

Project 3 Report

Matriculation numbers: 110011492, 150023775, 150022101, 150020423

This report, apart from section 7, was a group effort.

1.Introduction

This web application allows the user to look up weather information of cities supported by OpenWeatherMap. It provides current weather and the next 5 day forecast in 3-hour intervals. The information are presented in both text and graphic forms. In addition, the webpage displays summary of previously search results.

2.Setup

This application runs on nodejs server. The Node packages used are listed in package.json. All packages can be installed by changing to the root directory of the application and then running 'npm i' command.

Before using initServer.js and dataMashupsServer.js, replace URLs used for connecting to CouchDB with valid ones.

The initServer.js deletes existing database with the same name before creating a database called 'weather' which contains two documents for storing current and forecast weather respectively.

The server can be started by running dataMashupsServer.js. This will start a Secure HTTP server that runs through Port 8082 alongside the regular server through Port 8081. The https server provides a self-signed certificate which is stored in ssl folder with the private key file.

The webpage can then be accessed from <http://username.host.cs.st-andrews.c.uk:8081> or <https://username.host.cs.st-andrews.ac.uk:8082>. Google Chrome 50 and newer will automatically decline geolocation request from unsecured origins. Therefore, to use location services in these browsers, the user should use Port 8082. *When* connecting to server through https protocol, the user may have to disregard security warnings from the browser in order to proceed.

3.Design & Implementation

3.1.Database:

The database is kept consistent with the data from OpenWeatherApi. Upon initialisation with `initServer.js`, a document named 'weather' is created, or replaced if it has existed. The document is then split into two parts to provide a clean separation of the data of current weather and forecast weather. The database is then populated with data of a few cities. We do this by making API calls and storing the data in the correct part of the document. Extra information is added to each piece of data which converts the GMT time to the time local to the location.

3.2.Webpage:

We opted for a simple design for the interface to optimise the usability of the web app. Upon loading the page, the browser will ask the user for permission to use current location. If permission is granted, weather information for the detected location will be displayed; alternatively, the user can search manually with the search box.

3.2.1.Search

Content/data/citylist.js contains a list of more than 70000 city details with coordinates, which is assigned to variable ***cities***.

The function ***initCityAutoComplete*** is called when the user opted out for using the current location. The function uses jQuery UI feature 'autocomplete' to display the city names. ***module.common.getCityNames*** is used to filter and format the city names.

On click of select button function ***search*** is called with the selected city name as attribute. The location ID is extracted from the selected city name using function ***module.common.getCityDetails***. It is then posted to the server, where we make API calls to the OpenWeatherPaps from the server. This means that we do not have to post multiple piece of information from the client side to the database. We then retrieve the required from the database by using GET functions. The webpage is then updated with the information of the selected city. Certain parts of the page is hidden or showed depending on the user's selected view. Three buttons are available for the user to view the current weather, forecast weather and the weathers on a map, respectively.

Additionally a list of previously searched cities are displayed. These results correspond to the marked locations on the map in Weather Map section. In Weather Map section, clicking on a

city name from the list will centre the view to its position on the map. Deleting the record removes it from the map and the folding list.

The 'Route to' link will display a planned route from the user's current location (if supplied) to that city. On recent versions of iOS and OS X platforms, the Maps application will be launched to display the results; on other platforms, the web based Google Maps will be used.

3.2.2.Current Weather

The current weather gives a simple design of information display with relevant information in blocks and table, including a thermometer diagram and weather icon. The thermometer diagram is drawn manually using D3. The length of the alcohol illustration in the thermometer varies depending on the temperature data.

3.2.3.Forecast weather:

3.2.3.1.Forecast weather charts

The charts visualisations are implemented using D3.js. Once we obtained the data from the database, the data is then compiled in the right format for the use of the charts and tables. To reduce clutter, the information is shown to the user only on hover of the nodes or bars. The axes are formatted based on the domains of the data used for the charts. The points (circles or triangles) are added for each piece of data and are positioned with respect to the chart space. Lines path are also added to connect the points. As for the bar chart, the length and position is adjusted using the data. The colours of the points, triangles and bars are different for each day of the weather forecast.

The snow and precipitation are plotted as simple bar charts. The pressure, humidity and cloud is plotted as a simple line graph with interactive points. The wind data has triangles as points, which gives the direction of the wind. The line graph of the temperature is augmented with small triangles to also indicate the minimum and maximum temperature.

When the user mouse over a point, bar or icon, a function is called to changed the its feature. Besides that, we add a new div to give more information about that point. The orientations of the wind triangles are done by using the translation and rotation under the 'transform' function of D3.

The initial view under the forecast weather tab is the a summary chart and a table of weather details. The summary chart is a broad overview of data compilation. This includes the cloud coverage, temperature, weather, pressure, precipitation, wind speed and direction. The lines on top of the chart gives an illusion of the cloud coverage. The temperature line is appended with the weather icon. The icons, bars and the wind triangles are interactive: a label will appear

and the opacity and/or size will change on hover. The lengths of the wind triangles and the direction where the triangles are pointing in the summary chart are consistent with the wind speed and direction respectively.

3.2.3.2.Forecast weather table

At the bottom of the chart, the entire weather forecast detail is displayed in a tabular form with date, time (formatted using *DateFormat*) and weather icons.

Function ***displayForecastData*** is called to render the forecast table. The function loops through the forecastWeather json list and generates new table row element for each new forecast entry.

All the major functions are in *dataMashups.js*.

3.2.4.Map

3.2.4.1.Main design of map

Our map can show all selected cities on the map. We use leaflet.js for the map. When users click the “show selected Cities on the map”, the program will update (recall) all cities’ information in the database and show the information on the map.

We mainly use the mapIni function. Firstly, we remove the previous map DIV (if applicable) and create a new map DIV, so users can see the changes later (such as adding or deleting some new cities) on the map. Secondly, we set map’s centre by using map.fitBounds (center) and we also set the style of the map by using Stamen_Watercolor.addTo(map). Next, we can go drawmap function to draw some point and popup on the map.

In the drawmap function, firstly, we set the Icon for each point and add it to map as there is an inherent icon for each point. Secondly, we are going to draw the popup, which contains basic weather information. When users’ mouse over or out the point, the popup will appear or disappear respectively. There is also design to intentionally increase the effective area of the icon in which a cursor hovering can be detected. (`marker.bindPopup(url, {'offset': L.point(0,-10)}).openPopup();`)

3.2.4.2 Other functions related to the map

We also add 2 functions relate to the map. The first is that when users click the city on the accordion list above the map, the user will see the map’s center become to this city immediately. The second function is that when users click the red delete button, users will not only delete the city name on the accordion list but also delete the city point on the map.

For the first function, we add id for each element of the accordion list. When users click the element, a function that gets the selected city’s longitude and latitude will be activated. Next,

the program set the longitude and latitude as a center, run mapIni again. The map's center will be this city.

For the delete function, we add a delete button for each element of the accordion list. When users click the delete button, a function will send city id request to the backend and delete the id in the database in the backend. After that, we reload the accordion list and map asynchronously, so users can see the result immediately without reloading the whole page.

3.2.5. Record manipulation

The Delete button was responsible for removing each record from the database. For the sake of simplicity, these buttons were link elements styled to look like buttons and would send GET request to the server with the ID of that city. However, later it became clear that the solution would direct the browser to a new page, and if the server doesn't send a response, the communication wouldn't be terminated and the browser would be in a loading state until timeout. In view of this problem, the buttons were changed to proper button type input elements that invokes the `DeleteRowFunction` on the client side, which sends XMLHttpRequests in the convention way so server responses could be handled properly.

Once records in the database has been updated, the map and accordion on the page are refreshed to reflect the change.

3.5.6. Local times

Once we made a call from the server side and saved it to the database, we run a function to add information regarding the local times. We do this by using Google Maps Time Zone API. We make an API call with the coordinates of the city to obtain the time zone of the location. The time is then adjusted and added to the database.

5. Organisation/Communication

All team members were added to a group on Office 365. Information such as API keys, database address and links to useful materials was shared in a OneNote notebook, other files outside the application were shared on OneDrive.

For collaboration, we used Git hosted on Bitbucket. A basic setup with client and server side script, an HTML document and a database was created at the beginning upon which functionalities were added later. We then created a group project on Bitbucket and pushed all files into the single repository under that project.


Features	110011492	150023775	150022101	150020423
Client side	Basic functionalities			
Visualisation	All charts, current weather display			Forecast table
Map			All functionality	
Location service		Supply user coordinates, notify API status		
Search bar				Auto-complete
Server side	API calls using location ID and coordinates	Get all current data	Updating all current data and delete button server side	
Accordion		List database records, send ID to server, link to show navigation	Delete button with map	
Page/feature styling		Page width, button appearance		
Secure HTTP server		Enable https connection		
Wind direction		Degree to common expression		

Changes to code were mainly explained using commit messages.

6. Testing

6.1. Functionality

Browser security warnings when using secure HTTP connection:




Your connection is not private

Attackers might be trying to steal your information from **.host.cs.st-andrews.ac.uk** (for example, passwords, messages, or credit cards). `NET::ERR_CERT_AUTHORITY_INVALID`

HIDE ADVANCED Back to safety

This server could not prove that it is **.host.cs.st-andrews.ac.uk**; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

[Proceed to .host.cs.st-andrews.ac.uk \(unsafe\)](#)



This Connection is Untrusted

You have asked Firefox to connect securely to **.host.cs.st-andrews.ac.uk:8082**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

Technical Details

I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Add Exception...

Browser requesting location information:

Please accept location request to display your local weather

Current Weather

Search a location:
Type in a city name


https:// .host.cs.st-andrews.ac.uk:8082 wants to:

Know your location

Allow Block select

Accept location request to disp... x +

.host.cs.st-andrews.ac.uk:8081



Would you like to share your location with .host.cs.st-andrews.ac.uk?

[Learn more...](#)

Share Location v

Search a location:
Type in a city name

The page on load when location access is denied:

Location Access Denied

Current Weather

Search a location:

Type in a city name

select

The autocomplete feature of the search bar:

Current Weather

Search a location:

dundee

Dundee, GB

Dundee, ZA

Dundee, US

East Dundee, US

West Dundee, US

select

The page when a location is selected:

Current Weather

Search a location:

Dundee, GB

select

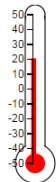
Current weather

Forecast

Show selected Cities on the map

Dundee,GB

[56.5°, -2.97°]



22°C

min temp: 16°C ; max temp: 26°C

Clouds



few clouds

retrieved at: Fri May 20 2016 13:00:00 GMT+0100 (GMT Daylight Time)

Clouds

20 %

Wind Speed

9.8 m/s

Wind Direction

West 250° (meteorological)

Humidity

55 %

Pressure

1006 hPa

Sunrise Time

04:45

Sunset Time

21:32

▼ Beijing, CN

27°C



clear sky

Delete

Route to

► Saint Andrews, GB

Part of the page when forecast weather is selected:

Weather Forecast

Search a location:

Dundee, GB

select

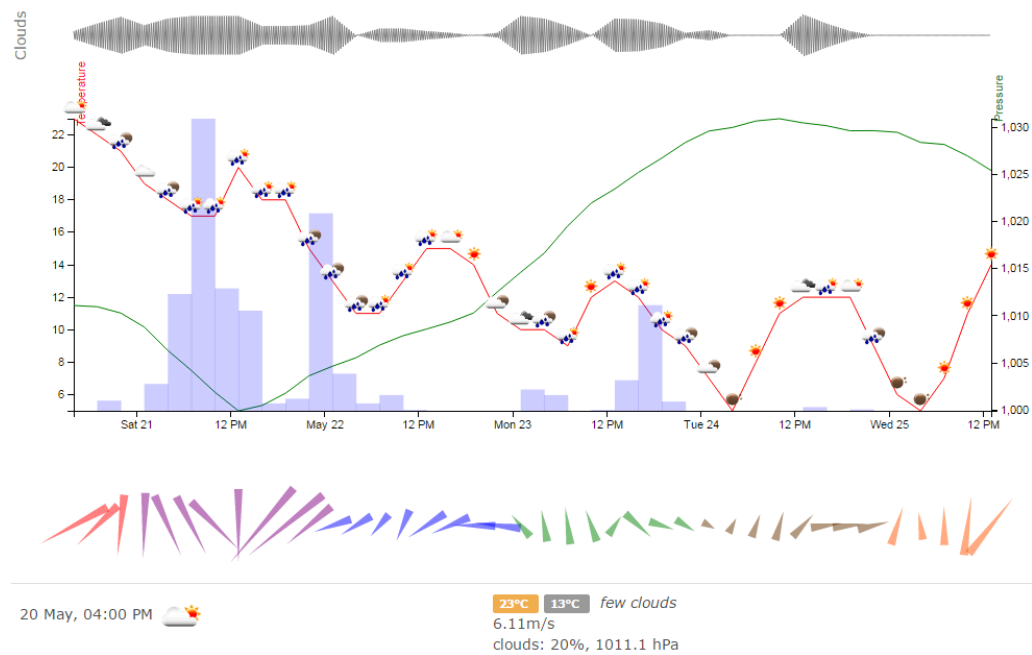
Current weather

Forecast

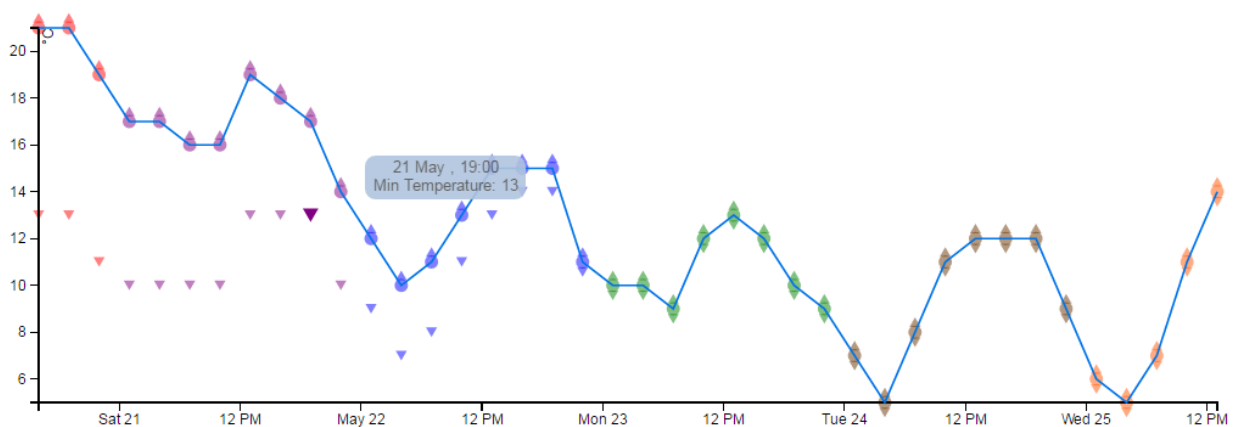
Show selected Cities on the map

Choose a data:

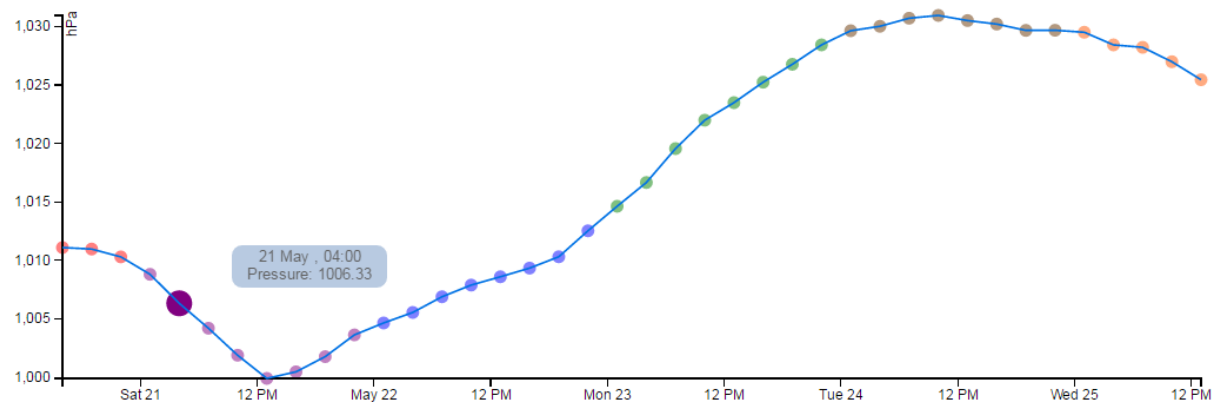
Summary



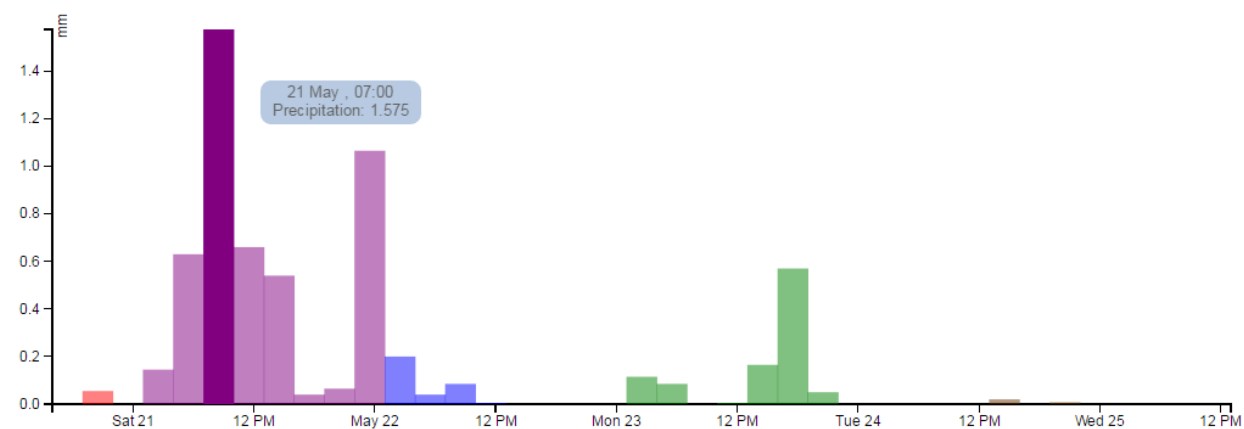
The temperature line graph:



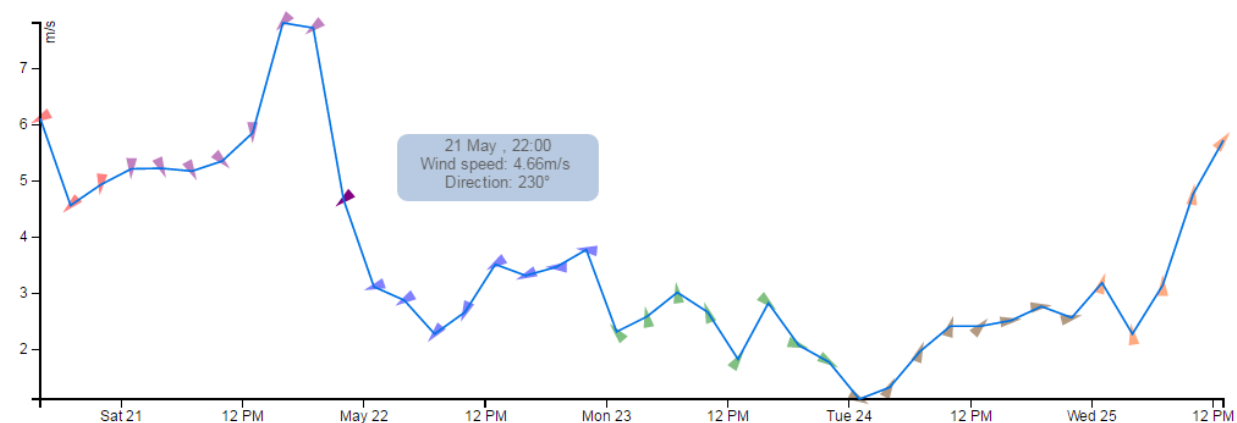
The pressure line graph: (the humidity and clouds charts are similar)







The precipitation bar chart:



The wind chart:



Forecast weather table:

20 May, 06:30 AM		25°C 25°C light rain 0.87m/s clouds: 76%, 983.96 hPa
20 May, 09:30 AM		27°C 27°C light rain 1.51m/s clouds: 80%, 982.67 hPa
20 May, 12:30 PM		27°C 27°C light rain 1.26m/s clouds: 92%, 983.03 hPa
20 May, 03:30 PM		25°C 25°C light rain 0.96m/s clouds: 92%, 984.47 hPa

The map's main interface:

Weather Map

Search a location:

Dundee, GB

select

Current weather Forecast Show selected Cities on the map

Beijing, CN

24°C ☀️ clear sky

Delete

Route to

London, GB

Dundee, GB

New York, US



Leaflet | Map tiles by Stamen Design, CC BY 3.0 — Map data © OpenStreetMap

Point's popup:

Weather Map

Search a location:

Dundee, GB

select

Current weather

Forecast

Show selected Cities on the map

Beijing, CN

London, GB

Dundee, GB

23°C scattered clouds

Delete

Route to

New York, US



Clicking the city of the accordion list will change the map center:

Weather Map

Search a location:

Dundee, GB

select

Current weather

Forecast

Show selected Cities on the map

Beijing, CN

London, GB

Dundee, GB

New York, US

17°C few clouds

Delete

Route to



Clicking the delete button will delete both element of the accordion list and the point on the map:

Weather Map

Search a location:

Dundee, GB

select

Current weather Forecast Show selected Cities on the map

London, GB

19°C scattered clouds

Delete

Route to

Dundee, GB



6.2. Usability

In addition to our own testing in development, a student which has no relations to the module is called upon to test our web app. It is noted that the axis label of the summary chart is hidden behind the axis ticks labels. Besides that, some labels of the charts are not visible due to having too small a margin. A comment is made on the colours of the tabs and no notification of which tab we are currently on. The student finds the web page easy to use and the visualisations interpretations intuitive.

In response to the comments, the axis labels are shifted, the margins of the charts are increased and the page title is updated on click of a tab, indicating which tabs the user is currently in.

7. Contributions

In this assignment, I mainly focus on the map and related functions. I cooperate with another teammate to finish the accordion list. I also learned how to use Bitbucket and remember some hard remembered code of Bitbucket. The most important thing is that I understand how to

cooperate with people in the bigger group (4 or 5 people). I will describe what I learn according to problems encountered, and what I do.

Bitbucket is easy to use, despite some hard remembered code. For us, if the code is pushed immediately, there is not conflict and little conflicts. However, because of exams, we do not contact with each other for 10 days. As a result, we spent 5 hours just to resolve the conflict and merge each branch to the master branch. If we need to use Bitbucket in the future, we will need to merge and resolve the conflicts immediately.

I also complete the map by using leaflet.js, and finish some details such as popup. I also finish two functions of maps: clicking the delete button will delete both elements of the accordion list and point on the map. Clicking the element (city) of the accordion list will change the centre of the map to the city. I finish those function from both backend and frontend. I mentioned the algorithm in the design of map part.

8. Extensions

The extensions are:

- Map integration using Leaflet
- Convert times to local times using Google Map Time Zone API
- Geo location API to get local coordinates
- Accordion feature: centre map on click, remove from map on deletion button
- Supports secure HTTP server