**Abdul Rehman 04071913025**

**M. Afaq Khan 04071913031**

**Assignment No. 02**

**Theory of Automata**

**Part – 1:**

**Language Used:** Python

**Libraries Used:** numpy, requests, re, bs4

**Regular Expressions:**

* Name: *‘(Dr\.\s[a-zA-Z\s\.]+(Associate|Assistant))|([a-zA-Z\s\.]+Lecturer)’*
* Phone: *‘\+92\-51\-9064\s[\d]{4}’*
* Email: *‘[a-z\.]+\sat’*

The regular expression of name will choose all the strings starting with Dr. followed by one or more of occurrences of alphabets or spaces and ending with either Associate or Assistant; or will accept the strings starting with one or more occurrences of alphabets or spaces and ending with Lecturer.

The regular expression of phone will accept all the strings starting with +92 then one occurrence of – then 51 then again – then 9064 then one occurrence of space followed by any four-digit number.

The regular expression of email will extract the usernames of all the faculty members starting with one or more occurrence of small alphabets or dot and ending with at.

**Explanation:**

Firstly, using the requests library we are requesting to get the data of the webpage using the get method, then we will extract the HTML content of the webpage. After that we will create a soup object through this HTML content using HTML parser. Then we will extract the text of the table tag in HTML where the record of the faculty is displayed.

Once the text is extracted, we will apply the regular expressions one by one on that text. As we can see in the name pattern, the names are either followed by Assistant or Associate or Lecturer. To remove that we have used the replace function and to remove the extra spaces or new line characters from the names we have split them using split function, this will split the string based on non-space or new line characters and then we have joined these splitted strings with a single space. The strings extracted by the phone pattern will be used as it is. The usernames extracted from the email pattern are ending with at, which is not required in the email so that we will remove that using replace function and then concatenating @qau.edu.pk to that username to form a valid email address.

Once the names, phone numbers and emails are in proper format then we are writing it to the text file using the basic file operations.

**Output:**

A screenshot of a computer

Description automatically generated with medium confidence

**Part – 2:**

**Language Used:** Python

**Libraries Used:** docx, re, docx2pdf

**Regular Expressions:**

* AM Time: *‘[\d]+\:[\d]+[\s]\*AM’*
* PM Time: *‘[\d]+\:[\d]+[\s]\*PM’*

The AM Time regular expression will accept any number of digits followed by : then any number of digits followed by any occurrence of space and ending with AM.

The PM Time regular expression will accept any number of digits followed by : then any number of digits followed by any occurrence of space and ending with PM.

**Explanation:**

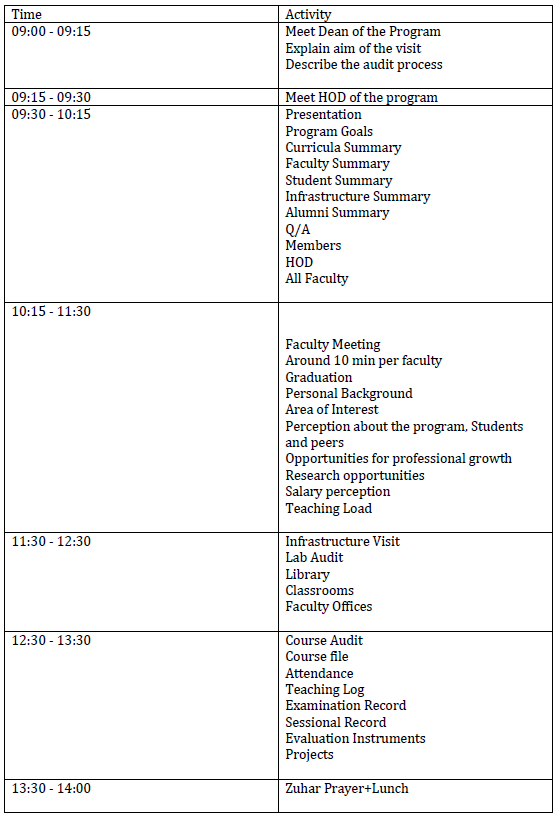
Firstly, we will create an object of Document using the .docx file we have got, named ‘schedule.docx’. Then we will extract the table from the document and store its data based on columns in a list of dictionaries where the key is the column name, and the data is the value on that row. Then we will concatenate all the table data in the form of a string and will apply the regular expressions of AM Time and PM Time on them one by one.

Once the AM and PM Times are extracted, we will simply remove the AM from AM Times using the replace function and will remove the PM from PM Times using replace function as well as replacing the hour part of PM Times with adding 12 in their hour part, using the similar functions.

Once, we have the updated lists of AM and PM Times, we will merge them in one list and update that list in the data dictionary using proper index of each time.

Then, we will create a new document and create a table in that using the updated data dictionary and save it as ‘new-schedule.docx’ and then will convert it into .pdf file.

**Output:**



Table

Description automatically generated