Developer's Manual
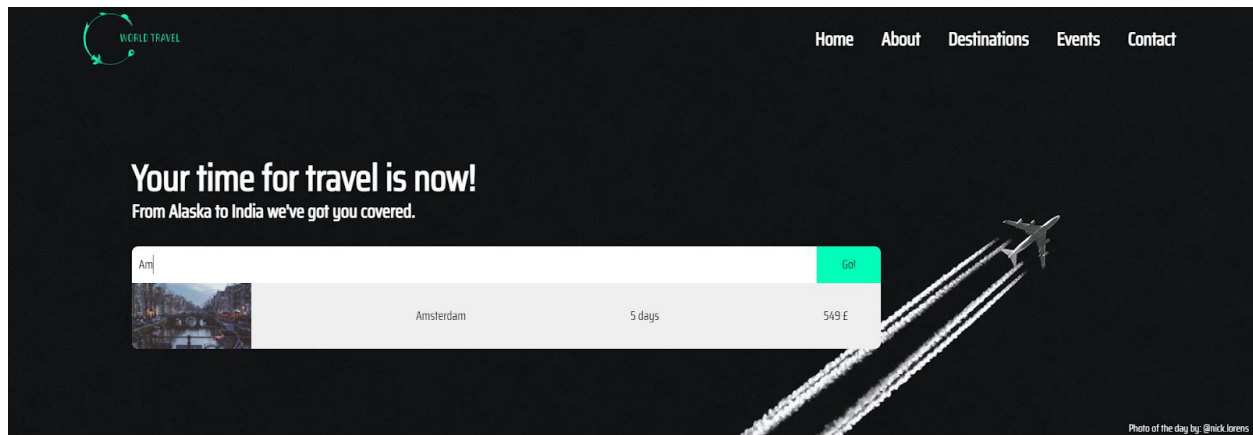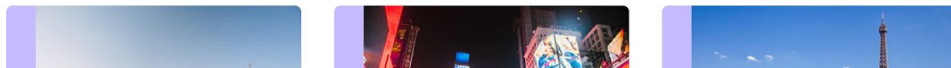# Travel Now Agency

Overview
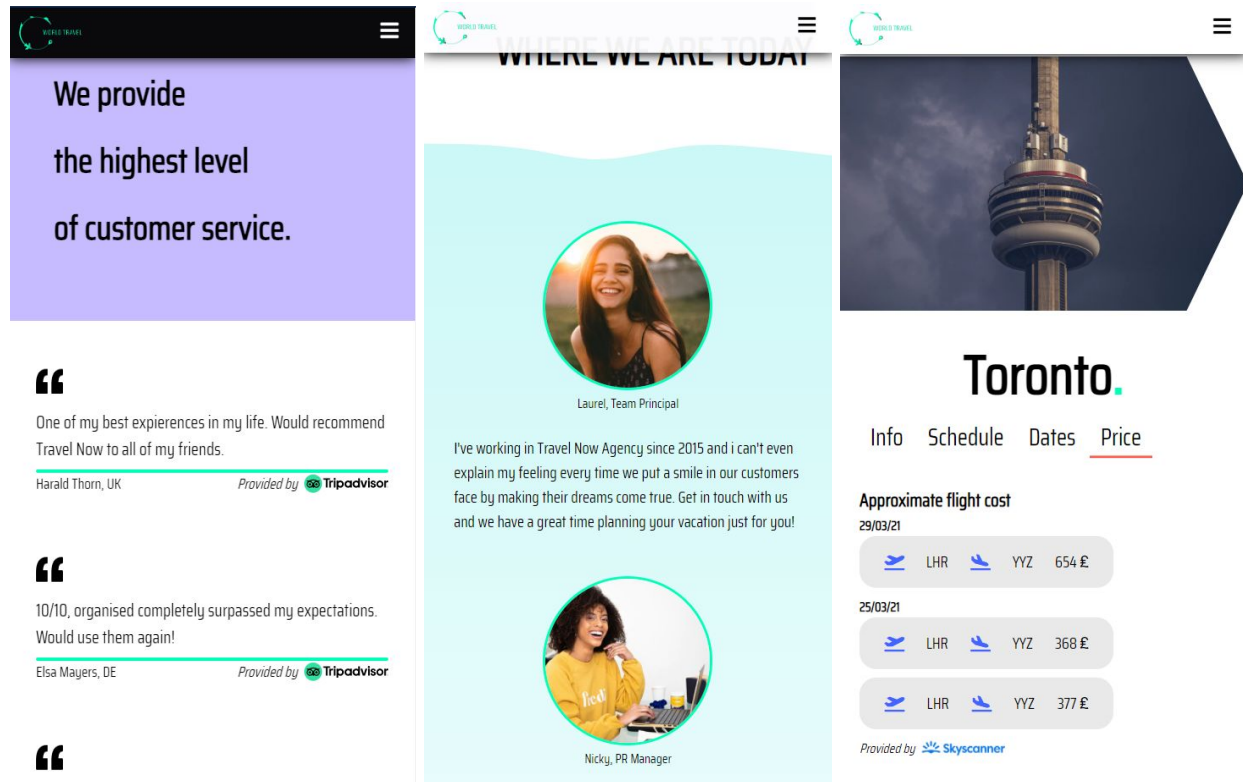

This project is about a Travel Agency named Travel Now Agency that offers destination packages from all around the world. For the purpose of this project only 5 destinations have been developed.

This website has been built using html, css and vanilla javascript. For the majority of the animations I used an open source library (AOS Animate On Scroll).

Responsiveness

I also took great care making the website responsive for some standard screen resolutions depending on the device f.e. Tablets 768px breakpoint, Mobiles 420px breakpoint. Also the website should be responsive for the screen resolutions in between of these breakpoints although its really difficult to test this for that many devices.

Folder structure

assets	File folder					
css	File folder					
images	File folder					
js	File folder					
about	Chrome HTML Document	4 KB	No	11 KB	70%	08/03/2021 15:48
contact-us	Chrome HTML Document	3 KB	No	13 KB	77%	08/03/2021 12:17
destinations	Chrome HTML Document	3 KB	No	14 KB	85%	08/03/2021 15:48
events	Chrome HTML Document	2 KB	No	8 KB	78%	08/03/2021 12:17
index	Chrome HTML Document	4 KB	No	13 KB	76%	08/03/2021 15:48

Note: The assets folder includes general icons and logos as well as the json file used in the search-dropdown and in the destinations page. The scripts used written in Javascript ES6 can be found in the js folder. The images folder contains subfolders of images depending on their usage on the webpage.

Comments

```javascript
const flightPromises = []; //will hold all of our promises from fetch requests to the skyscanner api

const getQuotes = (destination) => {
    const tab = document.querySelector(`#${destination.location} nav ul li:last-child`); //price tab of destination
        tab.addEventListener("click", () => {
            if (document.querySelector(`#${destination.location} .flightQuotes ul`).innerHTML===""){ //if list is empty (no results added yet)
                Promise.allSettled(flightPromises).then(() => { //ensure that the previous fetch requests have been "fullfilled" - wait / async
                    //Set destination parameter
                    quoteQuery.destinationPlace = `${destination.iata}`;
                    //Set date parameter (for every date from location > date > dates json)
                    destination.data.dates.forEach(date => {
                        let quoteURL = SKYURL + "/browsequotes/v1.0"; //quotes endpoint
                        quoteQuery.outboundPartialDate = convertDate(date); //covert date to correct format for the fetch request
                        quoteQuery.inboundPartialDate = calculateDate(quoteQuery.outboundPartialDate, destination.days); //create new date from departure day + days
                        for (const parameter in quoteQuery) { //create url from which to fetch - we could do part of this outside of getQuotes
                            quoteURL += `/${quoteQuery[parameter]}`;
                        }
                        //console.log(quoteURL);
                        requestQuotes(destination.location,quoteURL,date);
                    })
                })
            }
        });
};
```

Note: All html, css and javascript files have been commented appropriately when needed to explain the way it is supposed to work and in which section of the code we currently are.

Explanation of JS scripts

aosGlobalSettings.js

In this file we just pass the appropriate parameters in the AOS library to initialize the animations.

burgerMenu.js

Provides functionality in the burger menu when on tablet and mobile breakpoint.

createDestinationData.js

Fetches the destination data json file and inserts the data on the appropriate section in destination's page. The code has been written in that way that we can have as many destinations, days in schedule and dates as we want.

destinationsMenu.js

Simple script that added functionality to all the tabs of the destinations in the destinations page.

formValidation.js

This script validates the data entered by the user in the contact form using Regex on blur for most of the fields as well as if all the fields have been entered (not empty) on submit.

imageGallery.js

Provides functionality in the image gallery used in the destinations page using an array of images and alt tags.

navbar.js

With this script we get the offset heights of the header photos and the footer in order to show or hide the navigation bar appropriately.

searchDropdownMenu.js

With this script when the user starts typing a certain word/destination in the search input it will return a list of destinations corresponding to the word that has been typed in real time. The results are clickable and the data are provided by the json file and the user will be redirected to the appropriate destination that has been selected in the destinations page. The first 3 results to match the word will be displayed and not all (restricted to be more viewable).

skyScannerQuotesApi.js

In this script we fetch the skyscanner api (via rapidapi) to get flight quotes on certain dates. We first fetch our destination data json file and then we build on the skyscanner api endpoint creating every time the appropriate url to fetch the quotes from. For the flight quote dates we convert the dates to match the request format as well as for the return data we add the appropriate days from the package day length (using the Date class and its methods).