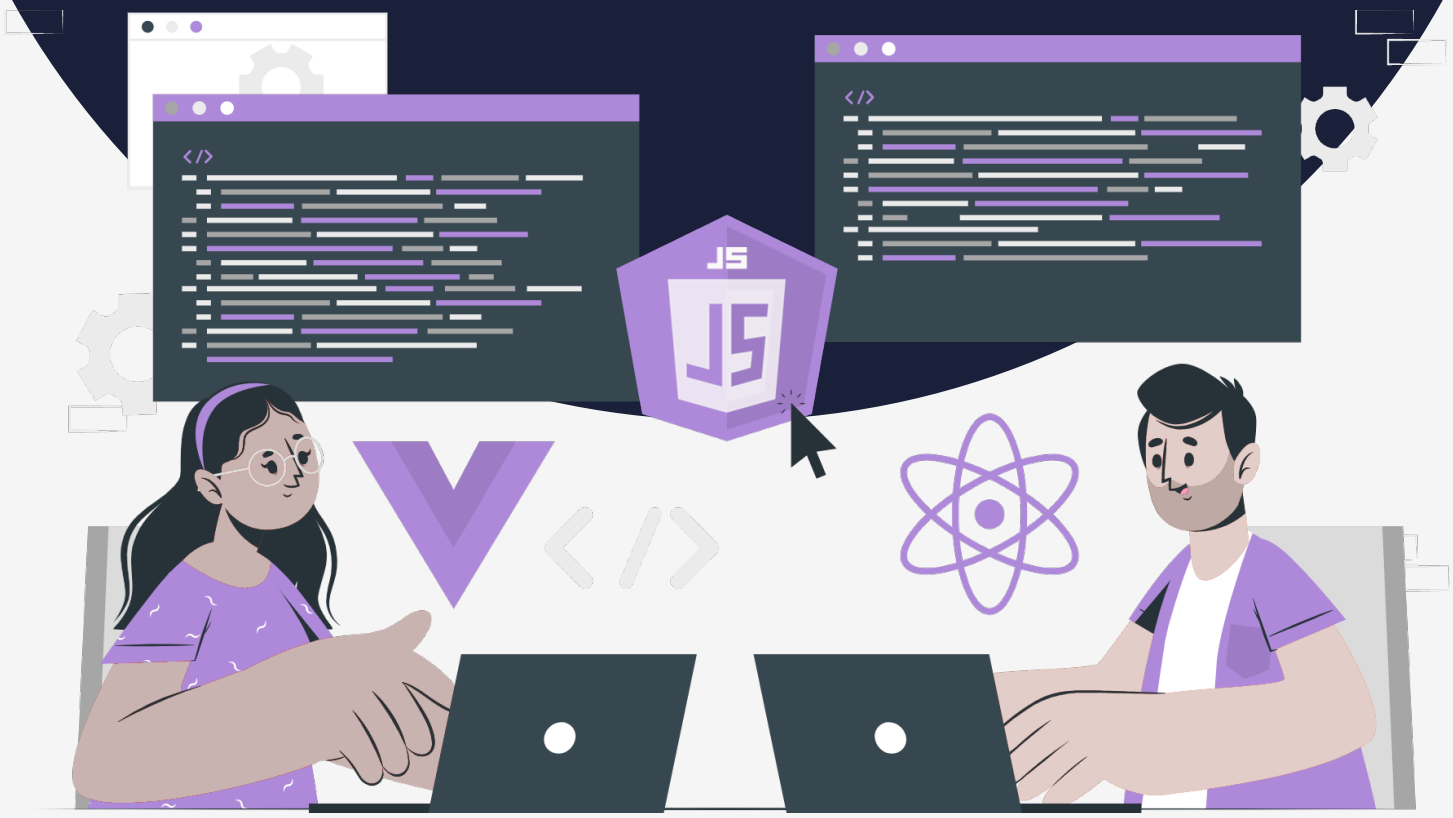
Lesson:

Starter Project with CDN



PW SKILLS

Topics Covered:

1. What is CDN?
2. Why do we use CDN?
3. Where to get react CDN?
4. How to use CDN in the project file.
5. Why we do not use React CDN in production?

What is CDN?
A CDN, or Content Delivery Network, is a distributed network of servers that work together to provide fast and reliable delivery of digital content, such as images, videos, web pages, and other static or dynamic assets, to end-users across the globe.

The main purpose of a CDN is to reduce latency and improve the speed of content delivery by caching and distributing content across multiple geographic locations, so that users can access the content from a server closest to their location. This helps to minimize the distance that the content has to travel, reducing the time it takes to load the content and improving the user experience.

CDNs also provide other benefits, such as improved scalability, increased reliability and availability, and better security, by offloading traffic from origin servers and protecting against DDoS attacks and other types of threats.

Overall, a CDN can be an essential tool for organizations that need to deliver content quickly and efficiently to a global audience, such as e-commerce sites, media companies, and other content-heavy websites.

Why do we use CDN?
Organizations utilize CDNs for a variety of purposes, including

1. Improved Performance: Reduction of the distance that data must travel between the server and the user allows CDNs to contribute to an increase in website performance. The user experience can be enhanced by speeding up page loading times as a result of this.

2. High Availability: By spreading a website's content over numerous servers in various places, CDNs can assist in enhancing a website's accessibility. If one of the servers fails, this can help to ensure that the website is still available.

3. Security: In addition to standard security measures, CDNs can assist against distributed denial of service (DDoS) assaults on websites.

4. Traffic Management: By dividing the load over several servers, CDNs can assist in managing traffic spikes. This may aid in preventing a website's slowdown or breakdown during periods of excessive traffic.

5. Cost Saving: By unloading some of the traffic from the origin server, CDNs can assist in lowering the cost of distributing content. This can help firms save money by reducing the need for extra infrastructures like servers and bandwidth.

Where to get react CDN?
You can get React CDNs from its official website.

https://reactjs.org/docs/cdn-links.html

How to use CDN in the project file
We can directly use CDN in our HTML file at the bottom of the body tag

```JavaScript
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="./index.css" />
    <title>React CDN</title>
  </head>
  <body>

    <script crossorigin
src="https://unpkg.com/react@18/umd/react.development.js"></sc
ript>
<script crossorigin
src="https://unpkg.com/react-dom@18/umd/react-dom.development.
js"></script>

  </body>
</html>
```

Why we do not use React CDN in production?
While it is possible to use React CDN (Content Delivery Network) in production, it is generally not recommended because of a few reasons:

1. **Security concerns**: Using an external CDN to host and deliver your React code means that you are relying on a third-party to serve your code, which can potentially introduce security vulnerabilities or risks.
2. **Performance issues**: CDNs can increase page load times if the user has to fetch the React code from the CDN. This can lead to a slower user experience, which can negatively impact your website or application.
3. **Control over dependencies**: Using a CDN means you have less control over which version of React is being served, and you may encounter compatibility issues with other dependencies in your application.

Instead, it is recommended to bundle and serve your React code along with your other assets from your own server or a cloud-based service like AWS S3 or Google Cloud Storage. This gives you more control over the security and performance of your application, and allows you to easily manage your dependencies.