VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

WEB MINING (CSE 3024)

Slot: B1+TB1

A GENERALISED WEB SCRAPING MECHANISM FOR TRAVEL RELATED WEBSITES

Submitted to: - Professor. Balakrishnan P.

Associate Professor

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Group members:-

Rishi Vamshi	15BCE0327
Devashish Mital	15BCE0260
Ishas Prasad Dikshit	15BCE0436
J V Sai Srikar	15BCE0520
Akhil Bharadwaj	15BCE0029

Index

ACKNOWLEDGEMENT

We would like to thank our Professor of **Web Mining, Balakrishnan P. sir** for giving us this opportunity to make an interactive **Web Mining** project which has not only helped better our coding skills but has also made our constructive and building skills refined. We would like to thank the **VIT Periyar Central library** for having a good amount of books for project development on Web Scrapping mechanism and also teachers.

ABSTRACT

Web scraping is a very popular and growing field. The presence of web scraping in travel related websites is gigantic. The advantages and difficulties of web information scrapping are significant to everybody in the travel business. Meta look locales, OTAs and travel providers are altogether being rejected and the most fascinating truth is that more than 30% of individuals who visit travel industry sites are web scrappers.

Through a generalised web scraping mechanism using selenium web driver people can easily scrap the website data without even opening the browser. Selenium is one of the most popular tools used nowadays to scrap the websites. Today, there are more advanced web data scrapping techniques and skills that enable you to visit the websites and get the data comfortably and easily (as mentioned above selenium is one of those web drivers). There is also software that scraps the web automatically and records the data.

For a very long time now, Bing has indexed travel suppliers. It has been also able to provide a direct inventory links to the resulting search results. The same route has been followed by Google as well. Fraud bot, which is bad, takes advantage of your system by looking at the vulnerable areas. These are those expensive hackers that fool websites by clicking ads, filling forms and look for other weaknesses in website.

INTRODUCTION

Web scraping is a most general and widely used field in the current day scenario. Mostly, web scraping in the current day scenario is done in order to save a lot of time. As a matter of fact it does save a huge amount of time. In our project we attempt to scrap some relevant travel related websites such as makemytrip.com, yatra.com, goibibo.com etc using he corresponding classes in order to extract the website data. The scraping tool used to scrap the website data is selenium web driver. The user interface we make is using the tkinter library, which is the most popular and commonly used python library. It provides a robust and highly profound method in order to make user interfaces. According to our implementation the user will enter the flight details and get a sorted output based on the searched query for the flight availability.

PROBLEM STATEMENT

To generalise a web scraping mechanism for travel related websites. A popular demonstration of the capability of automating websites using selenium drivers. It is used for automating web applications for testing purposes, specifically for travel related websites in this problem. There is need to provide a successful and a user friendly interface to users so that they can easily without even opening the website be able to get the flight details and make a comparison between the flight fares and book a good flight accordingly to their needs and requirements. This saves a huge amount of time for the customer as there is no need for the customer to visit different travel related websites and check for the corresponding flight fares and details, instead of that a single query inputted by the user would him/her a lot in order to view a broad range of categories.

TKINTER

Tkinter, a popular and a standard Graphical user Interface library for the python. When tkinter library is combined or integrated with python it provides a very quick and simple mechanism to create a Graphical User Interface Application. It provides a highly powerful object oriented interface to the GUI toolkit of Tk.

The mechanism for creating a simple GUI application on tkinter.

- **a.** Import the module tkinter.
- **b.** Create a Graphical user interface application main window.
- **c.** One can add the discussed modules or widgets to the application GUI.
- **d.** Then one has to enter an event loop so that a suitable action against the event can be taken which is triggered by the user.

Functionalities provided by tkinter:-

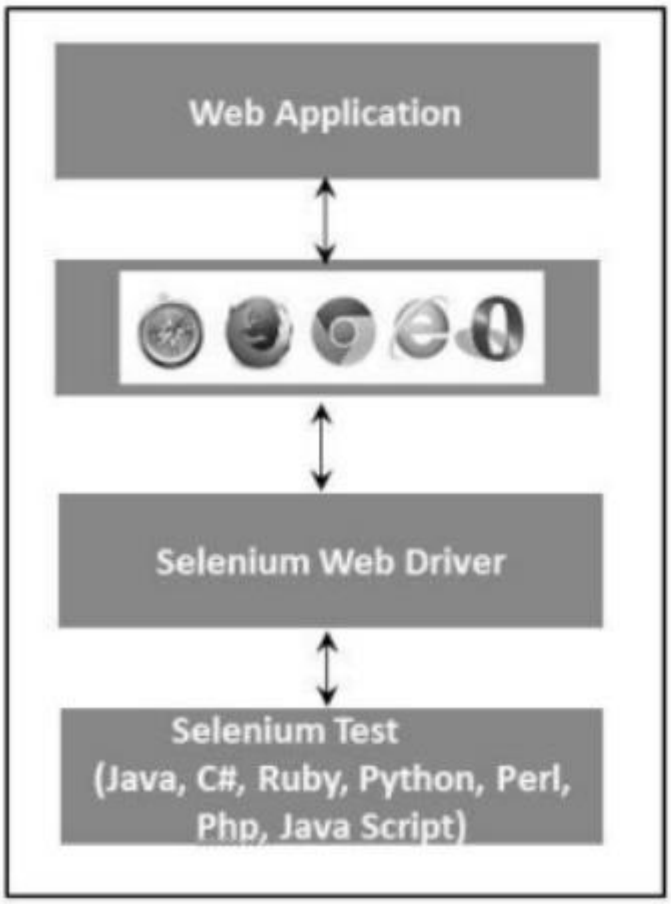
1. It provides classes for displaying and controlling the widgets.
2. Properties of widgets are specified using with keyword arguments.
3. Keyword arguments have the same name as the corresponding resource under Tk.
4. Top level widgets are Tk .
5. Widgets are being positioned with managers of geometry like Place, Pack or Grid. Methods such as place, pack and grid are called by these managers.

SELENIUM

Selenium is the tool used in order to automate testing of web applications. It is most commonly known as Selenium 2.0. This web driver uses an underlying framework which is very different when compared to selenium RC, as selenium RC uses a JavaScript embedded selenium core which has some amount of considerable limitations. This selenium web driver interacts with the web browser directly without the use of any sort of intermediary, whereas selenium RC has a limitation as it is completely dependent on the server. The context in which selenium web driver comes into action is as follows:-

1. It has the ability to handle multiple frames, popups, multi browser window, and some alerts.
2. Ability to navigate complex web pages.
3. Some advances navigation techniques such as drag and drop.
4. AJAX based User Interface elements.
5. It is also used in the multiple browser testing including an improved functionality for the browsers which is not at all well – supported by selenium.

ARCHITECTURE OF SELENIUM WEB DRIVER



Code and Output

Tables.py

```
from tkinter import *

import csv

def createStandardTable(f,window):

    handle = csv.reader(f)

    length = len(next(handle))


    sizes = [0] * length

    for record in handle:


        for p,column in enumerate(record):

            if len(column) > sizes[p]:

                sizes[p] = len(column)+3


    f.seek(0)

    trow = 0

    table = Frame(window)

    for record in handle:

        for w,column in enumerate(record):

            Label(table,text=column,width=sizes[w],border-
width=2,relief="groove",justify=LEFT,anchor=W, back-
ground='white').grid(column=w,row=trow,sticky=W)


        trow+=1


    return table
```

using_tables_with_frames.py

```
# Importing Libraries- Tkinter, CSV, numpy and pandas
from tkinter import *
from tkinter import filedialog
import csv
from tables import createStandardTable as cst
import numpy as np
import pandas as pd


# Declaring Variables of list datastructure for sorting.
pr, tim, air, depart, arrival, source, destination, duration, stopage, prgo = [], [], [], [], [], [], [], [], [], []
timgo, airgo, sourcegp, destgo, dash, flightnumgo, airname = [], [], [], [], [], [], []
departgo, arrivego, airname, flynum, finalgo, stopage2, final, airname1, flight1 = [], [], [], [], [], [], [], [], []



#Creating TK() object
window = Tk()


# Declaring all the functions
# Functions for taking inputs
def inputFrom():
    mText=fromCity.get()
    #mlabel1 = Label(tableFrame3,text=mText).grid()
def inputTo():
    mText=toCity.get()
    #mlabel1 = Label(tableFrame3,text=mText).grid()
def inputDepDate():
    mText=departDate.get()
    #mlabel1 = Label(tableFrame3,text=mText).grid()
```

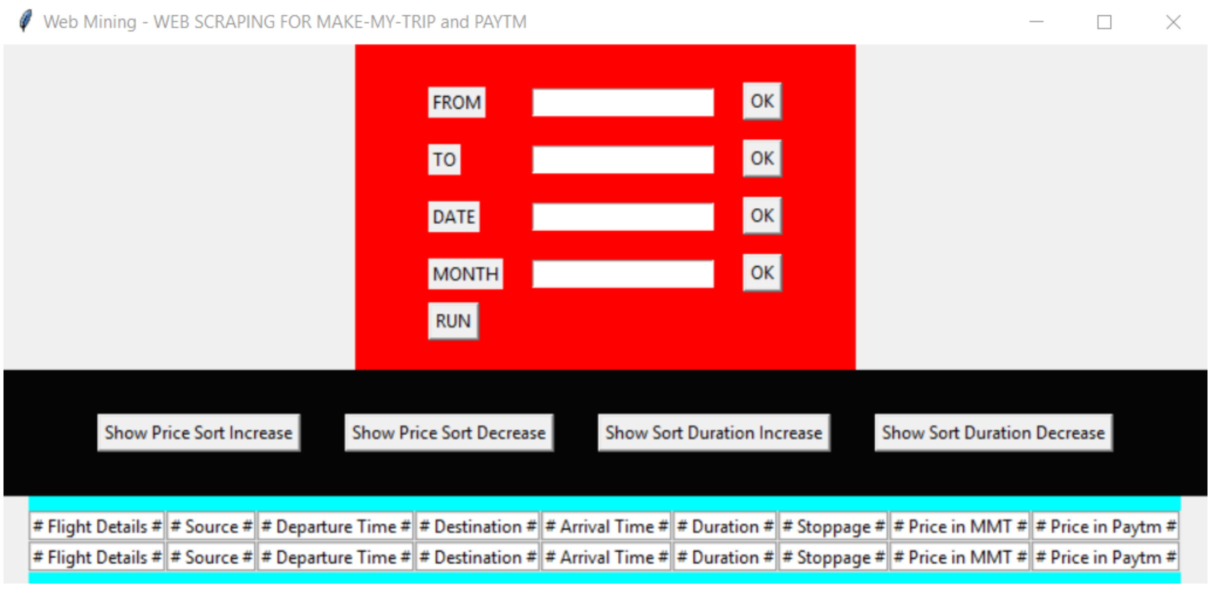
```
def inputDepMon():
    mText=departMonth.get()
    #mlabel1 = Label(tableFrame3,text=mText).grid()
def mAbout():
    messagebox.showinfo(title="About",message="This is the
about box")
def mQuit():
    mExit = messagebox.askyesno(title="Quit", message="Are you
sure")
    if mExit > 0:
        window.destroy()


# Functions For Sorting
def price_sort_increase(final):
    s=sorted(final,key=lambda x:x[7], reverse=False)
    return s
def price_sort_decrease(final):
    s=sorted(final,key=lambda x:x[7], reverse=True)
    return s
def sort_durationInc(flist):
    s=sorted(flist,key=lambda
x:int(str(x[5][0])+str(x[5][3])+str(x[5][4])), reverse=False)
    return s
def sort_durationDec(flist):
    s=sorted(flist,key=lambda
x:int(str(x[5][0])+str(x[5][3])+str(x[5][4])), reverse=True)
    return s


MMT = ["HYD", "MAA","BOM", "DEL", "CCU", "AMD", "BLR", "NAG",
"CCJ", "PNQ"]
```

```
Ptm= ["HYD-Hyderabad","MAA-Chennai","BOM-Mumbai","DEL-
Delhi","CCU-Kolkata","AMD-Ahmedabad","BLR-Bengaluru","NAG-Nag-
pur","CCJ-Kozhikode","PNQ-Pune"]


# Main Function where all the process happens

def run1():

    mtext1 = fromCity.get()

    mtext2 = toCity.get()

    mtext3 = departDate.get()

    mtext4 = departMonth.get()

    ptext1 = Ptm[MMT.index(mtext1)]

    ptext2 = Ptm[MMT.index(mtext2)]

    url = "https://flights.makemytrip.com/make-
mytrip/search/O/O/E/1/0/0/S/V0/"+mtext1+"_"+mtext2+"_"+mtext3+
"-"+mtext4+"-2017?contains=false&remove="

    url2= "https://paytm.com/flights/flight-
Search/"+ptext1+"/"+ptext2+"/1/0/0/E/2017-"+mtext4+"-"+mtext3

    from selenium import webdriver

    chrome_path = r"C:\Python27\chromedriver.exe"

    driver = webdriver.Chrome(chrome_path)

    driver2= webdriver.Chrome(chrome_path)

    driver.get(url)

    driver2.get(url2)


    ##Getting data for make my trip

    airline_info = driver.find_elements_by_class_name("air-
line_info_detls")

    price = driver.find_elements_by_class_name("num")

    time = driver.find_elements_by_class_name("time_info")

    for post in price:

        price=post.text

        price=price.replace(',','')

        price=int(price)
        pr.append(price)
        dash.append("-")
    for post in airline_info:
        airline=post.text
        airline=airline.replace('\n',' ')
        air.append(airline)
    for post in time :
        t=post.text
        tim.append(post.text)
    conarray=[i+"\n"+j+"\n"+k for i,j,k in zip(tim[::3],
tim[1::3],tim[2::3])]
    for element in conarray:
       parts = element.split('\n')
       depart.append(parts[0])
       source.append(parts[1])
       arrival.append(parts[2])
       destination.append(parts[3])
       duration.append(parts[4])
       stopage.append(parts[5])
    for k in range(0,len(pr)):
        final.append([air[k],source[k],depart[k],destina-
tion[k],arrival[k],duration[k],stopage[k],pr[k],dash[k]])
    newlist=sorted(final,key=lambda x:x[7], reverse=False)

    ##Getting data for Paytm

    airline_info2 = driver2.find_elements_by_class_name("_3H-
S")
    price2 = driver2.find_elements_by_class_name("_2gMo")
    fly=driver2.find_elements_by_class_name("NqXj")
    time2 = driver2.find_elements_by_class_name("vY4t")
    stopgo=driver2.find_elements_by_class_name("_7BOG")
    for post in price2:
        pri=post.text
        pri=pri.replace(',','')
        pri=int(pri)
        prgo.append(pri)
    for post in stopgo:
        stopage2.append(post.text)
    for post in airline_info2:
        airgo.append(post.text)
    for post in time2 :
        timgo.append(post.text)
    for post in fly :
        flynum.append(post.text)

    airname=airgo[::3]
    departgo=airgo[1::3]
    arrivego=airgo[2::3]
    flightnumgo=flynum[::4]
    flightnumgo = [w.replace(' ', '') for w in flightnumgo]
    sourcego=flynum[1::4]
    destgo=flynum[2::4]

    conarray2=[i+" "+j for i,j in zip(airname, flightnumgo)]
    for i in range(0,len(prgo)):
      finalgo.append([conarray2[i],sourcego[i], de-
partgo[i],destgo[i],arrivego[i],timgo[i],stopage2[i],prgo[i]])
      newlist2=sorted(finalgo,key=lambda x:x[7], re-
verse=False)

    ##Checking and storing price for flights in Paytm into the
array
    for j in range(0,len(newlist)):
        for i in range(0,len(newlist2)):
            if(newlist2[i][0]==newlist[j][0]):
                newlist[j][8]=(newlist2[i][7])
                break

    ## Exporting all the values into individual CSV files
    finallist = newlist[:15]

    my_df1 = pd.DataFrame(finallist)
    my_df1.to_csv('finallist.csv', index=False, header=False)

    price_sort_increase1 = price_sort_increase(finallist)

    my_df2 = pd.DataFrame(price_sort_increase1)
    my_df2.to_csv('priceSortIncrease.csv', index=False,
header=False)

    price_sort_decrease1 = price_sort_decrease(finallist)

    my_df3 = pd.DataFrame(price_sort_decrease1)
    my_df3.to_csv('priceSortDecrease.csv', index=False,
header=False)

    sortDurationInc1 = sort_durationInc(finallist)

    my_df4 = pd.DataFrame(sortDurationInc1)
    my_df4.to_csv('sort_durationInc.csv', index=False,
header=False)

    sortDurationDec1 = sort_durationDec(finallist)

    my_df5 = pd.DataFrame(sortDurationDec1)
    my_df5.to_csv('sort_durationDec.csv', index=False,
header=False)



## Creating Frames for input, Sorting buttons and output

tableFrame = Frame(window, bg='cyan', width=450, height=50,
pady=10)

tableFrame2 = Frame(window, bg='gray2', width=450, height=40,
padx=50, pady=20)

tableFrame3 = Frame(window, bg = 'red', width = 450, height =
300, padx= 50, pady = 20)


# Creating the Variables for Inputs

fromCity = StringVar()

toCity = StringVar()

departDate = StringVar()

departMonth = StringVar()


#Declaring the title for the window

window.title('Web Mining - WEB SCRAPING FOR MAKE-MY-TRIP')



def tableFrames():

    tableFrame3.grid()

    tableFrame2.grid()

    tableFrame.grid()


#create a global newtable grid.

f = open("my_csv.csv")

newtable = cst(f,tableFrame)


def createTableFrame():
    f = open("sort_durationDec.csv")
    global newtable
    newtable = cst(f,tableFrame)
    newtable.grid()

def Header():
    f = open("my_csv.csv")
    global newtable
    newtable = cst(f,tableFrame)
    newtable.grid()

def createTableFrame2():
    f = open("my_csv.csv")
    global newtable
    newtable = cst(f,tableFrame)
    newtable.grid()

def createTableFrame3():
    f = open("priceSortIncrease.csv")
    global newtable
    newtable = cst(f,tableFrame)
    newtable.grid()

def createTableFrame4():
    f = open("priceSortDecrease.csv")
    global newtable
    newtable = cst(f,tableFrame)
    newtable.grid()

def createTableFrame5():
    f = open("sort_durationInc.csv")
    global newtable
    newtable = cst(f,tableFrame)
    newtable.grid()

def ct():
    global newtable
    newtable.destroy()
    createTableFrame3()
    tableFrame.grid()

def th():
    global newtable
    newtable.destroy()
    createTableFrame()
    tableFrame.grid()

def t2h():
    global newtable
    newtable.destroy()
    createTableFrame4()
    tableFrame.grid()

def t3h():
    global newtable
    newtable.destroy()
    createTableFrame5()
    tableFrame.grid()

def sorting():
    Button(tableFrame2,text="Show Price Sort Increase",com-
mand=ct).grid(row = 0,column = 0 , padx = 15, pady = 10)
```

```
    Button(tableFrame2,text = "Show Price Sort Decrease", com-
mand = t2h).grid(row = 0,column = 1 , padx = 15, pady = 10)

    Button(tableFrame2,text = "Show Sort Duration Increase",
command = t3h).grid(row = 0,column = 2 , padx = 15, pady = 10)

    Button(tableFrame2,text = "Show Sort Duration Decrease",
command = th).grid(row = 0,column = 3 , padx = 15, pady = 10)







# Input

mLabel1 = Label(tableFrame3,text='FROM')

mLabel1.grid(row=0,column = 0,sticky = W )

mEntry1 = Entry(tableFrame3,textvariable=fromCity).grid(row =
0,column = 1 , padx = 20, pady = 10)

mButton1 = Button(tableFrame3,text ='OK', command = in-
putFrom).grid(row=0,column = 2)


mLabel2 = Label(tableFrame3,text='TO').grid(row=1,column = 0,
sticky = W)

mEntry2 = Entry(tableFrame3,textvariable=toC-
ity).grid(row=1,column = 1 , pady = 10)

mButton2 = Button(tableFrame3,text ='OK', command = in-
putTo).grid(row=1,column = 2)


mLabel3 = Label(tableFrame3,text='DATE').grid(row=2,column =
0, sticky = W)

mEntry3 = Entry(tableFrame3,textvariable=de-
partDate).grid(row=2,column = 1, pady = 10)

mButton3 = Button(tableFrame3,text ='OK', command = inputDep-
Date).grid(row=2,column = 2)


mLabel4 = Label(tableFrame3,text='MONTH').grid(row=3,column =
0, sticky = W)
```

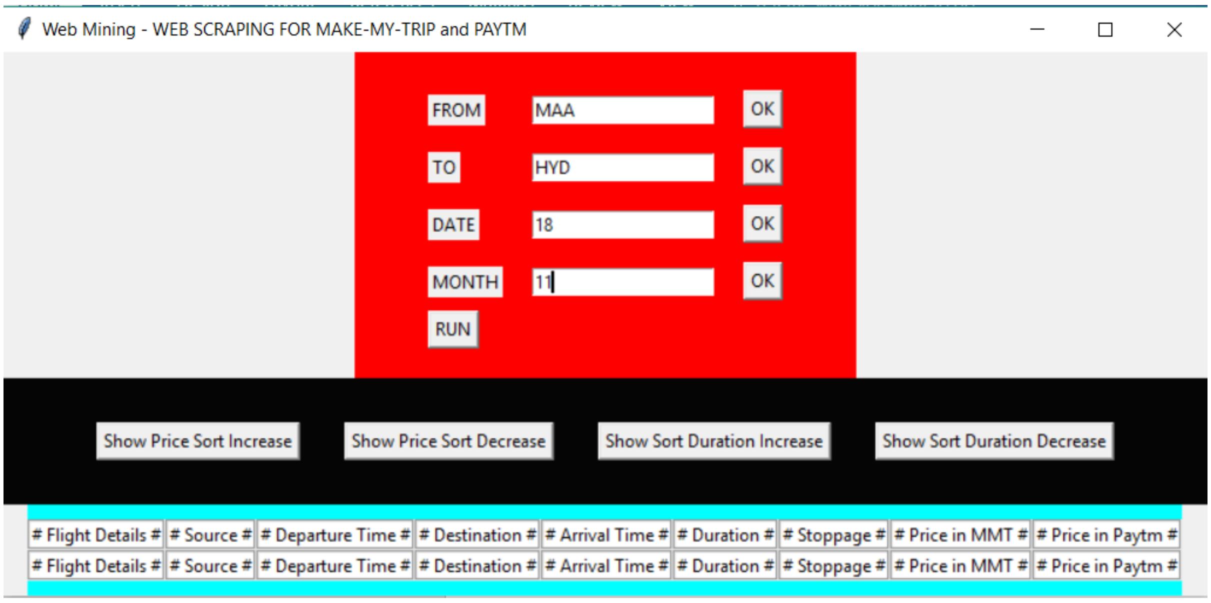
```
mEntry4 = Entry(tableFrame3,textvariable=depart-
Month).grid(row=3,column = 1, pady = 10)

mButton4 = Button(tableFrame3,text ='OK', command = inputDep-
Mon).grid(row=3,column = 2)



mRun = Button(tableFrame3,text ='RUN', command =
run1).grid(row = 4, column = 0, sticky = W)




tableFrames()

sorting()

Header()

createTableFrame2()




window.mainloop()
```

OUTPUT -


Web Mining - WEB SCRAPING FOR MAKE-MY-TRIP and PAYTM
FROM
OK
TO
OK
DATE
OK
MONTH
OK
RUN
Show Price Sort Increase
Show Price Sort Decrease
Show Sort Duration Increase
Show Sort Duration Decrease

```
# Flight Details # # Source # # Departure Time # # Destination # # Arrival Time # # Duration # # Stoppage # # Price in MMT # # Price in Paytm #
# Flight Details # # Source # # Departure Time # # Destination # # Arrival Time # # Duration # # Stoppage # # Price in MMT # # Price in Paytm #
```

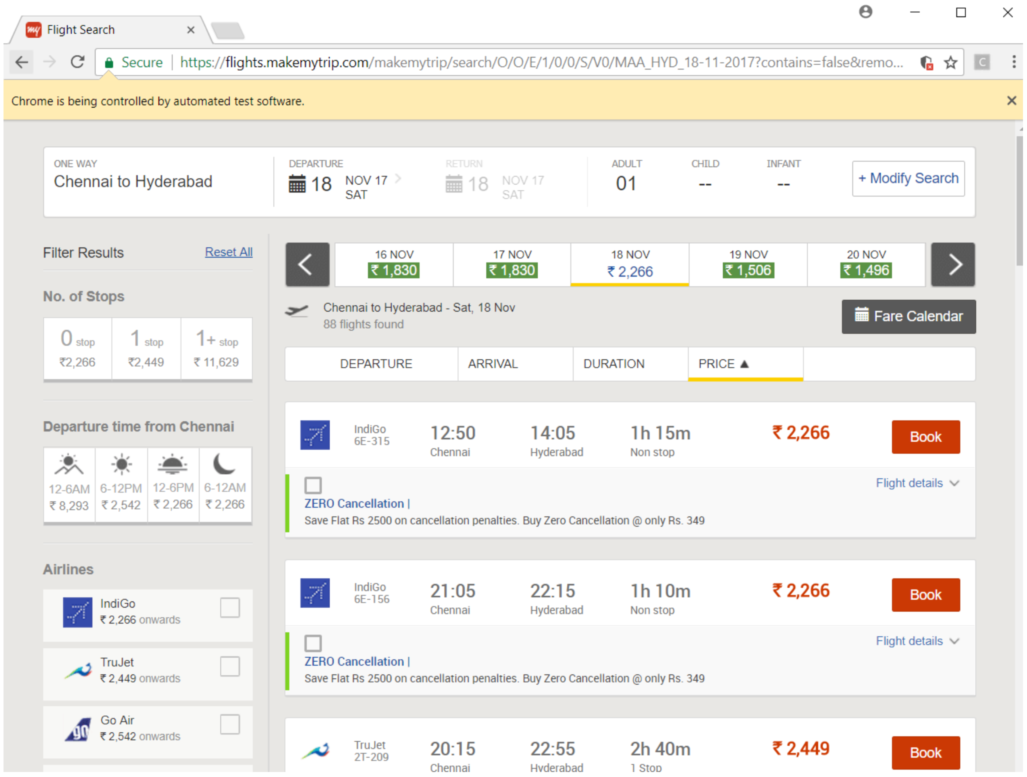
Figure 1 - Main Screen – Dashboard


Web Mining - WEB SCRAPING FOR MAKE-MY-TRIP and PAYTM
RUN
Show Price Sort Increase
Show Price Sort Decrease
Show Sort Duration Increase
Show Sort Duration Decrease

FROM MAA OK
TO HYD OK
DATE 18 OK
MONTH 11 OK

```
# Flight Details # # Source # # Departure Time # # Destination # # Arrival Time # # Duration # # Stoppage # # Price in MMT # # Price in Paytm #
# Flight Details # # Source # # Departure Time # # Destination # # Arrival Time # # Duration # # Stoppage # # Price in MMT # # Price in Paytm #
```

Figure 2 - Giving inputs to get flight details from Chennai to Hyderabad on 18/11/2017.


Flight Search

Secure | https://flights.makemytrip.com/makemytrip/search/O/O/E/1/0/0/S/V0/MAA_HYD_18-11-2017?contains=false&remo...

Chrome is being controlled by automated test software.

ONE WAY Chennai to Hyderabad | DEPARTURE 18 NOV 17 SAT | RETURN 18 NOV 17 SAT | ADULT 01 | CHILD -- | INFANT -- | + Modify Search

Filter Results Reset All

No. of Stops

0 stop	1 stop	1+ stop
₹2,266	₹2,449	₹ 11,629

Departure time from Chennai

12-6AM	6-12PM	12-6PM	6-12AM
₹ 8,293	₹ 2,542	₹ 2,266	₹ 2,266

Airlines

IndiGo ₹ 2,266 onwards

TruJet ₹ 2,449 onwards

Go Air ₹ 2,542 onwards

16 NOV	17 NOV	18 NOV	19 NOV	20 NOV
₹ 1,830	₹ 1,830	₹ 2,266	₹ 1,506	₹ 1,496

Chennai to Hyderabad - Sat, 18 Nov
88 flights found

Fare Calendar

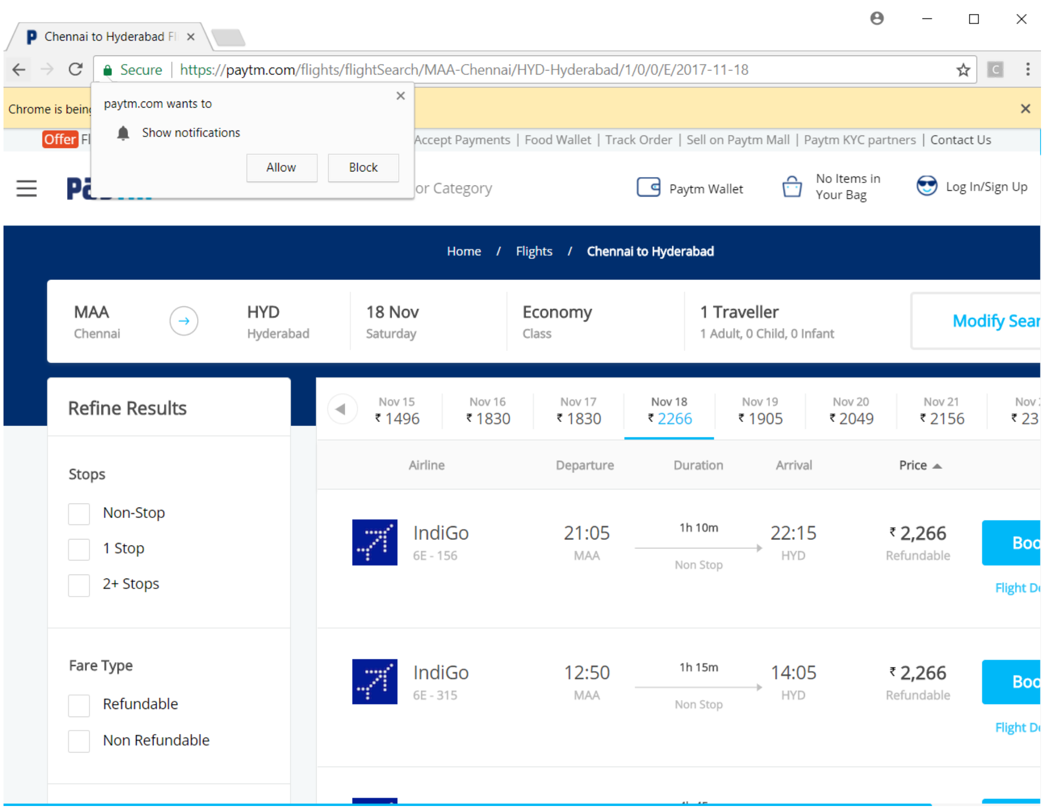
DEPARTURE | ARRIVAL | DURATION | PRICE ▲

Airline	Departure	Arrival	Duration	Price
IndiGo 6E-315	12:50 Chennai	14:05 Hyderabad	1h 15m Non stop	₹ 2,266
IndiGo 6E-156	21:05 Chennai	22:15 Hyderabad	1h 10m Non stop	₹ 2,266
TruJet 2T-209	20:15 Chennai	22:55 Hyderabad	2h 40m 1 Stop	₹ 2,449

ZERO Cancellation |
Save Flat Rs 2500 on cancellation penalties. Buy Zero Cancellation @ only Rs. 349

Flight details

Figure 3 - Make my trip screen


Chennai to Hyderabad Fl

Secure | https://paytm.com/flights/flightSearch/MAA-Chennai/HYD-Hyderabad/1/0/0/E/2017-11-18

Chrome is being

paytm.com wants to
Show notifications
Allow Block

Offer Fl | Accept Payments | Food Wallet | Track Order | Sell on Paytm Mall | Paytm KYC partners | Contact Us

or Category | Paytm Wallet | No Items in Your Bag | Log In/Sign Up

Home / Flights / Chennai to Hyderabad

MAA Chennai → HYD Hyderabad | 18 Nov Saturday | Economy Class | 1 Traveller 1 Adult, 0 Child, 0 Infant | Modify Sear

Refine Results

Stops
- Non-Stop
- 1 Stop
- 2+ Stops

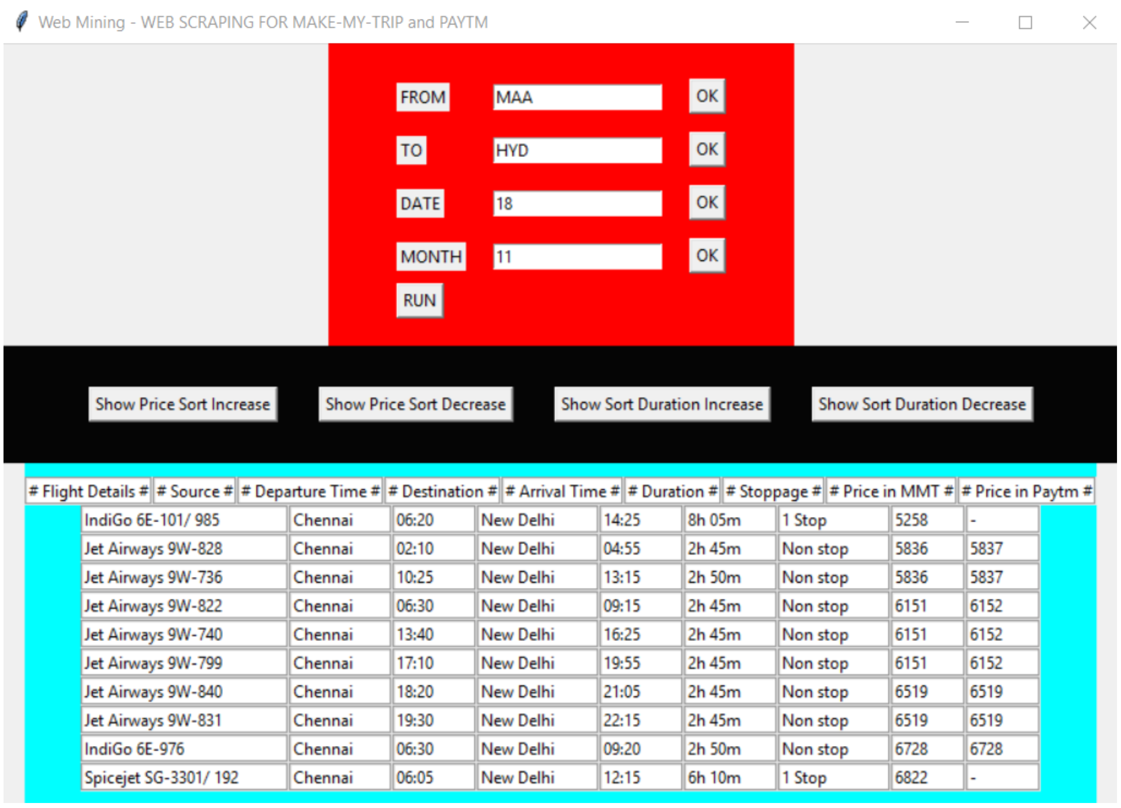
Fare Type
- Refundable
- Non Refundable

Nov 15	Nov 16	Nov 17	Nov 18	Nov 19	Nov 20	Nov 21	Nov
₹ 1496	₹ 1830	₹ 1830	₹ 2266	₹ 1905	₹ 2049	₹ 2156	₹ 23

Airline	Departure	Duration	Arrival	Price ▲
IndiGo 6E - 156	21:05 MAA	1h 10m Non Stop	22:15 HYD	₹ 2,266 Refundable
IndiGo 6E - 315	12:50 MAA	1h 15m Non Stop	14:05 HYD	₹ 2,266 Refundable

Boo
Flight D

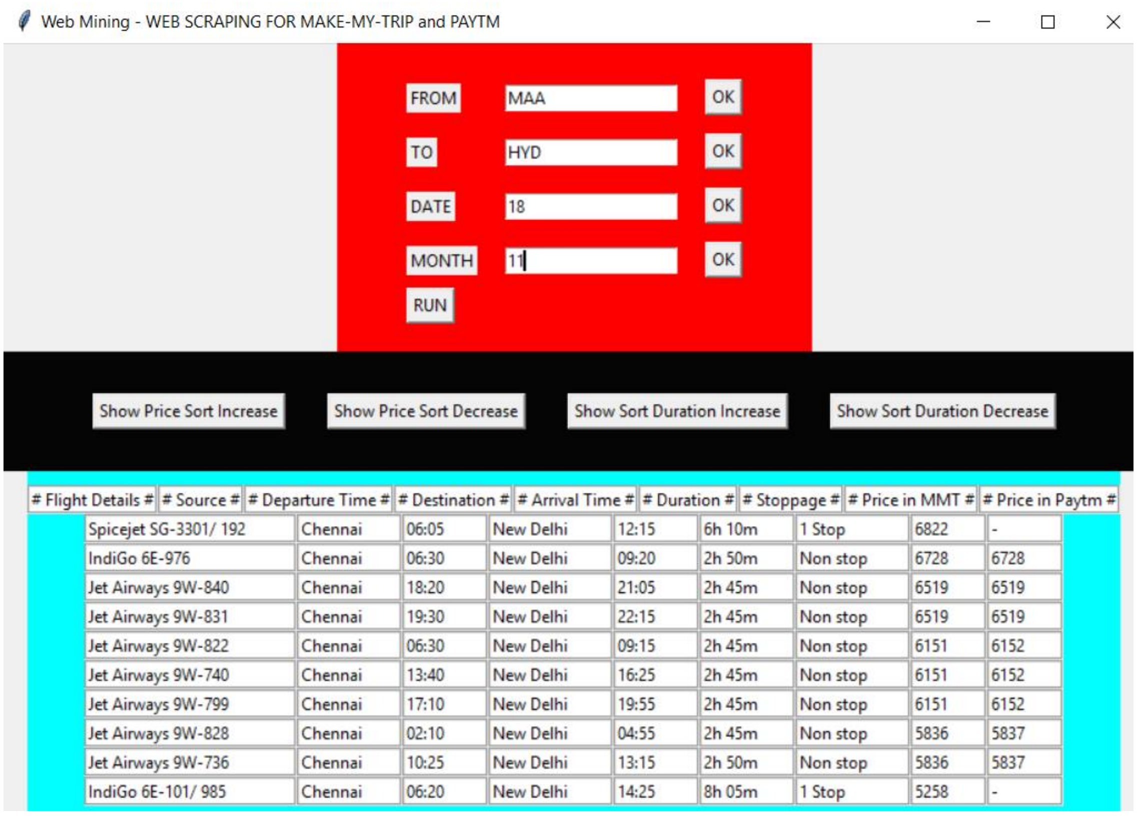
Figure 4 - PayTM Screen


Web Mining - WEB SCRAPING FOR MAKE-MY-TRIP and PAYTM
OK
OK
OK
OK
RUN
Show Price Sort Increase
Show Price Sort Decrease
Show Sort Duration Increase
Show Sort Duration Decrease

FROM MAA
TO HYD
DATE 18
MONTH 11

# Flight Details #	# Source #	# Departure Time #	# Destination #	# Arrival Time #	# Duration #	# Stoppage #	# Price in MMT #	# Price in Paytm #
IndiGo 6E-101/ 985	Chennai	06:20	New Delhi	14:25	8h 05m	1 Stop	5258	-
Jet Airways 9W-828	Chennai	02:10	New Delhi	04:55	2h 45m	Non stop	5836	5837
Jet Airways 9W-736	Chennai	10:25	New Delhi	13:15	2h 50m	Non stop	5836	5837
Jet Airways 9W-822	Chennai	06:30	New Delhi	09:15	2h 45m	Non stop	6151	6152
Jet Airways 9W-740	Chennai	13:40	New Delhi	16:25	2h 45m	Non stop	6151	6152
Jet Airways 9W-799	Chennai	17:10	New Delhi	19:55	2h 45m	Non stop	6151	6152
Jet Airways 9W-840	Chennai	18:20	New Delhi	21:05	2h 45m	Non stop	6519	6519
Jet Airways 9W-831	Chennai	19:30	New Delhi	22:15	2h 45m	Non stop	6519	6519
IndiGo 6E-976	Chennai	06:30	New Delhi	09:20	2h 50m	Non stop	6728	6728
Spicejet SG-3301/ 192	Chennai	06:05	New Delhi	12:15	6h 10m	1 Stop	6822	-

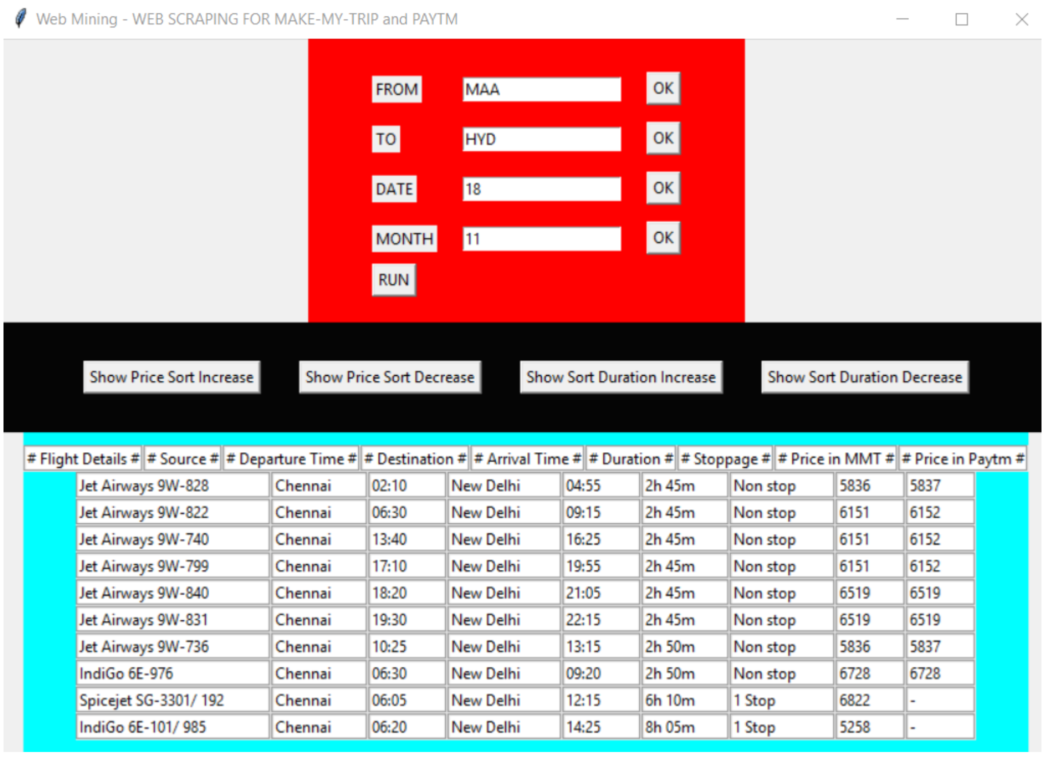
Figure 5 - Price Sort Increase


Web Mining - WEB SCRAPING FOR MAKE-MY-TRIP and PAYTM
OK
OK
OK
OK
RUN
Show Price Sort Increase
Show Price Sort Decrease
Show Sort Duration Increase
Show Sort Duration Decrease

FROM MAA
TO HYD
DATE 18
MONTH 11

# Flight Details #	# Source #	# Departure Time #	# Destination #	# Arrival Time #	# Duration #	# Stoppage #	# Price in MMT #	# Price in Paytm #
Spicejet SG-3301/ 192	Chennai	06:05	New Delhi	12:15	6h 10m	1 Stop	6822	-
IndiGo 6E-976	Chennai	06:30	New Delhi	09:20	2h 50m	Non stop	6728	6728
Jet Airways 9W-840	Chennai	18:20	New Delhi	21:05	2h 45m	Non stop	6519	6519
Jet Airways 9W-831	Chennai	19:30	New Delhi	22:15	2h 45m	Non stop	6519	6519
Jet Airways 9W-822	Chennai	06:30	New Delhi	09:15	2h 45m	Non stop	6151	6152
Jet Airways 9W-740	Chennai	13:40	New Delhi	16:25	2h 45m	Non stop	6151	6152
Jet Airways 9W-799	Chennai	17:10	New Delhi	19:55	2h 45m	Non stop	6151	6152
Jet Airways 9W-828	Chennai	02:10	New Delhi	04:55	2h 45m	Non stop	5836	5837
Jet Airways 9W-736	Chennai	10:25	New Delhi	13:15	2h 50m	Non stop	5836	5837
IndiGo 6E-101/ 985	Chennai	06:20	New Delhi	14:25	8h 05m	1 Stop	5258	-

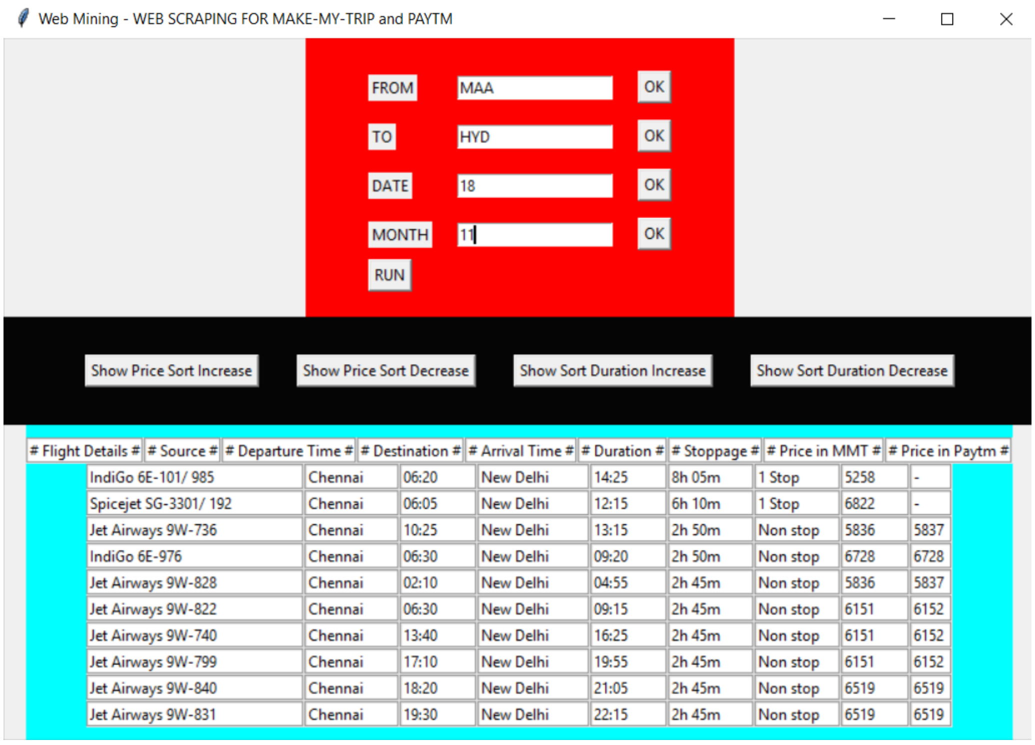
Figure 6 - Price Sort Decrease


Web Mining - WEB SCRAPING FOR MAKE-MY-TRIP and PAYTM
OK
OK
OK
OK
RUN
Show Price Sort Increase
Show Price Sort Decrease
Show Sort Duration Increase
Show Sort Duration Decrease

FROM MAA
TO HYD
DATE 18
MONTH 11

# Flight Details #	# Source #	# Departure Time #	# Destination #	# Arrival Time #	# Duration #	# Stoppage #	# Price in MMT #	# Price in Paytm #
Jet Airways 9W-828	Chennai	02:10	New Delhi	04:55	2h 45m	Non stop	5836	5837
Jet Airways 9W-822	Chennai	06:30	New Delhi	09:15	2h 45m	Non stop	6151	6152
Jet Airways 9W-740	Chennai	13:40	New Delhi	16:25	2h 45m	Non stop	6151	6152
Jet Airways 9W-799	Chennai	17:10	New Delhi	19:55	2h 45m	Non stop	6151	6152
Jet Airways 9W-840	Chennai	18:20	New Delhi	21:05	2h 45m	Non stop	6519	6519
Jet Airways 9W-831	Chennai	19:30	New Delhi	22:15	2h 45m	Non stop	6519	6519
Jet Airways 9W-736	Chennai	10:25	New Delhi	13:15	2h 50m	Non stop	5836	5837
IndiGo 6E-976	Chennai	06:30	New Delhi	09:20	2h 50m	Non stop	6728	6728
Spicejet SG-3301/ 192	Chennai	06:05	New Delhi	12:15	6h 10m	1 Stop	6822	-
IndiGo 6E-101/ 985	Chennai	06:20	New Delhi	14:25	8h 05m	1 Stop	5258	-

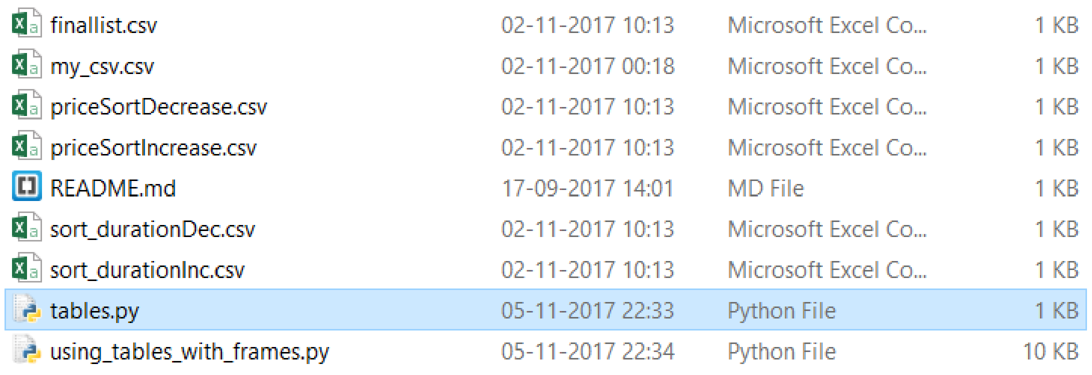
Figure 7 - Price Sort Duration Increase


Web Mining - WEB SCRAPING FOR MAKE-MY-TRIP and PAYTM
OK
OK
OK
OK
RUN
Show Price Sort Increase
Show Price Sort Decrease
Show Sort Duration Increase
Show Sort Duration Decrease

FROM MAA
TO HYD
DATE 18
MONTH 11

# Flight Details #	# Source #	# Departure Time #	# Destination #	# Arrival Time #	# Duration #	# Stoppage #	# Price in MMT #	# Price in Paytm #
IndiGo 6E-101/ 985	Chennai	06:20	New Delhi	14:25	8h 05m	1 Stop	5258	-
Spicejet SG-3301/ 192	Chennai	06:05	New Delhi	12:15	6h 10m	1 Stop	6822	-
Jet Airways 9W-736	Chennai	10:25	New Delhi	13:15	2h 50m	Non stop	5836	5837
IndiGo 6E-976	Chennai	06:30	New Delhi	09:20	2h 50m	Non stop	6728	6728
Jet Airways 9W-828	Chennai	02:10	New Delhi	04:55	2h 45m	Non stop	5836	5837
Jet Airways 9W-822	Chennai	06:30	New Delhi	09:15	2h 45m	Non stop	6151	6152
Jet Airways 9W-740	Chennai	13:40	New Delhi	16:25	2h 45m	Non stop	6151	6152
Jet Airways 9W-799	Chennai	17:10	New Delhi	19:55	2h 45m	Non stop	6151	6152
Jet Airways 9W-840	Chennai	18:20	New Delhi	21:05	2h 45m	Non stop	6519	6519
Jet Airways 9W-831	Chennai	19:30	New Delhi	22:15	2h 45m	Non stop	6519	6519

Figure 8 - Price Sort Duration Decrease



finallist.csv	02-11-2017 10:13	Microsoft Excel Co...	1 KB
my_csv.csv	02-11-2017 00:18	Microsoft Excel Co...	1 KB
priceSortDecrease.csv	02-11-2017 10:13	Microsoft Excel Co...	1 KB
priceSortIncrease.csv	02-11-2017 10:13	Microsoft Excel Co...	1 KB
README.md	17-09-2017 14:01	MD File	1 KB
sort_durationDec.csv	02-11-2017 10:13	Microsoft Excel Co...	1 KB
sort_durationInc.csv	02-11-2017 10:13	Microsoft Excel Co...	1 KB
tables.py	05-11-2017 22:33	Python File	1 KB
using_tables_with_frames.py	05-11-2017 22:34	Python File	10 KB

Figure 9 - All files

CONCLUSIONS:

We have successfully demonstrated how to extract data form travel related websites (like makemytrip and paytm) display the different flights available after giving the input for from and to, date and month. In this project we have displayed the different flights available for both the websites in both increasing and decreasing order of prices and duration of a journey. We have used functions for sorting flights based on prices and duration and the result is displayed in the form of a grid layout.