

Paint Web App Report

Name	<u>ID</u>
Muhammad El Kotb	19016258
Youssef Said	19016903
Abdelmoneim Hany	19017359
Ahmed Adel Abudef	19015264

Table Of Contents

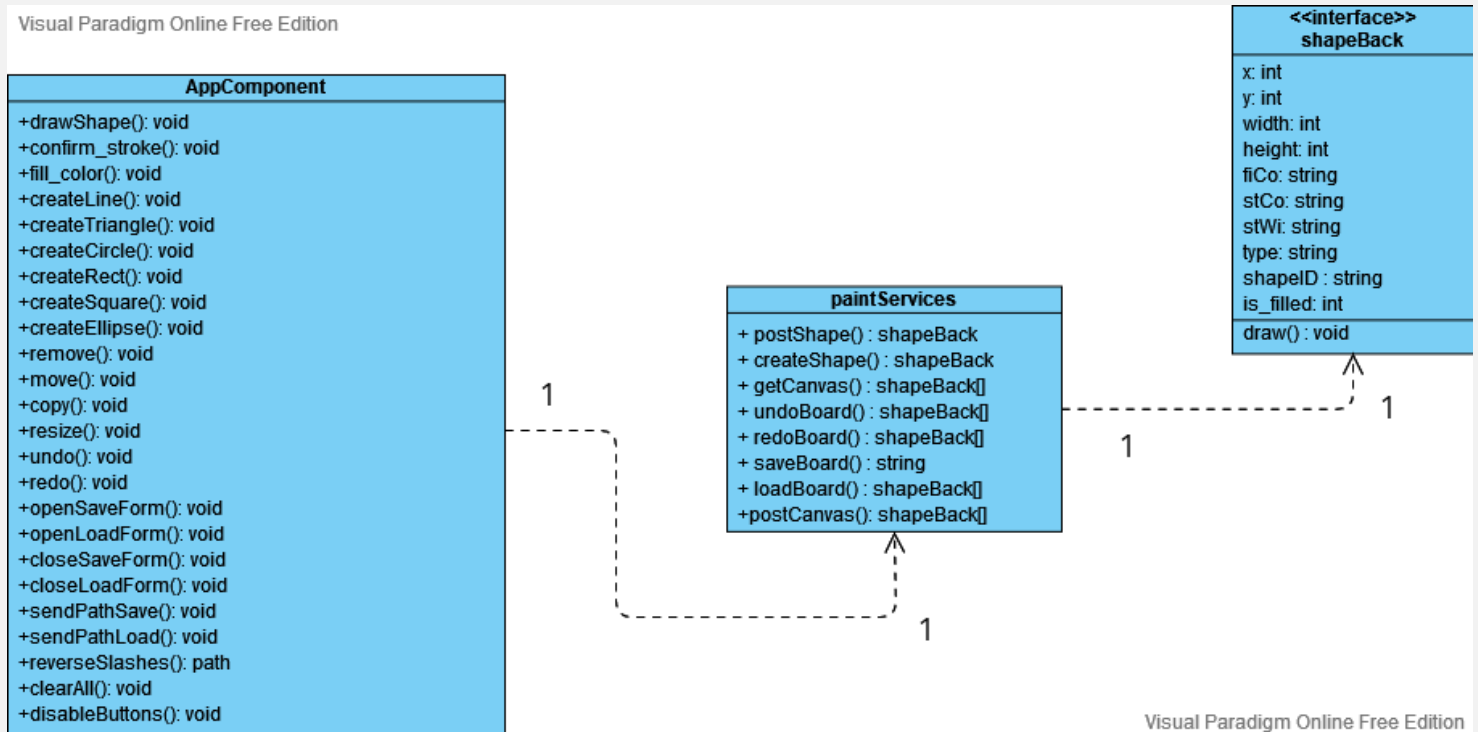
<u>1.</u>	<u>STEPS TO RUN CODE</u>	<u>2</u>
<u>2.</u>	<u>UML DIAGRAM</u>	<u>3</u>
<u>3.</u>	<u>DESIGN PATTERN EXPLANATION</u>	<u>5</u>
<u>4.</u>	<u>DESIGN DECISIONS</u>	<u>6</u>
<u>5.</u>	<u>SNAPSHOTS</u>	<u>7</u>
<u>6.</u>	<u>USER GUIDE</u>	<u>12</u>

1. Steps to run code

Firstly you have to create a new angular project using npm install and then copy our entire folder and replace the files in your created folder with our files, just because node modules may be missing. Spring Boot folder is straight forward just open the pom file using any IDE. Then you run the Angular project and on localhost:4200 , and the Spring Boot project on localhost:8080.

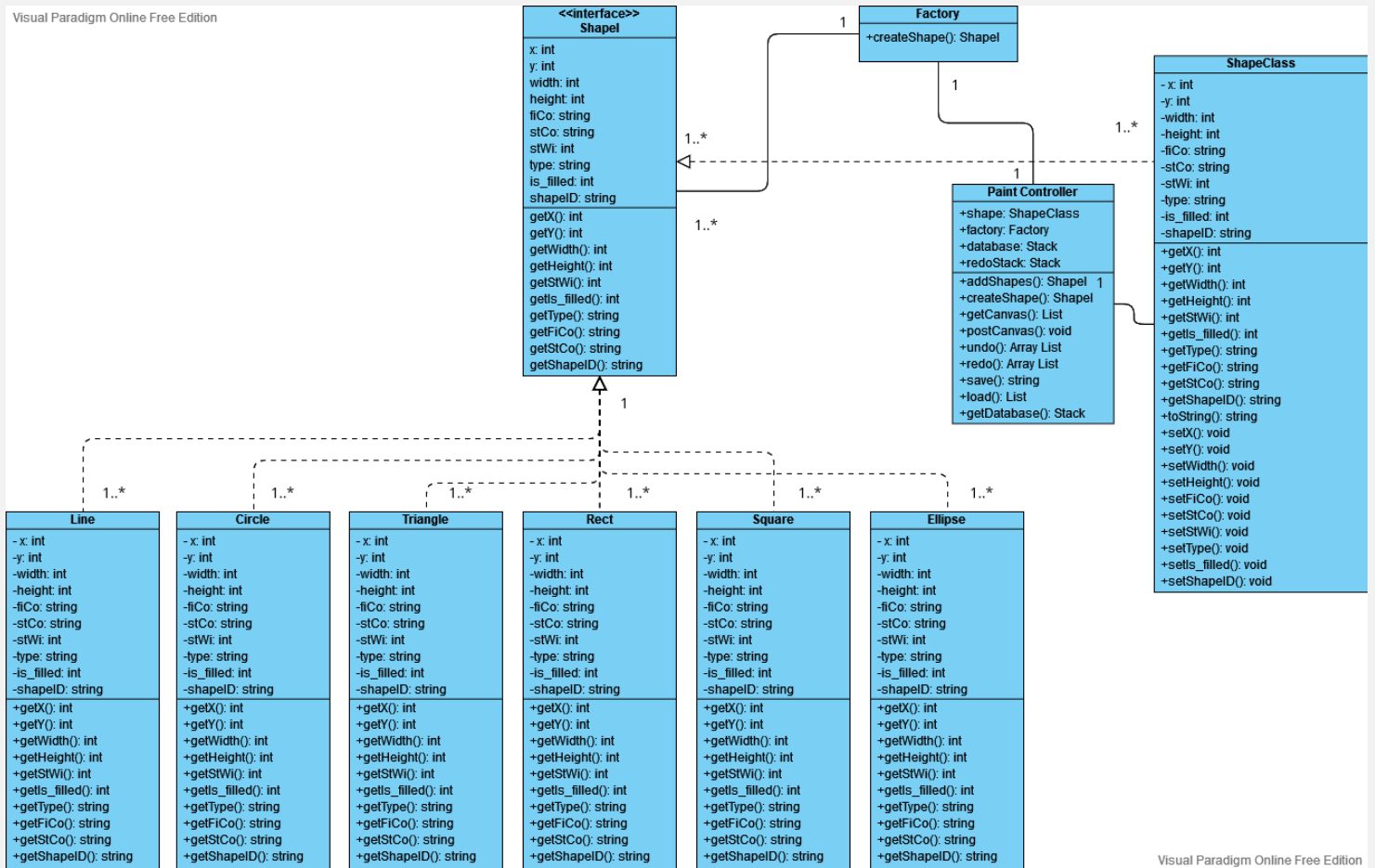
2. UML Diagram

1- Frontend Class Diagram:



2- Backend Class Diagram:

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

3. Design Pattern Explanation

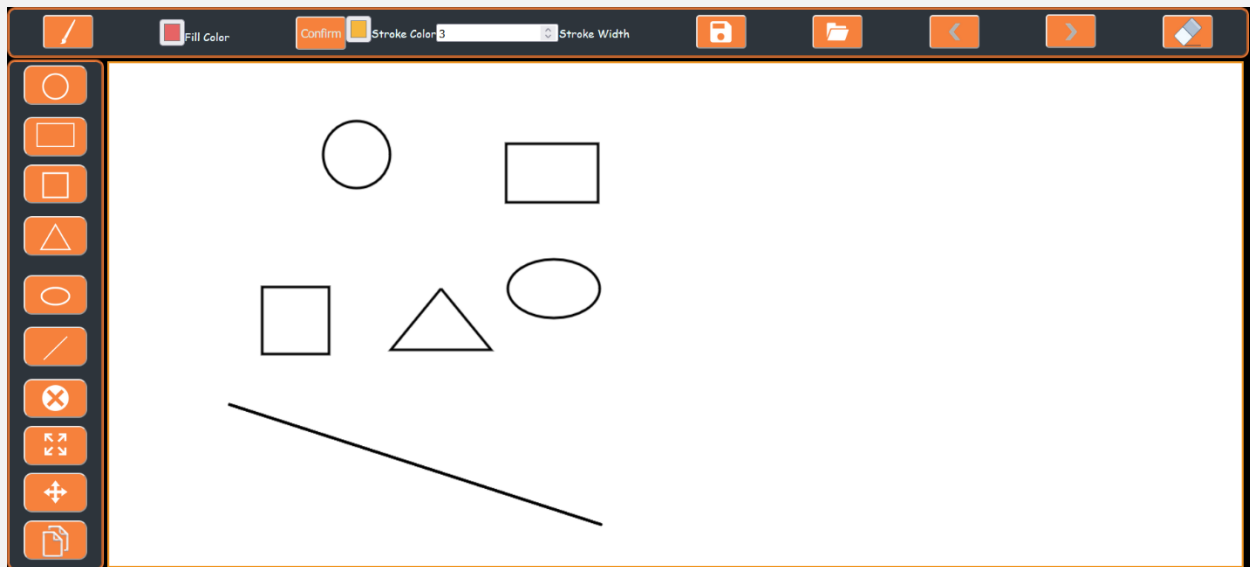
There is a factory design pattern present in our backend code which is responsible for creating shape objects using a shape interface and all other shapes implementing it and a controller instantiating the factory class and using its constructor to create shape objects. There are also basic design patterns present throughout our code like interface and the immutable design pattern which helps secure our code by making all our fields private and our methods public.

4. Design Decisions

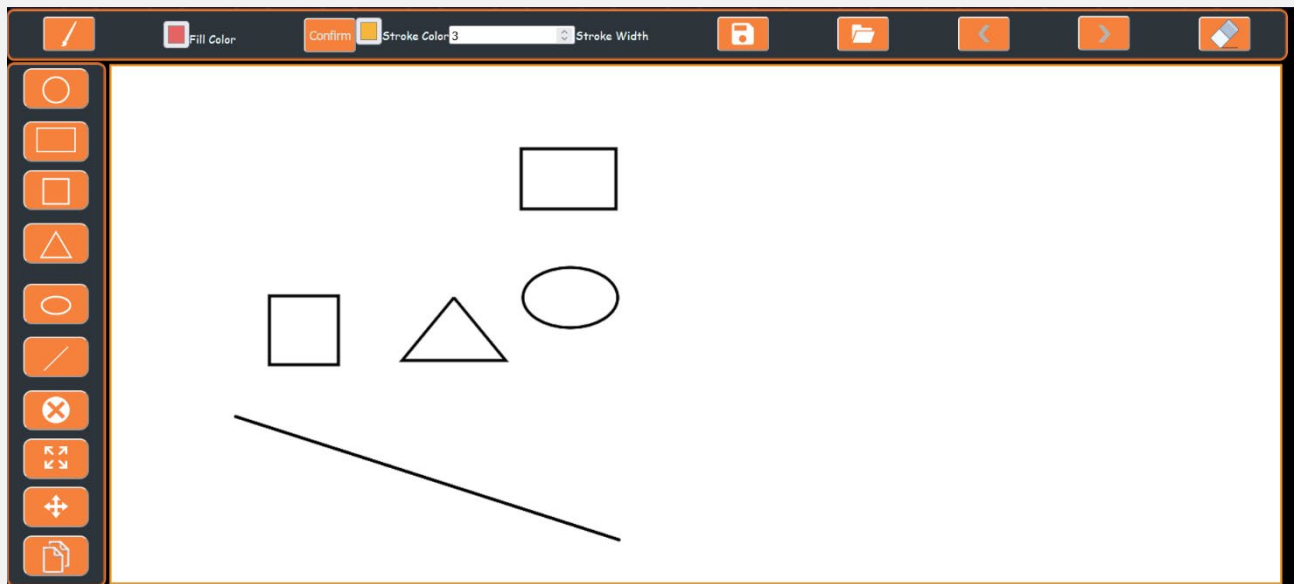
We assumed that the user would create a shape with a predefined size and then he can resize it. We assumed that the user would press to create an object, not drag. We assume that when the webpage is refreshed, the backend and database are restarted. We assumed the user wouldn't need more than fill color, stroke color, stroke width to change the appearance of an object. We assumed that the most important shapes are the ones present in our program, and that the line must be drawn with dragging not pressing to capture the angle of the line. We also assume that if the user doesn't enter an extension when saving then the default is a .json file.

5. Snapshots

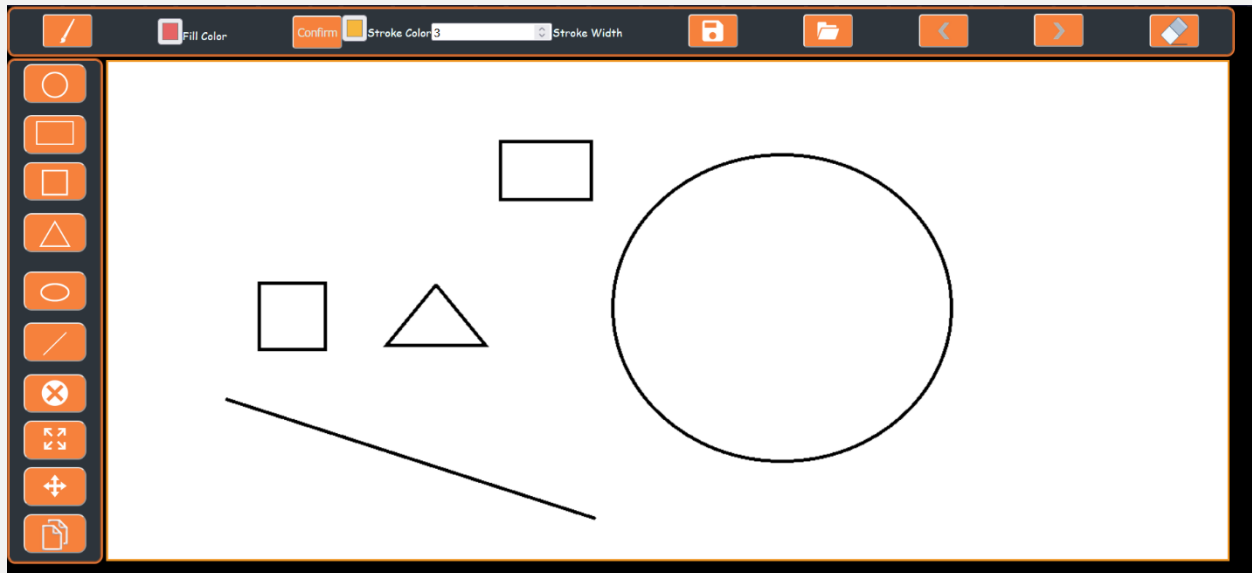
1- This is a snapshot of all objects on the canvas



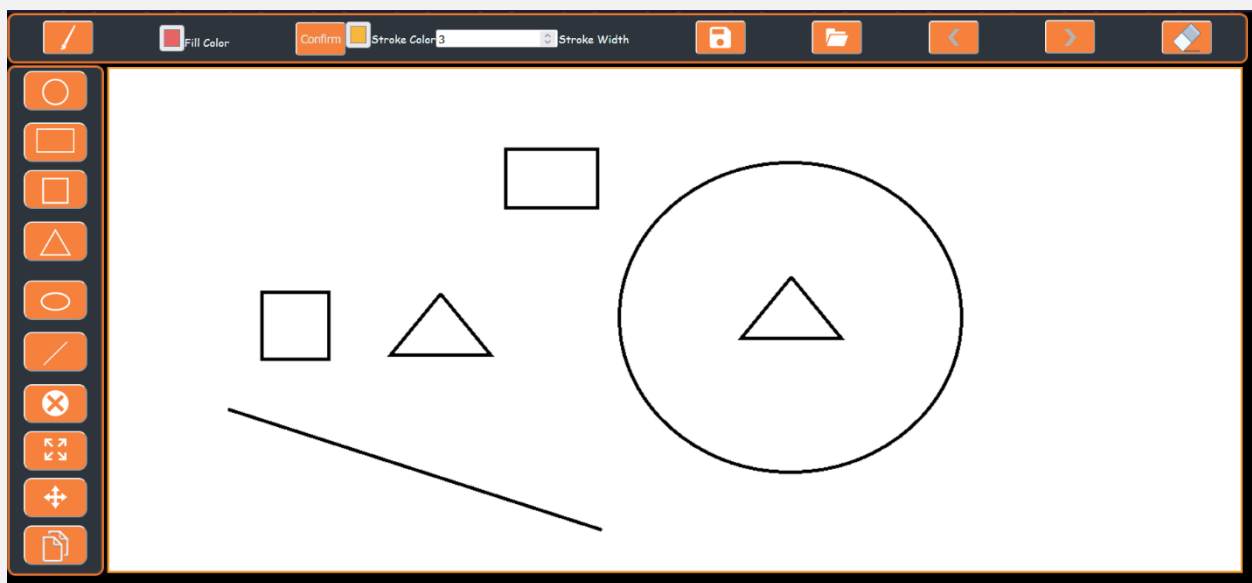
2- Next we removed the circle using the remove (X) button the left toolbar.



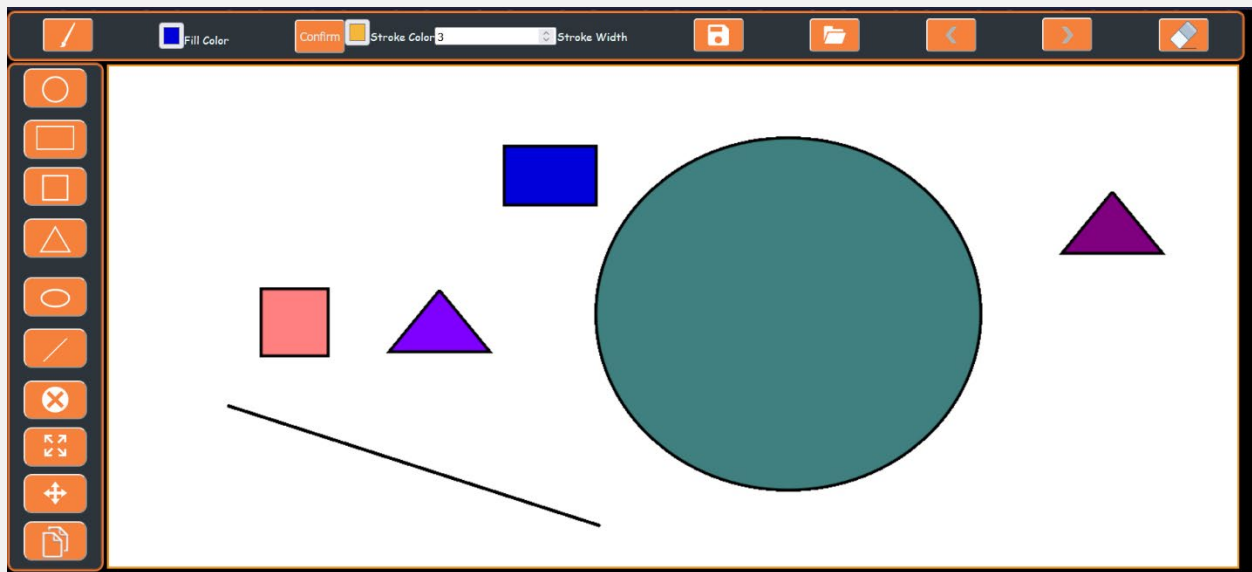
3- Next we increased the size of the ellipse and moved it to the right using resize and move buttons on the left toolbar.



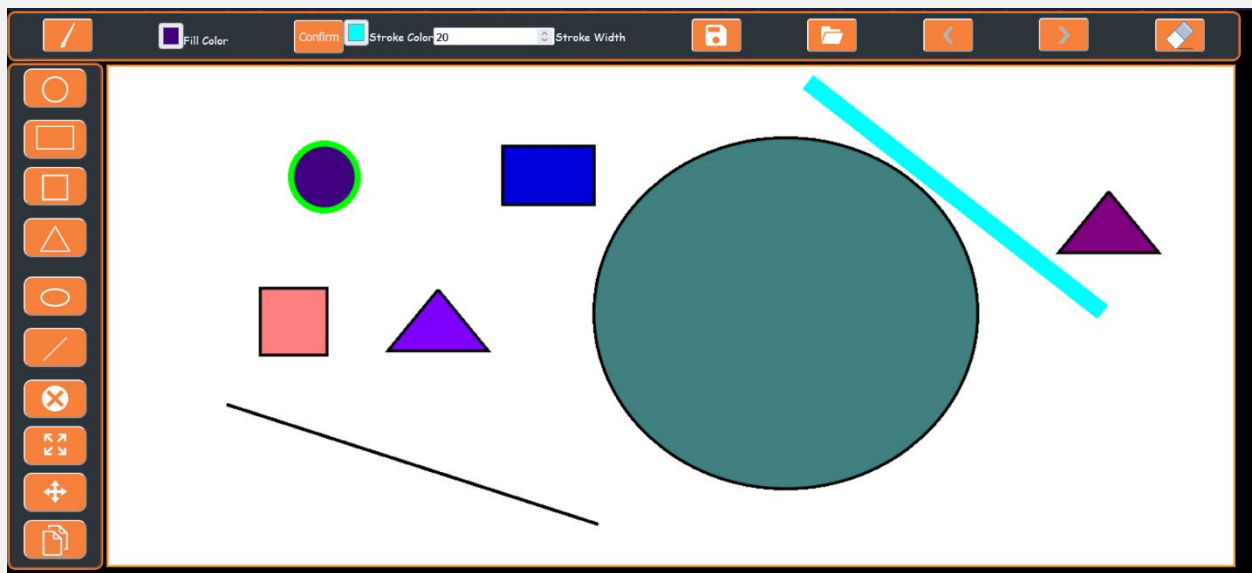
4- Next we copied the triangle inside the circle using the copy button on the left toolbar.



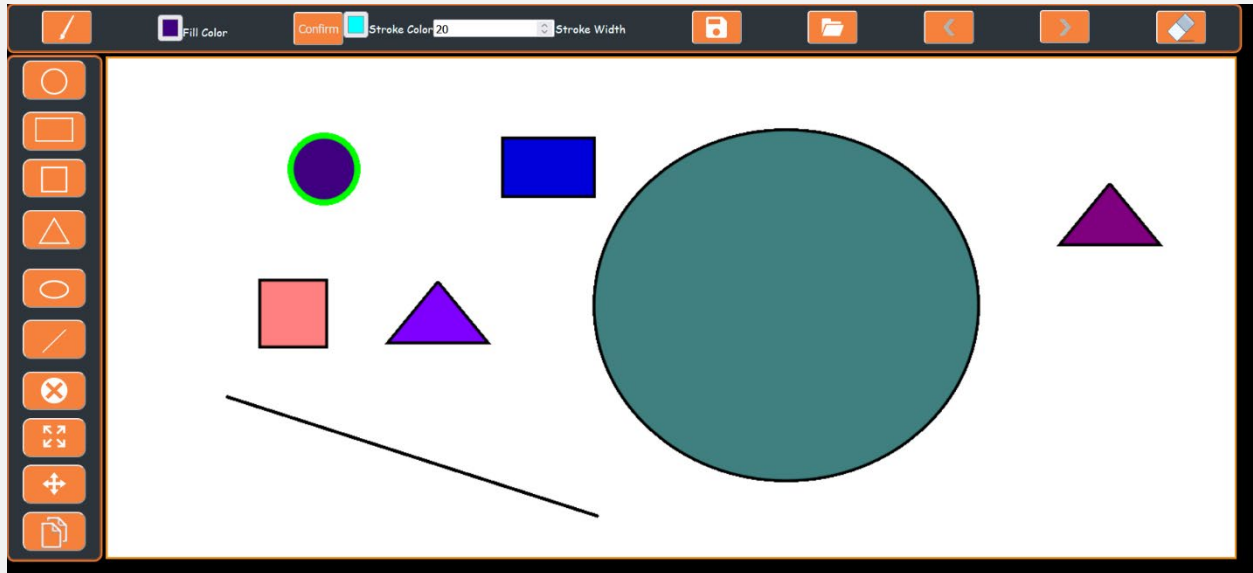
5- Next we moved the triangle to the right and colored the objects using the fill color and and brush icon on the top toolbar.



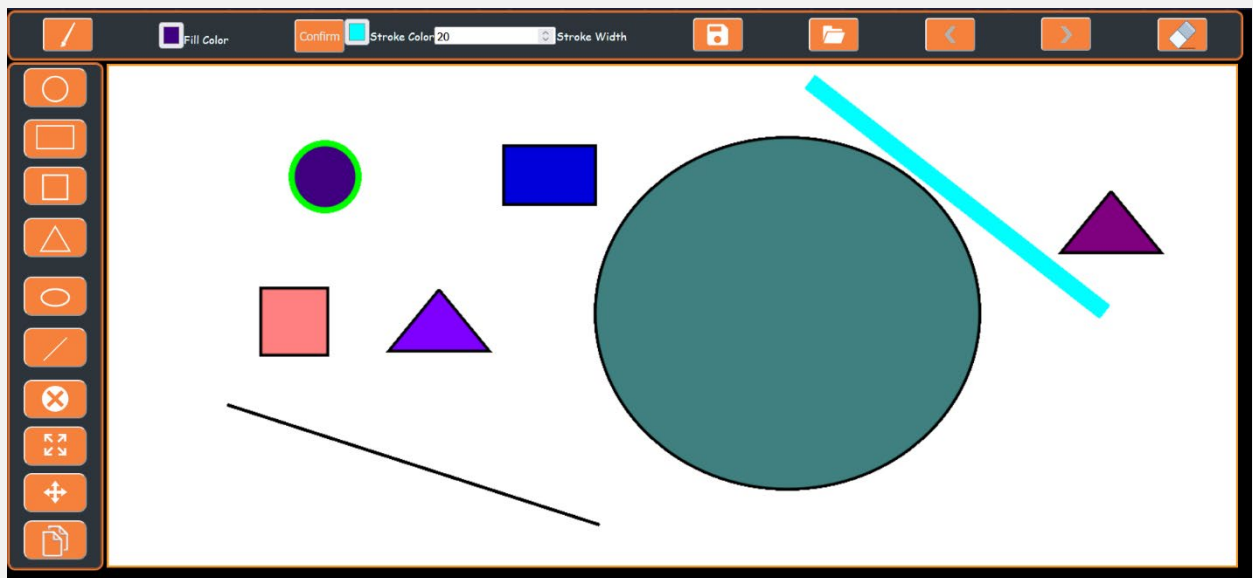
6- Next we created objects with different stroke styles like the dark purple circle and the cyan line.



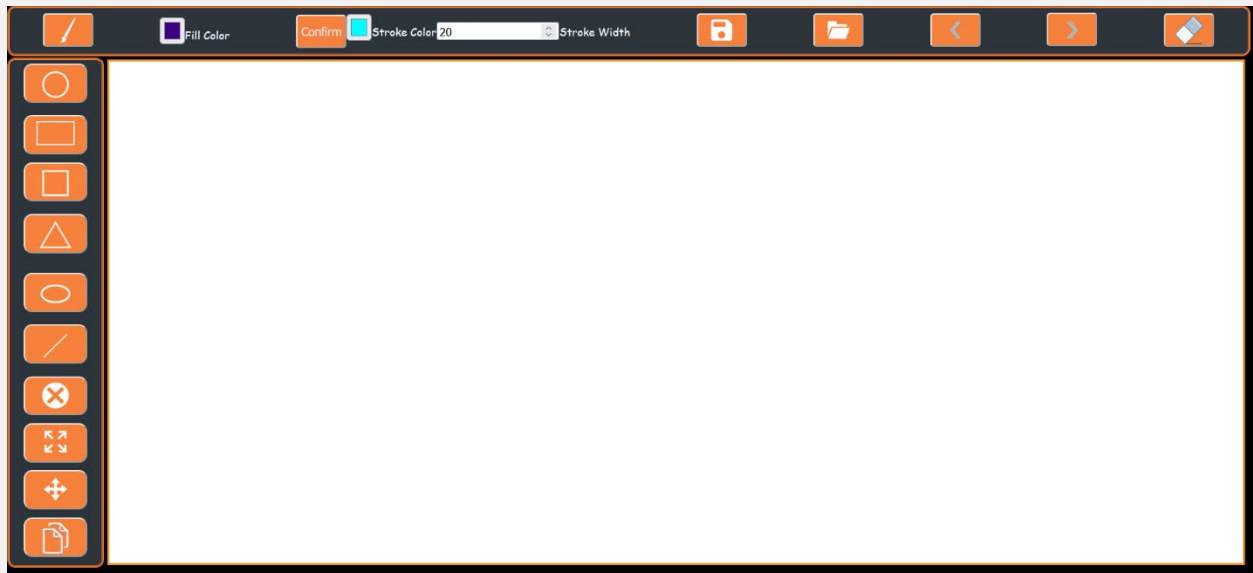
7- Next we used the undo button (<) on the top toolbar to remove the latest object which is the cyan line.



8- Next we used the redo button (>) on the top toolbar to bring back the undo-ed object which is the cyan line.



9- Next we used clear button on the top toolbar to clear the entire canvas.



6. User Guide

- 1- **Object Creation:** To create objects (other than line) you press on the object you want on the tools toolbar and then press once on the canvas in the place you want the object to be placed in, but to create a line you must drag the mouse from the starting position to the ending position.
- 2- **Object Coloring:** For objects to have different styles we have 3 functionalities in the coloring toolbar. First the fill color works by picking a color and then pressing inside the object you want colored. The stroke color and width are static functionalities that you can choose and apply to objects that are newly created, but objects that were previously created have a constant stroke color and width, but there is a workaround you can just remove the object and create a new one with the stroke color and width you want.
- 3- **Object Manipulation:** We have 4 functionalities in the tools toolbar. Move is straight forward you can press inside an object or in its stroke and drag it to the location you want. Resize is the same but you drag to increase or decrease size. Remove is easy you just press on the remove button and press inside an object or its stroke and it's gone. Copy works by choosing the copy button and then pressing inside or in the stroke of an object and then dragging to another location.

4- Object saving and loading: We have 5 functionalities. You'll find undo or redo buttons in the top toolbar. A save button is present and requires a valid directory, the same goes for the load button. A clear button in the top toolbar that clears the entire canvas. The save option works using .json file or .xml, if the user enters any of those extensions then it's okay if he doesn't enter an extension at all then it's assumed to be .json by default, but if he entered any other extension then he's prompted again. If a path is wrong either in save or load then you will be prompted again.