Experiment 3:

Consider one application and caliculate the distance between two pixels

Here we are considering a length of wave in image

1. Uploading the image:

In [3]:
```
from google.colab import files
files=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving tst.jpg to tst.jpg
```

In this the files are uploaded to the local runtime environment of google colaboratory.These files are locally saved into the environment till the runtime and when the runtime disconnects the user need to upload the files again.

2.Importing the libraries

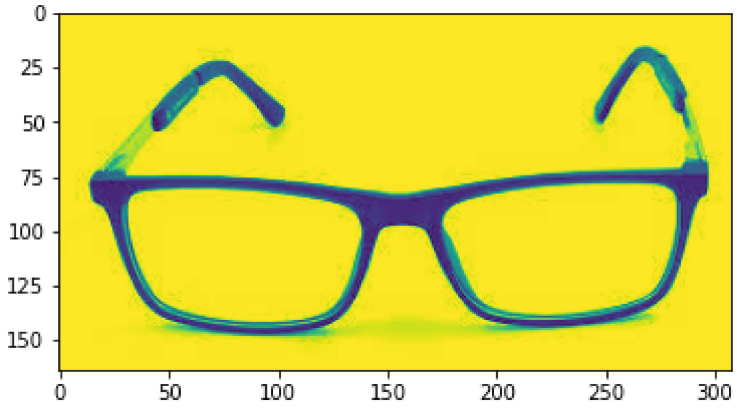
In [1]:
```
import cv2
import matplotlib.pyplot as plt
import numpy as np
```

Calculating the length needs cv2,matplotlib and numpy as the necessary libraries to be imported.Cv2 is used to perform all the image processing,Matplot is used for plotting the graphs and mentioning the measurement of the object in the image.All the pixels are stored in an numpy array of the objects of the image so numpy is imported as np

3.Reading the image:

In [4]:
```
img=cv2.imread('tst.jpg',0)
plt.imshow(img)
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x7ff31bc08550>
```

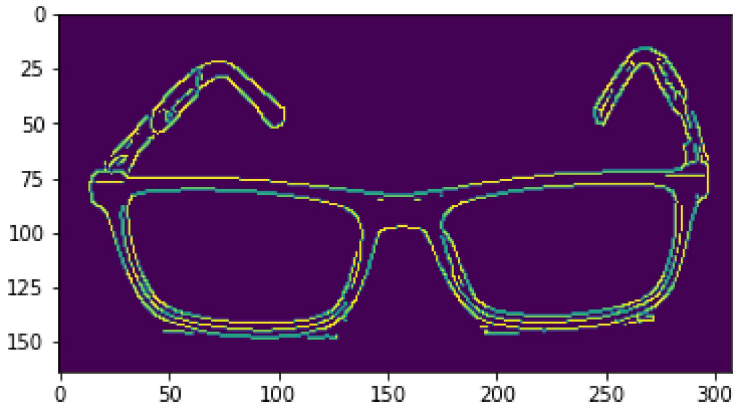


The image uploaded in the runtime in google colab is only for the runtime.To access the image in the program we must read the uploaded image using opencv.imread method in open cv is used to read the image in the google colab for program usage.

4.Edge detection:

In [5]:
```
edges = cv2.Canny(img,100,200)
plt.imshow(edges)
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x7ff31bb92050>
```



For measuring the length of the image it's important to detect the edges as the edges will help in finding the starting and ending of the pixel value. Edge detection is done with the help of a canny method present in the opencv library.Detected edges along with the image is used to locat the end and stating positions of the pixels which when processed will be helpful in find the length of the object.

In [6]:
```
img.shape
```

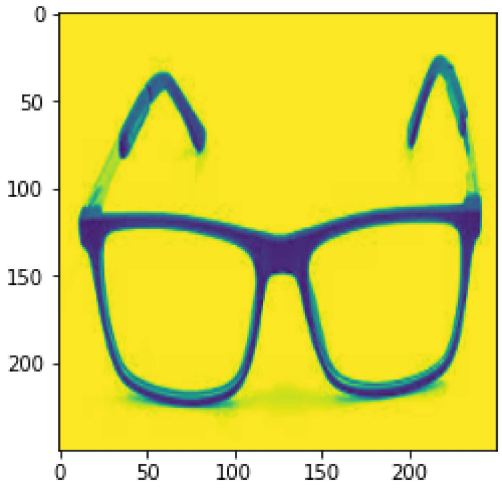
Out[6]:
```
(164, 308)
```

5. Reshaping to equal size

In [7]:
```
img=cv2.resize(img,(250,250))
```

In [8]:
```
plt.imshow(img)
```

Out[8]: <matplotlib.image.AxesImage at 0x7ff31bb06450>



5. Finding out two edge points (pixels)

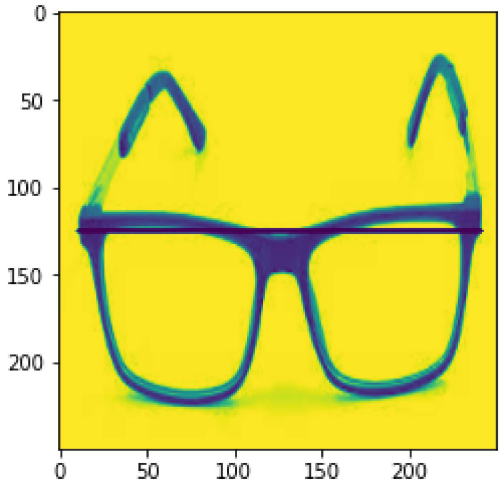
In [9]:
```
p1,p2=[],[]
for i in range(len(img)-1,-1,-1):
  #print(i,end=' ')
  for j in range(len(img[i])-1,-1,-1):
    #print(j,end=' ')
    if img[j][i]<225:
      p1=(i,j)
      break
  if not p1==[]:
    break

for i in range(len(img)):
  for j in range(len(img[i])-1,-1,-1):
    if img[j][i]<235:
      p2=(i,j)
      break
  if not p2==[]:
    break

p1,p2
```

Out[9]: ((241, 125), (11, 124))

7. Connecting two pixels

In [10]:
```
p2=(p2[0],p1[1])
cv2.line(img,p1,p2,(0,0,255),2)
plt.imshow(img)
```

Out[10]: <matplotlib.image.AxesImage at 0x7ff31baf5590>



6.Calculating the length of wave in image.

In [11]:
```
inches = 30     # 1 inch = 30 pixels
pixel_count = abs(p1[0]-p2[0])
length = pixel_count/inches
length
```

Out[11]: 7.666666666666667

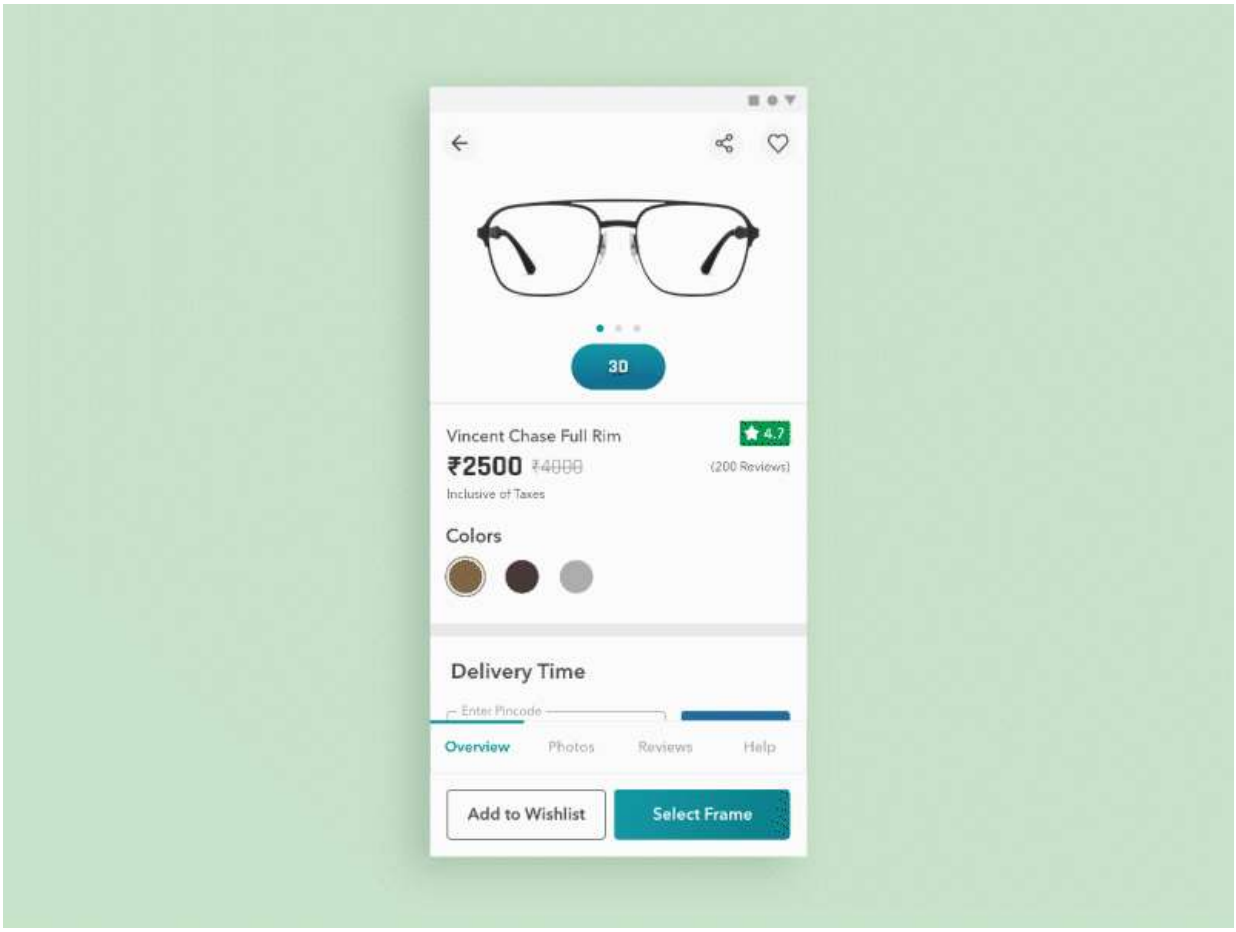
To calculate the length of the image firstly we have to define the measuring scale for the pixel.The measuring scale in this case is defined as 1 inch equals to 30pixels meaning that in every inch 30 pixels are present,so knowing stating and ending pixel location and dividing by the measuring scale the length of the object id calculated.

Real World Application:

Lenskart 3d Try on & Lenskart Frame size

In [12]:
```
from IPython.display import Image
Image(url='https://cdn.dribbble.com/users/4189149/screenshots/14948347/dribbble_shot_2_
```

Out[12]:


3D
Vincent Chase Full Rim
4.7
₹2500 ₹4000
(200 Reviews)
Inclusive of Taxes
Colors
Delivery Time
Enter Pincode
Overview
Photos
Reviews
Help
Add to Wishlist
Select Frame