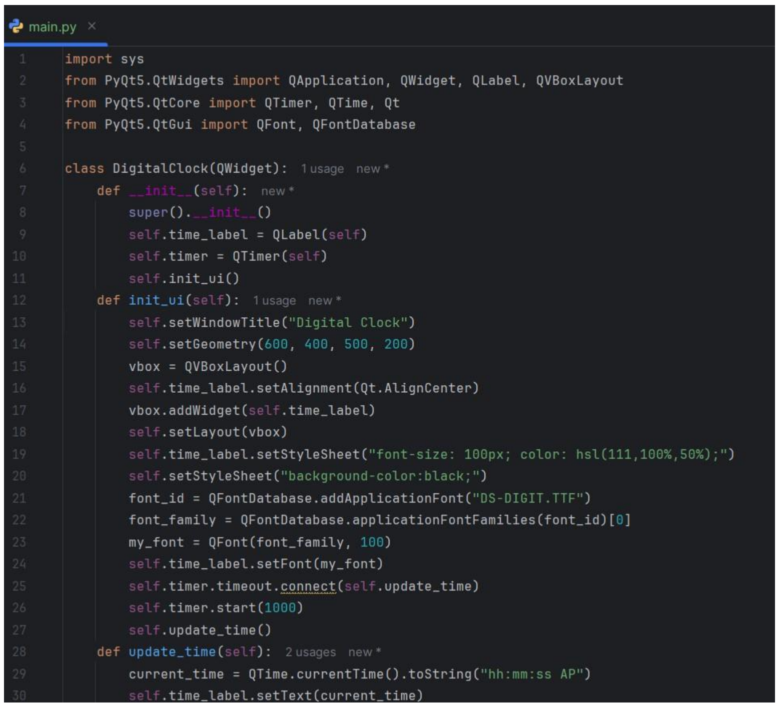
Name: Sadaf Riaz

Roll no: 2023-BSE-077

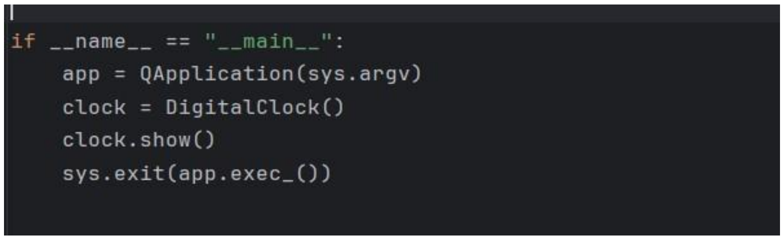
Class: BSE5-B

Lab 08

Digital Clock

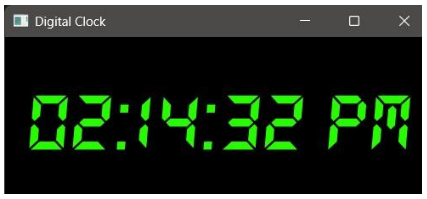


```python
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QVBoxLayout
from PyQt5.QtCore import QTimer, QTime, Qt
from PyQt5.QtGui import QFont, QFontDatabase

class DigitalClock(QWidget):
    def __init__(self):
        super().__init__()
        self.time_label = QLabel(self)
        self.timer = QTimer(self)
        self.init_ui()
    def init_ui(self):
        self.setWindowTitle("Digital Clock")
        self.setGeometry(600, 400, 500, 200)
        vbox = QVBoxLayout()
        self.time_label.setAlignment(Qt.AlignCenter)
        vbox.addWidget(self.time_label)
        self.setLayout(vbox)
        self.time_label.setStyleSheet("font-size: 100px; color: hsl(111,100%,50%);")
        self.setStyleSheet("background-color:black;")
        font_id = QFontDatabase.addApplicationFont("DS-DIGIT.TTF")
        font_family = QFontDatabase.applicationFontFamilies(font_id)[0]
        my_font = QFont(font_family, 100)
        self.time_label.setFont(my_font)
        self.timer.timeout.connect(self.update_time)
        self.timer.start(1000)
        self.update_time()
    def update_time(self):
        current_time = QTime.currentTime().toString("hh:mm:ss AP")
        self.time_label.setText(current_time)
```



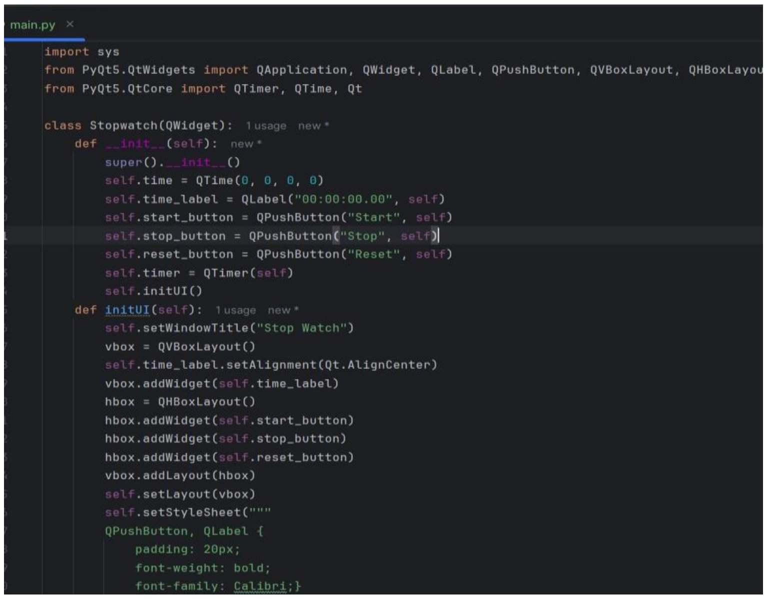
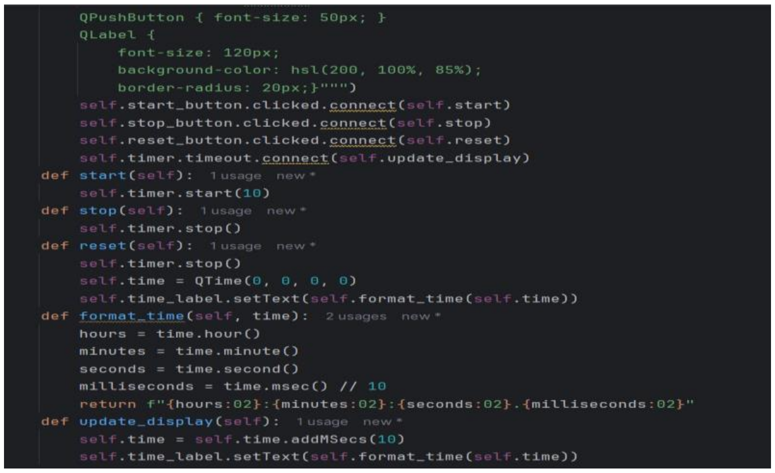
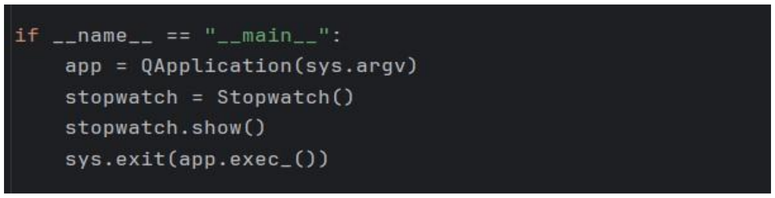
```python
if __name__ == "__main__":
    app = QApplication(sys.argv)
    clock = DigitalClock()
    clock.show()
    sys.exit(app.exec_())
```

Output:



```
02:14:32 PM
```

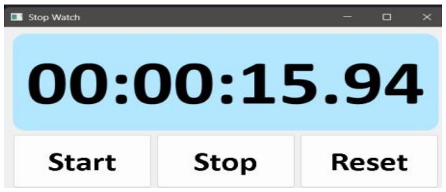
Stop Watch

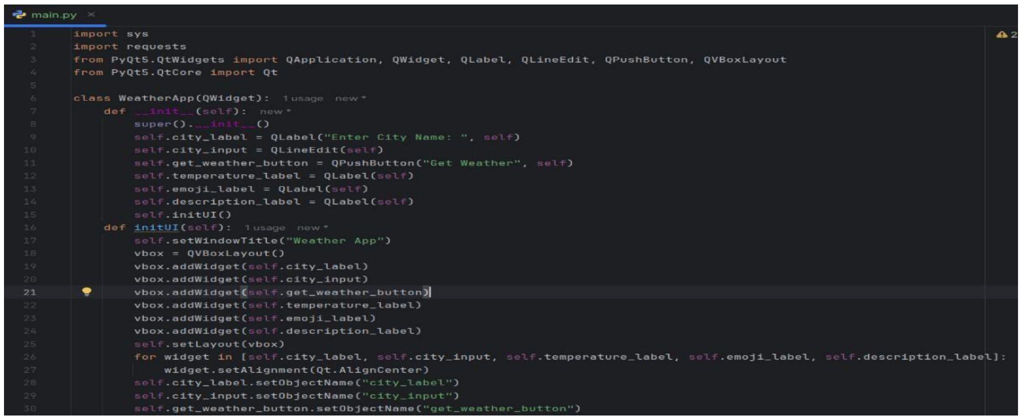
```python
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QPushButton, QVBoxLayout, QHBoxLayou
from PyQt5.QtCore import QTimer, QTime, Qt

class Stopwatch(QWidget):
    def __init__(self):
        super().__init__()
        self.time = QTime(0, 0, 0, 0)
        self.time_label = QLabel("00:00:00.00", self)
        self.start_button = QPushButton("Start", self)
        self.stop_button = QPushButton("Stop", self)
        self.reset_button = QPushButton("Reset", self)
        self.timer = QTimer(self)
        self.initUI()
    def initUI(self):
        self.setWindowTitle("Stop Watch")
        vbox = QVBoxLayout()
        self.time_label.setAlignment(Qt.AlignCenter)
        vbox.addWidget(self.time_label)
        hbox = QHBoxLayout()
        hbox.addWidget(self.start_button)
        hbox.addWidget(self.stop_button)
        hbox.addWidget(self.reset_button)
        vbox.addLayout(hbox)
        self.setLayout(vbox)
        self.setStyleSheet("""
        QPushButton, QLabel {
            padding: 20px;
            font-weight: bold;
            font-family: Calibri;}
        QPushButton { font-size: 50px; }
        QLabel {
            font-size: 120px;
            background-color: hsl(200, 100%, 85%);
            border-radius: 20px;}""")
        self.start_button.clicked.connect(self.start)
        self.stop_button.clicked.connect(self.stop)
        self.reset_button.clicked.connect(self.reset)
        self.timer.timeout.connect(self.update_display)
    def start(self):
        self.timer.start(10)
    def stop(self):
        self.timer.stop()
    def reset(self):
        self.timer.stop()
        self.time = QTime(0, 0, 0, 0)
        self.time_label.setText(self.format_time(self.time))
    def format_time(self, time):
        hours = time.hour()
        minutes = time.minute()
        seconds = time.second()
        milliseconds = time.msec() // 10
        return f"{hours:02}:{minutes:02}:{seconds:02}.{milliseconds:02}"
    def update_display(self):
        self.time = self.time.addMSecs(10)
        self.time_label.setText(self.format_time(self.time))
```

```python
if __name__ == "__main__":
    app = QApplication(sys.argv)
    stopwatch = Stopwatch()
    stopwatch.show()
    sys.exit(app.exec_())
```

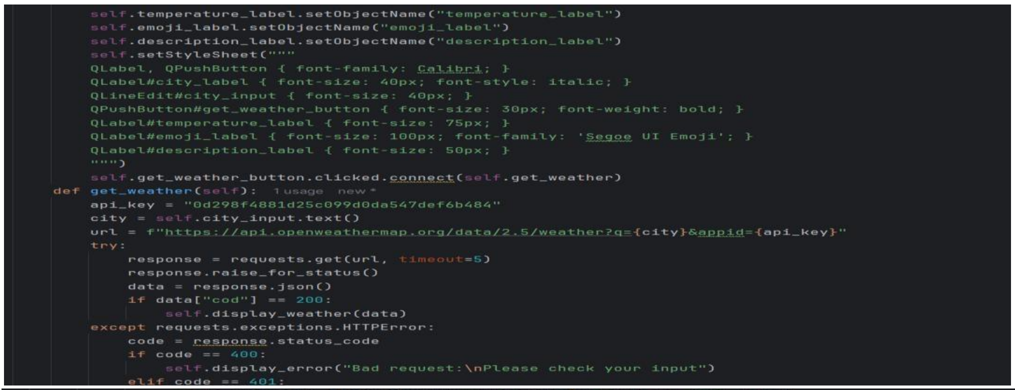
Output:



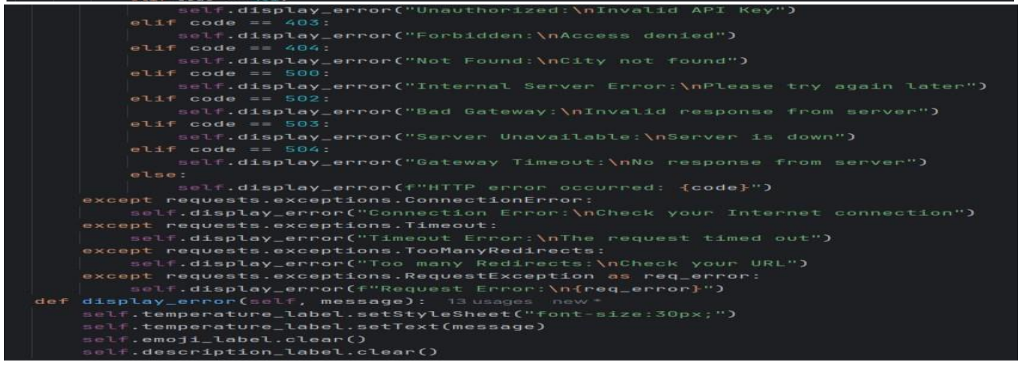
Weather Aap



```python
import sys
import requests
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QLineEdit, QPushButton, QVBoxLayout
from PyQt5.QtCore import Qt

class WeatherApp(QWidget):
    def __init__(self):
        super().__init__()
        self.city_label = QLabel("Enter City Name: ", self)
        self.city_input = QLineEdit(self)
        self.get_weather_button = QPushButton("Get Weather", self)
        self.temperature_label = QLabel(self)
        self.emoji_label = QLabel(self)
        self.description_label = QLabel(self)
        self.initUI()
    def initUI(self):
        self.setWindowTitle("Weather App")
        vbox = QVBoxLayout()
        vbox.addWidget(self.city_label)
        vbox.addWidget(self.city_input)
        vbox.addWidget(self.get_weather_button)
        vbox.addWidget(self.temperature_label)
        vbox.addWidget(self.emoji_label)
        vbox.addWidget(self.description_label)
        self.setLayout(vbox)
        for widget in [self.city_label, self.city_input, self.temperature_label, self.emoji_label, self.description_label]:
            widget.setAlignment(Qt.AlignCenter)
        self.city_label.setObjectName("city_label")
        self.city_input.setObjectName("city_input")
        self.get_weather_button.setObjectName("get_weather_button")
```

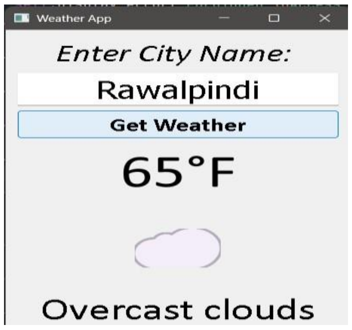


```python
        self.temperature_label.setObjectName("temperature_label")
        self.emoji_label.setObjectName("emoji_label")
        self.description_label.setObjectName("description_label")
        self.setStyleSheet("""
        QLabel, QPushButton { font-family: Calibri; }
        QLabel#city_label { font-size: 40px; font-style: italic; }
        QLineEdit#city_input { font-size: 40px; }
        QPushButton#get_weather_button { font-size: 30px; font-weight: bold; }
        QLabel#temperature_label { font-size: 75px; }
        QLabel#emoji_label { font-size: 100px; font-family: 'Segoe UI Emoji'; }
        QLabel#description_label { font-size: 50px; }
        """)
        self.get_weather_button.clicked.connect(self.get_weather)
    def get_weather(self):
        api_key = "0d298f4881d25c099d0da547def6b484"
        city = self.city_input.text()
        url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"
        try:
            response = requests.get(url, timeout=5)
            response.raise_for_status()
            data = response.json()
            if data["cod"] == 200:
                self.display_weather(data)
        except requests.exceptions.HTTPError:
            code = response.status_code
            if code == 400:
                self.display_error("Bad request:\nPlease check your input")
            elif code == 401:
```



```python
                self.display_error("Unauthorized:\nInvalid API Key")
            elif code == 403:
                self.display_error("Forbidden:\nAccess denied")
            elif code == 404:
                self.display_error("Not Found:\nCity not found")
            elif code == 500:
                self.display_error("Internal Server Error:\nPlease try again later")
            elif code == 502:
                self.display_error("Bad Gateway:\nInvalid response from server")
            elif code == 503:
                self.display_error("Server Unavailable:\nServer is down")
            elif code == 504:
                self.display_error("Gateway Timeout:\nNo response from server")
            else:
                self.display_error(f"HTTP error occurred: {code}")
        except requests.exceptions.ConnectionError:
            self.display_error("Connection Error:\nCheck your Internet connection")
        except requests.exceptions.Timeout:
            self.display_error("Timeout Error:\nThe request timed out")
        except requests.exceptions.TooManyRedirects:
            self.display_error("Too many Redirects:\nCheck your URL")
        except requests.exceptions.RequestException as req_error:
            self.display_error(f"Request Error:\n{req_error}")
    def display_error(self, message):
        self.temperature_label.setStyleSheet("font-size:30px;")
        self.temperature_label.setText(message)
        self.emoji_label.clear()
        self.description_label.clear()
```

Output:



```
Weather App
Enter City Name:
Rawalpindi
Get Weather
65°F
Overcast clouds
```