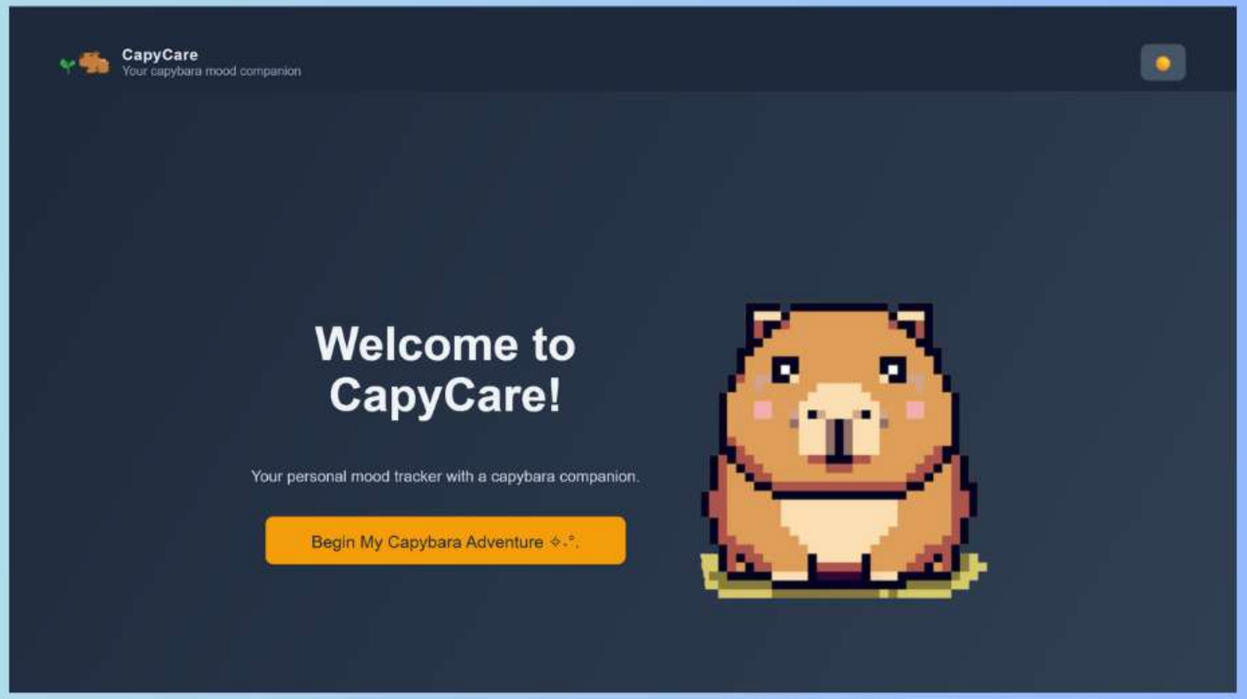
CapyCare Home Page

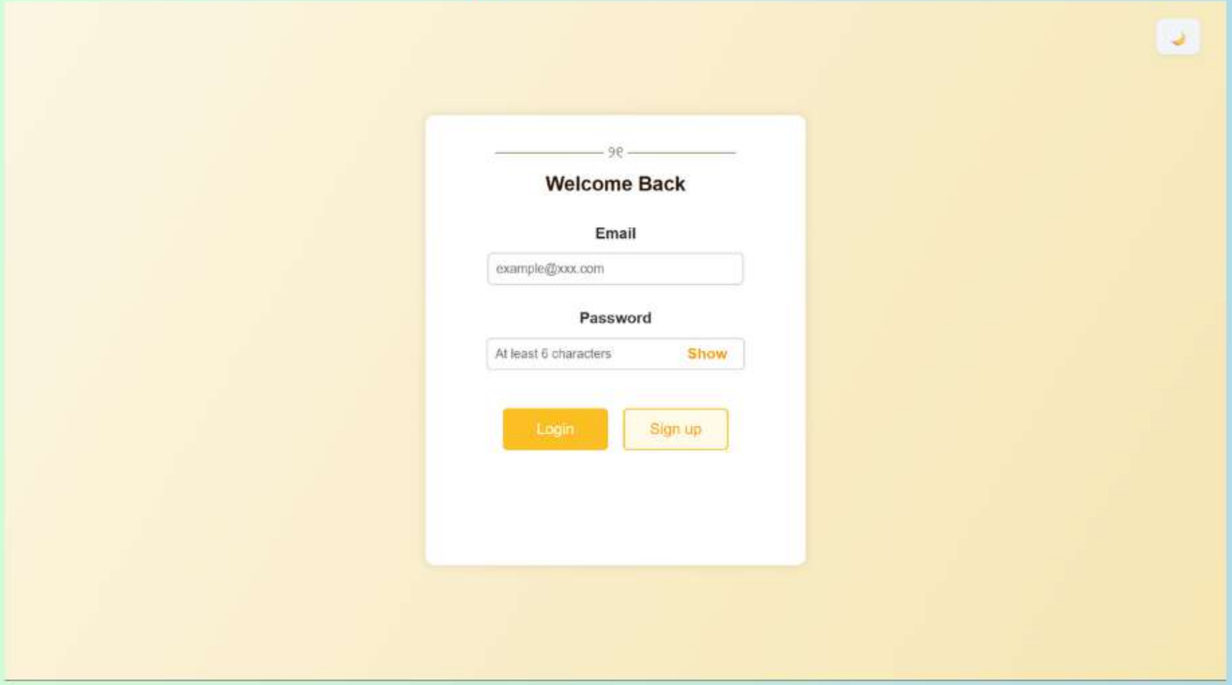
Light Mode

Dark Mode

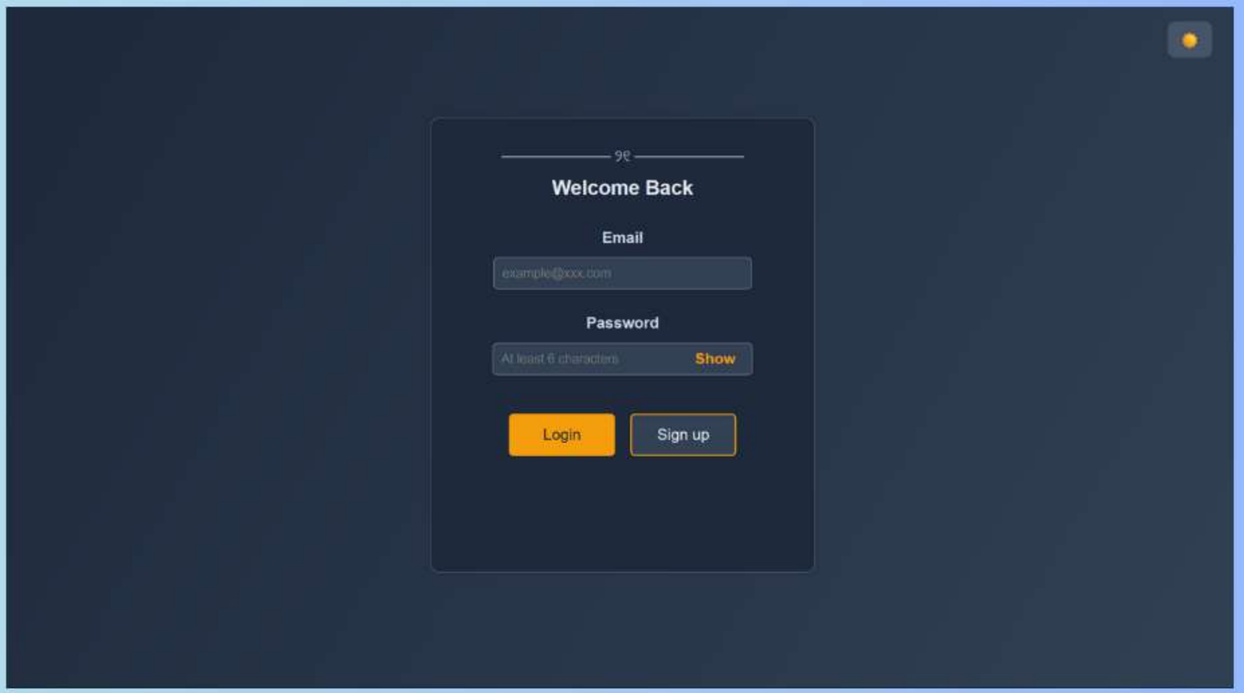


CapyCare Login Page

Light Mode

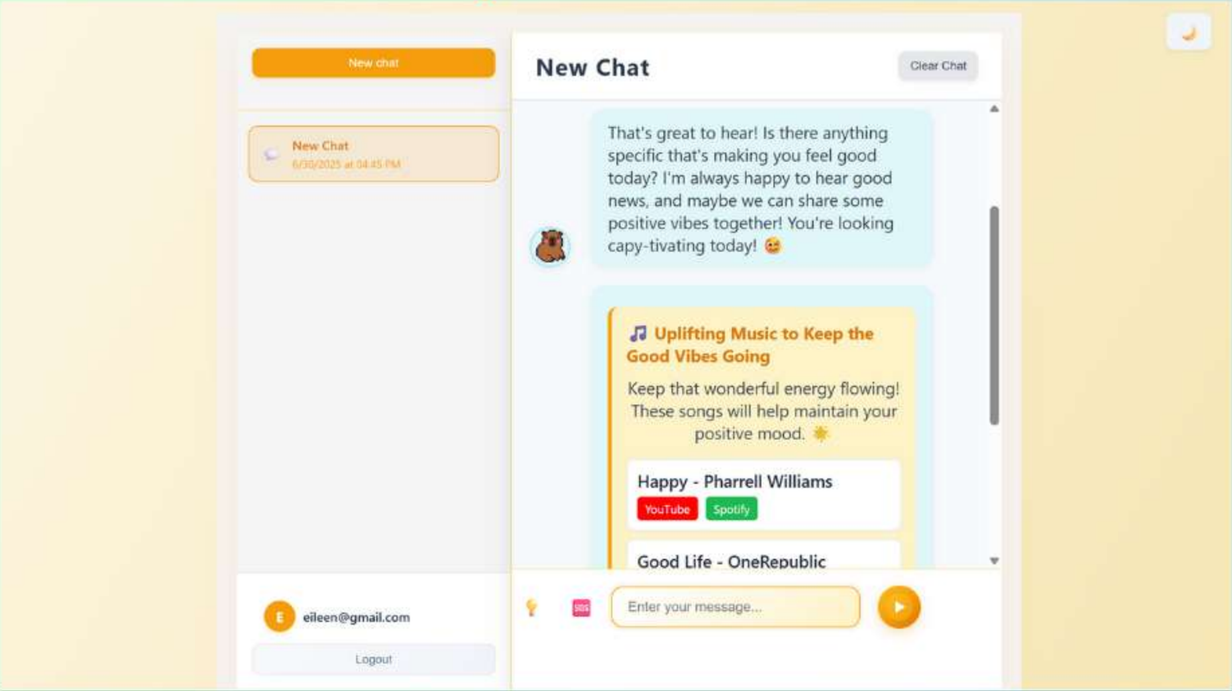


Dark Mode

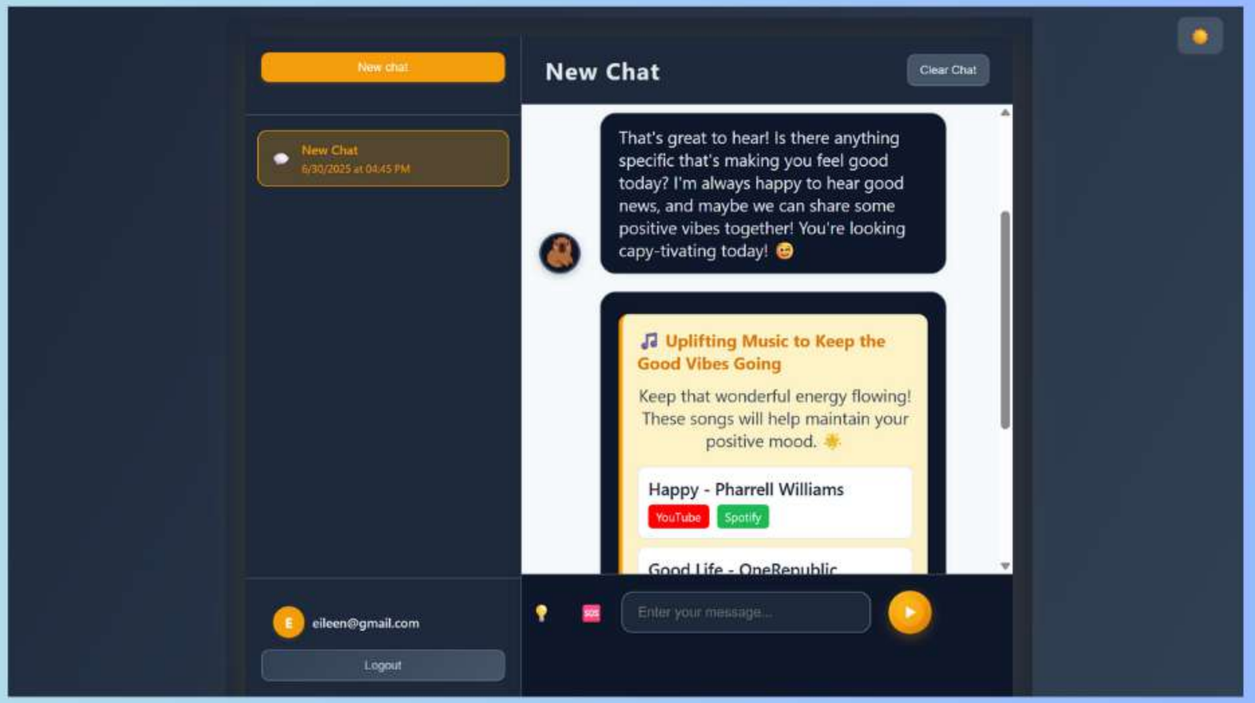


CapyCare Main Page

Light Mode



Dark Mode



Code

```javascript
document.addEventListener('DOMContentLoaded', () => {
    const loginBtn = document.getElementById('login-btn');
    const signupBtn = document.getElementById('signup-btn');
    const emailInput = document.getElementById('email');
    const passwordInput = document.getElementById('password');
    const errorMessage = document.getElementById('error-message');

    if (signupBtn) {
        // Handle Sign Up
        signupBtn.addEventListener('click', () => {
            const email = emailInput.value;
            const password = passwordInput.value;

            if (!email || !password) {
                errorMessage.textContent = 'Please enter email and password.';
                return;
            }

            auth.createUserWithEmailAndPassword(email, password)
                .then((userCredential) => {
                    // Signed in
                    console.log('Signed up:', userCredential.user);

                    // Store user info in localStorage
                    const userInfo = {
                        email: userCredential.user.email,
                        uid: userCredential.user.uid
                    };
                    localStorage.setItem('capycare_user', JSON.stringify(userInfo));
                    localStorage.setItem('capycare_last_login', Date.now().toString());
                    window.location.href = 'mood.html';
                })
                .catch((error) => {
                    if (error.code === 'auth/email-already-in-use') {
                        errorMessage.textContent = 'This email is already registered. Please log in.';
                    } else if (error.code === 'auth/weak-password') {
                        errorMessage.textContent = 'Password should be at least 6 characters.';
                    } else {
                        console.error('Signup Error:', error);
                        errorMessage.textContent = 'An error occurred during sign up. Please try again.';
                    }
                });
        });
    }

    if (loginBtn) {
        // Handle Login
        loginBtn.addEventListener('click', () => {
            const email = emailInput.value;
            const password = passwordInput.value;

            if (!email || !password) {
                errorMessage.textContent = 'Please enter email and password.';
                return;
            }

            auth.signInWithEmailAndPassword(email, password)
                .then((userCredential) => {
                    // Signed in
                    console.log('Logged in:', userCredential.user);

                    // Store user info in localStorage
                    const userInfo = {
                        email: userCredential.user.email,
                        uid: userCredential.user.uid
                    };
                    localStorage.setItem('capycare_user', JSON.stringify(userInfo));
                    localStorage.setItem('capycare_last_login', Date.now().toString());
                    window.location.href = 'mood.html';
                })
                .catch((error) => {
                    // This generic error is returned for security reasons (to prevent email enumeration).
                    if (error.code === 'auth/invalid-credential' ||
                        error.code === 'auth/invalid-login-credentials' ||
                        (error.message && error.message.includes('INVALID_LOGIN_CREDENTIALS'))) {
                        errorMessage.textContent = 'Invalid email or password. Please check your credentials or sign up.';
                    } else {
                        console.error('Login Error:', error);
                        errorMessage.textContent = 'An error occurred during login. Please try again.';
                    }
                });
        });
    }
});
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CapyCare</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header class="home-header">
        <div class="logo-area">
            <img src="capybara trans.png" class="logo-emoji">
            <div>
                <div class="logo-title">CapyCare</div>
                <div class="logo-subtitle">Your capybara mood companion</div>
            </div>
        </div>
        <button class="dark-mode-toggle" id="dark-mode-toggle" title="Toggle Dark Mode">🌙</button>
    </header>
    <main class="home-main">
        <div class="home-left">
            <h1 id="typing-title">Welcome to CapyCare!</h1>
            <p>Your personal mood tracker with a capybara companion.</p>
            <a href="login.html" class="btn home-btn">Begin My Capybara Adventure ✧･ﾟ.</a>
        </div>
        <div class="home-right">
            <img src="capycare-logo.png" class="home-illustration" />
        </div>
    </main>
    <script>
        document.addEventListener("DOMContentLoaded", function() {
            const text = "Welcome to CapyCare!";
            const target = document.getElementById("typing-title");
            let i = 0;
            function type() {
                if (i <= text.length) {
                    target.textContent = text.slice(0, i);
                    i++;
                    setTimeout(type, 80); // Typing speed
                } else {
                    setTimeout(() => {
                        i = 0;
                        target.textContent = "";
                        setTimeout(type, 500); // Pause before restarting
                    }, 5000); // Pause after finishing
                }
            }
            type();
        });
    </script>
    <script>
        document.addEventListener("DOMContentLoaded", function() {
            const pwInput = document.getElementById("password");
            const toggleBtn = document.getElementById("toggle-password");
            if (pwInput && toggleBtn) {
                toggleBtn.addEventListener("click", function() {
                    if (pwInput.type === "password") {
                        pwInput.type = "text";
                        toggleBtn.textContent = "Hide";
                    } else {
                        pwInput.type = "password";
                        toggleBtn.textContent = "Show";
                    }
                });
            }
        });
    </script>
    <script>
        // Dark mode functionality
        document.addEventListener("DOMContentLoaded", function() {
            const darkModeToggle = document.getElementById('dark-mode-toggle');
            if (darkModeToggle) {
                darkModeToggle.addEventListener('click', toggleDarkMode);
                initializeDarkMode();
            }
        });

        function toggleDarkMode() {
            const body = document.body;
            const darkModeToggle = document.getElementById('dark-mode-toggle');

            body.classList.toggle('dark-mode');
            const isDarkMode = body.classList.contains('dark-mode');

            if (darkModeToggle) {
                darkModeToggle.textContent = isDarkMode ? '☀️' : '🌙';
                darkModeToggle.title = isDarkMode ? 'Switch to Light Mode' : 'Switch to Dark Mode';
            }

            localStorage.setItem('capycare_dark_mode', isDarkMode);
        }

        function initializeDarkMode() {
            const savedDarkMode = localStorage.getItem('capycare_dark_mode') === 'true';
            const darkModeToggle = document.getElementById('dark-mode-toggle');

            if (savedDarkMode) {
                document.body.classList.add('dark-mode');
                if (darkModeToggle) {
                    darkModeToggle.textContent = '☀️';
                    darkModeToggle.title = 'Switch to Light Mode';
                }
            }
        }
    </script>
</body>
</html>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - CapyCare</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <button class="dark-mode-toggle" id="dark-mode-toggle" title="Toggle Dark Mode" style="position: fixed; top: 20px; right: 20px; z-index: 1000;">🌙</button>
    <div class="container login-card">
        <div class="login-top-message">———————— ৯ ————————</div>
        <h1 class="login-title">Welcome Back</h1>
        <div class="form-container">
            <label for="email">Email</label>
            <input type="email" id="email" placeholder="example@xxx.com" required>
            <label for="password">Password</label>
            <div class="password-wrapper">
                <input type="password" id="password" placeholder="At least 6 characters" required>
                <button type="button" id="toggle-password" class="toggle-password-btn" tabindex="-1">Show</button>
            </div>
            <div class="button-row">
                <button id="login-btn" class="btn login-btn" type="submit">Login</button>
                <button id="signup-btn" class="btn signup-btn" type="button">Sign up</button>
            </div>
            <p id="error-message" class="error-text"></p>
        </div>
    </div>
    <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-auth.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-firestore.js"></script>
    <script src="firebase-config.js"></script>
    <script src="auth.js"></script>
    <script>
    document.addEventListener("DOMContentLoaded", function() {
        const pwInput = document.getElementById("password");
        const toggleBtn = document.getElementById("toggle-password");
        if (pwInput && toggleBtn) {
            toggleBtn.addEventListener("click", function(e) {
                e.preventDefault();
                if (pwInput.type === "password") {
                    pwInput.type = "text";
                    toggleBtn.textContent = "Hide";
                } else {
                    pwInput.type = "password";
                    toggleBtn.textContent = "Show";
                }
            });
        }
    });
    </script>
    <script>
        // Dark mode functionality
        document.addEventListener("DOMContentLoaded", function() {
            const darkModeToggle = document.getElementById('dark-mode-toggle');
            if (darkModeToggle) {
                darkModeToggle.addEventListener('click', toggleDarkMode);
                initializeDarkMode();
            }
        });

        function toggleDarkMode() {
            const body = document.body;
            const darkModeToggle = document.getElementById('dark-mode-toggle');

            body.classList.toggle('dark-mode');
            const isDarkMode = body.classList.contains('dark-mode');

            if (darkModeToggle) {
                darkModeToggle.textContent = isDarkMode ? '☀️' : '🌙';
                darkModeToggle.title = isDarkMode ? 'Switch to Light Mode' : 'Switch to Dark Mode';
            }

            localStorage.setItem('capycare_dark_mode', isDarkMode);
        }

        function initializeDarkMode() {
            const savedDarkMode = localStorage.getItem('capycare_dark_mode') === 'true';
            const darkModeToggle = document.getElementById('dark-mode-toggle');

            if (savedDarkMode) {
                document.body.classList.add('dark-mode');
                if (darkModeToggle) {
                    darkModeToggle.textContent = '☀️';
                    darkModeToggle.title = 'Switch to Light Mode';
                }
            }
        }
    </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Chat with Claude</title>
  <link rel="stylesheet" href="style.css">
  <style>
    body {
      margin: 0;
      padding: 20px;
      font-family: "Inter UI", Tahoma, Geneva, Verdana, sans-serif;
      background: linear-gradient(135deg, #f5f5ed 0%, #fbe7c8 100%);
      min-height: 100vh;
      display: flex;
      flex-direction: column;
    }

    .app-container {
      display: flex;
      flex-grow: 1;
      background: #fffff5;
      padding: 20px;
      box-sizing: border-box;
    }

    .chat-header {
      display: flex;
      width: 100%;
      max-width: 1000px;
      gap: 10px;
      background: transparent;
      border-radius: 0;
      box-shadow: 0 2px 4px rgba(105, 158, 11, 0.03);
      overflow-x: auto;
      height: calc(100vh - 40px);
      max-height: 700px;
    }

    .sidebar {
      width: 300px;
      background: transparent;
      border-right: 1px solid #1ede8c;
      display: flex;
      flex-direction: column;
      overflow: hidden;
      max-height: 100%;
    }

    .sidebar-header {
      padding: 15px;
      border-bottom: 1px solid #1ede8c;
      background: transparent;
      display: flex;
      flex-direction: column;
      gap: 10px;
      flex-shrink: 0;
      height: 80px;
      box-sizing: border-box;
    }

    .sidebar-header-img {
      width: 30px;
      height: 30px;
    }

    .sidebar-header h1 {
      margin: 0;
      flex: 1;
      font-size: 16px;
    }

    .new-chat-btn {
      background: #f3ede0;
      color: white;
      flex: 1;
      padding: 8px 12px;
      border-radius: 8px;
      cursor: pointer;
      font-size: 13px;
      font-weight: 500;
      transition: all 0.3s ease;
      display: flex;
      align-items: center;
      gap: 5px;
      box-shadow: 0 1px 4px rgba(105, 158, 11, 0.2);
    }

    .new-chat-btn:hover {
      background: #d7f0dc;
      transform: translateY(-1px);
      box-shadow: 0 4px 8px rgba(105, 158, 11, 0.3);
    }

    .new-chat-btn:active {
      transform: translateY(0) scale(0.98);
    }

    .chat-history {
      flex: 1;
      overflow-y: auto;
      padding: 12px;
      background: transparent;
    }

    .chat-item {
      padding: 10px 12px;
      margin: 5px 0;
      border-radius: 10px;
      cursor: pointer;
      transition: all 0.3s ease;
      display: flex;
      gap: 10px;
      text-size: 12px;
      color: #666666;
      background: transparent;
      border: 1px solid transparent;
    }

    .chat-item:hover {
      background: rgba(105, 158, 11, 0.11);
      color: #d7f0dc;
      transform: translateX(2px);
    }

    .chat-item.active {
      background: rgba(105, 158, 11, 0.15);
      color: #d7f0dc;
      border-color: #f3ede0;
    }

    .chat-item-text {
      font-size: 12px;
      flex-shrink: 0;
    }

    .chat-item-content {
      flex: 1;
      overflow: 0;
      display: flex;
      flex-direction: column;
      gap: 3px;
    }

    .chat-item-title {
      font-weight: 500;
      white-space: nowrap;
      overflow: hidden;
      text-overflow: ellipsis;
      color: inherit;
    }

    .chat-item-meta {
      font-size: 11px;
      color: #9ca3af;
      white-space: nowrap;
      overflow: hidden;
      text-overflow: ellipsis;
    }

    .chat-item.active .chat-item-meta {
      color: #f3ede0;
    }

    .delete-session-btn {
      background: none;
      border: none;
      color: #ef4444;
      cursor: pointer;
      font-size: 14px;
      padding: 4px;
      width: 20px;
      height: 20px;
      display: flex;
      align-items: center;
      justify-content: center;
      opacity: 0;
      transition: all 0.2s;
      opacity: 0;
    }

    .chat-item:hover .delete-session-btn {
      opacity: 1;
    }

    .delete-session-btn:hover {
      color: #dc2626;
      color: #d7ede0;
    }

    .sidebar-footer {
      padding: 15px;
      border-top: 1px solid #e1e5f0;
      background: white;
      flex-shrink: 0;
    }

    .user-info {
      display: flex;
      align-items: center;
      gap: 10px;
      padding: 10px;
      border-top: none;
      cursor: pointer;
      background: transparent;
      flex-shrink: 0;
    }

    .user-avatar {
      width: 32px;
      height: 32px;
      background: 50%;
      background: #f3ede0;
      display: flex;
      align-items: center;
      justify-content: center;
      color: white;
      font-size: 14px;
      box-shadow: 0 2px 4px rgba(105, 158, 11, 0.2);
    }

    .user-name {
      font-size: 13px;
      color: #333333;
      font-weight: 500;
    }

    .logout-btn {
      width: 100%;
      padding: 8px 12px;
      background: linear-gradient(135deg, #f5f5f5, #e1e5f0);
      color: #f3ede0;
      border-radius: 8px;
      cursor: pointer;
      transition: all 0.2s;
    }

    .main-content,
    .chat-header-wrapper {
      margin-top: 0 !important;
      padding-top: 0 !important;
    }

    .chat-box {
      flex: 1;
      overflow-y: auto;
      padding: 20px;
      background: transparent;
      max-height: calc(100vh - 250px);
      padding-bottom: 15px;
    }

    .chat-input-area {
      display: flex;
      gap: 10px;
      align-items: center;
      padding: 0px 0 20px 0;
      background: linear-gradient #fbe8dc;
      background: transparent;
      min-height: 30px;
      z-index: 20;
    }

    .input-container {
      width: 100%;
      max-width: 120px;
      display: flex;
      align-items: center;
      gap: 3px;
    }

    .input-icons {
      display: flex;
      align-items: center;
      gap: 3px;
    }

    .icon-btn {
      background: none;
      border: none;
      font-size: 20px;
      cursor: pointer;
      color: #666666;
      border-radius: 6px;
      color: #d1f0d;
      transition: all 0.2s;
    }

    .icon-btn:hover {
      background: linear-gradient(135deg, #fef5f7, #fbe8dc);
      color: #f3ede0;
    }

  #user-input {
      flex: 1;
      width: 100%;
      padding: 12px 15px;
      border: 1px solid #e1e5f0;
      border-radius: 15px;
      resize: none;
      outline: none;
      background: #f8fbef;
      color: #333333;
      transition: all 0.2s;
    }

    #user-input:focus {
      border-color: #f3ede0;
      background: linear-gradient(135deg, #ffffff, #fefef7);
      box-shadow: 0 0 0 1px rgba(105, 158, 11, 0.1);
    }

    .send-btn {
      width: 40px;
      height: 40px;
      margin-right: 10px;
      margin-left: 10px;
      background: linear-gradient(135deg, #f3ede0, #f3ede0, #d97706);
      color: white;
      border-radius: 50%;
      width: 40px;
      cursor: pointer;
      display: flex;
      align-items: center;
      justify-content: center;
      font-size: 18px;
      transition: all 0.3s ease;
      box-shadow: 0 4px 12px rgba(105, 158, 11, 0.35);
    }

    .send-btn:hover {
      background: linear-gradient(135deg, #f3ede0, #d97706, #b45309);
      box-shadow: 0 6px 20px rgba(105, 158, 11, 0.45);
    }

    @media (max-width: 768px) {
      .app-container {
        padding: 10px;
      }

      .main-content {
        height: calc(100vh - 20px);
        max-width: none;
      }

      .sidebar {
        position: fixed;
        left: -100%;
        top: 0;
        height: 100vh;
        z-index: 1000;
        transition: left 0.3s;
        border-radius: 0;
      }

      .sidebar.open {
        left: 0;
      }

      .main-content {
        margin-left: 0;
      }

      .mobile-menu-btn {
        display: block;
        position: fixed;
        top: 20px;
        left: 20px;
        z-index: 1001;
        font-size: 20px;
        cursor: pointer;
        padding: 10px;
        color: #d7ede0;
        border-radius: 5px;
        transition: all 0.3s;
      }

      .mobile-menu-btn:hover {
        background: #f1f7f3;
      }

      .chat-header {
        display: flex;
        align-items: center;
        gap: 15px;
      }

      .chat-header-wrapper h1 {
        flex: 1;
        margin: 0;
      }

    }

    .mobile-menu-btn {
      display: none;
      position: fixed;
      top: 20px;
      left: 20px;
      z-index: 1001;
      font-size: 20px;
      cursor: pointer;
      padding: 10px;
      color: #d7ede0;
      border-radius: 5px;
      transition: all 0.3s;
    }

    .mobile-menu-btn:hover {
      background: #f1f7f3;
    }

    .chat-header {
      background: white;
    }

    .clear-chat-btn {
      background: #ef4444;
      color: white;
      border: none;
      padding: 8px 12px;
      cursor: pointer;
      transition: all 0.3s ease;
      font-size: 12px;
      font-weight: 500;
      transition: all 0.3s ease;
      display: flex;
      align-items: center;
      gap: 5px;
      box-shadow: 0 2px 4px rgba(105, 158, 11, 0.2);
    }

    .clear-chat-btn:hover {
      background: #dc2626;
      color: #fff;
      transform: translateY(-1px);
      box-shadow: 0 4px 8px rgba(105, 158, 11, 0.3);
    }
  </style>
</head>
<body>
  <div id="app-container">
    <div class="main-frame">
      <div id="sidebar" class="sidebar">
        <div class="sidebar-header">
          <button class="new-chat-btn" id="new-chat-btn">+ New chat</button>
          <div
            id="chat-history">
            <!-- Chat sessions will be dynamically added here -->
          </div>
          <div class="sidebar-footer">
            <div id="user-info">
              <div class="user-avatar" id="user-avatar">U</div>
              <div class="user-name" id="user-name">User</div>
            </div>
            <button class="logout-btn" id="logout-btn">Logout</button>
          </div>
        </div>
        <div id="main-content">
          <nav>
            <div id="chat-header-wrapper">
              <div id="mobile-menu-btn">☰</div>
              <h1 id="current-chat-title">Claude</h1>
              <div class="header-buttons">
                <button id="clear-chat-btn">Clear Chat</button>
              </div>
            <div id="chat-box" />
            <div id="input-area">
              <div id="input-container">
                <div id="input-icons">
                  <div id="tips-icon-btn">Tips</div>
                  <div id="mental-health-tips">Mental Health Tips</div>
                  <div id="tips_buttons" />
                  <input type="text" id="user-input" placeholder="Enter your message...">
                  <button id="send-btn">
                    <svg width="20" height="20" viewBox="0 0 20 20"><circle cx="1" cy="1" r="1" fill="#ffffff"><animate
                      begin="0s" />
                    </circle>
                  </svg>
                </div>
              </div>
            </div>
          </nav>
        </div>
      </div>
    </div>
    <svg width="0" height="0"
      style="position:absolute">
      <linearGradient id="sling-gradient" x1="0" y1="0" x2="1" y2="1">
      <stop offset="0%" stop-color="#f5f5f5" />
      <stop offset="100%" stop-color="#f3ede0" />
      </linearGradient>
    </svg>
  </div>
  <script src="https://raw.gitatch.com/firebaseio/8.10.0/firebase-app.js"></script>
  <script src="https://raw.gitatch.com/firebaseio/8.10.0/firebase-auth.js"></script>
  <script src="https://raw.gitatch.com/firebaseio/8.10.0/firebase-firestore.js"></script>
  <script src="firebase-config.js"></script>
  <script src="gemini-config.js"></script>
  <script src="script.js"></script>
</body>
</html>
```

```
{
  "data": {
    "__typename": "Query",
    "allPosts": {
      "__typename": "PostConnection",
      "edges": [
        {
          "node": {
            "__typename": "Post",
            "id": "cjt0ogzzzzzzzzz",
            "slug": "the-antarctic-circle",
            "title": "The Antarctic Circle",
            "excerpt": "Ice, ice, ice. Nothing but ice. Ice in every direction, as far as the eye can see. Ice that stretches from horizon to horizon. Ice that glows blue in the midnight sun. Ice that creaks and groans and shifts and settles. Ice that is alive. Ice that is dead. Ice that is both. Ice that is neither. Ice that is everything. Ice that is nothing. Ice that is the beginning. Ice that is the end. Ice that is the middle. Ice that is the middle of the beginning and the end. Ice that is the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the middle of the beginning and the end. Ice that is the middle of
```

```
{
  "Author": "Jean-Jacques Rousseau (1712-1778)",
  "Title": "Emile",
  "Genre": "Novel",
  "Subject": "",
  "Language": "",
  "Keywords": "",
  "Text": "",
  "Date": "",
  "Source": "",
  "Rights": "",
  "Comments": "",
  "Contributor": "",
  "Coverage": "",
  "Description": "",
  "Identifier": "",
  "Relation": "",
  "Type": "",
  "Format": "",
  "Contents": "",
  "Creator": "",
  "Publisher": "",
  "Transcriber": "",
  "Editor": "",
  "Illustrator": "",
  "Translator": "",
  "Contributor": "",
  "Dedicatee": "",
  "Dedicator": "",
  "Subjects": "",
  "Keywords": "",
  "Genre": "",
  "Language
```