# Homework # 4

## 13016215 Software Engineering Principle Laboratory Software Engineering Program

## Faculty of Engineering, KMITL

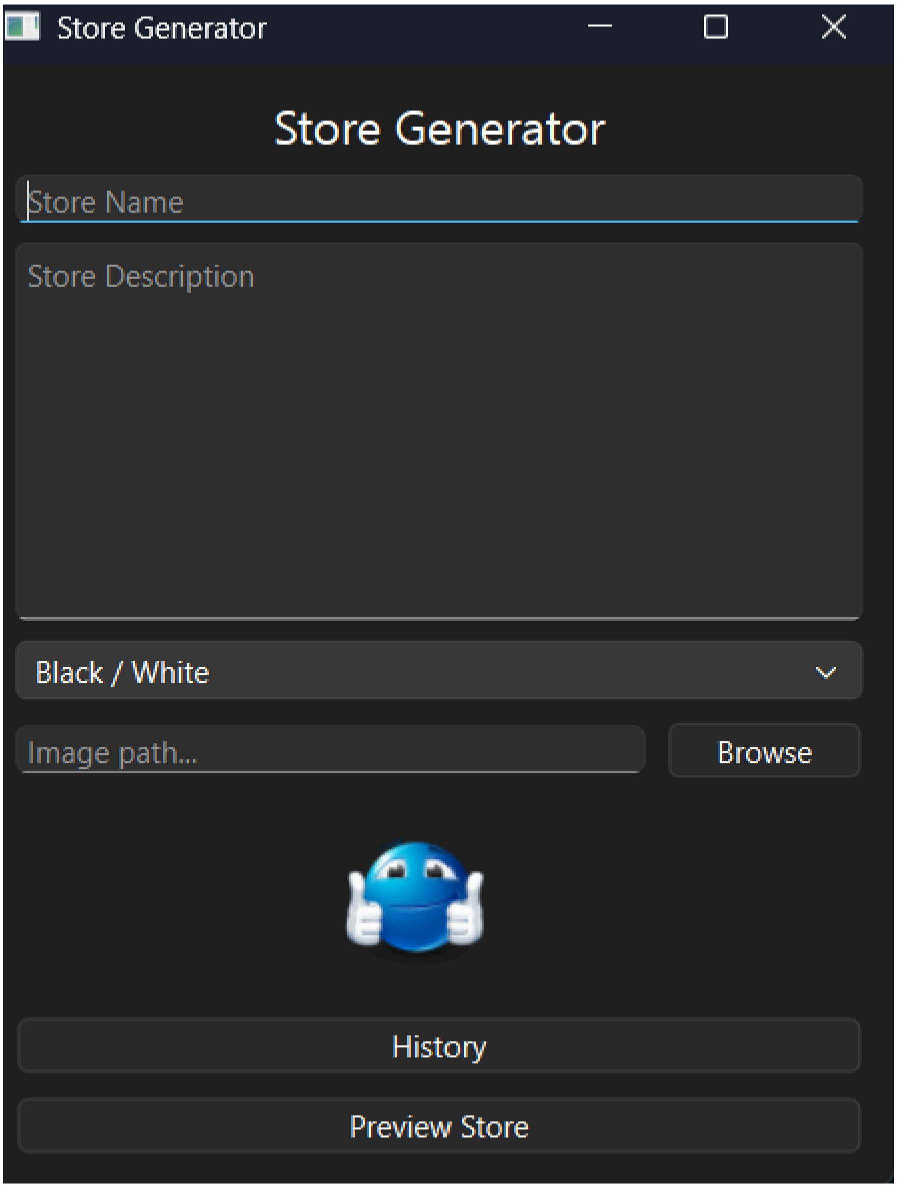
By

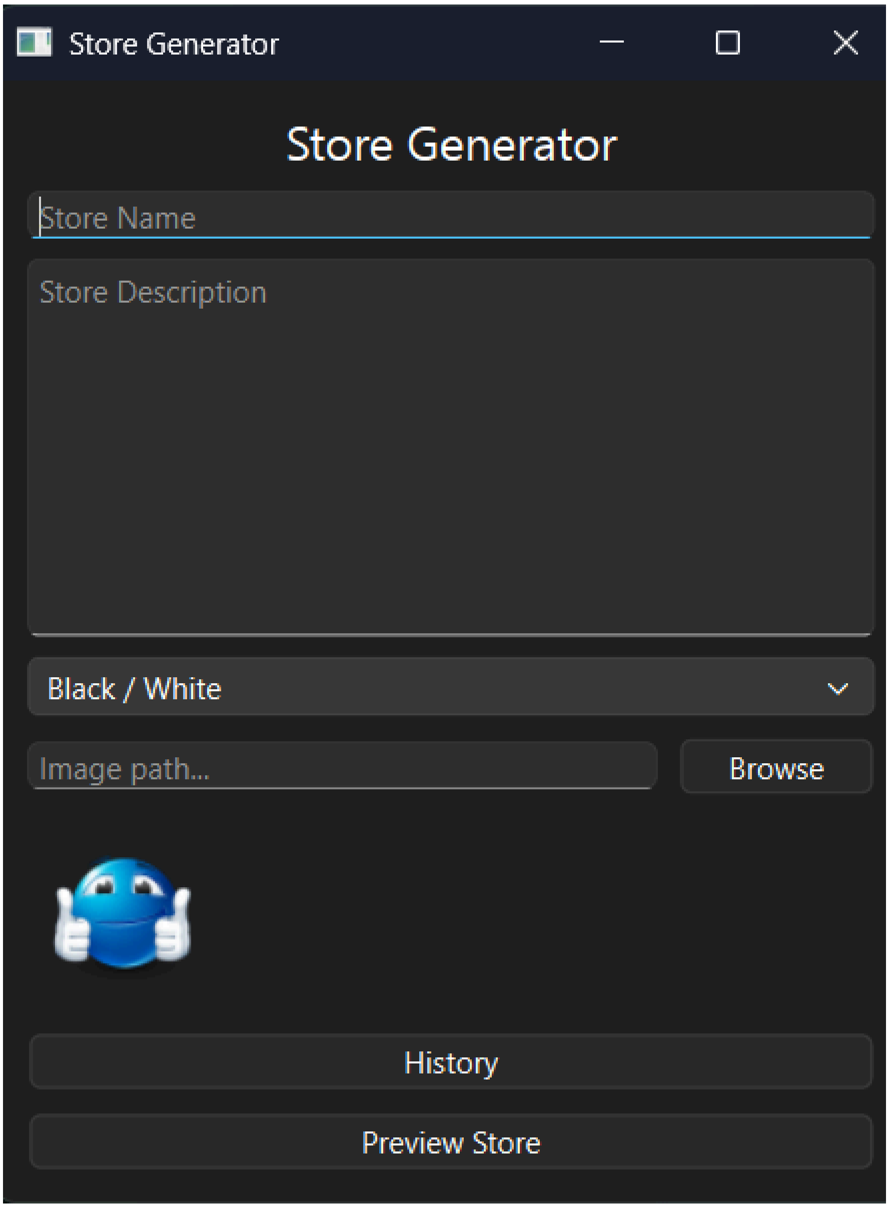
67011177 Napatrapee Chieowwit

Q. add animation and image to gui

123.png is

*note the program will use 👍instead if no image found!


Store Generator
Store Generator
Store Name
Store Description
Black / White
Image path...
Browse
History
Preview Store

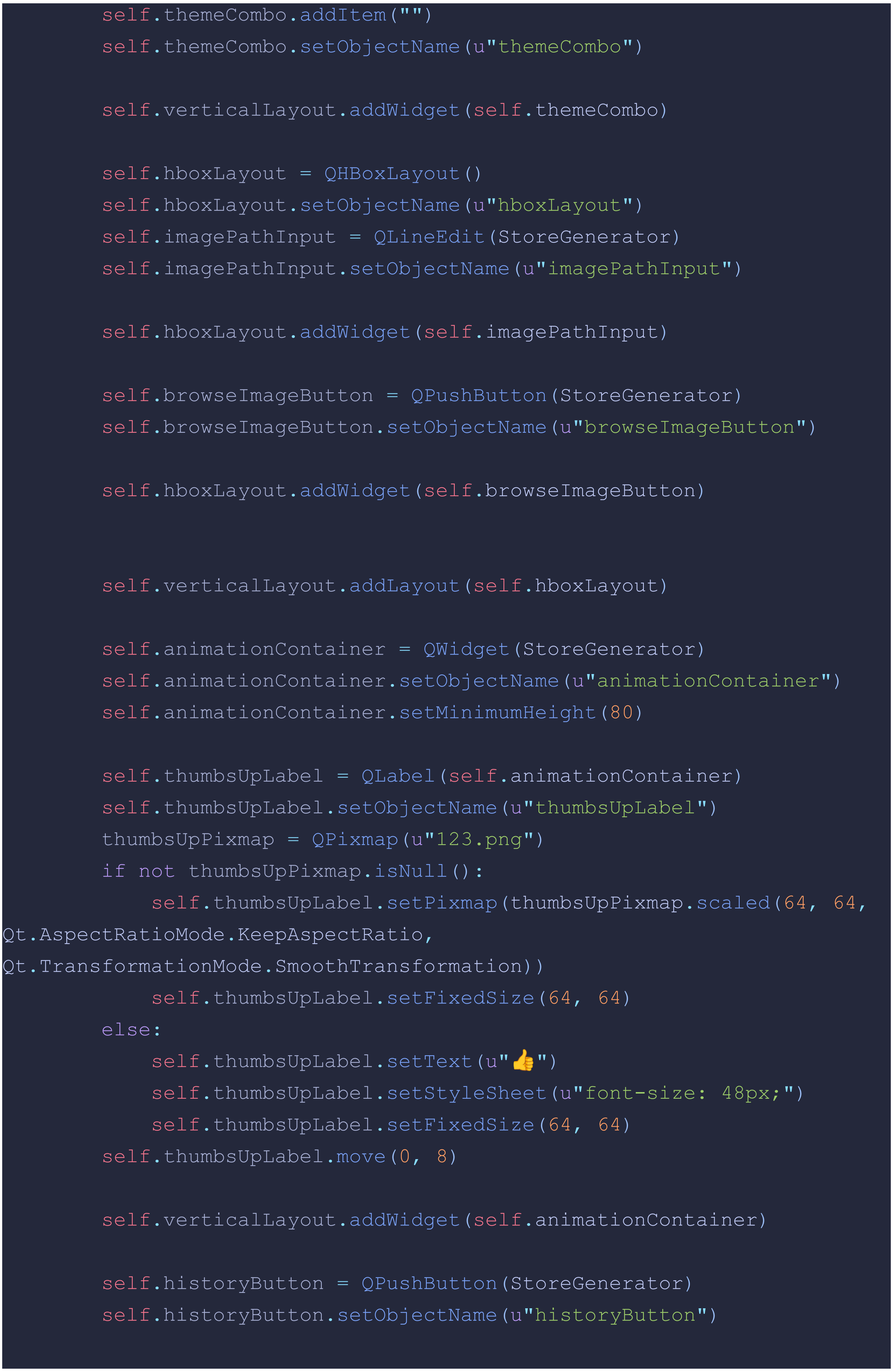


Note the thumbs up is moving

#code

```
# -*- coding: utf-8 -*-
# history page
################################################################################
## Form generated from reading UI file 'history.ui'
##
## Created by: Qt User Interface Compiler version 6.10.1
##
## WARNING! All changes made in this file will be lost when recompiling UI file!
```

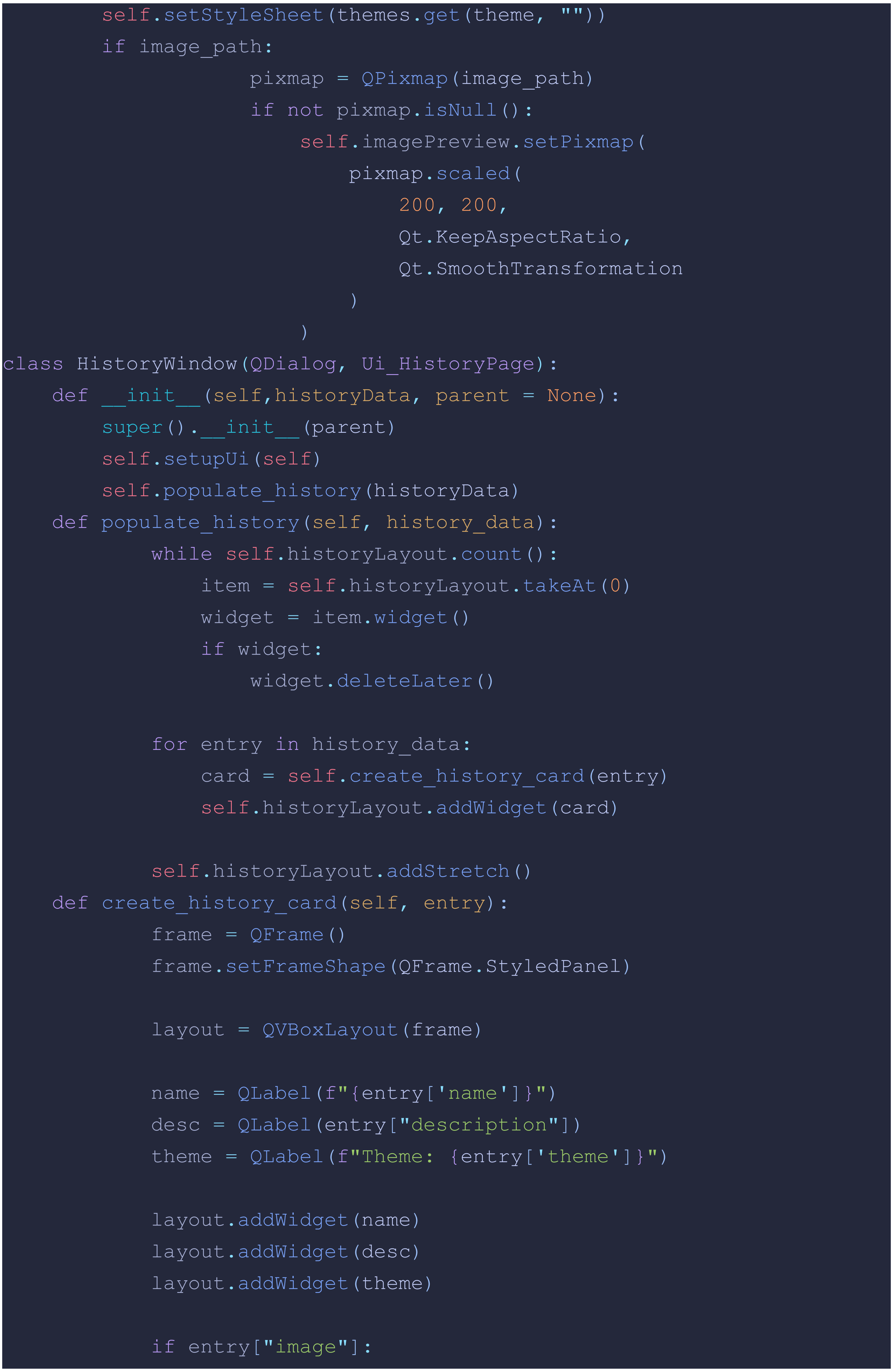
```
################################################################################
#########

from PySide6.QtCore import (QCoreApplication, QDate, QDateTime,
QLocale,
    QMetaObject, QObject, QPoint, QRect,
    QSize, QTime, QUrl, Qt)
from PySide6.QtGui import (QBrush, QColor, QConicalGradient, QCursor,
    QFont, QFontDatabase, QGradient, QIcon,
    QImage, QKeySequence, QLinearGradient, QPainter,
    QPalette, QPixmap, QRadialGradient, QTransform)
from PySide6.QtWidgets import (QApplication, QLabel, QScrollArea,
QSizePolicy,
    QSpacerItem, QVBoxLayout, QWidget)

class Ui_HistoryPage(object):
    def setupUi(self, HistoryPage):
        if not HistoryPage.objectName():
            HistoryPage.setObjectName(u"HistoryPage")
        HistoryPage.resize(381, 320)
        self.mainLayout = QVBoxLayout(HistoryPage)
        self.mainLayout.setSpacing(8)
        self.mainLayout.setObjectName(u"mainLayout")
        self.mainLayout.setContentsMargins(8, 8, 8, 8)
        self.titleLabel = QLabel(HistoryPage)
        self.titleLabel.setObjectName(u"titleLabel")

self.titleLabel.setAlignment(Qt.AlignmentFlag.AlignLeading|Qt.Alignment
Flag.AlignLeft|Qt.AlignmentFlag.AlignVCenter)

        self.mainLayout.addWidget(self.titleLabel)

        self.scrollArea = QScrollArea(HistoryPage)
        self.scrollArea.setObjectName(u"scrollArea")
        self.scrollArea.setWidgetResizable(True)
        self.scrollWidget = QWidget()
        self.scrollWidget.setObjectName(u"scrollWidget")
        self.scrollWidget.setGeometry(QRect(0, 0, 363, 270))
        self.historyLayout = QVBoxLayout(self.scrollWidget)
        self.historyLayout.setSpacing(6)
        self.historyLayout.setObjectName(u"historyLayout")
        self.historyLayout.setContentsMargins(4, 4, 4, 4)
        self.spacerItem = QSpacerItem(20, 40,
QSizePolicy.Policy.Minimum, QSizePolicy.Policy.Expanding)

        self.historyLayout.addItem(self.spacerItem)

        self.scrollArea.setWidget(self.scrollWidget)

        self.mainLayout.addWidget(self.scrollArea)


        self.retranslateUi(HistoryPage)

        QMetaObject.connectSlotsByName(HistoryPage)
    # setupUi

    def retranslateUi(self, HistoryPage):

HistoryPage.setWindowTitle(QCoreApplication.translate("HistoryPage",
u"History", None))

self.titleLabel.setStyleSheet(QCoreApplication.translate("HistoryPage",
u"font-size: 18px; font-weight: bold;", None))

self.titleLabel.setText(QCoreApplication.translate("HistoryPage",
u"History Container", None))
    # retranslateUi

#store generator page
# -*- coding: utf-8 -*-

########################################################################
#########
## Form generated from reading UI file 'store_generator.ui'
##
## Created by: Qt User Interface Compiler version 6.10.1
##
## WARNING! All changes made in this file will be lost when recompiling
UI file!
########################################################################
#########

from PySide6.QtCore import (QCoreApplication, QDate, QDateTime,
QLocale,
    QMetaObject, QObject, QPoint, QRect,
    QSize, QTime, QUrl, Qt, QPropertyAnimation, QEasingCurve)
from PySide6.QtGui import (QBrush, QColor, QConicalGradient, QCursor,
    QFont, QFontDatabase, QGradient, QIcon,
    QImage, QKeySequence, QLinearGradient, QPainter,
    QPalette, QPixmap, QRadialGradient, QTransform)
from PySide6.QtWidgets import (QApplication, QComboBox, QHBoxLayout,
QLabel,
    QLineEdit, QPushButton, QSizePolicy, QTextEdit,
    QVBoxLayout, QWidget)

class Ui_StoreGenerator(object):
    def setupUi(self, StoreGenerator):
        if not StoreGenerator.objectName():
            StoreGenerator.setObjectName(u"StoreGenerator")
        StoreGenerator.resize(363, 448)
        self.verticalLayout = QVBoxLayout(StoreGenerator)
        self.verticalLayout.setObjectName(u"verticalLayout")
        self.label = QLabel(StoreGenerator)
        self.label.setObjectName(u"label")
        font = QFont()
        font.setFamilies([u"Segoe UI Historic"])
        font.setPointSize(14)
        self.label.setFont(font)
        self.label.setAlignment(Qt.AlignmentFlag.AlignCenter)

        self.verticalLayout.addWidget(self.label)

        self.nameInput = QLineEdit(StoreGenerator)
        self.nameInput.setObjectName(u"nameInput")

        self.verticalLayout.addWidget(self.nameInput)

        self.descriptionInput = QTextEdit(StoreGenerator)
        self.descriptionInput.setObjectName(u"descriptionInput")

        self.verticalLayout.addWidget(self.descriptionInput)

        self.themeCombo = QComboBox(StoreGenerator)
        self.themeCombo.addItem("")
        self.themeCombo.addItem("")
        self.themeCombo.addItem("")
        self.themeCombo.addItem("")
```



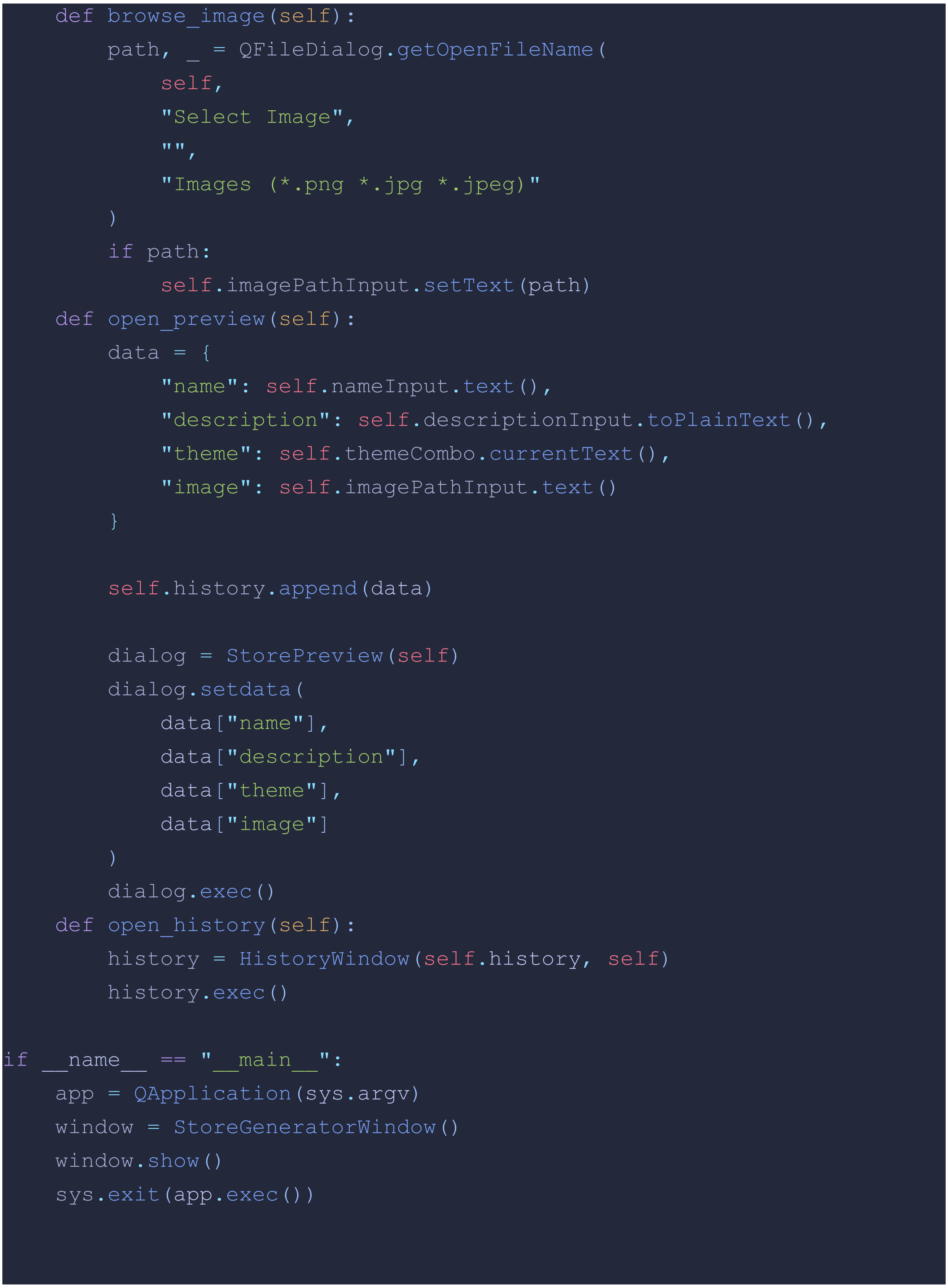
```
        self.themeCombo.addItem("")
        self.themeCombo.setObjectName(u"themeCombo")

        self.verticalLayout.addWidget(self.themeCombo)

        self.hboxLayout = QHBoxLayout()
        self.hboxLayout.setObjectName(u"hboxLayout")
        self.imagePathInput = QLineEdit(StoreGenerator)
        self.imagePathInput.setObjectName(u"imagePathInput")

        self.hboxLayout.addWidget(self.imagePathInput)

        self.browseImageButton = QPushButton(StoreGenerator)
        self.browseImageButton.setObjectName(u"browseImageButton")

        self.hboxLayout.addWidget(self.browseImageButton)


        self.verticalLayout.addLayout(self.hboxLayout)

        self.animationContainer = QWidget(StoreGenerator)
        self.animationContainer.setObjectName(u"animationContainer")
        self.animationContainer.setMinimumHeight(80)

        self.thumbsUpLabel = QLabel(self.animationContainer)
        self.thumbsUpLabel.setObjectName(u"thumbsUpLabel")
        thumbsUpPixmap = QPixmap(u"123.png")
        if not thumbsUpPixmap.isNull():
            self.thumbsUpLabel.setPixmap(thumbsUpPixmap.scaled(64, 64,
Qt.AspectRatioMode.KeepAspectRatio,
Qt.TransformationMode.SmoothTransformation))
            self.thumbsUpLabel.setFixedSize(64, 64)
        else:
            self.thumbsUpLabel.setText(u"👍")
            self.thumbsUpLabel.setStyleSheet(u"font-size: 48px;")
            self.thumbsUpLabel.setFixedSize(64, 64)
        self.thumbsUpLabel.move(0, 8)

        self.verticalLayout.addWidget(self.animationContainer)

        self.historyButton = QPushButton(StoreGenerator)
        self.historyButton.setObjectName(u"historyButton")
        self.verticalLayout.addWidget(self.historyButton)

        self.previewButton = QPushButton(StoreGenerator)
        self.previewButton.setObjectName(u"previewButton")

        self.verticalLayout.addWidget(self.previewButton)


        self.retranslateUi(StoreGenerator)

        QMetaObject.connectSlotsByName(StoreGenerator)
    # setupUi

    def retranslateUi(self, StoreGenerator):
        StoreGenerator.setWindowTitle(QCoreApplication.translate("StoreGenerator", u"Store Generator", None))
        self.label.setText(QCoreApplication.translate("StoreGenerator", u"Store Generator", None))
        self.nameInput.setPlaceholderText(QCoreApplication.translate("StoreGenerator", u"Store Name", None))
        self.descriptionInput.setPlaceholderText(QCoreApplication.translate("StoreGenerator", u"Store Description", None))
        self.themeCombo.setItemText(0, QCoreApplication.translate("StoreGenerator", u"Black / White", None))
        self.themeCombo.setItemText(1, QCoreApplication.translate("StoreGenerator", u"Purple / Black", None))
        self.themeCombo.setItemText(2, QCoreApplication.translate("StoreGenerator", u"Blue / Light", None))
        self.themeCombo.setItemText(3, QCoreApplication.translate("StoreGenerator", u"Green / Dark", None))
        self.themeCombo.setItemText(4, QCoreApplication.translate("StoreGenerator", u"Orange / Gray", None))

        self.imagePathInput.setPlaceholderText(QCoreApplication.translate("StoreGenerator", u"Image path...", None))
        self.browseImageButton.setText(QCoreApplication.translate("StoreGenerator", u"Browse", None))
```

```
self.historyButton.setText(QCoreApplication.translate("StoreGenerator",
u"History", None))

self.previewButton.setText(QCoreApplication.translate("StoreGenerator",
u"Preview Store", None))
    # retranslateUi

#store preview page
# -*- coding: utf-8 -*-

########################################################################
#########
## Form generated from reading UI file 'store_preview.ui'
##
## Created by: Qt User Interface Compiler version 6.10.1
##
## WARNING! All changes made in this file will be lost when recompiling
UI file!
########################################################################
#########

from PySide6.QtCore import (QCoreApplication, QDate, QDateTime,
QLocale,
    QMetaObject, QObject, QPoint, QRect,
    QSize, QTime, QUrl, Qt)
from PySide6.QtGui import (QBrush, QColor, QConicalGradient, QCursor,
    QFont, QFontDatabase, QGradient, QIcon,
    QImage, QKeySequence, QLinearGradient, QPainter,
    QPalette, QPixmap, QRadialGradient, QTransform)
from PySide6.QtWidgets import (QApplication, QDialog, QFrame, QLabel,
    QPushButton, QSizePolicy, QVBoxLayout, QWidget)

class Ui_StorePreviewDialog(object):
    def setupUi(self, StorePreviewDialog):
        if not StorePreviewDialog.objectName():
            StorePreviewDialog.setObjectName(u"StorePreviewDialog")
        StorePreviewDialog.resize(317, 413)
        self.verticalLayout = QVBoxLayout(StorePreviewDialog)
        self.verticalLayout.setObjectName(u"verticalLayout")
        self.previewName = QLabel(StorePreviewDialog)
        self.previewName.setObjectName(u"previewName")
        font = QFont()
        font.setPointSize(12)
        self.previewName.setFont(font)
        self.previewName.setAlignment(Qt.AlignmentFlag.AlignCenter)

        self.verticalLayout.addWidget(self.previewName)

        self.imagePreview = QLabel(StorePreviewDialog)
        self.imagePreview.setObjectName(u"imagePreview")
        self.imagePreview.setMinimumSize(QSize(200, 200))
        self.imagePreview.setFrameShape(QFrame.Shape.Box)
        self.imagePreview.setAlignment(Qt.AlignmentFlag.AlignCenter)

        self.verticalLayout.addWidget(self.imagePreview)

        self.previewDescription = QLabel(StorePreviewDialog)
        self.previewDescription.setObjectName(u"previewDescription")
        self.previewDescription.setWordWrap(True)

        self.verticalLayout.addWidget(self.previewDescription)

        self.closeButton = QPushButton(StorePreviewDialog)
        self.closeButton.setObjectName(u"closeButton")

        self.verticalLayout.addWidget(self.closeButton)


        self.retranslateUi(StorePreviewDialog)

        QMetaObject.connectSlotsByName(StorePreviewDialog)
    # setupUi

    def retranslateUi(self, StorePreviewDialog):

StorePreviewDialog.setWindowTitle(QCoreApplication.translate("StorePrev
iewDialog", u"Store Preview", None))

self.previewName.setText(QCoreApplication.translate("StorePreviewDialog
", u"Store Name", None))

self.imagePreview.setText(QCoreApplication.translate("StorePreviewDialo
g", u"Image Preview", None))
```

```
self.previewDescription.setText(QCoreApplication.translate("StorePreviewDialog", u"Description", None))

self.closeButton.setText(QCoreApplication.translate("StorePreviewDialog", u"Close", None))
    # retranslateUi

#main py program

import sys
from PySide6.QtWidgets import (
    QApplication, QWidget, QDialog, QFileDialog,
    QLabel, QFrame, QVBoxLayout
)
from PySide6.QtGui import QPixmap
from PySide6.QtCore import Qt
themes = {
    "Black / White": """
        QWidget { background: white; color: black; }
    """,
    "Purple / Black": """
        QWidget { background: #1e1b2e; color: #c084fc; }
    """,
    "Blue / Light": """
        QWidget { background: #e0f2fe; color: #075985; }
    """,
    "Green / Dark": """
        QWidget { background: #052e16; color: #bbf7d0; }
    """,
    "Orange / Gray": """
        QWidget { background: #374151; color: #fb923c; }
    """
}

class StorePreview(QDialog, Ui_StorePreviewDialog):
    def __init__(self, parent = None):
        super().__init__(parent)
        self.setupUi(self)
        self.closeButton.clicked.connect(self.close)
    def setdata(self, name, description, theme, image_path):
        self.previewName.setText(name)
        self.previewDescription.setText(description)
```



```
        self.setStyleSheet(themes.get(theme, ""))
        if image_path:
                    pixmap = QPixmap(image_path)
                    if not pixmap.isNull():
                        self.imagePreview.setPixmap(
                            pixmap.scaled(
                                200, 200,
                                Qt.KeepAspectRatio,
                                Qt.SmoothTransformation
                            )
                        )
class HistoryWindow(QDialog, Ui_HistoryPage):
    def __init__(self,historyData, parent = None):
        super().__init__(parent)
        self.setupUi(self)
        self.populate_history(historyData)
    def populate_history(self, history_data):
            while self.historyLayout.count():
                item = self.historyLayout.takeAt(0)
                widget = item.widget()
                if widget:
                    widget.deleteLater()

            for entry in history_data:
                card = self.create_history_card(entry)
                self.historyLayout.addWidget(card)

            self.historyLayout.addStretch()
    def create_history_card(self, entry):
            frame = QFrame()
            frame.setFrameShape(QFrame.StyledPanel)

            layout = QVBoxLayout(frame)

            name = QLabel(f"{entry['name']}")
            desc = QLabel(entry["description"])
            theme = QLabel(f"Theme: {entry['theme']}")

            layout.addWidget(name)
            layout.addWidget(desc)
            layout.addWidget(theme)

            if entry["image"]:
                img = QLabel()
                pixmap = QPixmap(entry["image"])
                if not pixmap.isNull():
                    img.setPixmap(
                        pixmap.scaled(150, 150, Qt.KeepAspectRatio)
                    )
                    layout.addWidget(img)

            return frame


class StoreGeneratorWindow(QWidget, Ui_StoreGenerator):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.history = []
        self.previewButton.clicked.connect(self.open_preview)
        self.browseImageButton.clicked.connect(self.browse_image)
        self.historyButton.clicked.connect(self.open_history)

        # Setup animation for thumbs up
        self.setupThumbsUpAnimation()

    def setupThumbsUpAnimation(self):
        self.thumbsUpAnimation = QPropertyAnimation(self.thumbsUpLabel, b"pos")
        self.thumbsUpAnimation.setDuration(3000)  # 3 seconds
        self.thumbsUpAnimation.setStartValue(QPoint(-64, 8))  # Start from left (off screen)
        self.thumbsUpAnimation.setEndValue(QPoint(self.animationContainer.width() + 64, 8))  # End at right (off screen)
        self.thumbsUpAnimation.setEasingCurve(QEasingCurve.Linear)
        self.thumbsUpAnimation.setLoopCount(-1)  # Loop forever
        self.thumbsUpAnimation.start()

    def resizeEvent(self, event):
        super().resizeEvent(event)
        # Update animation end value when window resizes
        if hasattr(self, 'thumbsUpAnimation'):
            self.thumbsUpAnimation.setEndValue(QPoint(self.animationContainer.width() + 64, 8))
```



```python
    def browse_image(self):
        path, _ = QFileDialog.getOpenFileName(
            self,
            "Select Image",
            "",
            "Images (*.png *.jpg *.jpeg)"
        )
        if path:
            self.imagePathInput.setText(path)
    def open_preview(self):
        data = {
            "name": self.nameInput.text(),
            "description": self.descriptionInput.toPlainText(),
            "theme": self.themeCombo.currentText(),
            "image": self.imagePathInput.text()
        }

        self.history.append(data)

        dialog = StorePreview(self)
        dialog.setdata(
            data["name"],
            data["description"],
            data["theme"],
            data["image"]
        )
        dialog.exec()
    def open_history(self):
        history = HistoryWindow(self.history, self)
        history.exec()

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = StoreGeneratorWindow()
    window.show()
    sys.exit(app.exec())
```