**Ibalita Mo!** is a real-time news update web application that fetches news articles from an external API and displays them in a user-friendly interface. With Ibalita Mo!, users can stay informed about the latest events and trends across various categories such as food, business, tech, movies, and what's popular now.

**Objectives:**

The main goal of Ibalita Mo! is to develop an intuitive web application that enables users to easily remain up to date on trends and current events. Keeping up with the latest news in today's fast-paced world can be difficult because there is so much information available. Mo Ibalita! attempts to make this procedure easier by combining news stories onto a single platform from a variety of trustworthy sources.  
  
The program offers users a customized news feed based on their interests and saves time by retrieving news articles from an external API and displays them in real-time. Ibalita Mo! offers updates on a variety of subjects, including politics, technology, entertainment, and more! is to provide its users with current and relevant information.

**Functionalities:**

These are the functionalities of the app:

* **Real-time updates:** Receive live news updates without refreshing the page.
* **Responsive design:** Accessible on both desktop and mobile devices.
* **Easy navigation:** Intuitive menu system for exploring news articles by category.
* **Search functionality:** Allows users to search for specific news topics.

**Features:**

These are the features of the app:

* **Live Updates**: Stay abreast of current events with real-time news updates, eliminating the need to manually refresh the page.
* **Cross-Device Accessibility**: Seamlessly access the platform on desktops, laptops, tablets, and smartphones, ensuring an optimal viewing experience across all devices.
* **Intuitive Navigation**: Effortlessly explore a diverse range of news articles categorized by topics such as politics, technology, entertainment, and more, facilitated by a user-friendly menu system.
* **Advanced Search**: Locate specific news topics with precision through the powerful search functionality, empowering users to find relevant articles quickly and efficiently.

**Program Structure and Code Documentation:**

**Frontend:**

**HTML**: Within the HTML code, you'll find the structure of the web pages, including elements such as <header>, <nav>, <main>, and <footer>. These elements define the layout and structure of the user interface, providing a foundation for displaying news articles and other content.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

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

    <title>Ibalita Mo!</title>

    <!--font awesome-->

    <link

        rel="stylesheet"

        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" referrerpolicy="no-referrer"

    />

    <!--Google font-->

    <link rel="preconnect" href="https://fonts.googleapis.com" />

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

    <link

        href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600&display=swap"

        rel="stylesheet"

    />

    <!---css style-->

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

</head>

<body>

<header>

    <img src="./logo.png" width="120px" />

    <div class="inputSearch desktop">

        <form id="searchForm">

            <input type="text" placeholder="Type to search..." id="searchInput" />

            <span> <i class="fa-solid fa-search"></i></span>

        </form>

    </div>

    <nav class="desktop">

        <ul>

            <li onclick="Search('food')">Food</li>

            <li onclick="Search('business')">Business</li>

            <li onclick="Search('tech')">Tech</li>

            <li onclick="Search('movies')">Movies</li>

            <li onclick="Search('popular now')">Popular Now</li>

        </ul>

    </nav>

    <div class="menuBtn">

        <i class="fa-solid fa-bars"></i>

    </div>

    <!------>

    <div class="mobile hidden">

        <nav>

            <ul>

                <li onclick="Search('food')">Food</li>

                <li onclick="Search('business')">Business</li>

                <li onclick="Search('tech')">Tech</li>

                <li onclick="Search('movies')">Movies</li>

                <li onclick="Search('popular now')">Popular Now</li>

            </ul>

        </nav>

        <div class="inputSearch">

            <form id="searchFormMobile">

                <input type="text" placeholder="Type to search..." id="searchInputMobile" />

                <span> <i class="fa-solid fa-search"></i></span>

            </form>

        </div>

    </div>

</header>

<main>

    <!-- Newsflash Section -->

    <div id="newsflash" class="newsflash">Waiting for news updates...</div>

</main>

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

<script>

    const newsflashElement = document.getElementById('newsflash');

    function showNotification(title, message) {

        if (Notification.permission === 'granted') {

            new Notification(title, { body: message });

        } else if (Notification.permission !== 'denied') {

            Notification.requestPermission().then(permission => {

                if (permission === 'granted') {

                    new Notification(title, { body: message });

                }

            });

        }

    }

    const ws = new WebSocket('ws://127.0.0.1:5500');

    ws.onopen = () => {

        console.log('Connected to WebSocket server');

        newsflashElement.textContent = 'Connected to WebSocket server';

    };

    ws.onmessage = event => {

        const data = JSON.parse(event.data);

        console.log('Message received:', data);

        newsflashElement.textContent = data.news;

        showNotification('News Update', data.news);

    };

    ws.onclose = (event) => {

        console.log('Disconnected from WebSocket server', event);

        newsflashElement.textContent = 'Disconnected from WebSocket server';

    };

    ws.onerror = error => {

        console.error('WebSocket error:', error);

        newsflashElement.textContent = 'WebSocket error: ' + error.message;

    };

</script>

</body>

</html>

**CSS**: Responsible for styling the HTML elements, CSS plays a crucial role in ensuring a visually appealing and consistent design across all pages. From setting colors, fonts, and spacing to defining layout grids and responsive breakpoints, CSS transforms static HTML content into dynamic and engaging interfaces.

\* {

    box-sizing: border-box;

}

body {

    font-family: 'Poppins', sans-serif;

    font-size: 16px;

    margin: 0;

    background: #2c2c2c;

    color: #f5f5f5;

}

header {

    background-color: #3a3a3a;

    height: 65px;

    display: flex;

    align-items: center;

    padding: 5px 10px;

    justify-content: space-between;

    box-shadow: 0px 3px 10px 0 rgba(80,119,242,0.3);

    transition: background-color 0.3s ease, box-shadow 0.3s ease;

}

header:hover {

    background-color: #4a4a4a;

    box-shadow: 0px 5px 15px 0 rgba(80,119,242,0.5);

}

nav ul {

    display: flex;

    list-style: none;

    gap: 15px;

    padding: 0;

}

nav ul li {

    cursor: pointer;

    transition: color 0.3s ease, transform 0.3s ease;

}

nav ul li:hover {

    color: #4361ee;

    transform: scale(1.1);

}

.inputSearch,

form {

    display: flex;

    align-items: center;

    height: 40px;

    width: 400px;

    border-radius: 35px;

    overflow: hidden;

    border: 1.5px solid #4361ee;

    transition: border-color 0.3s ease;

}

.inputSearch:hover,

form:hover {

    border-color: #3a49d8;

}

.inputSearch input {

    width: 100%;

    height: 100%;

    padding: 0px 15px;

    border: none;

    outline: none;

    font-size: 16px;

    background: #2c2c2c;

    color: #f5f5f5;

}

.inputSearch span {

    width: 70px;

    background-color: #4361ee;

    height: 100%;

    display: flex;

    justify-content: center;

    align-items: center;

    font-size: 18px;

    color: white;

}

.menuBtn {

    font-size: 20px;

    width: 40px;

    height: 40px;

    border-radius: 40px;

    display: flex;

    justify-content: center;

    align-items: center;

    cursor: pointer;

    transition: background-color 0.3s ease, color 0.3s ease;

}

.menuBtn:hover {

    background-color: #4361ee;

    color: #ffffff;

}

.desktop {

    display: none;

}

@media all and (min-width: 700px) {

    header {

        padding: 5px 30px;

    }

    .desktop {

        display: flex;

    }

    .menuBtn {

        display: none;

    }

}

.mobile {

    position: fixed;

    top: 70px;

    width: 100%;

    max-width: 700px;

    background-color: #2c2c2c;

    height: 100vh;

    padding: 20px;

    box-shadow: none;

    z-index: 2;

    margin-left: -10px;

    transition: transform 0.3s ease-in-out;

    transform: translateX(-100%);

}

.mobile.active {

    transform: translateX(0);

}

.mobile nav,

.mobile nav ul {

    flex-direction: column;

}

.mobile .inputSearch,

.mobile form {

    height: 35px;

}

.hidden {

    display: none;

}

@media all and (min-width: 700px) {

    .mobile {

        display: none;

    }

}

main {

    display: flex;

    flex-direction: column;

    gap: 25px;

    padding: 10px;

    margin: auto;

    animation: fadeIn 1s ease-in-out;

}

.card {

    display: flex;

    flex-direction: column;

    max-width: 400px;

    box-shadow: 7px 7px 25px 1px rgba(0, 0, 0, 0.06);

    border-radius: 10px;

    overflow: hidden;

    padding-bottom: 10px;

    background: #3a3a3a;

    color: #f5f5f5;

    transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.card:hover {

    transform: translateY(-10px);

    box-shadow: 10px 10px 30px 1px rgba(0, 0, 0, 0.1);

}

.card img {

    width: 100%;

    object-fit: cover;

    transition: transform 0.3s ease;

}

.card:hover img {

    transform: scale(1.05);

}

.card h4 {

    padding: 10px;

    font-size: 18px;

    color: #ffffff;

    background-color: #4a4a4a;

    margin: 0;

}

.card .publishbyDate {

    display: flex;

    align-items: center;

    padding: 5px 10px;

    font-size: 14px;

    color: #a5a1a1;

}

.card .publishbyDate p {

    margin: 0;

}

.card .publishbyDate span {

    padding: 0 10px;

}

.card .desc {

    padding: 10px;

    overflow: hidden;

    display: -webkit-box;

    -webkit-line-clamp: 3;

    -webkit-box-orient: vertical;

    margin: 0;

    color: #cccccc;

}

.card a {

    color: inherit;

    text-decoration: none;

}

@media all and (min-width: 700px) {

    main {

        padding: 20px;

        flex-direction: row;

        flex-wrap: wrap;

        justify-content: center;

    }

    .card {

        margin: 10px;

    }

}

@keyframes fadeIn {

    from {

        opacity: 0;

        transform: translateY(10px);

    }

    to {

        opacity: 1;

        transform: translateY(0);

    }

}

.newsflash {

    font-size: 18px;

    text-align: center;

    padding: 15px;

    background-color: #4a4a4a;

    border-radius: 5px;

    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

    transition: background-color 0.3s ease;

}

.newsflash:hover {

    background-color: #5a5a5a;

}

**JavaScript**: As the language of interactivity and dynamism, JavaScript breathes life into the applicationthe JavaScript code in the provided snippet primarily handles real-time updates, data fetching from external APIs, rendering news articles, search functionality, notification display, and menu button functionality, contributing to the dynamic and interactive nature of the Ibalita Mo! web application.

const API\_KEY = "5dd162e032bd435c8810227918f2c046";

const url = "https://newsapi.org/v2/everything?apiKey=";

async function fetchData(query) {

    try {

        const res = await fetch(`${url}${API\_KEY}&q=${query}`);

        const data = await res.json();

        if (res.ok) {

            return data;

        } else {

            throw new Error(data.message || "Error fetching data");

        }

    } catch (error) {

        console.error("Error fetching data:", error);

        return null;

    }

}

fetchData("all").then(data => {

    if (data) {

        renderMain(data.articles);

    } else {

        console.error("No data available");

    }

});

let mobilemenu = document.querySelector(".mobile");

let menuBtn = document.querySelector(".menuBtn");

menuBtn.addEventListener("click", () => {

    mobilemenu.classList.toggle("hidden");

});

function renderMain(arr) {

    if (!Array.isArray(arr)) {

        console.error("Expected an array of articles but got:", arr);

        return;

    }

    let mainHTML = '';

    for (let i = 0; i < arr.length; i++) {

        if (arr[i].urlToImage) {

            mainHTML += `<div class="card">

                            <a href="${arr[i].url}">

                                <img src="${arr[i].urlToImage}" alt="News Image" loading="lazy" />

                                <h4>${arr[i].title}</h4>

                                <div class="publishbyDate">

                                    <p>${arr[i].source.name}</p>

                                    <span>•</span>

                                    <p>${new Date(arr[i].publishedAt).toLocaleDateString()}</p>

                                </div>

                                <div class="desc">

                                    ${arr[i].description}

                                </div>

                            </a>

                         </div>`;

        }

    }

    document.querySelector("main").innerHTML = mainHTML;

}

const searchBtn = document.getElementById("searchForm");

const searchBtnMobile = document.getElementById("searchFormMobile");

const searchInputMobile = document.getElementById("searchInputMobile");

const searchInput = document.getElementById("searchInput");

searchBtn.addEventListener("submit", async (e) => {

    e.preventDefault();

    console.log("Searching for:", searchInput.value);

    const data = await fetchData(searchInput.value);

    if (data) {

        renderMain(data.articles);

    } else {

        console.error("No data available for search query:", searchInput.value);

    }

});

searchBtnMobile.addEventListener("submit", async (e) => {

    e.preventDefault();

    console.log("Searching for (mobile):", searchInputMobile.value);

    const data = await fetchData(searchInputMobile.value);

    if (data) {

        renderMain(data.articles);

    } else {

        console.error("No data available for search query:", searchInputMobile.value);

    }

});

async function Search(query) {

    console.log("Searching for:", query);

    const data = await fetchData(query);

    if (data) {

        renderMain(data.articles);

    } else {

        console.error("No data available for search query:", query);

    }

}

**Backend:**

**WebSocket server:** powers real-time communication between the client and server, ensuring live news updates without page refreshes. Built using Node.js and the WebSocket protocol, it enables seamless bidirectional communication, allowing the server to instantly push updates to connected clients. By leveraging this technology, Ibalita Mo! ensures smooth and uninterrupted news browsing, enhancing the user experience.

const WebSocket = require('ws');

const wss = new WebSocket.Server({ port: 5500 });

wss.on('connection', ws => {

    console.log('Client connected');

    // Send a message to the client every 5 seconds

    const sendInterval = setInterval(() => {

        if (ws.readyState === WebSocket.OPEN) {

            ws.send(JSON.stringify({ news: 'This is a real-time news update!' }));

        }

    }, 5000);

    ws.on('error', (error) => {

        console.log('WebSocket error: ', error);

    });

    ws.on('close', () => {

        clearInterval(sendInterval);

        console.log('Client disconnected');

    });

});

console.log('WebSocket server is running on ws://127.0.0.1:5500');