<!DOCTYPE html>

<html lang="zh-TW">

  <head>

    <meta charset="UTF-8" />

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

    <title>寄憶之海 | The Harbor of Memories</title>

    <link

      href="https://fonts.googleapis.com/css2?family=Noto+Serif+TC:wght@400;500;600;700&display=swap"

      rel="stylesheet"

    />

    <style>

      /\* --- 您的原始 CSS 開始 --- \*/

      \* {

        margin: 0;

        padding: 0;

        box-sizing: border-box;

      }

      body,

      html {

        height: 100%;

        width: 100%;

        font-family: "Noto Serif TC", serif;

        color: white;

        overflow: hidden;

      }

      body {

        position: relative;

        background-color: #0c141f; /\* 深藍色底色，防止圖片載入延遲時白屏 \*/

      }

      .background-container {

        position: fixed;

        top: -5%;

        left: -5%;

        width: 110%;

        height: 110%;

        z-index: -2;

        background-image: url("letter.jpg");

        background-position: center;

        background-size: cover;

        animation: pan-background 120s linear infinite alternate;

      }

      @keyframes pan-background {

        0% {

          transform: scale(1) translate(0, 0);

        }

        100% {

          transform: scale(1.05) translate(2%, 2%);

        }

      }

      .container {

        height: 100%;

        width: 100%;

        display: flex;

        flex-direction: column;

        justify-content: center;

        align-items: center;

        padding: 20px;

        position: relative;

        background-color: rgba(0, 0, 0, 0.2);

        transition: opacity 0.5s ease-out;

      }

      .content {

        text-align: center;

        max-width: 90%;

        margin-bottom: 25vh;

      }

      .main-title {

        font-size: 3.2rem;

        font-weight: 700;

        margin-bottom: 15px;

        letter-spacing: 5px;

        text-shadow: 0 0 5px rgba(255, 255, 255, 0.7),

          0 0 10px rgba(255, 255, 255, 0.5), 0 0 20px rgba(173, 216, 230, 0.5);

      }

      .sub-title {

        font-size: 1rem;

        font-weight: 500;

        margin-bottom: 40px;

        letter-spacing: 1px;

        text-shadow: 0 0 5px rgba(255, 255, 255, 0.6);

      }

      .text-content {

        font-size: 0.95rem;

        line-height: 1.6;

        margin-bottom: 15px;

        letter-spacing: 1px;

      }

      .buttons {

        display: flex;

        flex-direction: row;

        justify-content: center;

        gap: 15px;

        width: 100%;

        position: absolute;

        bottom: 30vh;

        left: 50%;

        transform: translateX(-50%);

      }

      button {

        background-color: rgba(255, 255, 255, 0.2);

        border: 1px solid rgba(255, 255, 255, 0.5);

        color: white;

        padding: 12px 20px;

        border-radius: 25px;

        cursor: pointer;

        font-family: "Noto Serif TC", serif;

        font-size: 0.9rem;

        transition: all 0.3s ease;

        backdrop-filter: blur(5px);

      }

      button:hover {

        background-color: rgba(255, 255, 255, 0.3);

        transform: translateY(-2px);

      }

      .copyright {

        position: absolute;

        bottom: 20px;

        font-size: 0.7rem;

        opacity: 0.7;

        width: 100%;

        text-align: center;

      }

      .page {

        position: absolute;

        top: 0;

        left: 0;

        width: 100%;

        height: 100%;

        display: flex;

        justify-content: center;

        align-items: center;

        flex-direction: column;

        padding: 20px;

        background-color: rgba(0, 0, 0, 0.2);

        opacity: 0;

        visibility: hidden;

        transition: opacity 0.5s ease-in-out, visibility 0.5s;

        backdrop-filter: blur(5px);

      }

      .page.active {

        opacity: 1;

        visibility: visible;

      }

      .page-content {

        width: 100%;

        max-width: 600px;

        text-align: center;

      }

      textarea {

        background-image: url("removebg.png");

        background-color: transparent;

        background-size: 100% 100%;

        background-repeat: no-repeat;

        background-position: center;

        border: none;

        box-shadow: none;

        filter: drop-shadow(0 5px 15px rgba(0, 0, 0, 0.4));

        color: #4a3c2d;

        font-family: "Noto Serif TC", serif;

        font-size: 1rem;

        resize: none;

        overflow-wrap: break-word;

        width: 85vw;

        max-width: 380px;

        height: auto;

        min-height: 0;

        aspect-ratio: 3 / 4;

        margin-left: auto;

        margin-right: auto;

        padding: 12% 15%;

        margin-bottom: 10px;

      }

      textarea::placeholder {

        color: rgba(74, 60, 45, 0.6);

      }

      textarea:focus {

        outline: none;

        border: none;

      }

      .page-buttons {

        display: flex;

        gap: 15px;

        justify-content: center;

      }

      .letter-display {

        width: 100%;

        min-height: 50vh;

        background-color: rgba(0, 0, 0, 0.4);

        border: 1px solid rgba(255, 255, 255, 0.3);

        border-radius: 10px;

        padding: 25px;

        font-size: 1.1rem;

        line-height: 1.8;

        white-space: pre-wrap;

        margin-bottom: 20px;

        text-align: left;

        color: #e5e7eb;

        overflow-wrap: break-word;

        transition: all 0.5s ease;

      }

      .letter-display.system-message {

        display: flex;

        justify-content: center;

        align-items: center;

        text-align: center;

      }

      .letter-display.parchment-bg {

        background-image: url("removebg.png");

        background-color: transparent;

        background-size: 100% 100%;

        background-repeat: no-repeat;

        background-position: center;

        border: none;

        box-shadow: none;

        filter: drop-shadow(0 5px 15px rgba(0, 0, 0, 0.4));

        color: #4a3c2d;

        text-align: left;

        width: 85vw;

        max-width: 380px;

        height: auto;

        min-height: 0;

        aspect-ratio: 3 / 4;

        margin-left: auto;

        margin-right: auto;

        padding: 12% 15%;

      }

      .bottle-animation {

        height: 50vh;

        display: flex;

        justify-content: center;

        align-items: center;

        position: relative;

        margin-bottom: 20px;

      }

      .wave {

        width: 100px;

        height: 100px;

        border-radius: 50%;

        border: 2px solid rgba(255, 255, 255, 0.8);

        position: absolute;

        animation: wave-animation 2.5s ease-out infinite;

      }

      .wave:nth-child(2) {

        animation-delay: 0.5s;

      }

      .wave:nth-child(3) {

        animation-delay: 1s;

      }

      @keyframes wave-animation {

        0% {

          transform: scale(0);

          opacity: 1;

        }

        100% {

          transform: scale(3);

          opacity: 0;

        }

      }

      #message-box {

        height: 20px;

        margin-top: 15px;

        transition: opacity 0.5s;

      }

      .message-success {

        color: #6ee7b7;

      }

      .message-error {

        color: #9ceaef;

        font-weight: bold;

        text-shadow: 0 0 8px rgba(156, 234, 239, 0.8);

      }

      .char-counter {

        width: 100%;

        text-align: right;

        font-size: 0.8rem;

        color: #eae1c9; /\* 修改：改成羊皮紙的米白色 \*/

        font-weight: bold;

        margin-bottom: 15px;

        padding-right: 5px;

      }

      .char-counter.limit-reached {

        color: #c94a4a;

      }

      #dynamic-effects-container {

        position: fixed;

        top: 0;

        left: 0;

        width: 100%;

        height: 100%;

        z-index: -1;

        pointer-events: none;

      }

      .star {

        position: absolute;

        background-color: white;

        border-radius: 50%;

        animation: twinkle 5s infinite ease-in-out;

      }

      @keyframes twinkle {

        0%,

        100% {

          opacity: 0.2;

        }

        50% {

          opacity: 1;

        }

      }

      #shooting-star {

        position: absolute;

        width: 2px;

        height: 80px;

        background: linear-gradient(

          to top,

          rgba(255, 255, 255, 0) 0%,

          rgba(255, 255, 255, 0.8) 100%

        );

        opacity: 0;

        transform: rotate(20deg);

      }

      #shooting-star.animate {

        animation: shoot 2.5s ease-in-out;

      }

      @keyframes shoot {

        0% {

          opacity: 1;

          transform: translate(-100vw, 100vh) rotate(20deg);

        }

        100% {

          opacity: 0;

          transform: translate(100vw, -100vh) rotate(20deg);

        }

      }

      #music-control {

        position: fixed;

        bottom: 20px;

        right: 20px;

        width: 40px;

        height: 40px;

        background-color: rgba(255, 255, 255, 0.2);

        border: 1px solid rgba(255, 255, 255, 0.5);

        border-radius: 50%;

        display: flex;

        justify-content: center;

        align-items: center;

        cursor: pointer;

        z-index: 50;

        backdrop-filter: blur(5px);

      }

      #music-control svg {

        width: 20px;

        height: 20px;

        fill: white;

      }

      body.initial-view .content {

        margin-bottom: 0;

      }

      body.initial-view .sub-title,

      body.initial-view .text-content {

        opacity: 0;

        pointer-events: none;

        font-size: 0;

        line-height: 0;

        margin-bottom: 0;

      }

      body.initial-view .buttons,

      body.initial-view .copyright,

      body.initial-view #music-control {

        opacity: 0;

        pointer-events: none;

      }

      .content,

      .sub-title,

      .text-content,

      .buttons,

      .copyright,

      #music-control {

        transition: all 1.2s cubic-bezier(0.25, 1, 0.5, 1);

      }

      .sub-title,

      .text-content {

        transition-delay: 0.4s;

      }

      .buttons,

      .copyright,

      #music-control {

        transition-delay: 0.6s;

      }

      @media (min-width: 768px) {

        .background-container {

          background-image: url("banner.jpg");

        }

        .mobile-break {

          display: none;

        }

        .letter-display.parchment-bg,

        textarea {

          width: 60vw;

          max-width: 550px;

          aspect-ratio: 4 / 3; /\* 電腦版橫式比例 \*/

          padding: 10% 12%; /\* 電腦版內邊距 \*/

        }

      }

    </style>

  </head>

  <body class="initial-view">

    <audio id="background-music" loop>

      <source src="background-music.mp3" type="audio/mpeg" />

      您的瀏覽器不支援音訊播放。

    </audio>

    <div id="music-control" style="opacity: 0">

      <svg

        id="icon-sound-on"

        xmlns="http://www.w3.org/2000/svg"

        viewBox="0 0 24 24"

      >

        <path

          d="M3 9v6h4l5 5V4L7 9H3zm13.5 3c0-1.77-1.02-3.29-2.5-4.03v8.05c1.48-.73 2.5-2.25 2.5-4.02zM14 3.23v2.06c2.89.86 5 3.54 5 6.71s-2.11 5.85-5 6.71v2.06c4.01-.91 7-4.49 7-8.77s-2.99-7.86-7-8.77z"

        />

      </svg>

      <svg

        id="icon-sound-off"

        style="display: none"

        xmlns="http://www.w3.org/2000/svg"

        viewBox="0 0 24 24"

      >

        <path

          d="M16.5 12c0-1.77-1.02-3.29-2.5-4.03v2.21l2.45 2.45c.03-.2.05-.41.05-.63zm2.5 0c0 .94-.2 1.82-.54 2.64l1.51 1.51C20.63 14.91 21 13.5 21 12c0-4.28-2.99-7.86-7-8.77v2.06c2.89.86 5 3.54 5 6.71zM4.27 3L3 4.27 7.73 9H3v6h4l5 5v-6.73l4.25 4.25c-.67.52-1.42.93-2.25 1.18v2.06c1.38-.31 2.63-.95 3.69-1.81L19.73 21 21 19.73l-9-9L4.27 3zM12 4L9.91 6.09 12 8.18V4z"

        />

      </svg>

    </div>

    <div class="background-container"></div>

    <div id="dynamic-effects-container">

      <div id="shooting-star"></div>

    </div>

    <div id="home-container" class="container">

      <div class="content">

        <h1 class="main-title">寄憶之海</h1>

        <h2 class="sub-title">一個只屬於你與偶然的療癒角落</h2>

        <p class="text-content">

          寫一封信，<br class="mobile-break" />給過去的遺憾、給未來的自己，<br

            class="mobile-break"

          />或給一個素未謀面的溫柔靈魂。

        </p>

        <p class="text-content">

          在這裡，每一封信都會<br

            class="mobile-break"

          />被溫柔地保存在時光的長河裡，<br

            class="mobile-break"

          />等待一位有緣人偶然拾起。

        </p>

      </div>

      <div class="buttons">

        <button id="send-button">投遞一封回憶信</button>

        <button id="receive-button">拾起一個漂流瓶</button>

      </div>

      <div class="copyright">© 2025 The Harbor of Memories by Vanessa</div>

    </div>

    <div id="write-page" class="page">

      <div class="page-content">

        <textarea

          id="letter-input"

          placeholder="寫給過去、未來，或某個再也無法對話的溫柔靈魂…"

        ></textarea>

        <div id="char-counter" class="char-counter">0 / 200</div>

        <div class="page-buttons">

          <button id="commit-send-button">投入大海</button>

          <button class="back-button">返回</button>

        </div>

        <div id="message-box"></div>

      </div>

    </div>

    <div id="read-page" class="page">

      <div class="page-content">

        <div

          id="animation-container"

          class="bottle-animation"

          style="display: none"

        >

          <div class="wave"></div>

          <div class="wave"></div>

          <div class="wave"></div>

        </div>

        <div id="letter-display" class="letter-display"></div>

        <div class="page-buttons">

          <button id="pick-another-button">拾起下一個</button>

          <button class="back-button">返回</button>

        </div>

      </div>

    </div>

    <script type="module">

      import { initializeApp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";

      import {

        getAuth,

        signInAnonymously,

        signInWithCustomToken,

      } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-auth.js";

      import {

        getFirestore,

        collection,

        addDoc,

        getDocs,

        doc,

        deleteDoc,

        serverTimestamp,

        limit,

        query,

      } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore.js";

      document.addEventListener("DOMContentLoaded", async () => {

        let db, auth, lettersCollectionRef;

        try {

          const firebaseConfig = JSON.parse(

            typeof \_\_firebase\_config !== "undefined" ? \_\_firebase\_config : "{}"

          );

          const app = initializeApp(firebaseConfig);

          db = getFirestore(app);

          auth = getAuth(app);

          if (

            typeof \_\_initial\_auth\_token !== "undefined" &&

            \_\_initial\_auth\_token

          ) {

            await signInWithCustomToken(auth, \_\_initial\_auth\_token);

          } else {

            await signInAnonymously(auth);

          }

          const appId =

            typeof \_\_app\_id !== "undefined" ? \_\_app\_id : "default-app-id";

          lettersCollectionRef = collection(

            db,

            `artifacts/${appId}/public/data/letters`

          );

        } catch (error) {

          console.error("Firebase 初始化失敗:", error);

        }

        const homeContainer = document.getElementById("home-container");

        const writePage = document.getElementById("write-page");

        const readPage = document.getElementById("read-page");

        const sendButton = document.getElementById("send-button");

        const receiveButton = document.getElementById("receive-button");

        const backButtons = document.querySelectorAll(".back-button");

        const letterInput = document.getElementById("letter-input");

        const commitSendButton = document.getElementById("commit-send-button");

        const messageBox = document.getElementById("message-box");

        const animationContainer = document.getElementById(

          "animation-container"

        );

        const letterDisplay = document.getElementById("letter-display");

        const pickAnotherButton = document.getElementById(

          "pick-another-button"

        );

        const charCounter = document.getElementById("char-counter");

        const effectsContainer = document.getElementById(

          "dynamic-effects-container"

        );

        const shootingStar = document.getElementById("shooting-star");

        const bgMusic = document.getElementById("background-music");

        const musicControl = document.getElementById("music-control");

        const iconSoundOn = document.getElementById("icon-sound-on");

        const iconSoundOff = document.getElementById("icon-sound-off");

        const MAX\_CHARS = 200;

        let pickCounter = 0;

        const noLetterMessages = [

          "海面很平靜，還沒有漂流瓶...",

          "大海現在似乎沒有故事想說。",

          "也許下一個漂流瓶正在來的路上...",

        ];

        function showPage(pageElement) {

          homeContainer.style.opacity = "0";

          writePage.classList.remove("active");

          readPage.classList.remove("active");

          if (pageElement) {

            pageElement.classList.add("active");

          } else {

            homeContainer.style.opacity = "1";

          }

        }

        async function saveLetter() {

          if (!lettersCollectionRef) {

            console.error("資料庫未連接，無法儲存信件。");

            messageBox.textContent = "投遞失敗，請稍後再試。";

            messageBox.classList.remove("message-success");

            messageBox.classList.add("message-error");

            setTimeout(() => (messageBox.textContent = ""), 3000);

            return;

          }

          const content = letterInput.value.trim();

          if (content === "") {

            messageBox.textContent = "信件不能是空的喔！";

            messageBox.classList.remove("message-success");

            messageBox.classList.add("message-error");

            setTimeout(() => (messageBox.textContent = ""), 3000);

            return;

          }

          commitSendButton.disabled = true;

          try {

            await addDoc(lettersCollectionRef, {

              content: content,

              createdAt: serverTimestamp(),

            });

            letterInput.value = "";

            updateCharCounter();

            messageBox.textContent = "您的回憶已投入大海。";

            messageBox.classList.remove("message-error");

            messageBox.classList.add("message-success");

            setTimeout(() => (messageBox.textContent = ""), 3000);

          } catch (error) {

            console.error("儲存信件失敗:", error);

            messageBox.textContent = "投遞失敗，請稍後再試。";

            messageBox.classList.remove("message-success");

            messageBox.classList.add("message-error");

          } finally {

            commitSendButton.disabled = false;

          }

        }

        async function displayContent() {

          letterDisplay.style.display = "none";

          animationContainer.style.display = "flex";

          setTimeout(async () => {

            animationContainer.style.display = "none";

            let isSystemMessage = false;

            let contentToShow = "";

            if (pickCounter === 3) {

              contentToShow = "偷偷傳給朋友，看能不能看到他的心事?";

              isSystemMessage = true;

              pickCounter = 0;

            } else if (!lettersCollectionRef) {

              console.error("資料庫未連接，無法讀取信件。");

              contentToShow = "大海的訊息似乎迷路了...";

              isSystemMessage = true;

            } else {

              try {

                const q = query(lettersCollectionRef, limit(20));

                const querySnapshot = await getDocs(q);

                const letters = [];

                querySnapshot.forEach((doc) => {

                  letters.push({ id: doc.id, ...doc.data() });

                });

                if (letters.length > 0) {

                  const randomIndex = Math.floor(

                    Math.random() \* letters.length

                  );

                  const pickedLetter = letters[randomIndex];

                  contentToShow = pickedLetter.content;

                  isSystemMessage = false;

                  await deleteDoc(

                    doc(

                      db,

                      `artifacts/${appId}/public/data/letters`,

                      pickedLetter.id

                    )

                  );

                } else {

                  const randomIndex = Math.floor(

                    Math.random() \* noLetterMessages.length

                  );

                  contentToShow = noLetterMessages[randomIndex];

                  isSystemMessage = true;

                }

              } catch (error) {

                console.error("讀取信件失敗:", error);

                contentToShow = "大海的訊息似乎迷路了...";

                isSystemMessage = true;

              }

            }

            if (isSystemMessage) {

              letterDisplay.classList.add("system-message");

              letterDisplay.classList.remove("parchment-bg");

              letterDisplay.style.display = "flex";

            } else {

              letterDisplay.classList.remove("system-message");

              letterDisplay.classList.add("parchment-bg");

              letterDisplay.style.display = "block";

            }

            letterDisplay.textContent = contentToShow;

          }, 2500);

        }

        function updateCharCounter() {

          const currentLength = letterInput.value.length;

          charCounter.textContent = `${currentLength} / ${MAX\_CHARS}`;

          if (currentLength > MAX\_CHARS) {

            letterInput.value = letterInput.value.substring(0, MAX\_CHARS);

            charCounter.textContent = `${MAX\_CHARS} / ${MAX\_CHARS}`;

            charCounter.classList.add("limit-reached");

          } else {

            charCounter.classList.remove("limit-reached");

          }

        }

        function createStars() {

          const numStars = 50;

          for (let i = 0; i < numStars; i++) {

            let star = document.createElement("div");

            star.className = "star";

            star.style.top = `${Math.random() \* 100}%`;

            star.style.left = `${Math.random() \* 100}%`;

            const size = Math.random() \* 2 + 1;

            star.style.width = `${size}px`;

            star.style.height = `${size}px`;

            star.style.animationDelay = `${Math.random() \* 5}s`;

            effectsContainer.appendChild(star);

          }

        }

        function triggerShootingStar() {

          shootingStar.classList.remove("animate");

          setTimeout(() => {

            shootingStar.style.top = `${Math.random() \* 60 - 10}%`;

            shootingStar.style.left = `${Math.random() \* 80 - 40}%`;

            shootingStar.classList.add("animate");

          }, 100);

        }

        function toggleMusic() {

          if (bgMusic.paused) {

            bgMusic.play();

            iconSoundOn.style.display = "block";

            iconSoundOff.style.display = "none";

          } else {

            bgMusic.pause();

            iconSoundOn.style.display = "none";

            iconSoundOff.style.display = "block";

          }

        }

        function initExperience() {

          bgMusic

            .play()

            .catch((e) => console.log("瀏覽器需要使用者互動才能播放音樂。"));

          document.body.classList.remove("initial-view");

          document.body.removeEventListener("click", initExperience);

          document.body.removeEventListener("touchstart", initExperience);

        }

        sendButton.addEventListener("click", () => showPage(writePage));

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

          pickCounter = 0;

          showPage(readPage);

          pickCounter++;

          displayContent();

        });

        backButtons.forEach((button) => {

          button.addEventListener("click", () => showPage(null));

        });

        commitSendButton.addEventListener("click", saveLetter);

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

          pickCounter++;

          displayContent();

        });

        letterInput.addEventListener("input", updateCharCounter);

        musicControl.addEventListener("click", toggleMusic);

        createStars();

        setInterval(triggerShootingStar, 30000);

        document.body.addEventListener("click", initExperience);

        document.body.addEventListener("touchstart", initExperience);

      });

    </script>

  </body>

</html>