Introduction

Mashup Purpose & description

The purpose of the mashup is to provide an easy format to view and gain information on conflicts around the world. The application shows an interactive map with the location of conflicts, and when selected, displays the latest information, images and tweets that are relevant to the selected conflict. These images and tweets are shown in a list on the left-hand side of the page, and the map with markers is shown on the right-hand side of the page. A map like this is important to increase the awareness of both past and current events, since larger conflicts attract the most news attention and the news cycle is quick to move on, especially when the occurring events have minimal impact on the lives of those in the "first-world". Similar maps do exist such as the *Global Conflict Tracker* run by the Council on Foreign Relations, which serves a very similar purpose. It doesn't show recent tweets about the conflict but does have more information and a nicer UI. Additionally, the application has a feature which determines the current location of the user and shows the conflicts closest to them. This in the hopes that having unknown conflicts that are so close may trigger them to explore the site and end up learning more about our world.

Services used

Twitter Recent Search API (v.2)

Twitter is a social media platform where users can post short form text-based content, occasionally accompanied by media. The API returns a collection of relevant Tweets from the last 7 days matching a specified query. This query can filter based on the terms, hashtags, media type, tweet type, language, time, geocoding, polls and relevancy. The results contain the tweet, author, time, media links, location, metrics and more.

- Endpoint: https://api.twitter.com/2/tweets/search/recent
- Docs: https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction

ACLED REST API v2.5

ACLED is the Armed Conflict Location and Event Data Project, which collects data on reported conflicts and makes weekly updates on the state of the world. The API returns a collection of conflicts that have or are occurring all over the world. The conflict data includes the date, location, type, actors, fatalities, source, description and more. Queries can filter based on all entries of returned data.

- Endpoint: https://api.acleddata.com/{data}/{command}
- Docs: https://www.acleddata.com/wp-content/uploads/dlm_uploads/2017/10/API-User-Guide.pdf

Flickr Photos Search API

Flickr is a photo-sharing platform and social network where users upload photos for others to see. The API returns a collection of images and their meta-data based on a query. The query method of search allows for searching by authors, tags, dates, geo-tagged status, media type and text, as well as be sorted.

- Endpoint: https://api.flickr.com/services/rest/
- Docs: https://www.flickr.com/services/api/flickr.photos.search.html

Leaflet API V1.8.0

Leaflet is an open-source JavaScript library for interactive maps. The API provides an interactive map that can be modified to display information. It gathers the map tile data from the OpenStreetMap API to show streets and detailed information. The map can be edited by adding pins, shapes, areas and popups.

- Endpoint: https://unpkg.com/leaflet@1.8.0/dist/leaflet.js
- Docs: https://leafletjs.com/reference.html

OpenStreepMap API v0.6

The open street map API returns the titles that show the streets and places that are in the leaflet map. The information can be returned in a range of formats. Leaflet handles the calls to the API to gather the tiles.

- Endpoint: https://tile.openstreetmap.org/
- Docs: https://wiki.openstreetmap.org/wiki/API_v0.6

IP API

The IP-API returns an estimated location of an IP address sent to it. The location returned can include data about country, city, latitude, longitude and isp in either XML, XML, PHP, CSV or JSON format.

- Endpoint: http://ip-api.com/
- Docs: https://ip-api.com/docs/

Persistence Service

S3 was used for the persistent view counter. S3 is an object storage service that uses buckets as containers for objects, with each object having a key which is the unique identifier for the object within the bucket.

- S3: https://aws.amazon.com/s3/

Docker was used to storing the image of the server. Using a dockerfile an image of the server was created. These image can then be downloaded and run from dockerhub as a container to run the server. To pull the image use the link: Declanbarrett/conflict-mapper

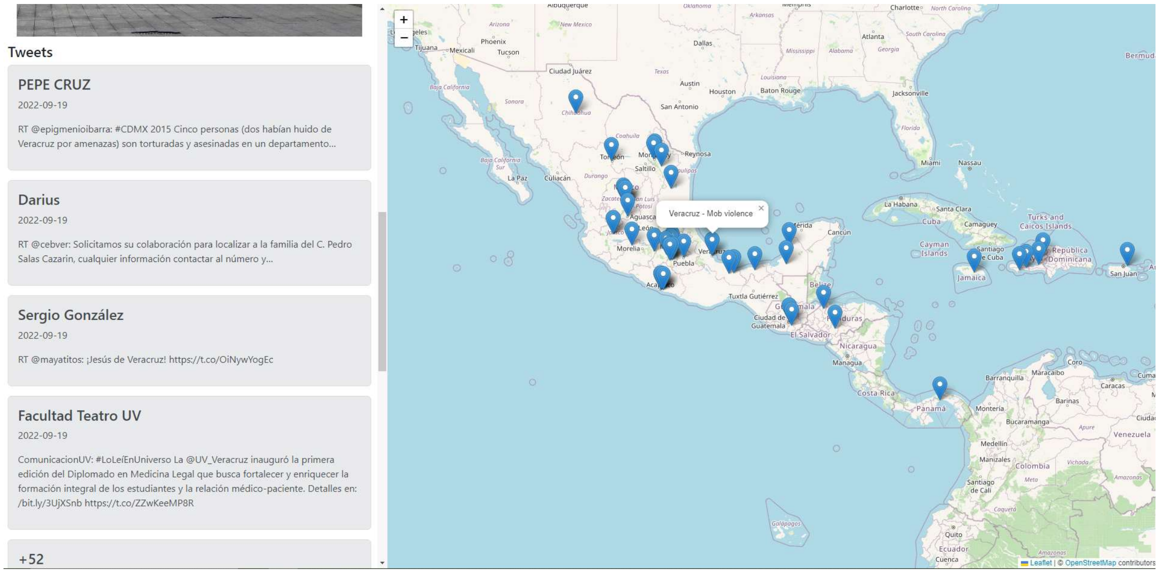
Mashup Use Cases and Services

User Story – Gather Conflict Tweets

As a	Journalist
I want	To find tweets about conflicts in specific geographic locations
So that	I can include the tweets in articles I write

The Leaflet and Open Street Map APIs are called on the client side to create the map that is shown to users. The client-side calls the server side to serve all ACLED conflicts by calling the

ACLED API. The client side uses the list of current conflicts with latitude, longitude and image to create markers on the map. When a marker is selected it then displays the further ACLED information in the left-hand panel as well as calls the server API to call Twitter with the name of the conflict. The Twitter API returns a list of 10 current tweets which are displayed in the left-hand column.



Tweets

PEPE CRUZ
2022-09-19
RT @epigmenioibarra: #CDMX 2015 Cinco personas (dos habían huido de Veracruz por amenazas) son torturadas y asesinadas en un departamento...

Darius
2022-09-19
RT @cebver: Solicitamos su colaboración para localizar a la familia del C. Pedro Salas Cazarin, cualquier información contactar al número y...

Sergio González
2022-09-19
RT @mayatitos: ¡Jesús de Veracruz! https://t.co/OiNywYogEc

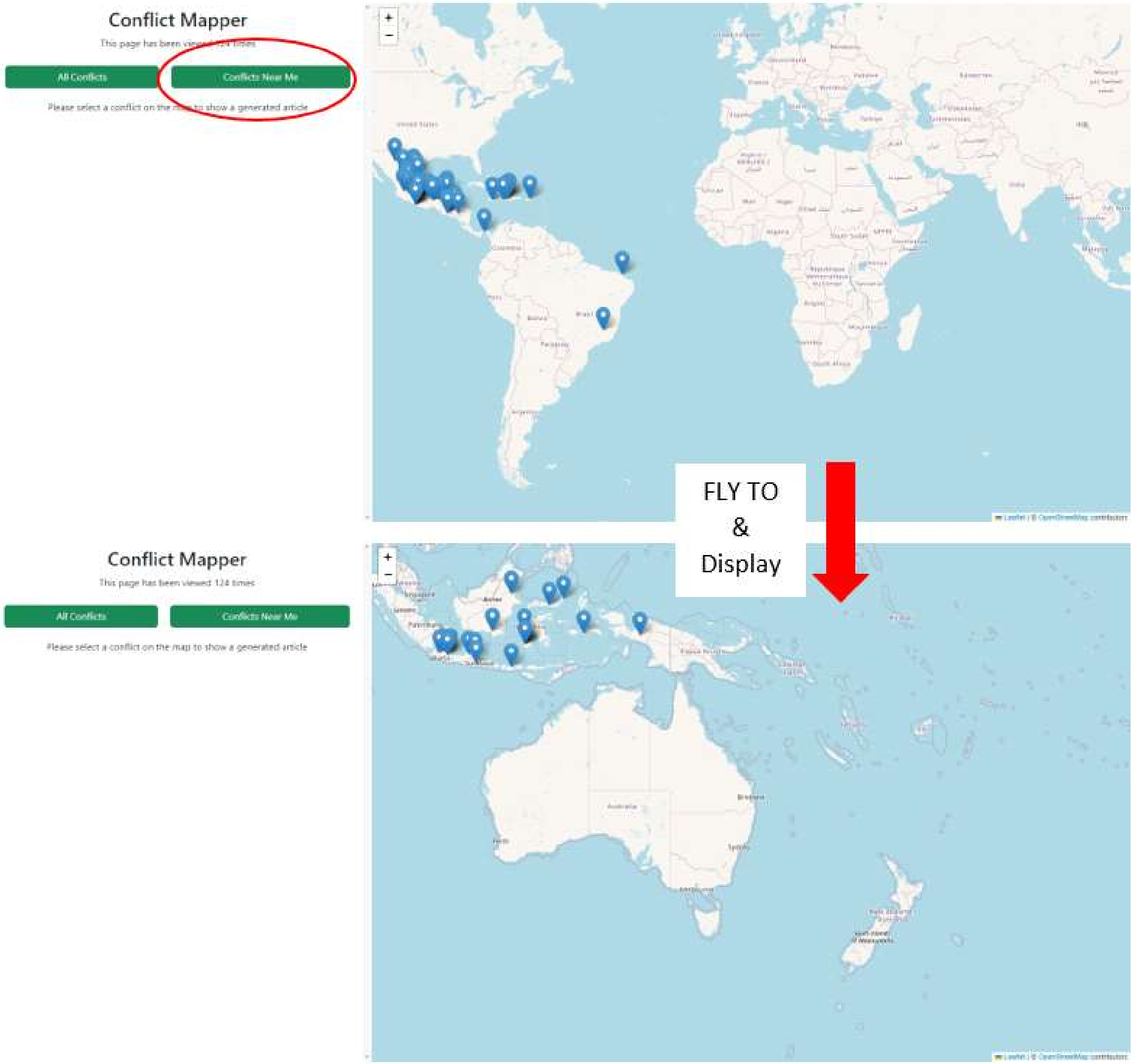
Facultad Teatro UV
2022-09-19
ComunicacionUV: #LoLeíEnUniverso La @UV_Veracruz inauguró la primera edición del Diplomado en Medicina Legal que busca fortalecer y enriquecer la formación integral de los estudiantes y la relación médico-paciente. Detalles en: /bit.ly/3UjXSnb https://t.co/ZZwKeeMP8R

+52

User Story – Gather Nearby Conflicts

As a	Geography Teacher
I want	To find wars that are closest to my students
So that	I can educate them on current international politics in an engaging way

The Leaflet and Open Street Map APIs are called on the client side to create the map that is shown to users. Once the “Near Me” button is pressed the HostIP API is called by the client side to gather the IP address of the user. The IP address is sent to the server which makes the same call except uses the latitude and longitude returned to call the ACLED API and an inaccuracy constant to get the closest conflicts. The ACLED API returns a list of conflicts and their locations. The server receives this and calls the FLICKR API to get a single image for each conflict. These locations and images are sent in response to the client side to be shown using markers on the leaflet API as well as displayed in the left-hand column.



Conflict Mapper

This page has been viewed 124 times

Please select a conflict on the map to show a generated article

FLY TO & Display

Conflict Mapper

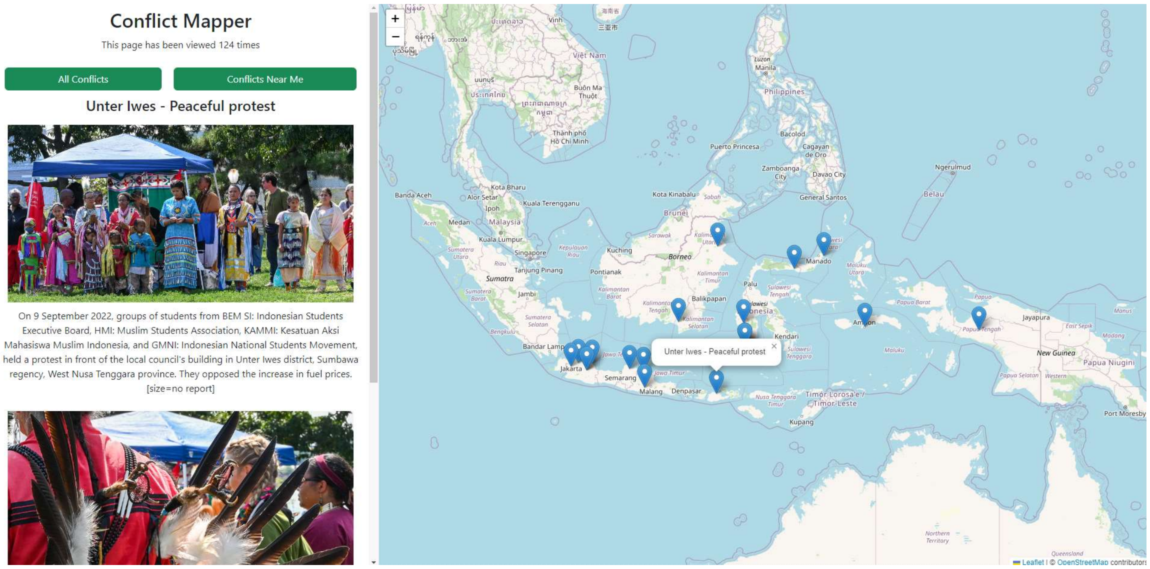
This page has been viewed 124 times

Please select a conflict on the map to show a generated article

User Story – Gather Conflict Images

As an	Expat
I want	To find images of conflicts from the country I moved from
So that	I can sense the level of destruction that has occurred

The Leaflet and Open Street Map APIs are called on the client side to create the map that is shown to users. The client-side calls the server side to serve all ACLED conflicts by calling the ACLED API. The client side uses the list of current conflicts with latitude, longitude and image to create markers on the map. When a marker is selected it then displays the ACLED information in the left-hand panel as well as calls the server to call Flickr with the name of the conflict. The Flickr API returns a list of 2 relevant images which are displayed in the left-hand column, above the Tweets.

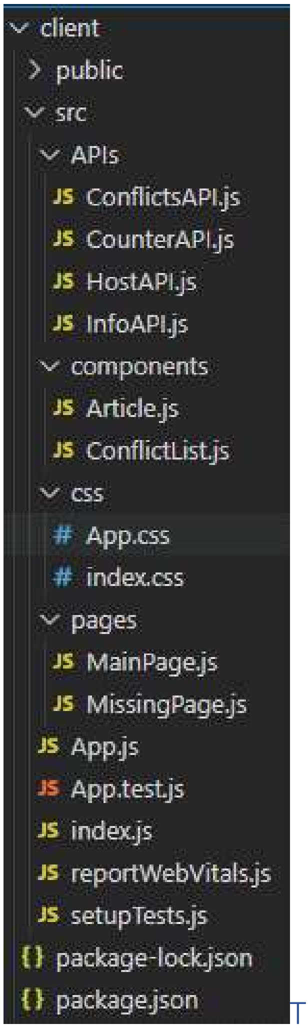


Conflict Mapper

This page has been viewed 124 times

Unter Iwes - Peaceful protest

On 9 September 2022, groups of students from BEM SI: Indonesian Students Executive Board, HMI: Muslim Students Association, KAMMI: Kesatuan Aksi Mahasiswa Muslim Indonesia, and GMNI: Indonesian National Students Movement, held a protest in front of the local council's building in Unter Iwes district, Sumbawa regency, West Nusa Tenggara province. They opposed the increase in fuel prices. [size=no report]

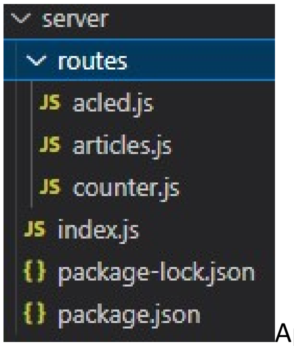


```
client
  public
  src
    APIs
      ConflictsAPI.js
      CounterAPI.js
      HostAPI.js
      InfoAPI.js
    components
      Article.js
      ConflictList.js
    css
      App.css
      index.css
    pages
      MainPage.js
      MissingPage.js
    App.js
    App.test.js
    index.js
    reportWebVitals.js
    setupTests.js
  package-lock.json
  package.json
```

Technical breakdown

Architecture

To begin the project is split into two separate parts: the front end and the back end. The front end uses React JS while the back end uses Express. Express serves the React front end build via Express Static allowing the use of React Router on the front end. There are four different API calls for the client side, located in the API directory. Conflicts is to gather the conflicts from the server side, counter to gather the view counter from the server side, info to gather the articles from the server side and host to gather the IP address of the client. Two components have been separated from the page, “Article” and “ConflictList”. These were separated to reduce the clutter on the “MainPage”. “MainPage” is the main page of the application which contains the map, title of the applications and buttons at the top. “Article” displays the information regarding a conflict on the left and “ConflictList” displays the map markers on the right.



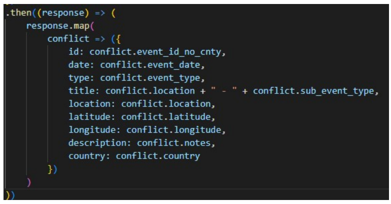
```
server
  routes
    acled.js
    articles.js
    counter.js
  index.js
  package-lock.json
  package.json
```

App contains the ReactRouter which routes between “MissingPage” and “MainPage”. Missing page is a very empty page stating that the application is located at “/Home” with a link to go there. This was simply to stop the “MainPage” being returned on broken API calls thus to signify that something has indeed gone wrong.

On the express server side there are three routers, “acled’, “articles” and “counter”. “Index” is used as the main entrance to the express app and uses these routers for the functionality of the app: “acled” handles “/api/acled/:number” and “/api/acled/:ipaddress/:number” to gather conflicts, “articles” handles “/api/articles/:location/:type/:country” to gather articles and “counter” handles “/api/counter/add” to update and gather the view counter.

Data Flow

Upon server start-up, an S3 bucket is created if it isn’t already to later use as a view counter store. At / is a very basic “Missing Page” 404 page with a link the user can use to go to the /home “Main Page” page where the main application resides.



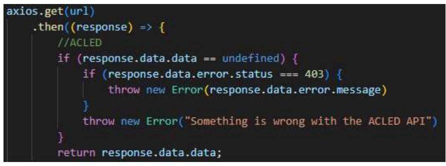
```
.then((response) => (
    response.map(
        conflict => ({
            id: conflict.event_id_no_cnty,
            date: conflict.event_date,
            type: conflict.event_type,
            title: conflict.location + " - " + conflict.sub_event_type,
            location: conflict.location,
            latitude: conflict.latitude,
            longitude: conflict.longitude,
            description: conflict.notes,
            country: conflict.country
        })
    )
))
```

Upon entering the main page multiple API calls are immediately made separately to one another. The ‘counter’ express API is called which when received by express at /api/counter/add and immediately attempts to retrieve the S3 bucket and get the “viewcounter” object. If this object is returned then it grabs the count, adds 1, pushes the new view count to S3 and returns the view count to the client. If the view counter object is missing then it creates a new one starting at 1, pushes the new view count to S3 and returns the view count of 1 to the client.

The client requesting conflicts via express at “/api/acled/” with no IP address listed also occurs immediately. This requests the ACLED API for the first 50 conflicts to display, with an array of conflicts returned being slightly modified and returned to the client. All API requests use Axios.

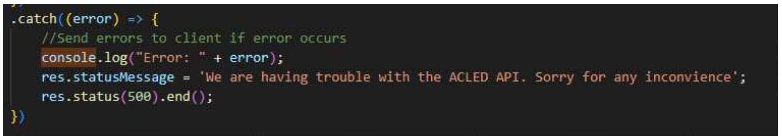
Leaflet and React-Leaflet are in the node modules so the leaflet API isn’t explicitly called but it is still gathered from the leaflet API on start-up. The tiles of OpenStreetMap are called by leaflet to display as the map moves around. Once the ACLED conflicts arrive at the client the locations are used to show markers on the map.

If the “conflicts near me” button is pressed then the IP API is called directly from the client. This doesn’t use a query cause the IP API gets the IP of the caller. This IP is then used in a call to the conflicts via express at “/api/acled/” which makes a separate call to the IP API using the IP to gather the location of the client. After this, the “/api/acled/” route collects all ACLED API conflicts (max 500) and then the distance is calculated between the location of the conflict and the client on the server with the closest conflicts then returned to the client and displayed on the map. If the “all conflicts” button is pressed then the IP address is deleted and the “/api/acled/” route is called without the IP address.

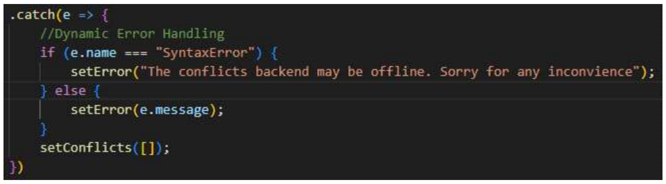


```
axios.get(url)
    .then((response) => {
        //ACLED
        if (response.data.data == undefined) {
            if (response.data.error.status === 403) {
                throw new Error(response.data.error.message)
            }
            throw new Error("Something is wrong with the ACLED API")
        }
        return response.data.data;
```

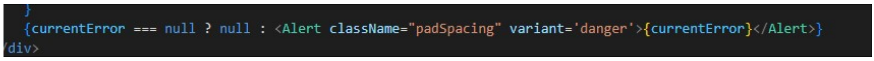
When a marker on the map is pressed the “/api/articles/” route is called with the location, type and country of the conflict selected, gathered from the returned ACLED object from the ACLED API call, thus forming the “mash-up”. The articles route calls both the Twitter and Flickr APIs. First, the Twitter API is called with the location and type combined. If this doesn’t return any tweets then just the location is used to find tweets. After this, then the Flickr API is called with the location and if no images are returned then the type is used. Both the tweets and images are then returned to the client and the tweets, images and existing ACLED conflict information are displayed to form an article in the menu on the left-hand side.



```
.catch((error) => {
    //Send errors to client if error occurs
    console.log("Error: " + error);
    res.statusMessage = 'We are having trouble with the ACLED API. Sorry for any inconvience';
    res.status(500).end();
})
```



```
.catch(e => {
    //Dynamic Error Handling
    if (e.name === "SyntaxError") {
        setError("The conflicts backend may be offline. Sorry for any inconvience");
    } else {
        setError(e.message);
    }
    setConflicts([]);
})
```



```
  }
  {currentError === null ? null : <Alert className="padSpacing" variant='danger'>{currentError}</Alert>}
/div>
```

Errors that occur when requesting the express API are set usually modified before displaying that something has gone wrong to the user. Custom errors on the server side such as both tweets and images being empty set the status message in the response header to enable the specific error to be displayed by the client. Such an example is the error with the ACLED API gathering of regular conflicts which is simply printed out in the server log and a sanitized version returned to the client

Deployment and the Use of Docker

Deployment was able to be done via Docker. The docker base image used is node (Figure A). Originally a Dockerfile with ubuntu as the base image that installed node and npm during the docker build but this took longer than simply using a node image base. Then “apt-get update && apt-get install -y” is used to update all the existing packages in the image. Both the client and server are copied into the image. To reduce build time and make sure only the correct

dependencies are included, the node_modules and react build are deleted prior to creating the image. The working directory is set to the client to then install the node_modules and create a build to serve. Afterwards, the working directory is set to server to install the server-side node modules. Port 3000 is exposed since it's a node application. When the image is used to make a container the index.js file is used to run the application.
DockerHub was used as the repository to store the docker image. The docker image is located at declanbarrett/conflict-mapper but is set to private to prevent plagiarism.

A custom run command was created which is used to set the AWS and API keys which is located in figure 3.

Test plan

Manual testing was performed. The application is designed that the amount of user input is quite limited. Due to the API routes being REST and being exposed it can be queried directly to test both client and server side.

Type	Task	Expected Outcome	Result	Screenshot/s (Figure B)
Positive	Update View Counter	Updated counter displayed, S3 updated with new value	Pass	1
	Click All Conflicts Button	Conflicts displayed as markers on map	Pass	2
	Click Conflicts Near Me Button	Closest ACLED conflicts displayed as markers on map nearby to the user	Pass	3
	Click Conflict Marker On Map	Article displayed showing information, images and tweets	Pass	4
	Have Conflicts Load by Default	Current conflicts load by default when the page is loaded	Pass	5
Negative	Handle No Tweets & Images	Show "It is not possible to generate an article for this conflict" error	Pass	6
	Handle Server Side Offline	Show "The conflicts backend may be offline" error in red	Pass	7
	Handle IP-API offline	Show "Please turn off your ad-blocker" error in red	Pass	8
	Handle AWS credentials wrong	Show AWS error on server log and show "unknown" as the view count	Pass	9
	Handle Flickr credentials wrong	Show "our article generator is having trouble" as error and print actual error to server	Pass	10
	Handle Twitter credentials wrong	Show "our article generator is having trouble" as error and print actual error to server	Pass	11
	Handle ACLED credentials wrong	Show "ACLED is having trouble" as error and print actual error to server	Pass	12
	Handle / Being Called	Show the missing page screen	Pass	13
Edge	Handle No Tweets but images	Show only images with no tweet title	Pass	14

	Handle No Images but tweets	Show only tweets	Pass	15
	Handle multi-word queries	Convert call to HTML encoding, retrieve results	Pass	16
	Handle rapid clicking	Finish call list with correct state	Pass	17
NF	Gather ACLED within 7 seconds	A time below 7000ms displayed in the outcome	Pass	18
	Gather Article within 5 seconds	A time below 5000ms displayed in the outcome	Pass	19
	Gather view counter within 1 second	A time below 1000ms displayed in the outcome	Pass	20

Difficulties / Exclusions / unresolved & persistent errors /

At the time of writing everything specified in the assignment has been included and is functional. The main problems that persist with the application are content relevancy.

There were a lot of minor errors that were handled throughout the development of the conflict mapper. To begin with it was difficult to get my hands on an ACLED API key as they requested I make a formal enquiry which needed to be reviewed. Thankfully they thought the usage of the API was acceptable.

ACLED is only able to serve 500 lines of objects to me as limited by my level of API access and also changes the content every week. This results in some weeks where the “closest” conflict is in Africa and some where it is in New Zealand, making it appear on some queries that the button doesn’t work, but when scrolling out find out it does.

Most tweets from foreign countries are unsurprisingly in another language making it difficult to tell what is being said without translation. Flickr also provides images which are not always relevant. Since only two images are retrieved there might be a lot of really good images available that are simply not shown since pictures of cats or crystals share similar names and thus are presented. There are also foreign place names which sometime return ‘lewd’ imagery from flickr which may not be appropriate to include in classrooms.

The choice to use leaflet in combination with React was difficult as they do not play well together, thus React-Leaflet was used to mitigate this. This prompted other problems since the documentation for React-Leaflet is scarce. Implementing the markers displayed on the map was relatively easy but having the map fly to them was difficult.

The original “host ip” API that I was using too often returned an error as a result very early in development thus the switch was made to “IP API”. A lot of time was spent dealing with error handling at the last minute when the test plan needed to be completed, combined with job assessments and other assignments causing a level of stress that caused me to have a mental breakdown and be unable to continue work.

Extensions

To make the application better the conflicts should probably be moved to a Redis or S3 Cache. The ACLED API is incredibly slow even when requesting small quantities of queries. The conflicts near me also collects all 500 available conflicts and compares them manually making this call extremely slow.

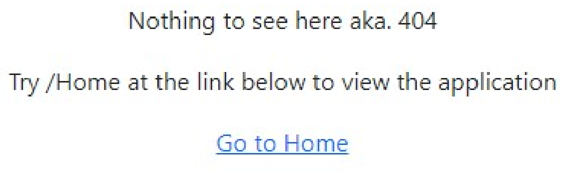
Further investigation into optimizing Flickr and Tweet relevancy needs to be done since a lot of articles retrieved could not be used by journalists. Having a filter which is either using a black or white list to remove results which are not appropriate could be used, but this would increase response times.

The main upgrade that needs to occur is the use of a translation API for the tweets as serving tweets in a foreign language makes it basically impossible for students or other end users to engage and care about those being affected by the conflict since they cannot understand what is being said.

User guide

To start the server run the docker run command in figure 3 with the correct AWS and API keys. The server should automatically run in the container once it has been created.

The application should be opened to /Home on port 80 (HTTP) as this is where the main application is. Otherwise, the missing page screen will be displayed and the link to the application should be followed.

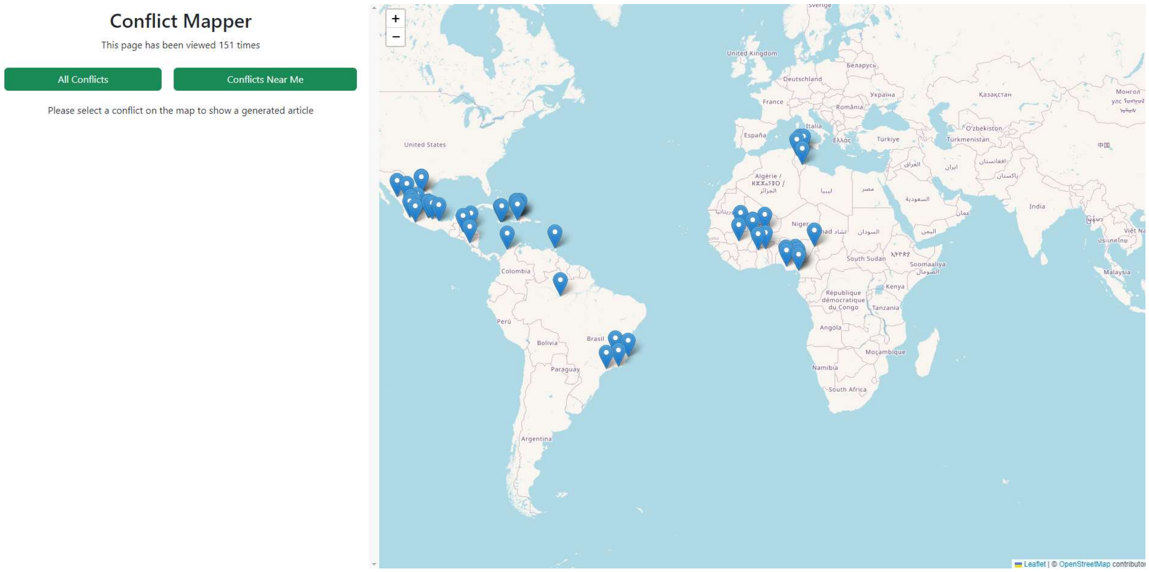


Nothing to see here aka. 404

Try /Home at the link below to view the application

Go to Home

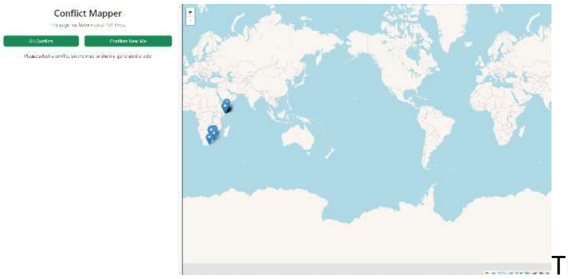
The application will open to the home page



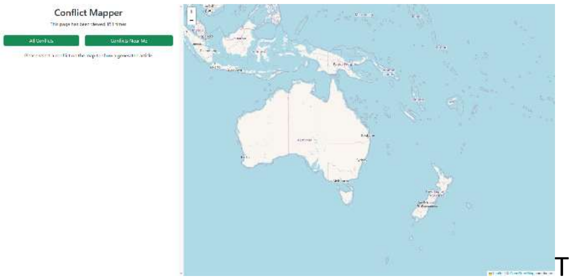
Conflict Mapper

This page has been viewed 151 times

Please select a conflict on the map to show a generated article

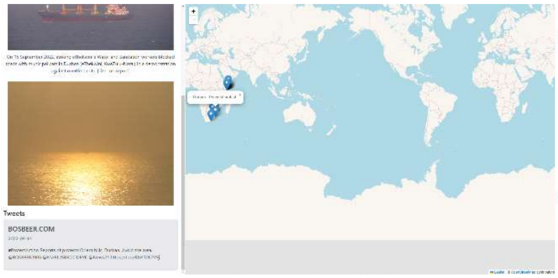


The entire page can be navigated with the mouse. Click and drag to move around the map and use the scroll wheel or the + and – symbols to zoom in and out. The page counter is displayed underneath the title. By default, the page is loaded with the first 50 conflicts which is being labelled as 'all'. Clicking the 'conflicts near me' button will load the conflicts nearest to the user and send the user to their location. Sometimes the zoom in is too much but the closest conflicts are still loaded once zoomed out. In this case, the closest conflicts this week to Australia were in Africa. This changes frequently.



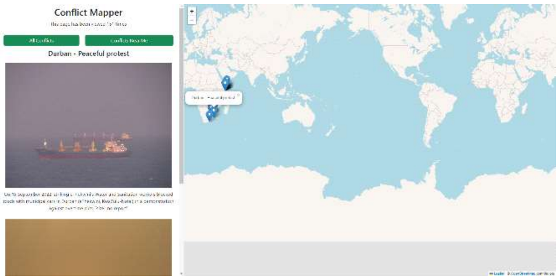
To display 'all conflicts' again the all conflicts button can be pressed.

Each of the blue icons on the map can be selected to generate an article as is stated on the left-hand side. This left-side menu is scrollable. If the pictures extend beyond the page, then the menu can be scrolled down to see the tweets. This makes up the entirety of the application.



Tweets

BOSBEER.COM



Conflict Mapper

Durban - Peaceful protest

Analysis

Question 1: Vendor Lock-in

The replacement of the APIs used would take a significant rework but would be able to be done.

To replace OpenStreepMap and Leaflet would not be too difficult as there are a lot of map API providers. Google Maps API, Mapbox API, Fencer, Foursquare API are all examples of APIs that have basic free tiers that could be used to replace the map used. However, if the load on the service increases then these would begin with cost money (RapidAPI, 2022a). The Google Maps API is the API that the most similar service (UCDP) provides to show their conflict information which was revealed monitoring the network traffic (UCDP, 2022a). A completely free alternative to leaflet is OpenLayers which provides even more functionalities than Leaflet in terms of map customization and layering (OpenLayers, 2022). The same latitude and longitude data of conflicts could be used with these options.

IP API was already switched out from Host IP midway through the development of the application and is free (HostIp, 2022). However, it produced substandard results thus a better alternative along the lines ipinfo or geoip-db could be used for IP lookup. Instead of violating the clients privacy without their permission, the HTML5 geolocation API can be used which asks the client whether they want their location shared. This is what the majority of other applications do to ethically gather the user's location information, is free and will be a user experience that a lot of users are used to. However, as of Chrome 50, this will only work with HTTPS applications thus the site would need to be updated (Amadi, 2022).

The ACLED API is the most difficult to replace as there are very few APIs that cover conflicts and global politics. The only other similar API is the UCDP API which is free (UCDP, 2022b). This API provides similar data such as the coordinates, location, actors, types of violence and dates. This information would be enough to be used in the API calls to gather the images and tweets, as well as enable the markers to be displayed on the map. However, this API has little documentation and also does not produce a summary of the conflicts, meaning this information would be missing from the article and need to be sourced elsewhere.

The Flickr and Twitter APIs are both "social media" APIs with Flickr focused on images and Twitter focused on text-based content (RapidAPI, 2022b). Both of these could be swapped out with any other social media API to share insights on the conflicts that are being gathered. Facebook has both Graph API and Pages API which can gather public posts and pages based on a search query. Within graph, images can also be gathered allowing the articles to be completely replaced with the use of Facebook APIs (Facebook, 2022). Alongside Facebook, Instagram also has a Graph API of its own able to search for public posts and gather the information and images pertained within. Alternatively, and more informatively, the YouTube API could be used to search for relevant informational videos on the conflicts and provide these videos to watch in the left-hand column (YouTube, 2022). This would be more focused on the education use case rather than generating articles for journalists to use. All of the above options are free and would not impact the user experience greatly.

Changing the persistence service from S3 would not be too difficult. There is a range of companies that provide similar object storage solutions. The main providers such as Google and Microsoft both have their blob storage equivalents such as Microsoft Azure Blob and Google Cloud Platform Cloud Storage. GCP has a free tier similar to AWS while Azure is pay-as-you-go immediately. The code to create a bucket in GCP is quite similar to AWS with 'createBucket' and 'file' and 'save' used to create a bucket and save a file with contents. Even setting the credentials is similar with Google needing the "GOOGLE_APPLICATION_CREDENTIALS" set with the correct key (Google Cloud, 2022). There are also a lot of smaller providers that offer S3 clones.

However, using blob storage for an object which is less than a kb is a bit overkill thus a memory cache view counter updating the value in a key-value pair could also be suitable, be quicker and act in a very similar way, if it was persistent. Redis with Elastic-cache could be suitable if Append-Only-Files is turned on which re-runs all writes on the cache data, but this isn't available on newer versions of Redis nor does it protect against all failure scenarios. Unsurprisingly using a non-persistent data store to attempt to store persistent data isn't a great idea and shouldn't be attempted (Amazon, 2022a)

Lastly, there are the database options, both SQL and No-SQL. Using SQL would require the creation of a database and table with a column name of "counter" and a single row with the current count. This would be with AWS RDS, Azure SQL Database or GCP Cloud SQL with an RDBM such as MySQL running. As relational databases have difficulty scaling out, it would be more appropriate to use a No-SQL database. DynamoDB, MongoDB Atlas, Azure Cosmos DB or Google Cloud Firestore. These offer a key-value solution similar to Redis with the persistence needed for a view counter (Amazon, 2022b). As schema is determined per item and the object being stored in S3 is already JSON with 'view-counter' being the key, there would not be a change in approach other than the specific functions required to create a database, get and add the key and value.

Question 2: Container Deployment

Under the assumption that a service to manager container deployment is being used for a more substantial version of the application, there are very few downsides to using docker in comparison to VMs.

The downside of using docker in this scenario is that it is already running on a virtual machine meaning that docker and each container are adding overhead that would not normally be there if the application was running directly on the virtual machine. Having multiple containers run on the same virtual machine also means that port collisions become a problem as multiple containers could be attempting to serve on the same outward-facing port. When 'docker run' is used to create the container a port mapping is specified, and it appears docker only allows 1 to 1 port mapping. With a service to manage container deployment, it may be possible to have each of these containers given a separate port or separate IP address and this service map port 80 HTTP requests to multiple containers. One of these services such as the Nginx gateway, a reverse proxy that allows for requests to be diverted to multiple containers (S, 2022). There is also a security risk with using docker as it has process-level isolation which very slightly less secure than the full virtualization that a VM offers. The application was built in WSL Ubuntu thus it is indeed compatible with Linux.

The benefit to using docker rather than the virtual machine for this application is that even when ‘substantial’ it is still not a resource-intensive application and thus benefits from a large quantity of smaller servers. A lot of the client-to-server calls are only waiting due to the external APIs being slow or requested multiple times to guarantee a response. Having only one application running per EC2 instance would be a waste of the EC2 instance’s CPU and main memory since they could be more efficiently used by Docker. Each container is 1.5GB due to the node modules taking up most of that space with the standard 8GB of space on the EC2 micro meaning at least four containers could be running simultaneously. Docker can support many more containers than VMs as it doesn’t need to maintain an entire OS stack, with attempting to run a VM on a VM not being a great idea. Running the application multiple times on the same EC2 instance is possible but creates security risks due to there not being a proper separation between application instances. Rebooting is also quicker with docker (Clancy, 2022). If the application became incredibly large and resource intensive from expansion, then at that point running one docker container rather than running directly on the VM is a waste and the VM would be used directly.

With an expanded application this would need to remain stateless and thus a multitude of storage solutions would be needed to improve the user experience. An in-memory cache such as Elastic-cache and Redis would be used for the most frequent API calls, which as it currently stands would be the 50 ACLED conflicts since these are called every time the application loads (Amazon, 2022c). For the slightly less used API calls the S3 datastore could be used to be another ‘cache’ for the lists of conflicts which are larger or the most requested Flickr images and tweets, especially for conflicts that have been gaining a lot of media attention. To determine which information to cache and where to cache, a unique request counter could be added which uses either object or NoSQL database which could be checked against frequently to determine what is the most used.

To serve the client-side React application, a CDN could be used since all of this content is static and is currently being served through express static. This will improve the page load times and reduce demand for the software containers. If static images were added to the site they would be included here, since the only static images currently are the tab icons (Amazon, 2022d).

To support the addition of user accounts as well as user sessions and other features that would be added, the view counter and all of the data related to these features would be included in a DynamoDB database. Even an upgraded version of the application would not need ACID since it is not incredibly important that passwords and other new data are updated immediately, but is important that they are eventually updated. The main reason to use NoSQL is the scalability that SQL databases either cannot provide or have performance impacts due to their difficulty with sharding (ACG Team, 2022). DynamoDB will be able to provide availability and partition tolerance while sacrificing consistency which is important when considering scaling out to a global context. DynamoDB also has atomic operations to prevent overwrite problems. Using a single major provider will also improve performance slightly as each major provider invests heavily in improving the latency between their internal server farms.

References

ACG Team. (2022). ACID and AWS DynamoDB transactions: The ins and outs of all-or-nothing. A Cloud Guru. Retrieved 20 September 2022, from https://acloudguru.com/blog/engineering/acid-aws-dynamodb-transactions.

Amadi, L. (2022). Getting Users locations, The Easy Way For Developers.. Medium. Retrieved 20 September 2022, from https://medium.com/@w3bh4ck/different-ways-to-get-a-users-location-bef34f732fec.

Amazon. (2022a). Append only files (AOF) in ElastiCache for Redis. Retrieved 20 September 2022, from https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/RedisAOF.html.

Amazon. (2022b). What Is a Key-Value Database?. Amazon Web Services, Inc. Retrieved 20 September 2022, from https://aws.amazon.com/nosql/key-value/.

Amazon. (2022c). Elastic-Cache Redit. Retrieved 20 September 2022, from https://aws.amazon.com/elasticache/redis/

Amazon. (2022d). Cloudfront. Retrieved 20 September 2022, from https://aws.amazon.com/cloudfront/

Clancy, M. (2022). Docker Containers vs. VMs: A Look at the Pros and Cons. Backblaze Blog | Cloud Storage & Cloud Backup. Retrieved 20 September 2022, from https://www.backblaze.com/blog/vm-vs-containers/.

Facebook. (2022). Overview - Graph API - Documentation - Facebook for Developers. Developers.facebook.com. Retrieved 20 September 2022, from https://developers.facebook.com/docs/graph-api/overview.

Google Cloud. (2022). Create storage buckets | Cloud Storage | Google Cloud. Google Cloud. Retrieved 20 September 2022, from https://cloud.google.com/storage/docs/creating-buckets#storage-create-bucket-nodejs.

HostIp. (2022). Lookup IP Address Location With Hostip.info. Hostip.info. Retrieved 20 September 2022, from https://www.hostip.info/use.php.

OpenLayers. (2022). OpenLayers - Welcome. Openlayers.org. Retrieved 20 September 2022, from https://openlayers.org/.

RapidAPI. (2022a). The Top 10 Mapping & Map APIs (for Developers in 2018) | RapidAPI. The Last Call - RapidAPI Blog. Retrieved 20 September 2022, from https://rapidapi.com/blog/top-map-apis/#:~:text=%EF%BB%BF-.

RapidAPI. (2022b). 18 Top Social Media APIs & Free Alternatives List - September, 2022 | RapidAPI. Rapidapi.com. Retrieved 20 September 2022, from https://rapidapi.com/collection/social-media-apis.

S, V. (2022). Want multiple Docker containers on the same port? Here's how. Bobcares. Retrieved 20 September 2022, from https://bobcares.com/blog/docker-multiple-containers-same-port/.

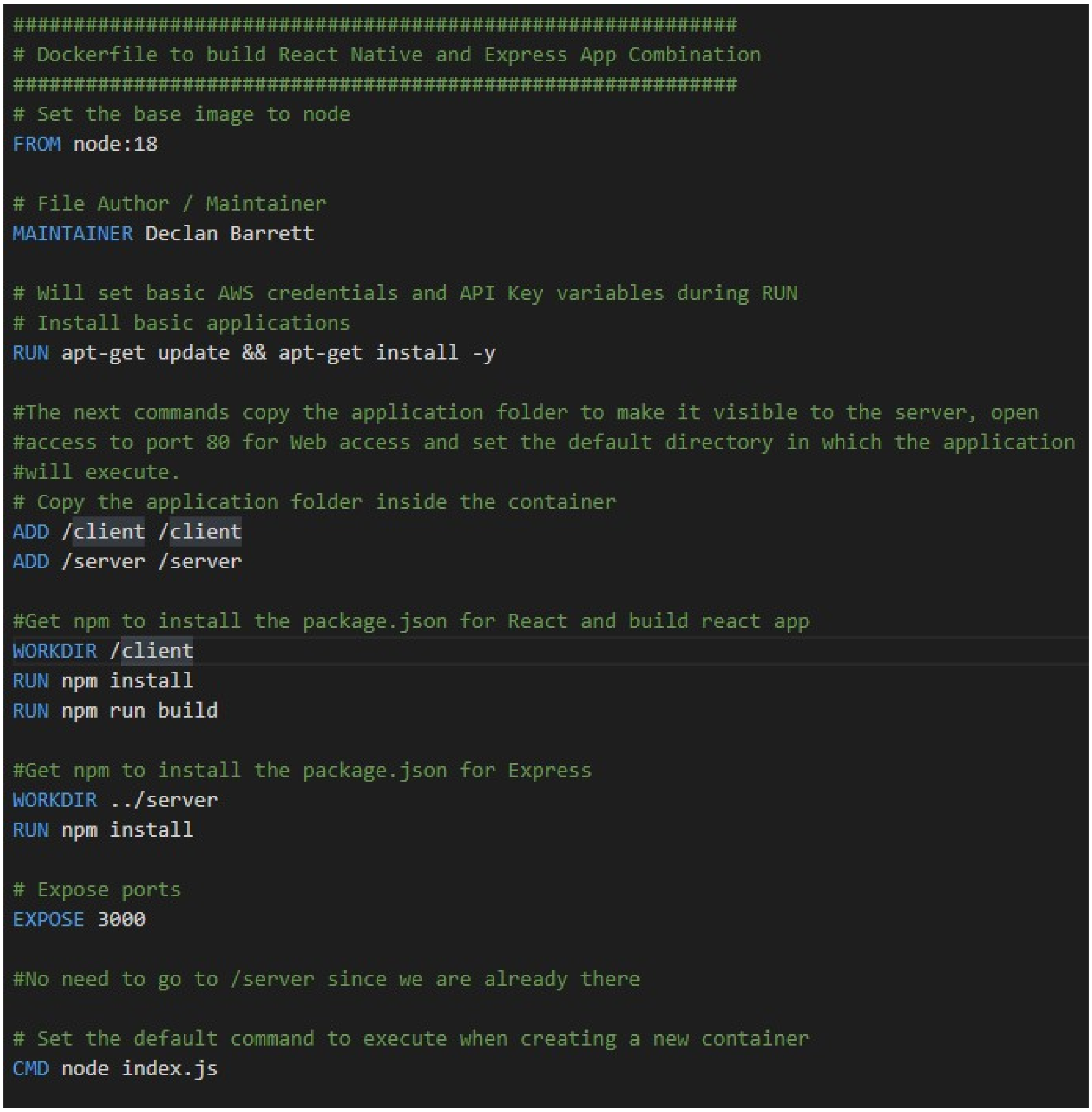
UCDP. (2022a). UCDP - Uppsala Conflict Data Program. Ucdp.uu.se. Retrieved 20 September 2022, from https://ucdp.uu.se/country/365.

UCDP. (2022b). UCDP Application Programming Interface (API). Ucdp.uu.se. Retrieved 20 September 2022, from https://ucdp.uu.se/apidocs/.

YouTube. (2022). YouTube Data API | Google Developers. Google Developers. Retrieved 20 September 2022, from https://developers.google.com/youtube/v3. Appendices

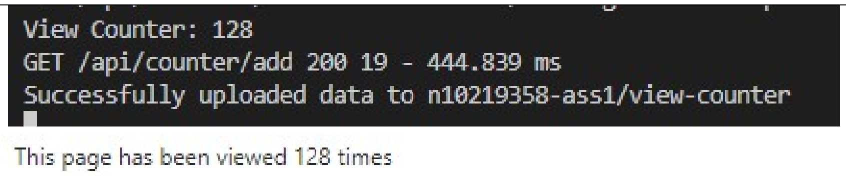
Appendix

Figure A: Dockerfile

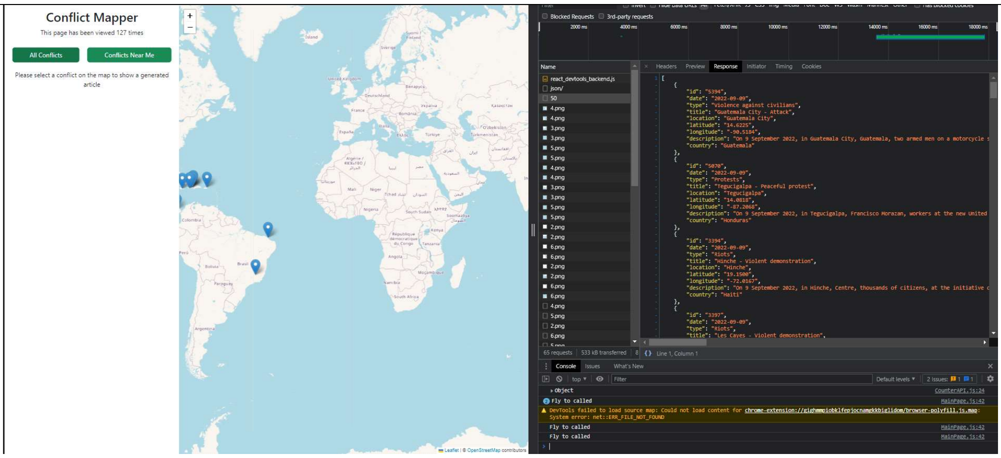


```
############################################################
# Dockerfile to build React Native and Express App Combination
############################################################
# Set the base image to node
FROM node:18

# File Author / Maintainer
MAINTAINER Declan Barrett

# Will set basic AWS credentials and API Key variables during RUN
# Install basic applications
RUN apt-get update && apt-get install -y

#The next commands copy the application folder to make it visible to the server, open
#access to port 80 for Web access and set the default directory in which the application
#will execute.
# Copy the application folder inside the container
ADD /client /client
ADD /server /server

#Get npm to install the package.json for React and build react app
WORKDIR /client
RUN npm install
RUN npm run build

#Get npm to install the package.json for Express
WORKDIR ../server
RUN npm install

# Expose ports
EXPOSE 3000

#No need to go to /server since we are already there

# Set the default command to execute when creating a new container
CMD node index.js
```

Figure B:

1	 View Counter: 128 GET /api/counter/add 200 19 - 444.839 ms Successfully uploaded data to n10219358-ass1/view-counter This page has been viewed 128 times

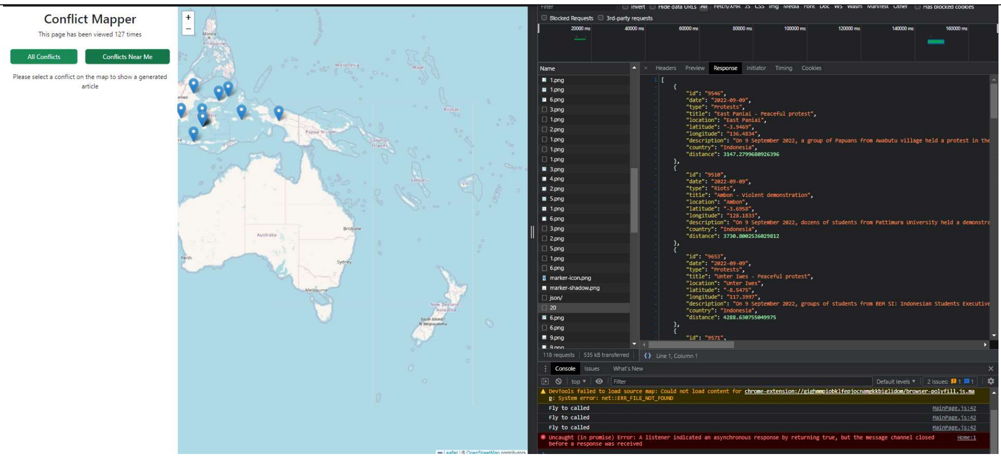
2



```
[
    {
        "id": "5394",
        "date": "2022-09-09",
        "type": "Violence against civilians",
        "title": "Guatemala City - Attack",
        "location": "Guatemala City",
        "latitude": "14.6225",
        "longitude": "-90.5184",
        "description": "On 9 September 2022, in Guatemala City, Guatemala, two armed men on a motorcycle s
        "country": "Guatemala"
    },
    {
        "id": "5070",
        "date": "2022-09-09",
        "type": "Protests",
        "title": "Tegucigalpa - Peaceful protest",
        "location": "Tegucigalpa",
        "latitude": "14.0818",
        "longitude": "-87.2068",
        "description": "On 9 September 2022, in Tegucigalpa, Francisco Morazan, workers at the new United
        "country": "Honduras"
    },
    {
        "id": "3394",
        "date": "2022-09-09",
        "type": "Riots",
        "title": "Hinche - Violent demonstration",
        "location": "Hinche",
        "latitude": "19.1500",
        "longitude": "-72.0167",
        "description": "On 9 September 2022, in Hinche, Centre, thousands of citizens, at the initiative c
        "country": "Haiti"
    },
    {
        "id": "3397",
        "date": "2022-09-09",
        "type": "Riots",
        "title": "Les Cayes - Violent demonstration",
```

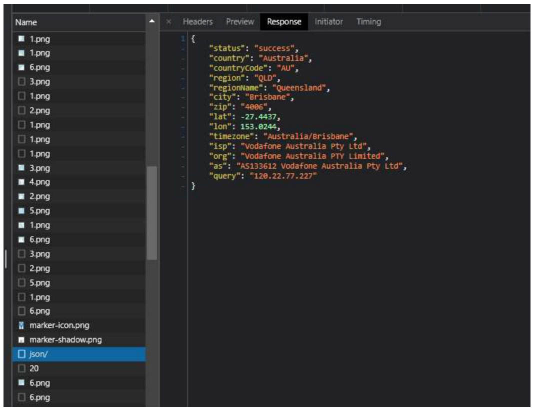
```
▸ Object                                                                CounterAPI.js:24
2 Fly to called                                                         MainPage.js:42
⚠ DevTools failed to load source map: Could not load content for chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/browser-polyfill.js.map:
  System error: net::ERR_FILE_NOT_FOUND
  Fly to called                                                         MainPage.js:42
  Fly to called                                                         MainPage.js:42
```

3

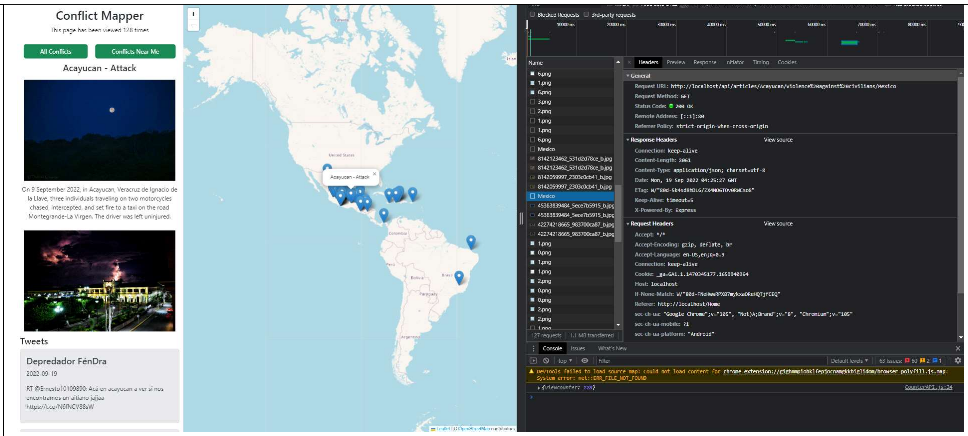
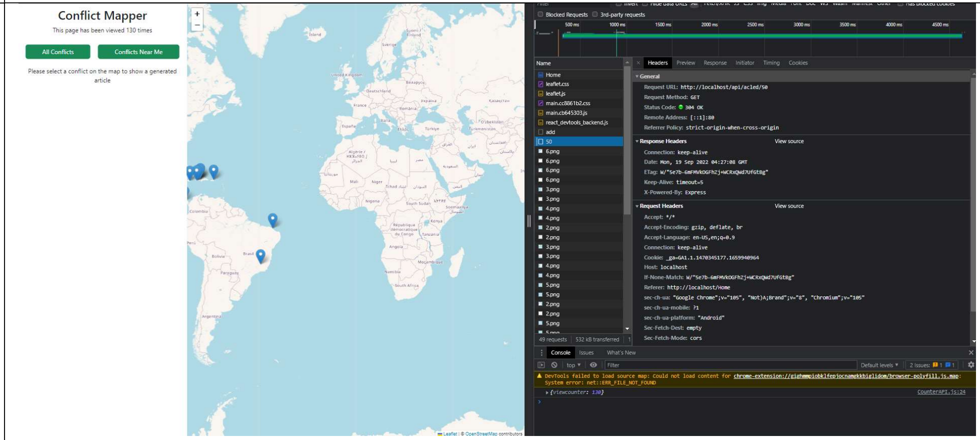
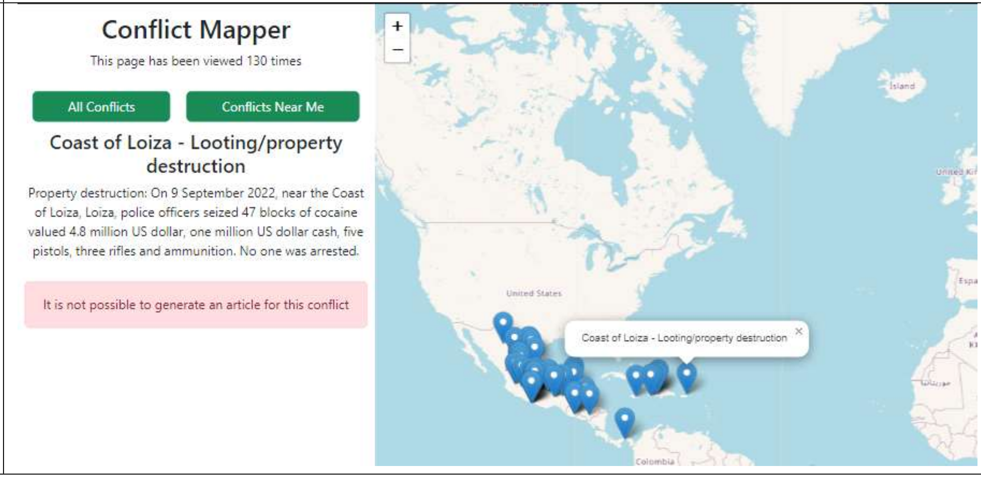


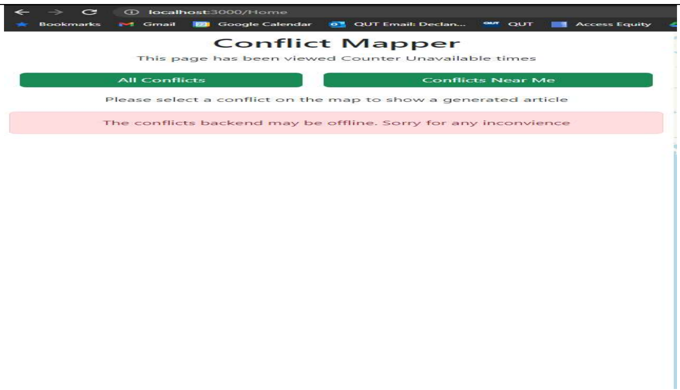
```
[
    {
        "id": "9546",
        "date": "2022-09-09",
        "type": "Protests",
        "title": "East Paniai - Peaceful protest",
        "location": "East Paniai",
        "latitude": "-3.9469",
        "longitude": "136.4834",
        "description": "On 9 September 2022, a group of Papuans from Awabutu village held a protest in the
        "country": "Indonesia",
        "distance": 3147.2799680926396
    },
    {
        "id": "9510",
        "date": "2022-09-09",
        "type": "Riots",
        "title": "Ambon - Violent demonstration",
        "location": "Ambon",
        "latitude": "-3.6958",
        "longitude": "128.1833",
        "description": "On 9 September 2022, dozens of students from Pattimura University held a demonstra
        "country": "Indonesia",
        "distance": 3730.8002536029812
    },
    {
        "id": "9653",
        "date": "2022-09-09",
        "type": "Protests",
        "title": "Unter Iwes - Peaceful protest",
        "location": "Unter Iwes",
        "latitude": "-8.5475",
        "longitude": "117.3997",
        "description": "On 9 September 2022, groups of students from BEM SI: Indonesian Students Executive
        "country": "Indonesia",
        "distance": 4288.630755049975
    },
    {
        "id": "9571",
```

```
⚠ DevTools failed to load source map: Could not load content for chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/browser-polyfill.js.ma
  p: System error: net::ERR_FILE_NOT_FOUND
  Fly to called                                                         MainPage.js:42
  Fly to called                                                         MainPage.js:42
  Fly to called                                                         MainPage.js:42
⊗ Uncaught (in promise) Error: A listener indicated an asynchronous response by returning true, but the message channel closed   Home:1
  before a response was received
```



```
{
    "status": "success",
    "country": "Australia",
    "countryCode": "AU",
    "region": "QLD",
    "regionName": "Queensland",
    "city": "Brisbane",
    "zip": "4006",
    "lat": -27.4437,
    "lon": 153.0244,
    "timezone": "Australia/Brisbane",
    "isp": "Vodafone Australia Pty Ltd",
    "org": "Vodafone Australia PTY Limited",
    "as": "AS133612 Vodafone Australia Pty Ltd",
    "query": "120.22.77.227"
}
```

4	 Conflict Mapper This page has been viewed 128 times All Conflicts Conflicts Near Me Acayucan - Attack On 9 September 2022, in Acayucan, Veracruz de Ignacio de la Llave, three individuals traveling on two motorcycles chased, intercepted, and set fire to a taxi on the road Montegrande-La Virgen. The driver was left uninjured. Tweets Depredador FénDra 2022-09-19 RT @Ernesto10109890: Acá en acayucan a ver si nos encontramos un aitiano jajjaa https://t.co/N6fNCV88sW Acayucan - Attack Leaflet | © OpenStreetMap contributors Blocked Requests 3rd-party requests 10000 ms 20000 ms 30000 ms 40000 ms 50000 ms 60000 ms 70000 ms 80000 ms Headers Preview Response Initiator Timing Cookies Name 6.png 1.png 6.png 3.png 2.png 1.png 1.png 6.png Mexico 8142123462_531d2d78ce_b.jpg 8142123462_531d2d78ce_b.jpg 8142059997_2303c0cb41_b.jpg 8142059997_2303c0cb41_b.jpg Mexico 45383839484_5ece7b5915_b.jpg 45383839484_5ece7b5915_b.jpg 42274218665_983700ca87_b.jpg 42274218665_983700ca87_b.jpg 1.png 0.png 1.png 1.png 2.png 0.png 0.png 2.png 2.png 127 requests | 1.1 MB transferred General Request URL: http://localhost/api/articles/Acayucan/Violence%20against%20civilians/Mexico Request Method: GET Status Code: 200 OK Remote Address: [::1]:80 Referrer Policy: strict-origin-when-cross-origin Response Headers View source Connection: keep-alive Content-Length: 2061 Content-Type: application/json; charset=utf-8 Date: Mon, 19 Sep 2022 04:25:27 GMT ETag: W/"80d-5k4sd8hDLG/ZX4NO6TOv0RWCso8" Keep-Alive: timeout=5 X-Powered-By: Express Request Headers View source Accept: */* Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Connection: keep-alive Cookie: _ga=GA1.1.1470345177.1659940964 Host: localhost If-None-Match: W/"80d-FNeHwwRPX87mykxaOReHQTjfCEQ" Referer: http://localhost/Home sec-ch-ua: "Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105" sec-ch-ua-mobile: ?1 sec-ch-ua-platform: "Android" Console Issues What's New top Filter Default levels 63 Issues: 60 2 1 DevTools failed to load source map: Could not load content for chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/browser-polyfill.js.map: System error: net::ERR_FILE_NOT_FOUND {viewcounter: 128} CounterAPI.js:24
5	 Conflict Mapper This page has been viewed 130 times All Conflicts Conflicts Near Me Please select a conflict on the map to show a generated article Island Suomi / Finland Sverige United Kingdom Беларусь Deutschland Brasil Leaflet | © OpenStreetMap contributors Blocked Requests 3rd-party requests 500 ms 1000 ms 1500 ms 2000 ms 2500 ms 3000 ms 3500 ms 4000 ms 4500 ms Headers Preview Response Initiator Timing Cookies Name Home leaflet.css leaflet.js main.cc8861b2.css main.cb645303.js react_devtools_backend.js add 50 6.png 6.png 6.png 6.png 3.png 3.png 4.png 4.png 2.png 2.png 3.png 3.png 4.png 4.png 5.png 5.png 2.png 2.png 5.png 49 requests | 532 kB transferred General Request URL: http://localhost/api/acled/50 Request Method: GET Status Code: 304 OK Remote Address: [::1]:80 Referrer Policy: strict-origin-when-cross-origin Response Headers View source Connection: keep-alive Date: Mon, 19 Sep 2022 04:27:08 GMT ETag: W/"5e7b-6mFMVkOGFhZj+WCRxQWd7UfGtBg" Keep-Alive: timeout=5 X-Powered-By: Express Request Headers View source Accept: */* Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Connection: keep-alive Cookie: _ga=GA1.1.1470345177.1659940964 Host: localhost If-None-Match: W/"5e7b-6mFMVkOGFhZj+WCRxQWd7UfGtBg" Referer: http://localhost/Home sec-ch-ua: "Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105" sec-ch-ua-mobile: ?1 sec-ch-ua-platform: "Android" Sec-Fetch-Dest: empty Sec-Fetch-Mode: cors Console Issues What's New top Filter Default levels 2 Issues: 1 1 DevTools failed to load source map: Could not load content for chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/browser-polyfill.js.map: System error: net::ERR_FILE_NOT_FOUND {viewcounter: 130} CounterAPI.js:24
6	 Conflict Mapper This page has been viewed 130 times All Conflicts Conflicts Near Me Coast of Loiza - Looting/property destruction Property destruction: On 9 September 2022, near the Coast of Loiza, Loiza, police officers seized 47 blocks of cocaine valued 4.8 million US dollar, one million US dollar cash, five pistols, three rifles and ammunition. No one was arrested. It is not possible to generate an article for this conflict Island United Ki Espa United States Coast of Loiza - Looting/property destruction Colombia

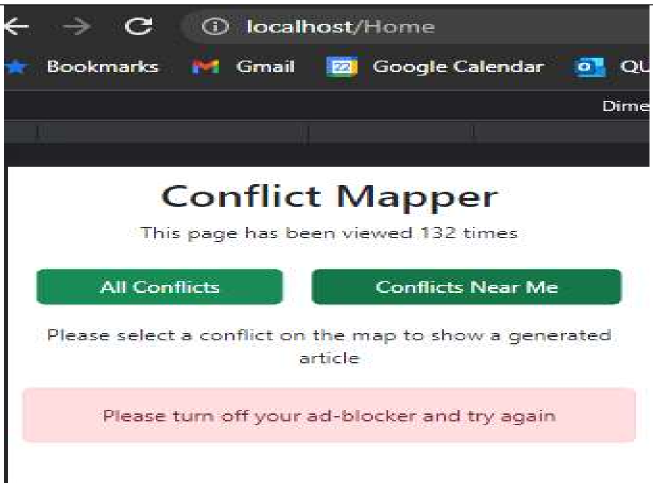
<table>
<tr><td>7</td><td>


Conflict Mapper

This page has been viewed Counter Unavailable times

All Conflicts | Conflicts Near Me

Please select a conflict on the map to show a generated article

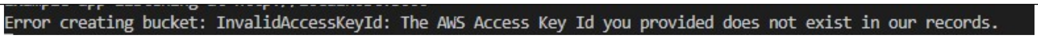
The conflicts backend may be offline. Sorry for any inconvience
</td></tr>
<tr><td>8</td><td>


Conflict Mapper

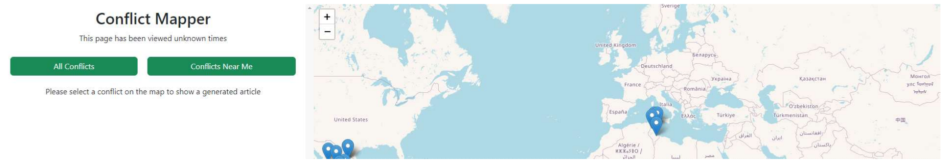
This page has been viewed 132 times

All Conflicts | Conflicts Near Me

Please select a conflict on the map to show a generated article

Please turn off your ad-blocker and try again
</td></tr>
<tr><td>9</td><td>


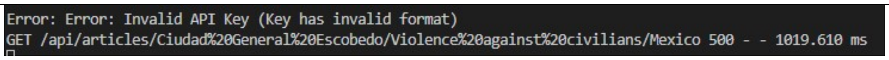
<pre>Error creating bucket: InvalidAccessKeyId: The AWS Access Key Id you provided does not exist in our records.</pre>



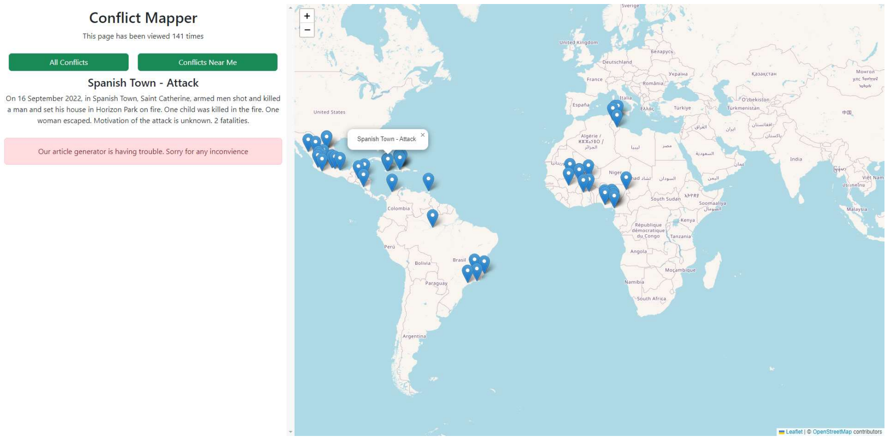
Conflict Mapper

This page has been viewed unknown times

All Conflicts | Conflicts Near Me

Please select a conflict on the map to show a generated article
</td></tr>
<tr><td>10</td><td>


<pre>Error: Error: Invalid API Key (Key has invalid format)
GET /api/articles/Ciudad%20General%20Escobedo/Violence%20against%20civilians/Mexico 500 - - 1019.610 ms</pre>



Conflict Mapper

This page has been viewed 141 times

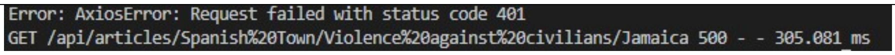
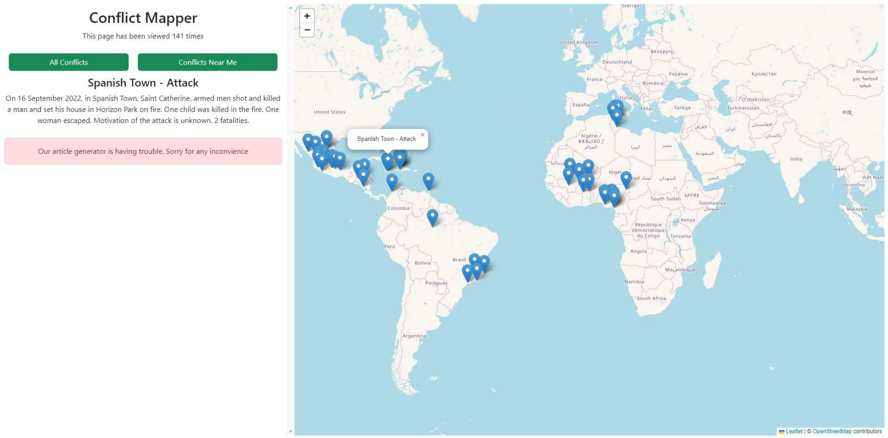
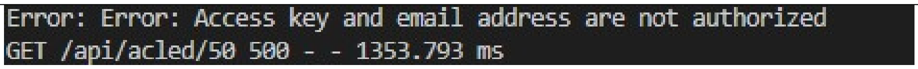
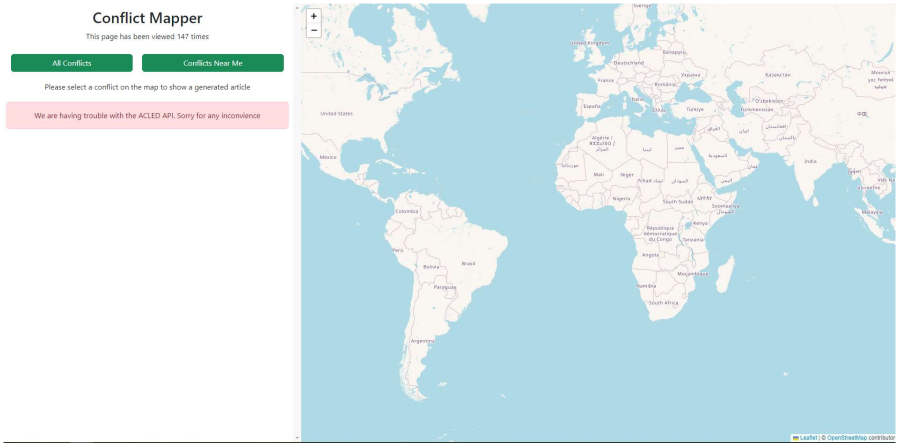
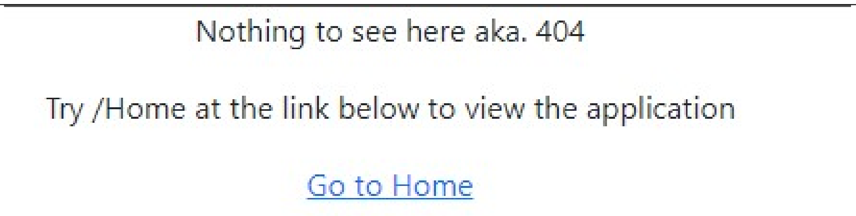
All Conflicts | Conflicts Near Me

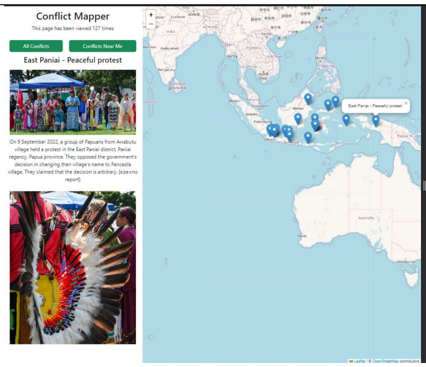
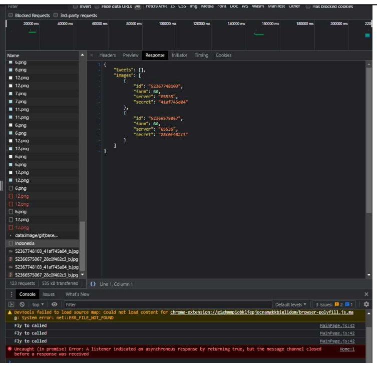
Spanish Town - Attack

On 16 September 2022, in Spanish Town, Saint Catherine, armed men shot and killed a man and set his house in Horizon Park on fire. One child was killed in the fire. One woman escaped. Motivation of the attack is unknown. 2 fatalities.

Our article generator is having trouble. Sorry for any inconvience

</td></tr>
</table>

11	 Error: AxiosError: Request failed with status code 401 GET /api/articles/Spanish%20Town/Violence%20against%20civilians/Jamaica 500 - - 305.081 ms  Conflict Mapper This page has been viewed 141 times All Conflicts Conflicts Near Me Spanish Town - Attack On 16 September 2022, in Spanish Town, Saint Catherine, armed men shot and killed a man and set his house in Horizon Park on fire. One child was killed in the fire. One woman escaped. Motivation of the attack is unknown. 2 fatalities. Our article generator is having trouble. Sorry for any inconvience + − Spanish Town - Attack × Leaflet | © OpenStreetMap contributors	
12	 Error: Error: Access key and email address are not authorized GET /api/acled/50 500 - - 1353.793 ms  Conflict Mapper This page has been viewed 147 times All Conflicts Conflicts Near Me Please select a conflict on the map to show a generated article We are having trouble with the ACLED API. Sorry for any inconvience 	
13	 Nothing to see here aka. 404 Try /Home at the link below to view the application Go to Home	

14	
	Conflict Mapper
	This page has been viewed 127 times
	All Conflicts
	Conflicts Near Me
	East Paniai - Peaceful protest
	On 9 September 2022, a group of Papuans from Awabutu village held a protest in the East Paniai district, Paniai regency, Papua province. They opposed the government's decision in changing their village's name to Pancasila village. They claimed that the decision is arbitrary. [size=no report]
	 +
	−
	East Paniai - Peaceful protest ×
	Leaflet | © OpenStreetMap contributors
	
	 Filter · Invert · Hide data URLs · All · Fetch/XHR · JS · CSS · Img · Media · Font · Doc · WS · Wasm · Manifest · Other · Has blocked cookies
	Blocked Requests · 3rd-party requests
	Name · Headers · Preview · Response · Initiator · Timing · Cookies
	6.png
	6.png
	12.png
	12.png
	7.png
	7.png
	11.png
	11.png
	6.png
	6.png
	12.png
	12.png
	6.png
	6.png
	12.png
	12.png
	6.png
	12.png
	12.png
	6.png
	12.png
	12.png
	data:image/gif;base...
	Indonesia
	52367748103_41af745a04_b.jpg
	52366575067_28c0f402c3_b.jpg
	52367748103_41af745a04_b.jpg
	52366575067_28c0f402c3_b.jpg
	123 requests · 535 kB transferred
	{
	"tweets": [],
	"images": [
	{
	"id": "52367748103",
	"farm": 66,
	"server": "65535",
	"secret": "41af745a04"
	},
	{
	"id": "52366575067",
	"farm": 66,
	"server": "65535",
	"secret": "28c0f402c3"
	}
	]
	}
	Line 1, Column 1
	 Console · Issues · What's New
	top · Filter · Default levels · 3 Issues: 2 1
	DevTools failed to load source map: Could not load content for chrome-extension://gighmmpiobklfepjocnamgkkbiglidom/browser-polyfill.js.map: System error: net::ERR_FILE_NOT_FOUND
	Fly to called MainPage.js:42
	Fly to called MainPage.js:42
	Fly to called MainPage.js:42
	Uncaught (in promise) Error: A listener indicated an asynchronous response by returning true, but the message channel closed before a response was received Home:1	

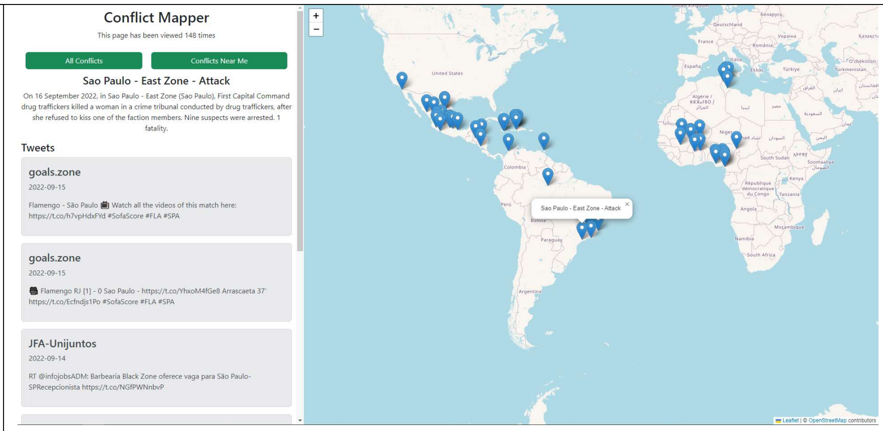
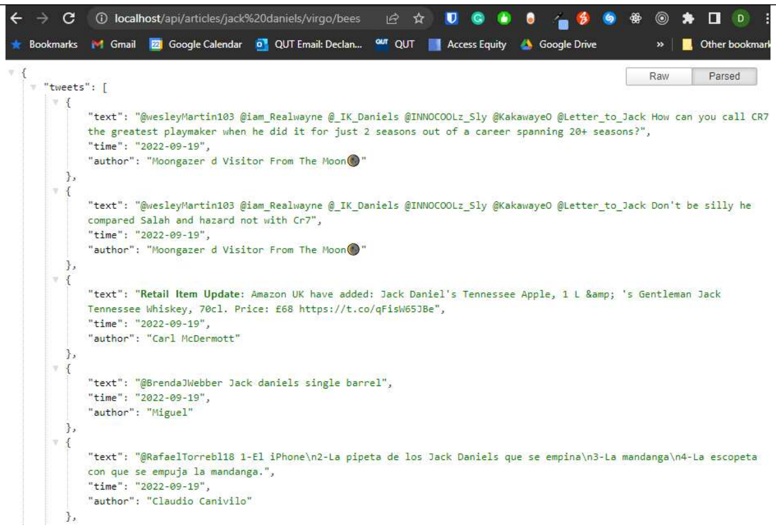
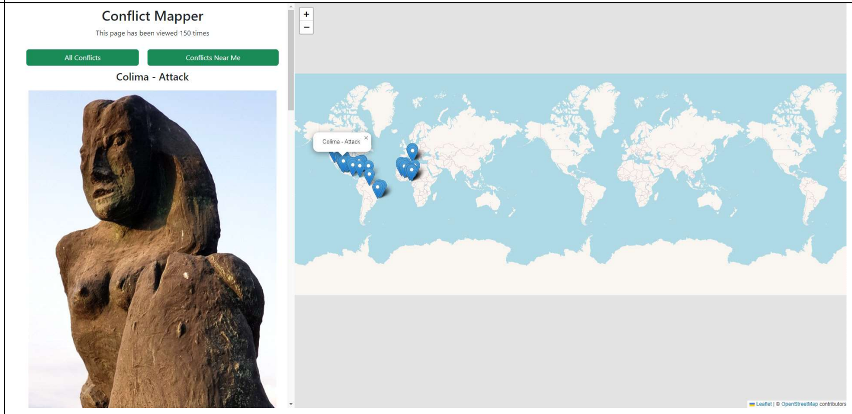
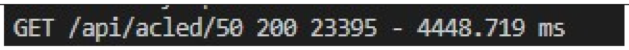
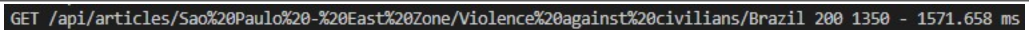
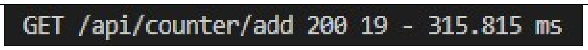
15	 Conflict Mapper This page has been viewed 148 times All Conflicts Conflicts Near Me Sao Paulo - East Zone - Attack On 16 September 2022, in Sao Paulo - East Zone (Sao Paulo), First Capital Command drug traffickers killed a woman in a crime tribunal conducted by drug traffickers, after she refused to kiss one of the faction members. Nine suspects were arrested. 1 fatality. Tweets goals.zone 2022-09-15 Flamengo - São Paulo 📺 Watch all the videos of this match here: https://t.co/h7vpHdxFYd #SofaScore #FLA #SPA goals.zone 2022-09-15 Flamengo RJ [1] - 0 Sao Paulo - https://t.co/YhxoM4fGe8 Arrascaeta 37' https://t.co/Ecfndjs1Po #SofaScore #FLA #SPA JFA-Unijuntos 2022-09-14 RT @infojobsADM: Barbearia Black Zone oferece vaga para São Paulo-SPRecepcionista https://t.co/NGfPWNnbvP + − Sao Paulo - East Zone - Attack Leaflet | © OpenStreetMap contributors
16	 localhost/api/articles/jack%20daniels/virgo/bees Bookmarks Gmail Google Calendar QUT Email: Declan... QUT Access Equity Google Drive Other bookmark Raw Parsed { "tweets": [{ "text": "@wesleyMartin103 @iam_Realwayne @_IK_Daniels @INNOCOOLz_Sly @KakawayeO @Letter_to_Jack How can you call CR7 the greatest playmaker when he did it for just 2 seasons out of a career spanning 20+ seasons?", "time": "2022-09-19", "author": "Moongazer d Visitor From The Moon🌑" }, { "text": "@wesleyMartin103 @iam_Realwayne @_IK_Daniels @INNOCOOLz_Sly @KakawayeO @Letter_to_Jack Don't be silly he compared Salah and hazard not with Cr7", "time": "2022-09-19", "author": "Moongazer d Visitor From The Moon🌑" }, { "text": "**Retail Item Update**: Amazon UK have added: Jack Daniel's Tennessee Apple, 1 L &amp; 's Gentleman Jack Tennessee Whiskey, 70cl. Price: £68 https://t.co/qFisW65JBe", "time": "2022-09-19", "author": "Carl McDermott" }, { "text": "@BrendaJWebber Jack daniels single barrel", "time": "2022-09-19", "author": "Miguel" }, { "text": "@RafaelTorrebl18 1-El iPhone\n2-La pipeta de los Jack Daniels que se empina\n3-La mandanga\n4-La escopeta con que se empuja la mandanga.", "time": "2022-09-19", "author": "Claudio Canivilo" },
17	 Conflict Mapper This page has been viewed 150 times All Conflicts Conflicts Near Me Colima - Attack + − Colima - Attack Leaflet | © OpenStreetMap contributors
18	 GET /api/acled/50 200 23395 - 4448.719 ms
19	 GET /api/articles/Sao%20Paulo%20-%20East%20Zone/Violence%20against%20civilians/Brazil 200 1350 - 1571.658 ms
20	 GET /api/counter/add 200 19 - 315.815 ms

Figure 3:

docker run \

```
-e AWS_ACCESS_KEY_ID=" " \
-e AWS_SECRET_ACCESS_KEY=”" \
-e AWS_SESSION_TOKEN=" " \
-e ACLED_KEY="" \
-e TWITTER_KEY=" " \
-e FLICKR_KEY="" \
-p 80:3000 \
-it declanbarrett/conflict-mapper
```