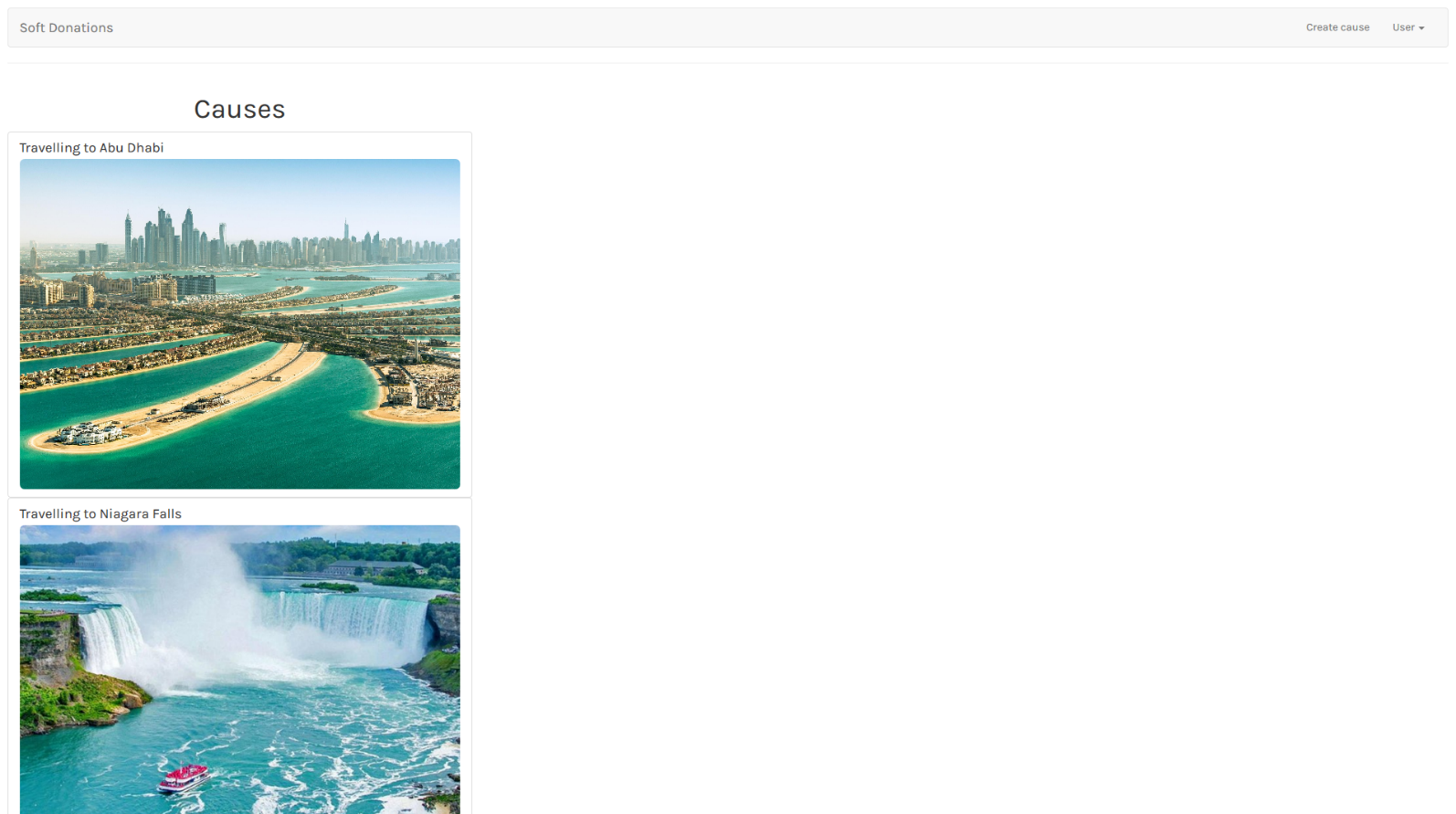
# Soft Donations Workshop - Part 1

This application is created as a workshop for the [Angular Corse](https://softuni.bg/trainings/2760/angular-january-2020) @ [Software University](https://softuni.bg)

# Task Requirements

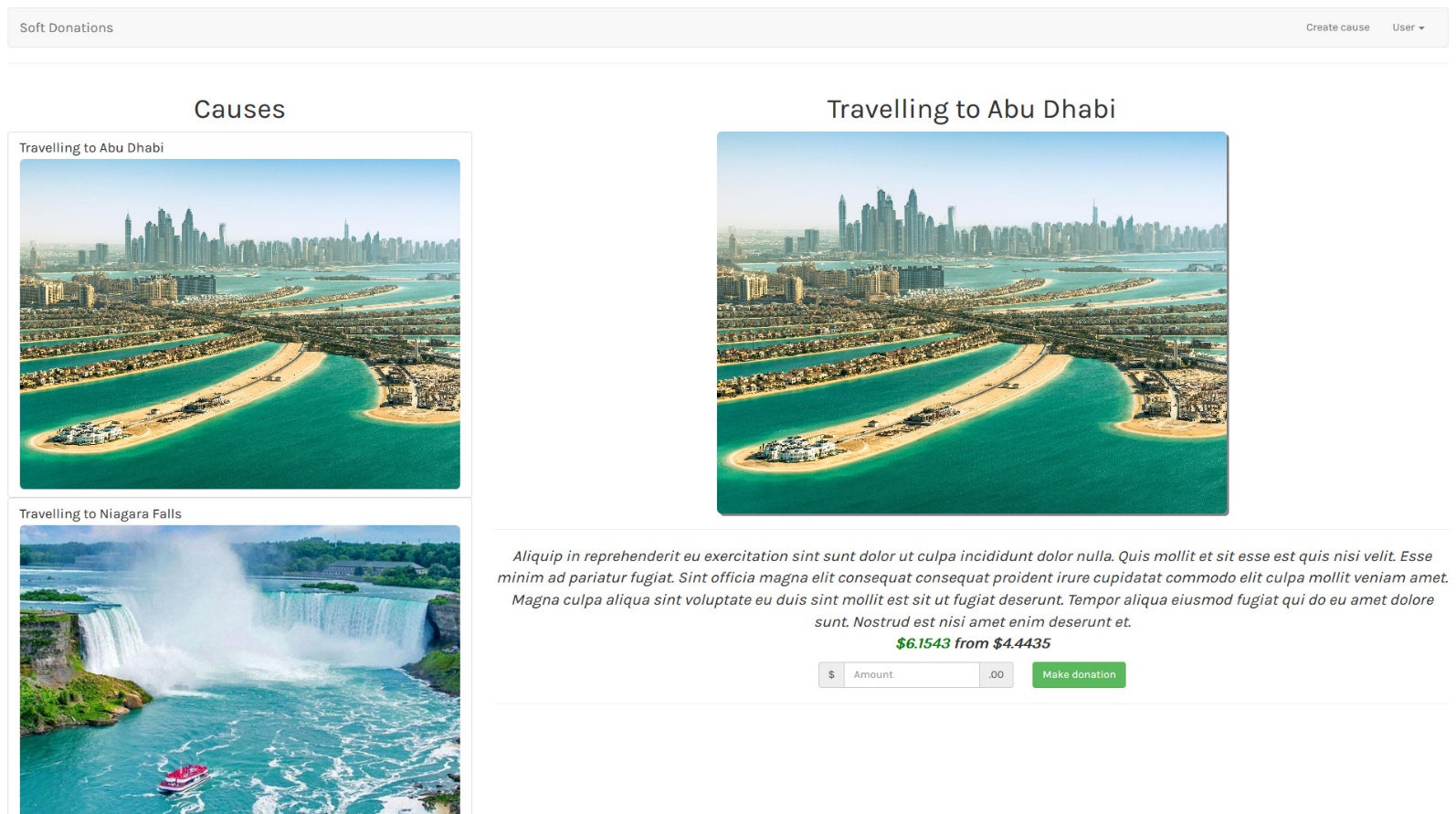
You will be provided with skeleton (**HTML**, **CSS** and **JSON** file). You should install the skeleton dependencies with "**npm install**" to be able to see the state of the application. After successful package installation the following page will appear

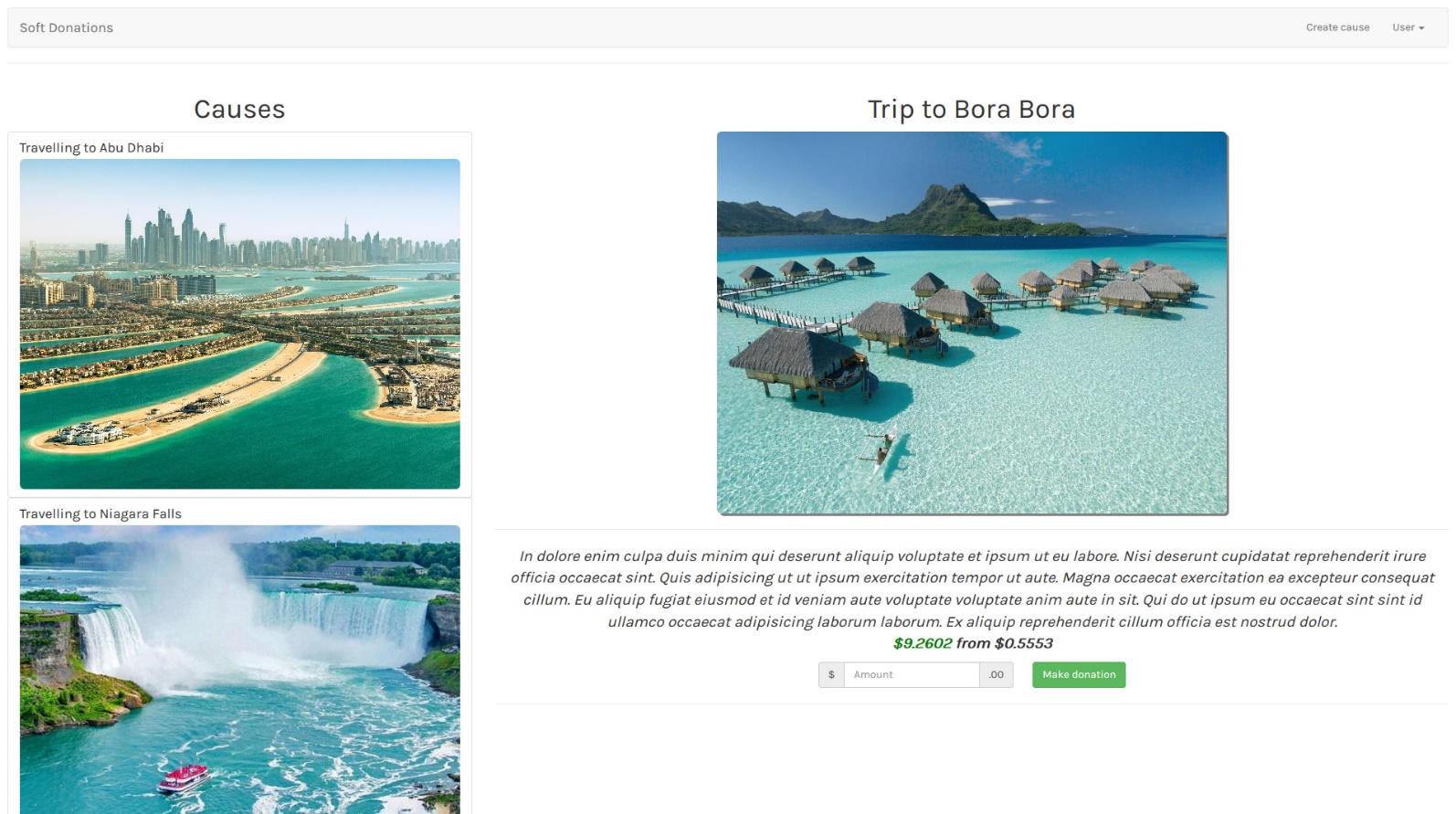


The application is a platform for **donations** where each registered user can create his own cause and the other users can contribute and donate to it. For now, your task is to check the given skeleton (HTML and CSS files) and **split** **them** **into** **components**, so each part can be **reusable**. Be careful when you structure the components.

After that you should **fetch all causes** and **list them** on the home page in the format you saw in the skeleton. (List of all causes on the left side and the details view on the right).

Every cause should have something like "**details view**" so users can read some description about each cause before they donate to it. When some of the causes being clicked, on the right side should be displayed the current cause details view which shows (**Cause name, image url, description, collected amount, needed amount and form for donation**). If other cause is clicked the current details view should be replaced with the information on the last clicked cause.





So far, we don’t saw how to route between pages, so you need to do this with some help with **@Input** and **@Output** **decorators**.

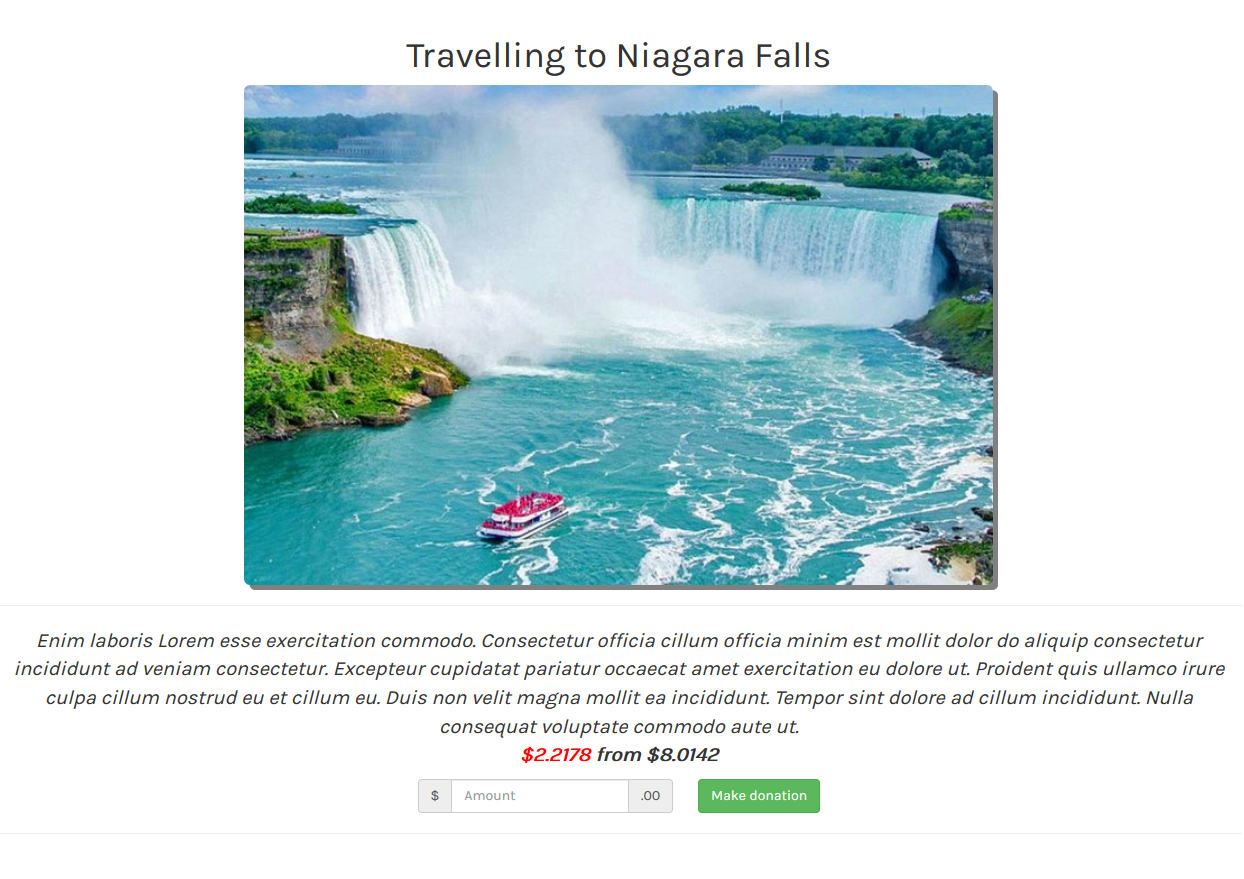
## REST API

You are also provided with **full fake REST API** (JSON file from the skeleton). To bring life on that API you should start it with "**json-server --watch causes.json**". In order to fetch the causes, you will have to make a **GET** **request** on "**localhost:3000/causes**".

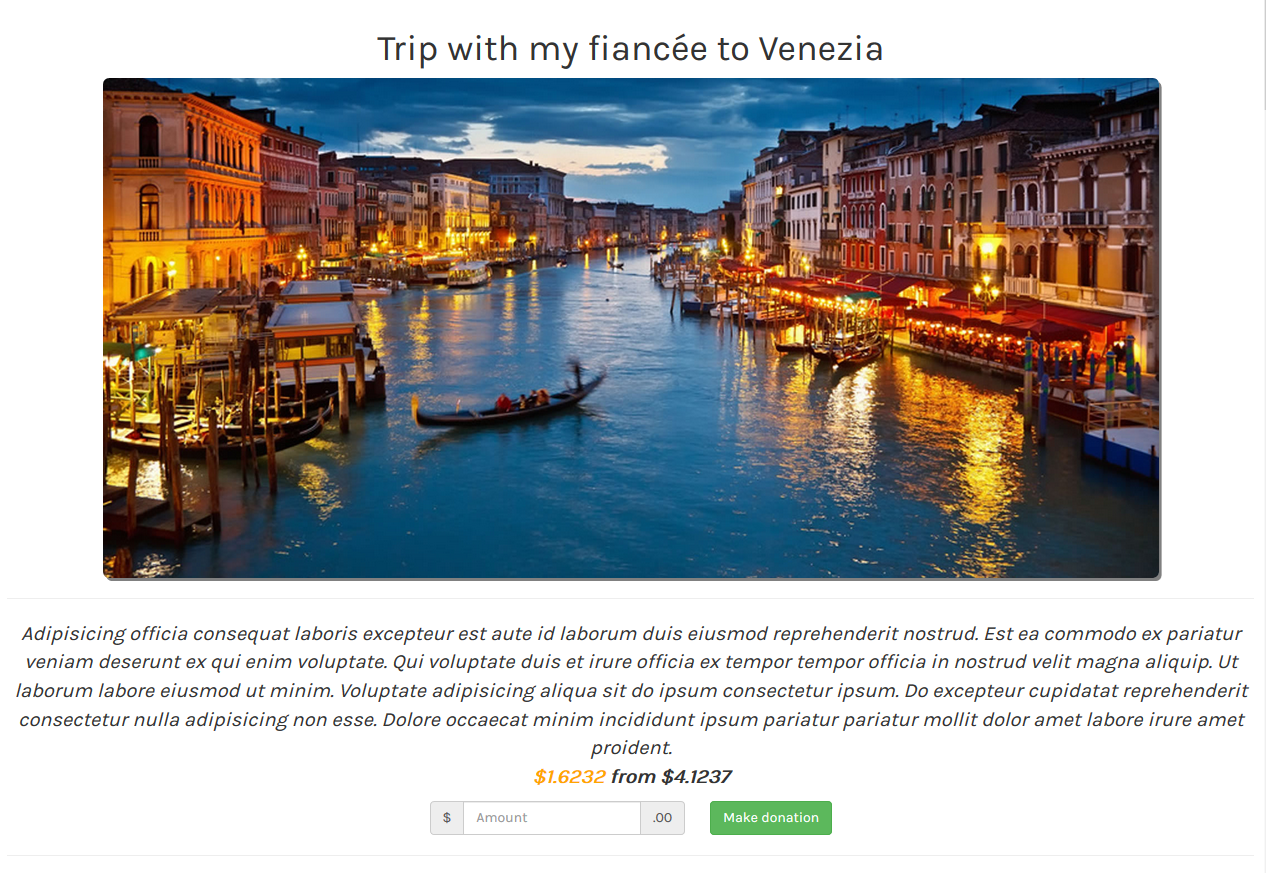
## Implementation

As you already know, each cause has a (**name**, **imageURL**, **description**, **collectedAmount** and **neededAmount**). There are three different cases when you show details for cause

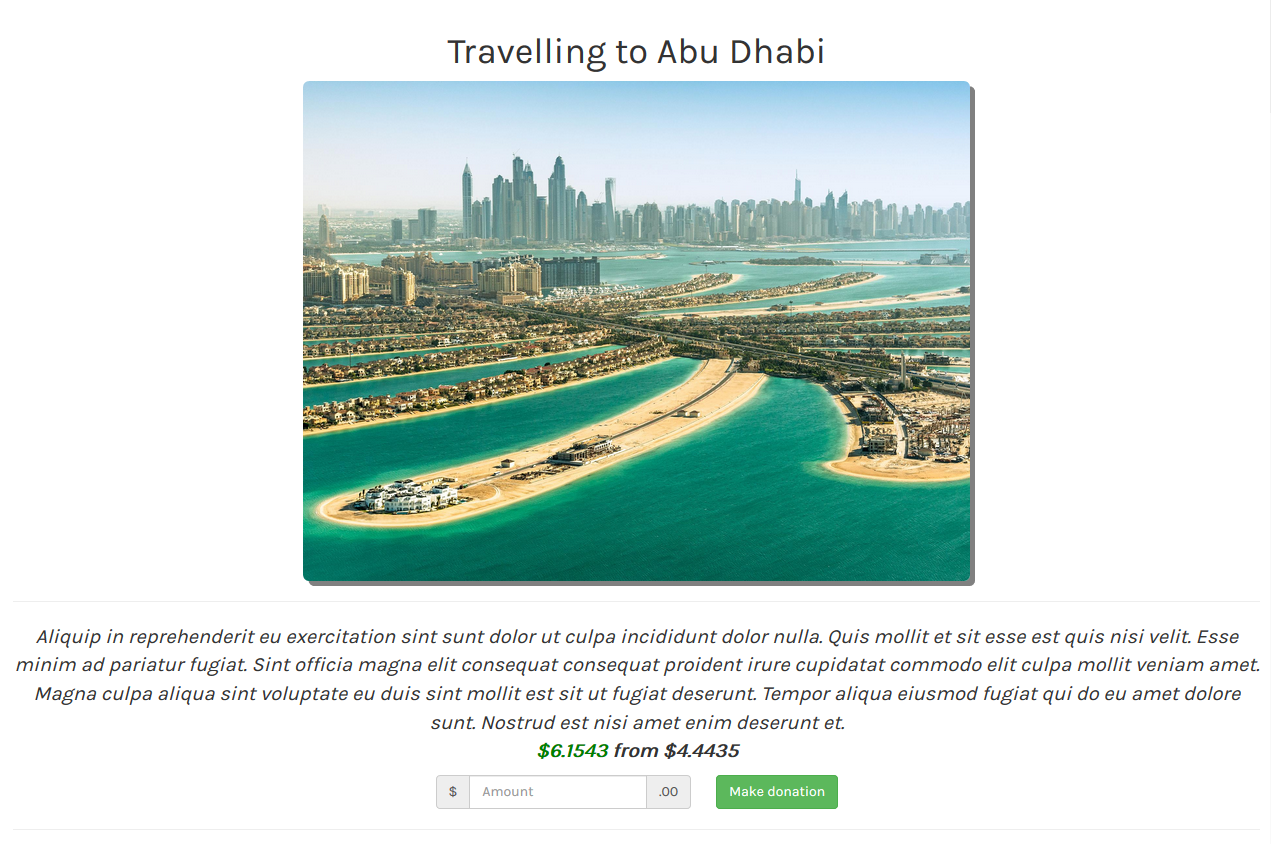
* If the **collected** **amount** is **less** or **equal** than **1/3** of the **needed** **amount** the collected amount should be displayed in **red color**



* If the collected amount is **greater** than **1/3** of the needed amount and **less** than **2/3** of the needed amount. The color should be **orange**



* If the **collected** **amount** is **equal** or **higher** than the **needed** **amount,** the color should be **green**.



### \*\*Note

Check the CSS classes to achieve this effect.

## Services

Try to complete this task using services, which serve some functionality to different components (TS files) via **Dependency** **Injection**.

***GOOD LUCK* 😊**