Learn with Rabbil

ব্যাকেন্ড নাকি ফ্রন্টেন্ড
বিগিনাররা কোনটা আগে শিখবেন?

Learn with Rabbil

যে কোনো কাজ শুরুর আগেই সেই কাজ সম্পর্কে আগে জানা দরকার

Learn with Rabbil

JOURNEY TO BECOME

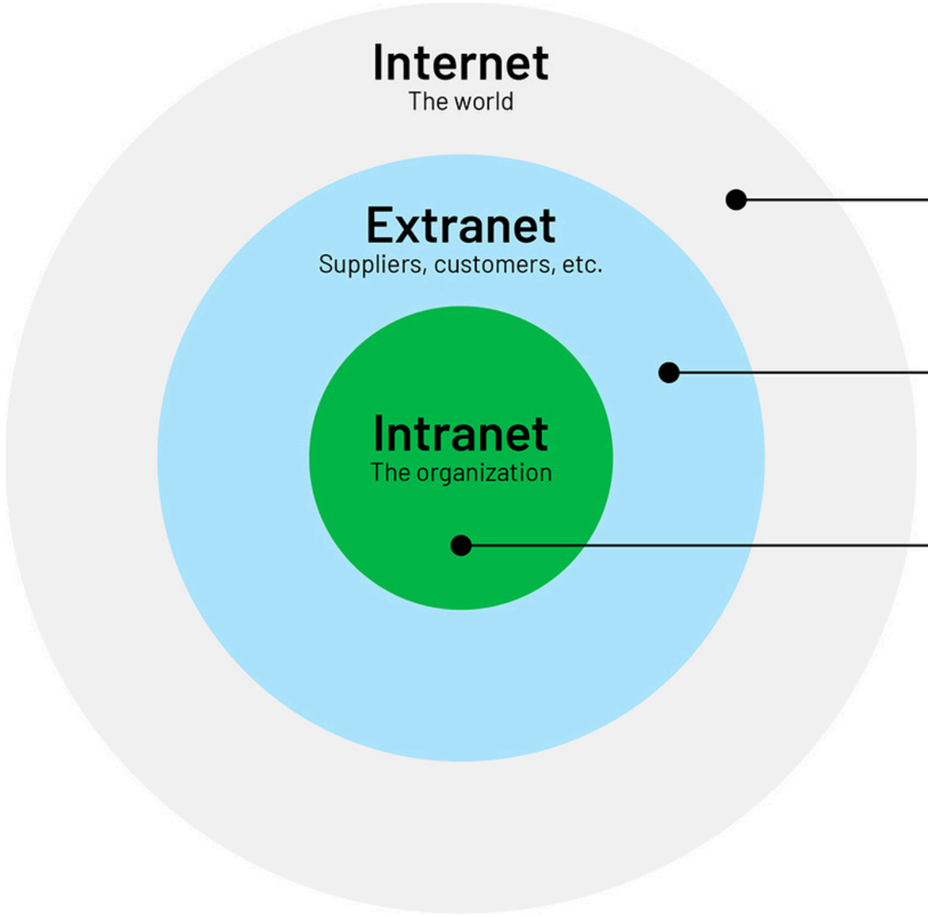
A Web Developer

Learn with Rabbil

THE WEB

Interconnected system of internet-based documents and resources accessed through browsers

INTERNET VS. INTRANET VS. EXTRANET



The **internet** creates connections between computers around the world.

An **extranet** creates connections beyond (or outside) an organization.

An **intranet** creates connections inside an organization.

Learn with Rabbil

তথ্য থাকবে সার্ভারে সবাই পাবে ব্রাউজার দিয়ে

- সার্ভার কম্পিউটার
- ইন্টারনেট/নেটোয়ার্ক মাধ্যম
- ব্রাউজার ইন ডিভাইস

সার্ভার কম্পিউটার


aws
DigitalOcean
Microsoft Azure


bluehost
SiteGround
GreenGeeks
HOSTINGER
HostGator.com
inmotion hosting
A2 HOSTING
OUR SPEED, YOUR SUCCESS
GoDaddy
DreamHost
iPage

ইন্টারনেট/নেটোয়ার্ক মাধ্যম

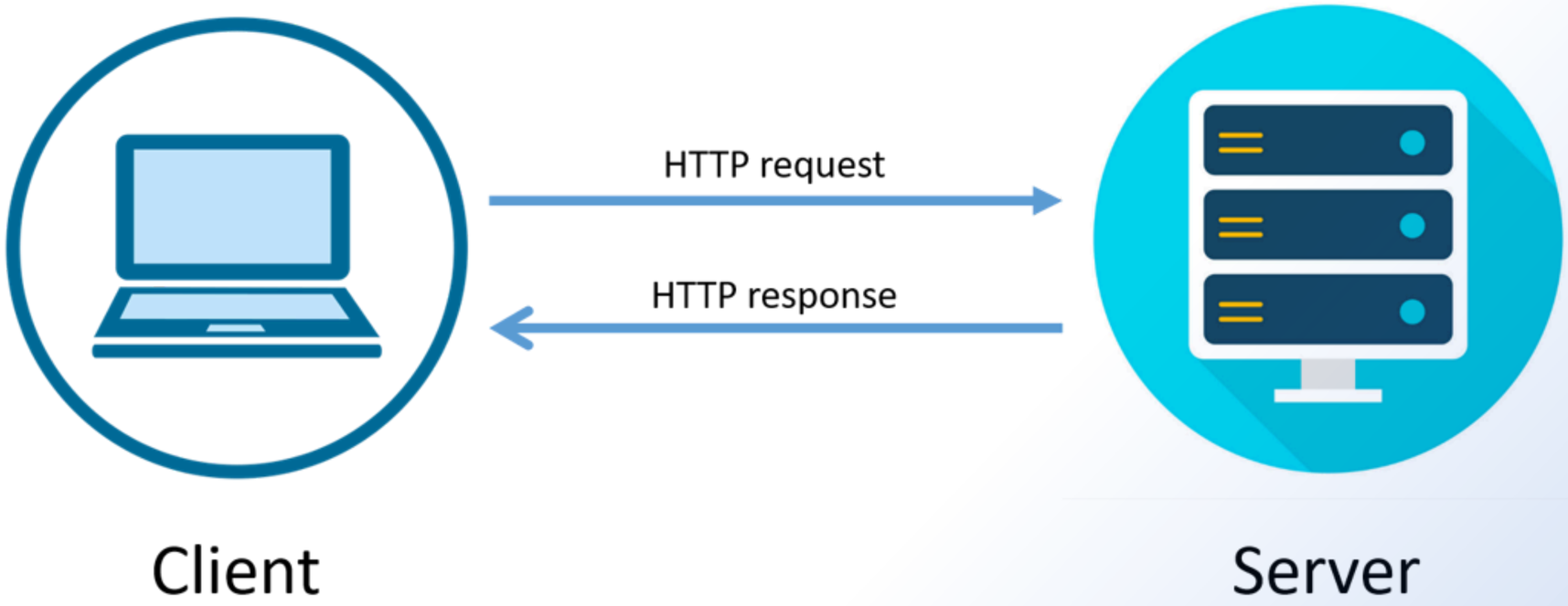
ব্রাউজার ইন ডিভাইস

Learn with Rabbil

তথ্য থাকবে সার্ভারে
সবাই পাবে ব্রাউজার দিয়ে

সার্ভার কম্পিউটার	→	সার্ভার সাইড/ ব্যাক-ইন্ড
ইন্টারনেট মাধ্যম	→	আনা নেওয়া মধ্যাম
ব্রাউজার ইন ডিভাইস	→	ক্লায়েন্ট সাইট/ ফ্রন্ট-ইন্ড

Request-Response Model



Learn with Rabbil

ডেটাবেজ সার্ভার

স্ট্যাটিক ওয়েব সাইট

- সার্ভার কম্পিউটার
- ইন্টারনেট মাধ্যম
- ব্রাউজার ইন ডিভাইস

Learn with Rabbil

ডাইন্যামিক ওয়েব সাইট

- সার্ভার কম্পিউটার
- ইন্টারনেট মাধ্যম
- ব্রাউজার ইন ডিভাইস

ডাইন্যামিক ওয়েব এপ্লিকেশন

- সার্ভার কম্পিউটার
- ইন্টারনেট মাধ্যম
- ব্রাউজার ইন ডিভাইস

Learn with Rabbil

ওয়েব UI/UX

Learn with Rabbil

ওয়েব টেম্পেলেট/ থিম ডেভেলপমেন্ট

Learn with Rabbil

ওয়েব ফ্রন্ট-ইন্ট ডেভেলপমেন্ট

Learn with Rabbil

ওয়েব ব্যাক-ইন্ড ডেভেলপমেন্ট

Learn with Rabbil

ওয়েব ফুল-স্ট্যাক ডেভেলপমেন্ট

Learn with Rabbil

ওয়েব এপ্লিকেশন সফটওয়্যার এর একটি ধরন

HTTP Client

HTTP Client Library	Language
Volly	Java
Retrofit	Java
Rest Sharp	C#
Axios	JavaScript
cURL	PHP

Application Level Client: Library used in client side application to generate request and receive response.

Browser Client: Browser is the primary HTTP Client responsible for load the web application.

POSTMAN Http Client

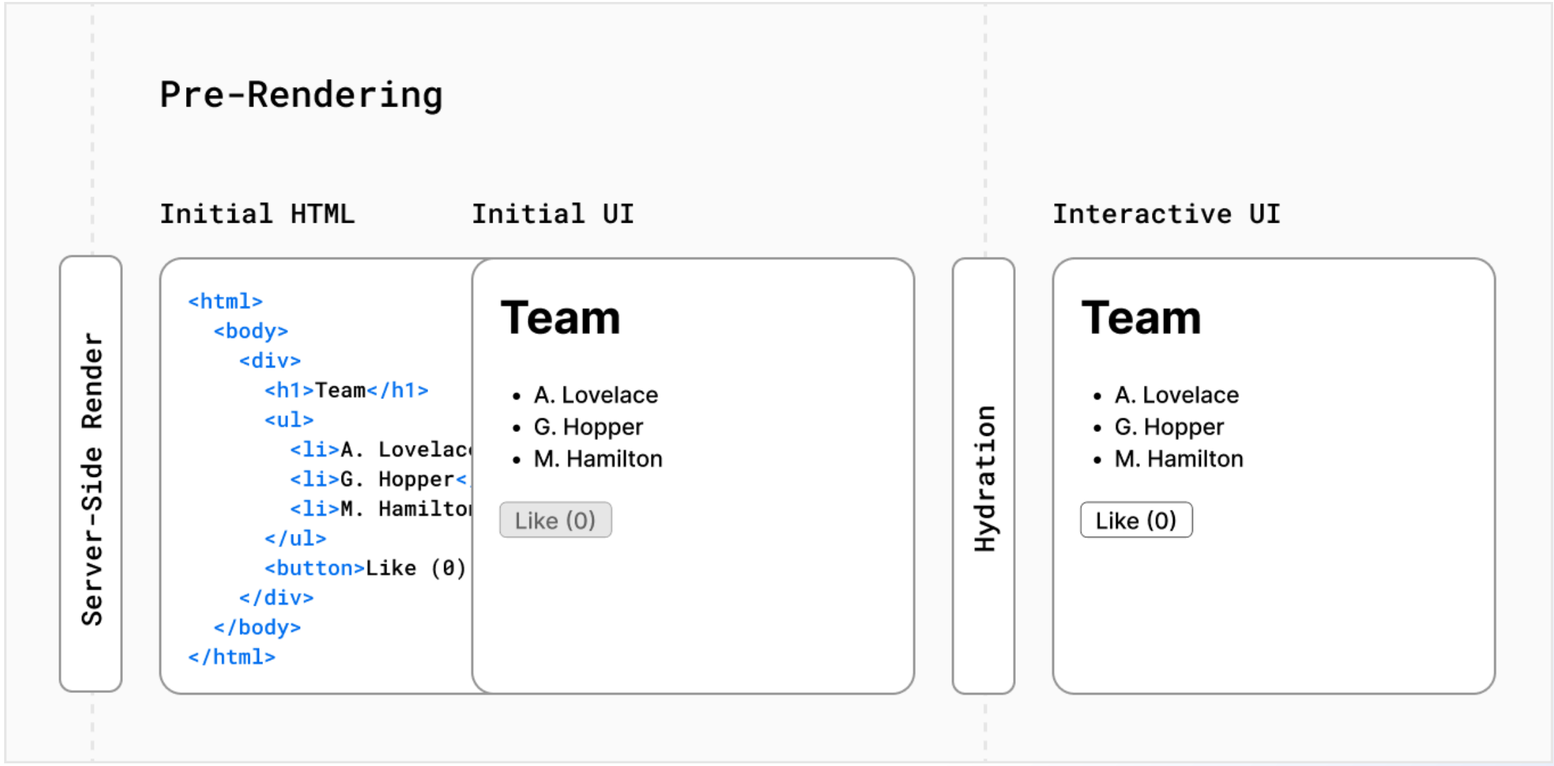
Postman is an HTTP Client application, used to test request-response communication.

Postman is widely used for API testing and generating documentation

- Quickly and easily send REST, SOAP, and GraphQL requests directly within Postman.
- Generate and publish beautiful, machine-readable API documentation.
- Checking performance and response times at scheduled intervals.
- Communicate the expected behavior of an API by simulating endpoints and their responses

Server-Side Rendering

- With server-side rendering, the HTML of the page is generated on a server for each request.
- The generated HTML, JSON data, and JavaScript instructions to make the page interactive are then sent to the client.



BENEFITS OF SSR

Better SEO

Server-side rendering allows search engine bots to crawl and index the content of your website easily, leading to better search engine rankings.

Faster initial loading

Server-side rendering provides faster initial loading of web pages by sending HTML content to the browser before the JavaScript is executed. This helps reduce page load time and improves user experience.

Improved accessibility

Server-side rendering can improve website accessibility by ensuring that content is available to all users, including those with slow or unreliable internet connections, or those using assistive technologies.

Improved security:

Server-side rendering can improve website security by reducing the risk of cross-site scripting (XSS) attacks that are common with client-side rendering.

Better performance on low-powered devices:

Server-side rendering can improve the performance of web pages on low-powered devices such as mobile phones and tablets by reducing the processing requirements on the client-side.

OPPOSITE OF SSR

Higher server load

Server-side rendering requires more server resources to generate and serve pages, which can increase server load and affect website performance.

Limited interactivity

Server-side rendering does not allow for complex interactive features that require heavy client-side processing, such as real-time updates or animations.

Limited flexibility

Server-side rendering may limit the ability to customize the user interface or provide personalized experiences, as the HTML is generated on the server.

Higher development costs

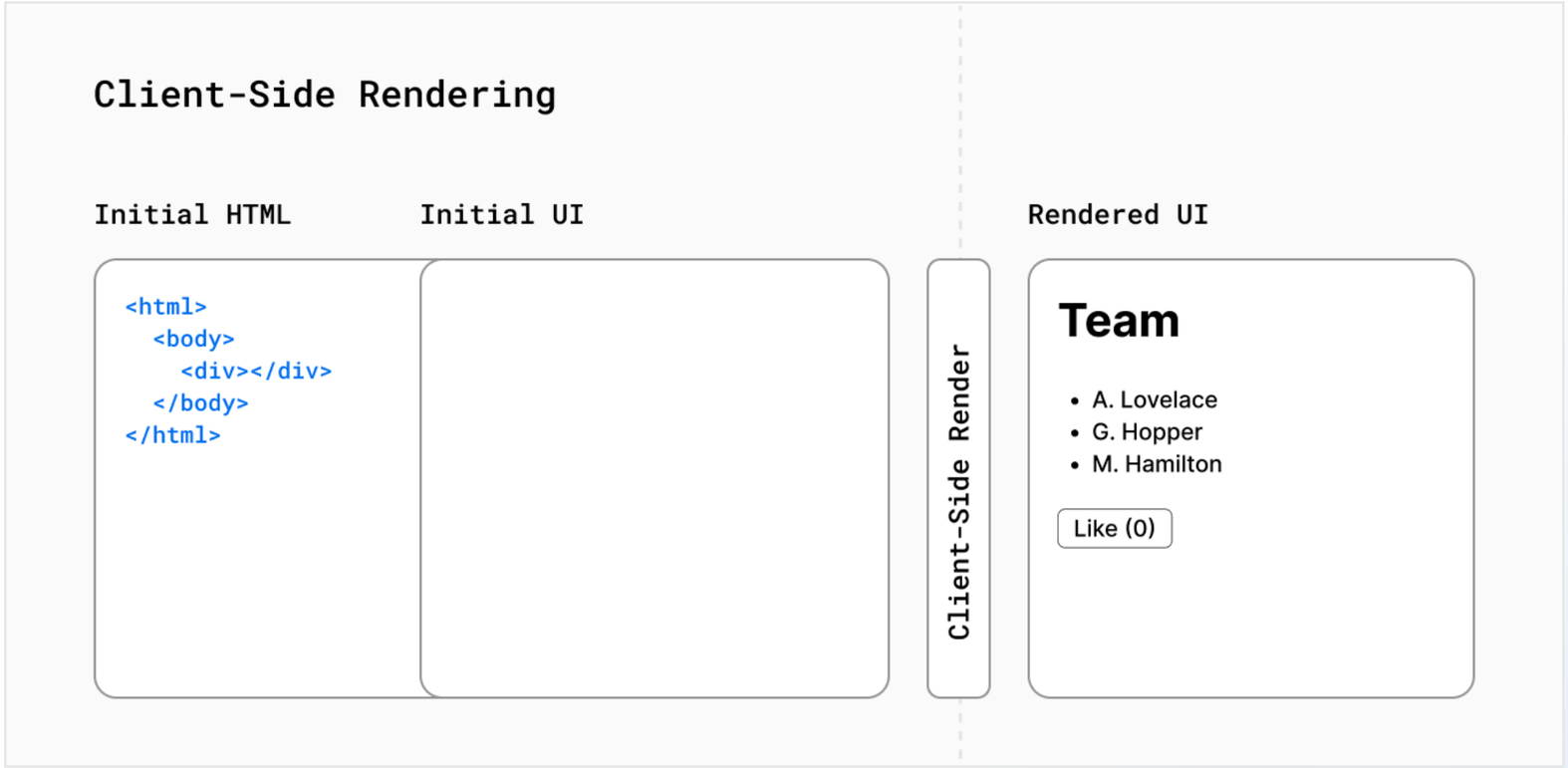
Server-side rendering may require additional development resources and expertise, leading to higher development costs.

Complexity

Implementing server-side rendering can be complex and require specific technologies and configurations, which can be difficult for some developers to manage.

Client-Side Rendering

- With client-side rendering, the HTML of the page is generated inside the web browser.
- This approach allows for faster and more dynamic interactions with the web page.as the browser can update specific parts of the page without having to reload the entire page from the server



Learn with Rabbil

BENEFITS OF CSR

Improved Performance

Client-side rendering reduces server load and bandwidth consumption, initial page load requires a smaller amount of data to be transferred from the server to the client rendering process is performed on the client-side, which reduces the server's workload and frees up resources.

Better User Experience

Client-side rendering provides a more interactive and dynamic user experience. The content can be updated dynamically, and page navigation is faster, as only the necessary data is fetched from the server.

Easier Development

Client-side rendering frameworks, such as React, Angular, and Vue.js, provide developers with tools and libraries that simplify the development process.

Faster Iteration

With client-side rendering, developers can iterate quickly and test changes in real-time without having to wait for server-side rendering. This speeds up the development process and reduces the time it takes to release new features and updates.

OPPOSITE OF CSR

SEO Challenges

While modern search engines are better at indexing client-side rendered pages, there can still be challenges with search engine optimization (SEO). It can be difficult to ensure that the content is fully visible and crawlable by search engines, which can negatively impact search rankings.

Initial Load Time:

The initial load time of a client-side rendered web page can be slower than that of a server-side rendered page because the web browser needs to download the JavaScript code and execute it before displaying the content. This can result in slower page load times, especially on slower devices or slower internet connections.

Browser Compatibility:

Different web browsers may interpret and execute JavaScript differently, which can lead to compatibility issues. This can require additional testing and development work to ensure that the web page functions correctly across all major web browsers.

Security Concerns:

Client-side rendering can increase the risk of security vulnerabilities, such as cross-site scripting (XSS) attacks. This is because the client-side JavaScript code can be modified by malicious actors, which can lead to data theft or other security breaches.

Maintenance Complexity:

Client-side rendering frameworks and libraries can be complex and require a high level of technical expertise to maintain and update. This can make it challenging for smaller development teams to keep up with the latest updates and ensure that the web page remains secure and functional.

Learn with Rabbil

THE OPTMIZED SOLUTION

- Use Server and Client Components.
- We can build a good combination of SSR, CSR using modern JavaScript .
- Use SSR,CSR Where ncesseassary.