GREENPRINTS BIOREGIONS MAP – MANUAL

CONTENTS

1. USING THE MAP

The Map can be accessed by visiting the following link: www.greenprints.org.au/map-app.
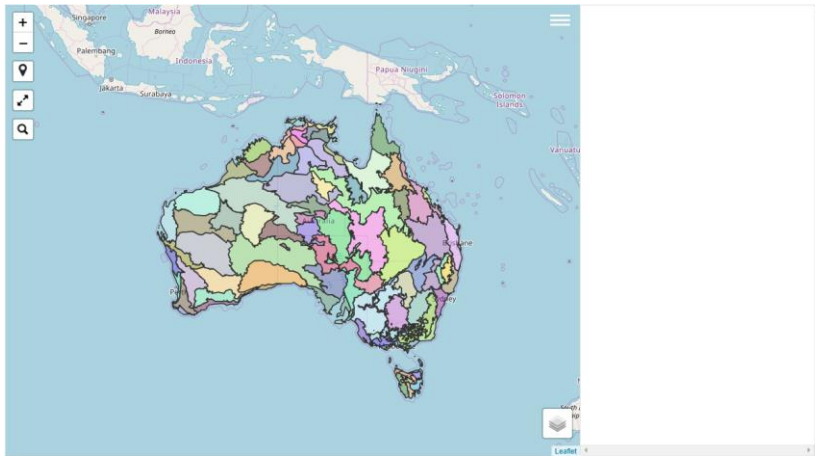
Image 1

MAP CONTROL

The top left corner has four buttons to help navigate through the Map:

- **Zoom level**: You can zoom in by clicking '+' and out by clicking '-'. You can also use the mouse wheel.
- **Geolocation**: Clicking this will show your location on the map. You might have to allow the site to access your location.
- **Full screen toggle**: Click this to see the map in full screen. Click again to exit full screen.
- **Search:** Use this to search for addresses.

LAYER CONTROL

The top right corner of the map has four checkboxes to toggle between both layers for better visibility (Image 2).

You can access by clicking this button .

Image 2

BIOREGIONS

When it first loads, the map shows the Bio-regions of Australia (see Image 1). Clicking in a Bio-region will show a pop up with the word More and information on the right side of the Map. (image 3)

The pop-up shows information about the fauna in that region.

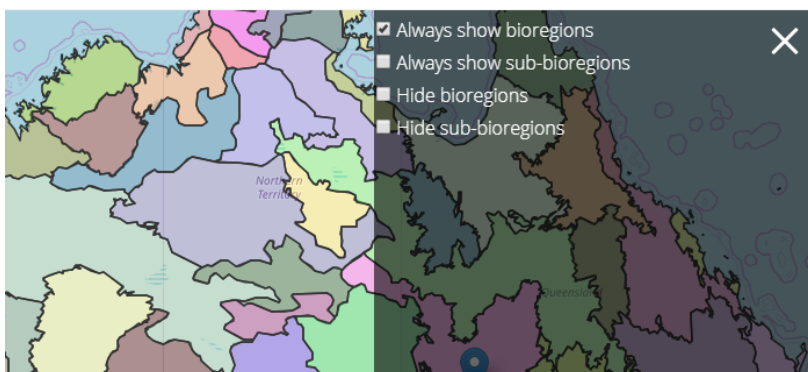The name of the region will show on the right side. Below it additional information will show if previously added. This information comes from the submissions made by the users and admin team of Greenprints.

If there's no information about that region, then the following message will appear: "*This region currently has no information. You can add information about this bioregion here*".

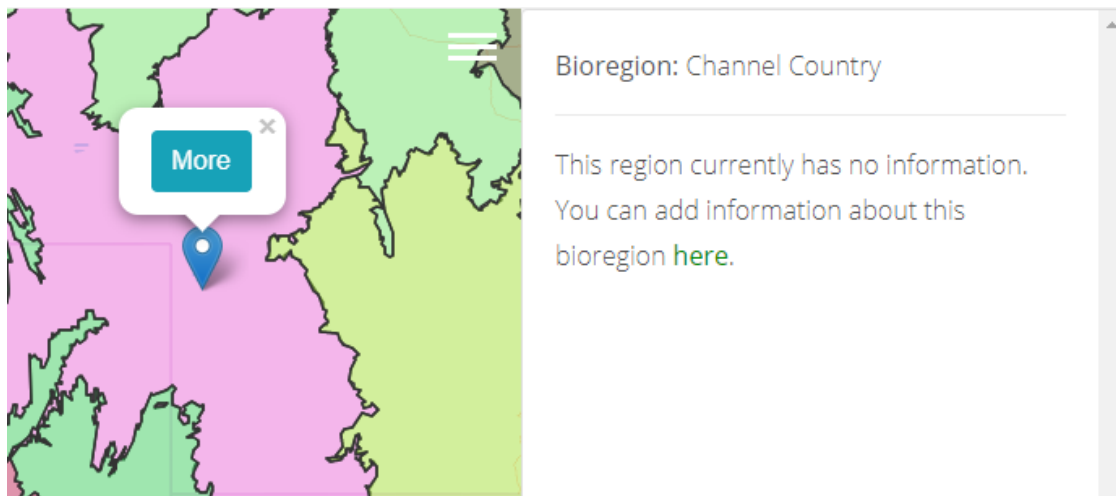With "*here*" being a link to the submissions page.

Image 3

SUBREGIONS

The Sub-bioregions can be visible by zooming in into a region or activating the layer from the Layer Control.

Clicking in a subregion will activate the same amount of information as the Bioregions as well as the name of the Sub-bioregion selected.
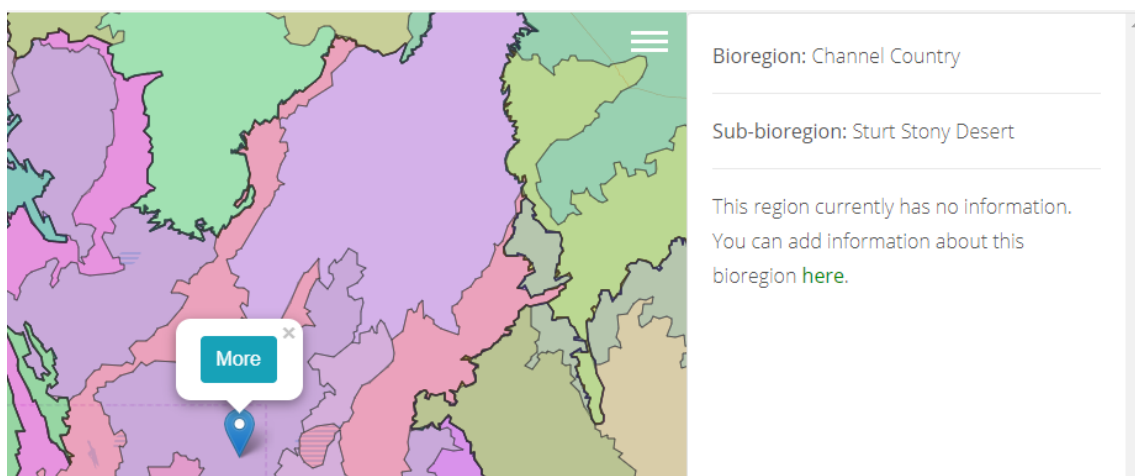
Image 4
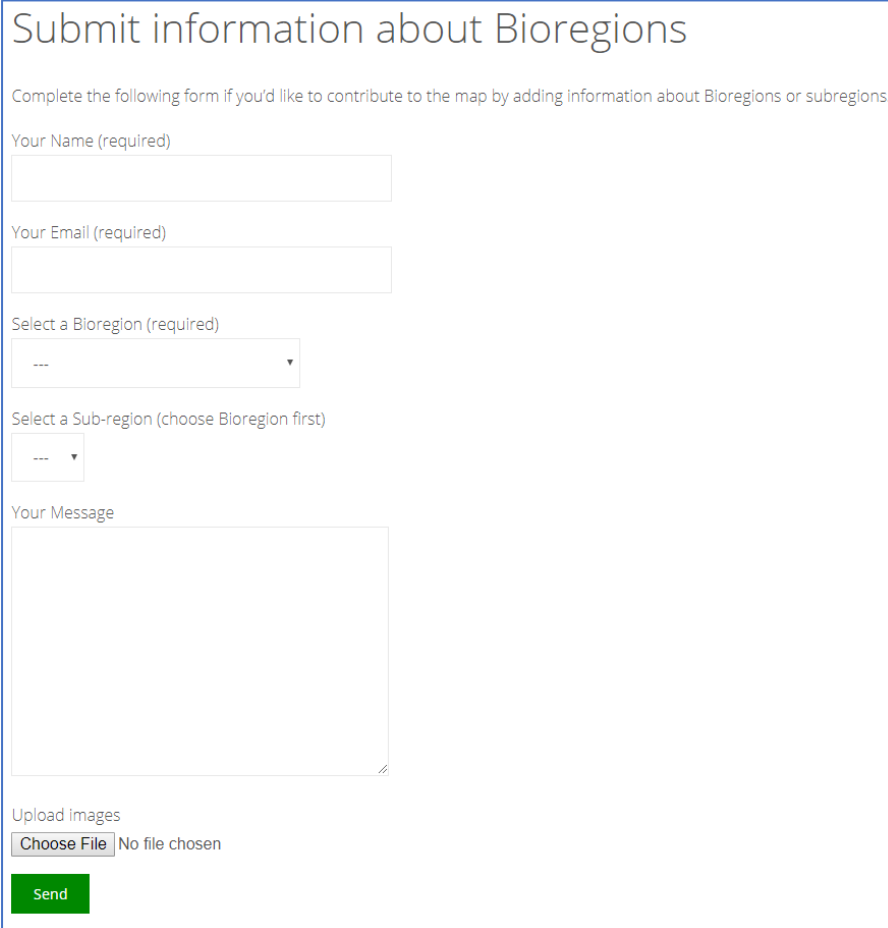
2. ADDING INFORMATION (USER)

Users can add information about a specific Bioregion or Sub-bioregion through the following link: www.greenprints.org.au/submit-bioregions.

There they will find a form (image 5) where they can submit information and images about a Bioregion.

Once the form is submitted it is saved and emailed to the admin's account.

Submit information about Bioregions

Complete the following form if you'd like to contribute to the map by adding information about Bioregions or subregions.

Your Name (required)

Your Email (required)

Select a Bioregion (required)

--- ▼

Select a Sub-region (choose Bioregion first)

--- ▼

Your Message

Upload images

Choose File | No file chosen

Send

Image 5

3. ADDING INFORMATION (ADMIN)
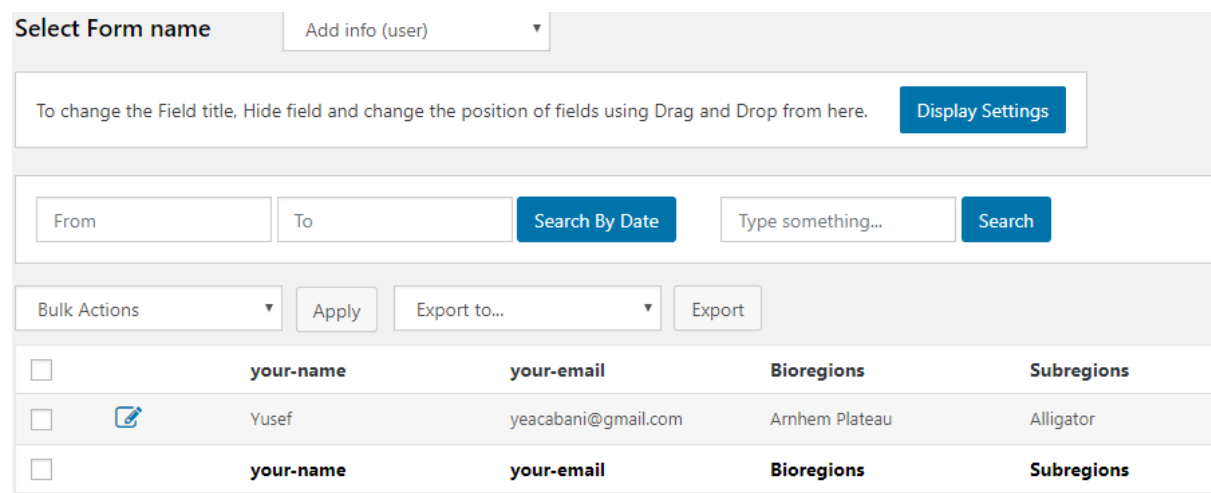
MANAGING THE SUBMISSION FORM

The submission form can be found under "Contact/Contact Forms" on the right side of the WordPress Dashboard. The name of the form is "Add info (user)". Clicking the title (or "Edit") will open the form to give the option to add or edit the fields.

The next tab "Mail" opens the settings to set up the email that will be sent to the admin once the user submits the form. Here you can choose the email account where the email will be sent as well as the body of the email with the content of the submission. You can also add the images as attachments.

WHERE CAN I SEE THE INFORMATION SUBMITTED BY THE USERS?

When the user submits information about a Bioregion, the data gets sent to the email account set in the "Add Info" Contact Form Settings and gets display in the plugin called "Advanced CF7 DB".

Image 6

CREATING WORDPRESS POST

Once the information has been submitted by the user and the admin has received the email, it can be added to the map by creating a Post through the WordPress Dashboard.

Important: The post should have the name of the Bioregion as the title and "Regions Info" as the category.

4. ACCESSING THE MAP DATA

Using the cPanel the map data can be found under '/public_html/map-app'. This folder contains the HTML and CSS data as well as the resources needed for the map.

Note: Access to the full repository can be found here: https://github.com/Yuseldin/Greenprints

FOLDERS

- **dist:** Contains 'bundle.js'
- **leaflet-providers:** Contains the resources for the tile layers (base maps) used in the map.
- **leaflet-search**: Contains the resources and styling of the search bar. Note: the search function is inside bundle.js (main.js)

Outside of the 'map-app' folder there's another folder containing important data:

- **'/public_html/submission-form'** contains two files which make the dropdowns for the Regions and Subregions in the Submission form possible. The Subregion dropdown only shows the relevant Subregions depending on the Region selected above.

FILES

- **Index.php** contains all the HTML in the map-app page except for the header and footer, which are part of the theme and are included using get_header() and get_footer(). This also calls the bundle.js script at the bottom of the body.
- **Head.php** is included in a separate file to prevent conflicts with the theme. The resources needed for index to work are inside this file.
- **Style.css** is used for the styling of this page.
- **regions.json, subregions.json & subregions_simplified.json:** This files contain the layer data for the regions and subregions, including the coordinates and names.

JAVASCRIPT

The development was done using NodeJS and Webpack to bundle all the JavaScript into one file. The readable JavaScript file can be found in the GitHub repository which can be cloned to have full access to all the files.

The file will be under '/src'. There are two files here:

- utils.js : Contains the HTTPRequest needed for calling the layers, the Bioregion info saved in posts and the data from ala.org.au (Atlas of Living Australia).
- main.js : This is the main JavaScript file where all the data is loaded.

5. MAIN.JS

While the code will have comments to help with future development, there are a few blocks of code worth mentioning.

INITIALIZE MAP

This is the first important function since it loads the base map and its functionality.

```
/**
 * Initialize map.
 *
 */
main.prototype.initMap = function() {
    this.map = L.map('mapid').setView([DEFAULT_LAT, DEFAULT_LNG], DEFAULT_ZOOM);
    this.zoomLevels = {
        start: this.map.getZoom(),
        end: this.map.getZoom()
    }

    var defaultLayer = L.tileLayer.provider('OpenStreetMap.Mapnik').addTo(this.map);
```

Image 7- This loads the map with the zoom limits around Australia and the default base layer. The default values can be found at the top of the file.

```
/**
 * ref: https://github.com/leaflet-extras/leaflet-providers
 */
let baseLayers = {
    'OpenStreetMap Default': defaultLayer,
    'Esri WorldStreetMap': L.tileLayer.provider('Esri.WorldStreetMap'),
    'Esri WorldImagery': L.tileLayer.provider('Esri.WorldImagery'),
    'Esri DeLorme': L.tileLayer.provider('Esri.DeLorme'),
    'Esri WorldTopoMap': L.tileLayer.provider('Esri.WorldTopoMap'),
    'Esri WorldTerrain': L.tileLayer.provider('Esri.WorldTerrain'),
    'Esri WorldShadedRelief': L.tileLayer.provider('Esri.WorldShadedRelief'),
    'Esri WorldPhysical': L.tileLayer.provider('Esri.WorldPhysical'),
    'Esri OceanBasemap': L.tileLayer.provider('Esri.OceanBasemap'),
    'Esri NatGeoWorldMap': L.tileLayer.provider('Esri.NatGeoWorldMap'),
    'Esri WorldGrayCanvas': L.tileLayer.provider('Esri.WorldGrayCanvas'),
    "NASAGIBS": L.tileLayer.provider('NASAGIBS.ViirsEarthAtNight2012'),
    'OpenStreetMap Black and White': L.tileLayer.provider('OpenStreetMap.BlackAndWhite'),
    'OpenStreetMap H.O.T.': L.tileLayer.provider('OpenStreetMap.HOT'),
    'Stamen Toner': L.tileLayer.provider('Stamen.Toner'),
    'Stamen Terrain': L.tileLayer.provider('Stamen.Terrain'),
    'Stamen Watercolor': L.tileLayer.provider('Stamen.Watercolor')
}
```

Image 8 – The rest of base layers added into the map. Using the strings from the leaflet-providers.js file.

```
let searchCtlOption = {
    url: 'https://nominatim.openstreetmap.org/search?format=json&q={s}',
    jsonpParam: 'json_callback',
    propertyName: 'display_name',
    propertyLoc: ['lat','lon'],
    marker: L.marker([0,0]),
    autoCollapse: true,
    autoType: false,
    minLength: 2,
    zoom: 10
};
```

Image 9 – Search function using Nominatim as the source for the addresses.

ADD LAYERS

Using the HTTPRequest in util.js we get the layers with the links to the json files as endpoints.

```
//Get geojson for the layers
main.prototype.initCarto = function() {
    sendRequest({method: "GET", url: 'https://www.greenprints.org.au/map-app/regions.json'})
    .then((data) => this.handleGeoJson(data, this.onEachFeatureRegions.bind(this), {
        color: '#333',
        weight: 1.5,
        opacity: 1,
        fillOpacity: 0.4
    }))
    .then((layer) => {
        this.regions = layer;
        this.regions.addTo(this.map)
    })
```

Image 10- Adding Bioregions from the json file

SHOW INFORMATION

The information displays once the user clicks on a region, so we bind a click event for each feature in the json file.

```
//Handler for click events for each feature in geojson layer (Regions)
main.prototype.onEachFeatureRegions = function(feature, layer) {
    layer.on({
        click: (e) => {
            this.currentRegionName = e.target.feature.properties.n;
            this.detailElement.innerHTML = '<strong>Bioregion: </strong>'+this.currentRegionName+'<hr/>';

            if (this.marker != undefined) {
                this.map.removeLayer(this.marker);
            }

            this.marker = L.marker(e.latlng).addTo(this.map);
            this.marker.bindPopup(this.showMoreButton).openPopup();
```

Image 11- Click event for Bioregion layer that creates a marker a shows the name.

```
//Get information from posts about bioregions
var titlePostRegion = this.currentRegionName.replace(/ /g,"-");
let url =  'https://www.greenprints.org.au/wp-json/wp/v2/posts?categories=39&slug='+titlePostRegion;

if (!this.data[titlePostRegion]) this.data[titlePostRegion] = {};
if (this.data[titlePostRegion].loading == true) return;

if (this.data[titlePostRegion].data) {
    this.detailElement.innerHTML += this.data[titlePostRegion].data
} else {
    this.data[titlePostRegion].loading = true;

    sendRequest({method: 'GET', url})
    .then((result) => {
        let data = JSON.parse(result);
        if (!data.length) {
            this.detailElement.innerHTML += '<p>This region currently has no information. You can add
        } else {
            this.data[titlePostRegion].data = data[0].content.rendered;
            this.detailElement.innerHTML += this.data[titlePostRegion].data
        }

        this.data[titlePostRegion].loading = false;
    }, () => {
        this.data[titlePostRegion].loading = false;
    })
}
```

Image 12- Inside the click event, using REST API to access the posts as json files using the link above and filtering by category and title. Category 39 is "Regions info". If there's no post with that title it shows a message with a link to the submission form.