document.addEventListener('DOMContentLoaded', function() {

const header = document.querySelector('header');

const navLinks = document.querySelectorAll('.nav-link');

// Scroll effect for header

window.addEventListener('scroll', () => {

**if** (window.scrollY > 50) {

header.classList.add('header-scrolled');

} **else** {

header.classList.remove('header-scrolled');

}

});

// Active link highlighting

function setActiveLink() {

const scrollPosition = window.scrollY;

navLinks.forEach(link => {

const sectionElement = document.querySelector(link.hash);

**if** (sectionElement.offsetTop <= scrollPosition + 150 &&

sectionElement.offsetTop + sectionElement.offsetHeight > scrollPosition + 150) {

link.classList.add('active');

} **else** {

link.classList.remove('active');

}

});

}

window.addEventListener('scroll', setActiveLink);

setActiveLink(); // Set initial active link

// Smooth scrolling

navLinks.forEach(link => {

link.addEventListener('click', function(e) {

e.preventDefault();

const targetId = this.getAttribute('href');

const targetElement = document.querySelector(targetId);

window.scrollTo({

top: targetElement.offsetTop - 100,

behavior: 'smooth'

});

});

});

});

document.addEventListener('DOMContentLoaded', function() {

const menuToggle = document.querySelector('.menu-toggle');

const navUl = document.querySelector('nav ul');

menuToggle.addEventListener('click', function() {

navUl.classList.toggle('show');

});

// Close menu when clicking outside

document.addEventListener('click', function(event) {

**if** (!event.target.closest('header')) {

navUl.classList.remove('show');

}

});

// Close menu when resizing to larger screen

window.addEventListener('resize', function() {

**if** (window.innerWidth > 768) {

navUl.classList.remove('show');

}

});

});

document.addEventListener('DOMContentLoaded', (event) => {

// Smooth scrolling for navigation links

document.querySelectorAll('a[href^="#"]').forEach(anchor => {

anchor.addEventListener('click', function (e) {

e.preventDefault();

document.querySelector(this.getAttribute('href')).scrollIntoView({

behavior: 'smooth'

});

});

});

// Function to handle reveal animation

const revealElement = (entries, observer) => {

entries.forEach(entry => {

**if** (entry.isIntersecting) {

entry.target.classList.add('visible');

observer.unobserve(entry.target);

}

});

};

// Create Intersection Observer

const observer = new IntersectionObserver(revealElement, {

root: **null**,

threshold: 0.1

});

// Apply animation to experience cards

const experienceCards = document.querySelectorAll('.experience-card');

experienceCards.forEach(card => {

observer.observe(card);

});

// Apply animation to project cards

const projectCards = document.querySelectorAll('.project-card');

projectCards.forEach(card => {

observer.observe(card);

});

// Intro section animations

const introElements = document.querySelectorAll('#intro .profile-container, #intro .profile-picture, #intro .profile-info h1, #intro .profile-info h2, #intro .profile-info p, #intro .cta-buttons, #intro .social-links');

setTimeout(() => {

introElements.forEach(element => {

element.classList.add('loaded');

});

}, 100);

// Skill bars animation

const skillBars = document.querySelectorAll('.skill-progress');

const animateSkillBars = (entries, observer) => {

entries.forEach(entry => {

**if** (entry.isIntersecting) {

entry.target.style.width = entry.target.getAttribute('data-width');

observer.unobserve(entry.target);

}

});

};

const skillObserver = new IntersectionObserver(animateSkillBars, {

threshold: 0.5

});

skillBars.forEach(bar => {

skillObserver.observe(bar);

});

});

// Add a simple loading animation

window.addEventListener('load', () => {

document.body.classList.add('loaded');

});

function scrollToSection(sectionId) {

const section = document.getElementById(sectionId);

**if** (section) {

section.scrollIntoView({ behavior: 'smooth' });

}

}

document.addEventListener('DOMContentLoaded', function() {

const contactButtons = document.querySelectorAll('.contact-button');

contactButtons.forEach(button => {

button.addEventListener('click', function() {

const textToCopy = this.getAttribute('data-copytext');

copyToClipboard(textToCopy, this);

});

});

function copyToClipboard(text, button) {

// Create a temporary textarea element

const tempTextArea = document.createElement('textarea');

tempTextArea.value = text;

document.body.appendChild(tempTextArea);

// Select and copy the text

tempTextArea.select();

let success = **false**;

**try** {

success = document.execCommand('copy');

} **catch** (err) {

console.error('Unable to copy to clipboard', err);

}

// Remove the temporary textarea

document.body.removeChild(tempTextArea);

// Provide visual feedback

**if** (success) {

button.classList.add('copied');

setTimeout(() => {

button.classList.remove('copied');

}, 500); // Remove the 'copied' class after 2 seconds

alert('Copied to clipboard: ' + text);

} **else** {

alert('Failed to copy. Please try again or copy manually.');

}

}

});