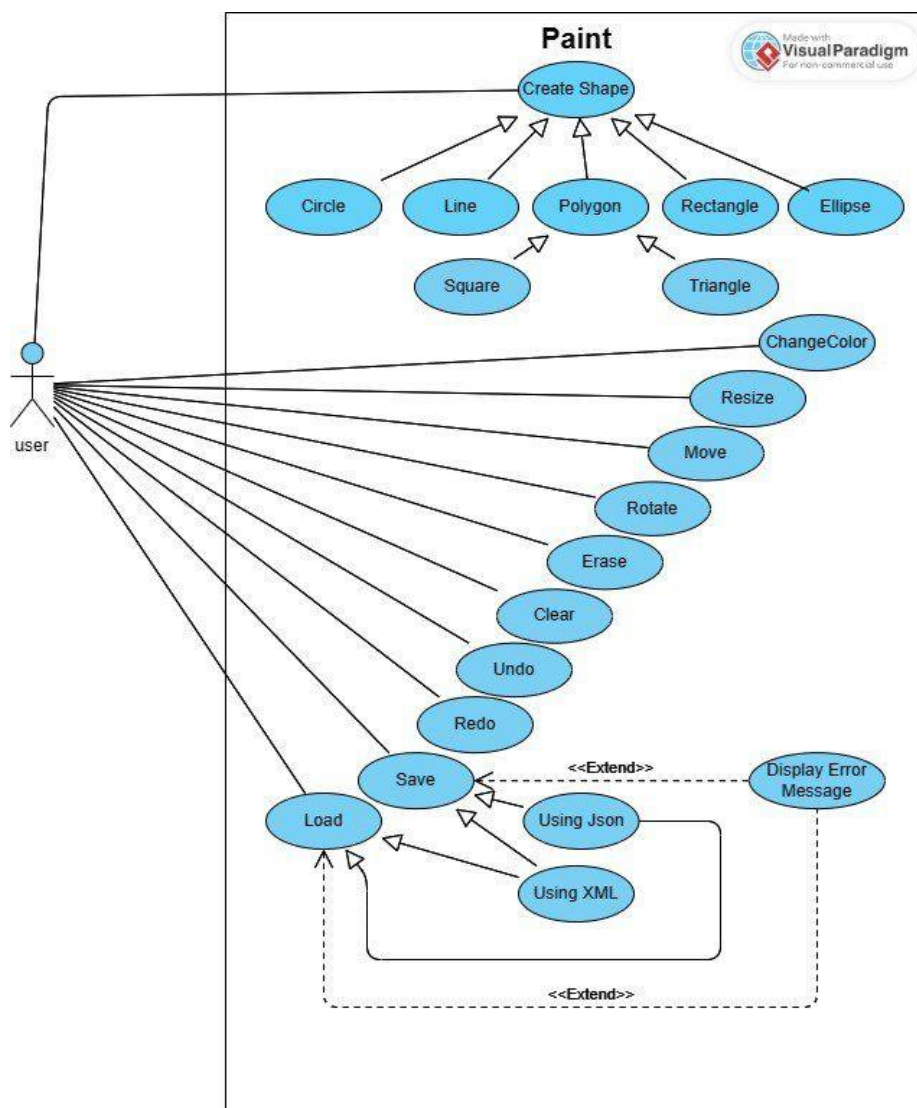# Assignment #3:Paint

Team Members:
1. **Rowan Gamal Ahmed Elkhouly** - 21010539
2. **Yomna Yasser Sobhy** - 21011566
3. **Yousef Mahmoud Mohamed Ahmed** - 21011630
4. **Mustafa Mohamed Elkaranshawy** - 21011375
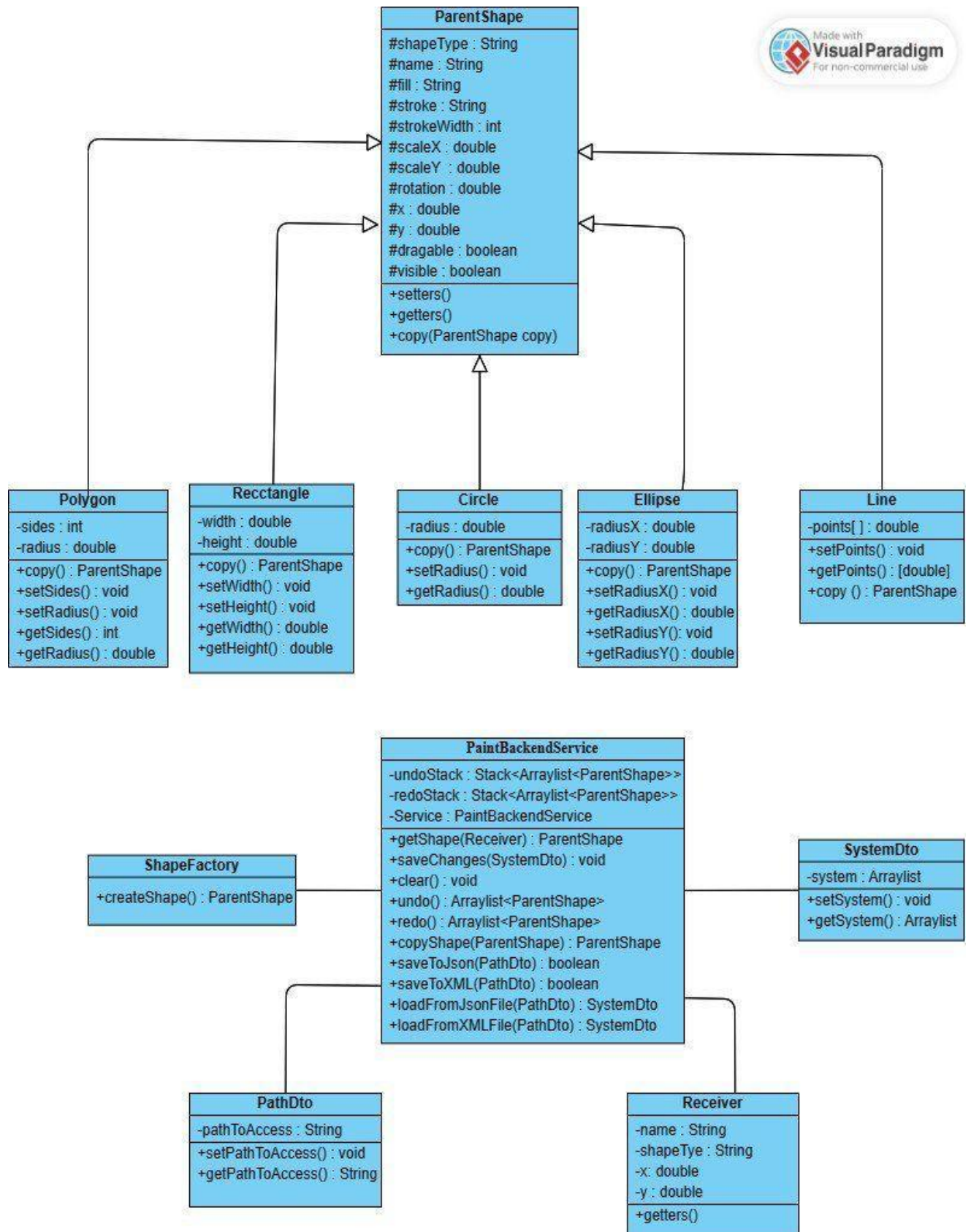
Steps of running the code:
- The backend and the frontend files should both be installed with all the included dependencies and packages
- The backend file should run first as we made it run on a default port of 8081
- The frontend file should be run afterward, and the local host is opened in the browser

UML diagrams:

Use-case diagram: shows the interaction between the user and system

Class diagram: shows the used classes, interfaces, and the relations between t themselves

ParentShape

#shapeType : String
#name : String
#fill : String
#stroke : String
#strokeWidth : int
#scaleX : double
#scaleY : double
#rotation : double
#x : double
#y : double
#dragable : boolean
#visible : boolean

+setters()
+getters()
+copy(ParentShape copy)

Polygon

-sides : int
-radius : double

+copy() : ParentShape
+setSides() : void
+setRadius() : void
+getSides() : int
+getRadius() : double

Recctangle

-width : double
-height : double

+copy() : ParentShape
+setWidth() : void
+setHeight() : void
+getWidth() : double
+getHeight() : double

Circle

-radius : double

+copy() : ParentShape
+setRadius() : void
+getRadius() : double

Ellipse

-radiusX : double
-radiusY : double

+copy() : ParentShape
+setRadiusX() : void
+getRadiusX() : double
+setRadiusY(): void
+getRadiusY() : double

Line

-points[ ] : double

+setPoints() : void
+getPoints() : [double]
+copy () : ParentShape

PaintBackendService

-undoStack : Stack<Arraylist<ParentShape>>
-redoStack : Stack<Arraylist<ParentShape>>
-Service : PaintBackendService

+getShape(Receiver) : ParentShape
+saveChanges(SystemDto) : void
+clear() : void
+undo() : Arraylist<ParentShape>
+redo() : Arraylist<ParentShape>
+copyShape(ParentShape) : ParentShape
+saveToJson(PathDto) : boolean
+saveToXML(PathDto) : boolean
+loadFromJsonFile(PathDto) : SystemDto
+loadFromXMLFile(PathDto) : SystemDto

ShapeFactory

+createShape() : ParentShape

SystemDto

-system : Arraylist

+setSystem() : void
+getSystem() : Arraylist

PathDto

-pathToAccess : String

+setPathToAccess() : void
+getPathToAccess() : String

Receiver

-name : String
-shapeTye : String
-x: double
-y : double

+getters()

Features:

- Shapes: Line Segment, Circle, Ellipse, Triangle, Rectangle and Square
- Colouring
- Resizing
- Erasing / Deleting
- Clearing whole board
- Copying a shape
- Moving and Dragging
- Undo-ing
- Redo-ing
- Saving in JSON or XML file
- Loading previous saved JSON or XML file

Extra Features:

- Rotating shape
- Saving board as PNG image
- Changing the mode of the page to Pink or Blue.

—--

How We Applied Required Design Patterns:
Design Patterns applied:
1. We have applied **Factory design pattern** such that when the user wants to create a shape, the shape factory:
    a. Creates it for him as we send the shapeType to the backend
    b. We create the required shape according to the shapeType given from frontend, choosing the right class that satisfies the required shape and sending to frontend the shape we have created with its default values.
2. We have applied **Prototype design pattern** such that when the user wants to copy a shape:
    a. we send the shape we want to copy to the backend
    b. Backend creates a shape from the same class sent from Frontend
    c. It gives its attributes the same values as the sent shape
    d. Save it in the Backend and send it to the frontend.

Design Decisions:

- We placed the **tool bars** at the top of the window to make the board isolated from the tools, so nothing distracts the user's painting and to maximize the board width as maximum as possible.
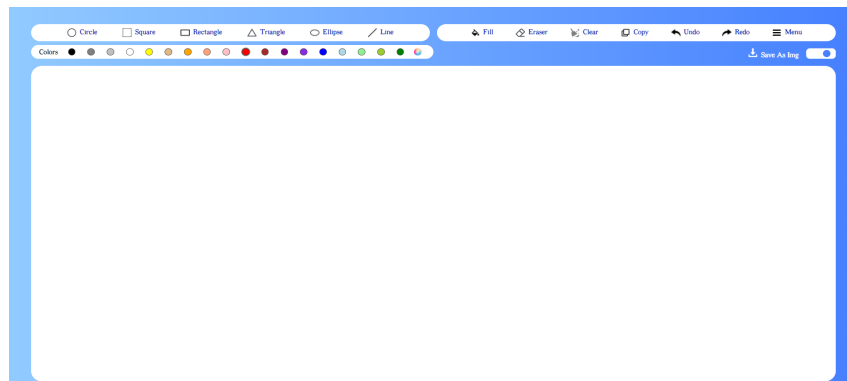

- We put the most used colors to make it easier for the user to pick the desired color, we also add the gradient colors box if the user wants to customise the color.


 - We also made the color of the fill button the selected color to add more Usability.
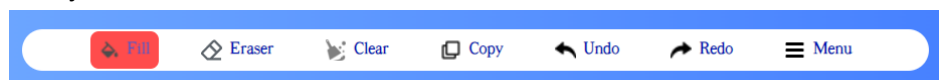
- We separate the **Shape** bar and the **toolbar** to make it easier for the user to choose the action he wants.
- The **Menu** button has the option to save or load paint files from the local storage and we decided to put them in one icon to separate them from the other tools.
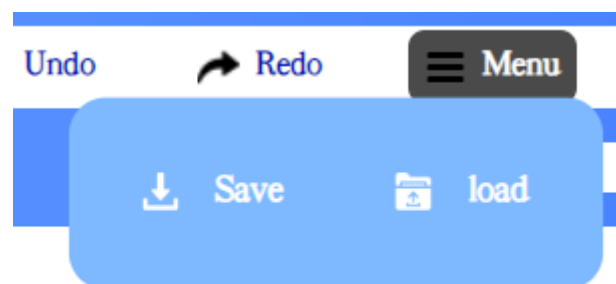
- We added **the active button property** to show the user the currently enabled tool and use it repeatedly, so he doesn't take any wrong or unneeded actions.
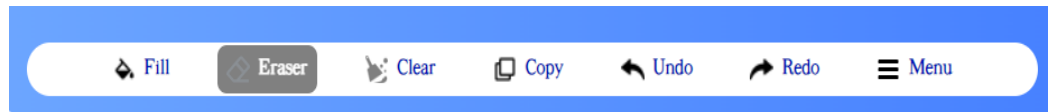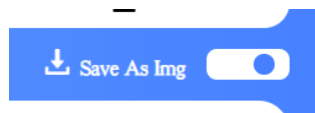

  - We also provide the active button property for the **shape bar** but can be used only once.
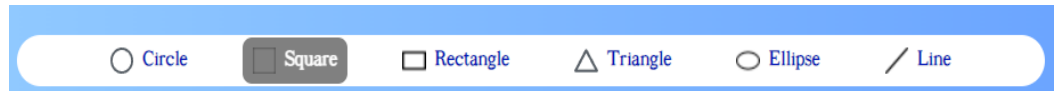
  - We also added the active color property to show to the user which color from the **Main colors** is used now.

- We added the **Change Mode** button between **pink** and **blue** to provide a variety of modes to cover all users.

  - The two modes' color contrast is very comfortable for the eye.

—--

User Guide:

- Upon opening the application the drawing board is fully white and empty
- **To create a shape**, just choose it from the menu and press where you want to put it on the board.
- **The fill, erase, and copy buttons** must be selected and deselected to be activated or not, it doesn't get activated or deactivated by itself.
- **To change the shape color** :
    a. You choose any color you want from the pallet
    b. You <u>MUST activate the fill button</u>
    c. Then choose the shape you want to apply the color to
- You can **change the position** of a shape by dragging it
- When you click on a shape, it gets selected and then you can **resize and rotate** it
- **To erase** a drawn shape you activate the erase button and then choose the shape you want to delete
- By pressing the **clear button**, all the shapes on the board will be deleted
- **To make a copy** from a shape you activate the copy button and then choose the shape you want to copy
    > *(please note that a copied shape is generated with <u>a default offset</u> so if you copied the same shape multiple times the copies will be completely overlapped so you will have to start moving one to see the copy under it)*

- The **undo and the redo buttons** simply work by pressing on them
- Once a file is loaded, you cannot undo the load and you cannot undo previous work on that load but you can add/create shapes, make changes, and undo them normally.
- There is a menu on the right that contains **the save and load buttons**
- **To save the current file** you enter the folder path separated from the folder name and then choose the required extension to save with
- **To load a file** you copy the file as a path and enter it as input in the load text box
- You can export your drawing and save it to your device as an image by pressing the **(save as image)** button
- We made a button on the right that activates our one and only **PINK MODE :)**

User Interface:
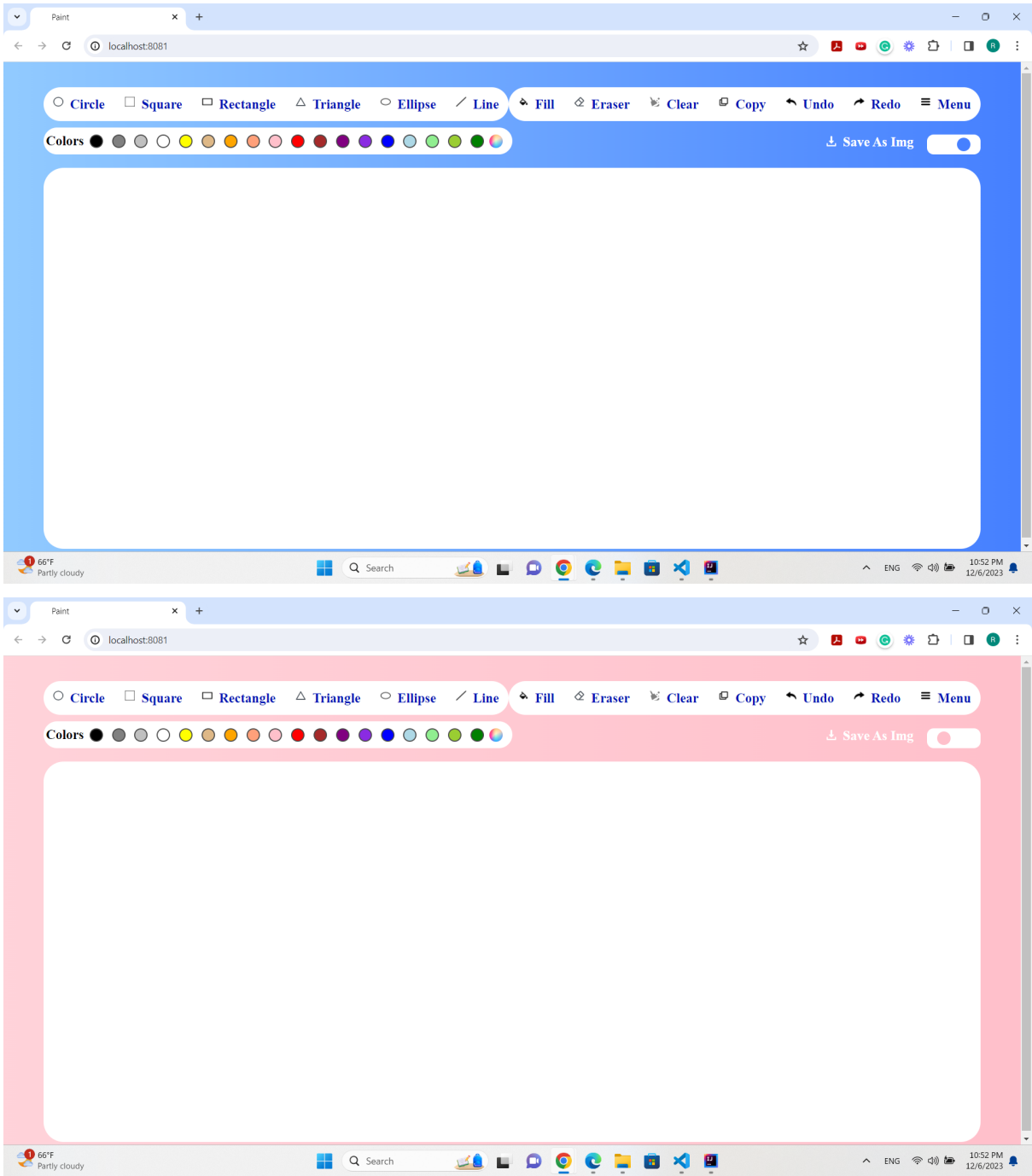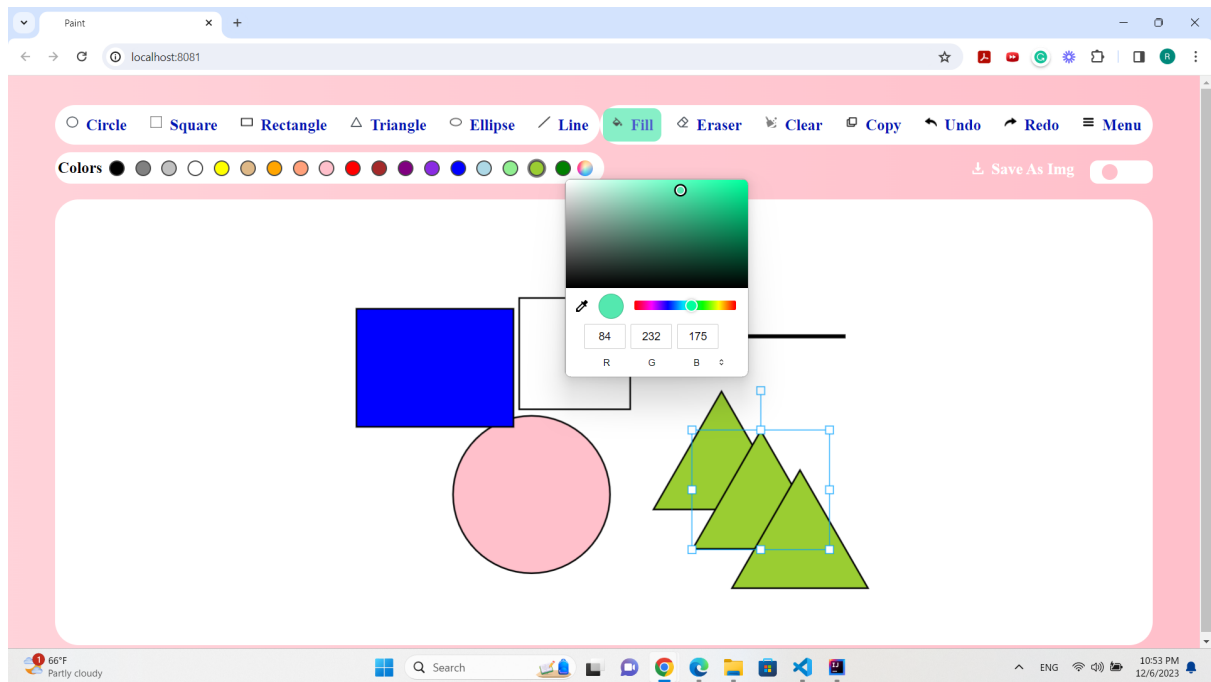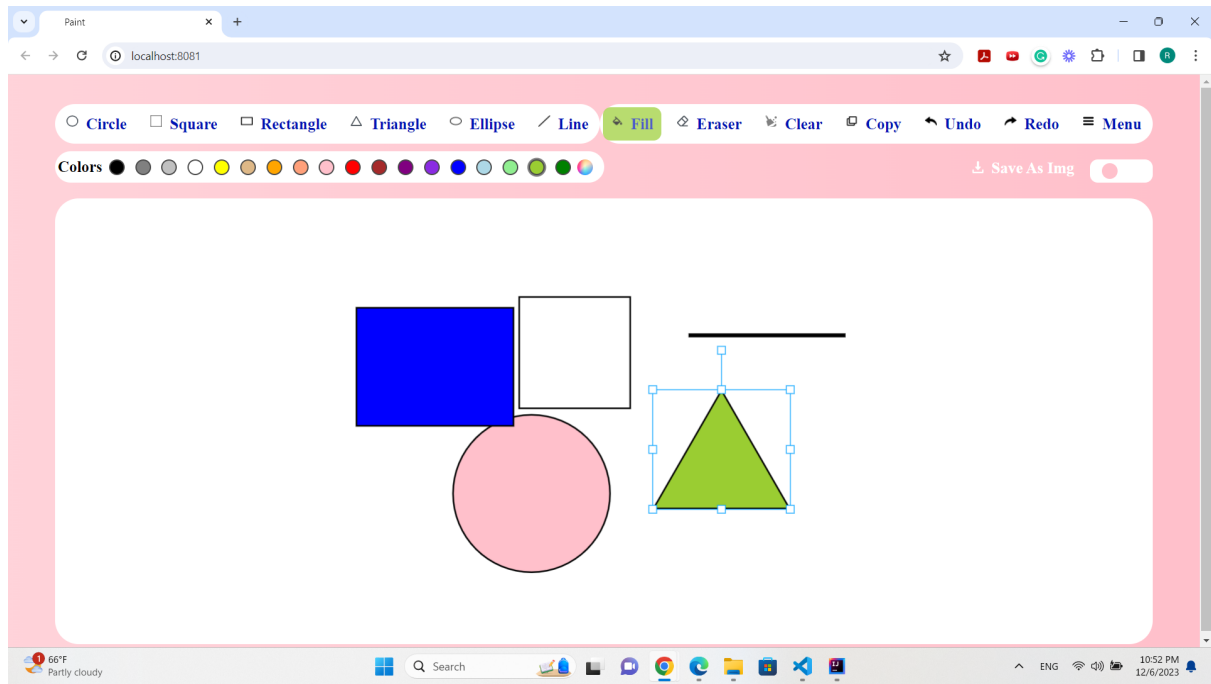
Illustrative video:
https://drive.google.com/file/d/1YdA_pGO2gUErPPAxQ9t97K95gNHPF9UW/view

External Resources:
https://konvajs.org/index.html