CS 373 Software Engineering IDB Group 03

Team Members: Milan Dhaduk, Audrey Tan, Zakaria Sisalem, Pooja Vasanthan

Link to Website: https://wildwareness.net/

Phase 1

WildWareness

- ➢ WildWareness is a web application that provides users with information about wildfires in the state of California. The intended audience is those affected by the fires who need help and/or those interested in helping those affected. Given how recent the issue is, and the severe devastation caused by the fires, we decided that developing this application would be valuable.
- ➢ The website provides information about wildfire incidents, emergency shelters, and community reports (the models). Each has specific instances and information related to those instances. The information is also related in terms of proximity (for example, on a page about a specific wildfire incident, the user can find information about the specific wildfire incident, as well as information on wildfire incidents and community reports that are located near that incident).
- ➢ Users can easily navigate the pages and find the information they seek.

Research and Data Sources

- ➢ First, using the internet, we performed the necessary research on the topic, compiling information about the wildfires in California and sources from which we can acquire data and support the backend of the application in a future phase.
- ➢ After our research, we decided to use these APIs/data sources to acquire the information:
  - CAL FIRE – Reports active and past wildfires with containment details (2025)
  - NewsData.io API- Community/local reports and headlines fetched from NewsData.io
  - Google Places API- Google's API is used for fetching information about a location on Google Maps.
  - Google Maps Geocoding API - a service that allows you to convert addresses into geographic coordinates (latitude & longitude) and vice versa (reverse geocoding).
  - List of Emergency Shelters - a website containing a list of emergency shelters by county and city in the form of text.

Initial Setup

- The project repository was created on GitLab and is available publicly. We own the repo, and we utilized the issue tracker to keep track of the tasks we needed to complete. GitLab provided us with an easy way to make changes and share it with the rest of the team.
- In this phase, we focused on coding the information from the data sources and APIs in HTML for the website using the Visual Studio Code IDE. After git cloning the repo to our local machines, we created a folder called "frontend" to house our HTML files, a "scripts" file to contain calls to our APIs (not required for this phase but to test out being able to get the information), a Makefile, a .gitignore file, a README file, and a .gitlab-ci.yml file.
- Since we kept the files in our code repo minimal to only what we need for Phase 1, the .gitignore file doesn't have many files that should be ignored. In the future, once we start using React, we will also include the node_modules in the .gitignore, which are libraries related to React that will clutter our repo if left.
- The README contains information relevant to our project. It's a condensed version of this technical report with details on project team members, website purpose/audience, data sources, models, and instances (and attributes).
- We included some common useful commands in the Makefile, such push, pull, status, and check. Throughout the project, we will add more as needed.
- The .gitlab-ci.yml file also doesn't contain much as of now. However, we are including a check for the index.html file. We will also expand this file as we go.

Website Architecture

- The general layout of the website includes five pages: the Splash page, Wildfire Incidents page, Emergency Shelters page, Community Reports Page, and About Page. These pages can be navigated using the navigation bar at the top of the website. Each page has an associated HTML file in the code repo.
- Models: Wildfire Incidents, Emergency Shelters, Community Reports
- Model Instances and Attributes
    - Wildfire Incidents
        - Instances: Palisades Fire, Eaton Fire, Oak Fire
        - Attributes: Name, County, Location, Date/Time, Acres Burned
    - Emergency Shelters
        - Instances: Sonoma Veterans Building, Sebastopol Community Center, Contra Costa Community College
        - Attributes: Name, Address, City, Contact, Open/Closed Status
            - ****Note - the open and closed status will be based on hours of operation, not if they are, still providing sheltering; this is*

*mentioned on the webpage and users are encouraged to contact the location for up-to-date information****

- Community Reports
  - Instances: BBC News Firefighters battle huge blaze near Los Angeles as winds pick up, BBC News Maps and images reveal scale of LA wildfire devastation, BBC News LA firefighters battle to contain monster inferno as death toll rises
  - Attributes: Source, Date, Category, Region, Reporter

➢ Website Pages:

- Splash Page
  - This is the main page containing information about the overall purpose of the website, the intended audience, the kind of information users can find on the website, and an image carousel.
- About Page
  - This page contains information about the website, the website developers (the four of us), the data sources and tools used, and the significance of the data we used.
  - Team member information is displayed in a similar card format and contains the team member's name, short bio, major responsibilities, and GitLab stats (# of commits, issues, and unit tests contributed).
- Wildfire Incidents Page (Model #1)
  - The page contains information about recent wildfires in California. Each fire is presented in card format with five attributes and an image.
  - The cards are clickable, each leading to a page with more details on the incident (instance page). The details consist of text (description and attributes) and multimedia (images and embedded videos). The cards at the bottom of the page include information on emergency shelters and community reports near the wildfire incident. These cards are also clickable and lead to the instance pages of those respective items.
- Emergency Shelters Page (Model #2)
  - This model page has information about locations providing shelter for those affected by the fires. Like the Wildfire Incidents, the emergency shelters (instances) are laid out as clickable cards containing the five attributes and images, which lead to pages containing more information about the shelter.
  - The instance pages again contain text and multimedia. There is an image carousel, the five attributes, description, embedded Google Maps of shelter location, cards for nearby wildfire incidents and community reports, and images and links to websites relevant to the location. There

are other embedded features (for example, an embedded page for a volunteering form on the location's website).

- Community Reports (Model #3)
  - This model page contains information about community reports (i.e. news) regarding fire updates. Like for the other model pages, this model page contains clickable cards for community reports (instances) that lead to pages that include more information on that report. The cards again have the five attributes and images.
  - The instance pages have text listing the five attributes, a description of the report, and multimedia, such as images, Google Maps locations, embedded videos, and graphs. At the bottom of the page, there are clickable cards to emergency shelters and wildfire incidents nearby.
- At the bottom of each instance page, there's a back button leading back to the instance's model page, and at the bottom of each model, the total number of instances on the page is listed.

Code for Website

- In the "frontend" folder, we have all the HTML files that code the information and media displayed on the website. To develop this code, we referenced previous projects, the internet, and ChatGPT. Also, we included a javascript file called about.js to call the GitLab API and obtain the GitLab stats for each of the team members.
- In the "scripts" folder, we have Python scripts (.py files) that make calls to the APIs listed to get the necessary information. We aren't actively using this for our website yet, but we wanted to test the APIs to see how they work.

User Stories

- Our customer was the ASPIRE group, and they provided us with five stories to implement on our website. The stories were added to our Issue Tracker in GitLab.

- User Stories We Gave to the Developer Team by Adding it to Their GitLab Issue Tracker
  - User Story #1: For the "resources/services" cards, can you include the contact information of that center if you can find It and the cost for the center if there is one? Including the contact information can make it easier for the users to reach the resources they need quickly. Moreover, adding any associated costs, if there are any, and if you are able to find it, will be nice, so the users aren't surprised when they follow through with that service.

- User Story #2: For the instances of the community events model, can you include the full name of the Organizer instead of the acronym only? Sometimes, people may not be familiar with the acronym, despite how popular the organizer may be. Adding this information could be useful, so users don't need to research this information themselves.
- User Story #3: For the resources/services instances that don't have a website available, can you include other relevant links such as Facebook, Yelp, or county websites about those resources/services? Even if the website isn't directly created for that service or resource, community discussion forums like Facebook and Yelp will expose the users to others who share similar interests. Any information is information, so this might be a good addition if possible.
- User Story #4: For the instances pages for the community events, can you include descriptions of the event, too, and any links to relevant websites that have information about the event (a community website, Facebook page, yelp page, etc)? Most of the time, community events are publicized using social media forums like Facebook and Yelp. With these pages, users can find real-time updates regarding the event and others who share their interests.
- User Story #5: For the community events instances, can you also include the cost for the event, if there is one, and the purpose of the event (fundraising, raising awareness, performance/entertainment, networking, etc)? This can be helpful information for users, as it can help them narrow down which events they want to prioritize attending if they can't attend them all. Also, including the cost, if there is one and if possible, lets the users know upfront rather than finding out at the event.

➢ User Stories Provided to Us by Our Customer along with the Our Responses Provided in GitLab Issue Tracker Comments

- User Story #1: It would be nice to see how many structures were destroyed in the wildfire. This would give us a better picture on the impact of each wildfire.
  - Our Response: We agree that this is a good attribute to have. If you go to the instance pages for each of the wildfire incident instances, we added an attribute called "Number of Structures Destroyed."
- User Story #2: Could you also include how well the fire was contained (I'm not sure if this is what you guys mean by severity).
  - Our Response: Yes, we added this attribute now. Previously, we were going to do severity, but after reading through the data source more, we found a measure of containment instead, which indicates how well the fire was contained. On the instance pages for each of the wildfire incidents, we included a "Containment Percentage" attribute under the attributes section.

- User Story #3: We would also like to be able to see how large (in acres) the wildfire was.
  - Our Response: Yes, this is a useful attribute. The wildfire incident instances have an "Acres Burned" attribute, which is on their respective cards on the wildfire incidents model page and on the specific instance pages under the attributes section.
- User Story #4: Could you also make it so that we can see information on which fire departments were managing each wildfire. This would give us information on which fire departments may need assistance or extra funding.
  - Our Response: This is a good idea. We listed this information under the attributes section on each of the wildfire incident's instance pages. The attribute is listed as "Fire Department(s) Involved."
- User Story #5: We would also love to be able to see the amount of casualties due to each fire (once again, not too sure if this is what is meant by severity). Knowing this amount would help us be aware and spread awareness about these dangerous wildfires.
  - Our Response: Yes, we agree. It would be a good idea to include the casualties information. Like for the previous requests, we added this information under the attributes section of each of the wildfire incident's instance pages. The attributes are listed as "Number of Fatalities" and "Number of Injuries."

➢ We were able to implement all the user stories provided to us. They were all related to adding more information about the wildfire incidents, so we referenced the CAL FIRE API and made the changes (changes are described next to “Our Response” above).

Hosting

➢ We decided to host our static website using AWS Amplify and also obtain a custom domain using Amazon Route 53 service.Amazon

➢ Why Amazon Amplify?

- **Continuous Deployment:** Automatic deployments from GitHub, GitLab, Bitbucket, and AWS CodeCommit.
- **Scalability:** Supports both static and dynamic applications with automatic scaling.
- **Secure Hosting:** Provides HTTPS by default**.**
- **Built-in CI/CD:** Streamlines development by automating builds, tests, and deployments.

    - **Supports SSR & SPA:** Works with frameworks like React, Vue, Angular, and Next.js.
- ➢ Other than simplification, why Amazon Route 53?
    - Register and Manage Custom Domains.
    - Configures DNS Routing to direct traffic to your Amplify app.
    - Enable HTTPS with AWS Certificate Manager for secure connections.

- ➢ **Note**: In order to enable HTTPS, we had to do the following:
    - In Amazon Route 53, go to Hosted Zones and find your domain.
    - In AWS Certificate Manager, request an SSL certificate for your domain.
    - Validate the certificate using DNS verification.
    - After validation, Amplify will use HTTPS for secure connections.

API Documentation

- ➢ For API design/documentation, we used Postman to easily organize our initial API endpoint design. The Postman collection contains three directories: Wildfires, Emergency Shelters, and Community Reports. As the names suggest, each directory is associated with each model on our architecture. As of right now, we've only implemented two forms of GET requests: one to retrieve all instances of a certain model, and another to retrieve a single instance of a given model.
- ➢ The former contains optional query parameters that can be used to filter and/or sort the instances retrieved by this HTTP requests, like being able to sort by county or filter by location
- ➢ The latter is more simple and contains no optional query parameters but rather a single required path variable called 'id'. This is the variable that will be used to select a certain instance of a model and MUST be added to the URI in order to make a successful request.

Challenges

- ➢ During this phase, we ran into some obstacles regarding obtaining information and hosting the website through a third party, which we overcame.

- ➢ Hosting on 3rd Party
    - We are hosting the website on AWS. We aren't familiar with the technology, so it took us some time to configure the environment and have our website up and running.

- ➢ Emergency Shelters Information

- When researching emergency shelters in California, we found that only two shelters were provided by the state, which wasn't enough to meet the required number of instances. We were going to pick another model, but after some digging, we found a website containing a list of shelters provided by other institutions, such as churches, schools, and community centers. Therefore, we were able to move forward with the emergency shelters model.
- Although we were able to obtain a list of emergency shelters, the website didn't contain information on whether those locations were still offering sheltering services (which would be tied to the Open/Closed Status attribute for the instances). Therefore, we had to make use of a Google API to acquire information on the hours of operation of the shelters. This isn't the same as whether the location is still open for sheltering, so we include a disclaimer on the model page telling the reader to contact the location to check with them.

➢ Community Reports Information
- Initially, we intended to use a Twitter API, which we could call and filter to get data on tweets made by users that were relevant to the California fires. However, we noticed that the information we received from the API call didn't contain enough information to provide us with five attributes. Thus, we switched to using a News API, which gave us information on news reports related to the fires with enough attributes.

Tools

➢ Bootstrap Framework - used to design the website
➢ Google Maps- to obtain locations for certain instances
➢ AWS Hosting-used to host website
➢ Amazon Route 53- to obtain URL
➢ Postman- for API documentation/design
➢ Google Documents- for technical report documentation
➢ Visual Studio Code- IDE used for frontend development of the website
➢ GitLab- houses project repository and make it available for public view
➢ Grammarly
➢ Google (Stack Overflow, GeeksforGeeks, Research, etc)
➢ Slack for Communication

Other Sources

➢ HomelessAid- Referenced previous project to develop layout of website.

- ChatGPT- Used to develop the code for the website by asking about implementing features (i.e. embedded maps and videos) in HTML and how to write python scripts to use the public APIs. Also used to figure out how to use AWS and host the website.