In [1]:

```
# Basic overview of pytesseract. Python-tesseract is an optical character recognition
 (OCR) tool for python.
# That is, it will recognize and “read” the text embedded in images.
# Python-tesseract is a wrapper for Google’s Tesseract-OCR Engine.
# It is also useful as a stand-alone invocation script to tesseract,
# as it can read all image types supported by the Pillow and Leptonica imaging librarie
s, including jpeg, png,
# gif, bmp, tiff, and others. Additionally, if used as a script,
# Python-tesseract will print the recognized text instead of writing it to a file.
```

In [9]:

```
import pytesseract
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

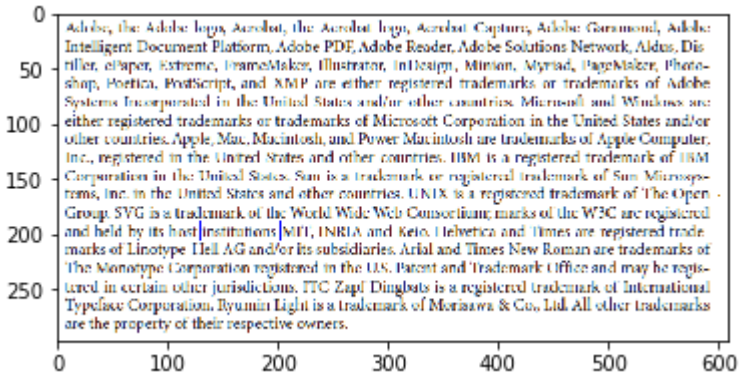
```
#the address of the tesseract present in your system
pytesseract.pytesseract.tesseract_cmd = r'C:\\Program Files\\Tesseract-OCR\\tesseract.e
xe'
```

In [5]:

```
# recognizing and extracing the text of any computerized document is efficient, but rec
ognizing the
# the text from an image or handwriting is tricky. We need to remove all the noise from
the image to make it efficient
```

In [17]:

```
# lets check for computerized image
img_color = cv2.imread('E:\\Learning\\Computer_Vision\\Digit Recognition\\text3.png')
plt.imshow(img_color)
plt.show()
```



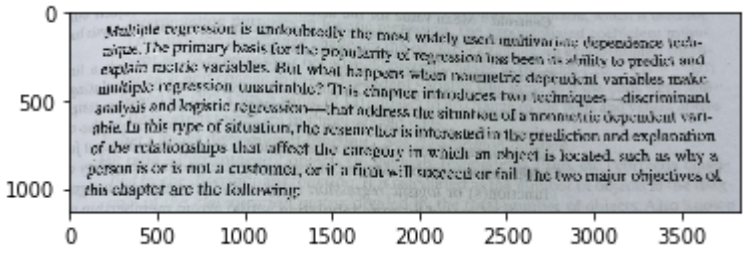
In [15]:

```
# image_to_string function is use for extracting text from image
# the text is extracted quite easily
print(pytesseract.image_to_string(img_color))
```

```
Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Capture, Adobe G
aramond, Adobe
Intelligent Document Platform, Adobe PDF, Adobe Reader, Adobe Solutions Ne
twork, Aldus, Dis-
tiller, ePaper, Extreme, FrameMaker, Illustrator, InDesign, Minion, Myria
d, PageMaker, Photo-
shop, Poetica, PostScript, and XMP are either registered trademarks or tra
demarks of Adobe
‘Systems Incorporated in the United States and/or other countries. Microso
ft and Windows are
either registered trademarks or trademarks of Microsoft Corporation in the
United States and/or
other countries. Apple, Mac, Macintosh, and Power Macintosh are trademarks
of Apple Computer,
Inc,, registered in the United States and other countries. IBM is a regist
ered trademark of IBM
Corporation in the United States. Sun is a trademark or registered tradema
rk of Sun Microsys-
tems, Inc. in the United States and other countries. UNIX is a registered
trademark of The Open
Group. SVG is a trademark of the World Wide Web Consortium; marks of the W
3C are registered
and held by its host[institutions MIT, INRIA and Keio. Helvetica and Times
are registered trade-
marks of Linotype-Hell AG and/or its subsidiaries. Arial and Times New Rom
an are trademarks of
‘The Monotype Corporation registered in the US. Patent and Trademark Offic
e and may be regis-
tered in certain other jurisdictions. ITC Zapf Dingbats is a registered tr
ademark of International
‘Typeface Corporation. Ryumin Light is a trademark of Morisawa & Co., Ltd.
All other trademarks
are the property of their respective owners.
```

In [19]:

```
# but if we look at any handwritten document or photographed image , it is a bit diffic
ult
img_color = cv2.imread('E:\\Learning\\Computer_Vision\\Digit Recognition\\text3.jpg')
plt.imshow(img_color)
plt.show()
```



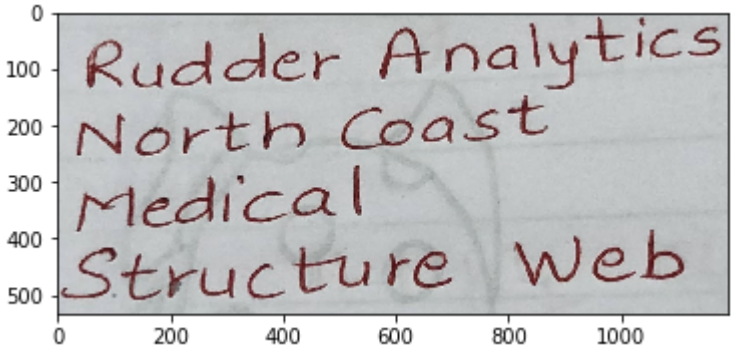
In [21]:

```
# lets try to extract without cleaing the image
print(pytesseract.image_to_string(img_color))
# output is gibrish, we need to clean the image first
```

```
y the most widely used multivariate dependence tech-

Multiple regression is undoubted]
Opularity of regression has been its ability to predict and

nique. The primary basis for the p

explain metric variables. But what happens when nonmetric dependent variab
les make

multiple regression unsuitable? This chapter introduces two techniques—dis
criminant

analysis and logistic regression—that address the Situation of a nonmetric
dependent vari-

able. In this type of situation, the researcher is interested in the predi
ction and explanation
he category in which an object is located, such as why a

of the relationships that affect t
person Is or is not a customer, or if a firm will succeed or fail. The two
major objectives of

this chapter are the following:
```

In [43]:

```
# not lets try hand written image
img = cv2.imread('E:\\Learning\\Computer_Vision\\Digit Recognition\\shand.jpg')
plt.imshow(img)
plt.show()
```



In [44]:

```
# lets try to extract without cleaning the image
print(pytesseract.image_to_string(img))
```

```
Rudder Arnalytics
North Goast

medica |
Structure Web
```

In [36]:

```
# here the spelling of 'medical' is categorized differently
```

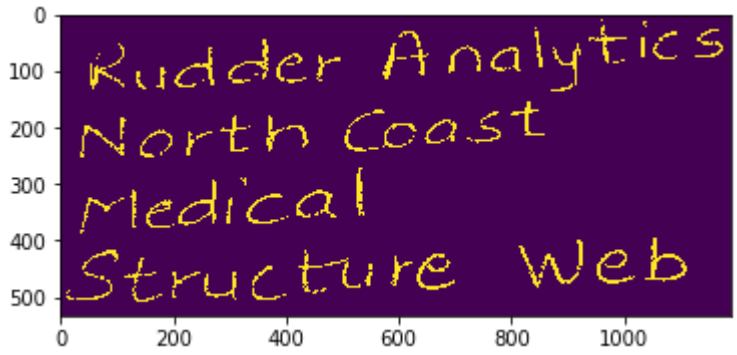
In [45]:

```
# lets the clean the image and check the output
blue_lower=np.array([25,10,10],np.uint8)
blue_upper=np.array([255,60,60],np.uint8)

mask = cv2.inRange(img, blue_lower, blue_upper)

print(pytesseract.image_to_string(mask))
```

```
Rudder Analytics
North Coast

ae leres
penne Web
```

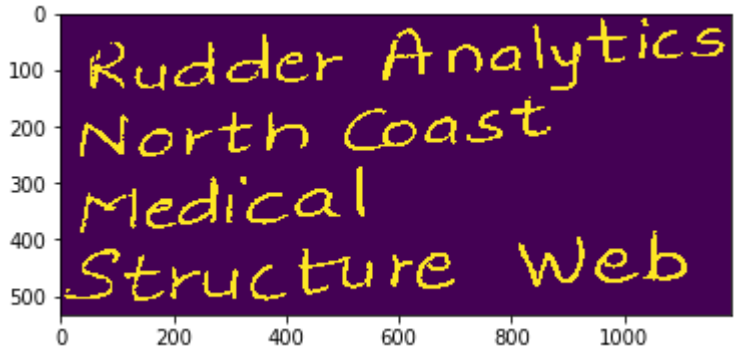
In [47]:

```
# extracting only blue color and making everything else black
blue_lower=np.array([25,10,10],np.uint8)
blue_upper=np.array([255,60,60],np.uint8)
mask = cv2.inRange(img, blue_lower, blue_upper)
plt.imshow(mask)
plt.show()
```



In [50]:

```
# dialating the image
kernel = np.ones((3,3), np.uint8)
img_dilation = cv2.dilate(mask, kernel, iterations=1)
plt.imshow(img_dilation)
plt.show()
```



In [51]:

```
# lets try to run on this image
print(pytesseract.image_to_string(img_dilation))
```

```
Rudder Analytics
North Coast

medical
Structure Web
```

In [52]:

```
# yeeahhh, its working
# but still the accuracy of pytesseract is a lot dependent on the quality of image, so
 we should not entirely reply on
# pytesseract
# we need to create our own model
```