- **Features implemented:**

1. Eraser Tool allows user to remove drawing made to canvas. Holding down on left-click and moving curser over canvas will erase marks made by user. Tool was implemented using the Line P5.js function and mouseX/Y functions to set the start/end of the line.

2. Shape Tool (circle, square, triangle, diamond, pentagon, polygon) allows user to drag different size shapes to canvas using the vertex function (default coordinates from array) which can vary in size or click on the canvas to default sized shape. It is implemented using an array of objects which contain the shapes and its properties (e.g default coordinates). When user selects a shape, it uses the ID of the shape and edits the specific shapes location as the user moves it. Another function loops through the array and displays all shapes in the array.

3. Stroke Thickness changer to change the stroke size of freehand tool. User can change size using slider (made using html <input> tag). Value of the slider is then extracted using JQuery and set as the strokeWeight().

4. Opacity Changer changes the transparency of freehand/shape tools. Similar to stroke changer, user change opacity value using slider.

5. Select Tool allows the user to select an object in the array by getting the ID of the object the Mouse is over and then using event handles to change the objects x and y coordinates. It is used in the shape, image and crop tool to allow the user to move the objects around. When the user leaves that specific tool the canvas is then saved.

6. Load canvas allows user to load a previously saved canvas to continue working on. Load input implemented using createFileInput() P5 function which creates <input> in the DOM with type “input”, this allows user to load local files for use in sketch.

7. Colour palette allows user to select wider range of colours(using selector or entering RGB) than previously available. Implemented using <input> with type “color”. Fill is then set to selected colour using our ColourPalette() function.

8. Image tool allows the user to add images to the canvas. The user can change the size of the image, upload images from there directory and select an image and move it around. When an image is created its data including the x, y coordinates, the width, the height are saved as an object and pushed into an array. A loop then displays every image in the array on the canvas but does not permanently save it allowing the user to move the image around and the loop will display the image in the new position.

9. Crop tool allows the user to crop the canvas, and drag the new crop to a new location. It involves using getImageData and putImageDatet to get the image data from the canvas, storing it as an image, then saving it in an object and then displaying it on screen. The user can then drag the cropped portion ("image") of the canvas around to a new location.

- **How you planned and coordinated development:**

At the start of the project, an initial plan was created with a list of features we wished to implement, after which, we estimated how long each feature would take to implement. Based on these estimations, we created a timeline of order of implementation that we will follow.

GitHub was used to manage the project files online, using Git version control made it easy see what changes have been made and allowed us to work on one “Master copy” rather than duplicates.

Weekly meeting in library/lab were used to discuss progress made and yet to make the prior week, details of which kept on wiki. Discord was used for online-conferencing, allowing partner and I to work together from home.

- **How you structured your code/ coding techniques used:**

Features implemented were divided into separate files to keep consistent with template. By keeping it modular, it allows for reusability of code for future projects and also easier to test for bugs via module-testing. coding conventions such as "*camelCasing*" and indentation were kept consistent throughout the project to allow easier maintenance and legibility. Also, by creating functions, the amount of lines is reduced as duplicating code isn't needed, improving reusability of code.

Comments were made throughout the code, making updating, debugging and analysis of code more easy/efficient, this also helps other programmers to read/work on the code. Testing frequently done after new code/feature added to catch bugs at its infancy.

- **Challenges faced:**

There were a couple of challenges faced during the project. For example, bugs found in the original template such as the mirror tool were affecting the program where the last "drawing"/ change to the canvas before selecting the mirror tool would disappear. This was fixed by adding loadPixels() to the end of each tool to correct the bug. Another challenge faced was with the crop/copy tool, where the selected area was not copied properly. This issue took some time to debug as the root cause wasn't clear (caused by canvas size changing depending on user screen zoom level). Another challenge was keeping time-management, even though a time plan was created, sticking to the plan was a challenge as it wasn't completely thought out initially.

- **Self-evaluation: what would you do differently next time?**

If we were to redo the project, one thing we would do differently is to be more time efficient with the implementation/debugging as spending too much time on either affects the final outcome. Another thing we would do differently is to test the initial code given more to eliminate bugs, whilst having less code to prevent issues later.

- **Project wiki page included as an appendix (does *not* count towards word count)** [1 SEP]

Final Project

Group 69 Final Project Plan

Extra ideas:

- Eraser Tool
- Shapes: Triangle,Square,Circle
- Thickness stroke
- Stroke Type eg: Thin, medium, thick
- Select tool to move parts of the drawing to other places
- Brightness/Contrast
- Canvas size
- Change the UI
- Load Tool
- Zoom Tool
- Opacity
- Dropdown menu
- RGB Colour system/Colour pallette
- Fill Tool
- Canvas colour
- Spiragraph Tool
- text tool
- crop tool

Expected Time

1h-2h	• Eraser Tool
1h-2h	• Text Tool

3h-4h	• Shapes: Triangle,Square,Circle, Pentagon, Polygon
2h	• Thickness stroke
3h	• Image tool
5h	• Crop Tool
6h	• Select tool to move parts of the drawing to other places
4h	• Brightness/Contrast
2h	• Canvas size
2h	• Change the UI
4h	• Load Tool
5h	• Zoom Tool
2h	• Opacity
5h	• RGB Colour system/Colour pallette
5h	• Fill Tool
2h	• Canvas colour
10h	• Spiragraph tool

week 1: 22nd - 26th

We went over the pre-existing code of the drawing app, commented throughout the code, marking on sections we shouldn't touch and what we need to understand.

Learned how to use Git & GitHub to manage the project. Plan to work on branches before committing to master copy. Will help prevent bugs from affecting already working code.

Week 2: 5th - 12th

shape tools were added, but not yet finished.

Stroke size changer feature added.

Load image feature added, had to use P5 reference to implement (CreateFileInput)

colour picker feature added, used W3School to help.

Week 3: 12th - 19th

Fixed glitch where last drawing would disappear when mirror tool is selected.

Finished the shape tools. Still need to work on fixing some problems with the triangle tool.

Added size function to work with freehand tool, line tool and eraser tool.

Week 4: 20th - 27th

A lot of html and css added.

Jquery scripts added to fade the second toolbar in and out.

Fixed the issue where when you would save for the first time, the whole image would be blank.
Fixed the issue where the colour would not be saved between switching from the eraser to another tool.

Week 5: 27th - 3rd

Created a function to change the opacity, which is passed into the colourPallette as a parameter.

Sliders added to change the size and opacity, rather than manually typing the value.

Changed the freehandtool to use vertex function instead of the line function. This was because it was not keeping a consistent opacity when drawing.

Used JQuery for the size and opacity slider.

Week 6: 3rd - 10th

size and opacity changed. Added opacity to colour palette. fixed outline on shape tools.

Added opacity to work with the slider. Made it so it increments in 0.01 steps so it moves evenly with the size slider. Added the Jquery for the opacity slider.

fixed size function. Added size to work universally. toolSettings.js created. Still need to fix opacity

Fixed the opacity and size sliders.

Moved all the tools into shapeTools. Added new icon boxes to the .options class. Removed old tools js files. Used JQuery to find the current shape tool selected. Fixed issue when switching between tools and freehand tool, line tool by using push pop. We will add a proportional shape sizer. We will also add a pre defined shape size which draws the shape when clicking on the screen.

Week 7: 10th - 17th

Added moveable shapes. The shapes function totally reworked to allow moveable shapes using points. New moveable ui popup box, still needs work. New shape tools added. triangle tool fixed. bug when saving yet to fix.

Added outline selector. shape size changer. fixed some drag bugs, Saving bug fixed.

Added image tool. upload image, image drag, select tool. Used CreateFileInput, fixed image tool function.

Added image move with size and width change feature.

Week 8: 17th - 22nd

crop tool to select part of the canvas. Crop issue where had to *2 to x y width and height. Used push image data and get image data.

Fixed canvas issue with x2.Added crop drag feature.

Icon changes, fixed crop tool with negative width and height.

textTool.js created, implemented text tool to type string of characters onto canvas.

References:

https://www.w3schools.com/howto/tryit.asp?filename=tryhow_js_draggable

https://stackoverflow.com/questions/23451267/how-to-make-drawing-draggable-on-canvas-in-html5

http://api.jquery.com/append/

https://stackoverflow.com/questions/34369053/how-to-drag-shapes-like-rect-circle-other-polygons-etc-around-the-canvas-using

https://www.w3schools.com/jquery/jquery_dom_add.asp

https://www.w3schools.com/tags/canvas_getimagedata.asp

https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/putImageData
https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/getImageData