📜 History of ReactJS

🔹 2011 – Creation at Facebook

- **Inventor:** Jordan Walke, a software engineer at Facebook.
- Built as an internal tool to manage complex UIs in Facebook's News Feed.
- Inspired by **XHP**, an HTML component framework for PHP used at Facebook.

🔹 2013 – Open Source Release

- **May 2013:** ReactJS was **open-sourced** at **JSConf US**.
- Early adoption was slow due to:
  - JSX (mixing HTML with JavaScript) being unconventional.
  - Component-based approach being a new concept.

🔹 2014–2015 – Rising Popularity

- Facebook started using React more broadly.
- **Instagram**, owned by Facebook, also adopted React.
- **Community started to grow** rapidly thanks to performance and developer experience.

🔹 2015 – React Native Released

- React Native enabled **native mobile app development** using the same React architecture.
- Developers could build iOS and Android apps using JavaScript + React.

🔹 2016 – React 15

- Major improvements to performance and bug fixing.
- Started preparing for deeper changes under the hood.

◆ 2017 – React 16 & React Fiber

- Complete rewrite of React’s core (called **Fiber**).
- Allowed features like **error boundaries**, **portals**, and better rendering control.
- Focused on enabling **asynchronous rendering**.

◆ 2019 – Introduction of Hooks (React 16.8)

- **Hooks** revolutionized how developers write components:
  - Allowed functional components to manage state and lifecycle.
  - Made code cleaner, easier to reuse logic.

◆ 2020–2021 – React 17

- A "no new features" release meant to ease gradual upgrades.
- Focused on improving the ability to upgrade React apps piece by piece.

◆ 2022 – React 18

- Introduced:
  - **Concurrent rendering** (better UI responsiveness).
  - **Automatic batching** of state updates.
  - **Suspense for data fetching**.
- Huge step toward more efficient and scalable apps.

◆ Today (2025)

- React remains one of the most popular libraries for building UIs.
- Widely used in startups, enterprises, and open-source projects.
- Supported by **Meta (Facebook)** and a strong open-source community.

🔧 React's Legacy

- React helped shape modern frontend development.
- It influenced many other frameworks (like Vue and Svelte).
- Encouraged trends like:
  - Component-based UI
  - Declarative programming
  - Reusable logic with hooks