Writing and Enhancing HTML with Copilot

Generative AI tools like Microsoft Copilot can simplify HTML writing and improve the quality of your code. AI tools can be used as coding assistants that create code based on your instructions.

In this activity, you will use Copilot to write HTML code for a portfolio website. This project will give you practice writing code yourself and using Copilot to help. You'll use Copilot to generate HTML elements, create semantic HTML structures, and ensure accessibility and SEO best practices.

This is the second of five activities where you will develop and code a front-end development project. The final output will be a portfolio website that you can use to show your mastery of front-end development with Copilot.

Activity Instructions

Step 1: Create a new HTML file

First, in your IDE, create a new file named index.html.

Step 2: Generate HTML structure

Next, you can add the HTML structure using Copilot.

1. Start by typing <!DOCTYPE html> and observe Copilot auto-complete the basic HTML structure.
2. Use Copilot to generate the <head> and <body> sections.
3. Include essential elements such as <title>, <meta>, and <link> for CSS.
4. Incorporate semantic HTML. Semantic HTML refers to using HTML tags that have meaning, such as <header>, <article>, and <footer>, which help with accessibility and SEO

Step 3: Develop the portfolio sections

Then, you can start creating different sections of your portfolio site using Copilot.

- Header: Type header and use Copilot to generate a header section with a navigation menu.
- About Me: Type section and name it "about-me." Use Copilot to generate a semantic structure with an <h2> heading and a <p> paragraph describing yourself.

- Projects: Create a "projects" section with appropriate headings and placeholder project descriptions. Use semantic tags like <article> and <figure> for each project.
- Skills: Use <ul> and <li> elements to list your skills. Let Copilot suggest common skills for a developer portfolio.
- Contact Form: Create a "contact" section with a form. Use <form>, <input>, <textarea>, and <button> elements. Ensure semantic naming and proper label association for accessibility.

Step 4: Incorporate accessibility features

Next, it's important to incorporate accessibility features into your HTML code with Copilot.

- Use Copilot to add alt attributes to images.
- Ensure all interactive elements (e.g., buttons, links) are accessible with keyboard navigation.
- Add ARIA roles where necessary (e.g., role="navigation" for the nav bar). ARIA roles are attributes added to HTML elements to improve accessibility, such as role='navigation' for the navigation bar."
- When adding styles, ensure that the text color contrasts well with the background to meet WCAG standards. Use tools like the WebAIM Contrast Checker to verify this.
- For text alternatives, include descriptive alt text for images, especially in the Projects section.
- Consider font choices that enhance readability and are supported across different devices.

Step 5: Follow SEO best practices

Finally, you can use Copilot to assist in implementing SEO best practices in your code.

- Add a meaningful <title> and <meta name="description">.
- Use semantic tags and meaningful heading hierarchies.

Step 6: Save your work

After you have your HTML code usable and complete, save your work. You will use this code in later activities.

Writing and Enhancing CSS with Copilot

Activity Introduction

Generative AI tools like Microsoft Copilot can help you create CSS and apply styling to your code. AI tools can help you work on basic styling and ensure that your code is responsive to different screen sizes.

In this activity, you will use Copilot to create a CSS file to style your previously written HTML code. This project will give you practice working with CSS and testing compatibility and also using Copilot to help with the process. You'll use Copilot to generate CSS and set styling across your portfolio website project.

This is the third of five activities where you will develop and code a front-end development project. The final output will be a portfolio website that you can use to show your mastery of front-end development with Copilot.

Activity Instructions

Step 1: Create a new CSS file

First, in your IDE, create a new file named styles.css.

Step 2: Link CSS to HTML

Then, ensure the CSS file is linked correctly in the <head> of your index.html file using the <link> tag. Make sure you're linking to the HTML code you wrote previously in the activity Writing and Enhancing HTML with Copilot.

Step 3: Generate basic styles

Next, you can begin adding basic styles to your project.

- Use Copilot to set basic styles for the body (e.g., font-family, margin, padding, background-color).
- Style the header and navigation bar with Copilot's suggestions for layout, color scheme, and typography.

Step 4: Style the portfolio sections

Then, you can start adding styling to the different sections of your portfolio site.

- About Me: Use Copilot to style the "About Me" section, focusing on text alignment, font size, and spacing.

- Projects: Using Copilot, create a card-like layout for the “Projects” section. Use properties like display: flex, margin, and padding.
- Skills: Style the “Skills” list with icons, using Copilot to suggest CSS pseudo-classes like ::before for adding icons.
- Contact Form: Create a "contact" section with a form. Use <form>, <input>, <textarea>, and <button> elements. Ensure semantic naming and proper label association for accessibility.

Step 5: Ensure responsive design

Next, it’s important to make sure your design is responsive to different screen sizes.

- Use Copilot to add media queries that adjust the layout for different screen sizes.
- Practice adding a breakpoint at 768px width. Adjust navigation to a mobile-friendly version using a hamburger menu icon.
- Ensure that images and videos are responsive with max-width: 100% and height: auto.

Step 6: Confirm cross-browser compatibility

Finally, you can use Copilot to ensure your code works across different browsers.

- Use Copilot to add vendor prefixes for CSS properties that require it.
- Test the website in different browsers (e.g. Edge, Chrome, Firefox) and note any discrepancies.
- Use Copilot to fix any issues by suggesting compatible CSS properties or syntax.

Step 7: Save your work

After you have your CSS in place, save your work. You will continue this project in later activities.

Writing and Enhancing JavaScript with Copilot

Activity Introduction

As a front-end developer, you can add interactivity to your projects with JavaScript. Generative AI tools like Microsoft Copilot can generate this code and check to ensure it works as expected.

In this activity, you will use Copilot to create JavaScript to make your previously written HTML code interactive. This project will give you practice working with JavaScript and Copilot. You'll use Copilot to generate JavaScript for your portfolio website project.

This is the fourth of five activities where you will develop and code a front-end development project. The final output will be a portfolio website that you can use to show your mastery of front-end development with Copilot.

Activity Instructions

Step 1: Create a new JavaScript file

First, in your IDE, create a new file named script.js.

Step 2: Link JavaScript to HTML

Then, ensure the JavaScript file is linked correctly at the bottom of your index.html file using the <script> tag. Make sure you're linking to the HTML code you wrote previously in the activity Writing and Enhancing HTML with Copilot and added CSS to in Writing and Enhancing CSS with Copilot.

Step 3: Add basic interactivity

Next, you can begin adding basic interactivity to your project.

- Use Copilot to write a function that toggles the navigation menu's visibility when the hamburger icon is clicked. Example: toggleMenu().
- Implement smooth scrolling behavior using Copilot for links in the navigation that reference sections within the page.

Step 4: Add interactivity to portfolio sections

Then, you can add interactive elements to specific parts of the website, like the "Project" section.

- Use Copilot to create a filter feature for the “Projects” section. Example: filterProjects(category).
- Implement a lightbox effect for project images. Use Copilot to write functions that display images in a modal view when clicked.

Step 5: Add form validation

Next, you can create an interactive “Contact” form that gives users feedback on their submission.

- Use Copilot to add form validation for the “Contact” form. Ensure fields like name, email, and message are filled in before submission.
- Provide real-time feedback using JavaScript to inform users of incorrect or missing input.

Step 6: Test and debug your JavaScript

Finally, you can use Copilot to ensure your JavaScript works properly.

- Test each function for expected behavior.
- Use Copilot to debug any issues, such as unexpected errors or incorrect behavior.
- Utilize console logs and Copilot’s debugging suggestions to fix issues.
- When debugging JavaScript, use the browser's developer tools to inspect the console for errors. If an error is found, Copilot can suggest potential fixes. For instance, if you see an 'undefined variable' error, Copilot might suggest initializing the variable before use.

Step 7: Save your work

After you have your JavaScript in place, save your work. You will continue this project in later activities.

Developing and Testing a Web Project with Copilot

Activity Introduction

At the end of any development project, you'll need to test to ensure that all of your code works together as expected and debug any issues that might arise. Microsoft Copilot can help with this process, including both testing and debugging.

In this activity, you will integrate the different parts of your portfolio website project. Then, you'll test to make sure everything works as expected and debug any issues that come up. You can use Microsoft Copilot in all parts of this process.

This is the fifth of five activities where you will develop and code a front-end development project. The final output will be a portfolio website that you can use to show your mastery of front-end development with Copilot.

Activity Instructions

Step 1: Integrate your project

First, integrate all the parts of your project: HTML, CSS, and JavaScript.

- Review the HTML, CSS, and JavaScript files to ensure all components are correctly linked and working together.
- Check that all styles are applied correctly and that interactive elements function as intended.

Step 2: Test your project

Then, ensure that your completed project works well in different situations.

- Test the website on multiple devices (desktop, tablet, mobile) and screen sizes.
- Use Copilot to simulate user interactions, like navigating through the site, submitting the contact form, and filtering projects.
- Document any usability issues or bugs encountered during testing.

Step 3: Debug your project

Next, debug any issues that you encountered.

- Use Copilot to debug identified issues. Look for common problems like broken links, missing styles, or JavaScript errors.

- Test cross-browser functionality to ensure consistent appearance and behavior.
- Make adjustments as needed, using Copilot's suggestions to refine the code for optimal performance.

Step 4: Save your work

When you're done, save your work. You'll submit this final project for review and feedback from peers.

HTML:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
  <meta name="robots" content="noindex, nofollow" />
  <meta name="description"
    content="Portfolio website of Dmitrii Bartash - Junior Full-Stack
Developer specializing in .NET & ASP.NET Core." />
  <title>Dmitrii Bartash | Junior Full-Stack Developer</title>
  <link rel="stylesheet" href="style.css" />
</head>

<body>

  <!-- Header -->
  <header role="banner">
    <button id="menu-toggle" aria-label="Toggle navigation">
      <span class="bar"></span>
      <span class="bar"></span>
      <span class="bar"></span>
    </button>

    <nav role="navigation" aria-label="Main navigation">
      <ul>
        <li><a href="#about-me">About</a></li>
        <li><a href="#experience">Experience</a></li>
        <li><a href="#education">Education</a></li>
        <li><a href="#skills">Skills</a></li>
        <li><a href="#certifications">Certifications</a></li>
        <li><a href="#projects">Projects</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </nav>
  </header>

  <!-- About Section -->
  <section id="about-me">
    <h2>About Me</h2>
    <p><strong>Dmitrii Bartash</strong> - Junior Full-Stack Developer
based in Chișinău, Moldova.</p>
    <p>Aspiring software developer with hands-on experience designing
and building full-stack web applications and
      distributed backend systems. Skilled in clean code, system
architecture, and cloud integration.</p>
    <p><strong>Email:</strong> <span class="email-
obfuscated">moc.liamg@hsatrab.iirtimd</span></p>
    <p><strong>GitHub:</strong> <a
href="https://github.com/DmitriiBartash" target="_blank"
        rel="noopener noreferrer">github.com/DmitriiBartash</a> |
      <strong>LinkedIn:</strong> <a
href="https://www.linkedin.com/in/dmitrii-bartash" target="_blank"
        rel="noopener noreferrer">linkedin.com/in/dmitrii-bartash</a>
    </p>
  </section>
```

```
  <!-- Experience Section -->
  <section id="experience">
    <h2>Experience</h2>

    <article>
      <h3>Biocond S.R.L – Intern Full-Stack Developer (Graduation
Project)</h3>
      <p><em>01/2024 – 06/2025 | Chișinău, Moldova</em></p>
      <ul>
        <li>Designed a microservice-based management system for climate
equipment installation</li>
        <li>Built backend with ASP.NET Core and FastAPI</li>
        <li>Integrated PostgreSQL, MongoDB, Redis</li>
        <li>Developed BTU Calculator for thermal load estimation</li>
        <li>Enabled REST, WebSocket, and RabbitMQ communication</li>
        <li>Containerized services with Docker for consistent
deployment</li>
        <li>Contributed to full-stack system design and integration</li>
      </ul>
    </article>

    <article>
      <h3>Emirat Travel – Intern Full-Stack Developer (Technological
Internship)</h3>
      <p><em>06/2023 – 12/2023 | Chișinău, Moldova</em></p>
      <ul>
        <li>Led team workflow and communication as team manager</li>
        <li>Built a full-stack ASP.NET Core web app with EF Core and
SQLite</li>
        <li>Developed admin panel and coordinated with client on
planning and delivery</li>
      </ul>
    </article>
  </section>

  <!-- Education Section -->
  <section id="education">
    <h2>Education</h2>

    <article>
      <h3>Master's Degree – Information Technology</h3>
      <p><em>Technical University of Moldova | 09/2025 –
01/2027</em></p>
    </article>

    <article>
      <h3>Bachelor's Degree – Information Technology</h3>
      <p><em>Technical University of Moldova | 09/2021 –
06/2025</em></p>
      <ul>
        <li>Average GPA: 9.59</li>
        <li>Highest grade (10) for bachelor's thesis</li>
        <li>Served as class representative</li>
      </ul>
    </article>
  </section>

  <!-- Skills Section -->
```

```
  <section id="skills">
    <h2>Skills</h2>
    <ul>
      <li><strong>Programming Languages:</strong> C#, Python,
JavaScript, SQL</li>
      <li><strong>.NET Frameworks & Tools:</strong> ASP.NET Core,
ASP.NET Web API, ASP.NET MVC, Razor Pages, Blazor,
        Entity Framework Core, LINQ</li>
      <li><strong>Python Frameworks:</strong> FastAPI</li>
      <li><strong>Frontend & UI:</strong> HTML5, CSS3, JavaScript</li>
      <li><strong>Databases & Storage:</strong> PostgreSQL, Microsoft
SQL Server, SQLite, MongoDB, Redis</li>
      <li><strong>Architecture & Integration:</strong> Monolithic,
Microservices, REST APIs, WebSocket, RabbitMQ</li>
      <li><strong>DevOps & Infrastructure:</strong> Docker, Docker
Compose, CI/CD, Git</li>
      <li><strong>Testing & Monitoring:</strong> xUnit, Postman, k6,
Grafana</li>
      <li><strong>Modeling & Design:</strong> UML, API & Database
Design</li>
    </ul>
  </section>

  <!-- Certifications Section -->
  <section id="certifications">
    <h2>Certifications</h2>
    <ul>
      <li>
        <strong>Introduction to Programming With C#</strong> - Microsoft
(July 22, 2025)<br />
        <a
href="https://coursera.org/share/7e4101de80eb49503dd0f0e393913064"
target="_blank"
          rel="noopener noreferrer">View Certificate</a>
      </li>
      <li>
        <strong>Foundations of Coding Full-Stack</strong> - Microsoft
(July 7, 2025)<br />
        <a
href="https://coursera.org/share/cf252a5528e0e4fbdbd1c0e0bd795b1c"
target="_blank"
          rel="noopener noreferrer">View Certificate</a>
      </li>
    </ul>
  </section>

  <!-- Projects Section -->
  <section id="projects">
    <h2>Projects</h2>

    <!-- Project Filters -->
    <div id="project-filters">
      <button data-category="all" class="active">All</button>
      <button data-category="web">Web</button>
      <button data-category="api">API</button>
      <button data-category="tool">Tool</button>
    </div>

    <!-- Project Cards -->
```

```
    <div class="project-card" data-category="web">
      <h3>Personal Portfolio</h3>
      <p>A responsive portfolio built with HTML, CSS, and
JavaScript.</p>
      <img src="images/portfolio.png" alt="Portfolio screenshot" />
    </div>

    <div class="project-card" data-category="api">
      <h3>Weather API</h3>
      <p>REST API built with ASP.NET Core and OpenWeather
integration.</p>
      <img src="images/portfolio.png" alt="Weather API screenshot" />
    </div>

    <div class="project-card" data-category="tool">
      <h3>BTU Calculator</h3>
      <p>Thermal load estimation tool built with FastAPI and JS
frontend.</p>
      <img src="images/btu.png" alt="BTU Calculator screenshot" />
    </div>
  </section>

  <!-- Lightbox Modal -->
  <div id="lightbox" class="lightbox" style="display: none;">
    <span id="lightbox-close">×</span>
    <img id="lightbox-img" src="" alt="Project Preview" />
  </div>

  <!-- Contact Section -->
  <section id="contact">
    <h2>Contact Me</h2>
    <form class="contact-form" action="#" method="post">
      <label for="name">Your Name</label>
      <input type="text" id="name" name="name" placeholder="Enter your
name" required />

      <label for="email">Your Email</label>
      <input type="email" id="email" name="email" placeholder="Enter
your email" required />

      <label for="message">Your Message</label>
      <textarea id="message" name="message" rows="5" placeholder="Type
your message here..." required></textarea>

      <button type="submit">Send Message</button>
    </form>
  </section>

  <!-- Footer -->
  <footer>
    <p>&copy; 2025 Dmitrii Bartash. All rights reserved.</p>
    <p>Languages: Russian (Native) | Romanian (Native) | English
(Improving)</p>
  </footer>

  <!-- Script -->
  <script src="script.js"></script>

</body>
```

```
</html>
```

CSS:

```css
/* ========= Root Variables ========= */
:root {
  --bg-color: #f9fafb;
  --text-color: #1f2937;
  --accent: #2563eb;
  --accent-dark: #1e40af;
  --card-bg: #ffffff;
  --card-border: #e5e7eb;
  --highlight-bg: #e0f2fe;
  --pill-bg: #eef2ff;
  --pill-text: #3730a3;
  --font-main: 'Segoe UI', Roboto, sans-serif;
  --shadow-sm: 0 1px 4px rgba(0, 0, 0, 0.06);
  --shadow-md: 0 4px 12px rgba(0, 0, 0, 0.1);
}

/* ========= Reset & Base ========= */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: var(--font-main);
  background-color: var(--bg-color);
  color: var(--text-color);
  line-height: 1.7;
  font-size: 16px;
  padding: 0 1rem;
  scroll-behavior: smooth;
}

img {
  max-width: 100%;
  display: block;
  border-radius: 8px;
}

/* ========= Layout Containers ========= */
header,
section,
footer {
  max-width: 1000px;
  margin: auto;
  padding: 3rem 1rem;
}

section {
  scroll-margin-top: -10px;
}

/* ========= Header & Navigation ========= */
header {
  background: linear-gradient(90deg, #1f2937, #111827);
  color: #fff;
  position: sticky;
```

```
  top: 0;
  z-index: 100;
  box-shadow: var(--shadow-md);
  border-radius: 0 0 16px 16px;
  padding: 1.5rem 1rem;
  display: flex;
  flex-direction: column;
  gap: 1rem;
}

/* Hamburger Icon */
#menu-toggle {
  width: 30px;
  height: 24px;
  position: relative;
  display: none;
  cursor: pointer;
  z-index: 101;
  background: none;
  border: none;
}

#menu-toggle .bar {
  position: absolute;
  width: 100%;
  height: 3px;
  background-color: #fff;
  left: 0;
  transition: all 0.3s ease;
  border-radius: 2px;
}

#menu-toggle .bar:nth-child(1) {
  top: 0;
}

#menu-toggle .bar:nth-child(2) {
  top: 50%;
  transform: translateY(-50%);
}

#menu-toggle .bar:nth-child(3) {
  bottom: 0;
}

#menu-toggle.active .bar:nth-child(1) {
  transform: rotate(45deg);
  top: 50%;
}

#menu-toggle.active .bar:nth-child(2) {
  opacity: 0;
}

#menu-toggle.active .bar:nth-child(3) {
  transform: rotate(-45deg);
  top: 50%;
  bottom: auto;
}
```

```css
/* Navigation */
nav ul {
  display: flex;
  justify-content: center;
  gap: 2rem;
  list-style: none;
  flex-wrap: wrap;
  transition: all 0.4s ease;
}

nav a {
  color: #fff;
  text-decoration: none;
  font-weight: 500;
  position: relative;
  padding: 0.5rem 0;
  transition: color 0.3s ease;
}

nav a::after {
  content: '';
  position: absolute;
  height: 2px;
  width: 0;
  left: 50%;
  bottom: 0;
  background-color: #3b82f6;
  transition: all 0.3s ease;
}

nav a:hover {
  color: #dbeafe;
}

nav a:hover::after {
  left: 0;
  width: 100%;
}

/* ========= Headings ========= */
h2 {
  font-size: 2.2rem;
  margin-bottom: 2rem;
  border-bottom: 2px solid var(--card-border);
  padding-bottom: 0.6rem;
  color: var(--accent-dark);
}

h3 {
  font-size: 1.4rem;
  margin-top: 1.8rem;
  margin-bottom: 0.7rem;
  color: var(--text-color);
}

/* ========= Card Sections ========= */
section>article,
section>div.project-card {
```

```css
  background: var(--card-bg);
  border: 1px solid var(--card-border);
  border-radius: 12px;
  padding: 2rem;
  margin-bottom: 2rem;
  box-shadow: var(--shadow-sm);
  transition: transform 0.2s ease, box-shadow 0.3s ease;
}

section>article:hover,
section>div.project-card:hover {
  transform: translateY(-4px);
  box-shadow: var(--shadow-md);
}

/* ========= Paragraphs & Lists ========= */
p {
  margin-bottom: 1.1rem;
}

ul {
  padding-left: 1.5rem;
  margin-top: 0.5rem;
  margin-bottom: 1.5rem;
}

ul li {
  margin-bottom: 0.5rem;
}

/* ========= Links ========= */
a {
  color: var(--accent);
  transition: color 0.2s ease;
}

a:hover {
  color: var(--accent-dark);
}

/* ========= Skills Section ========= */
#skills ul {
  display: flex;
  flex-wrap: wrap;
  gap: 0.6rem;
  list-style: none;
  padding-left: 0;
}

#skills li {
  background-color: var(--pill-bg);
  color: var(--pill-text);
  padding: 0.5rem 0.9rem;
  border-radius: 20px;
  font-size: 0.9rem;
  font-weight: 500;
  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.05);
}
```

```css
/* ========= Certifications Section ========= */
#certifications a {
  display: inline-block;
  margin-top: 0.4rem;
  color: var(--accent);
  font-size: 0.95rem;
}

/* ========= Projects Section ========= */
#project-filters {
  display: flex;
  flex-wrap: wrap;
  gap: 0.8rem;
  margin-bottom: 2rem;
  justify-content: center;
}

#project-filters button {
  padding: 0.5rem 1rem;
  border: 1px solid var(--accent);
  background-color: white;
  color: var(--accent);
  border-radius: 20px;
  font-weight: 500;
  cursor: pointer;
  transition: all 0.3s ease;
}

#project-filters button:hover,
#project-filters button.active {
  background-color: var(--accent);
  color: white;
}

/* ========= Lightbox ========= */
.lightbox {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.85);
  display: flex;
  justify-content: center;
  align-items: center;
  z-index: 1000;
}

.lightbox img {
  max-width: 90%;
  max-height: 90%;
  border-radius: 8px;
}

#lightbox-close {
  position: absolute;
  top: 2rem;
  right: 2rem;
  color: #fff;
```

```css
  font-size: 2rem;
  cursor: pointer;
}

/* ========= Contact Section ========= */
#contact {
  background-color: var(--highlight-bg);
  border-left: 5px solid var(--accent);
  border-right: 5px solid var(--accent);
  padding: 2rem;
  border-radius: 12px;
}

#contact h2 {
  text-align: center;
}

#contact p {
  font-weight: 600;
  font-size: 1.1rem;
  text-align: center;
}

/* ========= Contact Form ========= */
.contact-form {
  margin-top: 1.5rem;
  display: flex;
  flex-direction: column;
  gap: 1rem;
}

.contact-form label {
  font-weight: 600;
  color: var(--text-color);
}

.contact-form input,
.contact-form textarea {
  padding: 0.7rem;
  border: 1px solid var(--card-border);
  border-radius: 6px;
  font-size: 1rem;
  font-family: var(--font-main);
  background-color: #fff;
}

.contact-form textarea {
  resize: vertical;
}

.contact-form button {
  background-color: var(--accent);
  color: white;
  padding: 0.7rem 1.2rem;
  border: none;
  border-radius: 6px;
  font-size: 1rem;
  font-weight: 600;
  cursor: pointer;
```

```css
  transition: background 0.3s ease;
}

.contact-form button:hover {
  background-color: var(--accent-dark);
}

/* ========= Footer ========= */
footer {
  text-align: center;
  font-size: 0.9rem;
  color: #6b7280;
  border-top: 1px solid var(--card-border);
  padding: 2rem 1rem;
  margin-top: 3rem;
}

/* ========= Obfuscated Email ========= */
.email-obfuscated {
  unicode-bidi: bidi-override;
  direction: rtl;
  cursor: pointer;
  display: inline-block;
  background: rgba(37, 99, 235, 0.1);
  padding: 0.2rem 0.5rem;
  border-radius: 4px;
  font-weight: 500;
  transition: background 0.3s ease;
}

.email-obfuscated:hover {
  background: rgba(37, 99, 235, 0.2);
}

/* ========= Responsive ========= */
@media (max-width: 768px) {
  #menu-toggle {
    display: block;
    margin-left: auto;
    margin-right: auto;
  }

  nav ul {
    display: none;
    flex-direction: column;
    gap: 1rem;
    padding: 1rem;
    border-radius: 12px;
    text-align: center;
  }

  nav ul.open {
    display: flex;
  }

  h2 {
    font-size: 1.8rem;
  }
```

```
  h3 {
    font-size: 1.2rem;
  }

  section>article,
  section>div.project-card {
    padding: 1.2rem;
  }

  #skills li {
    font-size: 0.85rem;
    padding: 0.4rem 0.8rem;
  }

  .contact-form input,
  .contact-form textarea {
    font-size: 0.95rem;
  }

  .contact-form button {
    font-size: 0.95rem;
  }

}
```

JS:

```
// ===== Toggle navigation menu (burger menu with animation) =====
const menuToggle = document.getElementById('menu-toggle');
const navList = document.querySelector('nav ul');

menuToggle?.addEventListener('click', () => {
  navList.classList.toggle('open');
  menuToggle.classList.toggle('active');
});

// ===== Smooth scroll for internal links =====
document.querySelectorAll('a[href^="#"]').forEach(anchor => {
  anchor.addEventListener('click', function (e) {
    e.preventDefault();
    const target = document.querySelector(this.getAttribute('href'));

    if (navList.classList.contains('open')) {
      navList.classList.remove('open');
      menuToggle.classList.remove('active');

      setTimeout(() => {
        target?.scrollIntoView({ behavior: 'smooth' });
      }, 400);
    } else {
      target?.scrollIntoView({ behavior: 'smooth' });
    }
  });
});

// ===== Filter projects by category =====
const filterButtons = document.querySelectorAll('#project-filters
button');
const projectCards = document.querySelectorAll('.project-card');

filterButtons.forEach(button => {
  button.addEventListener('click', () => {
    const category = button.getAttribute('data-category');
    filterProjects(category);

    filterButtons.forEach(btn => btn.classList.remove('active'));
    button.classList.add('active');
  });
});

function filterProjects(category) {
  projectCards.forEach(card => {
    const match = category === 'all' || card.dataset.category ===
category;
    card.style.display = match ? 'block' : 'none';
  });
}

// ===== Lightbox effect for project images =====
const lightbox = document.getElementById('lightbox');
const lightboxImg = document.getElementById('lightbox-img');
const lightboxClose = document.getElementById('lightbox-close');

document.querySelectorAll('.project-card img').forEach(img => {
```

```
  img.addEventListener('click', () => {
    lightboxImg.src = img.src;
    lightbox.style.display = 'flex';
  });
});

lightboxClose.addEventListener('click', () => {
  lightbox.style.display = 'none';
});

lightbox.addEventListener('click', (e) => {
  if (e.target === lightbox) {
    lightbox.style.display = 'none';
  }
});

// ===== Contact form validation =====
const form = document.querySelector('.contact-form');

form?.addEventListener('submit', function (e) {
  e.preventDefault();

  const name = form.name;
  const email = form.email;
  const message = form.message;

  let valid = true;

  [name, email, message].forEach(input => {
    input.style.borderColor = '';
    if (!input.value.trim()) {
      input.style.borderColor = 'red';
      valid = false;
    }
  });

  if (valid) {
    alert('Message sent successfully!');
    form.reset();
    [name, email, message].forEach(input => input.style.borderColor =
'');
  } else {
    alert('Please fill in all required fields.');
  }
});

// ===== Real-time input feedback =====
['name', 'email', 'message'].forEach(id => {
  const input = document.getElementById(id);
  input?.addEventListener('input', () => {
    input.style.borderColor = input.value.trim() ? 'green' : 'red';
  });
});

// ===== Reveal and open obfuscated email on click =====
document.querySelectorAll('.email-obfuscated').forEach(el => {
  el.addEventListener('click', () => {
    const reversed = el.textContent.split('').reverse().join('');
    window.location.href = 'mailto:' + reversed;
```

```
    });
});
```