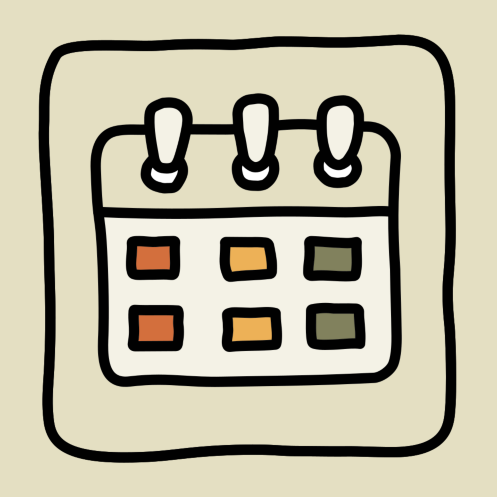
REPORT ON WEB DESIGNING

[CALENDER APP]



Submitted By: AISHWARYA N

[312321205013]

ABSTRACT:

The main objective of our project is to create a calender application that simplifies the process of managing tasks and events.

With our application, users can conveniently add,modify and delete tasks, ensuring nothing falls through the cracks.

Tools and Technology Used:

Some tools and technology were used In this project to form an website

Some of them are listed below:

1. HTML [HTML stands for Hyper Text Mark-up Language. Itr is

Used to design web pages using mark-up language.

2. CSS: [CSS stands for Cascading style sheets]. It is been used to design the web page in a creative way

3. JS:[JS stands for JavaScript].It is also used for validation process.

HTML CODE:

<!DOCTYPE html>

<html lang="en" dir="ltr">

  <head>

    <meta charset="utf-8">

    <title>CALENDER</title>

    <link rel="stylesheet" href="style.css">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Google Font Link for Icons -->

    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200">

    <script src="script.js" defer></script>

  </head>

  <body>

    <div class="wrapper">

      <header>

        <p class="current-date"></p>

        <div class="icons">

          <span id="prev" class="material-symbols-rounded">chevron\_left</span>

          <span id="next" class="material-symbols-rounded">chevron\_right</span>

        </div>

      </header>

      <div class="calendar">

        <ul class="weeks">

          <li>Sun</li>

          <li>Mon</li>

          <li>Tue</li>

          <li>Wed</li>

          <li>Thu</li>

          <li>Fri</li>

          <li>Sat</li>

        </ul>

        <ul class="days"></ul>

      </div>

    </div>

  </body>

</html>

ML CODE:

CSS CODE:

/\* Import Google font - Poppins \*/

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap');

\*{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: 'Poppins', sans-serif;

}

body{

  display: flex;

  align-items: center;

  padding: 0 10px;

  justify-content: center;

  min-height: 100vh;

  background: #e7a138;

}

.wrapper{

  width: 450px;

  background: #f5eaea;

  border-radius: 10px;

  box-shadow: 0 15px 40px rgba(0,0,0,0.12);

}

.wrapper header{

  display: flex;

  align-items: center;

  padding: 25px 30px 10px;

  justify-content: space-between;

}

header .icons{

  display: flex;

}

header .icons span{

  height: 38px;

  width: 38px;

  margin: 0 1px;

  cursor: pointer;

  color: #eb4770;

  text-align: center;

  line-height: 38px;

  font-size: 1.9rem;

  user-select: none;

  border-radius: 50%;

}

.icons span:last-child{

  margin-right: -10px;

}

header .icons span:hover{

  background: #921111;

}

header .current-date{

  font-size: 1.45rem;

  font-weight: 500;

}

.calendar{

  padding: 20px;

}

.calendar ul{

  display: flex;

  flex-wrap: wrap;

  list-style: none;

  text-align: center;

}

.calendar .days{

  margin-bottom: 20px;

}

.calendar li{

  color: #333333;

  width: calc(100% / 7);

  font-size: 1.07rem;

}

.calendar .weeks li{

  font-weight: 500;

  cursor: default;

}

.calendar .days li{

  z-index: 1;

  cursor: pointer;

  position: relative;

  margin-top: 30px;

}

.days li.inactive{

  color: #aaaaaa;

}

.days li.active{

  color: #fff;

}

.days li::before{

  position: absolute;

  content: "";

  left: 50%;

  top: 50%;

  height: 40px;

  width: 40px;

  z-index: -1;

  border-radius: 50%;

  transform: translate(-50%, -50%);

}

.days li.active::before{

  background: #b6595e;

}

.days li:not(.active):hover::before{

  background: #f2f2f2;

}

JS CODE:

const daysTag = document.querySelector(".days"),

currentDate = document.querySelector(".current-date"),

prevNextIcon = document.querySelectorAll(".icons span");

// getting new date, current year and month

let date = new Date(),

currYear = date.getFullYear(),

currMonth = date.getMonth();

// storing full name of all months in array

const months = ["January", "February", "March", "April", "May", "June", "July",

              "August", "September", "October", "November", "December"];

const renderCalendar = () => {

    let firstDayofMonth = new Date(currYear, currMonth, 1).getDay(), // getting first day of month

    lastDateofMonth = new Date(currYear, currMonth + 1, 0).getDate(), // getting last date of month

    lastDayofMonth = new Date(currYear, currMonth, lastDateofMonth).getDay(), // getting last day of month

    lastDateofLastMonth = new Date(currYear, currMonth, 0).getDate(); // getting last date of previous month

    let liTag = "";

    for (let i = firstDayofMonth; i > 0; i--) { // creating li of previous month last days

        liTag += `<li class="inactive">${lastDateofLastMonth - i + 1}</li>`;

    }

    for (let i = 1; i <= lastDateofMonth; i++) { // creating li of all days of current month

        // adding active class to li if the current day, month, and year matched

        let isToday = i === date.getDate() && currMonth === new Date().getMonth()

                     && currYear === new Date().getFullYear() ? "active" : "";

        liTag += `<li class="${isToday}">${i}</li>`;

    }

    for (let i = lastDayofMonth; i < 6; i++) { // creating li of next month first days

        liTag += `<li class="inactive">${i - lastDayofMonth + 1}</li>`

    }

    currentDate.innerText = `${months[currMonth]} ${currYear}`; // passing current mon and yr as currentDate text

    daysTag.innerHTML = liTag;

}

renderCalendar();

prevNextIcon.forEach(icon => { // getting prev and next icons

    icon.addEventListener("click", () => { // adding click event on both icons

        // if clicked icon is previous icon then decrement current month by 1 else increment it by 1

        currMonth = icon.id === "prev" ? currMonth - 1 : currMonth + 1;

        if(currMonth < 0 || currMonth > 11) { // if current month is less than 0 or greater than 11

            // creating a new date of current year & month and pass it as date value

            date = new Date(currYear, currMonth, new Date().getDate());

            currYear = date.getFullYear(); // updating current year with new date year

            currMonth = date.getMonth(); // updating current month with new date month

        } else {

            date = new Date(); // pass the current date as date value

        }

        renderCalendar(); // calling renderCalendar function

    });

});

OUTPUT:

