|  |  |
| --- | --- |
|  | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»  (МГТУ им. Н.Э. Баумана) |
| ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»  КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)  Лабораторные работы  «Парадигмы и конструкции языков программирования»  38.03.05 «Бизнес-информатика» (бакалавриат)  Студент ИБМ3-34Б Сергеева М. А.  (Подпись, дата)    2024 г. | |

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

**Тема:** Создание датасета на основе данных веб-сайта

**Цель работы:** Разработка программы для сбора данных с веб-сайта, их обработки и сохранения в формате CSV.

**1. Разбор данных с веб-сайта**

Был осуществлён парсинг данных сайта с использованием библиотеки BeautifulSoup. Для автоматизированного получения данных использовался Selenium. Данные о погоде были извлечены с веб-сайта https://yandex.ru/pogoda/moscow.

Использованные библиотеки:

* requests – для отправки HTTP-запросов;
* selenium – для автоматизации браузера и получения HTML-кода страницы;
* BeautifulSoup – для парсинга HTML-кода и извлечения нужных данных;
* csv – для сохранения данных в формате CSV.

**2. Код программы**

import requests

from selenium import webdriver

from bs4 import BeautifulSoup

import csv

import time

# Настройка Selenium

options = webdriver.ChromeOptions()

options.add\_argument("--headless") # Запуск браузера в фоновом режиме

driver = webdriver.Chrome(options=options)

# Открываем страницу

url = "https://yandex.ru/pogoda/moscow"

driver.get(url)

# Ожидание загрузки страницы

time.sleep(5)

# Заголовки для обхода блокировки

HEADERS = {

"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"

}

# Получение HTML-кода

html = driver.page\_source

soup = BeautifulSoup(html, "html.parser")

# Извлечение данных

state = soup.select\_one(".link\_\_condition").text.strip()

humidity = soup.select\_one(".term\_\_value").text.strip()

temp = soup.select\_one(".temp\_\_value").text.strip()

weather\_data = [[state, humidity, temp]]

# Закрытие браузера

driver.quit()

# Сохранение данных в CSV

with open("weather.csv", "w", newline="", encoding="utf-8-sig") as file:

writer = csv.writer(file)

writer.writerow(["Погода", "Влажность", "Температура"])

writer.writerows(weather\_data)

print("Данные успешно сохранены в weather.csv!")

**3. Формирование датасета**

После выполнения парсинга был сформирован датасет в табличном виде, содержащий сведения о погодных условиях:

* **Состояние погоды** (ясно, облачно, дождь и т. д.);
* **Влажность** (в процентах);
* **Температура** (в градусах Цельсия).

Пример сформированного датасета:

Погода,Влажность,Температура

Облачно,75%,-3°C

**4. Сохранение данных**

Датасет был сохранён в формате CSV, обеспечивая удобный доступ к собранной информации. Формат CSV позволяет легко импортировать данные в различные аналитические системы и таблицы.

Файл weather.csv был успешно создан и содержит актуальные данные о погоде в Москве.

**Лабораторная работа выполнена успешно.**