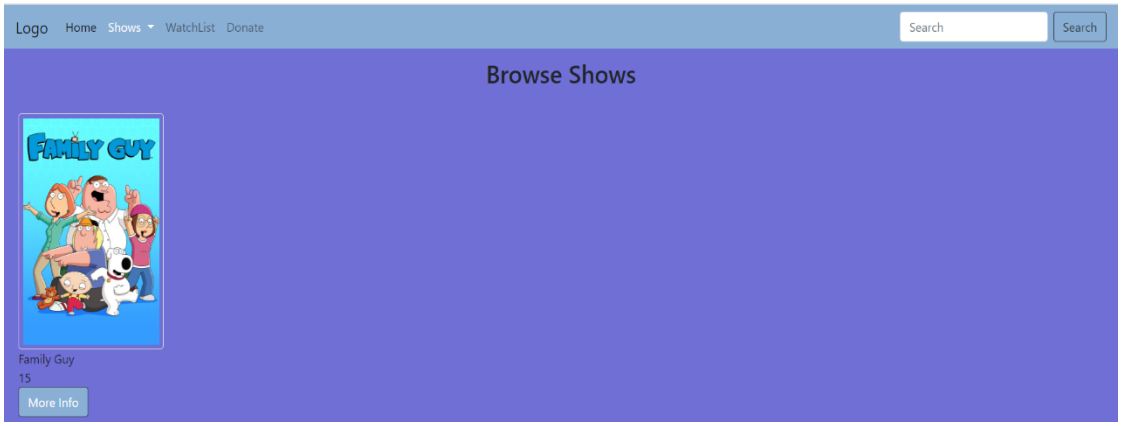
**Extracting information from the database:**

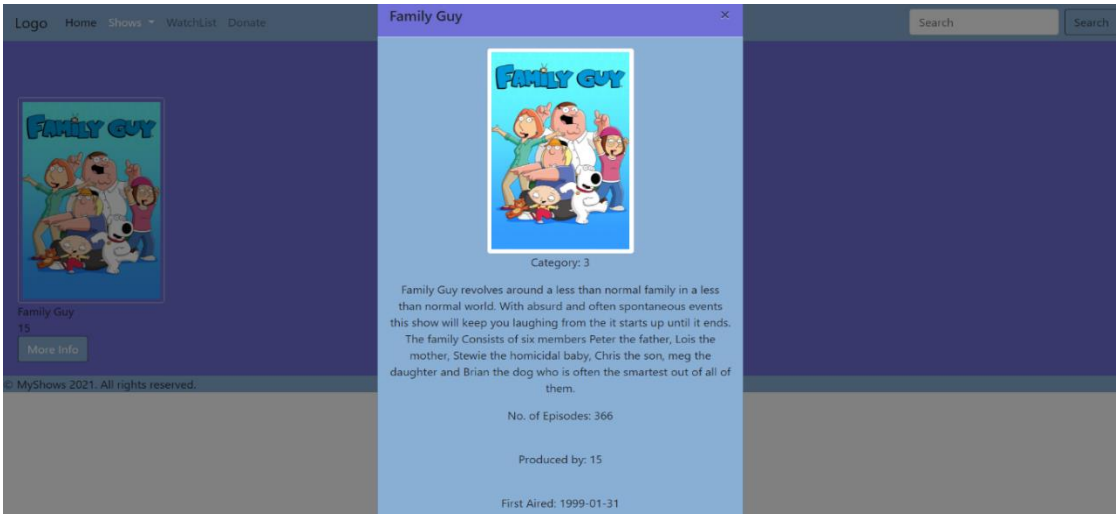
```php
<?php
require_once('header.php');

$showList = "SELECT showID, showImage, showName, showStu, showCat, showDesc, showEp, startDate FROM shows
    INNER JOIN category ON shows.showCat = category.catID
    INNER JOIN studio ON shows.showStu = studio.stuID";

    $showPrep = $dbConn->prepare($showList);

    $showPrep->execute();

while($showRow = $showPrep->fetchObject()) { ?>
```

The above PHP code extracts key information from the database: the show ID, name, studio, category, description, number of episodes and initial air date. On the main page, only the show name and studio are extracted and if the user wishes for more, they can click the 'more info' button to open a modal that will show all the information. The image below showcases the website's main page. Only one show can be seen but more will be added by the end of the project.




**Modals:**

As mentioned above, there is a 'more info' button the user can click on to bring up a more detailed profile of the show. On clicking the button, a modal will pop-up on the screen displaying the image of the show and all the information about it held in the database. Given the page limitation for this assignment is ten pages, Modals are an appropriate choice, so each show does not require its own separate page. In addition, they can be closed easily, either by clicking the small cross in the header of the modal or the 'close' button in the footer. The following images showcase this feature and code:

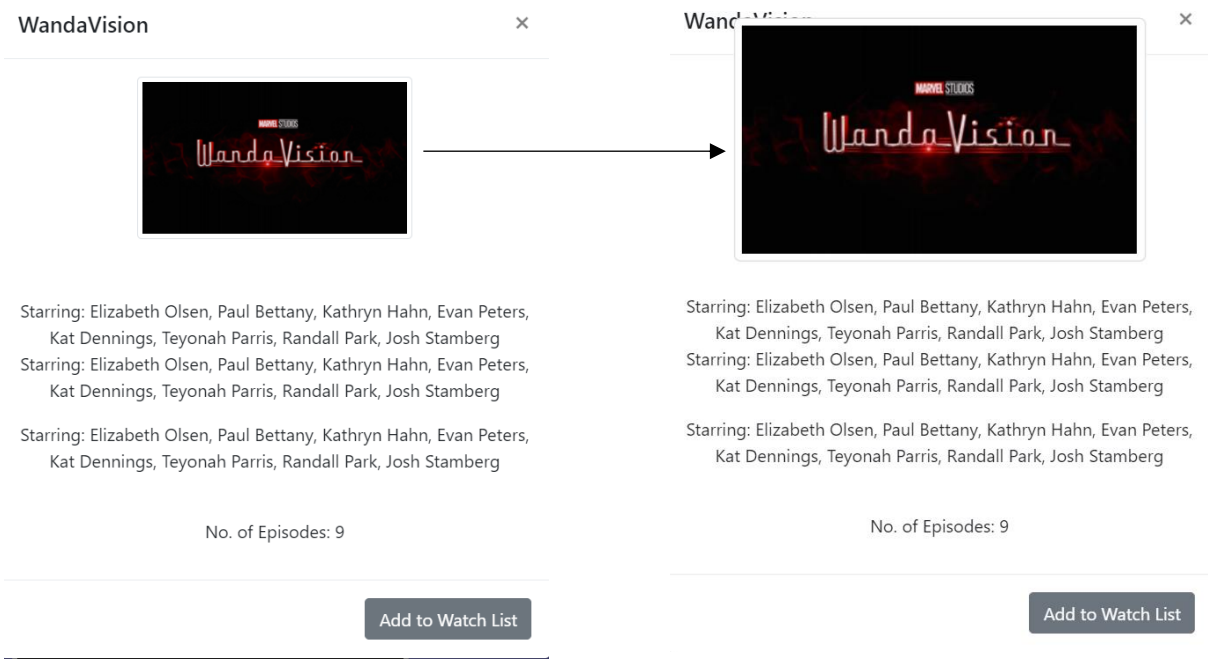


```
  <!-- Modal -->
<div class="modal fade" id="showModal" tabindex="-1" role="dialog" aria-labelledby="showModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="showModalLabel"><?php echo $showRow->showName; ?></h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
          <div class="zoom-image">
          <img class="img-thumbnail" src="img/<?php echo $showRow->showImage; ?>" alt="<?php echo $showRow->showName; ?>">
          </div>
          <div class="info">
          <p>Category: <?php echo $showRow->showCat; ?></p>
          <p><?php echo $showRow->showDesc; ?> <br />
          <p>No. of Episodes: <?php echo $showRow->showEp; ?></p> <br />
          <p>Produced by: <?php echo $showRow->showStu; ?></p> <br />
          <p>First Aired: <?php echo $showRow->startDate; ?></p> <br />
          </div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Add to Watch List</button>
      </div>
    </div>
  </div>
</div>
```

To make the data more readable for the user, labels have been added before the data, for example: "No. of Episodes:" This provides the user with confirmation of what data they are reading so they are not rendered confused. There is also an 'Add to Watch List' button which will link with another feature of the website – when they press this button, the show is added to the watchlist page and the user is provided with an opportunity to download a text file of the shows they have selected. The modal colors also follow the theme of the website, providing design consistency.

**Modal feature – Image expansion:**

An additional feature for the modal I have included is to have the image expand when the user hovers over (desktop) or clicks (mobile, tablet etc.) it. This extra feature allows the user to get a clearer view of the show's concept art/title card whilst the modal is open. This is demonstrated below:




The screenshots above are based on a test webpage I created before adding to the actual project – this was to ensure there were no errors before its implementation onto the website. As shown, the image

expands on mouse hover causing the image of size 250px by 200px to expand by a factor of 1.5 in a .2s transition. The enlargement does not need to be too large as it does not need to go outside of the modal. The CSS for this feature is shown below.

```css
.zoom-image {
  transition: transform .2s;
  width: 250px;
  height: 200px;
  margin: 0 auto;
}
.zoom-image:hover {
  -ms-transform: scale(1.5);
  transform: scale(1.5);
}
```

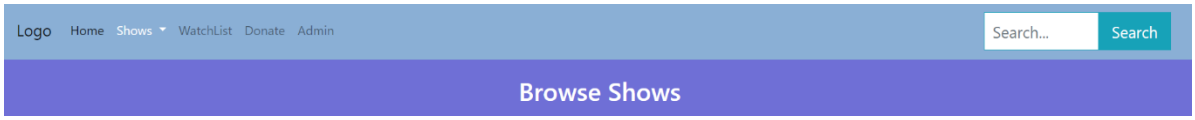
This feature was inspired from the W3Schools website and follows a remarkably similar CSS structure. Further plans for this feature are to have it push the modal text down the page when it expands so as not to block any of the text from the user.

***Note: This feature has been removed from the final website due to its conflictions with Bootstrap's CSS framework.***

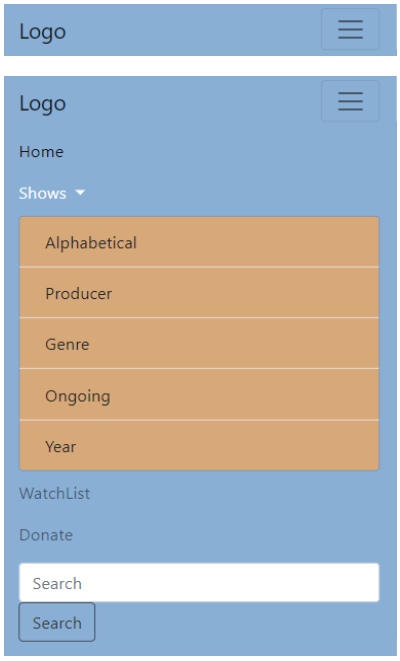
**Navigation menu with Bootstrap:**

Bootstrap provides a way to create a dynamically responsive navigation menu for any website which includes space for a logo, several links (including a dropdown menu for filtering, which is Title by default) and a search bar. Since this website is to be mobile first, it needs to be suitable for mobile devices, therefore on smaller screens, the navigation collapses into a 'hamburger' icon menu which drops from the top of the screen.

Navigation on desktop screens:




Navigation on mobile screens:



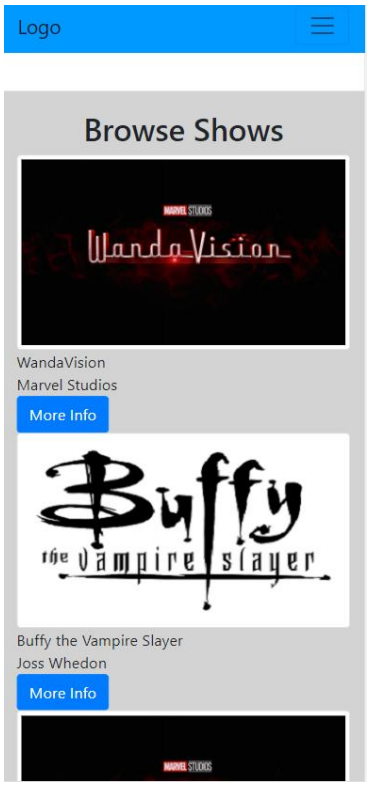

As can be seen, the navigation has collapsed into the menu icon which can be toggled by the user. The second image shows what the screen looks like once the menu icon has been toggled by the user. The links fall from the top of the screen in the same order as on desktop and the dropdown menu works in a similar fashion. The 'Shows' dropdown menu is also active, and appear in a different color for the user, able to be toggled on and off by the little arrow. Bootstrap has built in dropdown-dividers, allowing the links to be split apart from one another. The code for the navigation menu is shown below:

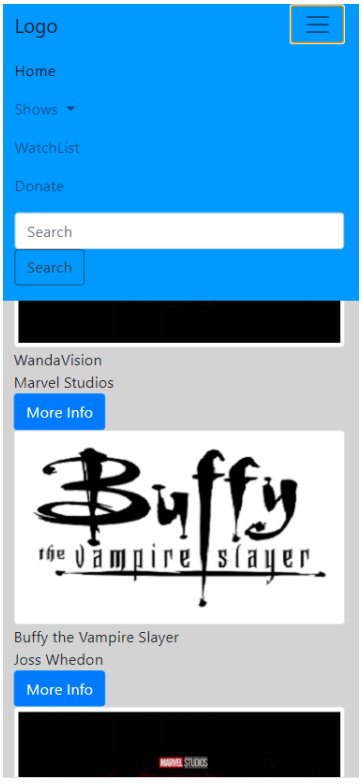
```
<nav class="navbar navbar-expand-md navbar-light" style="background-color: #8AAFD5;position: relative">
    <a class="navbar-brand" href="#">Logo</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarToggler" aria-controls="navbarToggler" aria-expanded="false" aria-label="T
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarToggler">
        <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
            <li class="nav-item active">
                <a class="nav-link" href="index.php">Home <span class="sr-only">(current)</span></a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" area-expanded="false">
                    Shows
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdown" style="background-color: #D7A878">
                    <a class="dropdown-item">Alphabetical</a>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item">Producer</a>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item"> Genre</a>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item"> Ongoing</a>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item"> Year</a>
                </div>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="watchlist.php">WatchList</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="donate.php">Donate</a>
            </li>
        </ul>
    </div>
```

**Color Schemes:**

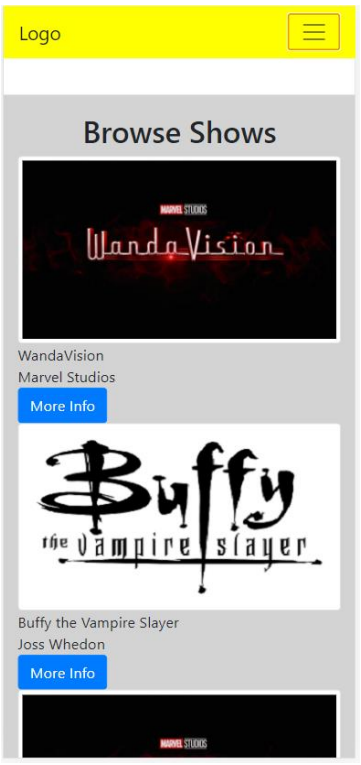
Numerous color schemes were proposed by our web designer and all tried out before deciding on a look for the website. The following four were all trialed:

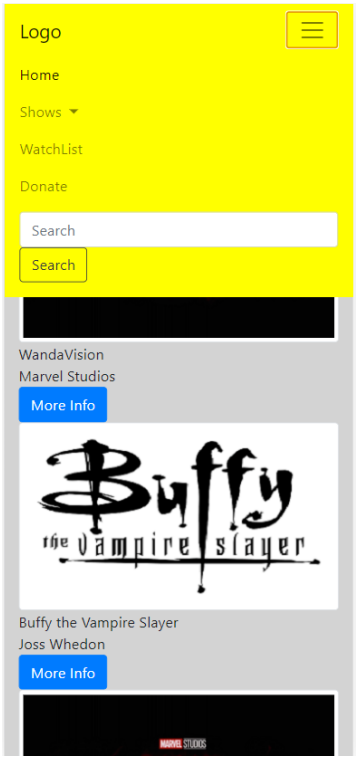
1) Light gray background with blue navigation and a dark blue for the navigation sub-menu.

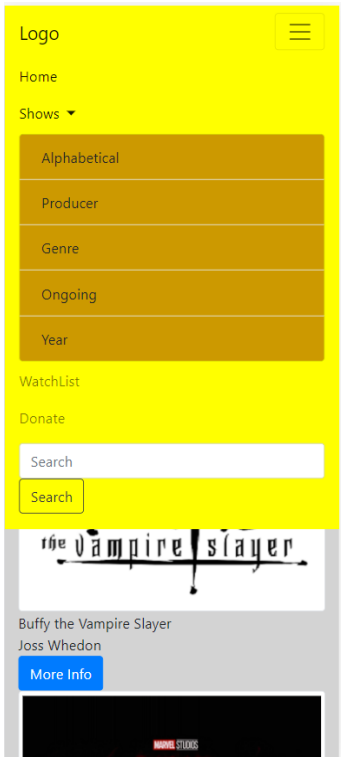


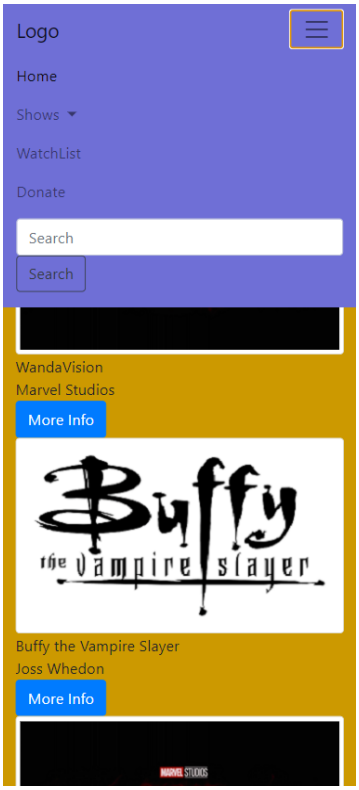

2) Light gray background with a yellow navigation and a brown for the navigation sub-menu.



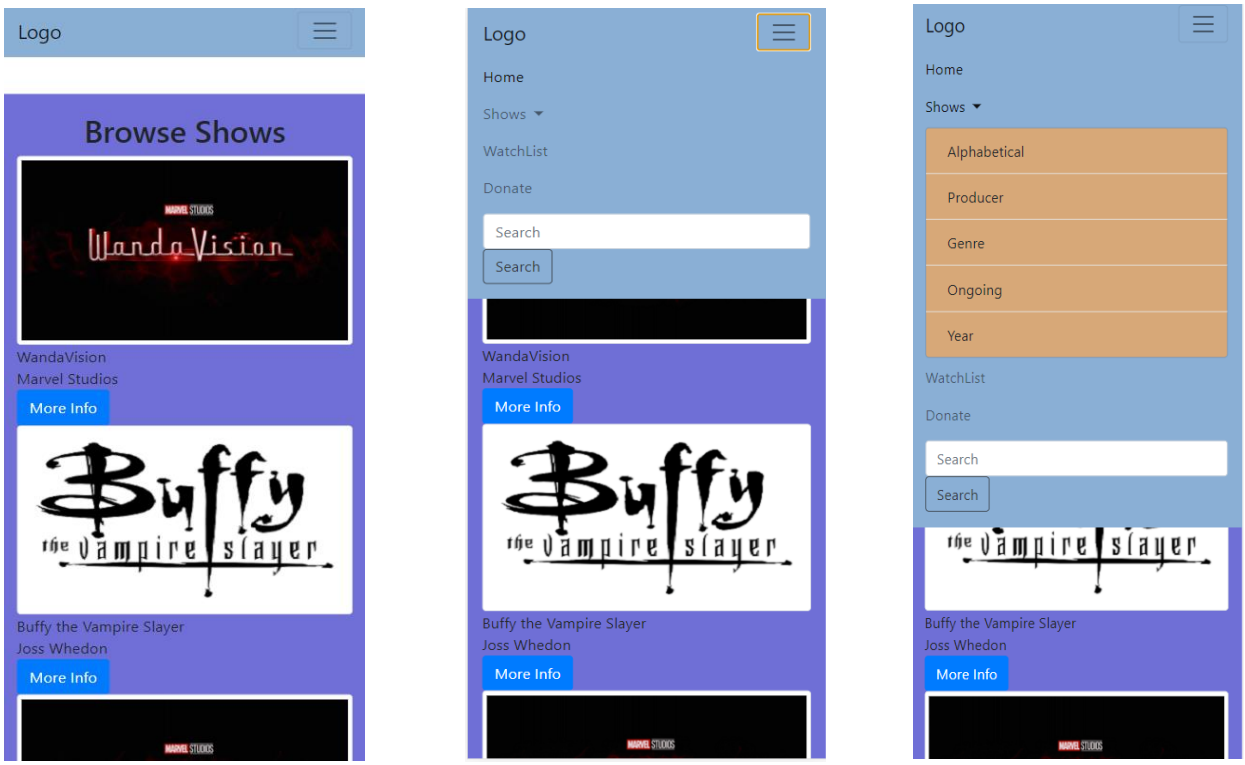




3) Brown background with a purple/blue navigation and dark blue for navigation sub-menu.


4) Purple background with a pale blue navigation and a beige/brown sub-navigation.




Deciding on a color scheme is an important decision to make, as it expresses the website and is analysed in a user's judgement of a website. If discouraged by the theme, users are more likely to avoid the site altogether and find a better service from a competitor. Theme 4 was chosen for the website due to its contrast of colors which are not too harsh on the eye nor too discouraging that they appear off-putting. It does not appear too vibrant and the orange-brown blends nicely with the blue. In theme 3, the contrast between the purple-blue and the dark-blue dropdown menu makes it difficult to make out the black text, making it the least ideal choice. Similarly, the two blue colors of theme 1 also make the text difficult to read, though it does work well against the light grey background of the webpage. Theme 2 was my secondary choice given the yellow pairs nicely with the light grey but if it were to be chosen, the yellow would most likely have been made lighter so it does not look too bright, and the brown would be slightly altered for a cleaner match.

**Logo implementation and font choice:**


As shown, the logo fits nicely into the website's navigation bar on both browser and mobile view. It serves as a button which redirects the user to the homepage. Most sites utilize this feature to ensure

users can always make their way back to the homepage if they happen to get lost. This has been implemented with a single line embedded in the code for the navigation bar:

```
<a class="navbar-brand" href="index.php"><img src="Logo.png" width="50" height="50" alt="Logo"></a>
```

At a size of 50 by 50, the logo fits well so as not to unnecessarily expand the length of the navigation bar and the text is clear enough to be read without any difficulty.

Selecting a font for the website is a more difficult task than just finding one that looks good – fonts affect the readability of website content, have an impact on user experience and complete the design. Users will judge a website based on its design immediately on arrival – font style has a significant impact on the user's opinion, conveying messages regarding trust, stability, sophistication, and creativity. The font must be easy to read across all screen sizes, and when in doubt, Open Sans is versatile and suitable for all websites, making it a safe choice. The font I have chosen is from Google, Merriweather Sans. While basic and of the Sans family of typefaces, it is not too sophisticated and reads well – achieving the goal of conveying readable information to the user.


The font as initially tested on the Google Fonts page.


The font as it appears on the donations page of our website. Clear and unsophisticated so it can clearly be read by the user. The font style will remain consistent across the whole website in accordance with design.

The font has been imported within the header of the website along with the CSS stylesheet:

```
<!-- Google Font import -->
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Merriweather+Sans&display=swap" rel="stylesheet">
```

When using the font for text styling, the following line is required:

font-family: 'Merriweather Sans', sans-serif;

**Image slideshow feature on donations page:**

The website has a donations page which utilizes a payment gateway in the form of a PayPal button. As well as a QR code image and some text, I thought an image slideshow would be a great addition to the page. The slideshow was created with guidance from the W3Schools website (much like the image expansion feature) and would rotate between three images. A line of grey dots beneath the slideshow would highlight progress through the slideshow, and each image was programmed to have

a caption providing background on the image. JavaScript was utilized to get the slideshow functional, combined with HTML and appropriate CSS styling.

```
<!-- JavaScript for image slideshow -->
<script>
var slideIndex = 0;
showSlides();

function showSlides() {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" activeSlide", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " activeSlide";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
```

A timer function within the JavaScript sets the slide rotation at two seconds and it will endlessly loop between the three images in their respective order.

```
    <!-- Image slideshow -->
    <div class="slideshow-container">

    <div class="mySlides fade">
        <div class="numbertext">1 / 3</div>
        <img class="slideImg" src="#" style="width:100%">
        <div class="textCaption">Caption Text</div>
    </div>

    <div class="mySlides fade">
        <div class="numbertext">2 / 3</div>
        <img class="slideImg" src="#" style="width:100%">
        <div class="textCaption">Caption Two</div>
    </div>

    <div class="mySlides fade">
        <div class="numbertext">3 / 3</div>
        <img class="slideImg" src="#" style="width:100%">
        <div class="textCaption">Caption Three</div>
    </div>

    </div>
    <br>

<div class="dots">
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
</div>
<!-- End of image slideshow -->
```

The HTML code shown here has ‘fade’ attached to the div class for each slide. This allows for a smooth transition between each image, rather than just jumping from one to the next. The class ‘numberText’ refers to the slide number which appears at the top left corner of the image and ‘textCaption’ refers to the caption label of each image. The former may be seen as unnecessary, given the grey dots will pinpoint the progress through the slides.

***Note: After discussion, the team decided to remove this feature due to the fact it seemed unnecessary on the page and there was difficulty in deciding what the images would be as they needed to be relevant to the website.***