**React Installation**

Pre-requisite of creating a react-app is having Node Js. NodeJs is an open-source cross platform back-end JavaScript Run Time Environment, which also install some tools like npm & npx in our local system in order to start server and run the website. Basically, npm & npx tool contain itself by Node Js.

**npm:**npm is a website having its own server, which is a packet manager for JavaScript Programming Language.   
Note: There is a packet manager for every Programming Language which manages stuffs like dependencies.

Facebook built react library on node and store it over there as package to access and download it from anywhere across the world.

Basically, any package place there at npm website can access and download by running either npm command or npx command.

So, here we’re going to use **npx command** to get the react package/files in our local system, that placed there by facebook at npm website.

**Steps to create react App:-**

1. Install NodeJs and NPM  
   --Check installation using CMD using these command, node –v and npm –v for Node and NPM respectively.
2. Install Visual Studio Code
3. Create React App from terminal (Here, terminal is CMD (in-built from VS Code) or Original CMD Terminal) using NPX Commands:-  
   a). npx create-react-app <app-name>   
   b). cd <app-name>  
   c). npm start

**For fast creation of React App (In just few Seconds) using Vitejs.dev. Here’s the another method of creating react App:**

1. Go to [*https://vitejs.dev/guide/#overview*](https://vitejs.dev/guide/%23overview)
2. Open window powershell in VS Code.
3. Type this command **npm create vite@latest** (Make sure to install node.js inside the system in order to run npm command and establish nodejs Server).
4. Type **Project Name** (If we’re already inside the project folder means that folder where we want to store our project files [Ex: Web 3.0/client], then just type **./** to create all react app files under client folder).
5. Type **Package Name** whatever we would like it to call. Ex: krypt.
6. Select a **Framework** and its variant**.** Ex: react for framework & variant.
7. Now run **npm install**
8. Then run **npm run dev**

**Install Tailwind CSS with Create React App**

1. Go to [*https://tailwindcss.com/docs/guides/create-react-app*](https://tailwindcss.com/docs/guides/create-react-app)
2. Open Window Poweshell terminal in VS Code.
3. Type **npm install -D tailwindcss postcss autoprefixer**
4. Then run **npx tailwindcss init -p** to create *tailwindconfig.js* file as well as *postcss.config.js* files under the project.
5. Now, copy the **template paths** and paste it to the tailwind css config file i.e. (tailwind.config.js).
6. Now, copy the tailwind directives to our css file under src file. i.e. ./src/index.css
7. And finally Again run npm command in order to execute the project and see the tailwindcss result. Run *npm run dev* (If react app created using vet) or *npm run start* (if react add created using default npx )
8. **Install Bootstrap CSS with Create React App**
9. Open terminal (ctrl + j)
10. Run this command: **npx create-react-app@latest boot –use-npm**
11. Hit Enter, to create a new react app with bootstartp.

**Deploy React Application**

If we build the react app using **npx create-react-app**, then follow the following steps to deploy it.

1. Open terminal (ctrl + j)
2. Run this command: **npm run build**  
   this is going to create a “**deployed** **version”** which would be ready for production.
3. Finally, deployed version (as **build folder**) of our react app has created consisting various files for deployment.

Now, we just need to upload “**all those deployed files**” of build folder onto our root directory or repository of hosting to deploy our react app.

If we build the react app using **Vitejs.dev**, then follow the following steps to deploy it.

1. Open terminal (ctrl + j)
2. Move to the ‘**client**’ directory (cd client)
3. Run this command: **npm run build**  
   this is going to create a “**deployed** **version”** which would be ready for production.
4. Finally, deployed version (as **dist folder**) of our react app has created under client folder consisting (assests & index.html).
5. Now, we just need to upload “**assets folder & index.html file**” of dist folder onto our file directory or repository to deploy our react app.