Section 2: Implement Haar Cascade Classifier using OpenCV

Step 1: Import the modules

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

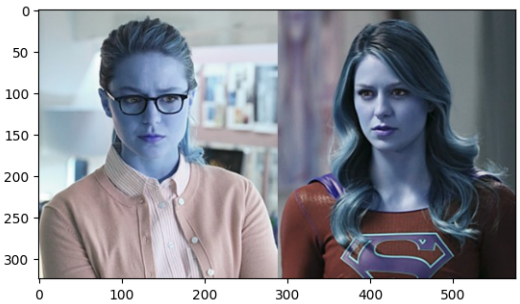
Step 2: Load the face detection trained model

In [2]:
```
face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
```

Step 3: Read and convert the image

In [3]:
```
img = cv2.imread("img/sample.jpg")
if img is None:
    raise FileNotFoundError("img/sample.jpg not found. Make sure the image exists in the 'img' folder.")
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

In [4]:
```
plt.imshow(img)
img_copy1 = img.copy()
img_copy2 = img.copy()
img_copy3 = img.copy()
```



Step 4: Define detect_face function

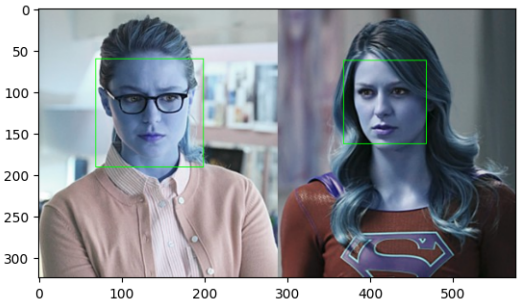
In [5]:
```
def detect_face(img):
    face_img = img.copy()
    face_rect = face_cascade.detectMultiScale(face_img,
                                              scaleFactor = 1.1,
                                              minNeighbors = 5)

    for (x, y, w, h) in face_rect:
        # red box for face
        cv2.rectangle(face_img, (x, y),
                      (x + w, y + h), (0, 255, 0), 1)
    return face_img
```

Step 5: Invoke function and display image

In [6]:
```
face = detect_face(img_copy1)
plt.imshow(face)
cv2.imwrite('face.jpg', face)
```

Out[6]: True



Step 6: Load the eye detection trained model

In [7]:
```
eye_cascade = cv2.CascadeClassifier("haarcascade_eye.xml")
```

Step 7: Define detect_eyes function

In [8]:
```
# create a function to detect eyes
def detect_eyes(img):
    eye_img = img.copy()
    eye_rect = eye_cascade.detectMultiScale(eye_img,
                                            scaleFactor = 1.1,
                                            minNeighbors = 5)

    for (x, y, w, h) in eye_rect:
        # green box for eyes
        cv2.rectangle(eye_img, (x, y),
                      (x + w, y + h), (255, 0, 0), 1)
    return eye_img
```

Step 8: Invoke function and display image

In [9]:
```
# combine face and eye detection
eyes_face = detect_face(img_copy3)
eyes_face = detect_eyes(eyes_face)
plt.imshow(eyes_face)
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x2416dc5a6d0>
```