sự cố bảo mật và tấn công một cách hiệu quả.

Thử thách và kiểm tra bảo mật (Penetration Testing): Kiểm tra và đánh giá sự bảo mật của hệ thống và ứng dụng bằng cách mô phỏng các cuộc tấn công.

An ninh mạng di động và IoT: Tập trung vào bảo vệ các thiết bị di động và thiết bị kết nối mạng thông qua Internet of Things (IoT).

Công cụ và kỹ thuật quan trọng trong An ninh mạng và bảo mật:

Firewall và Intrusion Detection/Prevention Systems (IDS/IPS): Giám sát và kiểm soát lưu lượng mạng để ngăn ngừa các tấn công.

Mã hóa dữ liệu (Encryption): Sử dụng mã hóa để bảo vệ dữ liệu quan trọng khỏi việc truy cập trái phép.

Authentication and Access Control Systems: Đảm bảo quyền truy cập được kiểm soát và đảm bảo tính xác thực của người dùng.

Vulnerability Assessment Tools: Công cụ đánh giá lỗ hổng để xác định điểm yếu trong hệ thống và ứng dụng.

Security Information and Event Management (SIEM): Theo dõi, phân tích và báo cáo về các sự kiện và thông tin bảo mật trên mạng.

Các kỹ thuật kiểm tra bảo mật (e.g., Penetration Testing, Ethical Hacking): Sử dụng các phương pháp kiểm thử để xác định lỗ hổng và kiểm tra tính bảo mật của hệ thống.

Công nghệ điều tra số (Digital Forensics): Phân tích và thu thập dữ liệu điện tử để tìm hiểu và xác định các hành vi phạm pháp. Các kỹ thuật và ngôn ngữ: Kỹ thuật hacking etical (ethical hacking), kiểm thử bảo mật, bảo mật mạng, Python.

Trí tuệ nhân tạo (Artificial Intelligence - AI) và Học sâu (Deep Learning):

Phát triển các ứng dụng và mô hình học sâu để giải quyết các vấn đề phức tạp.

Các ngôn ngữ và thư viện phổ biến: Python, TensorFlow, PyTorch, Keras.

Trò chơi điện tử (Game Development):

Phát triển trò chơi (Game Development) là quá trình tạo ra các trò chơi điện tử từ ý tưởng ban đầu đến sản phẩm hoàn chỉnh. Đây là một ngành công nghiệp phát triển nhanh chóng và đa dạng, liên quan đến nhiều lĩnh vực kỹ thuật và nghệ thuật.

Unity: Phổ biến, hỗ trợ nhiều nền tảng và dễ học.

Unreal Engine: Mạnh mẽ về đồ họa và hỗ trợ nhiều nền tảng.

Cocos Creator: Đặc biệt cho phát triển trò chơi di động.

Ngôn ngữ lập trình:

- C#: Phổ biến cho Unity.
- C++: Thường được sử dụng cho Unreal Engine.
- Python: Được sử dụng cho phát triển gameplay và công cụ.

Công cụ thiết kế (Design Tools):

- Adobe Photoshop, Illustrator, Blender, 3ds Max, Maya: Được sử dụng để tạo đồ họa và mô hình.
- Audacity, Fmod, Wwise: Được sử dụng để tạo âm thanh và âm nhạc.

Thư viện và khung phát triển (Frameworks):

- Phaser: Cho phát triển trò chơi 2D.
- LibGDX: Hỗ trợ phát triển đa nền tảng (desktop, mobile, web).
- Pygame: Dành cho phát triển trò chơi sử dụng Python.

Công cụ quản lý dự án và version control:

- Git: Quản lý phiên bản và tương tác đồng thời giữa các thành viên trong nhóm phát triển.
- Jira, Trello: Quản lý dự án và tiến độ phát triển.

Tổng kết:

Phát triển trò chơi là một quy trình đa dạng và phức tạp, bao gồm các bước từ ý tưởng ban đầu đến việc triển khai sản phẩm hoàn chỉnh. Các công cụ và nền tảng chơi vai trò quan trọng trong việc xây dựng và quản lý quá trình phát triển trò chơi, trong khi quản lý dự án và kiến thức về thiết kế và lập trình cũng là yếu tố then chốt trong thành công của một dự án phát triển trò chơi.

Quản lý dự án phần mềm (Software Project Management):

Ngành quản lý dự án phần mềm (Software Project Management) là một vai trò quan trọng trong lĩnh vực công nghệ thông tin, nhằm đảm bảo việc phát triển, triển khai, và quản lý các dự án phần mềm được thực hiện một cách hiệu quả và hiệu suất cao. Quản lý dự án phần mềm đòi hỏi kiến thức sâu về công nghệ thông tin, quy trình phát triển phần mềm, và kỹ năng lãnh đạo.

Vai trò và trách nhiệm chính của một quản lý dự án phần mềm:

Quy hoạch và lập kế hoạch (Planning and Scheduling):

- Xác định mục tiêu dự án, tài nguyên cần thiết và lịch trình thực hiện.
- Tạo kế hoạch công việc và xác định các bước tiến trình cần thiết để hoàn thành dự án.

Quản lý tài nguyên (Resource Management):

- Quản lý tài nguyên như con người, vật liệu, thiết bị cần thiết cho dự án.
- Xác định nhu cầu tài nguyên và đảm bảo rằng chúng đủ để hoàn thành dự án theo lịch trình.

Quản lý rủi ro (Risk Management):

- Xác định và đánh giá các rủi ro có thể xảy ra trong quá trình phát triển và triển khai.
- Phát triển kế hoạch để giảm thiểu và ứng phó với các rủi ro.

Quản lý nhóm và lãnh đạo (Team Management and Leadership):

- Xây dựng và quản lý đội ngũ phát triển, tạo môi trường làm việc tích cực.
- Lãnh đạo và hỗ trợ thành viên trong đội để đạt được mục tiêu dự án.

Theo dõi tiến độ (Monitoring Progress):

- Theo dõi và đánh giá tiến độ thực hiện dự án, đảm bảo tuân thủ lịch trình và mục tiêu.
- Đưa ra các điều chỉnh và cải thiện dựa trên thông tin thu được.

Quản lý chất lượng (Quality Management):

- Đảm bảo rằng sản phẩm hoàn thành đáp ứng chất lượng đã quy định.
- Áp dụng các phương pháp kiểm thử và kiểm tra để đảm bảo tính chính xác và tin cậy của sản phẩm.

Giao tiếp và tương tác (Communication and Stakeholder Management):

- Xây dựng và duy trì một quan hệ tốt với khách hàng, bên ngoài và các bên liên quan khác.
- Báo cáo tiến độ và thông tin dự án một cách hiệu quả.

Kỹ năng và đặc điểm quan trọng của một quản lý dự án phần mềm:

Kỹ năng quản lý: Kiến thức về các phương pháp và kỹ thuật quản lý dự án, quản lý thời gian, quản lý nguồn lực, quản lý rủi ro, và quản lý chất lượng.

Kỹ năng giao tiếp và lãnh đạo: Khả năng giao tiếp hiệu quả, lắng nghe, tạo động lực cho đội ngũ, và xây dựng môi trường làm việc tích cực.

Kiến thức về công nghệ thông tin: Hiểu biết về quy trình phát triển phần mềm, ngôn ngữ lập trình, cơ sở dữ liệu, kiến thức về hệ thống và môi trường phát triển.

Tư duy logic và quyết định: Có khả năng phân tích vấn đề, tìm ra giải pháp và đưa ra quyết định đúng đắn trong tình huống khó khăn.

Linh hoạt và kiên trì: Linh hoạt trong tư duy, sẵn sàng thích ứng với sự thay đổi và kiên trì để đạt được mục tiêu dự án.

Khả năng làm việc nhóm: Có khả năng làm việc cùng đội ngũ, xây dựng môi trường hợp tác và đóng góp vào sự phát triển của đồng đội.

Kỹ năng quản lý xung đột: Có khả năng giải quyết xung đột và vấn đề nội bộ một cách hiệu quả.

Quản lý dự án phần mềm đóng vai trò quan trọng trong việc đảm bảo rằng dự án được thực hiện một cách hiệu quả, đáp ứng yêu cầu của khách hàng và đạt được mục tiêu đã đề ra.

Công cụ và phương pháp: Agile, Scrum, Kanban, Jira.

Kỹ năng cần thiết để có việc làm lập trình

1. Kiến thức về ngôn ngữ lập trình và công nghệ:

Hiểu biết sâu về ít nhất một ngôn ngữ lập trình chính và có kiến thức cơ bản về một số ngôn ngữ khác.

Nắm vững các khái niệm cơ bản và nâng cao của lập trình như cấu trúc dữ liệu, thuật toán, design patterns, etc.

Hiểu biết về các công nghệ và framework liên quan đến lĩnh vực bạn muốn làm việc.

Đây là kỹ năng cần thiết nhất vì nhà tuyển dụng cần bạn làm được công việc họ giao. Và họ tuyển dụng bạn vì bạn có kỹ năng này mà họ cần.

2. Kỹ năng xử lý lỗi và debugging:

Khả năng xác định và sửa lỗi hiệu quả.

Hiểu biết về quy trình debugging và sử dụng các công cụ hỗ trợ.

Kỹ năng này cực kỳ quan trọng cho lập trình viên để có thể giải quyết vấn đề trong cả lập trình lẫn cuộc sống thường ngày.

3. Kỹ năng quản lý mã nguồn (Version Control):

Hiểu và sử dụng các hệ thống quản lý mã nguồn như Git.

Có kỹ năng làm việc với nhóm và giải quyết xung đột mã nguồn.

Không chỉ có làm việc nhóm mới cần quản lý mã nguồn. Ngay cả làm freelancer cá nhân cũng cần quản lý source code và version tốt để tránh rủi ro như hư hỏng ổ cứng, lỡ tay xóa code, lỡ tay ghi đè lên code.

4. Kiến thức về cơ sở dữ liệu:

Hiểu biết về cơ sở dữ liệu, các loại cơ sở dữ liệu khác nhau và truy vấn dữ liệu.

Kỹ năng làm việc với hệ quản trị cơ sở dữ liệu (Database Management Systems) như MySQL, PostgreSQL, MongoDB, etc.

Bạn không nhất thiết phải biết tất cả hệ quản trị cơ sở dữ liệu nhưng cần biết 1 trong những CSDL này nhằm hỗ trợ cho ngôn ngữ lập trình chính của bạn. Sau này bạn sẽ phát triển lên tư duy phát triển phần mềm.

5. Kỹ năng tư duy logic và phân tích:

Khả năng phân tích vấn đề và đưa ra giải pháp logic.

Tư duy hệ thống, có khả năng xác định yêu cầu, thiết kế và triển khai.

Để giải quyết được 1 bài toán bất kỳ bạn phải có khả năng chia nhỏ bài toán để giải quyết từng phần nhỏ sau đó bài toán tổng thể sẽ được giải quyết.

6. Kỹ năng giải quyết vấn đề:

Khả năng xác định và giải quyết vấn đề một cách hiệu quả và sáng tạo.

Hiểu biết về quy trình giải quyết vấn đề và áp dụng nó vào công việc hàng ngày.

7. Kỹ năng làm việc nhóm:

Có khả năng làm việc hiệu quả trong môi trường nhóm.

Kỹ năng giao tiếp, lắng nghe và tương tác xã hội tốt.

Dù có làm freelancer hay không thì kỹ năng làm việc nhóm vẫn luôn cần thiết. Đôi khi có những công việc cần kết hợp nhiều nền tảng, kỹ thuật, ngôn ngữ lập trình và kỹ thuật lập trình chuyên sâu khác nhau. Lúc này chúng ta sẽ cần hỗ trợ từ nhiều người khác.

8. Kỹ năng quản lý dự án:

Hiểu và áp dụng các nguyên tắc quản lý dự án.

Xây dựng và duy trì kế hoạch dự án, quản lý tài nguyên và thời gian một cách hiệu quả.

Tự bản thân quản lý tốt dự án cá nhân là bước đi đầu tiên trong việc quản lý dự án lớn hơn, 1 nhóm, 1 Cty. Quản lý dự án tốt cũng giúp tiết kiệm nhiều chi phí bảo trì, phát triển, chuyển giao. Cũng như có sự chuyên nghiệp trong công việc.

9. Kỹ năng tự học và cập nhật:

Khả năng học và tiếp nhận kiến thức mới nhanh chóng.

Luôn cập nhật và nâng cao kiến thức về công nghệ mới và xu hướng ngành.

Ngành công nghệ thông tin là 1 ngành phát triển thần tốc. Kỹ năng tự học là cực kỳ cần thiết để không bị bỏ lại phía sau.

10. Kỹ năng viết tài liệu và giao tiếp:

Khả năng viết tài liệu kỹ thuật rõ ràng và chi tiết.

Kỹ năng giao tiếp hiệu quả với đồng đội và khách hàng.

Bất kỳ bước bàn giao dự án nào cũng sẽ cần tài liệu bàn giao, thiết kế dự án cũng cần tài liệu thiết kế. Nên việc soạn tài liệu cũng rất cần thiết để giúp chúng ta hoàn thiện dự án, hiểu được dự án sau lâu ngày quay lại hay bàn giao cho đồng nghiệp, Cty, khách hàng. Cũng là cách để chuyên nghiệp hóa công việc của chúng ta.

Tóm lại:

Làm việc trong lĩnh vực lập trình đòi hỏi sự kết hợp giữa kiến thức kỹ thuật và các kỹ năng mềm quan trọng. Hãy liên tục nâng cao kỹ năng của bạn, tìm kiếm cơ hội học hỏi mới và tham gia vào các dự án thực tế để phát triển sự nghiệp của mình trong ngành công nghiệp lập trình.

Sản phẩm nên có khi đi xin việc(URL)

Kết luận

Bài biên tập chỉ tóm gọn nội dung để bạn dễ theo dõi, mọi kinh nghiệm được trao đổi & chia sẻ, chém gió hết sức thú vị chi tiết trong Video Live Stream. Nên bạn đừng lười coi video nhé!

Mọi nội dung luôn được Kteam cố gắng truyền tải một cách chi tiết và thực tế nhất. Tuy nhiên, không tránh được góc nhìn 1 phía từ trải nghiệm cá nhân, nếu bạn thấy vấn đề nào đó chưa đủ chính xác hay còn nhiều nội dung cần thảo luận thì đừng ngại để lại bình luận trao đổi, chia sẻ kinh nghiệm hiểu biết cá nhân với cộng đồng Kter nhé!

[Kteam](#) luôn chào đón mọi câu hỏi, trao đổi, bình luận góp ý từ phía các bạn để định hướng và phát triển bài viết tốt hơn. Đừng quên tham gia Live stream hàng tuần tại [Channel Youtube của Kteam](#) nhé! Cảm ơn bạn đã theo dõi!

