

Tomiwa Kukoyi - Frontend Web Developer

Email: Tomiwakukoyi07@gmail.com
GitHub: <https://github.com/Tomiwakukoyi>

Website: <https://tomiwa-portfolio.vercel.app>
Phone: +2347083325053

Summary

Creative, skilled and hardworking **Frontend Developer** with over three years of experience in developing high-quality, user-friendly and responsive websites and web applications. I use modern Frontend technology like React JS and Tailwind CSS to build user interface that improves the user experience. I've worked in cross-functional teams harmonically and delivered excellently. I stay up-to-date with the best practices and latest technologies in frontend development to ensure I'm always ahead and I stay delivering quality and effective web apps.

Education and Certifications

Bachelor of Science (Ed) Mathematics Education from the **University of Lagos**
CS50 Introduction to Computer Science from **Harvard X**
The Complete JavaScript Course from **Udemy**
React, The Complete Guide from **Udemy**
Python and Django Full Stack Developer **Udemy**

Work Experience

Mein Sherpa, Vue JS Frontend Developer, Feb 2023 – Present

- I worked on a team of Frontend developers and Backend developers and Database Administrators
- I built interactive and responsive frontend for the company using their Tech stack of Vue JS, Tailwind CSS
- I always worked with the Backend Developer to ensure there was proper integration between the Frontend and Backend which also made me learn a little bit of PHP
- An achievement on this job was I was able to learn Vue JS while working as prior to this job, React was my main stack but I was able to switch and be comfortable with Vue JS because of my in-depth knowledge of programming syntax
- I used GitHub everyday to pull/push code so the team and I can always have the updated code
- I developed a good relationship with my co-workers which increased productivity and delivery time.
- Some of the tools I used frequently were **Vue JS, Tailwind CSS/Flowbite, Git & GitHub, XAMPP and PHP**

Linux Jobber, React JS Frontend Developer, Apr 2022 – Oct 2022

- I worked alongside other interns to implement Frontend designs using React JS
- I implemented a fully functional login and signup form with unique inputs like a range slider
- I implemented the design of a task manager that had a drag and drop feature
- I put to good use the advantage of working in teams and it helped me speed up my development process
- I used React practices like hooks, state management and props on various other projects I implemented while working as an intern
- I always ensured I turned in my work earlier than the deadline day
- I also helped out other interns who had difficulties doing their tasks.
- Some of the tools I used were **React JS, Tailwind CSS, SASS, API** and some **External React Libraries**.

Personal Projects

Spotify Clone – <https://github.com/TomiwaKukoyi/spotify-clone>

Spotify Clone is a web App built in the similitude of the real Spotify and is really not functional in terms of playing music, at least at the moment but is mainly presenting data gotten from my own Spotify. Some of the tools I used on this project were **React JS, Context API, Material UI, CSS, Spotify API** and **Spotify Web API JS**

- A major challenge I faced was reading the data from the Spotify API as it was quite complex for me to do so. I however found a way round the problem by looking at resources and asking for help from friends.
- I also integrated Context API as my main tool for ensuring data goes round the whole components while avoiding complex props trees
- The layout of the website was styled throughout by Vanilla CSS and no external framework or library apart from Material UI which I used for the icons.

Next JS Ecommerce Website - <https://github.com/Tomiwakukoyi/shopit>

I used Next JS for the first time on a project and it wasn't that bad. The idea was to make an Ecommerce site and integrate a payment system but I left that out and did the rest things.

Some technologies I used on this project were, **Next JS, Sanity, Context API, CSS**

- I used Next JS as the main stack for this project
- I also used sanity for the first time for content management as they provide very good options and features for managing a stores content
- Also, I used Context API for the first time for managing the state of the whole project, it was quite difficult but I had help from a friend
- I used an external React library called React-Icons for the icons used throughout the project
- I enjoyed the Server-Side Rendering of next JS, it made navigating through the pages fast and seamless

React JS Disney X Clone - <https://github.com/Tomiwakukoyi/disneyclone>

The Disney X Clone is a project done to replicate the original Disney X website and the main purpose was to put into practice somethings I had learnt.

Some technologies used on this project were, **React JS, Redux-Toolkit, React-Router, Firebase, Styled-Components** and some external React libraries.

- I built the main UI using React JS and I styled it using Styled Components
- I used React-Router for routing between pages on the project
- I used Firebase Google Auth to implement login and signup with Gmail. There was an obstacle here for me because the Firebase syntax is quite different from it was before so I had to do lots of reading the docs and checking with stack overflow to fix some bugs
- I also used Firebase like my database where I stored some data and retrieved it dynamically from there
- React Slick was an external React library I used to implement a carousel

Google Clone - <https://github.com/Tomiwakukoyi/googleclone>

Google Clone is a web app made to look like google and also have the functionalities google has.

Some technologies used on this project were, **React JS, Context API, Material UI, CSS, Google Custom Search API**

- I made the google UI from scratch using React and CSS for styling
- I used material UI to get the icons used on this project
- I also created a custom hook which was quite fun and exciting to do
- I used the custom hook, useGoogleSearch to manage the calling of the API
- I used Context API to seamlessly pass important data through components
- All the data displayed on the web app are displayed in real time from the Google Custom Search API

Skills

Html, CSS, JavaScript, React JS/Next JS, Vue JS, Bootstrap, Tailwind CSS, Material UI, Styled-Components, Context API, Redux, Git/GitHub, Responsive Web-Design, Sanity, Firebase