

Final Project Worth (20%)

General

- The project due date is **August 15th, 11:59pm**. Please be sure you do not wait until the last minute to finish/submit it.
 - **NO LATE SUBMISSIONS ARE ACCEPTED as it's going to be the last week. If not submitted the project grade is zero.**
 - There are two options for the final project: creating a weather App or creating your own idea. Check the details in the following paragraphs
 - You will be required to present your project during week 14's session. More details about the presentation will be announced later
-

Deliverables

The key **deliverables** will be ...

- **MS Word Document** uploaded to *Final Project* drop box by due date indicated that includes...
 - Group Name
 - Names of team members
 - Information to connect/access - URL (on Sheridan Lamp server), user name, password, etc.
 - The URL of the weather API you used in your project.
 - **A zip file contains all the files of your project**
 - If you choose to implement your own idea, you need to explain the purpose of your App and the functionalities it provides
-

Final Project Option #1: Weather App

Description

In this final project, you will use the skills you learned in class to build a realistic weather App. The App should be able to check the weather at the user's location or at other cities. It should provide information about the current/forecast weather conditions and visual charts to show the weather changes.

Please read **carefully** the following requirements and make sure to include all of them in your project. Please send me an email if anything seems not clear to you.

Weather data

Your App should pull dynamic and realistic weather data from the web to provide to the user. A lot of free weather services are available on the web. They provide weather data in Json format. In order to use the API, you should usually register and get a key. You can use one of the following APIs:

- <http://www.openweathermap.org/api>
- <https://darksky.net/dev/>

Otherwise, you can search any other API on the web. Check this page as an example:

- <https://superdevresources.com/weather-forecast-api-for-developing-apps/>

App Features

The weather App should provide the following information:

- Current weather conditions: this information should include (at least):
 - o the temperature,
 - o a short description about the weather,
 - o an icon that represents the current weather conditions (cloud, sun, etc.)
- Weather forecast: that could be for one or multiple days (according to your API options)
- Graphic displaying of weather data (charts): The App should show a chart showing temperature changes over a period of time:
 - o that could be for the same day (e.g. maximum, minimum, morning, evening, etc.) or the forecast temperatures (today, tomorrow, after two days, etc.)
- By default, the App shows the current weather conditions at the user's current location (using HTML geolocation API). The user can also check the weather in different locations:
 - o Your app should provide a predefined list of cities (of your choice) that the user can chose from. Use data file (xml or JSON) to save that list and load it using Ajax call
 - o The user can also search a city by typing it. Make sure to choose an API that provides search by name feature.

Misc.

- Show error message if the API doesn't reply to your Ajax call.
 - Show an error message if you don't get an answer from the API for a custom user's search.
 - Save the user's searches using HTML localStorage. Show the saved cites to the user under the list of the predefined cities in a category called (recent locations)
-

Option #2: Your Own Project Idea

We can come with your own project idea. You need to discuss with me and get the approval before you start it (**should be done by week 11**)

Your project should provide the following requirements (minimum)

- It should be mobile friendly (use jQuery Mobile or any other mobile library)
- Read raw content from an API or from a Json file that you create on the server and use it to build the interface (or the main part of it)
- Has at least 2 pages
- Use LocalStorage to load and save user input (e.g. load/save user's search)
- Provide different presentations of remote data (textual, icons, charts)
- Show appropriate error message (in case of errors)

Submission:

- **Publish** the project to your web space (be sure to include all necessary files/folders)
 - **Upload MS word document in the Project Drobox folder.**
 - **Zip and upload all files** to the Project Dropbox in SLATE by the Due Date indicated and **include the URL** to your file in the **Comment** section
 - **URL must be included** or project will not be graded
-

Marking Rubric

Feature Name	Description	Value
General		
Project Submission	Project is submitted as described in the submission section. MS Word document provides the required information.	2
Overall UI	Overall usability, richness, attractiveness of UI on all pages. <ul style="list-style-type: none"> If you use multiple pages: make sure that the design of the pages should be consistent with each other with ease of navigation between pages, consistency in design of all pages, etc. If you use single page architecture: make sure that all the information is nicely represented in the page. (e.g. you could use collapsible components to not overload the interface) 	3
Code/Script Quality	Appropriate use of jQuery to pull remote data from JSON file. Do your best to avoid mixing JS and HTML (e.g. use Handlebar templates to generate HTML.)	3
Functionality	App functionalities provided as described	12

Functionality breakdown for Weather App		
Current weather conditions	App shows current temperature, a short description about the weather, an icon that represents the current weather conditions (cloud, sun, etc.)	3
Weather forecast	App shows some weather forecasts (one or multiple days)	2
Graphic displaying of weather data (charts)	App includes a chart showing temperature changes over a period of time (same day, forecast or both)	3
Appropriate error message	<ul style="list-style-type: none"> Show error message if the API doesn't reply to your Ajax call. Show an error message if you don't get an answer from the API for a custom user's search. 	2
Save user's searches	Use HTML5 localStorage to save the locations that the user searched. Save only the cities/locations that your App can found (e.g. when it gets an answer from the API)	2