**Video Link:**[**https://github.com/suhanpz/gfgdoclab/blob/main/gfg%20video%20hospital.mp4**](https://github.com/suhanpz/gfgdoclab/blob/main/gfg%20video%20hospital.mp4)

**Description:**

In this Project we aim to solve the traditional issues of hospital management. Medical website designed is to help hospitals and it is also beneficial for different patient to classify their respective disease and their concern doctors.As It is having a special System Checker option which is very beneficial for new patient.

**Problem Statement/Opportunity:**

These days, due to the pandemic, there has been a great increase in no of patients in the hospitals and one must wait a lot to get any test done. It has become difficult in the selection of suitable medicine for the type of illness and this usually takes time and makes the patient or customer wait. Therefore, the time is being wasted for the customer to be waiting.

**Project Description/Solution:**

* This is a medical system designed to manage the activities in the hospital. The new patients can request an appointment with the help of this website.
* The website confirms the appointment based on the availability of free slots for the respective doctors and the patient is informed. It is very important to maintain efficient software to handle the information of a hospital.
* Website offer online services, such as appointment scheduling, bill payment, and access to medical records, making it convenient for patients to interact with the hospital and manage their healthcare needs remotely. Facilitate communication and engagement with patients, their families, and the community.
* It can include features such as contact forms, feedback forms, live chat, and social media integration, allowing patients and visitors to connect with the hospital, provide feedback, and ask questions . This application provides a way to record this information and to access these in a simple way.
* Online Hospital Management Systems enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital. Since we are entering details of the patients electronically in the Hospital Management System, the data will be secured. Thus, processing information will be faster.
* It guarantees accurate maintenance of patient’s appointments details. It provides articles, videos, and resources on various health conditions, treatments, and preventive care, helping patients better understand their health and make informed choices .
* It provide comprehensive and up-to-date information about the hospital's services, specialties, doctors, and other important details. Patients and their families can access this information to make informed decisions about their healthcare needs, such as finding the right doctor, booking appointments, or understanding treatment options.

**BUILT WITH:**

1. HTML
2. CSS
3. JAVASCRIPT
4. REACT JS
5. BOOTSTRAP

**HTML:**

HTML, which stands for Hypertext Markup Language, is the standard markup language used to create web pages and web applications. HTML provides a set of tags and attributes that define the structure, content, and presentation of a web page. Web browsers interpret HTML code and render it into visual web pages that users can view in their web browsers.HTML is the foundation of web development and is used in conjunction with CSS and JavaScript to create web pages and web applications with different layouts, styles, and interactivity. It is a markup language that is easy to learn and essential for creating static and dynamic web content.

Some key concepts in HTML include:

* Tags: HTML uses tags to define different elements on a web page, such as headings, paragraphs, images, links, lists, forms, and more. Tags are enclosed in angle brackets (<>) and can have attributes that provide additional information about the element.
* Elements: Elements in HTML are created using tags and define the different parts of a web page, such as headings, paragraphs, images, and more.
* Attributes: Attributes provide additional information about HTML elements and are used to customize their behavior or appearance. Attributes are defined within the opening tag of an element and have a name and a value.
* Nesting: HTML elements can be nested inside other elements, allowing for the creation of complex and structured web pages.
* Semantics: HTML includes semantic elements, such as <header>, <nav>, <main>, <footer>, etc., that provide meaning and structure to the content of a web page, making it more accessible and search engine friendly.
* Links: HTML allows for creating hyperlinks that allow users to navigate between web pages, websites, or external resources.
* Forms: HTML provides form elements, such as <form>, <input>, <select>, <textarea>, etc., that allow users to input data and submit it to a server for processing.

**CSS:**

CSS, which stands for Cascading Style Sheets, is a stylesheet language used in web development to describe the presentation and appearance of HTML (Hypertext Markup Language) documents. CSS allows developers to control the layout, design, and formatting of web pages, including elements such as fonts, colors, spacing, positioning, and more.CSS uses a set of rules and selectors to target HTML elements and apply styles to them. Styles can be defined inline within HTML elements, embedded within HTML documents, or included in separate external CSS files that are linked to HTML documents.CSS is a powerful tool for web developers to control the visual appearance of web pages, and it is widely used in conjunction with HTML and JavaScript to create modern and interactive web experiences.

Some key concepts in CSS include:

* Selectors: These specify which HTML elements to target with styles. Selectors can target elements by their element type, class, ID, or other attributes.
* Properties: These define the visual properties of HTML elements, such as font-size, color, margin, padding, etc.
* Values: These specify the values of the properties, such as "red" for color or "12px" for font-size.
* Cascading: CSS follows a cascade model, which means that styles can be overridden and combined based on their specificity, order, and inheritance.
* Responsive Design: CSS allows for creating responsive designs that adapt to different screen sizes and devices, using techniques such as media queries and flexible layouts.

**JAVASCRIPT:**

JavaScript is a widely-used programming language that allows developers to add dynamic and interactive elements to websites and web applications. JavaScript is a client-side scripting language, which means it runs on the client's web browser and can be used to manipulate web page elements, handle user interactions, and perform various actions in response to events.Some key concepts in JavaScript include:

* Variables: JavaScript allows developers to declare and use variables to store and manipulate data, including numbers, strings, arrays, objects, and more.
* Operators: JavaScript supports various operators for performing arithmetic, comparison, logical, and other operations on values.
* Control Flow: JavaScript includes conditional statements (such as if/else and switch) and loops (such as for and while) for controlling the flow of program execution.
* Functions: JavaScript supports defining and using functions, which are blocks of reusable code that can be called with different arguments to perform specific tasks.
* Events: JavaScript can handle events, such as user interactions like clicks or key presses, and can respond to those events by executing code.
* DOM Manipulation: JavaScript can manipulate the Document Object Model (DOM) of a web page, allowing developers to dynamically change the content, structure, and styling of web page elements.
* AJAX: JavaScript can make asynchronous requests to servers to fetch data or update web page content without requiring a full page reload.
* Error Handling: JavaScript provides error handling mechanisms, such as try/catch blocks, for handling runtime errors and exceptions.

**REACT JS :**

React.js, often simply referred to as React, is an open-source JavaScript library for building user interfaces (UIs) and front-end web applications. Developed and maintained by Facebook, React.js is one of the most popular and widely-used libraries for front-end web development.

Some key concepts in React.js include:

* Components: React.js uses a component-based approach to building UIs, where UI elements are encapsulated as reusable components that can be composed together to create complex user interfaces.
* JSX: JSX is a syntax extension for JavaScript that allows developers to write HTML-like code directly in their JavaScript files, making it easier to define the structure and appearance of UI components in React.
* Virtual DOM: React uses a virtual DOM (Document Object Model) to optimize rendering performance by efficiently updating only the parts of the UI that need to be changed, rather than re-rendering the entire page.
* State: React components can have state, which represents the current data and state of the UI. State allows components to be dynamic and respond to user interactions or changes in data.
* Props: Props (short for properties) are used to pass data and behavior from parent components to child components in React. Props allow components to be easily composed and reused with different data.
* Lifecycle Methods: React components have lifecycle methods that are called at various stages of the component's lifecycle, allowing developers to perform actions such as initialization, updating, and cleanup.
* Events: React components can handle events, such as user interactions like clicks or input changes, and respond to them by updating state or triggering actions.
* Component Styling: React provides various ways to style components, including inline styles, CSS modules, and popular CSS-in-JS libraries like styled-components or emotion.
* React.js is a powerful and flexible library that enables developers to create complex and interactive UIs with ease. It is widely used in modern web development, and its popularity has led to a large ecosystem of tools, libraries, and community support.

**Bootstrap:**

It is a popular open-source front-end framework for building responsive and mobile-first websites and web applications. It provides a collection of CSS, JavaScript, and pre-designed UI components that can be easily integrated into web projects to create modern and visually appealing user interfaces. Bootstrap simplifies the process of designing and developing responsive websites with its grid system, typography, forms, buttons, navigation, and other UI elements. It also offers a variety of customization options, allowing developers to tailor the appearance and behavior of their web projects to suit their specific needs. Bootstrap is widely used in web development and has a large community of developers and resources available for support and documentation.

**Web hosting and domain:**

**Web Hosting**: Web hosting is a service that provides storage space and resources on a server to store and serve web pages and files. When you build a website, you need a place to store all the files, images, and content that make up your website. Web hosting providers offer servers that are connected to the internet and can host your website files, making them accessible to visitors who access your website via the internet.

**Domain**: A domain is the web address or URL (Uniform Resource Locator) that users enter in their web browsers to access your website. For example, "www.example.com" is a domain. A domain is unique and serves as the address of your website on the internet. It typically consists of a name (e.g., "example") and a domain extension (e.g., ".com", ".org", etc.).

To launch a hospital website, you will need to register a domain name and subscribe to a web hosting service. Many web hosting providers also offer domain registration services, allowing you to purchase and manage your domain and hosting from the same provider. Once you have a domain and web hosting, you can upload your website files to the hosting server, configure the domain to point to the hosting server, and make your website accessible to visitors by entering the domain in their web browsers.When choosing a web hosting and domain provider for your hospital website, consider factors such as cost, reliability, performance, security, customer support, and any specific requirements or features that your website may need. It's important to select a reputable web hosting and domain provider to ensure your website's stability, security, and performance.

**STEPS TO HOST A WEBSITE:**

To host a website for free, follow these general steps:

1.Choose a free web hosting service like GitHub Pages, Netlify, or InfinityFree.

2.Register a free domain name, or use a subdomain provided by the hosting service.

3.Upload your website files using the hosting service's file manager or FTP.

4.Configure your website, including any necessary databases or settings.

5.Test your website to ensure it's functioning properly.

6.Update your domain DNS settings to point to the hosting service's server.

7.Your website is now live and accessible on the internet, hosted for free. Keep in mind that free hosting services may have limitations in terms of storage, bandwidth, and features, and may display ads on your site.