MIDTERM PROJECT PROPOSAL

Lightning Fast Weather

September 23, 2025

PREPARED BY

Nathan Copa
Erik Nissen

1. Introduction

Our names are Nathan Copa and Erik Nissen. We are both juniors in Computer Science at Iowa State University. Nathan does not have much background in HTML, CSS, or Javascript, but is taking this class to learn the basics of website creation, and improve their ability to use Javascript. Erik has used HTML and Javascript heavily for personal projects. One of these personal projects used a local WebSocket server to parse and display information from a game controller in a graphically pleasing and easy to understand manner.

We chose this topic to help improve our JavaScript capabilities and to learn how to more efficiently parse information from existing API services.

2. Purpose

Existing national weather websites are flawed in end-user experience. For example, weather.gov provides highly detailed information, but is difficult for a new user to understand and navigate. The Weather Channel's weather.com provides information in an easy to read manner, however it contains too much bloat on the website and is very slow as a result. We plan to create a website that is able to parse weather data quickly into an easily understandable format.

3. Goals/Objectives

- Create a website that parses data from a weather service API
- Must load and display data as fast as possible
- Refine our ability to use JavaScript
- Research and utilize efficient methods of using an API
- Modularity to add functionality should scope of the project increase
- Implement some form of graphical capability to provide information at a glance (i.e., clouds with rain and lightning to show that there is currently a thunderstorm)

4. Project

The core of this project will be the parsing of a weather service API, in an efficient, and easy to understand manner.

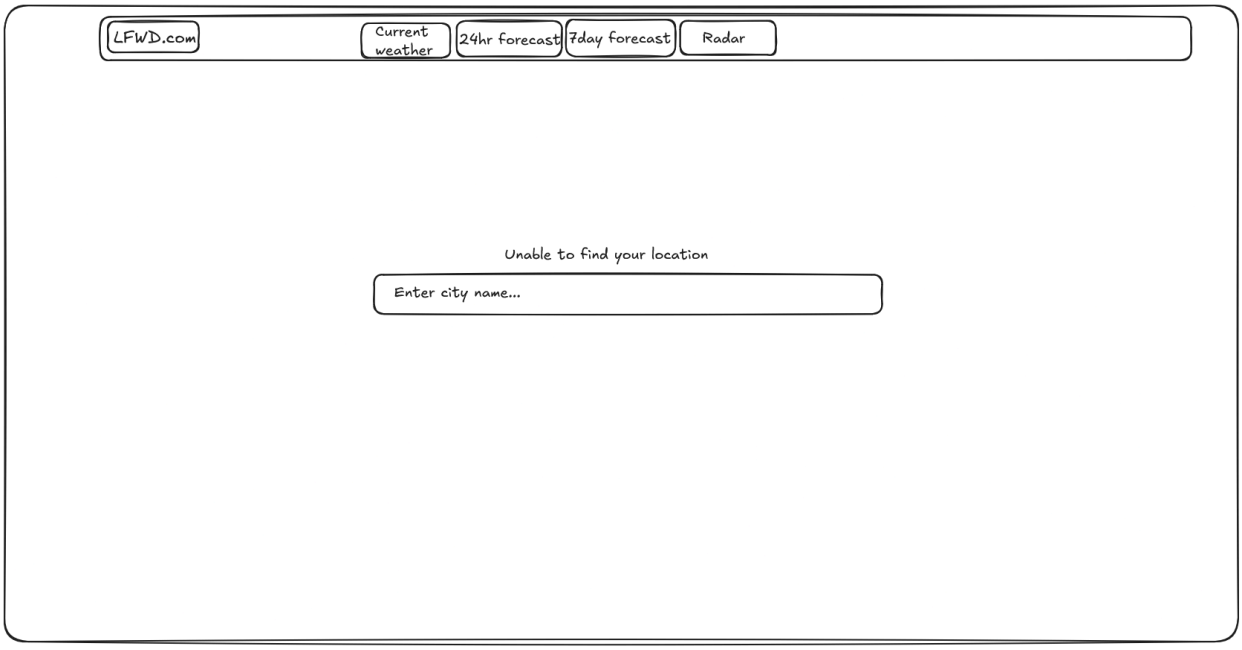
Core website functionality includes:

- The current weather at the time and location of website access.
- A 24 hour forecast of the current day (1 hour segments) showing temperature, weather events, humidity, etc.
- A seven day forecast.
- Weather radar, assuming API allows access.

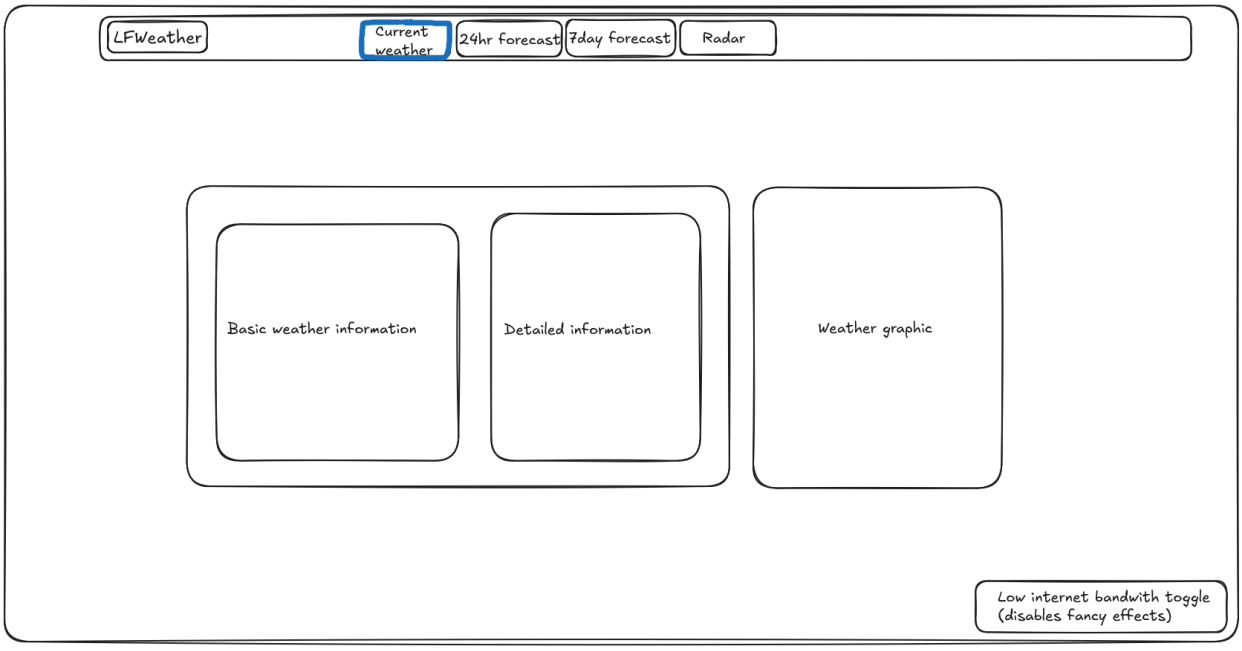
- A search feature to find the weather of cities in the United States.

We created an example wireframe for the website layout. By default, the webpage will use location services if allowed. If the browser has these services disabled, a different home page will be displayed. Otherwise, the website displays like normal. For users with low internet speed, a toggle button is available that prevents the user from requesting any assets that may increase bandwidth usage.

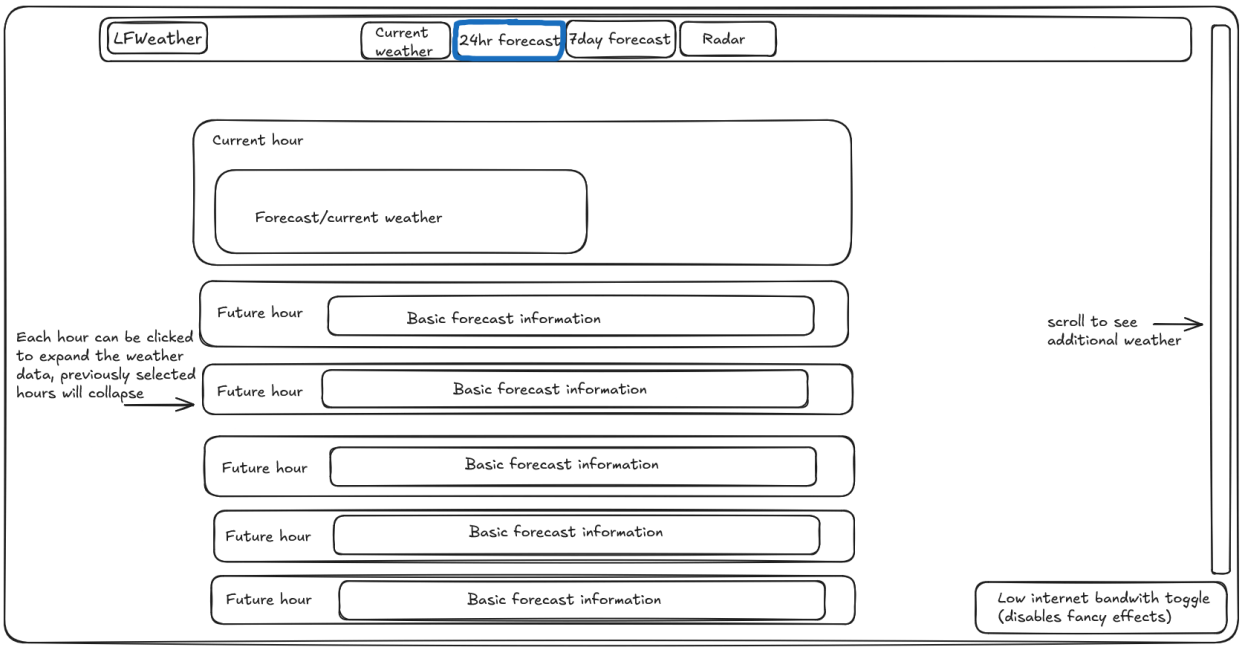
Home Page (w/o Location Access)



Home Page (w/Location Access)



24hr Forecast Page



The website should automatically detect the device's default dark mode settings. While there is a low bandwidth toggle, the website should still be very efficient by default, and it should only display if the webpage is loading slow, or manually toggled in a hidden menu should the user prefer.

5. Resources

- Programming languages - HTML, CSS, Javascript
- Github or Gitlab for cloud based version control
- National weather service API - https://www.weather.gov/documentation/services-web-api
- Time commitment from each team member should be no more than six hours per week per team member.

6. Future Work

Our modular approach should allow us to easily continue work for the purpose of the final project in this course. Website modularity will also allow us to increase or decrease the scope of the project to ensure it is finished by the due date, and allow us to implement our unfinished functionality into the final project. Some additional features could be automatically

embedding a current live weather report from the user's location, or a live video feed of weather using a city's highway camera feed.

7. Final Comments

For questions and additional clarification, please contact us at:

Nathan Copa
Email: ncopa@iastate.edu

Erik Nissen
Email: ernissen@iastate.edu