<u>Disaster Ready</u>

Linnea Jones, Abby Garner, Cameron Maynor, Connor Locke, Sam Mankin

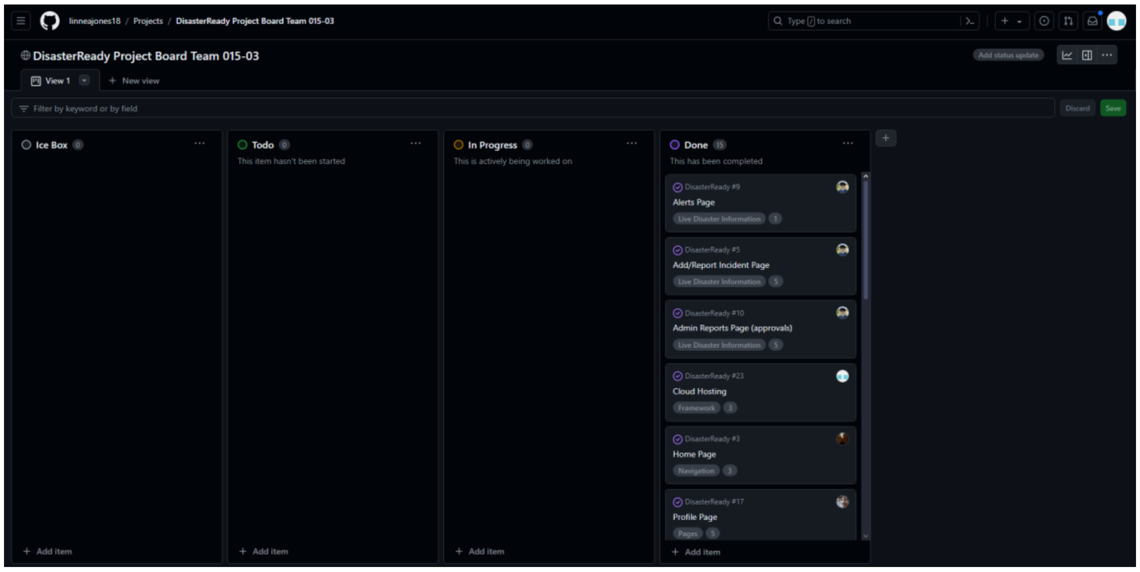
GitHub: linneajones18, AbbyGarner, Eduardo-Zone, connorlocke, sammankin

<u>Project Description</u>

Disaster Ready is a website which provides a hub for users to become informed of emergencies or hazards close to them. Users are unable to view any pages other than the login and register pages prior to logging in. Once a user logs in with an account they have created, they will be redirected to the home page where they can choose to make a report themselves, check alerts for their area, view available resources, and look around an embedded map for any nearby alerts. They may make a report themselves by providing the city the incident occurred in as well as the geographical coordinates for the pin to be placed on the map. The user may provide details of the incident and how they would classify it. The report is then dynamically added to the embedded map where other users can view it with the details the reporter provided. Users may also access a profile page where they can change their name, location, and provide a short bio. Information regarding local resources such as the fire department, police station, and emergency services are provided within the resources page along with educational material regarding emergency awareness and training.

<u>Github Project Board</u>

https://github.com/users/linneajones18/projects/2/views/1


linneajones18 / Projects / DisasterReady Project Board Team 015-03

DisasterReady Project Board Team 015-03

Ice Box 0

Todo 0
This item hasn't been started

In Progress 0
This is actively being worked on

Done 15
This has been completed

DisasterReady #9
Alerts Page
Live Disaster Information 1

DisasterReady #5
Add/Report Incident Page
Live Disaster Information 5

DisasterReady #10
Admin Reports Page (approvals)
Live Disaster Information 5

DisasterReady #23
Cloud Hosting
Framework 3

DisasterReady #3
Home Page
Navigation 3

DisasterReady #17
Profile Page
Pages 5

<u>Video</u>

https://youtu.be/S4dK7ih_eOw

VCS

https://github.com/linneajones18/DisasterReady.git

Contributions

Linnea Jones:

I designed the navbar in handlebars to provide a directory across any page of the site. I also designed and implemented the profile page. The edit profile page I created accepts user input for name, location, and bio, and stores changes made while keeping previous data if a box is left empty. I also wrote all the test functions besides the map page test functions. I also helped lead the group by checking in with the group regularly and ensuring that all parts of the project were being completed in a timely manner.

Abby Garner:

I helped format and make the docker files and set up, ensuring that the .YAML file was correctly configured and working so that our website could properly run. I also created the home page of the website which was linked to every other page. Along with this, I also created API routes in our index.JS file that linked all the pages together. Since the home page could only be accessed when a user is logged in, I also made sure that the session was set so that home page and others could only be accessed when logged in.

Cameron Maynor:

I helped design the database table and architectural structure for posting/approving alerts. This also entailed creating an admin approvals page to approve submitted reports (api routes, NodeJS, HTML front-end, bootstrap, handlebars, CSS, little SQL). However, this was unworking at the time of the website release and was not included in the final product. I also developed an alerts page which would list all posted alerts for the customer to see, but the format of this page was inferior to checking alerts on the map itself, as you could visualize location better, so we also decided to go another way on this page. I also performed most of the merging into GitHub and testing of the website after each commit to main, ensuring all main commits were fully working standalone products. I also designed a sketch logo for the website.

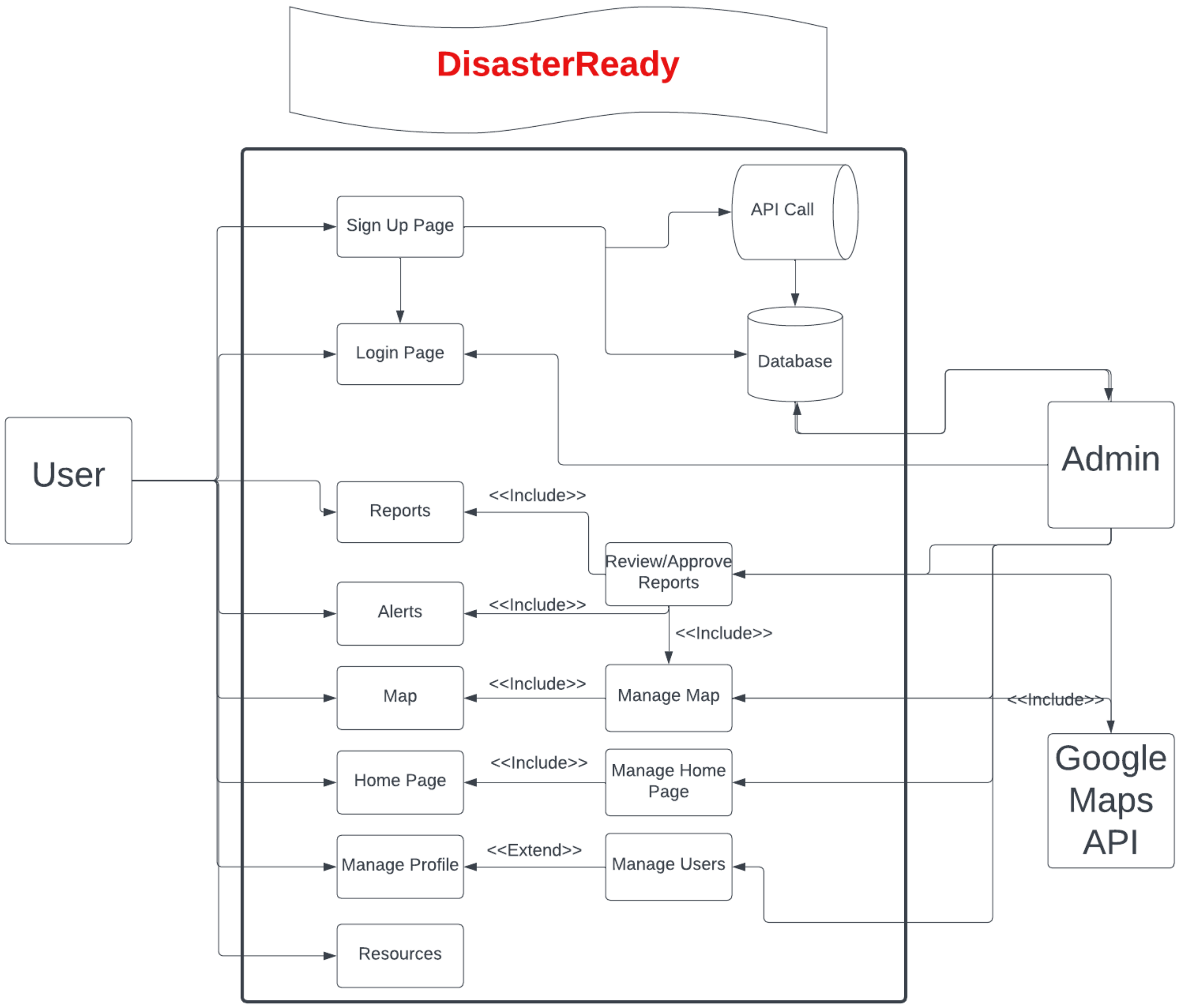
Connor Locke:

I developed the login and register pages front end and back end while ensuring that user credentials were hashed and properly stored into the connected database. Ensured users were only able to access specific pages of the website without authorized credentials. Constructed the necessary API routes for the login and register pages. Implemented the initial partials such as the navbar, footer, title, etc. which provided dynamic content and alerts across the site. Configured the Azure VM to host the website to be publicly accessible through an open domain name.

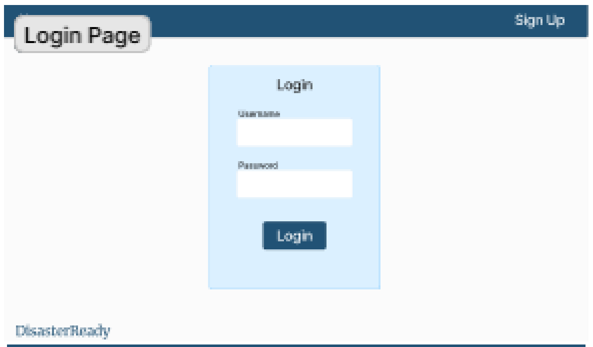
Sam Mankin:

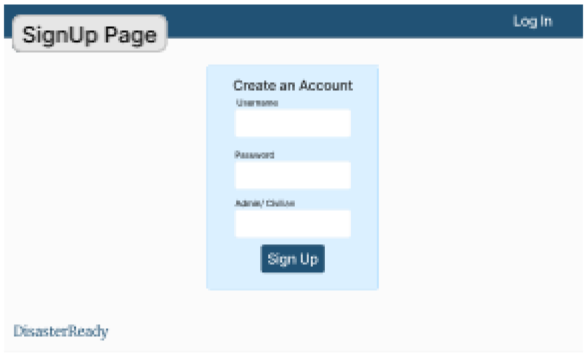
I developed the maps, check-alerts, and submit-report features for the application, I integrated the Google Maps API to display an interactive map, which allows users to visually locate and identify alerts. For the submit report functionality, users input data through a form, which is then stored in a SQL database. This stored information is subsequently retrieved and displayed as markers on the map in both the maps and check-alerts pages. I crafted both the front end and back end for these pages. The resources page was also designed by Me in which displays emergency information and pertinent information surrounding how to be better prepared for disasters. I did the back end for this page as well.

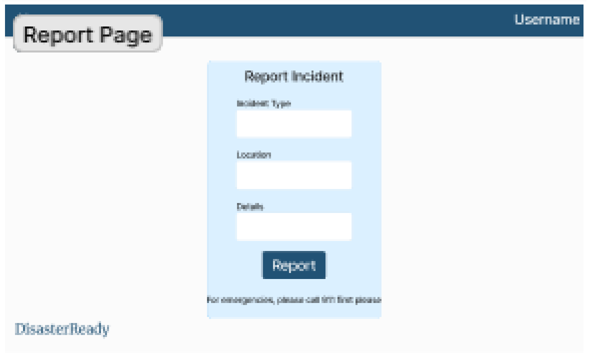
Use Case Diagram



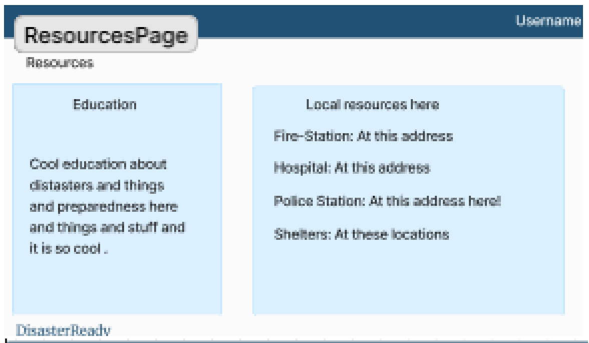
Original Wireframes







For emergencies, please call 911 first please



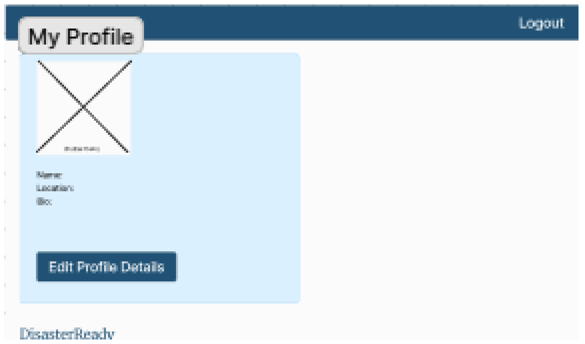
Cool education about distasters and things and preparedness here and things and stuff and it is so cool .

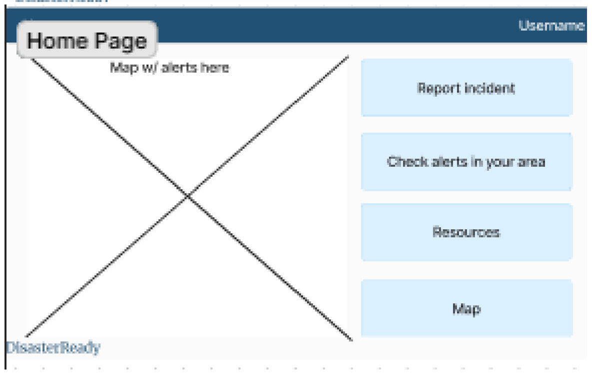
Fire-Station: At this address

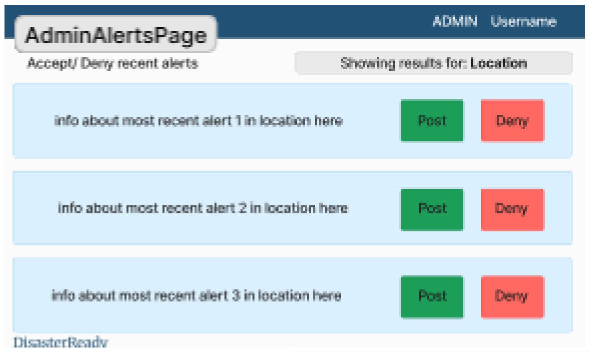
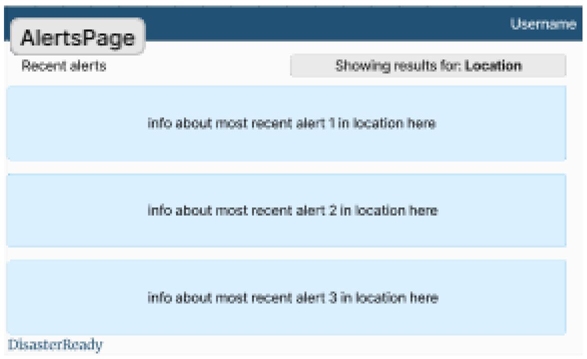
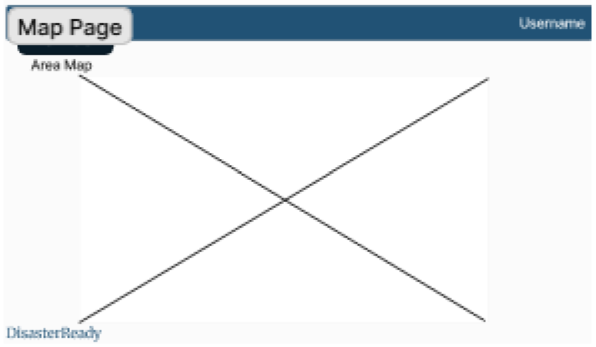
Hospital: At this address

Police Station: At this address here!

Shelters: At these locations






AdminAlertsPage
ADMIN Username
Accept/ Deny recent alerts
Showing results for: Location
info about most recent alert 1 in location here
Post
Deny
info about most recent alert 2 in location here
Post
Deny
info about most recent alert 3 in location here
Post
Deny
DisasterReady

AlertsPage
Username
Recent alerts
Showing results for: Location
info about most recent alert 1 in location here
info about most recent alert 2 in location here
info about most recent alert 3 in location here
DisasterReady


Test Results

Test Plan – Group 3

Sign-up page/Login in

User test cases:

Valid Credentials: Test logging in with a valid username and password to ensure the API authenticates correctly.

Invalid Username: Attempt to log in with a non-existent username and verify that the API returns an appropriate error response.

Invalid Password: Test logging in with a valid username but an incorrect password to confirm that the API rejects the request with the correct error message.

Database Entry: When sign in make sure that the login is added into the database and that the API correctly calls the new information.

Test data:

Valid Credentials:

Email: Sama8318@colorado.edu

Password: password123

Invalid Credentials:

Email: ihoppancakes
Password: I don't like syrup
Test environment: NodeJS environment set up with Mocha and Chai for unit testing. Mock data or test database with predefined user credentials for testing against.

<u>Test results:</u>
Valid Credentials: Redirect to the home page where information is listed.

Invalid Credentials: Redirect back to login page with a pop up showing that the information provided was not valid.

User acceptance testers: Nathan Martin-Bourne – Roommate

Navigation Bar
<u>User test cases:</u>
Valid Interactions: Users should be able to utilize the navigation bar when they are logged in with proper credentials and be redirected to the appropriate pages.

Invalid Interactions: If a user is not logged in with proper credentials, they should not be allowed to redirect off the login or register pages, and the navigation bar should not show up.

Proper Pages: The navigation bar should not be able to redirect a user to a page which they should not access without certain credentials, such as an admin page.

<u>Test data:</u>
Valid: A logged in user may use the navigation bar to navigate the various pages, but not other ones.

Invalid: A non-logged in user should not be able to navigate to pages besides the login and register pages.

Test environment: Localhost environment hosting the website utilizing Docker.

<u>Test results:</u>
Valid Results: Logged-in users should be able to navigate to the pages listed on the navigation bar, but not other ones.

Invalid Results: Not Logged-in users should only be able to access the login and register pages and not utilize the navigation bar.

User acceptance testers: Mitchell Kubina – CS Student/Roommate

Alerts Feature

<u>User test cases:</u>

Valid interactions: Users should be able to open the alerts page and be presented with chronological alerts from their local area. The page should update automatically when new alerts are added, and the user should also be able to filter alerts by location. The user should be able to select a new location to show alerts for and the alerts should match the chosen location of the user.

<u>Test data:</u>

Valid: A valid added alert using user login and pushing from Admin side (accepting alert) should show up on alerts page.

Invalid: Invalid user input alert (invalid fields) should not be accepted by administrators and should not show up on alerts page.

Test environment: Localhost environment hosting the website utilizing Docker.

Test results: When alerts are added by a user, they show up to all users on their alerts page.

User acceptance testers: Mitchell Kubina – CS Student/Roommate

Deployment

http://recitation-015-team-03.eastus.cloudapp.azure.com:3000/login

We successfully deployed the website using Azure and Docker containers. Docker was necessary to provide containerization which allowed for seamless and consistent deployment across the initial testing stages for the whole group. Azure was utilized as the cloud hosting platform due to its wide selection of machines and free usage for CU students. A docker container was deployed on the Azure VM which hosted the website and all of its data.