### **Examen Groepswerk 1: Landen App met OpenStreetMap en Leaflet.js**

## **Inleiding**

In deze opdracht bouw je samen met een teamgenoot een uitgebreide **landen-applicatie** die data gebruikt van de **REST Countries API**. Naast het tonen van landinformatie in een lijst en modal, voeg je een interactieve kaart toe aan de modal met behulp van **OpenStreetMap** en **Leaflet.js**.

### **Wat is OpenStreetMap?**

**OpenStreetMap (OSM)** is een gratis en open-source wereldkaart die door een wereldwijde gemeenschap van gebruikers wordt bijgehouden. Het is een toegankelijk alternatief voor commerciële kaartdiensten en vereist geen API-key. Met OSM kun je kaarten naadloos integreren in je eigen applicaties.

### **Wat is Leaflet.js?**

**Leaflet.js** is een populaire JavaScript-bibliotheek waarmee je interactieve kaarten kunt maken. Het werkt perfect samen met OpenStreetMap en stelt je in staat om markers, popups en andere interactieve elementen toe te voegen aan je kaart. Leaflet is lichtgewicht, flexibel en eenvoudig te gebruiken in moderne webapplicaties.

## **Opgave**

Bouw een **landen-applicatie** die de volgende functionaliteiten bevat:

1. **Lijstweergave van landen**:
   * Toon een overzicht van landen in **Bootstrap-cards** of een lijstweergave.
   * Elk item bevat:
     + De naam van het land
     + De vlag
     + Regio
     + Populatie
2. **Zoek- en filterfunctie**:
   * Voeg een zoekbalk toe waarmee gebruikers landen kunnen zoeken op naam.
   * Voeg een filter toe waarmee gebruikers landen kunnen filteren op regio (bijvoorbeeld "Europe", "Asia").
3. **Details via modal**:
   * Wanneer een gebruiker op een land klikt, opent een **modal** die meer informatie toont, zoals:
     + Hoofdstad
     + Talen
     + Valuta
     + Populatie
     + Vlag
   * Integreer in de modal een **kaart** (via Leaflet.js) waarop de locatie van het land wordt weergegeven met een marker.
4. **Kaartintegratie**:
   * Gebruik de coördinaten (latlng) uit de **REST Countries API** om een marker op de kaart te plaatsen.
   * Zorg ervoor dat de kaart automatisch inzoomt op de locatie van het land.
5. **Robuustheid en foutafhandeling**:
   * De applicatie wordt op alle aspecten getest, waaronder:
     + Wat gebeurt er als een gebruiker verkeerde gegevens invoert?
     + Hoe reageert de applicatie als er ontbrekende data in de API zit, zoals geen coördinaten of ontbrekende hoofdstad?
   * Voeg logica toe om gebruikersvriendelijke foutmeldingen te tonen wanneer er iets misgaat.
6. **Live Publicatie op Syntrawestcloud.be**:
   * De applicatie moet gepubliceerd worden op **syntrawestcloud.be**.
   * Zorg dat je de live URL deelt bij de indiening.

## **Instructies voor Indienen**

1. **Indieningstermijn**:
   * **Openen**: 5 december om 09u00.
   * **Sluiten**: 5 december om 16u30.
2. **Wat moet worden ingediend?**
   * De link naar de live applicatie gepubliceerd op **syntrawestcloud.be**.
   * De link naar de gedeelde Git-repository.
3. **Hoe indienen?**
   * Stuur een e-mail naar **syntraprogrammeurs@gmail.com** met:
     + De **titel**: Groepswerk 1 - Landen App.
     + De live link naar de applicatie.
     + De link naar de Git-repository.

## **Eisen**

1. **Functionele eisen**:
   * De lijstweergave toont alle landen met relevante informatie.
   * De zoek- en filterfunctie werken dynamisch.
   * De modal toont uitgebreide details over een geselecteerd land.
   * De kaart in de modal werkt correct en bevat een marker op de juiste locatie.
   * Er is foutafhandeling voor ontbrekende of foutieve data.
2. **Technische eisen**:
   * Gebruik **REST Countries API** (https://restcountries.com/v3.1/all) om de data op te halen.
   * Integreer **Leaflet.js** en **OpenStreetMap** om de kaart te implementeren.
   * Gebruik **Bootstrap** voor de layout en styling.
   * Voeg eigen **CSS** toe om de applicatie een unieke uitstraling te geven.
3. **Samenwerkingseisen**:
   * Werk samen via een gedeelde **Git-repository**.
   * Gebruik branches voor nieuwe features en maak pull requests om wijzigingen samen te voegen.

## **Beoordeling**

De applicatie wordt volledig getest. Zorg ervoor dat de volgende punten robuust en foutloos werken:

1. **Functionaliteit (40%)**:
   * Werkt de lijstweergave?
   * Zijn de zoek- en filterfuncties correct geïmplementeerd?
   * Worden details in de modal correct getoond?
2. **Kaartintegratie (40%)**:
   * Wordt de kaart correct geladen in de modal?
   * Wordt de juiste locatie met een marker weergegeven, ook wanneer data ontbreekt of foutief is?
3. **Robuustheid en foutafhandeling (10%)**:
   * Is de applicatie bestand tegen verkeerde invoer van gebruikers?
   * Hoe wordt ontbrekende of foutieve data afgehandeld?
4. **Samenwerking (10%)**:
   * Is de Git-repository correct gebruikt?
   * Zijn er duidelijke commits en pull requests?

## **Voorbeeldscenario**

* Een gebruiker opent de applicatie en ziet een lijst met landen.
* De gebruiker zoekt op "Belgium" en klikt op het land.
* Een modal opent met details over België, zoals:
  + Hoofdstad: Brussel
  + Regio: Europa
  + Populatie: 11.5 miljoen
  + Talen: Nederlands, Frans, Duits
* In de modal ziet de gebruiker een kaart met een marker op de locatie van België.
* Als een land geen coördinaten heeft, toont de applicatie een melding: "Locatiegegevens niet beschikbaar."

## **Tips**

1. **Test vroeg en vaak**:
   * Controleer hoe de applicatie reageert op foutieve of ontbrekende data.
2. **Focus op gebruikersvriendelijkheid**:
   * Voeg duidelijke foutmeldingen toe.
3. **Werk samen en communiceer**:
   * Gebruik Git om efficiënt samen te werken.

## **Veel succes!**

Werk efficiënt samen, test je applicatie uitgebreid, en zorg voor een robuuste en gebruiksvriendelijke applicatie. We kijken uit naar jullie creatie! 🚀