(i) DOM manipulation

Add a "dark mode" toggle button that changes the color scheme of the website using JavaScript to modify CSS classes.

For the dark mode toggle, I'll add a button that when clicked, applies a "dark-mode" CSS class to the <body> element. The dark mode CSS class will have styles that override the default light color scheme. This will be implemented using JavaScript to listen for the button click event and toggle the class.

html

```
<button id="darkModeToggle">Toggle Dark Mode</button>
```

css

```
body.dark-mode {
  background-color: #333;
  color: #fff;
}
```

js

```
const darkModeToggle = document.getElementById('darkModeToggle');
darkModeToggle.addEventListener('click', function() {
  document.body.classList.toggle('dark-mode');
});
```

(ii) Browser API

Implement a geolocation feature that detects the user's location and displays a personalized greeting or content based on their location.

The geolocation feature will use the browser's geolocation API to request the user's location. If the user allows access, the latitude and longitude will be retrieved. I can then use this information to display a customized message, such as "Greetings from [City]!" by reverse geocoding the coordinates with a service like Google Maps API.

js

```
if ("geolocation" in navigator) {
  navigator.geolocation.getCurrentPosition(function(position) {
    const latitude = position.coords.latitude;
    const longitude = position.coords.longitude;

    // Use Google Maps API to reverse geocode coordinates and get city name
    const geocodeUrl =
`https://maps.googleapis.com/maps/api/geocode/json?latlng=${latitude},${longitude}&key=YOUR_API_
KEY`;
```

```
  fetch(geocodeUrl)
    .then(response => response.json())
    .then(data => {
      const city = data.results[0].address_components.find(component =>
component.types.includes('locality')).long_name;
      document.getElementById('greeting').textContent = `Greetings from ${city}!`;
    });
  });
}
```

(iii) jQuery library

Create an image gallery with a lightbox effect using a jQuery plugin like Fancybox or Magnific Popup.

For the image gallery, I'll use a jQuery plugin like Fancybox to create an interactive lightbox. When an image thumbnail is clicked, the full-size image will open in an overlay with navigation controls. The plugin will handle the lightbox functionality, while I'll set up the HTML structure and initialize the plugin with the appropriate options.

```
html
<div class="gallery">
  <a href="image1.jpg" data-fancybox="gallery">
    <img src="thumbnail1.jpg" alt="Image 1">
  </a>
  <a href="image2.jpg" data-fancybox="gallery">
    <img src="thumbnail2.jpg" alt="Image 2">
  </a>
  <!-- more gallery images -->
</div>

js
$(document).ready(function() {
  $('[data-fancybox]').fancybox({
    // Fancybox options
  });
});
```