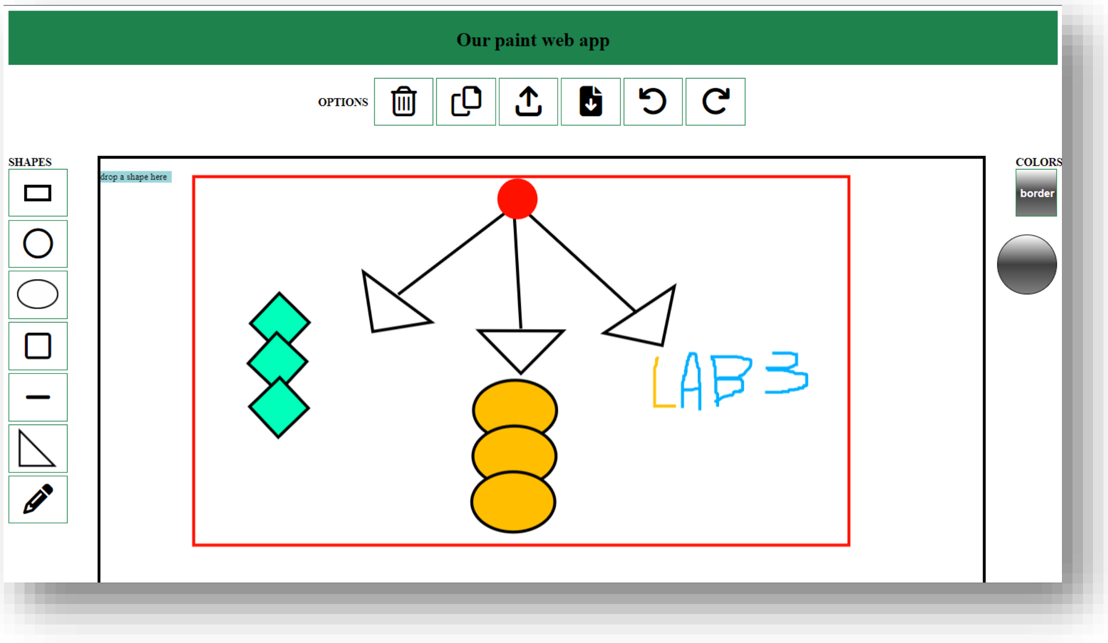
PAINT

Abdelrahman Wael.20010869 Ahmed Hesham.20010234 Abdelrahman Mohamed.20010851
Mohamed Amin.20011502

Our web application for geometric shapes modeling implementing an object-oriented drawing and efficient app.

OVERVIEW

DESIGN



FUNCTIONAL REQUIREMENTS

- Drawing basic shapes (Rectangle, Square, Circle, Triangle, Ellipse)
- Free hand drawing and Lines
- Editing (Resize, Move, Delete, Copy)
- Coloring (Fill color – Border Color)
- Undo and Redo
- Save and Load

NON-FUNCTIONAL REQUIREMENTS

- Handles user spamming buttons.
- Support many shapes in the same file.

IMPLEMENTATION

DESIGN PATTERNS

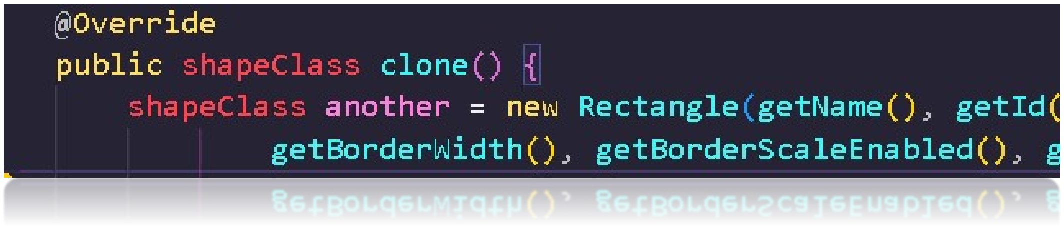
1) Factory 2) Prototypes 3) Polymorphism 4) MVC

Factory

Used in the backend as it receives the name of the shape to be drawn. This shape is sent back to the front end to be drawn with the standard attributes.

Prototypes

To easily copy shapes with same dimensions, position and colors. A clone was made from the same object. For object indication, each shape has a unique ID which helps in copy and delete and a reference for any shape in general.



```
@Override
public shapeClass clone() {
    shapeClass another = new Rectangle(getName(), getId(
            getBorderWidth(), getBorderScaleEnabled(), g
```

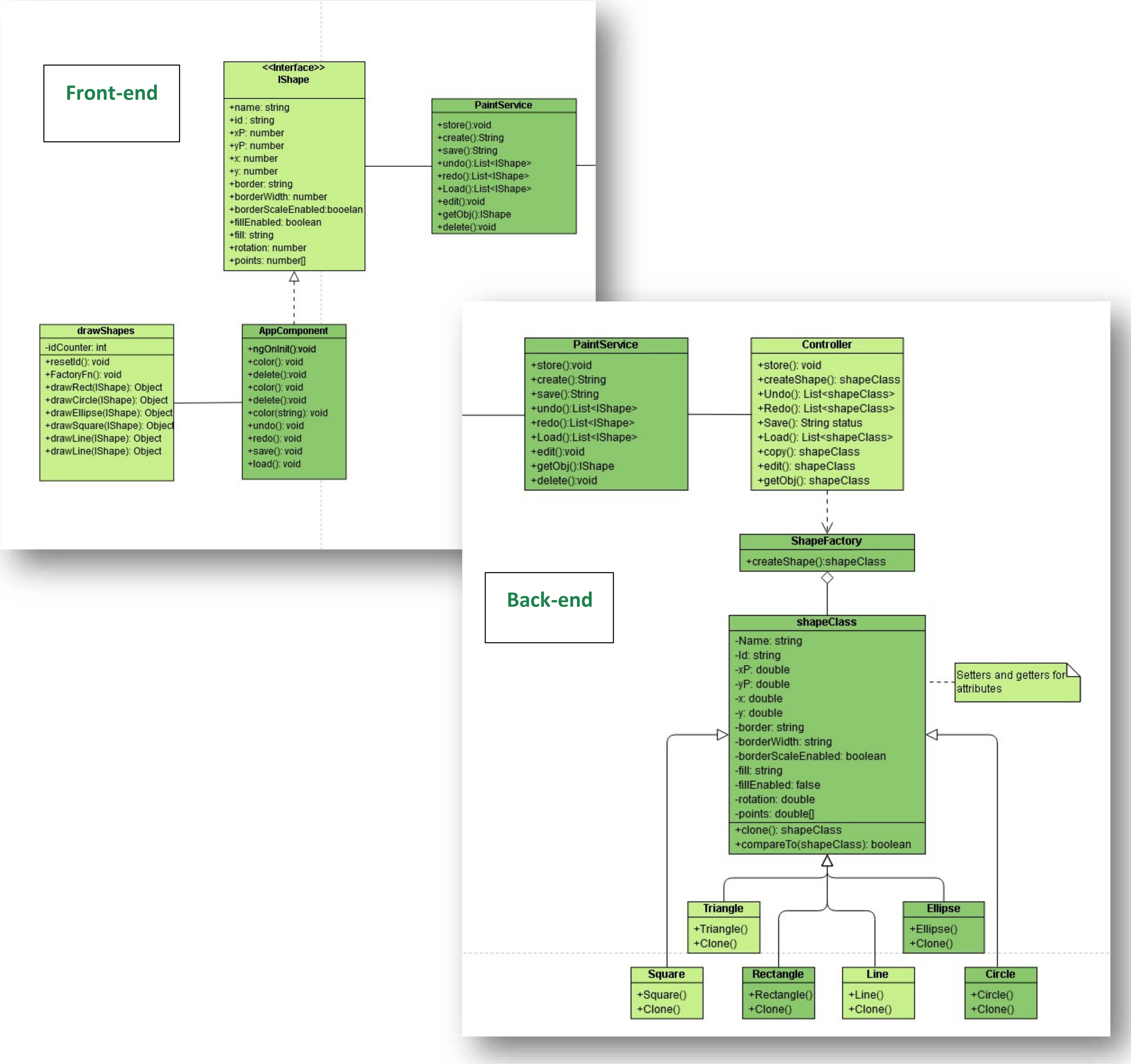
Polymorphism

Every shape will be assigned to a shape object. The behavior of the object will depend on the object passed.

MVC

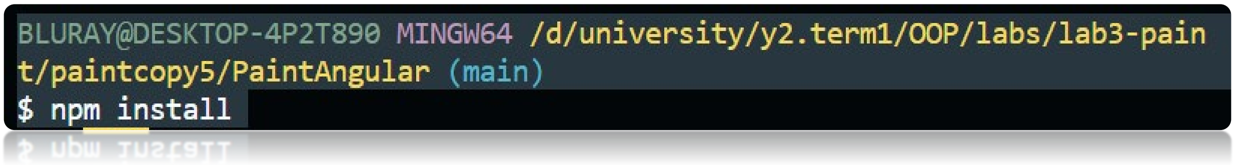
The app uses MVC to decouple the presentation code from data access code. View is represented in the front end where data is displayed and user inputs are taken. A Controller class in the backend is the only way to communicate with the Model where all the requests are sent to the controller and the business logic is implemented.

UML


Front-end
<<Interface>>
IShape
+name: string
+id : string
+xP: number
+yP: number
+x: number
+y: number
+border: string
+borderWidth: number
+borderScaleEnabled:booelan
+fillEnabled: boolean
+fill: string
+rotation: number
+points: number[]
PaintService
+store():void
+create():String
+save():String
+undo():List<IShape>
+redo():List<IShape>
+Load():List<IShape>
+edit():void
+getObj():IShape
+delete():void
drawShapes
-idCounter: int
+resetId(): void
+FactoryFn(): void
+drawRect(IShape): Object
+drawCircle(IShape): Object
+drawEllipse(IShape): Object
+drawSquare(IShape): Object
+drawLine(IShape): Object
+drawLine(IShape): Object
AppComponent
+ngOnInit():void
+color(): void
+delete():void
+color(): void
+delete():void
+color(string): void
+undo(): void
+redo(): void
+save(): void
+load(): void
PaintService
+store():void
+create():String
+save():String
+undo():List<IShape>
+redo():List<IShape>
+Load():List<IShape>
+edit():void
+getObj():IShape
+delete():void
Controller
+store(): void
+createShape(): shapeClass
+Undo(): List<shapeClass>
+Redo(): List<shapeClass>
+Save(): String status
+Load(): List<shapeClass>
+copy(): shapeClass
+edit(): shapeClass
+getObj(): shapeClass
ShapeFactory
+createShape():shapeClass
Back-end
shapeClass
-Name: string
-Id: string
-xP: double
-yP: double
-x: double
-y: double
-border: string
-borderWidth: string
-borderScaleEnabled: boolean
-fill: string
-fillEnabled: false
-rotation: double
-points: double[]
+clone(): shapeClass
+compareTo(shapeClass): boolean
Setters and getters for attributes
Triangle
+Triangle()
+Clone()
Ellipse
+Ellipse()
+Clone()
Square
+Square()
+Clone()
Rectangle
+Rectangle()
+Clone()
Line
+Line()
+Clone()
Circle
+Circle()
+Clone()

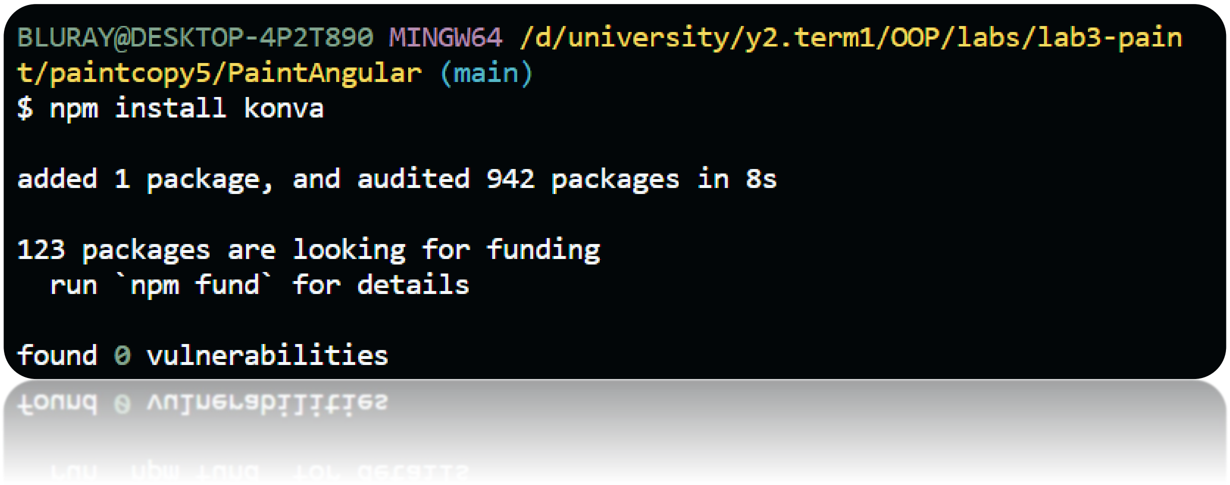
HOW TO RUN:

1) Run npm install:



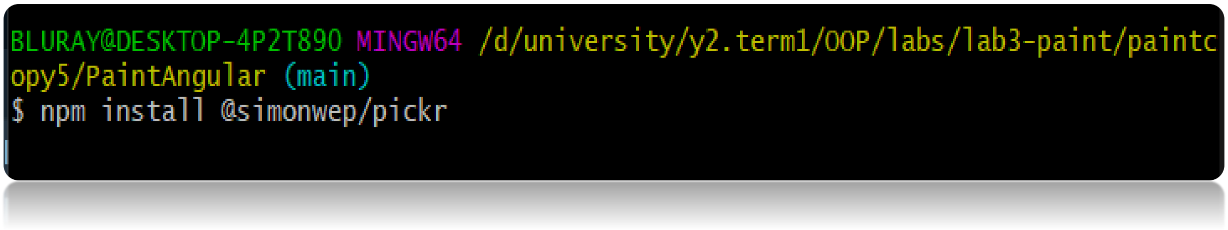
```
BLURAY@DESKTOP-4P2T890 MINGW64 /d/university/y2.term1/OOP/labs/lab3-pain
t/paintcopy5/PaintAngular (main)
$ npm install
```

2) Run npm install konva:



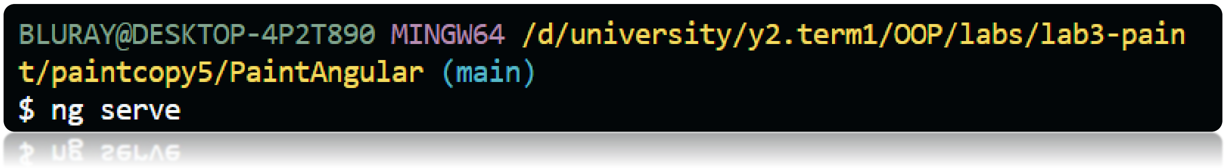
```
BLURAY@DESKTOP-4P2T890 MINGW64 /d/university/y2.term1/OOP/labs/lab3-pain
t/paintcopy5/PaintAngular (main)
$ npm install konva

added 1 package, and audited 942 packages in 8s

123 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

3) Run npm install @simonwep/pickr:



```
BLURAY@DESKTOP-4P2T890 MINGW64 /d/university/y2.term1/OOP/labs/lab3-paint/paintc
opy5/PaintAngular (main)
$ npm install @simonwep/pickr
```

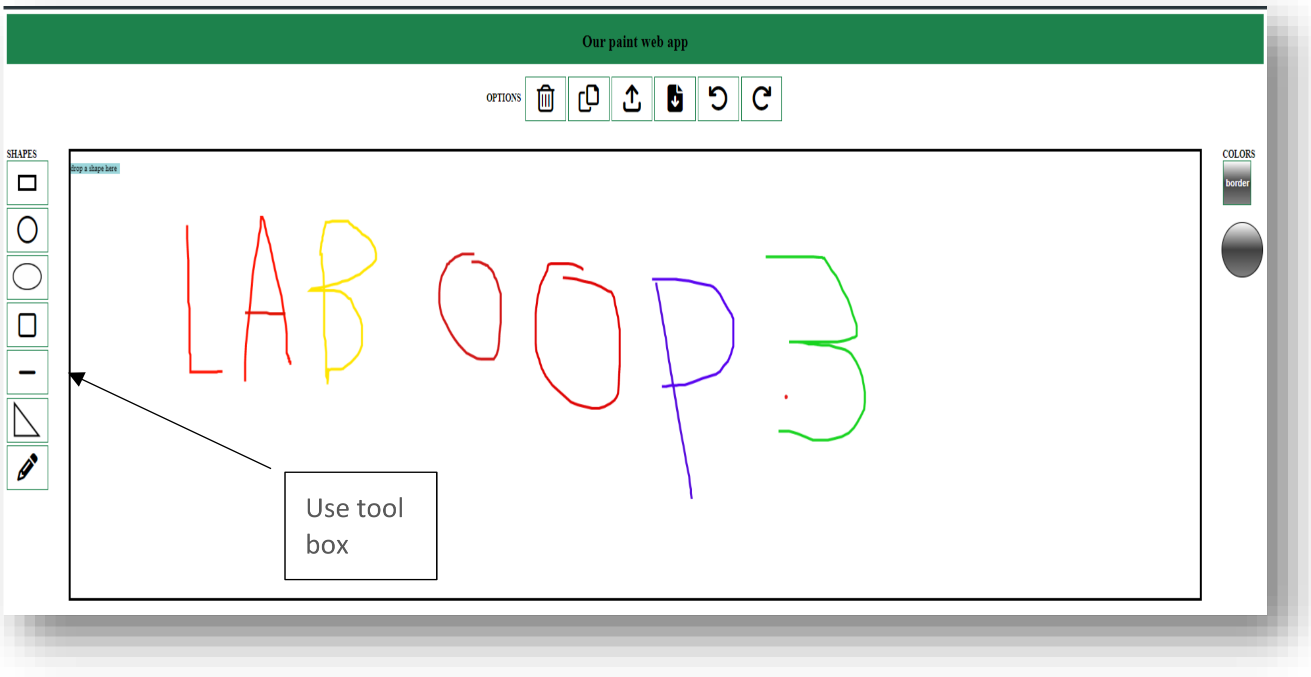
4) Run ng serve:



```
BLURAY@DESKTOP-4P2T890 MINGW64 /d/university/y2.term1/OOP/labs/lab3-pain
t/paintcopy5/PaintAngular (main)
$ ng serve
```

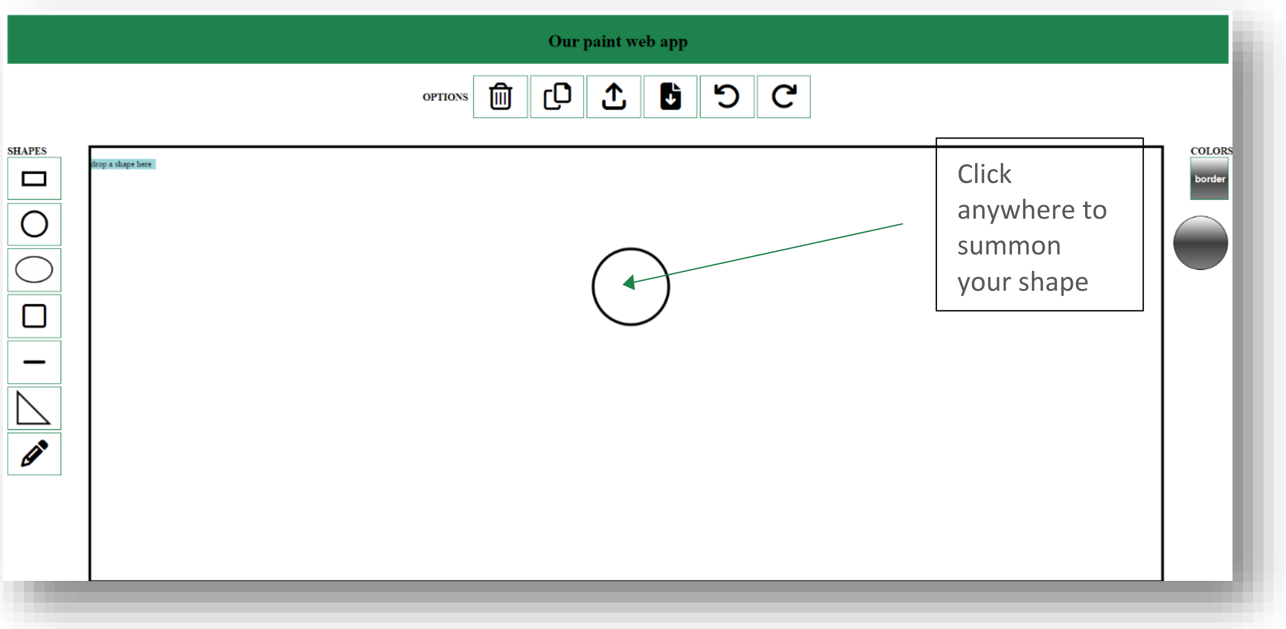
HOW TO USE

1) Choose to draw any shape or line or use the free hand tool to draw like as if you have a pen



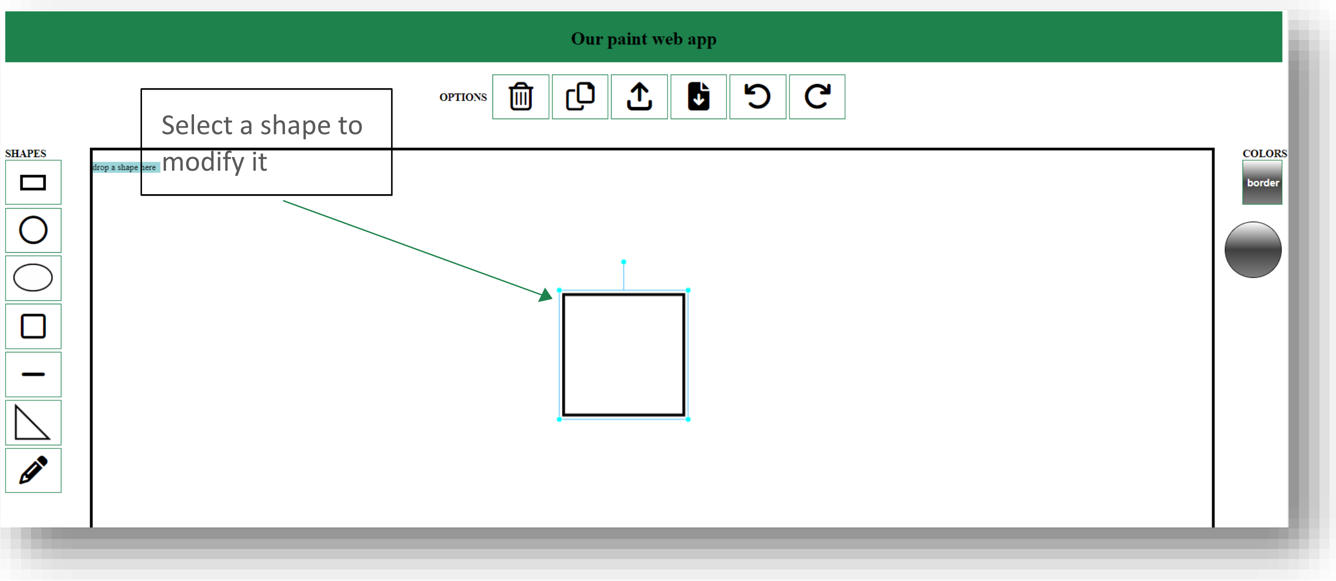
Use tool box

2) After choosing a shape click anywhere on the board to insert that shape.



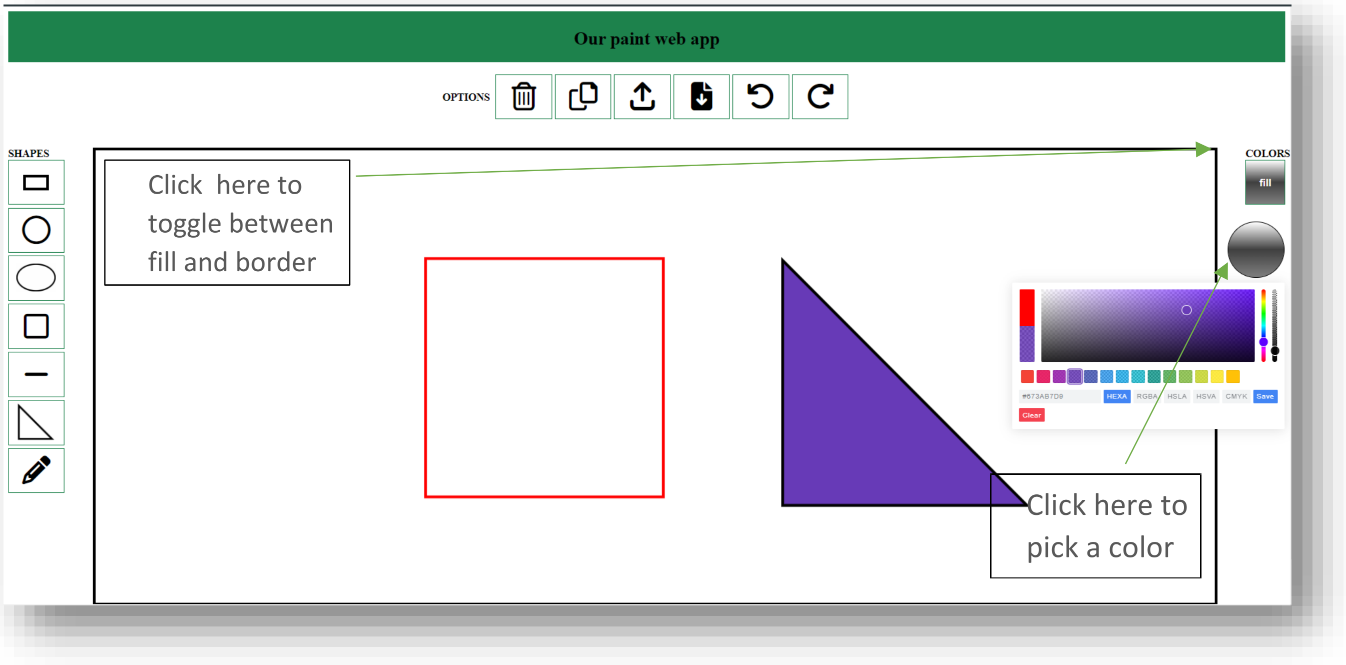
Click anywhere to summon your shape

3) If you want to resize, move or rotate the shape click on one of its edges and a blue outline will appear. Use it for modification.



Select a shape to modify it

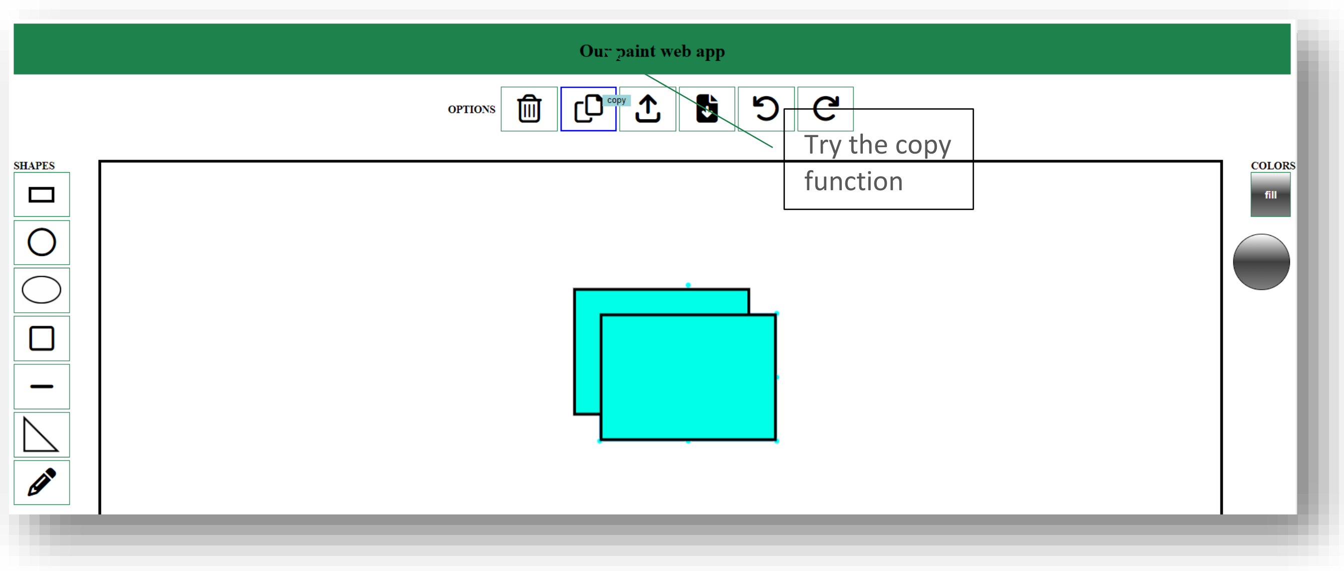
4) For coloring, select any drawing first and choose its color from the coloring bar on the right side. (Border and Fill are a toggle button)



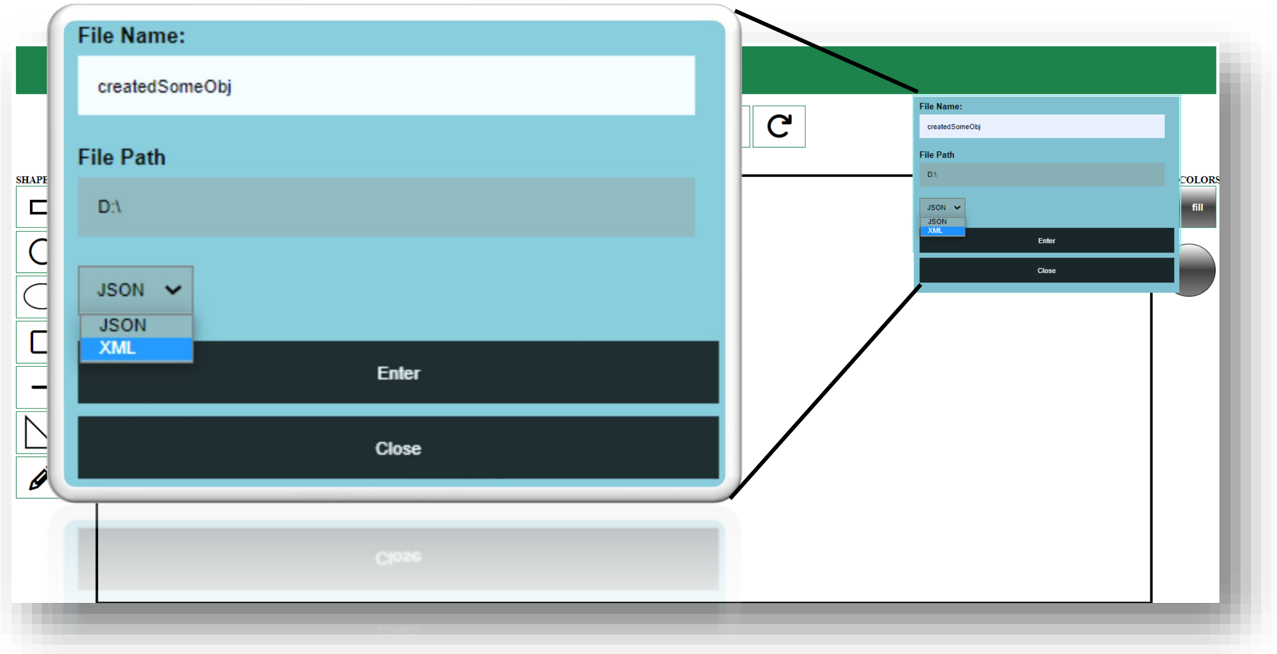
Click here to toggle between fill and border

Click here to pick a color

5) You can also copy or delete the figure by using the options bar at the top after selecting it.



6) Finally, to save your artwork. Click on the Save button in the options bar. A form will appear requesting a name for the file, the directory which you'd like to save your file and a format of two (JSON or XML)


File Name:

createdSomeObj

File Path

D:\

7) Loading your file will be similar to saving it. The board will appear with all drawings and you can undo or redo from that instance and continue your art.

PROJECT LINKS:

- https://drive.google.com/drive/folders/17h-UsV5q4yNNEHrMo7mzt1Yy5q_WsaUh?usp=sharing
- https://github.com/AbdelrahmanWael2/Paint.git

REFERENCES

- Prototyping: https://refactoring.guru/designpatterns/prototype/java/example#:~:text=Prototype%20is%20a%20creational%20design,their% 20concrete%20classes%20are%20unknown.

- POST request sending JSON: https://reqbin.com/code/javascript/wzp2hxwh/javascript-postrequestexample#:~:text=To%20send%20an%20HTTP%20POST,subscribe%20to%20the%20onrea dystatechange%20event.

- Konva drawing: https://www.youtube.com/watch?v=wXiXLS9eQLw

- Konva dragging: https://stackoverflow.com/questions/53138815/get-the-name-or-id-ofelement-i-drag-it

- Konva start: https://konvajs.org/docs/index.html

- Java to JSON: https://www.youtube.com/watch?v=pv1VCFWTP-I

- Color Picker: https://github.com/Simonwep/pickr