Jarurat Care Audit Report

While Auditing the site I majorly made these observations

Less Functionality

As we move towards sites being highly interactive, in this case, the site is designed with a simple and user-friendly interface, ensuring ease of use and accessibility but limiting the site on Functionality

New Features can and should be introduced for the convenience of the user to find the solution to most of his/her problems in a single place

Fundraiser Section

Most people fighting major diseases require funds so a Fundraiser section can be introduced in such a way that anyone can apply for a fundraiser but can only be started after verification.

For this, a Payment Gateway will be integrated with donators receiving proper receipts which will be available for tax-benefits

Easy Access to The Website(Progressive Web App)

Being the Company being very young people tend to need to remember the site. A Progressive web app functionality can be introduced, allowing our site to be installable on a mobile and desktop homepage. This will improve user experience and will provide an offline access feature to the site

Hospital Recommendation According to Diseases

A Feature which will take the user's location, income, diseases, and other informative details to recommend a list of hospitals based on criteria entered by the user with all the necessary information about hospitals and doctors and contact details

Code Improvement

Added new files like manifest.json sw.js and Jarurat.js this file includes a service worker file which activates the service worker and the manifest.json file makes the site natively installable by this approach and also lets us precache the shell files and then finally dynamically cache all the loaded data

This improves the speed of the application after the first render and will also simply user experience with cache memory involved the app once in connection to the internet will be able to work completely offline

Reasoning And Approach

For all of this code, google's documentation for Pwa was used. My approach in this code improvement was to add and easily accessible and faster user experience so implemented a way so by which after the initial render the the site will perform better and will have offline functionality

As the greatest downside toa website as compared to Native App is that for even a minute information the site requires internet connection but for an app some of the features can be accessed without internet so after using this technology we compiled it in an way that if opened from the installed pwa app and some files occupied in the cache the data will be atken from cache without any internet connection required

Though this is not not something which requires React or any other advanced tech this technique is advanced in itself for it works perfectly with any application on any tech stack