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

1. Read in Image, Grayscale

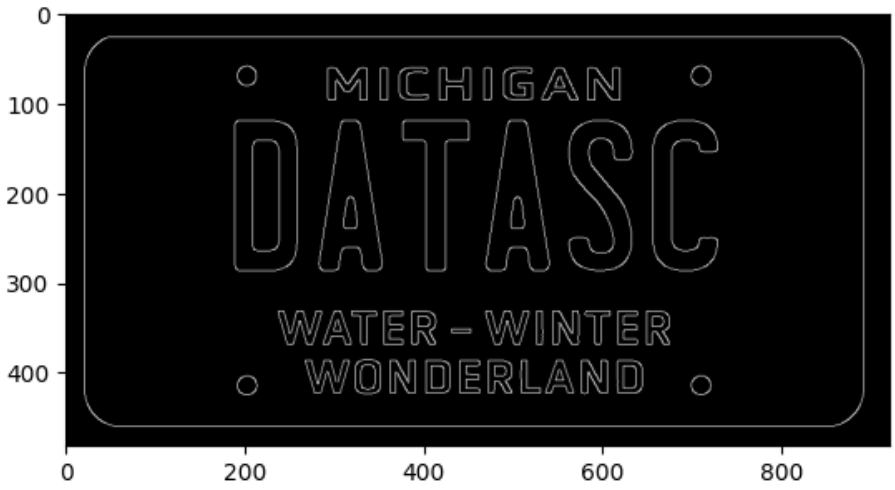
In [2]:
```
img = cv2.imread('../data/generated3.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray,cv2.COLOR_BGR2RGB))
```

Out[2]:
```
<matplotlib.image.AxesImage at 0x17f7c6b40>
```

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 0x289668680>
```



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([[[588, 121]],

       [[589, 120]],

       [[608, 120]],

       [[609, 121]],

       [[612, 121]],

       [[613, 122]],

       [[615, 122]],

       [[616, 123]],

       [[617, 123]],

       [[618, 124]],

       [[619, 124]],

       [[621, 126]],

       [[622, 126]],

       [[626, 130]],

       [[626, 131]],

       [[629, 134]],

       [[629, 135]],

       [[630, 136]],

       [[630, 137]],

       [[631, 138]],

       [[631, 140]],

       [[632, 141]],

       [[632, 144]],

       [[633, 145]],

       [[633, 152]],

       [[634, 153]],

       [[634, 156]],

       [[633, 157]],

       [[633, 160]],

       [[629, 164]],

       [[617, 164]],

       [[614, 161]],

       [[614, 160]],

       [[613, 159]],

       [[613, 152]],

       [[612, 151]],
```

```
[[612, 148]],
[[611, 147]],
[[611, 145]],
[[608, 142]],
[[607, 142]],
[[606, 141]],
[[604, 141]],
[[603, 140]],
[[594, 140]],
[[593, 141]],
[[592, 141]],
[[591, 142]],
[[590, 142]],
[[587, 145]],
[[587, 147]],
[[586, 148]],
[[586, 151]],
[[585, 152]],
[[585, 159]],
[[586, 160]],
[[586, 164]],
[[587, 165]],
[[587, 167]],
[[588, 168]],
[[588, 170]],
[[591, 173]],
[[591, 174]],
[[596, 179]],
[[596, 180]],
[[601, 185]],
[[601, 186]],
[[606, 191]],
[[606, 192]],
[[612, 198]],
[[612, 199]],
[[618, 205]],
```

[[618, 206]],
[[621, 209]],
[[621, 210]],
[[622, 211]],
[[622, 212]],
[[624, 214]],
[[624, 215]],
[[625, 216]],
[[625, 217]],
[[626, 218]],
[[626, 219]],
[[627, 220]],
[[627, 222]],
[[628, 223]],
[[628, 225]],
[[629, 226]],
[[629, 228]],
[[630, 229]],
[[630, 232]],
[[631, 233]],
[[631, 237]],
[[632, 238]],
[[632, 242]],
[[633, 243]],
[[633, 261]],
[[632, 262]],
[[632, 265]],
[[631, 266]],
[[631, 268]],
[[630, 269]],
[[630, 270]],
[[629, 271]],
[[629, 272]],
[[627, 274]],
[[627, 275]],
[[621, 281]],

[[620, 281]],
[[618, 283]],
[[616, 283]],
[[614, 285]],
[[611, 285]],
[[610, 286]],
[[604, 286]],
[[603, 287]],
[[592, 287]],
[[591, 286]],
[[586, 286]],
[[585, 285]],
[[583, 285]],
[[582, 284]],
[[580, 284]],
[[579, 283]],
[[578, 283]],
[[577, 282]],
[[576, 282]],
[[574, 280]],
[[573, 280]],
[[569, 276]],
[[569, 275]],
[[567, 273]],
[[567, 272]],
[[566, 271]],
[[566, 270]],
[[565, 269]],
[[565, 267]],
[[564, 266]],
[[564, 261]],
[[563, 260]],
[[563, 255]],
[[567, 251]],
[[569, 251]],
[[570, 250]],

[[579, 250]],
[[580, 251]],
[[583, 251]],
[[584, 252]],
[[584, 253]],
[[585, 254]],
[[585, 257]],
[[586, 258]],
[[586, 260]],
[[587, 261]],
[[587, 263]],
[[590, 266]],
[[592, 266]],
[[593, 267]],
[[605, 267]],
[[606, 266]],
[[608, 266]],
[[612, 262]],
[[612, 259]],
[[613, 258]],
[[613, 250]],
[[612, 249]],
[[612, 245]],
[[611, 244]],
[[611, 241]],
[[610, 240]],
[[610, 238]],
[[609, 237]],
[[609, 235]],
[[608, 234]],
[[608, 233]],
[[607, 232]],
[[607, 230]],
[[606, 229]],
[[606, 228]],
[[605, 227]],

```
[[605, 226]],
[[604, 225]],
[[604, 224]],
[[603, 223]],
[[603, 222]],
[[602, 221]],
[[602, 220]],
[[600, 218]],
[[600, 217]],
[[598, 215]],
[[598, 214]],
[[595, 211]],
[[595, 210]],
[[580, 195]],
[[580, 194]],
[[574, 188]],
[[574, 187]],
[[571, 184]],
[[571, 183]],
[[568, 180]],
[[568, 179]],
[[567, 178]],
[[567, 177]],
[[566, 176]],
[[566, 174]],
[[565, 173]],
[[565, 170]],
[[564, 169]],
[[564, 163]],
[[563, 162]],
[[563, 149]],
[[564, 148]],
[[564, 143]],
[[565, 142]],
[[565, 140]],
[[566, 139]],
```

```
       [[566, 137]],

       [[567, 136]],

       [[567, 135]],

       [[569, 133]],

       [[569, 132]],

       [[575, 126]],

       [[576, 126]],

       [[578, 124]],

       [[579, 124]],

       [[580, 123]],

       [[581, 123]],

       [[582, 122]],

       [[584, 122]],

       [[585, 121]]], 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))
```