

Дідріхе А. Г.,
здобувач вищої освіти спеціальності 122 Комп'ютерні науки
Науковий керівник: **Тищенко С. І.,**
канд.пед.наук, доцент кафедри економічної кібернетики, комп'ютерних
наук та інформаційних технологій
Миколаївський національний аграрний університет, м. Миколаїв

PYTHON-СЕРВІСИ ДЛЯ ВІДСТЕЖЕННЯ ТА КОНТРОЛЮ ТЕМПЕРАТУРНИХ РЕЖИМІВ ПРИ ТРАНСПОРТУВАННІ ШВИДКОПСУВНИХ ПРОДУКТІВ

У сучасному світі сільськогосподарська продукція часто транспортується на великі відстані, що вимагає забезпечення належних температурних режимів для збереження якості та свіжості швидкопсувних продуктів. Порушення температурного режиму під час транспортування може призвести до псування продуктів і значних фінансових втрат. Для вирішення цієї проблеми необхідно розробити ефективний спосіб відстеження та контролю температури під час транспортування.

Python, поширена мова програмування, пропонує різноманітні інструменти та бібліотеки для створення веб-сервісів та роботи з даними.

Переваги використання Python-сервісів для відстеження та контролю температурних режимів при транспортуванні швидкопсувних продуктів:

1. Python – це кросплатформна мова програмування, що дозволяє розробляти сервіси, які можуть працювати на різних операційних системах, таких як Windows, Linux та macOS, без необхідності внесення значних змін у код.

2. Python має величезну кількість бібліотек та фреймворків для розробки веб-сервісів (наприклад, Flask, Django), роботи з базами даних (SQLite, PostgreSQL, MySQL), обробки даних (NumPy, Pandas) та багато іншого, що полегшує розробку та інтеграцію різних компонентів системи.

3. Синтаксис Python є досить простим і зрозумілим, що полегшує написання, читання та підтримку коду, особливо для великих проектів.

4. Завдяки простоті та зрозумілості коду, а також наявності великої кількості бібліотек, розробка Python-сервісів може бути швидшою в порівнянні з іншими мовами програмування.

5. Python-сервіси можна легко масштабувати та інтегрувати з іншими системами, такими як системи моніторингу, сповіщення та аналітики.

6. Python має велику та активну спільноту розробників, що сприяє постійному розвитку мови та бібліотек, а також забезпечує підтримку та обмін досвідом.

Розглянемо розробку простих Python-сервісів для відстеження та контролю температурних режимів при транспортуванні швидкопсувних продуктів.

1. Веб-сервіс на базі Flask

```
from flask import Flask, render_template, request
import sqlite3
app = Flask(__name__)
```

```

# Обробка запитів для отримання даних про температуру
@app.route('/temperature', methods=['POST'])
def receive_temperature():
    data = request.get_json()
    temperature = data['temperature']
    timestamp = data['timestamp']
    # Зберігаємо дані про температуру в базі даних
    conn = sqlite3.connect('temperatures.db')
    c = conn.cursor()
    c.execute("INSERT INTO temperatures (temperature, timestamp)
VALUES (?, ?)", (temperature, timestamp))
    conn.commit()
    conn.close()
    return 'Temperature data received'
# Відображення поточних та історичних даних про температуру
@app.route('/')
def index():
    conn = sqlite3.connect('temperatures.db')
    c = conn.cursor()
    c.execute("SELECT * FROM temperatures ORDER BY timestamp
DESC LIMIT 10")
    temperatures = c.fetchall()
    conn.close()
    return render_template('index.html', temperatures=temperatures)
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)

```

2. Аналіз та контроль температурних режимів

```

import sqlite3
from datetime import datetime, timedelta
# Підключення до бази даних
conn = sqlite3.connect('temperatures.db')
c = conn.cursor()
# Визначення допустимого діапазону температур
min_temp = 2
max_temp = 8
# Перевірка останніх даних про температуру
c.execute("SELECT temperature, timestamp FROM temperatures ORDER
BY timestamp DESC LIMIT 1")
last_temp, last_time = c.fetchone()
# Перевірка на відхилення від допустимого діапазону
if last_temp < min_temp or last_temp > max_temp:
    # Генерація сповіщення
    time_diff = datetime.now() - datetime.strptime(last_time, '%Y-%m-%d
%H:%M:%S')
    if time_diff < timedelta(hours=1):

```

```
print(f"Warning: Temperature {last_temp}°C is out of range
({min_temp}-{max_temp}°C)")
conn.close()
```

Розробка Python-сервісу для відстеження та контролю температурних режимів при транспортуванні швидкопсувних продуктів є важливим завданням для забезпечення якості та безпеки продуктів. Використовуючи потужні інструменти Python, такі як Flask для створення веб-сервісу та SQLite для зберігання даних, можна розробити ефективну систему для збору, аналізу та моніторингу даних про температуру.

Ключовими компонентами такого сервісу є:

1. Веб-сервіс на базі Flask для отримання даних про температуру від датчиків та відображення поточних і історичних даних.
2. База даних SQLite для зберігання даних про температуру та часові мітки.
3. Модуль аналізу та контролю температурних режимів, який виявляє відхилення від встановлених параметрів та генерує сповіщення у разі критичних ситуацій.

Наведені приклади коду демонструють основні функції сервісу, такі як отримання даних про температуру від датчиків, зберігання даних у базі даних, візуалізація поточних та історичних даних, а також аналіз температурних режимів і генерація сповіщень у разі виявлення відхилень.

Такий Python-сервіс забезпечує ефективний моніторинг і контроль температурних режимів під час транспортування швидкопсувних продуктів, що допомагає запобігти їх псуванню та мінімізувати фінансові втрати. Крім того, система дозволяє генерувати звіти про температурні режими, що сприяє прозорості та підзвітності в ланцюгу постачання.

Загалом, розробка Python-сервісу для відстеження та контролю температурних режимів є актуальним та перспективним рішенням для сільськогосподарської галузі, що дозволяє підвищити якість продукції, забезпечити безпеку споживачів та зменшити втрати під час транспортування.

Список використаних джерел

1. Цибко Г.Ю., Горошко Ю.В., Костюченко А.О. Програмування у Python. Практичний курс: навчальний посібник. Ч.: ФОП Баликіна С.М., 2022. 180 с.
2. Controlling Devices by IoT. International Journal of Recent Technology and Engineering. 2019. Vol. 8, no. 4. P. 8173-8176.
URL: <https://doi.org/10.35940/ijrte.d8777.118419>.