############################################################################################

# #

# This code is written during team meet-up by team members as shown below ... #

# #

# HAFIFI BIN YAHYA - WQD170042 #

# NOR ASMIDAH BINTI MOHD ARSHAD - WQD180006 #

# MAS RAHAYU BINTI MOHAMAD - WQD180048 #

# LEE CHUN MUN - WQD180066 #

# JOJIE ANAK NANJU - WQD180029 #

# #

############################################################################################

# Batch 1

# Importing related Python modules/packages.

from lxml import html

from datetime import datetime

import requests

import pymysql

# List of companies that listed in KLSE Main Board.

companies = (

'ADVENTA','AHEALTH','HARTA','IHH','KOSSAN','KOTRA','KPJ','PHARMA','SUPERMX',

'TMCLIFE','TMCLIFE-WB','TOPGLOV','YSPSAH','ALAM','ARMADA','BARAKAH','CARIMIN','COASTAL','DAYA',

'DAYANG','DELEUM','DIALOG','HANDAL','HENGYUAN','HIBISCS','HIBISCS-WC','HUAAN','ICON','KNM',

'KNM-WB','MHB','PENERGY','PERDANA','PERISAI','PETRONM','REACH','REACH-WA','SAPNRG','SAPNRG-PA',

'SAPNRG-WA','SCOMI','SCOMI-WB','SCOMIES','SERBADK','SUMATEC','SUMATEC-WA','T7GLOBAL','THHEAVY','THHEAVY-PB',

'UZMA','VELESTO','VELESTO-WA','WASEONG','YINSON','CENSOF','CENSOF-WB','CUSCAPI','D%26O','DATAPRP',

'DIGISTA','DIGISTA-WB','DNEX','DNEX-WD','DSONIC','DSONIC-WA','EDARAN','EFORCE','EFORCE-WA','ELSOFT',

'FRONTKN','FSBM','FSBM-WA','GHLSYS','GRANFLO','GTRONIC','HTPADU','INARI','INARI-WB','ITRONIC',

'JCY','KESM','KEYASIC','MI','MMSV','MPI','MSNIAGA','MYEG','NOTION','NOTION-WC',

'OMESTI','PANPAGE','PENTA','PRESBHD','THETA','TRIVE','TRIVE-WB','TURIYA','UNISEM','VITROX',

'VSTECS','WILLOW','ACME','AMPROP','AMPROP-PA','AMPROP-PB','AMVERTON','ARK','ARK-WB','ASIAPAC',

'ASIAPAC-WB','AYER','BCB','BDB','BERTAM','BJASSET','BJASSET-WB','CHHB','CHHB-WB','CRESNDO',

'CVIEW','DAIMAN','DBHD','DBHD-WA','DPS','DPS-WB','E%26O','E%26O-WB','ECOFIRS','ECOFIRS-WC',

'ECOWLD','ECOWLD-WA','ENCORP','ENRA','EUPE','EWEIN','EWINT','EWINT-WA','FARLIM','GLOMAC',

'GMUTUAL','GOB','GOB-WA','GSB','GUOCO','HCK','HCK-WA','HOOVER','HUAYANG','IBHD',

'IBHD-WA','IBRACO','IDEAL','IDEAL-WB','IGBB','IGBB-PA','IGBB-PB','IOIPG','IVORY','IWCITY',

'JIANKUN','JIANKUN-WA','JKGLAND','KBUNAI','KBUNAI-WC','KEN','KSL','L%26G','LBICAP','LBS',

'LBS-PA','LBS-WB','LIENHOE','MAGNA','MAGNA-WB','MAHSING','MAHSING-WC','MALTON','MATRIX','MATRIX-WA',

'MBWORLD','MCT','MEDAINC','MEDAINC-WA','MEDAINC-WB','MEDAINC-WC','MENANG','MENANG-WB','MJPERAK','MKH',

'MKLAND','MPCORP','MRCB','MRCB-WB','MUH','MUIPROP','NAIM','OIB','OSK','OSK-WC',

'PARAMON','PASDEC','PASDEC-WA','PHB','PLB','PLENITU','RAPID','SAPRES','SBCCORP','SDRED',

'SEAL','SHL','SIMEPROP','SMI','SNTORIA','SNTORIA-WA','SNTORIA-WB','SPB','SPSETIA','SPSETIA-PA',

'SPSETIA-PB','SUNSURIA','SUNSURIA-WA','SYMLIFE','SYMLIFE-WB','TADMAX','TAGB','TALAMT','TAMBUN','TANCO',

'TANCO-WB','THRIVEN','THRIVEN-WB','TIGER','TIGER-WC','TITIJYA','TITIJYA-PA','TROP','TROP-WA','UEMS',

'UOADEV','WMG','Y%26G','Y%26G-WA','YNHPROP','YONGTAI','YONGTAI-PA','YONGTAI-WA','YTLLAND','EDEN',

'EDEN-WB','GASMSIA','KPS','MALAKOF','MFCB','MFCB-WA','PBA','PETGAS','RANHILL','SALCON',

'SALCON-WB','TALIWRK','TENAGA','YTL','YTLPOWR','ABMB','AEONCR','AFFIN','ALLIANZ','ALLIANZ-PA',

'AMBANK','APEX','BIMB','BIMB-WA','BURSA','CIMB','ECM','ELKDESA','HLBANK','HLCAP',

'HLFG','INSAS','INSAS-PA','INSAS-WB','JOHAN','KENANGA','KUCHAI','LPI','MAA','MANULFE',

'MAYBANK','MBSB','MNRB','MPHBCAP','P%26O','PBBANK','RCECAP','RHBBANK','TA','TAKAFUL')

# Defining AppCrawler class that use to crawl stock data from the hardcoded weblink.

class AppCrawler:

def \_\_init\_\_(self, starting\_url, depth):

self.starting\_url = starting\_url

self.depth = depth

self.apps = []

def crawl(self):

self.get\_app\_from\_link(self.starting\_url)

return

def get\_app\_from\_link(self, link):

start\_page = requests.get(link)

tree = html.fromstring(start\_page.text)

# Crawled data

# Company name

stock\_name = tree.xpath('//h1[@class="stock-profile f16"]/text()')[0]

# Company stock code in KLSE

stock\_code = tree.xpath('//li[@class="f14"]/text()')[1]

# Stock open price of the day

open\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_opentext"]/text()')[0]

# Stock high price of the day

high\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_hightext"]/text()')[0]

# Stock low price of the day

low\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lowtext"]/text()')[0]

# Stock last price of the day

last\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lastdonetext"]/text()')[0]

# Stock price change

price\_change\_up = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_up"]/text()')

price\_change\_down = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_down"]/text()')

if len(price\_change\_up) == 0 and len(price\_change\_down) != 0:

price\_change = price\_change\_down[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) != 0:

price\_change = price\_change\_up[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) == 0:

price\_change = 0

# Stock price change in percentage

price\_change\_percent = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgpercenttrext"]/text()')[0]

# Stock volume

stock\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_voltext"]/text()')[0]

# Buy volume

buy\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_buyvol"]/text()')[0]

# Sell volume

sell\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_sellvol"]/text()')[0]

# Defining database connection that use to store crawled stock data together with the date of the stock price

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

current\_Date = datetime.now()

formatted\_date = current\_Date.strftime('%Y-%m-%d %H:%M:%S')

#formatted\_date = current\_Date.strftime('%Y-%m-%d')

insert\_to\_database = (stock\_name, stock\_code[3:], open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, formatted\_date)

sql = 'INSERT INTO Stock (stock\_name, stock\_code, open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, date) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

print('Company Name: ' + str(stock\_name))

print('Stock Code: ' + str(stock\_code[3:]))

print('Open Price: ' + str(open\_price))

print('High Price: ' + str(high\_price))

print('Low Price: ' + str(low\_price))

print('Last Price: ' + str(last\_price))

print('Price Change: ' + str(price\_change))

print('Price Change in %: ' + str(price\_change\_percent))

print('Stock Volume: ' + str(stock\_volume))

print('Buy / Volume: ' + str(buy\_volume))

print('Sell / Volume: ' + str(sell\_volume))

print("Records were saved to database successfully\n\n")

return

# Crawling of stock data for the companies that set in the listing

# The crawling is based on the hardcoded weblink and AppCrawler class

for company in companies:

weblink = 'https://www.thestar.com.my/business/marketwatch/stocks/?qcounter=' + str(company)

crawler = AppCrawler(weblink, 0)

crawler.crawl()

# End of the code

# Batch 2

# Importing related Python modules/packages.

from lxml import html

from datetime import datetime

import requests

import pymysql

# List of companies that listed in KLSE Main Board.

companies = (

'TUNEPRO','AMEDIA','AMTEL','ASTRO','AXIATA','BJMEDIA','DIGI','ECOHLDS','GPACKET','GPACKET-WB',

'MAXIS','MEDIA','MEDIAC','OCK','OCK-WA','PPG','SASBADI','SJC','STAR','TIMECOM',

'TM','UTUSAN','3A','AAX','AAX-WA','ACOSTEC','AEON','AHB','AHB-WB','AIRASIA',

'AJI','AMTEK','AMWAY','APFT','APOLLO','ASB','ASIABRN','ASIAFLE','ATLAN','AVI',

'BAT','BAUTO','BIOOSMO','BJFOOD','BJLAND','BJTOTO','BONIA','BRAHIMS','CAB','CAB-WA',

'CAELY','CAELY-WA','CAMRES','CARING','CARLSBG','CCB','CCK','CCK-WA','CHEETAH','CIHLDG',

'CNI','CNOUHUA','COCOLND','CSCENIC','CSL','CWG','DBE','DBE-WB','DEGEM','DKSH',

'DLADY','DRBHCOM','EASTLND','EIG','EKA','EMICO','ENGKAH','EURO','EUROSP','F%26N',

'FCW','FIAMMA','FIHB','FIHB-PA','FIHB-PB','FPI','G3','G3-WA','GCB','GCE',

'GENM','GENTING','GETS','GREENYB','HAIO','HARISON','HBGLOB','HEIM','HLIND','HOMERIZ',

'HOMERIZ-WA','HUPSENG','HWATAI','IQGROUP','JADI','JAYCORP','JERASIA','JOHOTIN','KAMDAR','KAREX',

'KAWAN','KFM','KHEESAN','KHIND','KPOWER','KSTAR','KSTAR-WA','KTB','LANDMRK','LATITUD',

'LAYHONG','LAYHONG-WA','LEESK','LIIHEN','LONBISC','LONBISC-WA','LTKM','MAGNI','MAGNUM','MARCO',

'MAXWELL','MAXWELL-WA','MBG','MBMR','MESB','MESB-WA','MFLOUR','MFLOUR-WC','MILUX','MINDA',

'MSM','MSPORTS','MUIIND','MULPHA','MYNEWS','NESTLE','NHFATT','NICE','NICE-WB','NIHSIN',

'NTPM','OCB','OCNCASH','OFI','OLYMPIA','ORIENT','OWG','OWG-WA','PADINI','PANAMY',

'PAOS','PAOS-WA','PARAGON','PARKSON','PCCS','PCCS-WA','PELIKAN','PENSONI','PENSONI-WB','PERMAJU',

'PETDAG','PMCORP','PMHLDG','POHKONG','POHUAT','POHUAT-WB','PPB','PRG','PRG-WA','PRLEXUS',

'PRLEXUS-WA','PTRANS','PTRANS-WA','PWF','PWF-WA','PWROOT','PWROOT-WA','QL','REX','RGB',

'RHONEMA','SALUTE','SANBUMI','SAUDEE','SAUDEE-WA','SEG','SEM','SERNKOU','SERNKOU-WA','SHANG',

'SHH','SIGN','SIGN-WA','SIME','SINOTOP','SMCAP','SMCAP-WC','SNC','SOLID','SOLID-WA',

'SPRITZER','SUIWAH','SWSCAP','SWSCAP-WB','SYF','SYF-WB','TAFI','TCHONG','TECGUAN','TEKSENG',

'TEKSENG-WA','TEOSENG','TEOSENG-WA','TGL','TOMEI','TPC','TPC-WA','UMW','UPA','VERTICE',

'VERTICE-WA','WANGZNG','WARISAN','XDL','XDL-WD','XIANLNG','XINQUAN','XINQUAN-WA','XINQUAN-WB','YEELEE',

'YOCB','ZHULIAN','ADVCON','AZRB','AZRB-WA','BENALEC','BPURI','BREM','CRESBLD','DKLS',

'ECONBHD','ECONBHD-WA','EKOVEST','EKOVEST-WB','FAJAR','FAJAR-WB','GADANG','GADANG-WB','GAMUDA','GAMUDA-WE',

'GBGAQRS','GBGAQRS-WB','GKENT','HOHUP','HSL','IJM','IKHMAS','INTA','IREKA','IREKA-WB',

'JAKS','JAKS-WB','KERJAYA','KERJAYA-WB','KIMLUN','KIMLUN-WA','LEBTECH','MELATI','MERCURY','MERGE',

'MGB','MGB-WA','MITRA','MITRA-WD','MITRA-WE','MTDACPI','MUDAJYA','MUHIBAH','OCR','OCR-PA',

'OCR-WC','OCR-WD','PEB','PESONA','PESONA-WC','PRTASCO','PRTASCO-WA','PSIPTEK','PSIPTEK-WA','PTARAS')

# Defining AppCrawler class that use to crawl stock data from the hardcoded weblink.

class AppCrawler:

def \_\_init\_\_(self, starting\_url, depth):

self.starting\_url = starting\_url

self.depth = depth

self.apps = []

def crawl(self):

self.get\_app\_from\_link(self.starting\_url)

return

def get\_app\_from\_link(self, link):

start\_page = requests.get(link)

tree = html.fromstring(start\_page.text)

# Crawled data

# Company name

stock\_name = tree.xpath('//h1[@class="stock-profile f16"]/text()')[0]

# Company stock code in KLSE

stock\_code = tree.xpath('//li[@class="f14"]/text()')[1]

# Stock open price of the day

open\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_opentext"]/text()')[0]

# Stock high price of the day

high\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_hightext"]/text()')[0]

# Stock low price of the day

low\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lowtext"]/text()')[0]

# Stock last price of the day

last\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lastdonetext"]/text()')[0]

# Stock price change

price\_change\_up = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_up"]/text()')

price\_change\_down = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_down"]/text()')

if len(price\_change\_up) == 0 and len(price\_change\_down) != 0:

price\_change = price\_change\_down[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) != 0:

price\_change = price\_change\_up[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) == 0:

price\_change = 0

# Stock price change in percentage

price\_change\_percent = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgpercenttrext"]/text()')[0]

# Stock volume

stock\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_voltext"]/text()')[0]

# Buy volume

buy\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_buyvol"]/text()')[0]

# Sell volume

sell\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_sellvol"]/text()')[0]

# Defining database connection that use to store crawled stock data together with the date of the stock price

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

current\_Date = datetime.now()

formatted\_date = current\_Date.strftime('%Y-%m-%d %H:%M:%S')

#formatted\_date = current\_Date.strftime('%Y-%m-%d')

insert\_to\_database = (stock\_name, stock\_code[3:], open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, formatted\_date)

sql = 'INSERT INTO Stock (stock\_name, stock\_code, open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, date) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

print('Company Name: ' + str(stock\_name))

print('Stock Code: ' + str(stock\_code[3:]))

print('Open Price: ' + str(open\_price))

print('High Price: ' + str(high\_price))

print('Low Price: ' + str(low\_price))

print('Last Price: ' + str(last\_price))

print('Price Change: ' + str(price\_change))

print('Price Change in %: ' + str(price\_change\_percent))

print('Stock Volume: ' + str(stock\_volume))

print('Buy / Volume: ' + str(buy\_volume))

print('Sell / Volume: ' + str(sell\_volume))

print("Records were saved to database successfully\n\n")

return

# Crawling of stock data for the companies that set in the listing

# The crawling is based on the hardcoded weblink and AppCrawler class

for company in companies:

weblink = 'https://www.thestar.com.my/business/marketwatch/stocks/?qcounter=' + str(company)

crawler = AppCrawler(weblink, 0)

crawler.crawl()

# End of the code

# Batch 3

# Importing related Python modules/packages.

from lxml import html

from datetime import datetime

import requests

import pymysql

# List of companies that listed in KLSE Main Board.

companies = (

'PUNCAK','SENDAI','SUNCON','SYCAL','TRC','TSRCAP','TSRCAP-WB','VIZIONE','VIZIONE-WC','WCEHB',

'WCT','WCT-WE','ZECON','ZELAN','AHP','ALAQAR','ALSREIT','AMFIRST','ARREIT','ATRIUM',

'AXREIT','CMMT','HEKTAR','IGBREIT','KIPREIT','KLCC','MQREIT','PAVREIT','SUNREIT','TWRREIT',

'UOAREIT','YTLREIT','ABLEGRP','ADVPKG','AEM','AEM-WA','AFUJIYA','AISB','AJIYA','AJIYA-WA',

'ALCOM','ANALABS','ANCOM','ANNJOO','ANNJOO-PA','ANZO','ANZO-WA','ANZO-WB','ANZO-WC','APB',

'APM','ARANK','ASTINO','ATAIMS','ATTA','ATTA-PA','ATTA-WB','ATTA-WC','ATURMJU','ATURMJU-PA',

'AWC','AWC-WA','AYS','BIG','BINTAI','BINTAI-WA','BJCORP','BJCORP-WB','BJCORP-WC','BOILERM',

'BORNOIL','BORNOIL-WC','BORNOIL-WD','BOXPAK','BOXPAK-WA','BPPLAS','BRIGHT','BSLCORP','BSTEAD','BTM',

'BTM-WA','BTM-WB','CANONE','CAP','CBIP','CBIP-WA','CCM','CEPCO','CFM','CHINHIN',

'CHINWEL','CHOOBEE','CHUAN','CICB','CME','CME-WA','CMSB','CNASIA','COMCORP','COMFORT',

'COMPUGT','CSCSTEL','CYL','CYMAO','CYPARK','DAIBOCI','DAIBOCI-WB','DANCO','DANCO-WA','DESTINI',

'DNONCE','DNONCE-WA','DOLMITE','DOLPHIN','DOLPHIN-WA','DOMINAN','DOMINAN-WA','DUFU','EDGENTA','EFFICEN',

'EG','EG-PA','EG-WC','EITA','EKSONS','EMETALL','ENGTEX','EPMB','EVERGRN','FACBIND',

'FAVCO','FIBON','FIMACOR','FITTERS','FITTERS-WB','FLBHD','FPGROUP','GESHEN','GLOTEC','GLOTEC-WA',

'GOODWAY','GPA','GPA-WA','GPHAROS','GUH','HALEX','HAPSENG','HEVEA','HEVEA-WB','HEXZA',

'HIAPTEK','HIAPTEK-WB','HIGHTEC','HIL','HIL-WB','HOKHENG','HSSEB','HSSEB-WA','HUMEIND','HWGB',

'HWGB-WD','IMASPRO','IPMUDA','IRETEX','IRETEX-WA','JASKITA','JCBNEXT','JETSON','JMR','KARYON',

'KEINHIN','KFIMA','KGB','KGB-WA','KIALIM','KIANJOO','KIMHIN','KINSTEL','KKB','KNUSFOR',

'KOBAY','KOMARK','KOMARK-WB','KPSCB','KSENG','KSSC','KUB','KYM','LAFMSIA','LBALUM',

'LCTITAN','LEONFB','LEWEKO','LEWEKO-WB','LFECORP','LIONFIB','LIONIND','LSTEEL','LUSTER','LUSTER-WA',

'LUSTER-WB','LUXCHEM','LYSAGHT','MASTEEL','MASTER','MBL','MBL-WA','MCEHLDG','MELEWAR','MELEWAR-WB',

'MENTIGA','METROD','MIECO','MINETEC','MINETEC-WA','MINHO','MINHO-WC','MSC','MTRONIC','MUDA',

'MYCRON','MYCRON-WA','NGGB','NGGB-WA','NWP','NYLEX','OKA','ORNA','PA','PA-WB',

'PANSAR','PANSAR-WA','PANTECH','PANTECH-WA','PANTECH-WB','PCHEM','PECCA','PERSTIM','PESTECH','PGLOBE',

'PGLOBE-WA','PICORP','PIE','PJBUMI','PJBUMI-WA','PMBTECH','PMBTECH-WA','PMETAL','PMETAL-WC','PNEPCB',

'POLY','PPHB','PRESTAR','PWORTH','QUALITY','RALCO','RALCO-WB','RESINTC','RGTBHD','RGTBHD-WB',

'ROHAS','RUBEREX','SAB','SAM','SAMCHEM','SAPIND','SCABLE','SCGM','SCGM-WA','SCIB',

'SCICOM','SCIENTX','SCNWOLF','SEACERA','SEACERA-WB','SEACERA-WC','SEB','SGB','SGB-PA','SGB-WA',

'SIGGAS','SKBSHUT','SKPRES','SLP','SMISCOR','SSTEEL','STONE','STONE-WA','SUBUR','SUCCESS',

'SUNWAY','SUNWAY-WB','SUPERLN','TASEK','TASEK-PA','TAWIN','TECHBND','TEXCHEM','TGUAN','TGUAN-WA')

# Defining AppCrawler class that use to crawl stock data from the hardcoded weblink.

class AppCrawler:

def \_\_init\_\_(self, starting\_url, depth):

self.starting\_url = starting\_url

self.depth = depth

self.apps = []

def crawl(self):

self.get\_app\_from\_link(self.starting\_url)

return

def get\_app\_from\_link(self, link):

start\_page = requests.get(link)

tree = html.fromstring(start\_page.text)

# Crawled data

# Company name

stock\_name = tree.xpath('//h1[@class="stock-profile f16"]/text()')[0]

# Company stock code in KLSE

stock\_code = tree.xpath('//li[@class="f14"]/text()')[1]

# Stock open price of the day

open\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_opentext"]/text()')[0]

# Stock high price of the day

high\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_hightext"]/text()')[0]

# Stock low price of the day

low\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lowtext"]/text()')[0]

# Stock last price of the day

last\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lastdonetext"]/text()')[0]

# Stock price change

price\_change\_up = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_up"]/text()')

price\_change\_down = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_down"]/text()')

if len(price\_change\_up) == 0 and len(price\_change\_down) != 0:

price\_change = price\_change\_down[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) != 0:

price\_change = price\_change\_up[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) == 0:

price\_change = 0

# Stock price change in percentage

price\_change\_percent = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgpercenttrext"]/text()')[0]

# Stock volume

stock\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_voltext"]/text()')[0]

# Buy volume

buy\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_buyvol"]/text()')[0]

# Sell volume

sell\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_sellvol"]/text()')[0]

# Defining database connection that use to store crawled stock data together with the date of the stock price

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

current\_Date = datetime.now()

formatted\_date = current\_Date.strftime('%Y-%m-%d %H:%M:%S')

#formatted\_date = current\_Date.strftime('%Y-%m-%d')

insert\_to\_database = (stock\_name, stock\_code[3:], open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, formatted\_date)

sql = 'INSERT INTO Stock (stock\_name, stock\_code, open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, date) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

print('Company Name: ' + str(stock\_name))

print('Stock Code: ' + str(stock\_code[3:]))

print('Open Price: ' + str(open\_price))

print('High Price: ' + str(high\_price))

print('Low Price: ' + str(low\_price))

print('Last Price: ' + str(last\_price))

print('Price Change: ' + str(price\_change))

print('Price Change in %: ' + str(price\_change\_percent))

print('Stock Volume: ' + str(stock\_volume))

print('Buy / Volume: ' + str(buy\_volume))

print('Sell / Volume: ' + str(sell\_volume))

print("Records were saved to database successfully\n\n")

return

# Crawling of stock data for the companies that set in the listing

# The crawling is based on the hardcoded weblink and AppCrawler class

for company in companies:

weblink = 'https://www.thestar.com.my/business/marketwatch/stocks/?qcounter=' + str(company)

crawler = AppCrawler(weblink, 0)

crawler.crawl()

# End of the code

# Batch 4

# Importing related Python modules/packages.

from lxml import html

from datetime import datetime

import requests

import pymysql

# List of companies that listed in KLSE Main Board.

companies = (

'TIENWAH','TIMWELL','TOMYPAK','TOMYPAK-WA','TONGHER',

'TOYOINK','TOYOINK-WB','TURBO','UCHITEC','ULICORP','UMS','UMSNGB','UNIMECH','VERSATL','VS',

'WATTA','WELLCAL','WIDETEC','WONG','WOODLAN','WTHORSE','WTK','WZSATU','WZSATU-WA','YFG',

'YILAI','YKGI','YKGI-WB','YLI','AASIA','BKAWAN','BLDPLNT','BPLANT','CEPAT','CHINTEK',

'DUTALND','FAREAST','FGV','GENP','GENP-WA','GLBHD','GOPENG','HARNLEN','HSPLANT','IJMPLNT',

'INCKEN','INNO','IOICORP','JTIASA','KLK','KLUANG','KMLOONG','KMLOONG-WB','KRETAM','KWANTAS',

'MALPAC','MHC','NPC','NSOP','PINEPAC','PLS','RSAWIT','RVIEW','SBAGAN','SHCHAN',

'SIMEPLT','SOP','SWKPLNT','TAANN','TDM','THPLANT','TSH','UMCCA','UTDPLT','AIRPORT',

'BHIC','BIPORT','CHGP','CHGP-WA','CJCEN','COMPLET','EATECH','FREIGHT','GDEX','GDEX-WB',

'GUNUNG','GUNUNG-WB','HARBOUR','HARBOUR-WC','HUBLINE','HUBLINE-WA','HUBLINE-WB','HUBLINE-WC','ILB','LITRAK',

'M%26G','MAYBULK','MISC','MMCCORP','NATWIDE','PDZ','PDZ-WB','POS','PRKCORP','SEALINK',

'SEEHUP','SURIA','SYSCORP','TAS','TASCO','TNLOGIS','TOCEAN','WPRTS','XINHWA','CLIQ',

'RSENA','ICAP')

# Defining AppCrawler class that use to crawl stock data from the hardcoded weblink.

class AppCrawler:

def \_\_init\_\_(self, starting\_url, depth):

self.starting\_url = starting\_url

self.depth = depth

self.apps = []

def crawl(self):

self.get\_app\_from\_link(self.starting\_url)

return

def get\_app\_from\_link(self, link):

start\_page = requests.get(link)

tree = html.fromstring(start\_page.text)

# Crawled data

# Company name

stock\_name = tree.xpath('//h1[@class="stock-profile f16"]/text()')[0]

# Company stock code in KLSE

stock\_code = tree.xpath('//li[@class="f14"]/text()')[1]

# Stock open price of the day

open\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_opentext"]/text()')[0]

# Stock high price of the day

high\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_hightext"]/text()')[0]

# Stock low price of the day

low\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lowtext"]/text()')[0]

# Stock last price of the day

last\_price = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_lastdonetext"]/text()')[0]

# Stock price change

price\_change\_up = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_up"]/text()')

price\_change\_down = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgtext"] //span[@class="quote\_down"]/text()')

if len(price\_change\_up) == 0 and len(price\_change\_down) != 0:

price\_change = price\_change\_down[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) != 0:

price\_change = price\_change\_up[0]

elif len(price\_change\_down) == 0 and len(price\_change\_up) == 0:

price\_change = 0

# Stock price change in percentage

price\_change\_percent = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_chgpercenttrext"]/text()')[0]

# Stock volume

stock\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_voltext"]/text()')[0]

# Buy volume

buy\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_buyvol"]/text()')[0]

# Sell volume

sell\_volume = tree.xpath('//td[@id="slcontent\_0\_ileft\_0\_sellvol"]/text()')[0]

# Defining database connection that use to store crawled stock data together with the date of the stock price

connection = pymysql.connect(host='localhost', user='root', password='', db='KLSE')

cursor = connection.cursor()

current\_Date = datetime.now()

formatted\_date = current\_Date.strftime('%Y-%m-%d %H:%M:%S')

#formatted\_date = current\_Date.strftime('%Y-%m-%d')

insert\_to\_database = (stock\_name, stock\_code[3:], open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, formatted\_date)

sql = 'INSERT INTO Stock (stock\_name, stock\_code, open\_price, high\_price, low\_price, last\_price, price\_change, price\_change\_percent, stock\_volume, buy\_volume, sell\_volume, date) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'

cursor.execute(sql, insert\_to\_database)

connection.commit()

print('Company Name: ' + str(stock\_name))

print('Stock Code: ' + str(stock\_code[3:]))

print('Open Price: ' + str(open\_price))

print('High Price: ' + str(high\_price))

print('Low Price: ' + str(low\_price))

print('Last Price: ' + str(last\_price))

print('Price Change: ' + str(price\_change))

print('Price Change in %: ' + str(price\_change\_percent))

print('Stock Volume: ' + str(stock\_volume))

print('Buy / Volume: ' + str(buy\_volume))

print('Sell / Volume: ' + str(sell\_volume))

print("Records were saved to database successfully\n\n")

return

# Crawling of stock data for the companies that set in the listing

# The crawling is based on the hardcoded weblink and AppCrawler class

for company in companies:

weblink = 'https://www.thestar.com.my/business/marketwatch/stocks/?qcounter=' + str(company)

crawler = AppCrawler(weblink, 0)

crawler.crawl()

# End of the code