Below is the basic flow of the project:

● Based on the input data in the search form, construct a web service URL to retrieve the XML-formatted output from the Google Geocode API.
● If the current location checkbox is selected, the location details must be obtained from IP-API
● Parse the returned XML and extract the latitude and longitude values.
● Call the Forecast.io API (latitude and longitude are parameters in the web service URL) and retrieve the JSON-formatted output.
● Parse the returned JSON-formatted output and extract the weather information.
● Display the weather information in tabular format.

The project needs API keys for using **Google Geocode API** and **DarkSky API**.

Getting DarkSky API key:

- Go to https://darksky.net/dev

- Click on “Try For Free”.

- Fill out the form to get the key

- Copy this key and use it in the code of this project in place of <DARKSKY_API_KEY>

Getting Google API key:

- Go to the Google Developers Console (https://console.developers.google.com/flows/enableapi?apiid=geocoding_backend&keyType=SERVER_SIDE&reusekey=true)

- Create or select a project

- Click Continue to Enable the API

- Go to Credentials to get a Server key (and set the API Credentials)

- Copy this key and use it in the code of this project in place of <GOOGLE_API_KEY>

The initial webpage contains 2 important buttons:

1. **Search:** This button validates whether the user provided values for street address, city and state. The validation is implemented in JavaScript. If the user did not enter one of the data items, then an alert should be shown with an appropriate message prompting the user to provide complete information. A PHP script sends the data to Google Geocode and forecast.io in sequence
2. **Clear:** This button must clear the result area, all text fields, unselect the State value. The Clear operation is done using a JavaScript function.

Google Geocode API:

The PHP script uses the address information to construct a web service URL to query Google Geocode API:

https://maps.googleapis.com/maps/api/geocode/xml?address=[STREET,CITY,STATE]&key=[GOOGLE_API_KEY]

The API returns results in XML format. The latitude and longitude are extracted from this data and used further to query the forecast.io API

Fetching current location from IP-API:

The latitude and longitude details must be fetched from the API call made to:

http://ip-api.com/json

Forecast.io API:

This web service URL is constructed as follows:

https://api.forecast.io/forecast/[DARKSKY_API_KEY]/[LATITUDE,LONGITUDE]?exclude=minutely,hourly,alerts,flags

Here LATITUDE and LONGITUDE corresponds to the values extracted from Google Geocode API. The response is in JSON format. Required details are obtained from the parsed JSON file.

On click of the search button, a card with detailed current weather will be displayed. A table providing summary of weather for the next 7 days will be displayed below the card. On clicking on this summary, a more detailed weather for that day will be displayed in card format along with a 24-hour temperature chart.

Detailed Weather:

The detailed weather is obtained by making an API call to forecast.io with the below URL:

https://api.darksky.net/forecast/[DARKSKY_API_KEY]/[LATITUDE],[LONGITUDE],[TIME]?exclude=minutely

Here the TIME parameter is from the 'data' object selected from the table which is part of the 'daily' object from the 1st API call.

Google Charts:

The chart for hourly temperature is displayed using Google Charts:
https://developers.google.com/chart/interactive/docs/gallery/linechart

Icons:

The various icons used in the project are from https://www.iconfinder.com/

Note:

The project can be hosted on any cloud platform. The Google App Engine application makes it easier to run and test the application on localhost.