**Steps followed to clone Air ALgerie web site using react js for the front end and tailwind for css:**

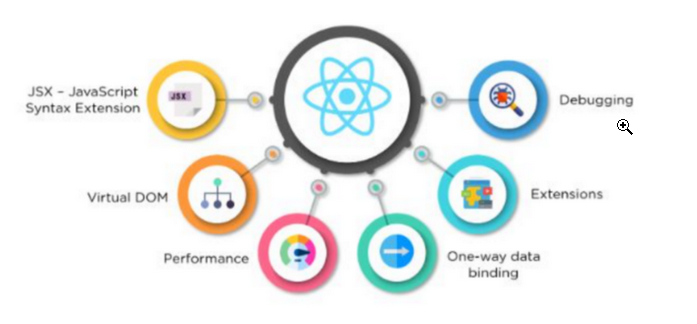
M**ERN** is one of several variations of the [MEAN stack](https://www.mongodb.com/mean-stack) (MongoDB Express Angular Node), where the traditional Angular.js frontend framework is replaced with React.js. Other variants include MEVN (MongoDB, Express, Vue, Node), and really any frontend JavaScript framework can work.

1. **Front End:**

* **React JS:** A JavaScript library for building user interfaces

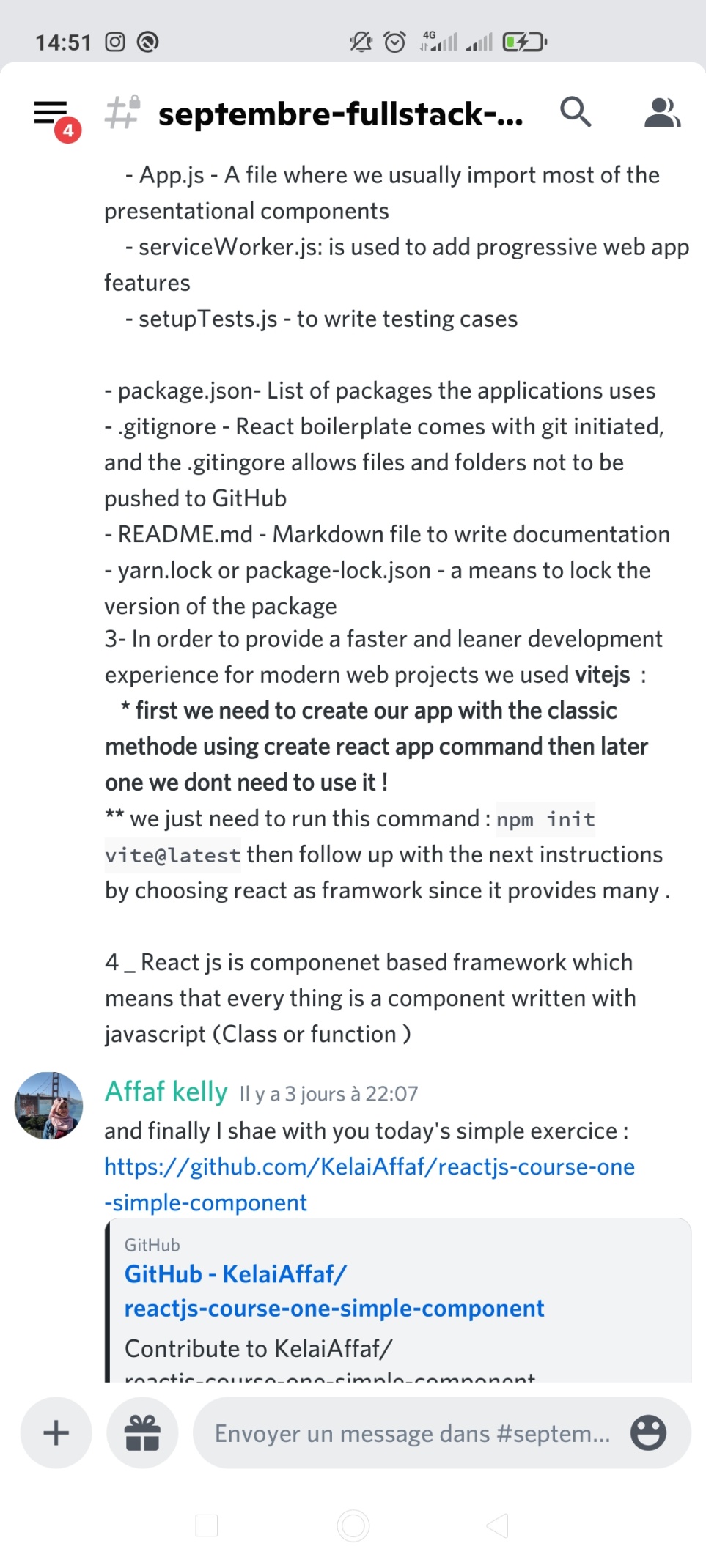
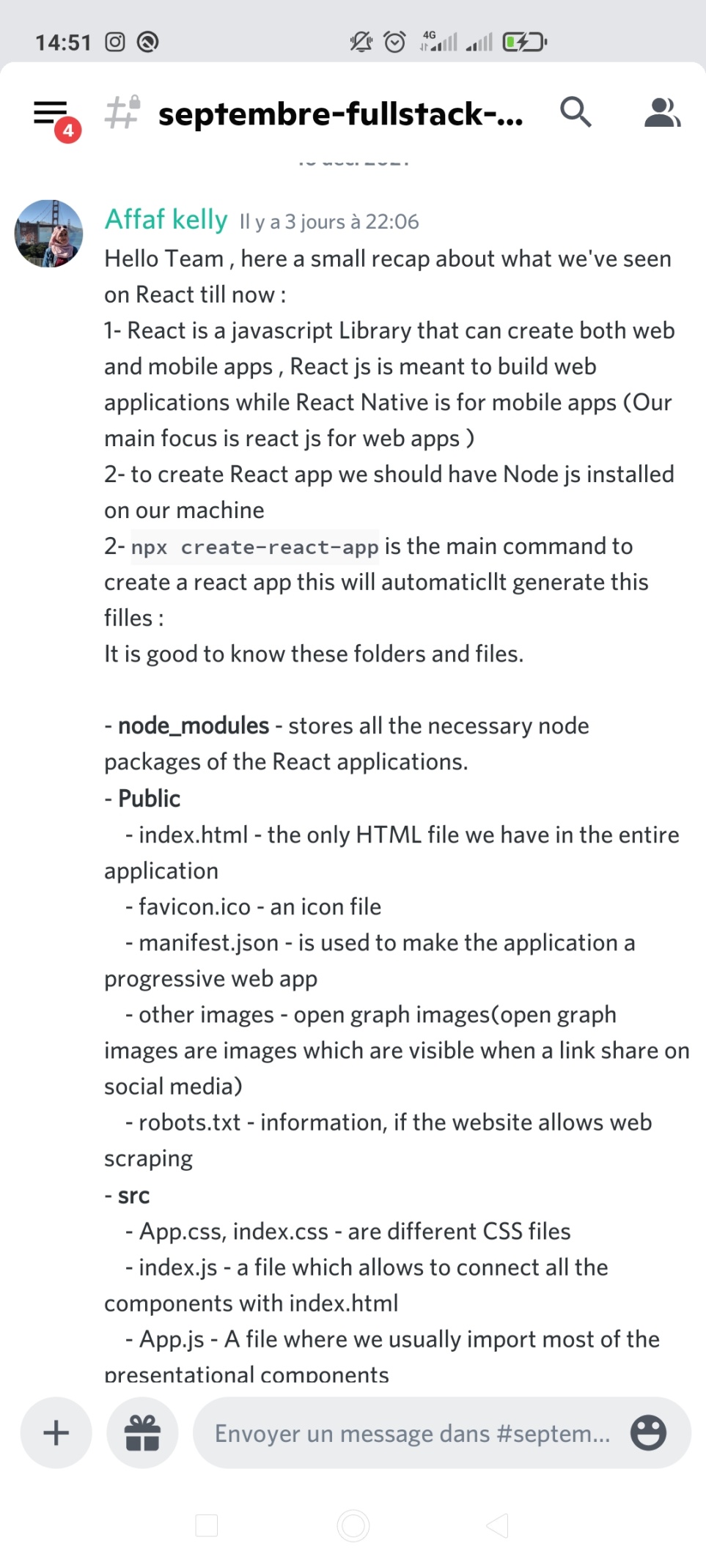
The top tier of the MERN stack is React.js, this library for Client-Side Applications in HTML is completely written in JS and lets us build complex interfaces through simple **components**! Yeah React UIs are just and only components (Keep it in mind 🙌)

React’s strong suit is handling stateful, data-driven interfaces with minimal code and minimal pain, and it has all the bells and whistles you’d expect from a modern web framework.

****

React is a javascript Library that can create both web and mobile apps ,

React js is meant to build web applications while React Native is for mobile apps

****

**How to install React?**

First, we need to have **Node js** installed on our machine since it’s the execution environment, coz we’ll be using **npm** a lot !!!

Secondo, I highly recommend one approach, and that’s using the officially recommended tool called create-react-app.

create-react-app is a command-line application, aimed at getting you up to speed with React in no time.

You start by using npx, which is an easy way to download and execute Node.js commands without installing them.

**BUT … Wait a minute**

A wise man said : *npx create-react-app* and *npm start* are a real a waste of time

**Vite.js — An Opinionated Frontend Build Tool**

**What’s Vite.js?**

Created by Evan You (also the creator of Vue.js), Vite.js is a next-generation, lightning-fast, front-end build tool that provides an amazing user experience.

Vite is a french word that means ‘fast’ and is pronounced as ‘vit’. and it is Trop vite meme 💖 😜 . We might talk about it in a later article, but let’s say that we’ll be using it to create our react App ***after the first time with npx create-react-app*** (the secret is in here !), and it will take less and less time to create our application and also to start the server with **npm run dev** command instead of **npm start**

So to recap :

**#For the first time or maybe if you're all time patient use this :**   
npx create-react-app my-app   
cd my-app   
npm start**#Later on and if you're nervous :p#If using NPM**$ npm init vite-app <project-name>  
$ cd <project-name>  
$ npm install  
$ npm run dev  
**#If using Yarn**$ yarn create vite-app <project-name>  
$ cd <project-name>  
$ yarn  
$ yarn dev

npx create-react-app will create some folders and files , I believe it’s good to know what they are exactly :

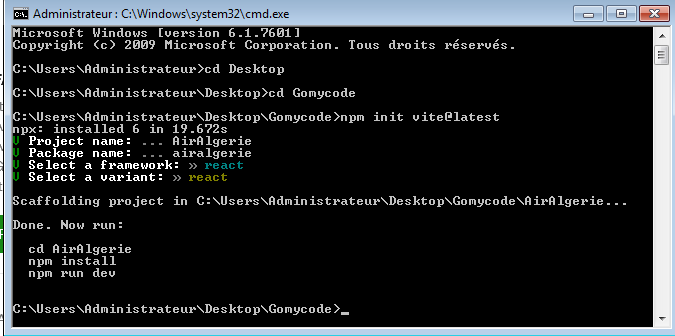
It is good to know these folders and files.

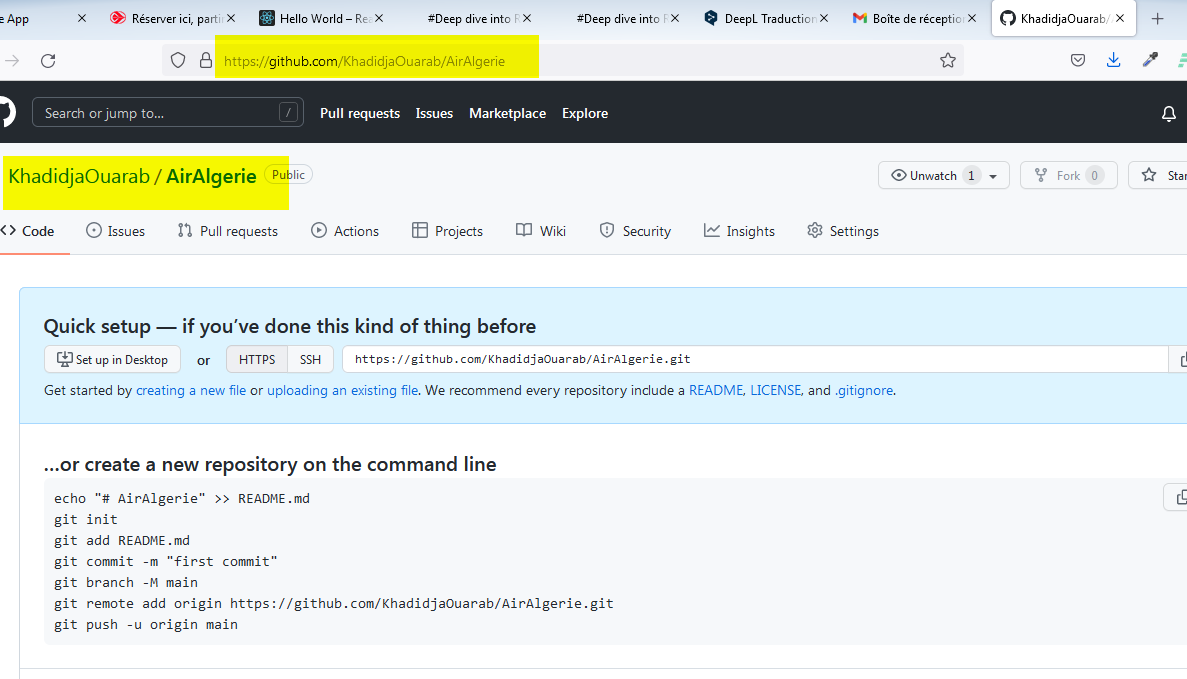
* **node\_modules** — stores all the necessary node packages of the React applications.
* **Public**
* index.html — the only HTML file we have in the entire application
* favicon.ico — an icon file
* manifest.json — is used to make the application a progressive web app
* other images — open graph images(open graph images are images which are visible when a link share on social media)
* robots.txt — information, if the website allows web scraping
* **src**
* App.css, index.css — are different CSS files
* index.js — a file which allows connecting all the components with index.html
* App.js — A file where we usually import most of the presentational components
* serviceWorker.js: is used to add progressive web app features
* setupTests.js — to write testing cases
* package.json- List of packages the applications uses
* .gitignore — React boilerplate comes with git initiated, and the .gitingore allows files and folders not to be pushed to GitHub
* [README.md](http://README.md) — Markdown file to write documentation
* yarn.lock or package-lock.json — a means to lock the version of the package

with **vite** project you’ll find a diffrent structure which is mainly the same but without the unuseful files . (so don’t panic !)

**VsCode Extentions that will save your time when coding with React :**

* The ***Bracket Pair Colorizer*** extension.
* The ***change-case*** extension.
* The ***Code Spell Checker*** extension.
* The ***Duplicate Selection*** extension.
* The ***EditorConfig for VS Code*** extension.
* The ***VSCode React Refactor*** extension.
* The ***npm Intellisense*** extension.
* The ***ESLint*** extension.
* The ***ES7 React/Redux/GraphQL/React-Native snippets*** extension.
* The ***Prettier — Code formatted*** extension.

****

****

**Files structure:**

**SRC:**

* **Components**
* **Pages : contains multiple component**
* **Assets : image, json file**

**Pour styler mes pages, j’ai plusieurs options:**

* **Dans un fichier CSS à part, après j’import le fichier puis j’utilise classname**
* **Styled component : si je veux écrire le css dans javascript 🡪 for some component**

**Il faut l’installer :**

**npm install styled-componet,**

**import styled from ‘styled-component’**

**const MyButton = styled.button ‘ color: green ‘;**

**< MyButton></ MyButton>**

* **Tailwind css for react 🡪 I choose it**

[**https://nerdcave.com/tailwind-cheat-sheet**](https://nerdcave.com/tailwind-cheat-sheet)

[**https://tailwindcss.com/docs/guides/create-react-app**](https://tailwindcss.com/docs/guides/create-react-app)

[**https://larainfo.com/blogs/how-to-install-tailwind-css-in-react**](https://larainfo.com/blogs/how-to-install-tailwind-css-in-react)

## Install Tailwind CSS

Install tailwindcss and its peer dependencies via npm, and then run the init command to generate both tailwind.config.js and postcss.config.js.

Terminal

**npm install -D tailwindcss postcss autoprefixer**

**npx tailwindcss init -p**

##  Configure your template paths

Add the paths to all of your template files in your tailwind.config.js file.

tailwind.config.js

**module.exports = { content: [ "./src/\*\*/\*.{js,jsx,ts,tsx}", ], theme: { extend: {}, }, plugins: [],}**

##  Add the Tailwind directives to your CSS

Add the @tailwind directives for each of Tailwind’s layers to your ./src/index.css file.

index.css

**@tailwind base;**

**@tailwind components;**

**@tailwind utilities;**

##  Start your build process

Run your build process with npm run start.

Terminal

**npm run build**

**Npm start if not vite, 🡪 if vite we run: npm run dev**

##  Start using Tailwind in your project

Start using Tailwind’s utility classes to style your content.

App.js

export default function App() { return ( <h1 className="text-3xl font-bold underline"> Hello world! </h1> )}