import { loadBooks } from "./books.js"

/\* --------------------------- GLOBAL ALERT & PROMPT --------------------------- \*/

window.showAlert = function (text) {

  const alertMsg = document.getElementById("alertMessage");

  const customAlert = document.getElementById("customAlertModal");

  const alertOK = document.getElementById("alertOkButton");

  alertMsg.textContent = text;

  customAlert.style.display = "flex";

  alertOK.onclick = () => {

    customAlert.style.display = "none";

  };

};

window.showPrompt = function (text, initial = "") {

  return new Promise((resolve) => {

    const promptMsg = document.getElementById("promptMessage");

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

    const promptOK = document.getElementById("promptOkButton");

    const promptCancel = document.getElementById("promptCancelButton");

    const customPrompt = document.getElementById("customPromptModal");

    promptMsg.textContent = text;

    promptInput.value = initial;

    customPrompt.style.display = "flex";

    promptOK.onclick = () => {

      customPrompt.style.display = "none";

      resolve(promptInput.value);

    };

    promptCancel.onclick = () => {

      customPrompt.style.display = "none";

      resolve(null);

    };

  });

};

/\* --------------------------- DOM READY --------------------------- \*/

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

  const navLinks = document.querySelectorAll(".nav-links a");

  const contentPages = document.querySelectorAll(".content-page");

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

  function showPage(id) {

    contentPages.forEach((p) => {

      p.classList.remove("active-page");

      p.classList.add("hidden-page");

    });

    const page = document.getElementById(id);

    if (page) {

      page.classList.remove("hidden-page");

      page.classList.add("active-page");

    }

  }

  showPage("homePageContent");

  loadBooks();

  navLinks.forEach((link) => {

    link.addEventListener("click", (e) => {

      e.preventDefault();

      const key = link.dataset.page;

      showPage(`${key}PageContent`);

      if (key === "home" || key === "library") {

        loadBooks();

      } else if (key === "profile") {

        renderProfile();

      }

    });

  });

  function applyThemePreference() {

    const savedTheme = localStorage.getItem("theme");

    if (savedTheme === "dark") {

      document.body.classList.add("dark-mode");

      document.body.classList.remove("light-mode");

    } else {

      document.body.classList.add("light-mode");

      document.body.classList.remove("dark-mode");

    }

  }

  function toggleTheme() {

    if (document.body.classList.contains("dark-mode")) {

      document.body.classList.remove("dark-mode");

      document.body.classList.add("light-mode");

      localStorage.setItem("theme", "light");

    } else {

      document.body.classList.remove("light-mode");

      document.body.classList.add("dark-mode");

      localStorage.setItem("theme", "dark");

    }

  }

  themeToggle.addEventListener("click", toggleTheme);

  applyThemePreference();

  /\* --------------------------- CAROUSEL FUNCTIONALITY --------------------------- \*/

  const carouselWrapper = document.querySelector('.book-carousel');

  const leftButton = document.querySelector('.scroll-button.left');

  const rightButton = document.querySelector('.scroll-button.right');

  if (carouselWrapper && leftButton && rightButton) { // Ensure all elements exist

    let isDown = false;

    let startX;

    let scrollLeft;

    // Mouse drag functionality for desktop

    carouselWrapper.addEventListener('mousedown', (e) => {

      isDown = true;

      carouselWrapper.style.cursor = 'grabbing';

      startX = e.pageX - carouselWrapper.offsetLeft;

      scrollLeft = carouselWrapper.scrollLeft;

    });

    carouselWrapper.addEventListener('mouseleave', () => {

      isDown = false;

      carouselWrapper.style.cursor = 'grab';

    });

    carouselWrapper.addEventListener('mouseup', () => {

      isDown = false;

      carouselWrapper.style.cursor = 'grab';

    });

    carouselWrapper.addEventListener('mousemove', (e) => {

      if (!isDown) return;

      e.preventDefault();

      const x = e.pageX - carouselWrapper.offsetLeft;

      const walk = (x - startX) \* 2; // Adjust multiplier for scroll speed

      carouselWrapper.scrollLeft = scrollLeft - walk;

    });

    // Scroll button functionality

    leftButton.addEventListener('click', () => {

      const scrollAmount = carouselWrapper.offsetWidth \* 0.7; // Scroll by 70% of visible width

      carouselWrapper.scrollBy({

        left: -scrollAmount,

        behavior: 'smooth'

      });

    });

    rightButton.addEventListener('click', () => {

      const scrollAmount = carouselWrapper.offsetWidth \* 0.7; // Scroll by 70% of visible width

      carouselWrapper.scrollBy({

        left: scrollAmount,

        behavior: 'smooth'

      });

    });

    // Optional: Hide/Show scroll buttons based on scroll position

    const toggleScrollButtons = () => {

      const scrollTolerance = 5;

      if (carouselWrapper.scrollLeft <= scrollTolerance) {

        leftButton.style.display = 'none';

      } else {

        leftButton.style.display = 'flex';

      }

      if (carouselWrapper.scrollLeft + carouselWrapper.clientWidth >= carouselWrapper.scrollWidth - scrollTolerance) {

        rightButton.style.display = 'none';

      } else {

        rightButton.style.display = 'flex';

      }

    };

    // Initial check on load

    toggleScrollButtons();

    // Check on scroll

    carouselWrapper.addEventListener('scroll', toggleScrollButtons);

    // Check on window resize

    window.addEventListener('resize', toggleScrollButtons);

  } else {

    console.warn("Carousel elements not found. Carousel functionality will not be active.");

  }

  /\* ------------------------ END CAROUSEL FUNCTIONALITY ------------------------ \*/

});

const librarySearchBar = document.getElementById('librarySearchBar');

const libraryBookList = document.getElementById('libraryBookList');

// Ensure librarySearchBar and libraryBookList exist before adding event listener

if (librarySearchBar && libraryBookList) {

  librarySearchBar.addEventListener('input', () => {

    const searchTerm = librarySearchBar.value.toLowerCase();

    const bookCards = libraryBookList.querySelectorAll('.book-card');

    bookCards.forEach(card => {

      const title = card.querySelector('h3').textContent.toLowerCase();

      const author = card.querySelector('p').textContent.toLowerCase();

      const matches = title.includes(searchTerm) || author.includes(searchTerm);

      card.style.display = matches ? 'block' : 'none';

    });

  });

} else {

  console.warn("Library search bar or book list not found.");

}

// homeSearchBar and homeBookList were removed from HTML, so this block might not be necessary.

// If you add them back, ensure they are correctly referenced.

const homeSearchBar = document.getElementById('homeSearchBar');

const homeBookList = document.getElementById('homeBookList');

if (homeSearchBar && homeBookList) {

  homeSearchBar.addEventListener('input', () => {

    const searchTerm = homeSearchBar.value.toLowerCase();

    const bookCards = homeBookList.querySelectorAll('.book-card');

    bookCards.forEach(card => {

      const title = card.querySelector('h3')?.textContent.toLowerCase() || '';

      const author = card.querySelector('p')?.textContent.toLowerCase() || '';

      const matches = title.includes(searchTerm) || author.includes(searchTerm);

      card.style.display = matches ? 'block' : 'none';

    });

  });

}

// Dummy renderProfile function to avoid errors, assuming it's imported from profile.js

// If profile.js isn't linked or defined, this will prevent an error.

function renderProfile() {

  console.log("Rendering profile...");

  // Add your profile rendering logic here if it's not handled by profile.js

}