Phase 4: Development part 2

Title: Environmental monitoring

Create a html file (index.html):

```
<!DOCTYPE html>
<html>
<head>
  <title>IoT Temperature and Humidity Monitor</title>
</head>
<body>
  <h1>Temperature and Humidity Monitoring</h1>

  <div id="temperature">Temperature: Loading...</div>
  <div id="humidity">Humidity: Loading...</div>

  <script>
    // Simulated sensor data for demonstration purposes
    var sensorData = {
      temperature: 25.5, // Replace with your actual temperature data
      humidity: 60.2,   // Replace with your actual humidity data
    };

    // Function to update sensor data on the web page
    function updateSensorData() {
      document.getElementById('temperature').textContent = 'Temperature: ' + sensorData.temperature + '°C';
```

```
            document.getElementById('humidity').textContent = 'Humidity: ' + sensorData.humidity + '%';
        }

        // Periodically update the sensor data (replace this with your actual data retrieval logic)
        setInterval(updateSensorData, 5000); // Update every 5 seconds
    </script>
</body>
</html>
```

Create a CSS file (styles.css):

```
/* Container for the sensor display */
.sensor-container {
    background-color: #f0f0f0;
    border: 2px solid #ccc;
    padding: 20px;
    width: 200px;
    text-align: center;
}

/* Styling for temperature reading */
.temperature {
    font-size: 24px;
    color: #ff5733;
}
```

```
/* Styling for humidity reading */
.humidity {
   font-size: 18px;
   color: #3385ff;
}
```

Create a JavaScript file (script.jss):

```
import com.pi4j.io.gpio.*;
import com.pi4j.temperature.TemperatureScale;
import com.pi4j.temperature.TemperatureSensor;
import com.pi4j.wiringpi.GpioUtil;

public class TemperatureHumiditySensor {

   public static void main(String[] args) throws Exception {
      // Initialize Pi4J GPIO
      GpioUtil.enableNonPrivilegedAccess();

      // Create a GpioController
      GpioController gpio = GpioFactory.getInstance();

      // DHT22 sensor is connected to GPIO pin 4 (BCM numbering)
      final GpioPinDigitalInput dhtPin = gpio.provisionDigitalInputPin(RaspiPin.GPIO_04,
PinPullResistance.PULL_UP);

      // Create a temperature sensor instance
```

```
    TemperatureSensor temperatureSensor = new TemperatureSensor.Builder()
        .temperatureScale(TemperatureScale.CELSIUS)
        .gpioProvider(new
RaspiGpioProvider(RaspiPinNumberingScheme.BROADCOM_PIN_NUMBERING))
        .build();

    while (true) {
      // Read temperature and humidity from the DHT22 sensor
      double temperature = temperatureSensor.getTemperature();
      double humidity = temperatureSensor.getHumidity();

      // Print the readings
      System.out.println("Temperature: " + temperature + "°C");
      System.out.println("Humidity: " + humidity + "%");

      // Sleep for a while before reading again
      Thread.sleep(2000);
    }
  }
}
```