To meet the project requirements, let's break down the tasks and ensure we cover all necessary aspects. Here's a detailed plan to create a website project hosted on GitHub Pages, consisting of at least three congruent pages with common navigation and theme, optimized for accessibility, usability, and visual appeal, and containing dynamic elements using JavaScript.

### 1. Set Up GitHub Repository and GitHub Pages

1. \*\*Create a GitHub repository\*\*: Name it appropriately for your project.

2. \*\*Enable GitHub Pages\*\*: In the repository settings, enable GitHub Pages and choose the main branch (or a `gh-pages` branch) as the source.

### 2. Website Structure

- \*\*Home Page (index.html)\*\*: Introduction and overview of the site.

- \*\*About Page (about.html)\*\*: Information about the purpose of the site and the target audience.

- \*\*Contact Page (contact.html)\*\*: Contact form and other relevant information.

### 3. Common Navigation and Theme

- \*\*Navigation Bar\*\*: Include links to Home, About, and Contact pages.

- \*\*CSS Theme\*\*: Consistent color scheme, typography, and layout across all pages.

### 4. Accessibility and Usability

- \*\*ARIA Landmarks\*\*: Use `nav`, `header`, `main`, `footer` roles.

- \*\*Alt Text\*\*: Ensure all images have meaningful alt text.

- \*\*Keyboard Navigation\*\*: Ensure all interactive elements are accessible via keyboard.

### 5. HTML Structure

```html

<!-- Example HTML Structure -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Website Title</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="about.html">About</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

</header>

<main>

<!-- Page-specific content -->

</main>

<footer>

<p>&copy; 2024 Your Name</p>

</footer>

<script src="scripts.js"></script>

</body>

</html>

```

### 6. CSS Styling

- \*\*Responsive Design\*\*: Use media queries to ensure the site looks good on both mobile and desktop.

- \*\*Optimized CSS\*\*: Avoid duplicate declarations and unused rules.

```css

/\* Example CSS \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header, nav, main, footer {

padding: 20px;

}

nav ul {

list-style: none;

padding: 0;

}

nav ul li {

display: inline;

margin-right: 10px;

}

@media (max-width: 600px) {

nav ul li {

display: block;

margin: 5px 0;

}

}

```

### 7. Images and Lazy Loading

- \*\*Optimize Images\*\*: Use appropriate formats and compression.

- \*\*Lazy Loading\*\*: Use `loading="lazy"` attribute on images.

```html

<img src="example.jpg" alt="Description" loading="lazy">

```

### 8. HTML Form

- \*\*Contact Form (contact.html)\*\*

```html

<form id="contactForm">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

<label for="message">Message:</label>

<textarea id="message" name="message" required></textarea>

<button type="submit">Submit</button>

</form>

```

### 9. JavaScript Functionality

- \*\*Dynamic Elements\*\*: Use JavaScript for form validation, interaction, and local storage.

```javascript

// Example JavaScript (scripts.js)

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

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

contactForm.addEventListener('submit', (e) => {

e.preventDefault();

const name = document.getElementById('name').value;

const email = document.getElementById('email').value;

const message = document.getElementById('message').value;

// Simple validation

if (name && email && message) {

localStorage.setItem('contactFormData', JSON.stringify({ name, email, message }));

alert('Thank you for your message!');

contactForm.reset();

} else {

alert('Please fill out all fields.');

}

});

});

```

### 10. Additional Requirements

- \*\*No spelling or grammatical errors\*\*: Proofread all content.

- \*\*Project Completion\*\*: Ensure all features are implemented and tested by the due date.

By following this plan, you'll create a functional, accessible, and responsive website that meets all project requirements. Good luck with your project!