In [1]:
```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

1. Read in Image, Grayscale

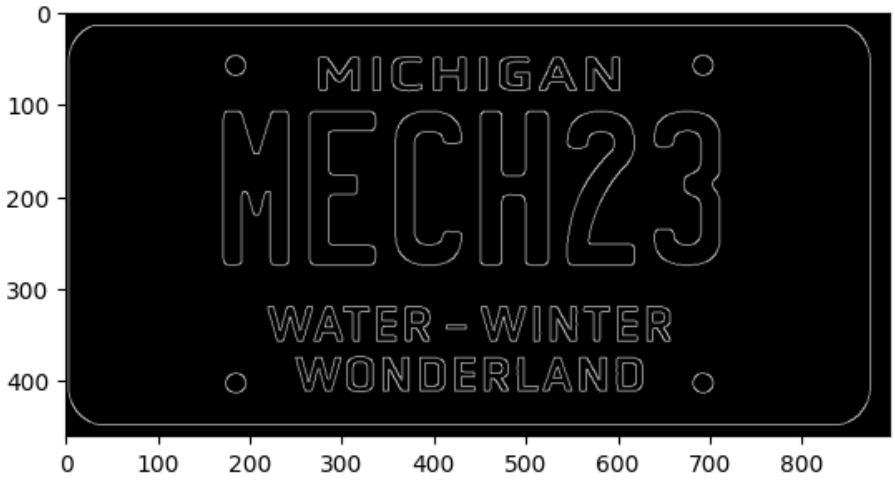
In [2]:
```
img = cv2.imread('../data/generated4.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray,cv2.COLOR_BGR2RGB))
```

Out[2]:
```
<matplotlib.image.AxesImage at 0x1761e9670>
```

2. Apply filter and find edges for localization

In [3]:
```
bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bfilter, 20, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x17f77ede0>
```



3. Find contours and apply masks

In [4]:
```
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

In [5]:
```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

In [6]:
```
contour
```

Out[6]:
```
array([[[269, 108]],

       [[331, 108]],

       [[332, 109]],

       [[333, 109]],

       [[336, 112]],

       [[336, 113]],

       [[337, 114]],

       [[337, 123]],

       [[336, 124]],

       [[336, 125]],

       [[335, 126]],

       [[335, 127]],

       [[334, 128]],

       [[333, 128]],

       [[332, 129]],

       [[290, 129]],

       [[289, 130]],

       [[287, 130]],

       [[286, 131]],

       [[286, 173]],

       [[287, 174]],

       [[287, 175]],

       [[288, 176]],

       [[289, 176]],

       [[290, 177]],

       [[314, 177]],

       [[316, 179]],

       [[316, 180]],

       [[317, 181]],

       [[317, 194]],

       [[316, 195]],

       [[316, 196]],

       [[314, 198]],

       [[290, 198]],

       [[289, 199]],

       [[288, 199]],
```

```
       [[287, 200]],

       [[287, 201]],

       [[286, 202]],

       [[286, 250]],

       [[287, 251]],

       [[287, 252]],

       [[288, 252]],

       [[289, 253]],

       [[330, 253]],

       [[331, 254]],

       [[333, 254]],

       [[336, 257]],

       [[336, 259]],

       [[337, 260]],

       [[337, 267]],

       [[336, 268]],

       [[336, 271]],

       [[333, 274]],

       [[332, 274]],

       [[331, 275]],

       [[271, 275]],

       [[270, 274]],

       [[269, 274]],

       [[266, 271]],

       [[266, 270]],

       [[265, 269]],

       [[265, 112]]], dtype=int32)
```

In [7]:
```
location
```

In [8]:
```
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0, 255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)

plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```

```
---------------------------------------------------------------------------
error                                     Traceback (most recent call last)
Cell In[8], line 2
      1 mask = np.zeros(gray.shape, np.uint8)
----> 2 new_image = cv2.drawContours(mask, [location], 0, 255, -1)
      3 new_image = cv2.bitwise_and(img, img, mask=mask)
      5 plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))

error: OpenCV(4.10.0) /Users/xperience/GHA-Actions-OpenCV/_work/opencv-python/opencv-python/openc
v/modules/imgproc/src/drawing.cpp:2433: error: (-215:Assertion failed) p.checkVector(2, CV_32S) >=
0 in function 'fillPoly'
```

In [ ]:
```
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

In [ ]:
```
plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

4. Use Easy OCR to Read Text

In [ ]:
```
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

5. Render Result

In [ ]:
```
text = result[0][-2]
font = cv2.FONT_HERSHEY_SIMPLEX
res = cv2.putText(img, text=text, org=(approx[0][0][0], approx[1][0][1]+60), fontFace=font, fontSc
                  thickness=2, lineType=cv2.LINE_AA)
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0,255,0),3)
plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))
```