Assessment 3

Advanced Project

CSE2WDX Web Development

Before you begin

Objectives

This is an individual assignment. Students are not permitted to work in a group when completing the quizzes.

Copying and Plagiarism

This is an individual assignment. Students are not permitted to work in a group when completing the quizzes. Plagiarism is the submission of another person's work in a manner that gives the impression that the work is their own. La Trobe University treats plagiarism seriously. When detected, penalties are strictly imposed.

Further information can be found on https://www.latrobe.edu.au/students/admin/academic-integrity

Submission Guidelines

Your assignment submission should be typed, not written/drawn by hand.

Submit the electronic copy of your assignment through the subject Learning Portal.

Submission after the deadline will incur a penalty of 5% of the available assignment mark per day capped at 5 days. No assignment will be accepted after 5 days. If you have encountered difficulties that lead to late submission or no submission, you should apply for special consideration.

Contents

Advanced Project 1
CSE2WDX Web Development 1
Before you begin 2
Objectives 2
Copying and Plagiarism 2
Submission Guidelines 2
Background 4
Tasks 1-7: Objectives report 4
Tasks 8-13: Implementation 4
Task 8: Geolocation component 5
Task 9: Forecast component 5
Task 10: Line chart component 7
Task 11: App content 7
Task 12: App presentation 7
Task 13: App logic 7
Task 14: Coding conventions 7
Task 15: Organisation 8
Submission 8
Assessment marking criteria [total 77 marks] 9
Appendix A: Page design before OpenWeatherMap request 12
Appendix B: Page design after OpenWeatherMap request 13

Background

In this assignment you will create a web page that behaves as an interactive weather tool for your location.

The web page will first detect and report your location via the geolocation API. Expect that this functionality will run over localhost only as geolocation is considered to be insecure over HTTP. You are not expected to host this page and/or pay for HTTPS.

Once the page has your location, it will use those coordinates to send a request for the weather data at that location to the OpenWeatherMap website. Specifically, you will use the "5 day weather forecast API". You can use this service by signing up for a free account and obtaining an API key.

The weather data received will be for 4 days of weather data at 3-hourly intervals making a total of 32 readings. The data will then be displayed in an attractive fashion in two formats: (1) a grid summary, and (2) a line graph produced with an HTML Canvas.

A preview of the expected interface is given in Appendices A and B.

Tasks 1-7: Objectives report

You have some freedom in how you design the interface for the page. However, a critical aspect of this assignment will be to prepare a **report** showing how parts of your solution map to the certification exam objectives. Marks for Tasks 1-7 are tied to completing the report and code together as a pair. You will not earn marks for Tasks 1-7 if you do the report without the code or vice versa.

An example report and a template have been provided. You may try the technical tasks suggested in the example as written, modify them, or choose alternatives from the certification exam objectives.

Marks are awarded for the following certification objectives and they must have **two examples** each:

- A.2: Write code that interacts with UI controls
- A.4: Implement HTML5 APIs
- B.3: Implement exception handling
- C.1: Validate user input by using HTML5 elements
- C.2: Validate user input by using JavaScript
- C.3: Consume data
- C.4: Serialise, deserialise and transmit data

These objectives all come from weeks 6-9 of the subject.

Tasks 8-13: Implementation

Regarding the implementation tasks, the work must go in the following files:

	weather.html	weather.css	geolocation.js	weather.js
Task 8: Geolocation	✓	✓	✓	
Task 9: Forecast	✓	✓		✓
Task 10: Line chart	✓	✓		✓
Task 11: HTML	✓			
Task 12: CSS		✓		
Task 13: JavaScript			✓	✓

Details for the implementation tasks are as follows.

Task 8: Geolocation component

The web page will first detect and report your location via the geolocation API. The results (latitude and longitude) will be injected into a form that will be used as the starting point for Tasks 9 and 10 later.

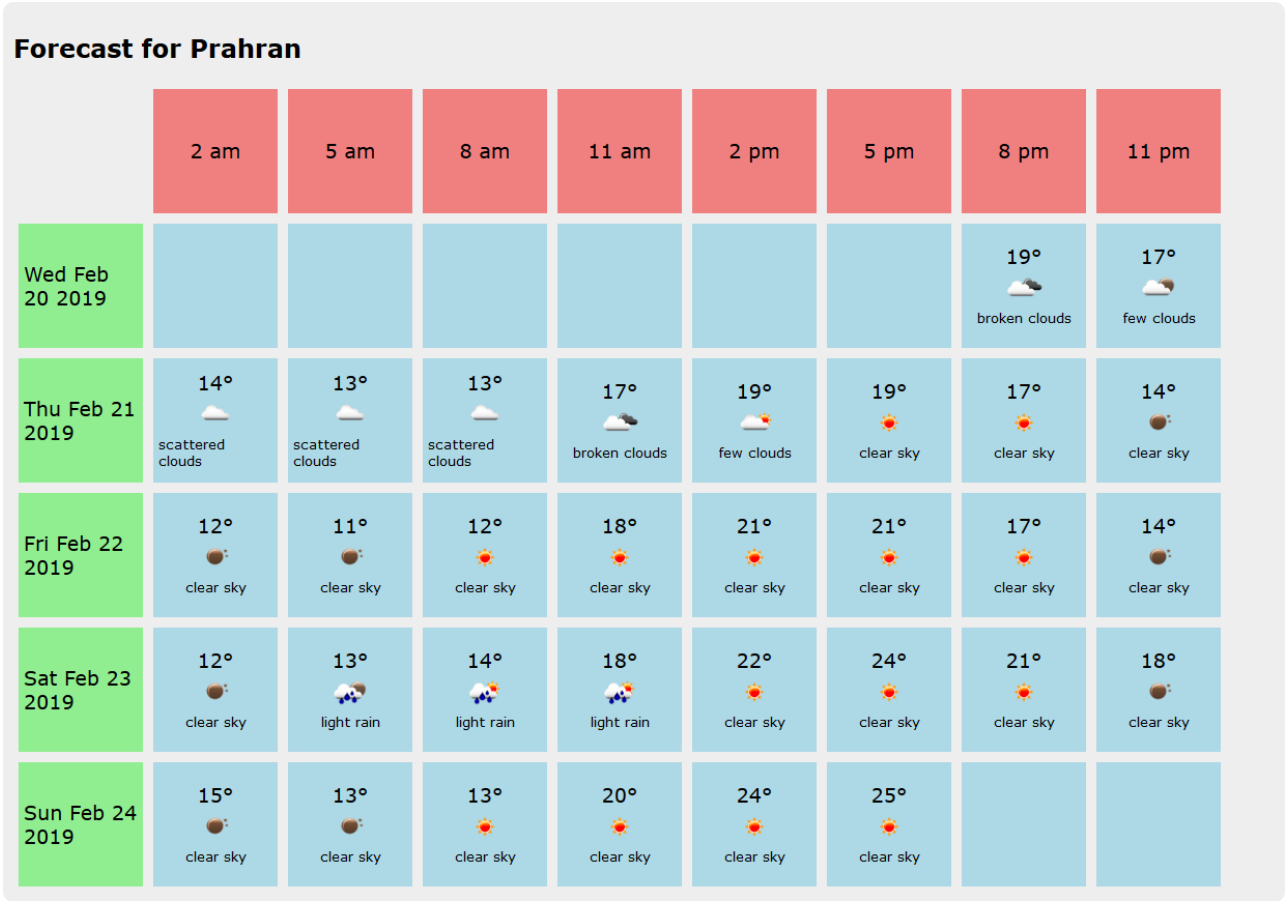
The geolocation request and response will be processed *immediately* on page launch without requiring a button click or similar. To be clear, the “Get Weather” button is used for Tasks 9 and 10 only.

A suggested design is shown here:

Task 9: Forecast component

The weather forecast will be for 4 days with 8 readings per day at 3 hourly intervals. This is a total of 32 readings to be organised in a grid. The data comes from the OpenWeatherMap 5 day weather forecast API. Note that this API only supplies forecasts from the current time onwards, so expect the need to report the weather forecast for a part-day at the start and end. The request is launched with the “Get Weather” button (Task 8).

A suggested design is shown here:



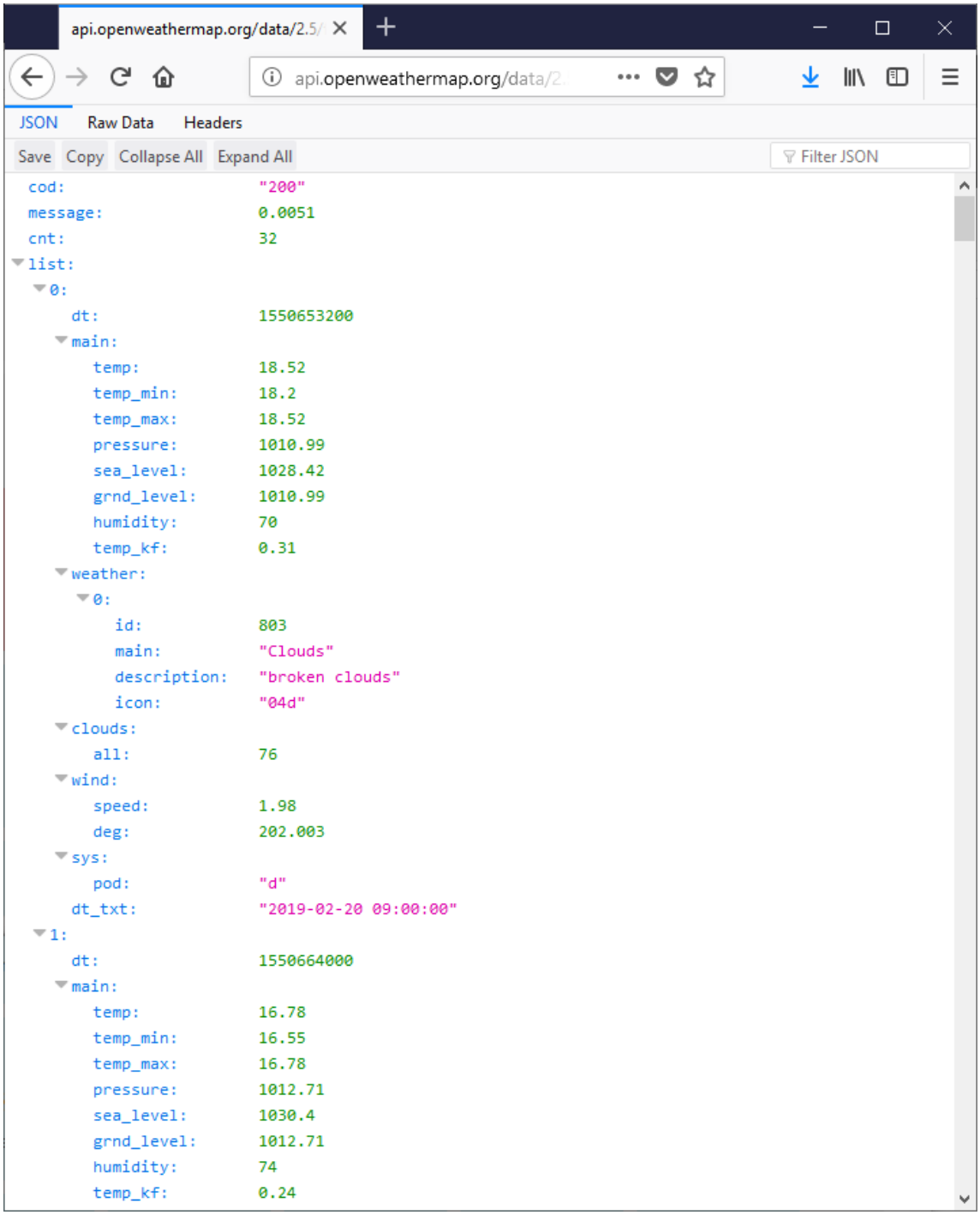
	2 am	5 am	8 am	11 am	2 pm	5 pm	8 pm	11 pm
Wed Feb 20 2019							19° broken clouds	17° few clouds
Thu Feb 21 2019	14° scattered clouds	13° scattered clouds	13° scattered clouds	17° broken clouds	19° few clouds	19° clear sky	17° clear sky	14° clear sky
Fri Feb 22 2019	12° clear sky	11° clear sky	12° clear sky	18° clear sky	21° clear sky	21° clear sky	17° clear sky	14° clear sky
Sat Feb 23 2019	12° clear sky	13° light rain	14° light rain	18° light rain	22° clear sky	24° clear sky	21° clear sky	18° clear sky
Sun Feb 24 2019	15° clear sky	13° clear sky	13° clear sky	20° clear sky	24° clear sky	25° clear sky		

Here is an example of the URL that you would submit as an AJAX request to source the above data:

http://api.openweathermap.org/data/2.5/forecast?lat=-37.850234&lon=144.977716&cnt=32&units=metric&APPID=XXX

… where XXX is the APPID that you get when you sign up to a free OpenWeatherMap account.

If you try submitting this URL to your browser rather than as part of an AJAX request, you should get a raw JSON response like below.



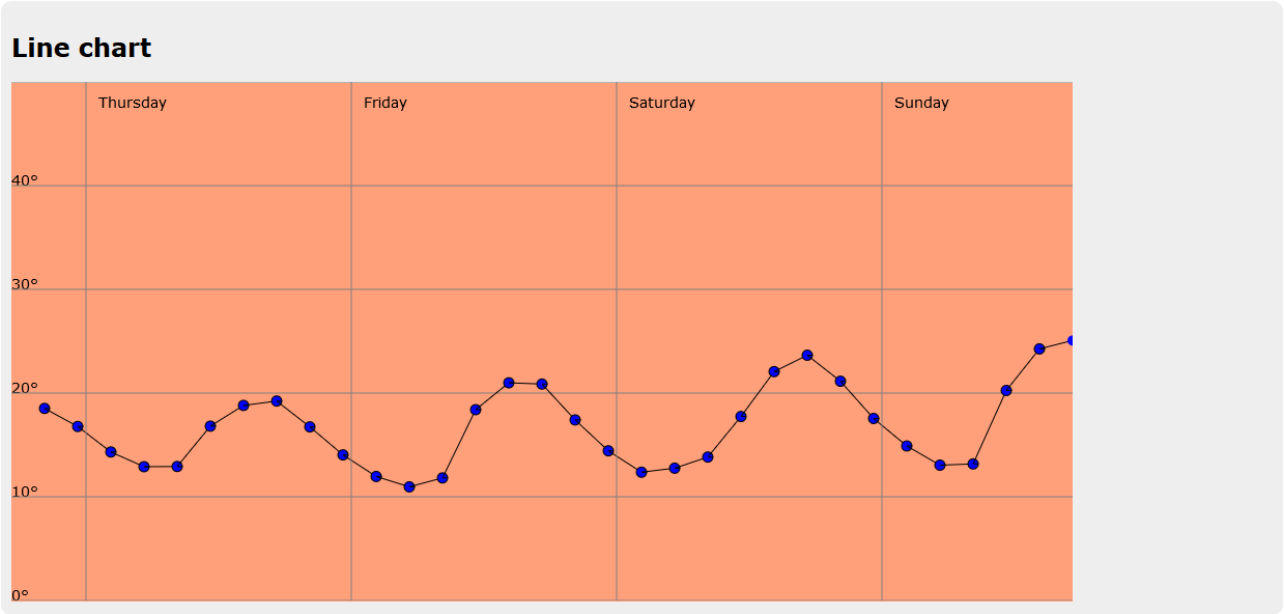
For example, the first temperature at timestamp 1550653200 is 18.52°. Then 16.78° at 1550664000. Etc.

Try retrieving a copy of the JSON data as a test like above before you start.

Task 10: Line chart component

The “Get Weather” button that launches the OpenWeatherMap request for Task 9 also feeds Task 10. For this task we want an HTML Canvas to plot the trend of the 32 temperatures as a line graph.

A suggested design is shown here:



Task 11: App content

Marks are awarded for your HTML for the content put into your page. This includes text, images, controls and the containers that hold them all. You should aim to create a well-organised page.

Task 12: App presentation

Marks are awarded for your CSS for the presentation of the page. This includes styling of the content and the layout. You should aim to create an attractive design. Refer to Appendices A and B for suggestions.

Task 13: App logic

Marks are awarded for your JavaScript/jQuery for the logic of the page. This includes logic based on sequence, selection and repetition structures, plus core concepts such as variables, functions, arrays, and event handling. You should aim to create a very functional page.

Task 14: Coding conventions

This task will award marks according to the quality of your code according to these aspects:

1. HTML, CSS and JavaScript indentation is consistent with one level of indentation per block.
2. The HTML, CSS and JavaScript files each use some comments (<!--comment -->, /* comment */ or // comment as appropriate, minimum of 3 per file type).
3. The W3C HTML markup validation service (https://validator.w3.org) reports zero errors and zero warnings on the code when tested with the “validate by direct input” method.
4. The W3C CSS markup validation service (https://jigsaw.w3.org/css-validator) reports zero errors and zero warnings on the code when tested with the “validate by direct input” method.
5. The Visual Studio Error List reports zero JavaScript errors and warnings.
6. The page footer includes hyperlinks to the W3C HTML and CSS markup validation services to indicate compliance to the W3C standards.

Extra: Although marks are not allocated, it is also good practice for HTML, CSS and JavaScript code to consistently use best-practice casing (lowercase for HTML/CSS and camelCase for JavaScript). (The DOCTYPE declaration is an exception.)

Task 15: Organisation

The following organisational aspects must be put in place for your solution:

(a) Programming language:

The web page will be a DHTML web page. DHTML refers to Dynamic HTML. DHTML is the combination of HTML5, CSS3 and JavaScript working together to create a web page that has a dynamic feel. Note that JavaScript may also include jQuery and is strongly encouraged.

(b) Third-party components:

A critical aspect of this assignment is to develop the coding skills by hand. Therefore, except for jQuery, third-party libraries are not permitted for this assignment (Bootstrap, etc.).

(c) Images:

Store the images in a subfolder called "images". Note that the web page cannot use copyrighted images. To avoid this problem, you are welcome to use images with an open licence. You can get these from free image websites or by performing an image search on your search engine with a "usage rights" filter set to "labelled for reuse" (or equivalent). Maintain a list of references in a file called references.txt in your solution with the name, author, location and access date of the images.

(d) Integrated development environment:

The website must be built using Visual Studio 2017.

(e) Template:

The website will be built from scratch beginning with an empty folder. Follow these steps:

a) Create a new folder called **xxx_cse2wdx_assignment3**, where xxx is your student ID.
b) Open Visual Studio 2017. Select "File" -> "Open" -> "Website…" and select the new folder.
c) Select "File" -> "Close Solution" immediately. This will prompt you to save a solution file (*.sln suffix). This must be saved in your new folder. Close Visual Studio 2017.
d) Double-click on the solution file to open Visual Studio 2017 and your solution again. Continue to use this method for opening your solution going forward for convenience.
e) You are now ready to go. Add your project assets next.

Submission

When you have completed, submit your answers via the link on the Learning Portal. You must submit the following two files:

a) Upload the objectives report to a file called **xxx_cse2wdx_assignment3.docx**.
b) Zip your Visual Studio 2017 solution folder to a file called **xxx_cse2wdx_assignment3.zip**. Take care to zip your solution folder rather than the contents of the folder to avoid a tarbomb effect. The following files will be expected in your folder:
    - HTML × 1: weather.html (Tasks 8-11).
    - CSS × 1: weather.css (Tasks 8-10 & 12).
    - JavaScript × 2: geolocation.js (Tasks 8 & 13) and weather.js (Tasks 9-10 & 13).
    - Subfolder: "images" subfolder with graphic assets (Task 15c).
    - Text file: references.txt for image references (Task 15c).
    - Solution file: xxx_cse2wdx_assignment3.sln.
    - Web.config: Visual Studio may generate this file, which you can leave alone.

… where xxx is your student ID. The two files must be submitted **individually**.

Note: The Learning Portal will impose a 10mb limit for uploading your files. You will need to make changes to your submission if you exceed this limit. A common reason for exceeding this limit is including unscaled photographs. Another common reason is including unnecessary assembly files in your Visual Studio solution. Refer to Task 17e on creating your Visual Studio 2017 solution to avoid unnecessary assemblies.

Assessment marking criteria [total 77 marks]

Task 1: A.2 - Write code that interacts with UI controls [4 marks]

- The report and code for example 1 are correct. (2 marks)
- The report and code for example 1 are partly correct. (1 mark)
- The report and code for example 1 are incorrect. (0 marks)
- The report and code for example 1 are not attempted. (0 marks)

- The report and code for example 2 are correct. (2 marks)
- The report and code for example 2 are partly correct. (1 mark)
- The report and code for example 2 are incorrect. (0 marks)
- The report and code for example 2 are not attempted. (0 marks)

Tasks 2-7: Other objectives [6 x 4 = 24 marks]

- (Marking is the same as Task 1.)

Task 8: Geolocation component [8 marks]

- Excellent implementation. The requirements were fully met meaning that the user's latitude and longitude is automatically and correctly loaded into the form when the page loads via localhost. (8 marks)
- Good implementation. The requirements were mostly met. (6 marks)
- Satisfactory implementation. The requirements were partly met. (4 marks)
- Below standard implementation. The requirements were mostly not met. (2 marks)
- Unsatisfactory implementation. The requirements were not met. (0 marks)
- No implementation. The requirements were not attempted. (0 marks)

Task 9: Forecast component [8 marks]

- Excellent implementation. The requirements were fully met meaning that the OpenWeatherMap weather forecast is populated in the grid for 32 weather readings when the "Get Weather" button is clicked. (8 marks)
- Good implementation. The requirements were mostly met. (6 marks)
- Satisfactory implementation. The requirements were partly met. (4 marks)
- Below standard implementation. The requirements were mostly not met. (2 marks)
- Unsatisfactory implementation. The requirements were not met. (0 marks)
- No implementation. The requirements were not attempted. (0 marks)

Task 10: Line chart component [8 marks]

- Excellent implementation. The requirements were fully met meaning that the 32 weather readings are plotted on an HTML Canvas as a line graph when the "Get Weather" button is clicked. (8 marks)
- Good implementation. The requirements were mostly met. (6 marks)
- Satisfactory implementation. The requirements were partly met. (4 marks)
- Below standard implementation. The requirements were mostly not met. (2 marks)
- Unsatisfactory implementation. The requirements were not met. (0 marks)
- No implementation. The requirements were not attempted. (0 marks)

Task 11: App content (weather.html) [4 marks]

- Page content is excellent. You have created a well-organised page with suitable choices for the text, images, controls and containers. (4 marks)
- Page content is good. There is room for minor improvement. (3 marks)
- Page content is satisfactory. There is room for moderate improvement. (2 marks)
- Page content is below standard. There is room for major improvement. (1 mark)
- Page content is unsatisfactory. There is a need for rework. (0 marks)
- Page content was not attempted. (0 marks)

Task 12: App presentation (weather.css) [4 marks]

- Page presentation is excellent. You have created an attractive page with styling for content and layout. (4 marks)
- Page presentation is good. There is room for minor improvement. (3 marks)
- Page presentation is satisfactory. There is room for moderate improvement. (2 marks)
- Page presentation is below standard. There is room for major improvement. (1 mark)
- Page presentation is unsatisfactory. There is a need for rework. (0 marks)
- Page presentation was not attempted. (0 marks)

Task 13: App logic (geolocation.js & weather.js) [4 marks]

- Page logic is excellent. You have created a functional page with logic based on sequence, selection and repetition structures, plus core concepts such as variables, functions, arrays, and event handling. (4 marks)
- Page logic is good. There is room for minor improvement. (3 marks)
- Page logic is satisfactory. There is room for moderate improvement. (2 marks)
- Page logic is below standard. There is room for major improvement. (1 mark)
- Page logic is unsatisfactory. There is a need for rework. (0 marks)
- Page logic was not attempted. (0 marks)

Task 14: Coding conventions [13 marks]

- Indentation is excellent. It was consistent with one level of indentation per block. (2 marks)
- Indentation is satisfactory. It has some inconsistencies that can be improved. (1 mark)
- Indentation is unsatisfactory. It needs attention to aid the readability of your code. (0 marks)
- Indentation was not attempted. (0 marks)

- HTML commenting was good with a minimum of 3 valid examples. (1 mark)
- HTML commenting needs improvement. (0 marks)
- HTML commenting was not attempted. (0 marks)

- CSS commenting was good with a minimum of 3 valid examples. (1 mark)
- CSS commenting needs improvement. (0 marks)
- CSS commenting was not attempted. (0 marks)

- JavaScript commenting was good with a minimum of 3 valid examples. (1 mark)
- JavaScript commenting needs improvement. (0 marks)
- JavaScript commenting was not attempted. (0 marks)

- The W3C HTML validator reported zero problems. (2 marks)
- The W3C HTML validator reported one or more warnings, but no errors. (1 mark)
- The W3C HTML validator reported one or more errors. (0 marks)
- The HTML code writing was not attempted. (0 marks)

- The W3C CSS validator reported zero problems. (2 marks)
- The W3C CSS validator reported one or more warnings, but no errors. (1 mark)
- The W3C CSS validator reported one or more errors. (0 marks)
- The CSS code writing was not attempted. (0 marks)

- The Visual Studio Error List reported zero JavaScript problems. (2 marks)
- The Visual Studio Error List reported one or more JavaScript warnings, but no errors. (1 mark)
- The Visual Studio Error List reported one or more JavaScript errors. (0 marks)
- The JavaScript code writing was not attempted. (0 marks)

- Your web page included a valid hyperlink to the W3C HTML validator to indicate compliance to its HTML standards. (1 mark)
- Your web page did not include a valid hyperlink to the W3C HTML validator. (0 marks)

- Your web page included a valid hyperlink to the W3C CSS validator to indicate compliance to its CSS standards. (1 mark)
- Your web page did not include a valid hyperlink to the W3C CSS validator. (0 marks)

Task 15: Organisation [0 marks]

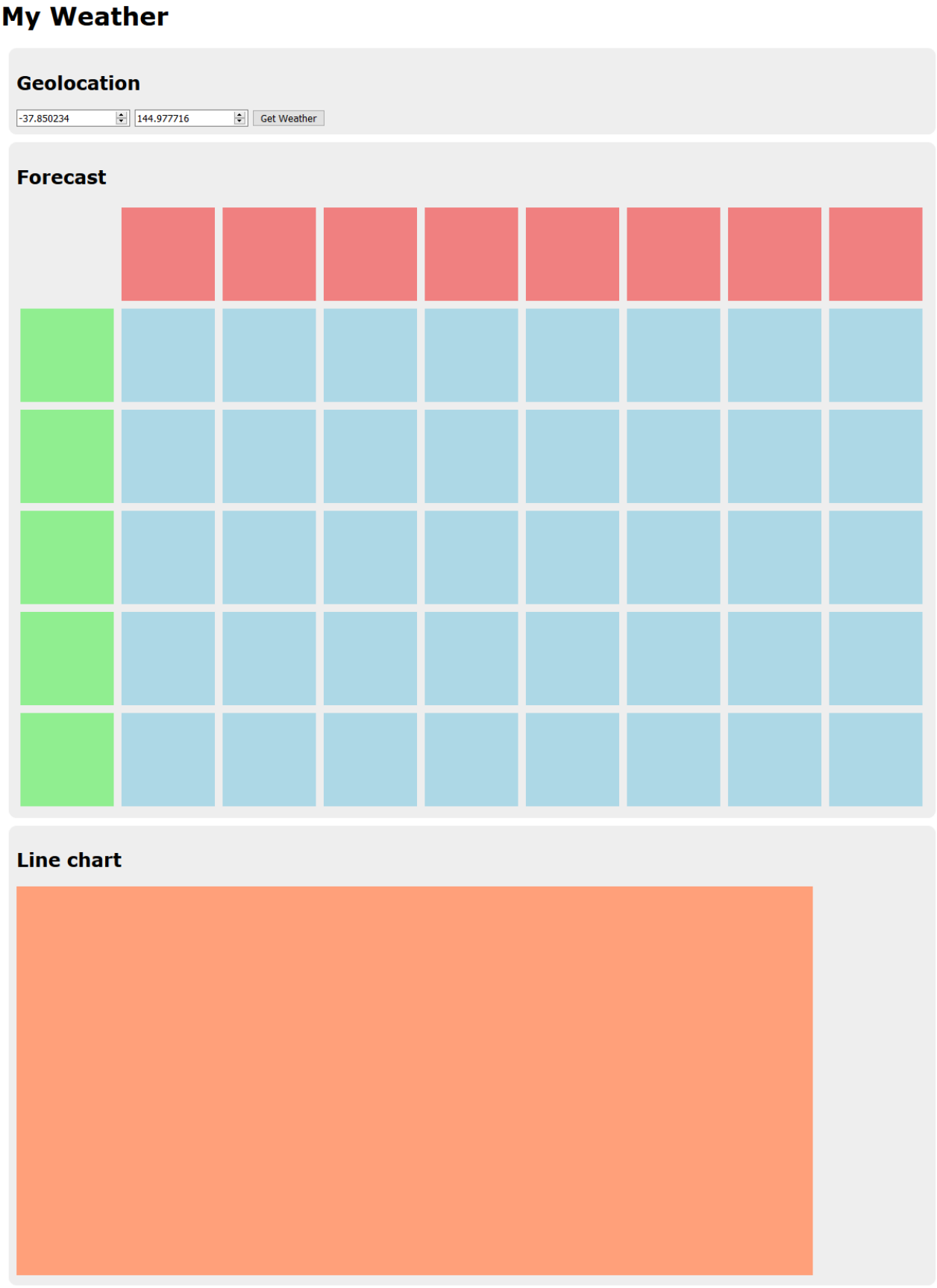
- Marks are not awarded for this task. Nevertheless, these are important starting points that are expected to be successful for the remainder of the assignment.

General:

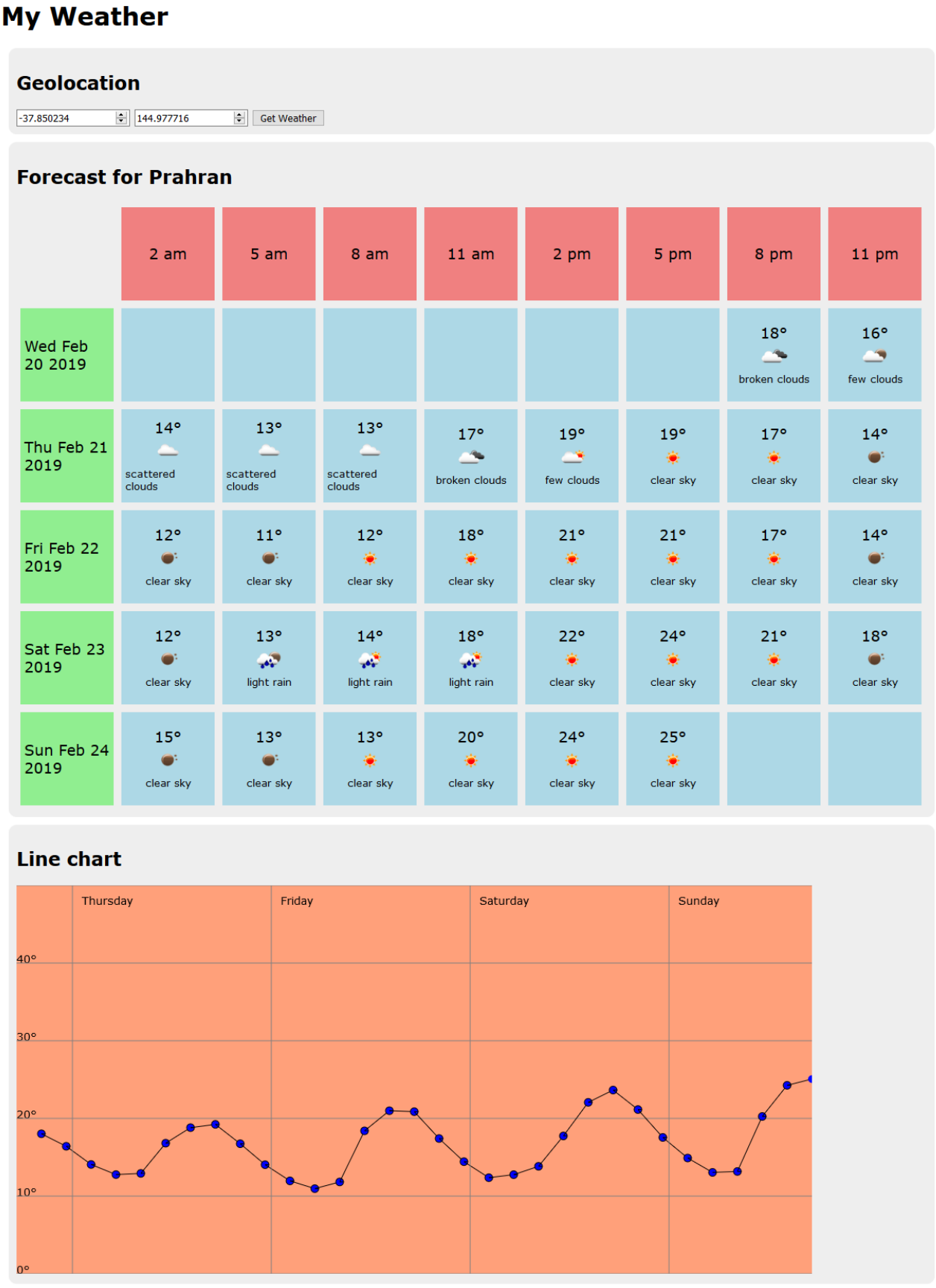
- Submission length is too short. This will be reflected in marks for individual criteria. (0 marks)
- Submission length is too long. There is merit is doing just what is asked. (0 marks)
- Submission length has met the expectations of the assignment. (0 marks)
- Submission format did not meet the assignment requirements. Examples are problems with number of files, file names, file types, compression, folders, and so on. Refer to academic comments. (0 marks)
- Submission format has met the assignment requirements. (0 marks)
- Submission timeliness: Deduction is 5% per day late up to 5 days, then 100% deduction.

(Continued over page.)

Appendix A: Page design before OpenWeatherMap request



Appendix B: Page design after OpenWeatherMap request



Data from OpenWeatherMap's five day weather forecast API. Images from OpenWeatherMap's weather icons. Valid HTML5 and CSS3.