

IT-IUH

Designing Web Applications

Presented by Group 2

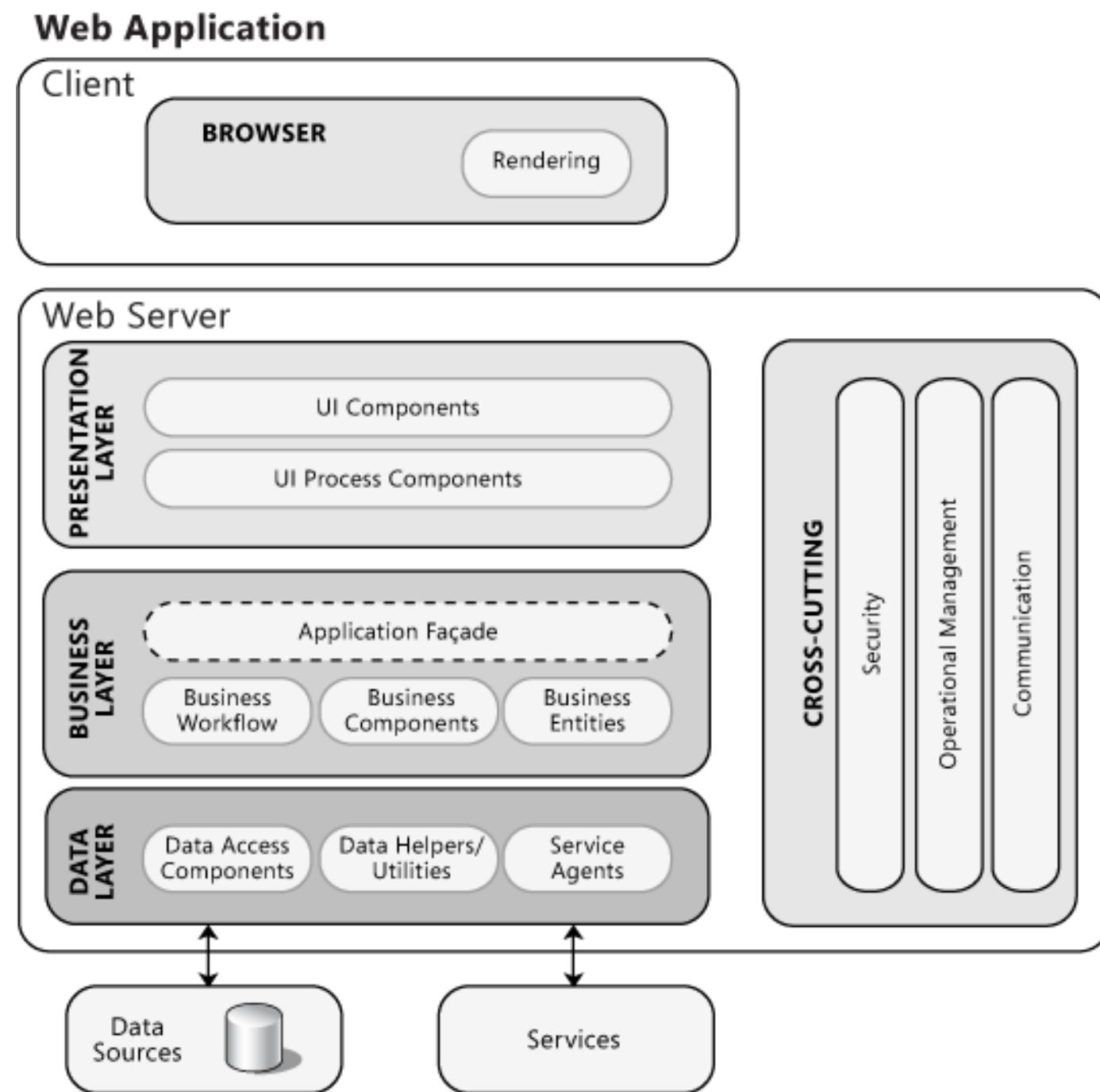
Nội dung

- 01 **Kiến trúc ứng dụng web là gì?**
- 02 **Kiến trúc ứng dụng web hoạt động như thế nào?**
- 03 **Các lớp của kiến trúc ứng dụng web**
- 04 **Các thành phần của ứng dụng web**
- 05 **Các mô hình ứng dụng web**
- 06 **Các loại kiến trúc ứng dụng web**
- 07 **Một số lưu ý khi thiết kế ứng dụng web**

I. Kiến trúc ứng dụng web là gì?



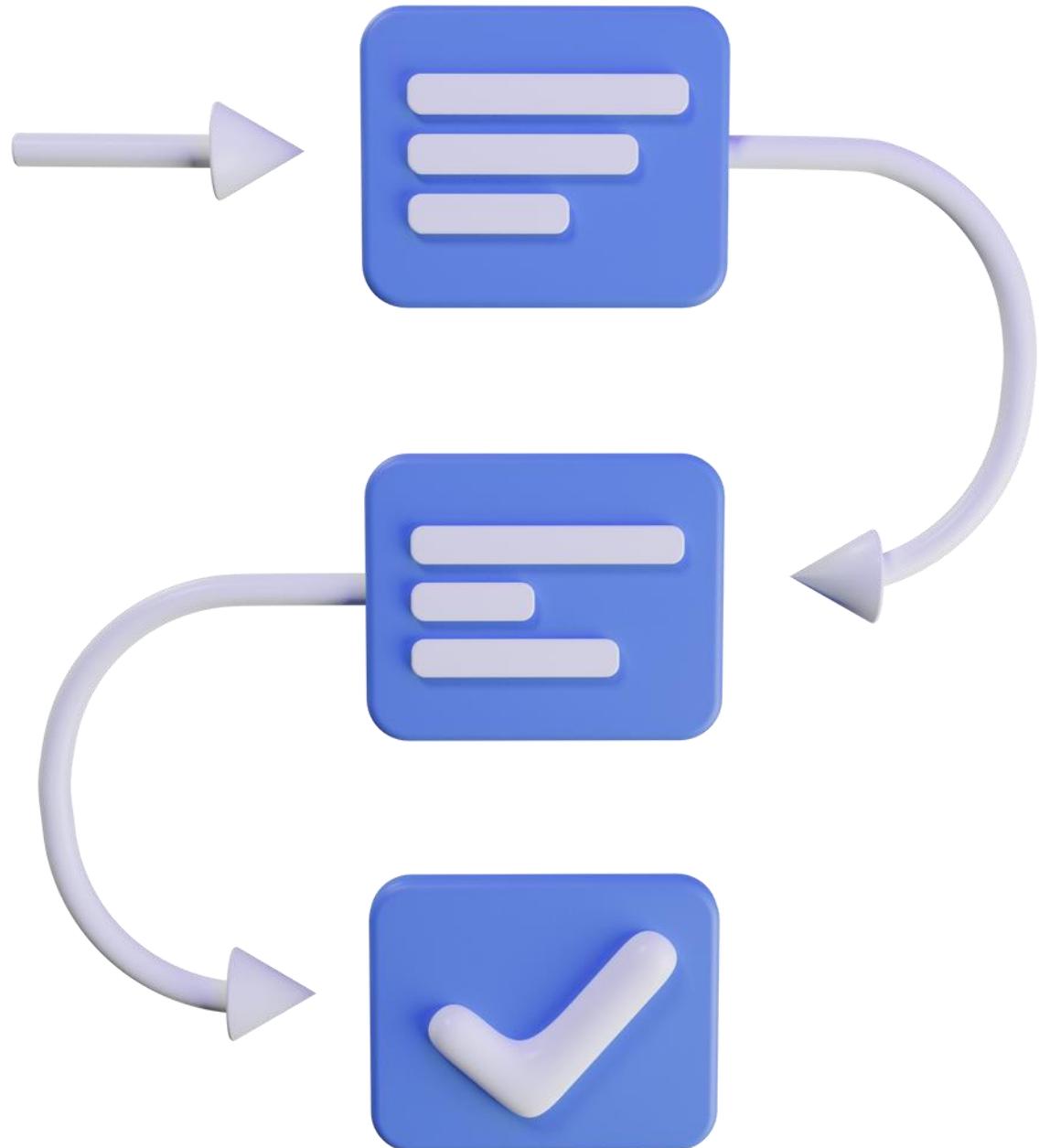
- Là bản thiết kế tương tác đồng thời giữa các thành phần, cơ sở dữ liệu, hệ thống phần mềm trung gian, giao diện người dùng và máy chủ trong một ứng dụng.
- Kiến trúc ứng dụng web cũng có thể được mô tả như một bố cục xác định hợp lý các kết nối giữa phía máy chủ và máy khách để người dùng có trải nghiệm tốt hơn.



Hình 1: Sơ đồ tổng thể kiến trúc ứng dụng web



Tại sao kiến trúc ứng dụng web quan trọng?



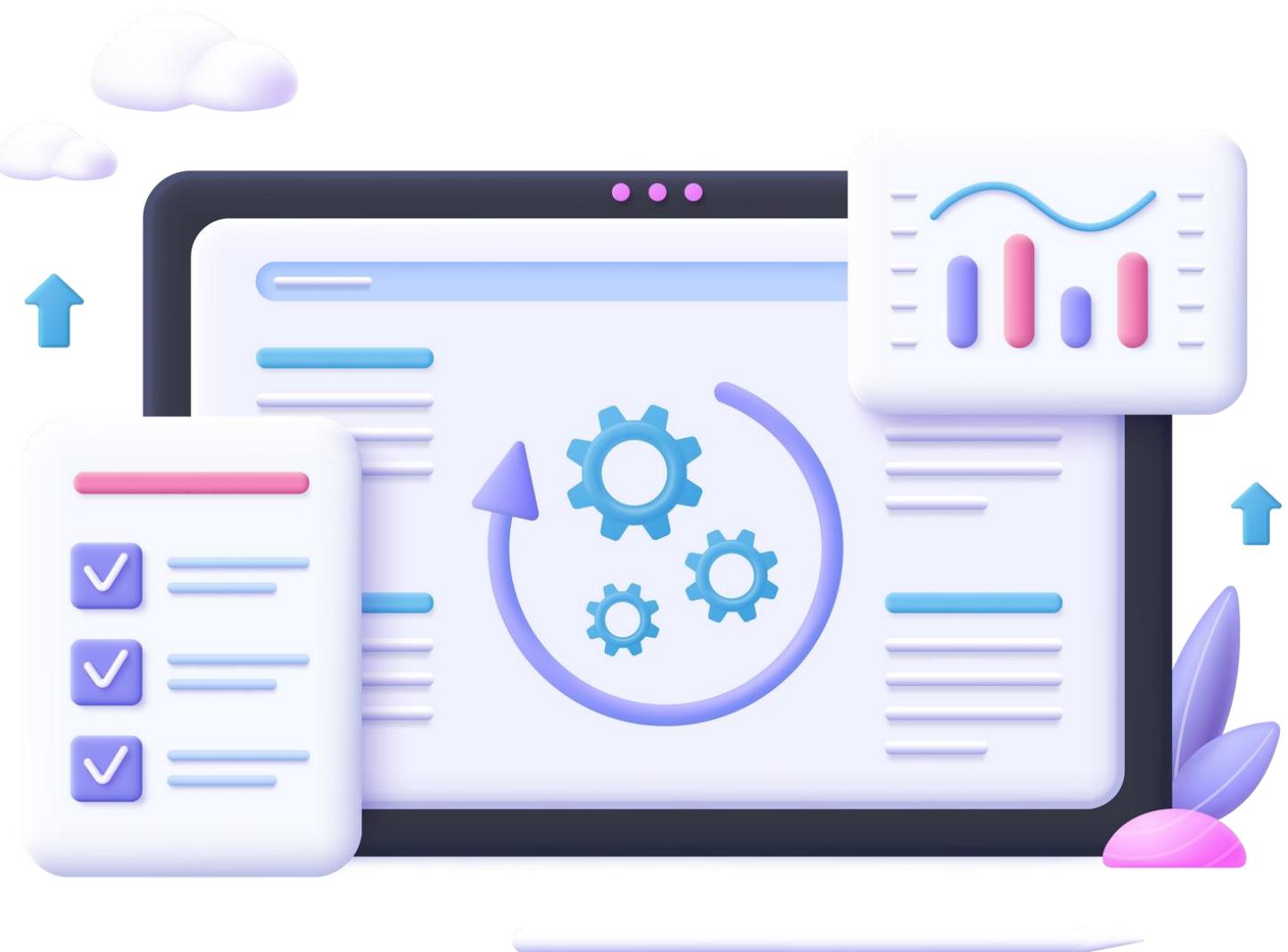
- Một ứng dụng web cần một kiến trúc để đặt nền tảng vững chắc.
- Một kiến trúc ứng dụng web tốt có thể xử lý các tác vụ khác nhau và thích ứng với các yêu cầu nghiệp vụ thay đổi một cách linh hoạt.
- Kiến trúc ứng dụng web chia ứng dụng thành các khối riêng biệt để giảm thiểu các mối đe dọa bảo mật, bao gồm cả rủi ro mã độc.

2. Kiến trúc ứng dụng web hoạt động như thế nào?



Client-side

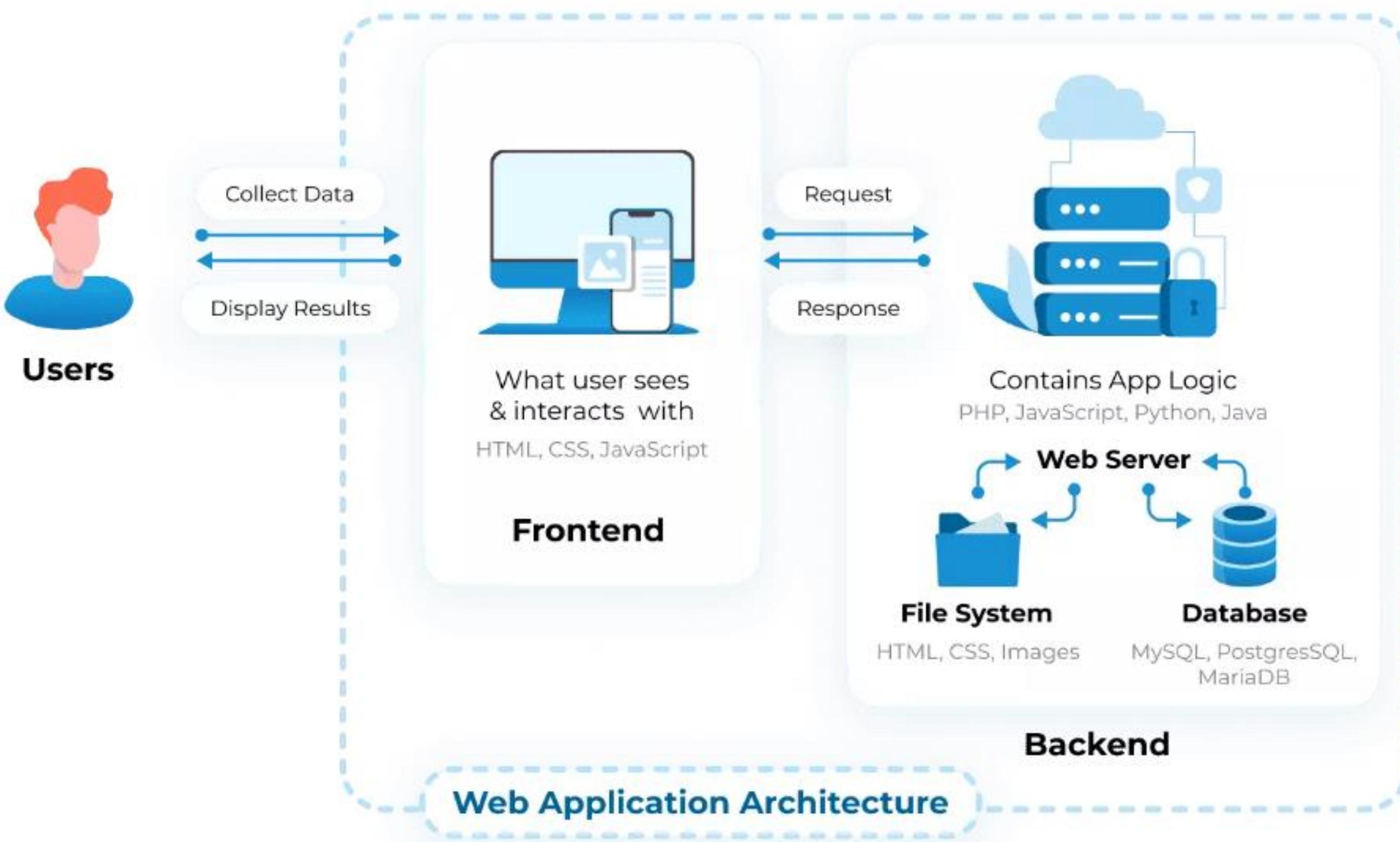
- Thường được gọi là Front-End.
- Được viết bằng HTML, CSS, JS và
được lưu trữ trên trình duyệt.





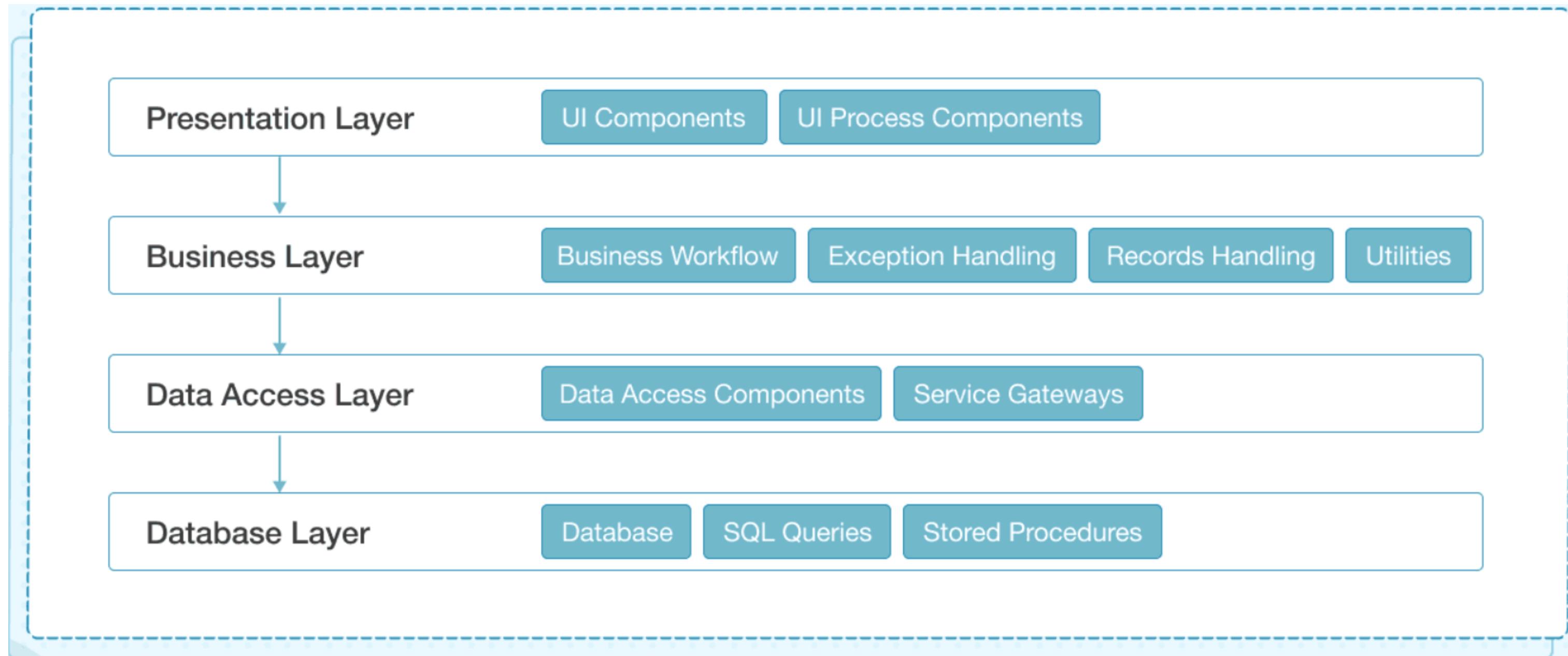
Server-side

- Thường được gọi là Back-End.
- Điều khiển các logic nghiệp vụ và phản hồi các yêu cầu http.
- Được viết bằng Java, PHP, Ruby, Python.



Hình 3: Sơ đồ hoạt động của ứng dụng web

3. Các lớp của kiến trúc ứng dụng web



Hình 4: Sơ đồ kiến trúc lớp ứng dụng web

4. Các thành phần của ứng dụng web



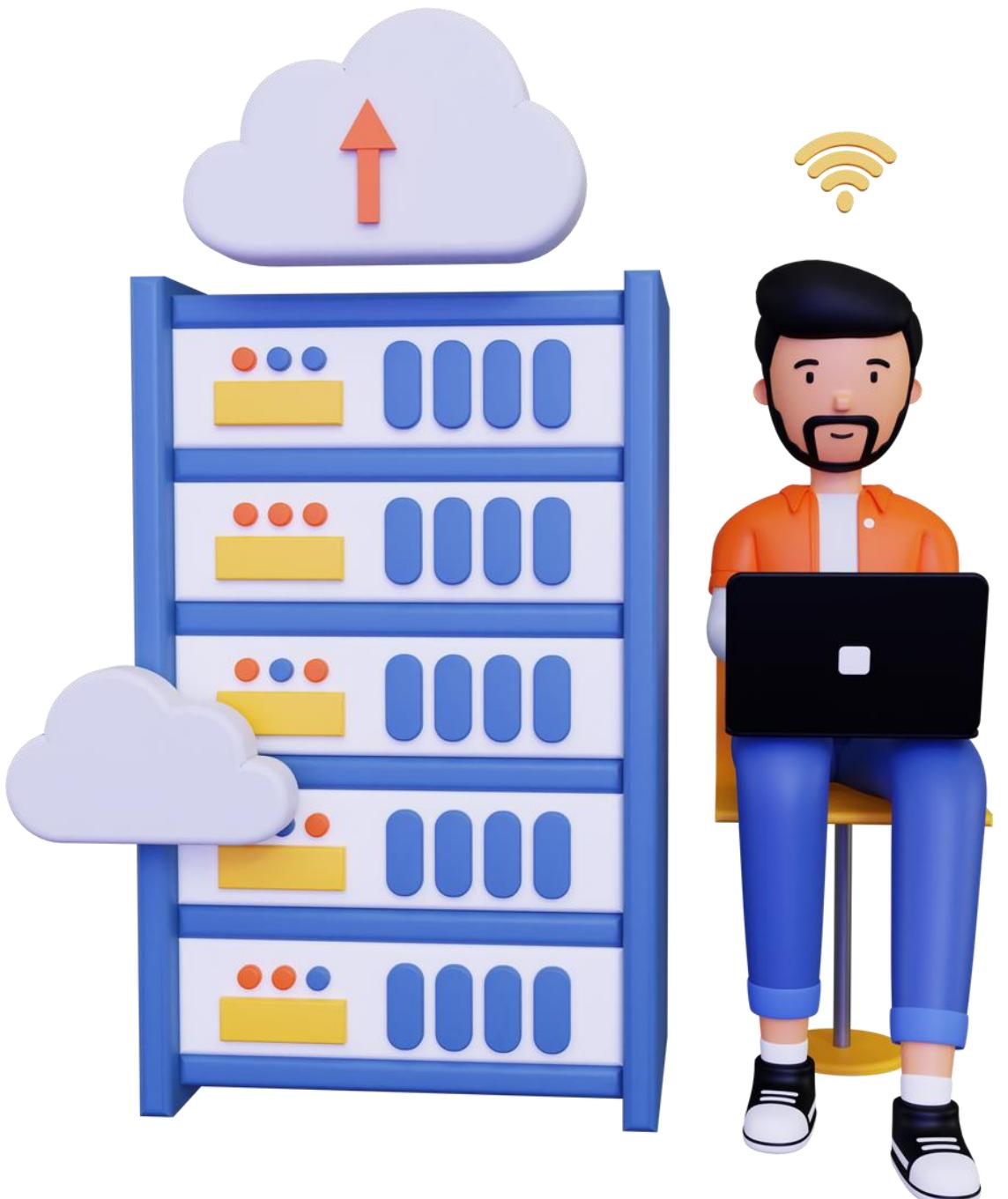


Giao diện người dùng

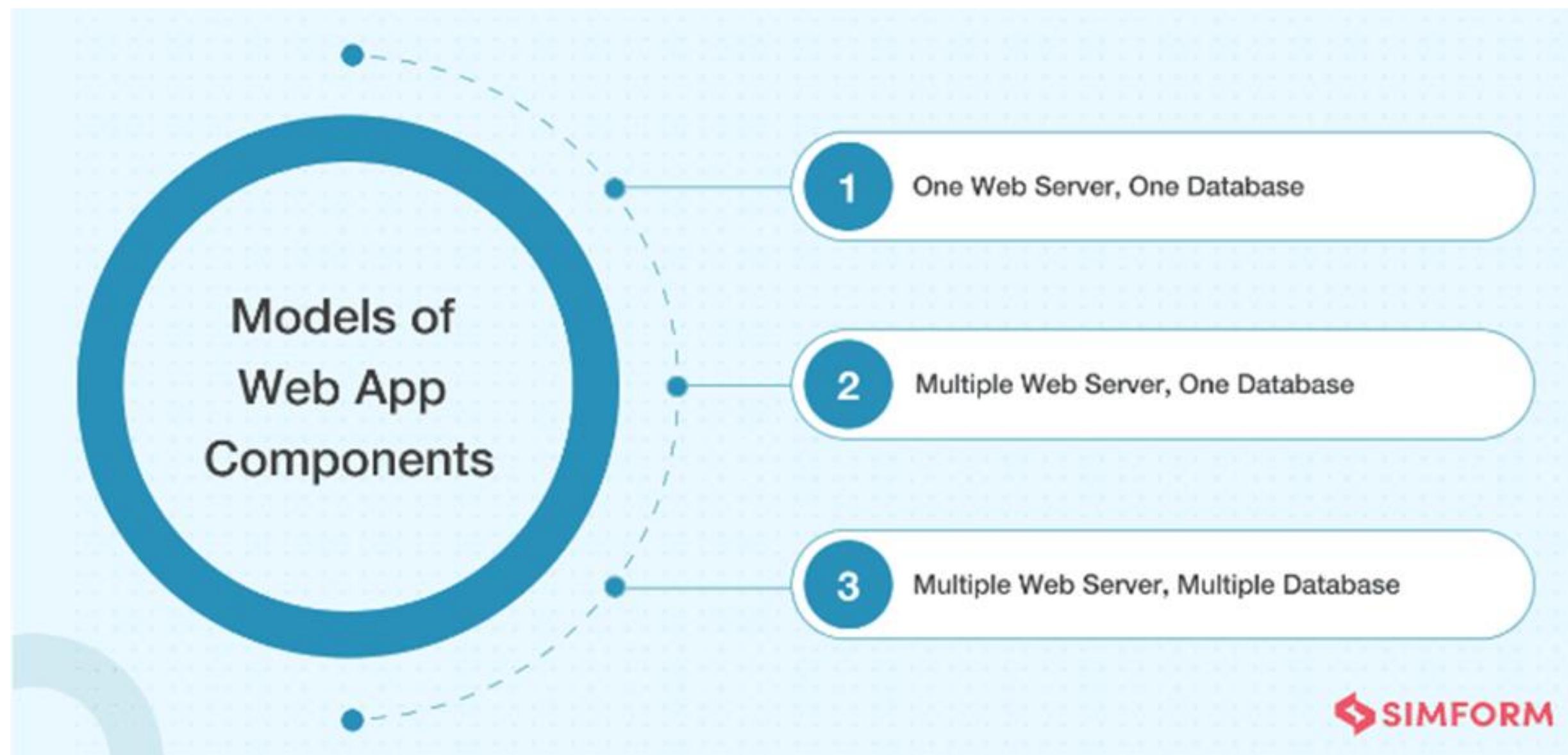
- Là phần giao diện trực quan của ứng dụng web và không có tương tác với kiến trúc.

Cấu trúc web:

- Các thành phần ứng dụng máy khách:
 - Tồn tại trong trình duyệt của người dùng.
 - Tương tác với các chức năng của ứng dụng web.
- Các thành phần phía máy chủ:
 - Một máy chủ ứng dụng web.
 - Một máy chủ cơ sở dữ liệu.



5. Mô hình ứng dụng web



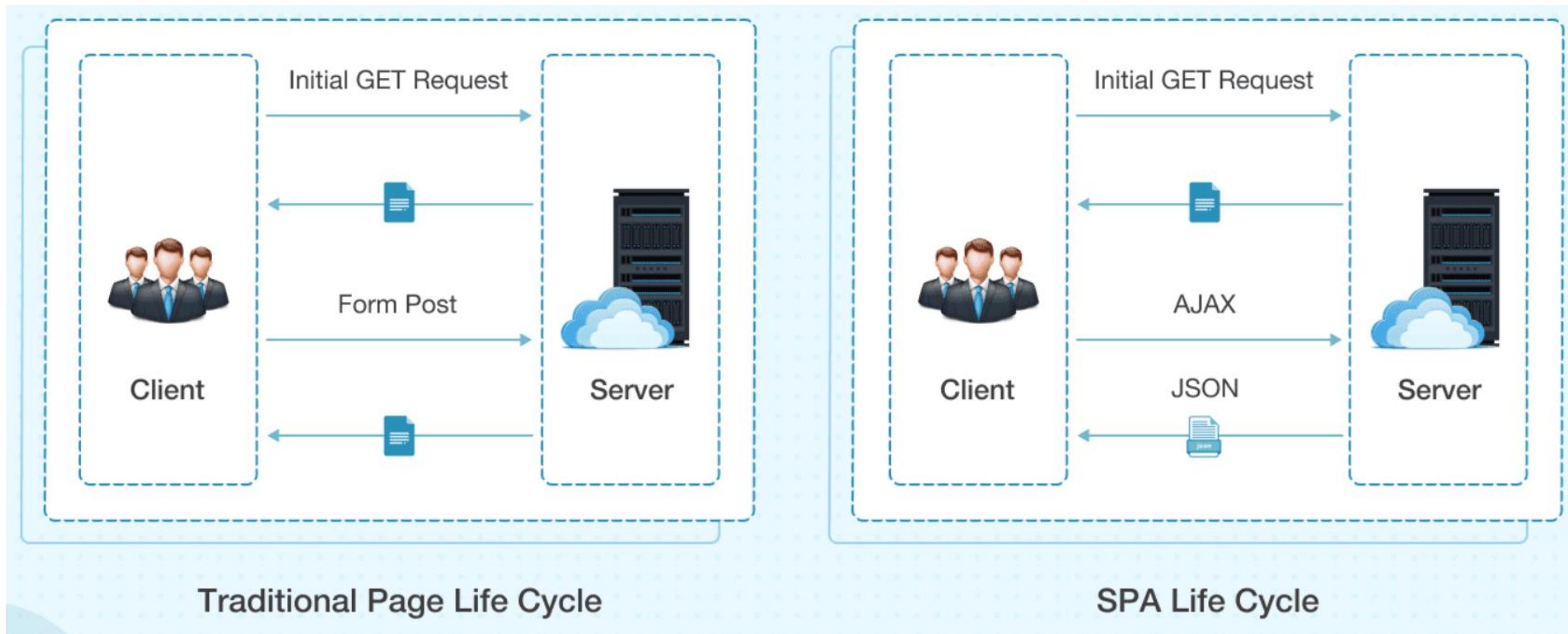
Hình 5: Sơ đồ các mô hình ứng dụng web

6. Các loại kiến trúc ứng dụng web



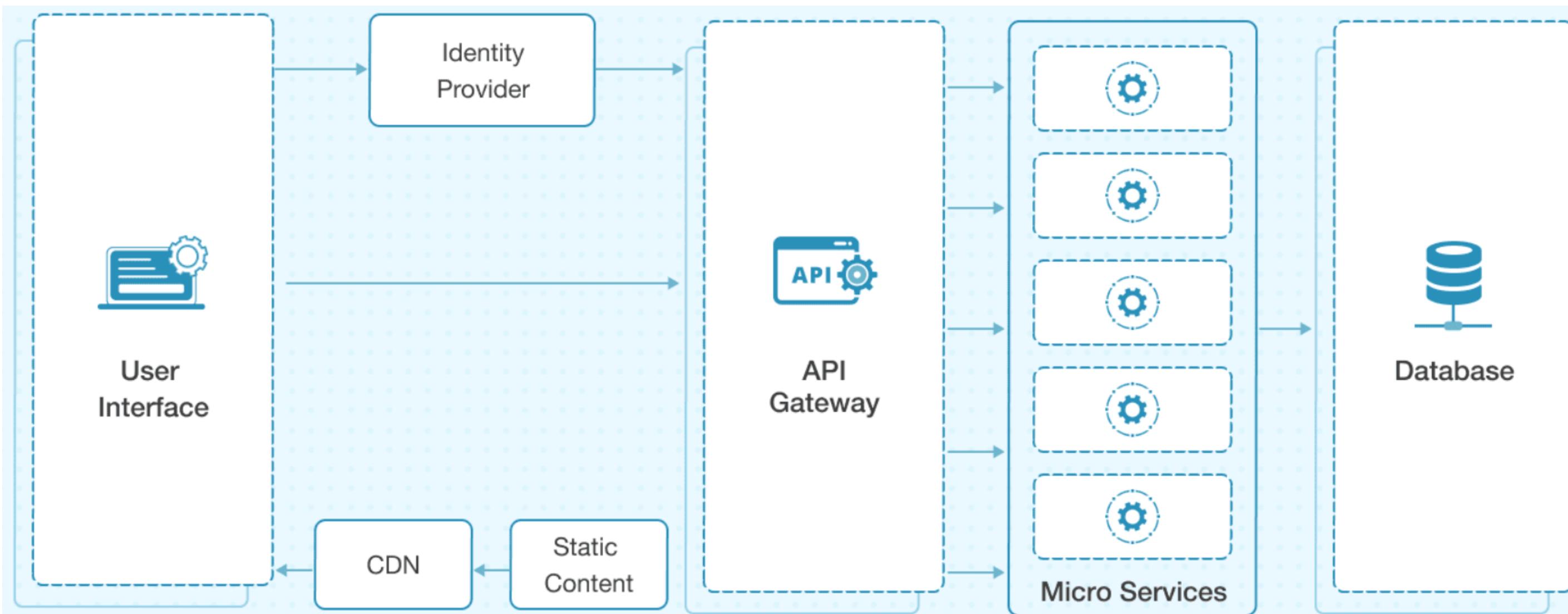
- Single Page Applications
- Microservice
- Serverless

• Single Page Applications



SPA được sinh ra khắc phục những hạn chế truyền thống để đạt được hiệu suất ứng dụng mượt mà, trải nghiệm người dùng trực quan.

2. Microservice



Microservice architecture: đơn giản hóa kết nối giữa các thành phần dịch vụ. Vì ứng dụng được tách ra nhiều thành phần dịch vụ riêng lẻ.

3. Serverless



Là một m^{odel} thiết kế trong đó các ứng dụng được xây dựng và chạy mà không có bất kỳ sự can thiệp thủ công nào trên các máy chủ được quản lý bởi các nhà cung cấp dịch vụ đám mây bên thứ ba như Amazon và Microsoft.

7. Một số lưu ý khi thiết kế ứng dụng web



- Authentication & Authorization
- Exception Management
- Validation

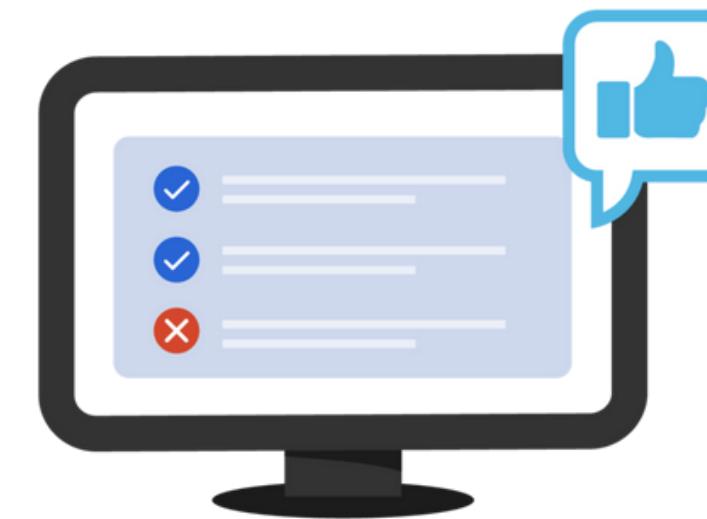
Authentication & Authorization

Authentication



Confirms users
are who they say they are.

Authorization



Gives users permission
to access a resource.

Authentication (xác thực) có nghĩa là xác nhận danh tính của riêng bạn, trong khi Authorization (ủy quyền) có nghĩa là cấp quyền truy cập vào hệ thống.

Authentication	Authorization
Authentication xác nhận danh tính của bạn để cấp quyền truy cập vào hệ thống.	Authorization xác định xem bạn có được phép truy cập tài nguyên không.
Đây là quá trình xác nhận thông tin đăng nhập để có quyền truy cập của người dùng.	Đó là quá trình xác minh xem có cho phép truy cập hay không.
Nó quyết định liệu người dùng có phải là những gì anh ta tuyên bố hay không.	Nó xác định những gì người dùng có thể và không thể truy cập.
Authentication thường yêu cầu tên người dùng và mật khẩu.	Các yếu tố xác thực cần thiết để authorization có thể khác nhau, tùy thuộc vào mức độ bảo mật.
Authentication là bước đầu tiên của authorization vì vậy luôn luôn đến trước.	Authorization được thực hiện sau khi authentication thành công.
Ví dụ, sinh viên của một trường đại học cụ thể được yêu cầu tự xác thực trước khi truy cập vào liên kết sinh viên của trang web chính thức của trường đại học. Điều này được gọi là authentication.	Ví dụ, authorization xác định chính xác thông tin nào sinh viên được phép truy cập trên trang web của trường đại học sau khi authentication thành công.

2. Exception Management

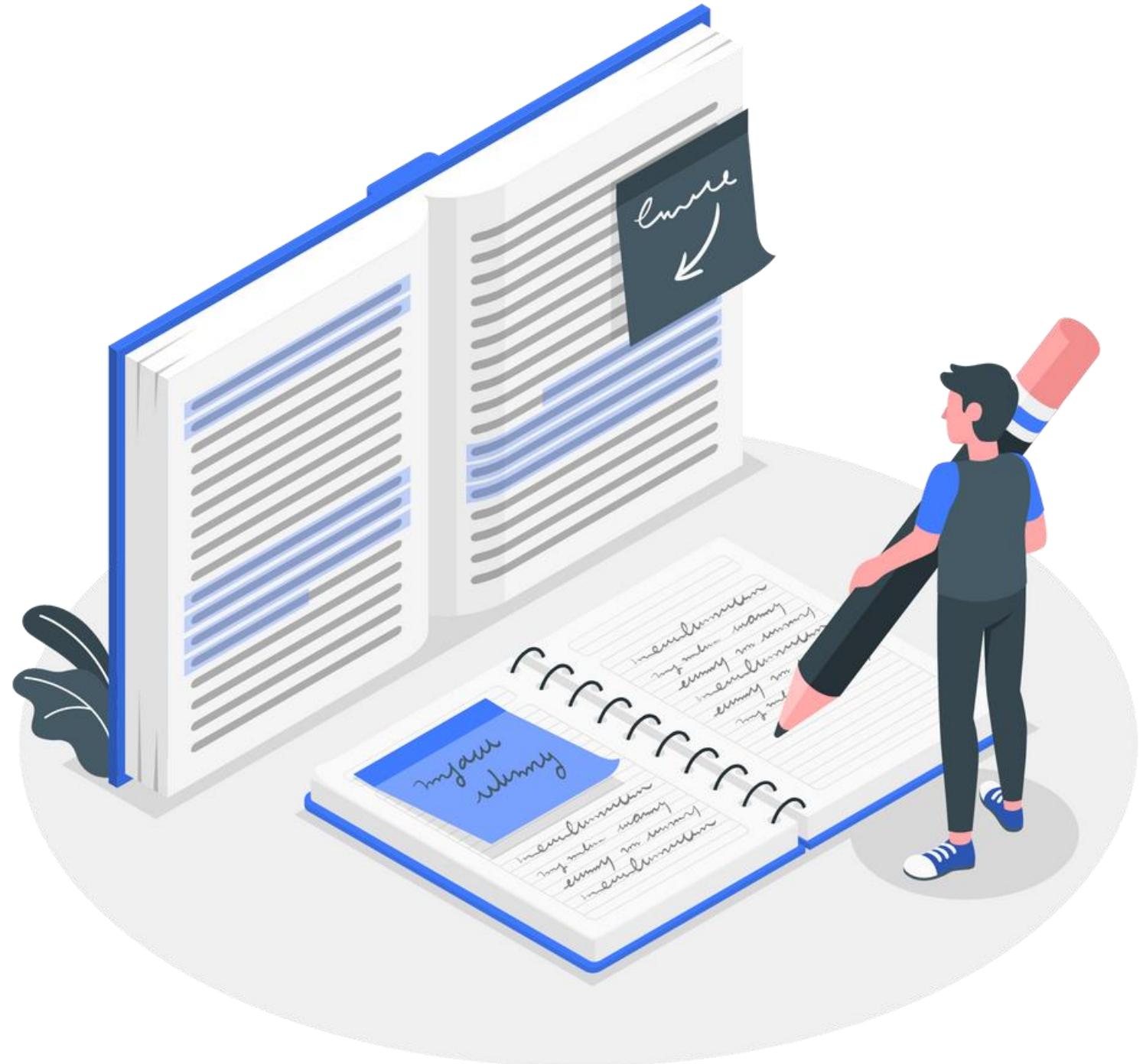


Ngoại lệ là vấn đề - lỗi phát sinh trong quá trình thực thi chương trình.

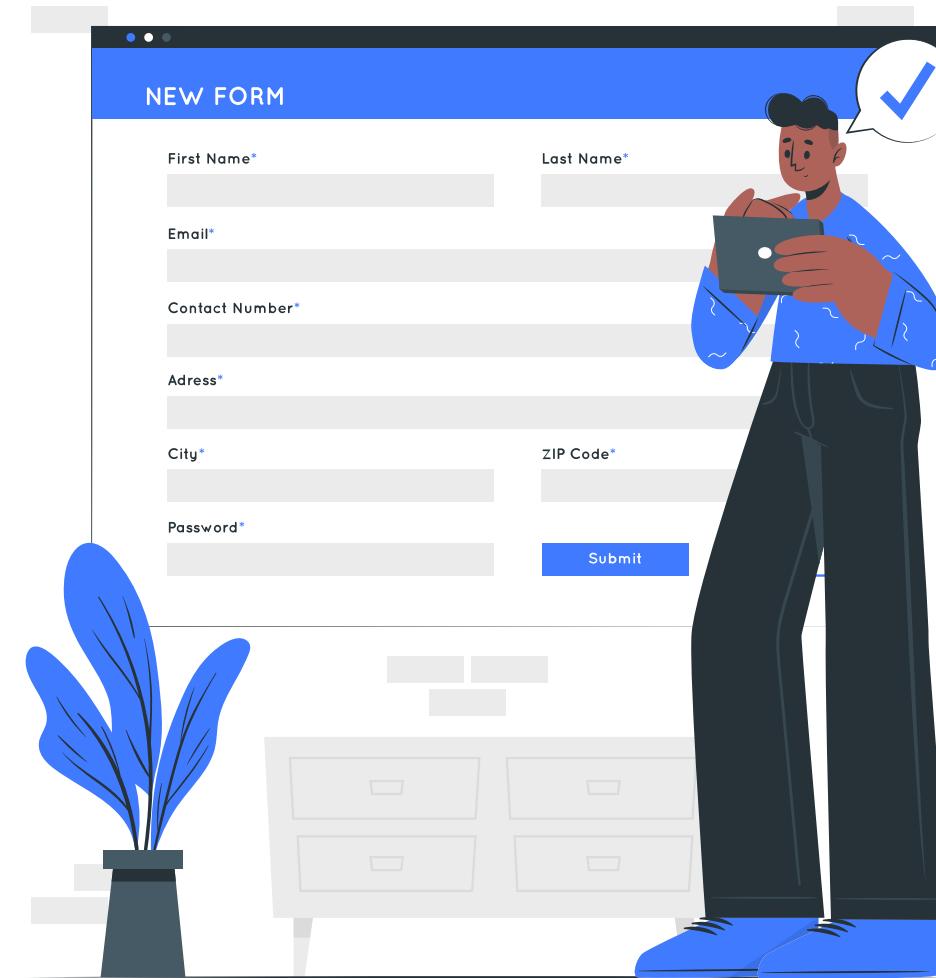
- Các cách xử lý ngoại lệ
 - Sử dụng khối lệnh try/catch/finally
 - Sử dụng throw và throws
 - Sử dụng đối tượng error



- **Lưu ý khi xử lý ngoại lệ**
 - Cung cấp thông báo lỗi thân thiện với người dùng nhưng tránh để lộ dữ liệu nhạy cảm.
 - Nắm bắt được các ngoại lệ chưa được xử lý, dọn dẹp tài nguyên và trạng thái khi xảy ra ngoại lệ.
 - Thiết kế trình xử lý ngoại lệ chung.
 - Tránh sử dụng các ngoại lệ tùy chỉnh khi không cần thiết.
 - Không sử dụng ngoại lệ để kiểm soát luồng logic ứng dụng.



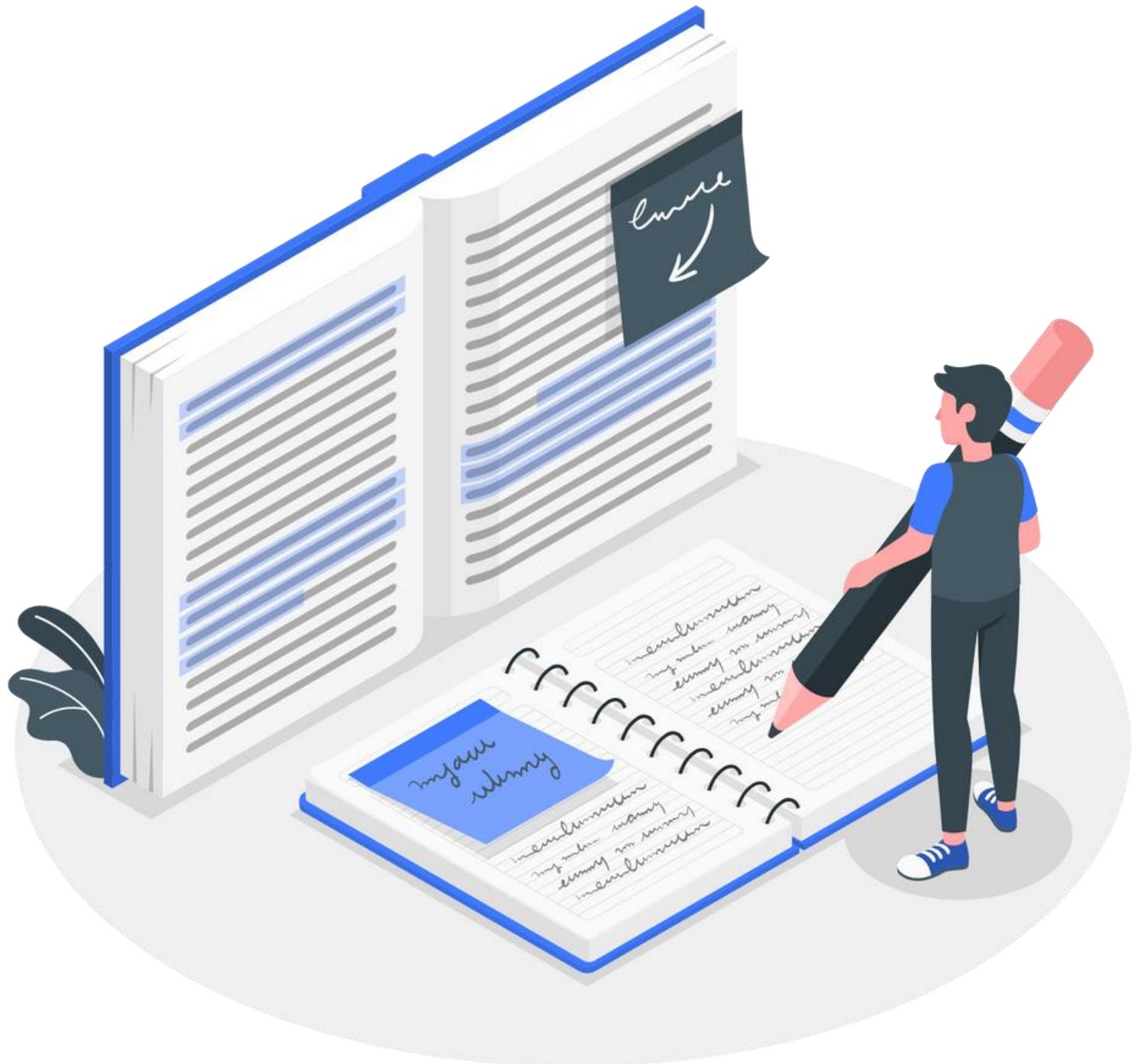
3. Validation



Kiểm tra dữ liệu là quá trình đảm bảo ứng dụng web của bạn hoạt động trên dữ liệu sạch, chính xác và hữu ích.

• Lưu ý khi kiểm tra dữ liệu

- Kiểm tra tất cả dữ liệu vượt qua ranh giới tin cậy của ứng dụng.
- Thiết kế chiến lược kiểm tra để hạn chế, từ chối và loại bỏ đầu vào độc hại và kiểm tra tất cả dữ liệu đầu vào dựa trên độ dài, phạm vi, định dạng và loại.
- Việc kiểm tra ở phía máy khách để có trải nghiệm người dùng tối ưu và giảm các yêu cầu đến máy chủ, nhưng luôn kiểm tra trên máy chủ vì lý do bảo mật.



Thanks for watching & QA

THANK
you