

Phase 1 Technical Report

[Website Link](#)

Group 06 Members: Aryan Samal, Rohan Damani, Will Matherne, Jeremy Nguyen, Kenny Nguyen

1. Introduction

Purpose and Motivation

PalestineWatch aims to raise awareness regarding the state of Palestine as well as inform others on what they can do to help out. We include a number of support groups that are currently aiding Palestine, countries that serve as places of asylum for Palestinian Refugees, and news articles documenting current events in Palestine.

This report serves as a technical background for anyone wishing to understand the tools/technologies used to build our website.

Questions We Answer

- What is the latest news regarding this conflict?
- What are people doing to help?
- What can you do to help?

2. Models and Instances

Model Overview

For this phase, each model has three instances. Every instance connects to 2 instances of different models.

- **News**
 - Represents articles that are related to the Palestine-Israeli conflict. May include articles about other countries involved in the conflict.
 - **Author:** Creator of the article.

- **Title:** Article title.
- **Description:** Represents a description of the article.
- **url:** Link to the actual article.
- **urlToImage:** Link to the article's image.
- **PublishedAt:** Date of which the article was published.
- **Content:** Represents the full content of the article.
- **Source:** Source from which the article was published by.
- **Countries**
 - Represents countries that are supporting Palestine by allowing for Palestinian Refugees to seek asylum.
 - **Country Name:** Name of the country displayed.
 - **Flag:** Flag of the country.
 - **Capital:** Capital city of the country.
 - **Population:** # of people living in this country.
 - **Region:** The overall region in which the country is located.
 - **Sub-Region:** Smaller, more specific region the country is located in.
 - **CountryData:** Displays a tuple that represents the # of refugees by year.
- **Support Groups**
 - Represents Support Groups that are helping Palestine by providing donation, supplies, etc.
 - **Name:** Support group name.
 - **Email:** Email of the support group.
 - **City:** City the group operates out of.
 - **State:** State the group operated out of.
 - **Zip Code:** Zip Code of the support group.
 - **Link:** Support group's website link.

3. API Documentation

<https://documenter.getpostman.com/view/38731121/2sAXxLBDwn>

4. Toolchain

The main tools we used were Bootstrap-React for styling components, Axios for fetching data from APIs, Next.js for creating our frontend interface, Docker for creating a consistent development environment, and Cheerio for parsing/manipulating HTML.

5. Hosting

Our website is hosted on AWS Amplify. We chose this provider because of its “easy to start, easy to scale” design and architecture.

6. Architecture

The site is primarily divided into pages that are built with individual components. We have a pages folder where all the pages can be found as well as a components folder to access all of the components.

The first page the user is greeted with is the splash page where they can then navigate to a multitude of other pages using the Navigation Bar at the top of the page. The splash page also contains some history about the Palestine-Israel conflict as well as our motivation for creating the site. The About page has information related to all of our group members, including bios, number of commits, pictures of us, as well as some general project information.

The next 3 nav bar items are the News, Support Groups, and Countries tabs. They each link to a page that houses the 3 instances of each particular model. Each model card displays a variety of information and is also clickable, allowing you to display more information. When you click on a card, it will navigate you to a page named [id].js in either the countries, News Pages, or support-groups folders. This will then allow you to view a full page of information on the given topic.

7. Challenges

A major challenge our group faced was related to acquiring data using APIs. Although we did vet the API in the RFP stage, we did not realize that some of the APIs would have hidden flaws. For example, many of the news APIs had a content paywall which would not allow you to access the full article content as a json object. This was quite frustrating at first, but we managed to make it work. Another major challenge we faced was deploying the website on AWS Amplify. We had a lot of issues where it said the website was deployed but when we tried to click on the link, it would say the website was not found.