

Grey Paper

Group Members: Abdul Wahab, Umme Habiba, Haniya Ghauri

1. HTML code for the navigation menu on top. It has four menu options that lead to their own separate pages on the website

```
<nav>
  <ul>
    <!--<h1 class="title">Pick Your Place</h1-->
    <li><a data-active="homePage" href="home.html">Home</a></li>
    <li><a data-active="placesPage" href="places.html">Places</a></li>
    <li><a data-active="feedbackPage" href="feedback.html">Feedback</a></li>
    <li><a data-active="aboutPage" href="aboutUs.html">About Us</a></li>
  </ul>
</nav>
```

2. HTML code for a div that contains three popular places of the week on the homepage

```
<div class="flex-container">
  <div class="flexed-item"> <a href="tourist.html">  </a> <br> <center> Brookly Bridge</center></div>
  <div class="flexed-item"> <a href="restaurant.html">  </a> <br> <center> Le Botaniste </center></div>
  <div class="flexed-item"> <a href="parks.html">  </a> <br> <center> Bryant Park </center> </div>
</div>
```

3. The animation for the hover effect on the nav options on each page. This gives a yellow underline effect on the option that is hovered

```
nav a {
  color: #fff;
  text-decoration: none;
  text-transform: uppercase;
  font-size: 14px;
}

nav a:hover {
  color: #FFEB98;
}

nav a::before {
  content: '';
  display: block;
  height: 5px;
  background-color: #FFEB98;
  position: absolute;
  bottom: 0;
  width: 0%;
  transition: all ease-in-out 250ms;
}

nav a:hover::before {
  width: 100%;
}
```

4. HTML code for the center image (banner) that is on the homepage. It is a banner that has a tagline for our website.

```
<div class="flex-container-banner">
  <div class="banner">  </div>
</div>
```

5. This is the CSS code how the yellow lines are styled for all headings, they vary on size based on the length of the heading

```
hr {
  width: 150px;
  border-radius: 5px;
  border: none;
  background-color: #FEB98;
  height: 3px;
  margin-top: -9px;
}
```

6. CSS code for how the images are displayed and styled. There are three images on homepage that are centered and have a fixed size and added radius

```
.flexed-item img{
  float: left;
  width: 325px;
  height: 325px;
  border-radius: 10%;
  overflow: hidden;
  margin-top: -6px;
  margin-bottom: 10px;
}
```

7. Below is the HTML code for the bucket list on the homepage. This idea was from the udeemy course and it adds locations to your list that you can cross off or delete

```
<div class="bucketlist">
  <h2>Make A List Of The Places You Would Love To Visit</h2>
  <form id="form" action="index.html" method="post">
    <input type="text" class="input" id="input" placeholder="Your Favourite Desitnations" autocomplete="off">
    <ul class="todos" id="todos">

    </ul>
  </form>
  <small>Left click to toggle completed <br>
    click to delete bucketlist</small>
</div>
```

8. CSS code for how the items in the bucket list are made to look crossed out and how they are displayed when added

```
.todos{
  background-color: #fff;
  padding: 0;
  margin: 0;
  list-style-type: none;
}
.todos li{
  border-top: 1px solid #e5e5e5;
  cursor: pointer;
  font-size: 1.5rem;
  padding: 1rem 2rem;
}
.todos li.completed{
  color: #b6b6b6;
  text-decoration: line-through;
}
```

9. This is the html code for displaying the categories on the “places” page. It is in 2 divs and each div contains 3 sub divs as each place

```

<div class="flex-container">
  <div class="flexed-item"> <a href="parks.html">  </a> <br> <p class = "catName"> <br> <br> Park <br><br> </p> </div>
  <div class="flexed-item"> <a href="cafe.html">  </a> <br> <p class = "catName"> <br> <br> Cafe <br> <br> </p> </div>
  <div class="flexed-item"> <a href="library.html">  </a> <br> <p class = "catName"> <br> <br> Library <br> <br> </p> </div>
</div>
<div class="flex-container">
  <div class="flexed-item"> <a href="restaurant.html">  </a> <br> <p class = "catName"> <br> <br> Restaurant <br> <br> </p> </div>
  <div class="flexed-item"> <a href="museum.html">  </a> <br> <p class = "catName"> <br> <br> Museum <br> <br> </p> </div>
  <div class="flexed-item"> <a href="tourist.html">  </a> <br> <p class = "catName"> <br> <br> Tourist Attractions <br> <br> </p> </div>
</div>

```

10. Below is the css code for how the titles of each of the categories in the places page is styled. It has a dark background that is styled to look like its under the image/icon card

```

.catName {
  text-align: center;
  margin-top: 10px;
  padding-top: 10px;
  background: #007B86;
  justify-content: center;
  align-items: center;
  border-radius: 20px;
  color: white;
}

```

11. This is the code for the container that holds the 2 rows of categories. It is styled to center and its flex direction is row because that's how the category cards are laid out on the website

```

.flex-container {
  display: -webkit-flex;
  display: flex;
  -webkit-justify-content: center;
  justify-content: center; /* center horizontally */
  /*width: 100%;*/
  flex-direction: row;
  flex-wrap: nowrap;
  margin-right: 55px;
  margin-left: 55px;
  margin-bottom: 55px;
}

```

12. HTML code for the title and some main highlights of the place in a specific category (in this example its a cafe)

```

<div class="flexed-item" style="width:30%"> <h3> Black Fox Coffee </h3>
  <ul>
    <li> a bright and spacious cafe </li>
    <li> has a full kitchen offering gourmet coffee & breakfast plates in chic digs </li>
    <li> has the best coffee selection day-to-day of any shop in the city </li>
    <li> showcasing coffees from a variety of esteemed roasters around the world </li>
  </ul>
</div>

```

13. This is the html code for the rightmost column of the section on each location page. It displays address, ratings for the place and a button that leads to the feedback page

```

<div class="flexed-item" style="width:22%">
  <h4> Address</h4><a href="https://www.google.com/maps/dir/40.6867129,-73.7
<br>
<h4> Rating</h4>
<center>
<div class="rating">
  <div class="rating-item" style="width:33%"> <h1> 18 </h1> Good </div>
  <div class="rating-item" style="width:33%"> <h1> 6 </h1> Average </div>
  <div class="rating-item" style="width:33%"> <h1> 2 </h1> Bad </div>
</div>
</center>
<br> <a href="feedback.html" class="button">Give Your Feedback</a>
</div>

```

14. Below is the css code for styling the feedback button so it has 2 states, default which is a green color with yellow border and hover state that turns the button yellow

```

.button {
  background-color: #FFEB98;
  background-color: rgb(14, 141, 153, 1);
  color: black;
  border: 2px solid #FFEB98;
  margin-bottom: -2px;
  border-radius: 10px;
}

.button:hover {
  background-color: rgb(14, 141, 153, 0.5);
  background-color: #FFEB98;
  color: black;
  border-radius: 10px;
}

```

15. This css code shows how the ratings are styled. It has one row and three columns for good, average, bad and each have title and numbers.

```

.rating {
  display: -webkit-flex;
  display: flex;
  -webkit-justify-content: center;
  justify-content: center; /* center horizontally */
  align-items: center;
  /*width: 100%;*/
  flex-direction: row;
  flex-wrap: nowrap;
  /*margin-right: 5px;
  margin-left: 5px;*/
  margin-bottom: 5px;
}

.rating-item {
  float: left;
  padding: 5px;
  color: black;
  -webkit-justify-content: center;
  justify-content: center; /* center horizontally */
  align-items: center;
  margin-top: -25px;
  /*margin-right: 1px;
  margin-left: 1px;*/
}

```

16. The following code shows the styling that is added to the title of the page for each location

```
.title {
  float: left;
  padding-top: 10px;
  padding-bottom: 10px;
  padding-left: 12px;
  font-size: 20px;
  font-family: 'Cinzel', serif;
  font-weight: 800;
  color: white;
}
```

17. HTML code for the form on the feedback page. This part of the code is getting text input from user such as their first and last name and name of the location they want to give feedback for

```
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname" placeholder="e.g. John" required>
<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname" placeholder="e.g. Doe" required>
<br><br>

<label for="pname">Place name:</label>
<input type="text" id="placeName" name="placeName" placeholder="e.g. Central Park">
<br><br>
```

18. This code asks for the user's experience given the 3 emojis that are clickable (good/satisfied, average, bad/unsatisfied). This idea was also from udeemy projects..\\

```
<label for="fname">How was your experience?</label>

<div class="radio-image">
  <input type="radio" id="good" class="satisfied" name="satisfied" value="Good">
  <label for="good"></label>

  <input type="radio" id="average" class="satisfied" name="satisfied" value="Average">
  <label for="average"></label>

  <input type="radio" id="bad" class="satisfied" name="satisfied" value="Bad">
  <label for="bad"></label>
</div>
```

19. This is html code that further asks for the experience in a different way. The user is asked to give a rating out of 10 using a scale they can drag. This is another idea from udeemy projects. Got stuck in updating the label everytime the cursor moved but then used javascript to implement this.

```
<label for="rating">Rate your experience out of 10</label>

<div class="range-container">
  <input type="range" id="range" min="0" name="ratings" max="10" value="">
  <label for="range">5</label>
```

20. The following code is for a text box that is draggable and users can explain here their experience in words. It also has a submit button (code will be discussed later)

```

<textarea name = "message" id="message" rows="4" placeholder="e.g. this place has a really nice view..." required> </textarea>

<br>
<br>
<center> <input type="submit" id = "submit" value="Submit"> </center>
<span style="text-align: center" id="success"></span>

```

21. CSS code for how the two sections on about us page are styled (who we are and what we do)

```

#who {
  float: left;
  padding: 5px;
  /*background-color: rgba(154,35,108,0.5);*/
  color: black;
  /*margin: 40px 195px;*/
  margin-right: 200px;
  margin-left: 60px;
  margin-bottom: 55px;
  width: 50%;
}

#what {
  float: right;
  padding: 5px;
  /*background-color: rgba(154,35,108,0.5);*/
  color: black;
  /*margin: 40px 195px;*/
  margin-right: 60px;
  margin-left: 200px;
  margin-bottom: 55px;
  width: 50%;
}

```

22. JavaScript code to connect our to a google sheets file where all feedback responses are saved.

```

const scriptURL = 'https://script.google.com/macros/s/AKfycbyXo0zykVzLsB7PvsvC30HowzSizN06T85B18Skpx-0iNwgYEGJjaavz0dUQ41-FYyV';
const form = document.forms['form-to-sheets']

```

23. The following is a javascript snippet for when the feedback is submitted on the feedback page. On success it adds it to the google sheets displays a success message. On failure it displays a message on console. Got stuck in submitting this request. Had some issues in the code but was able to eventually fix it.

```

const success = document.getElementById('success');
form.addEventListener('submit', e => {
  e.preventDefault()
  fetch(scriptURL, { method: 'POST', body: new FormData(form)})
    .then(response => {
      success.innerHTML = "Feedback Successfully Submitted"

      setTimeout(function(){
        success.innerHTML = ""
      }, 5000)
      form.reset()
    })
    .catch(error => console.error('Error!', error.message))
})

```

24. Javascript code for the range slider. We used a udemy course to get idea for this feature. Users can choose from 1 to 10 for their feedback.

```
//Range Code
```

```
const range = document.getElementById('range')
range.addEventListener('input', (e) => {
  const value = +e.target.value
  const label = e.target.nextElementSibling

  const range_width = getComputedStyle(e.target).getPropertyValue('width')
  const label_width = getComputedStyle(label).getPropertyValue('width')
  const num_width = +range_width.substring(0,range_width.length -2)
  const num_label_width = +label_width.substring(0,label_width.length -2)
```

25. Javascript code for travel bucketlist on the home page. Users can add their top destinations and create a list for it. This code makes a new li tag and adds a new destination. It also has the remove features. Got stuck on how to remove the item from the list. But then came up with the idea to do right click to remove.

```
//for bucketList container
form.addEventListener('submit', (e) => {
  e.preventDefault()
  addToDo()
})

function addToDo(todo){
  let todoText = input.value

  if(todo){
    todoText = todo.text
  }
  console.log(todoText)
  if(todoText){
    const todoEl = document.createElement('li')
    if(todo && todo.completed){
      todoEl.classList.add('completed')
    }
    todoEl.innerText = todoText

    todoEl.addEventListener('click', () =>
      todoEl.classList.toggle('completed'))

    todoEl.addEventListener('contextmenu', (e) =>{
      e.preventDefault()
      todoEl.remove()
    })

    todosUL.appendChild(todoEl)
```

26. Javascript code for the random image generator on home page. Random images are retrieved from unsplash website and added here.

```
//for random image generator
const container = document.querySelector('.image-container')
const unsplashURL= 'https://source.unsplash.com/random/'
const rows = 5

for(let i =0; i< rows*3; i++){

  const img = document.createElement('img')
  img.src = `${unsplashURL}${getRandomSize()}/?travel`
  console.log(img)
  container.appendChild(img)

}
function getRandomSize(){
  return `${getRandomNr()}x${getRandomNr()}`
}
function getRandomNr(){
  return Math.floor(Math.random()*10)+300
}
```

27. Javascript code that has been added to all pages of the project. It provides links to each tab on the website.

```
let links = document.querySelectorAll(".container a");
let bodyId = document.querySelector("body").id;

for (let link of links) {
  if (link.dataset.active == bodyId) {
    link.classList.add("active");
  }
}
```

28. This is the snippet of the node javascript. It allows node server to open and run different types of files such as png, jpeg, html, js, json, etc. At first the team got stuck on rendering other types of files besides html such png, jpeg but then used some youtube ideas to implement this simple solution.

```
const extname = String(path.extname(filePath)).toLowerCase();
let contentType = 'text/html';

switch (extname) {
  case '.html':
    contentType = 'text/html';
    break;
  case '.js':
    contentType = 'text/javascript';
    break;
  case '.css':
    contentType = 'text/css';
    break;
  case '.json':
    contentType = 'application/json';
    break;
  case '.png':
    contentType = 'image/png';
    break;
  case '.jpg':
    contentType = 'image/jpg';
    break;
  case '.jpeg':
    contentType = 'image/jpeg';
    break;
  case '.gif':
    contentType = 'image/gif';
    break;
  default:
    contentType = 'application/octet-stream';
}
```

29. Javascript code for the node file.


```

filePath = path.join(__dirname, filePath);
const readStream = fs.createReadStream(filePath);

readStream.on('error', function (err) {
  if (err.code === 'ENOENT') {
    res.writeHead(404, { 'Content-Type': 'text/html' });
    res.write('<h1>404 Not Found</h1>');
    res.end();
  } else {
    res.writeHead(500, { 'Content-Type': 'text/html' });
    res.write('<h1>500 Internal Server Error</h1> ${err}');
    res.end();
  }
});

res.writeHead(200, { 'Content-Type': contentType });
readStream.pipe(res);
});

```

30. Javascript code to set the node which is 3000

```

const PORT = process.env.PORT || 3000;

server.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});

```