

CS4530 Final Project: Weather Forecast Modal

Group 605

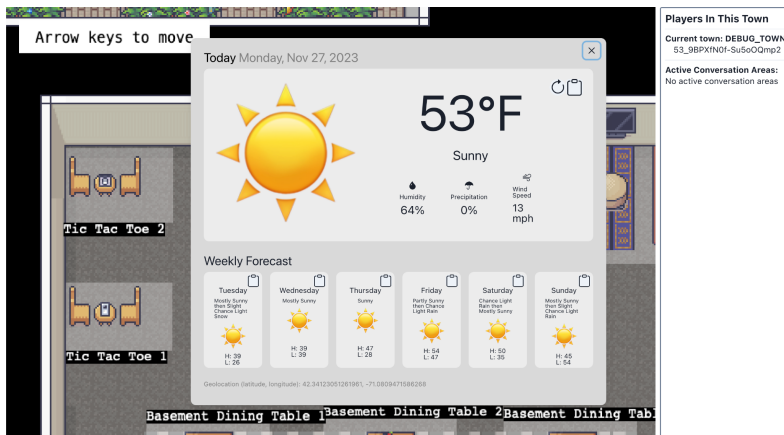
Our Feature: Weather Forecast Modal

Our feature aims to enhance Covey.Town's Conversation Area feature, which allows users to begin and join conversations in the town. Consider a situation where users want to talk to each other but do not have a relevant conversation starter. As the weather is a common conversation starter, we aim to solve the aforementioned problem by introducing a persistent weather forecast modal that can be opened by the user. Beyond this, a weather forecast display serves a practical purpose by allowing users to plan timely breaks from gaming based on the current forecast for their real-life current location. Additionally, this data adds a level of realism to the virtual town, making Covey.Town a more engaging digital world.

With just one simple step (pressing the **f** key) the weather forecast modal is opened. We automatically request access to the user's geolocation then fetch weather forecast data from the backend to display. Users can then exit the modal, or copy the information to their clipboard to easily share to others. Furthermore, to avoid unnecessary requests and ensure persistence, the forecast data is cached in the browser's local storage for reuse. If a user wishes to view an updated forecast from the backend, they simply click the modal's refresh button.

Demo and Source

View our demo site [here](#) and our code [here](#).



Our Technology Stack

Our weather forecast interface builds on the existing Covey.Town codebase by implementing the following features using their corresponding technologies,

- **Geolocation:** We utilize the *Geolocation API* to obtain accurate and seamless geolocation data from the user's browser. This API is supported by all major browsers
- **Forecast Data:** Our weather forecast information is pulled from the *National Weather Service (NWS) API*. From their open-source API, we fetch temperature, humidity, wind, and conditions data for the current and next 6 days for some given US-based location.
- **Forecast Caching:** To avoid unnecessary requests and offer a more robust solution, fetched forecast data is persisted in the browser's local storage using the *Web Storage API*.
- **User Interface:** We use *React* components alongside the *Chakra* library to create a visually pleasing and informative weather forecast interface. In addition, we render appropriate loading/error state messages to users to reflect the status of requests. Our weather display modal also includes a clipboard copy functionality for users to share their forecast in a text-based format.

Future Work

- **Location Support:** Our current implementation uses the free NWS API, which limits us to United States locations. Incorporating an international weather service will allow us to support users in different locations. With this support, we can then change default temperature units based on locale.
- **User Input:** The NWS API takes in coordinates to fetch weather data for that location, so we resorted to browser geolocation. Future implementations can allow for users to input city names, which can then be mapped to coordinates to fetch the data.