```
'''
Program name : SH-Convertor
Program Version : 1.17 Beta
This Code By : Hasan Mohammed Alshikh
Facebook Link : https://www.facebook.com/hasan.muhmet.syhe
Email Address : hasanalshik65@gmail.com

You can use this code to convert image files to PDF file.
Plz before you run this code you need to install libraries [ Pillow - PyQt5]

to install this libraries on Linux :
pip3 install Pillow [PIL]
apt install python3-pyqt5

to install this libraries on windows :
pip install Pillow
pip install PyQt5

Thanks for Shahed.
'''
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from sys import *
from PIL import Image
from os import *

class qShahed(QMainWindow):
    def __init__(self):
        super().__init__()
        self.path = getcwd()
        self.setWindowTitle('SH-Convertor')
        self.resize(800,500)
        # parts
        self.qshahed_menubar()
        self.program_body()
    def qshahed_menubar(self):

        menu_widget = self.menuBar()
        # menus
        file_menu = menu_widget.addMenu('File')
        view_menu = menu_widget.addMenu('View')
        edit_menu = menu_widget.addMenu('Edit')
        help_menu = menu_widget.addMenu('Help')
        # file actions
        open_action = QAction('Open image', self)
        open_action.setShortcut("Ctrl+O")
        open_action.triggered.connect(self.open_file)
        savepdf_action = QAction('Save as pdf', self)
        savepdf_action.setShortcut("Ctrl+Alt+P")
        savepdf_action.triggered.connect(self.img_pdf)
        file_menu.addAction(open_action)
        file_menu.addAction(savepdf_action)
        # help actions
        about_action = QAction('About Program', self)
        about_action.triggered.connect(self.about)
        help_menu.addAction(about_action)


    def program_body(self):
        body_widget = QWidget()
        main_layout = QVBoxLayout()
        body_widget.setLayout(main_layout)
        # set widget inside self
        self.setCentralWidget(body_widget)

        # layouts
        toplayout = QHBoxLayout()
        bottomlayout = QHBoxLayout()
        # add this layouts to main_layout
        main_layout.addLayout(toplayout)
        main_layout.addLayout(bottomlayout)

        self.img_label_display = QLabel()
        self.img_label_display.setAlignment(Qt.AlignLeft | Qt.AlignTop)
        self.img_label_display.setScaledContents(True)
        # get background img
        img_back = QPixmap('background.png')
        img_back = img_back.scaled(self.width(), self.height())
        self.img_label_display.setPixmap(img_back)
        # add this label iside toplayout
        toplayout.addWidget(self.img_label_display)
        self.status_bar(bottomlayout)

    def status_bar(self, layout):
        status_box = QGroupBox("Statusbar")
        status_layout = QHBoxLayout()
        status_box.setLayout(status_layout)
        layout.addWidget(status_box)
        self.width_img_label = QLabel("Image Width : none | ")
        self.height_img_label = QLabel("Image Height : none | ")
        self.format_img_label = QLabel("Image Format : none")

        status_layout.addWidget(self.width_img_label)
        status_layout.addWidget(self.height_img_label)
        status_layout.addWidget(self.format_img_label)

        spacer = QSpacerItem(50, 10, QSizePolicy.Minimum, QSizePolicy.MinimumExpanding)
        status_layout.addItem(spacer)

        self.progressline = QProgressBar()
        self.progressline.setValue(0)
        status_layout.addWidget(self.progressline)

    def open_file(self):
        self.progressline.setValue(0)
        self.opened_file = QFileDialog.getOpenFileName(self, "Select File", self.path)
        format_file = path.splitext(self.opened_file[0])
        pix_img = QPixmap(self.opened_file[0])
        if(self.opened_file[0] == ''):
            pass
        else:
            self.pix_img = pix_img.scaled(self.width(), self.height())
            self.img_label_display.setPixmap(self.pix_img)
            self.resize(self.pix_img.width(), self.pix_img.height())

        self.width_img_label.setText(f'Image Width : {pix_img.width()} px | ')
        self.height_img_label.setText(f'Image Width : {pix_img.height()} px | ')
        self.format_img_label.setText(f'Image Format : {format_file[1]}')
    def img_pdf(self):
        self.progressline.setValue(20)
        text = 'Please Enter your pdf file name !'
        self.name = QInputDialog.getText(self,"Convert Image", text, QLineEdit.Normal, "meryem")
        if(self.name[1] == True):
            myImge = Image.open(f'{self.opened_file[0]}')
            myImge_pdf = myImge.convert('RGB')
            myImge_pdf.save(f'{self.path}/{self.name[0]}.pdf')
            self.progressline.setValue(100)
        elif(self.name[1] == False):
            QMessageBox.about(self, 'Save As Pdf', 'Operation Canceled.')
        else:
            QMessageBox.about(self, 'Error', 'Operation Canceled.')

    def about(self):
        about_text = '''
        <b><h3>SH-Convertor</h3></b><br>
        You can use this program to convert format files to another one<br>
        you can convert more then 15 format file just using this program<br><br>
        <font color="#0087ff">Program Version : 1.17 Beta</font>
        '''
        QMessageBox.about(self, 'About SH-Convertor', about_text)


def main():
    app_sh = QApplication(argv)
    app_sh.setStyle('Fusion')
    window_sh = qShahed()
    window_sh.show()
    exit(app_sh.exec_())
if __name__ == '__main__':
    main()
```