# The Superior University, Lahore

## LAB-TASK (1) (Spring-2026)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Course Title: | Programming for AI (Lab) | | | Course Code: | - | Credit Hours | 1 |
| Instructor: | Sir Rasikh | | | Program: | BSDS | | |
| Semester: | 4th | Batch: | Morning | Section: | 4A | Date: | - |
| Time Allowed: | - | | | Maximum Marks: | | N/A | |
| Student's Name: | Shumaila Maryam | | | Roll No. | SU92-BSDSM-S25-062 | | |

**<u>Work Assigned:</u>**

**Question # 01:**

**Develop a web scraper that extracts all available email addresses from a given URL. The program should crawl the webpage, identify valid email patterns, and collect them efficiently. After extraction, the scraped email addresses must be stored in a structured Excel file for further use. Ensure the solution handles different webpage formats and avoids duplicate entries.**

DEPARTMENT OF
SOFTWARE
ENGINEERING

# Email Scrapper - Web Application

## Introduction:

The Email Scrapper is a professional web-based application designed to extract email addresses from websites efficiently and automatically. Built using Python and Flask framework, this tool provides a modern, user-friendly interface where users can input website URLs and instantly extract all email addresses found on those pages. The extracted emails can be viewed on screen and exported to Excel files for further use.

## Why I Made This Program:

The main idea behind this program is to practice and demonstrate concepts of web scraping, web development, and full-stack application building in Python. Emails were chosen as the target because collecting contact information is a common real-world need. By making this program, I wanted to:

- Learn how to scrape data from websites using BeautifulSoup library.
- Understand how to build web applications using Flask framework.
- Gain experience in creating modern user interfaces with HTML, CSS, and JavaScript.
- Develop skills in REST API design and JSON data handling.
- Learn how to generate and export data to Excel files using OpenPyXL.
- Practice using regular expressions (regex) for pattern matching.

This program serves both as a learning project and a practical tool, helping to strengthen programming skills while solving a real-world problem.

## How It Works:

**1. Starting the Application**

The user runs the Flask application by executing "python app.py" in the terminal. The server starts and the web interface becomes accessible at http://localhost:20002.

**2. Entering URL**

The user enters a website URL in the input field. The application accepts URLs with or without the http:// prefix and automatically adds it if missing.

**3. Scraping Process**

When the user clicks "Start Scraping", JavaScript sends a request to the Flask backend. The server uses the Requests library to fetch the webpage content and BeautifulSoup to parse it.

**4. Email Extraction**

The InfoReader module uses regular expressions to find all email patterns in the page content. The regex pattern matches standard email formats like username@domain.com. All unique emails are collected.

**5. Displaying Results**

The extracted emails are sent back to the frontend as JSON. JavaScript displays the results in beautiful cards showing the number of emails found and listing each email address.

**6. Excel Export**

Users can click "Export to Excel" to download the results. The backend generates an Excel file with two columns: URL and Email. The file is formatted with styled headers and proper column widths.

**7. Multiple URLs**

The application also supports batch processing. Users can enter multiple URLs (one per line) and scrape all of them at once. Results are shown for each URL separately.

## Summary:

1. The Email Scrapper is a Python web application built with Flask that extracts email addresses from any website. It features a modern dark-themed interface with glassmorphism effects, animated backgrounds, and responsive design that works on all devices.

2. The project uses BeautifulSoup for web scraping, regular expressions for email pattern matching, and OpenPyXL for generating Excel exports. It supports both single URL and bulk URL processing.

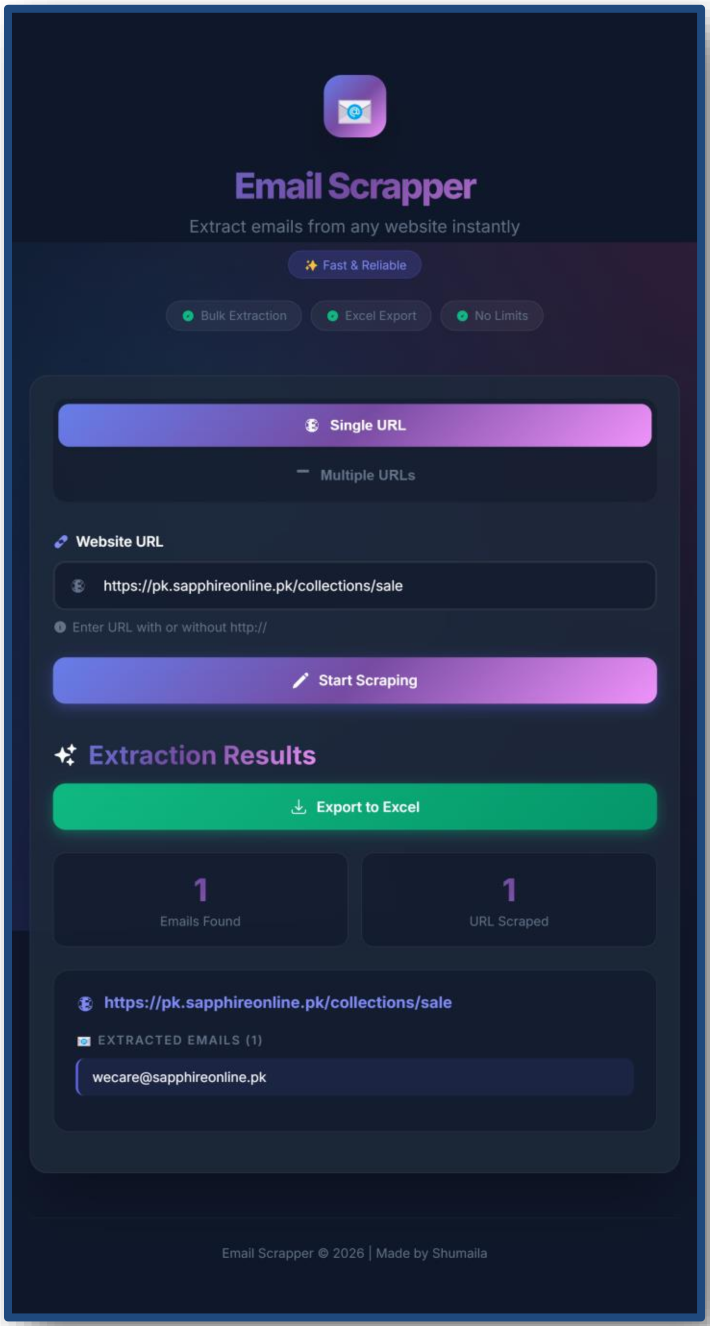
3. This project was created to practice Python web development, full-stack application building, and modern UI/UX design while applying them to solve a practical problem of collecting email addresses from websites efficiently.

## Technologies Used:

- Python 3.9+ - Backend programming language
- Flask - Web framework for server and routing
- BeautifulSoup4 - HTML parsing and web scraping
- Requests - HTTP library for fetching web pages
- OpenPyXL - Excel file generation

- HTML5/CSS3 - Frontend structure and styling
- JavaScript - Frontend interactivity and API calls

## **Output:**



Email Scrapper

Extract emails from any website instantly

✨ Fast & Reliable

● Bulk Extraction ● Excel Export ● No Limits

Single URL

Multiple URLs

Website URL

https://pk.sapphireonline.pk/collections/sale

Enter URL with or without http://

Start Scraping

Extraction Results

Export to Excel

1 Emails Found

1 URL Scraped

https://pk.sapphireonline.pk/collections/sale

EXTRACTED EMAILS (1)

wecare@sapphireonline.pk

Email Scrapper © 2026 | Made by Shumaila