In this program, we first import the necessary modules (**re** for regular expressions and **requests** for fetching HTML content from a URL). The **extract_data_from_url** function takes a URL as input, fetches the HTML content from that URL, and then uses regular expressions to search for emails, phone numbers and dates of different formats within the HTML content. The functions uses the `requests.get()` method to send an HTTP GET request to the specified URL and retrieve the response.

I) Email Address:

Regular expression pattern to match email addresses **\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b**

- \b denotes a word boundary to ensure the pattern matches whole email addresses.
- [A-Za-z0-9._%+-]+ matches one or more alphanumeric characters, dots, underscores, percent signs, plus signs, or hyphens, representing the email username.
- @ matches the "@" symbol.
- [A-Za-z0-9.-]+ matches one or more alphanumeric characters, dots, or hyphens, representing the domain name.
- \. matches a literal dot.
- [A-Za-z]{2,} matches two or more letters, representing the top-level domain.

The **re.findall()** function is used to find all matches of the email pattern in the **text_content**. It returns a list of email addresses. The list of extracted email addresses is returned by the function. The **extract_data_from_url()** function is called with the **url** as an argument, and the returned list of email addresses is stored in the **email_addresses** variable. Finally, a loop is used to iterate over the **email_addresses** list, and each email address is printed one by one.

II) Phone Numbers:

Two regular expression patterns are used to match phone numbers:

- **pattern1** is used to match phone numbers in the format "+91-xxxxxxxxxx", "xx-xxxxxxxx", "xxx-xxxxxxx", or "xxxx-xxxxxx".
- **pattern2** is used to match phone numbers in the format "xxx-xxx-xxxx".

The **re.findall()** function is used to find all matches of the phone number patterns in the **response.text**. It returns two lists: **tel_numbers** for landline numbers and **mobile_numbers** for mobile numbers. The function retaurns the extracted landline and mobile numbers as a tuple. The **extract_data_from_url()** function is called with the **url** as an argument, and the returned tuple of phone numbers is stored in the **phone_numbers** variable. Finally, two loops are used to iterate over the **phone_numbers** tuple, and each phone number is printed one by one.

III) Dates:

- A list **date_formats** is defined, containing different regular expressions to match date formats. The regular expressions cover formats such as "YYYY-MM-DD", "DD-MM-YYYY", "DD/MM/YYYY", "DD/MM/YY", and "DD-MM-YY".

An empty list **extracted_dates** is created to store the extracted dates. The code iterates over each **date_format** in the **date_formats** list and uses the **re.findall()** function to find all matches of the date format in the **html_content**. The extracted dates are stored in the **dates** variable. The **extend()** method is used to add the extracted dates to the **extracted_dates** list.

Once all date formats have been processed, the **extracted_dates** list is returned. The **extract_data_from_url()** function is called with the **url** as an argument, and the returned list of extracted dates is stored in the **dates** variable. Finally, a loop is used to iterate over the **dates** list, and each date is printed one by one.