DATA MINING

ASSIGNMENT: MILESTONE 1

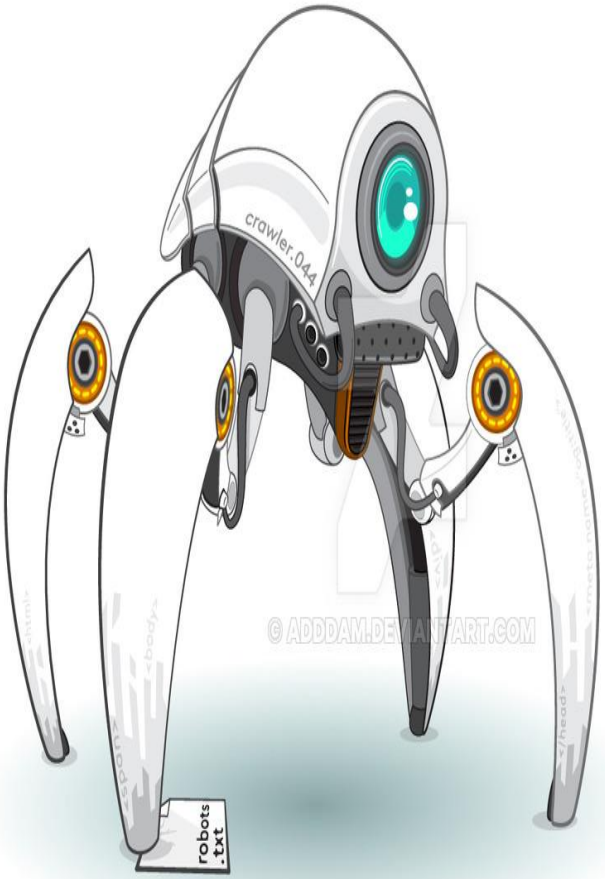
NORAISHA YUSUF (WQD180008)

NORBAIZURA MOHAMAD (WQD180043)

MARINA SHAH MUHAMMAD ZABRI TAN (WQD180011)

AMIRA AN-NUR BINTI RUSLI (WQD180016)

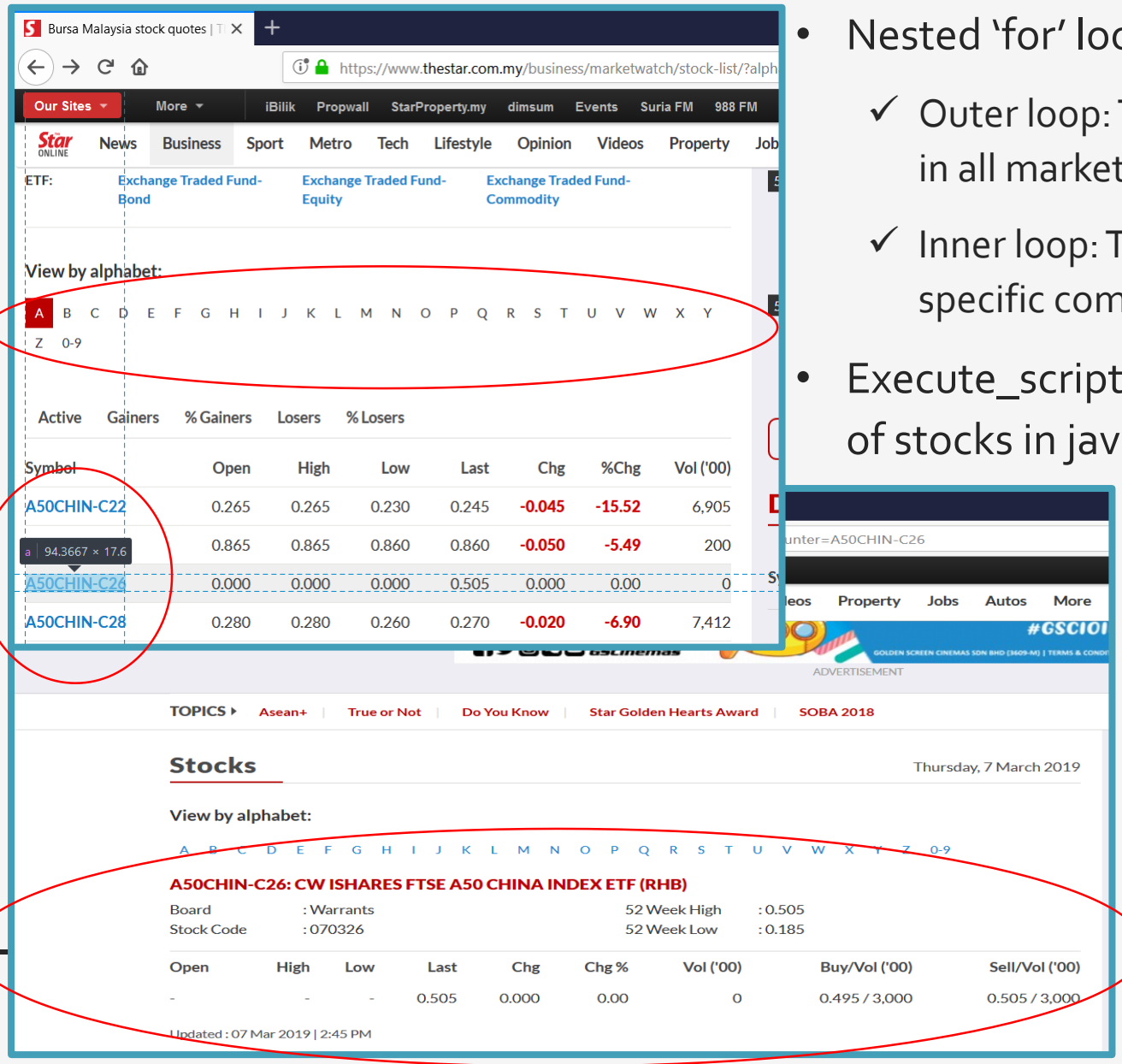
AMINAH SOFIA BINTI MAHAYUDIN (WQD180032)



Stock list

source: The Star

- Main modules: Selenium and BeautifulSoup
- Nested 'for' loop
  - ✓ Outer loop: To get urls of all companies, in all markets, in alphabetical order
  - ✓ Inner loop: To crawl stock data of each specific company
- Execute_script - to extract table of list of stocks in java script



JupyterLab

localhost:8892/lab

File Edit View Run Kernel Tabs Settings Help

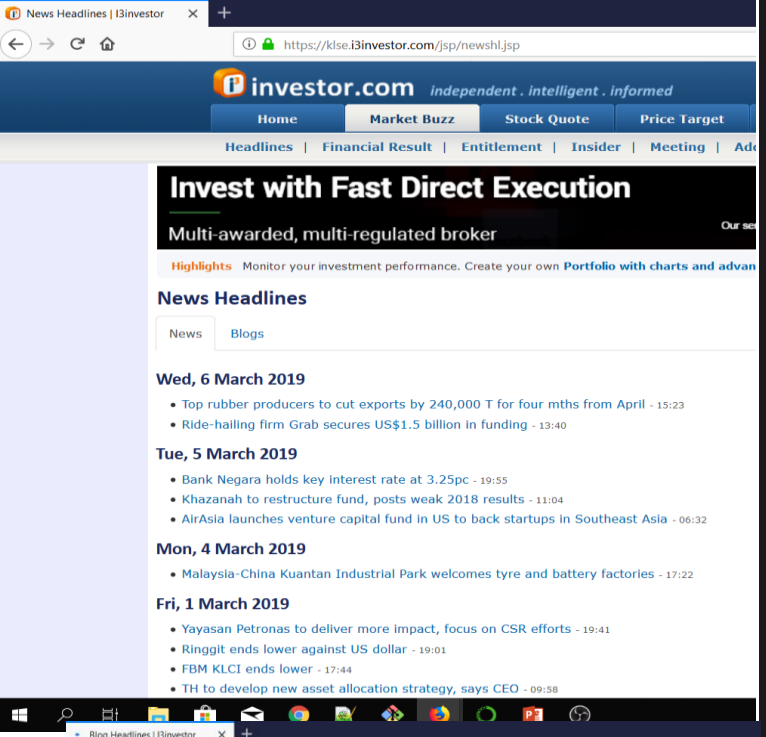
aisha_datamining_crawl.py

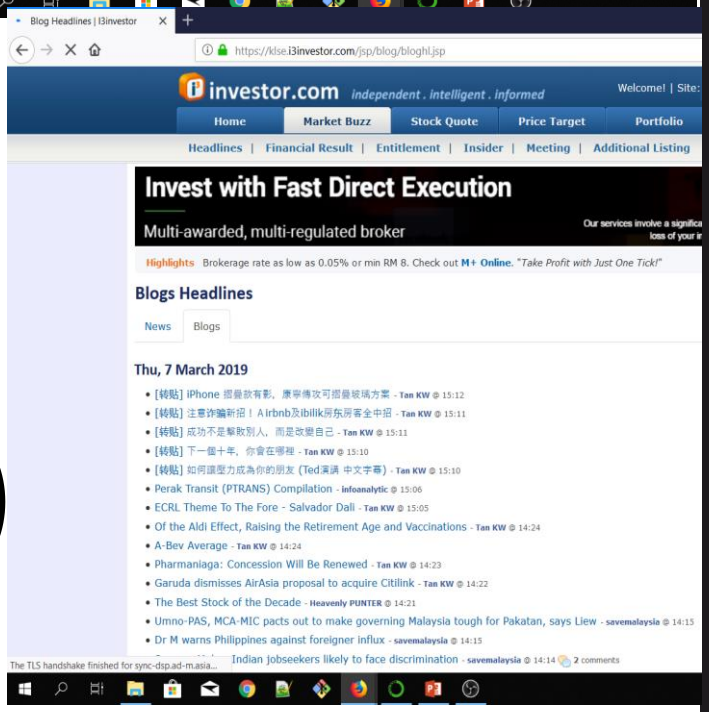
```
for i in alpha:
    print("!!!  Now char "+ i)
    browser = webdriver.Firefox(executable_path=r'C:\geckodriver.exe')
    browser.get(urlTheStar + i)
    innerHTML = browser.execute_script('return document.body.innerHTML')
    soup = BeautifulSoup(innerHTML, 'lxml')
    stock_table = soup.find('table',{'class':'market-trans'})
    links = stock_table.findAll('a')

    company = []
    for link in links:
        start_page = requests.get('https://www.thestar.com.my'+link.get('href'))
        tree = html.fromstring(start_page.text)

        url_link = 'https://www.thestar.com.my'+link.get('href')
        board = tree.xpath('//li[@class="f14"]/text()')[0]
        stock_code = tree.xpath('//li[@class="f14"]/text()')[1]
        name = tree.xpath('//h1[@class="stock-profile f16"]/text()')[0]
        w52high = tree.xpath('//li[@class="f14"]/text()')[2]
        w52low = tree.xpath('//li[@class="f14"]/text()')[3]
        updateDate = tree.xpath('//span[@id="slcontent_0_ileft_0_datetxt"]/text()')[0]
        updateTime = tree.xpath('//span[@class="time"]/text()')[0]
        open_price = tree.xpath('//td[@id="slcontent_0_ileft_0_lastdonetext"]/text()')[0]
        high_price = tree.xpath('//td[@id="slcontent_0_ileft_0_opentext"]/text()')[0]
        low_price = tree.xpath('//td[@id="slcontent_0_ileft_0_lowtext"]/text()')[0]
        last_price = tree.xpath('//td[@id="slcontent_0_ileft_0_lastdonetext"]/text()')[0]
        volume = tree.xpath('//*[@id="slcontent_0_ileft_0_voltext"]/text()')[0]
        buy_vol_hundred = tree.xpath('//*[@id="slcontent_0_ileft_0_buyvol"]/text()')[0]
        sell_vol_hundred = tree.xpath('//*[@id="slcontent_0_ileft_0_sellvol"]/text()')[0]
        date_crawl = str(datetime.datetime.now())

```

aisha_datamining_crawl.py

```
0.360
0.085
AHMAD ZAKI RESOURCES BERHAD- WA 14/24
07 Mar 2019
```

[ ]:





News & Blogs

source: Investor.com

- Selenium and BeautifulSoup
- Nested for loops
- Date, headlines and author

JupyterLab

localhost:8892/lab

File Edit View Run Kernel Tabs Settings Help

aisha_datamining_crawl.py

```
96
97  ############################
98  #   CRAWLING NEWS
99  ############################
100 print("!!! START CRAWLING NEWS")
101 urlNews='https://klse.i3investor.com/jsp/newshl.jsp'
102 browser = webdriver.Firefox(executable_path=r'C:\geckodriver.exe')
103 browser.get(urlNews)
104 innerHTML = browser.execute_script('return document.body.innerHTML')
105 soup = BeautifulSoup(innerHTML, 'lxml')
106 date = soup.select('div > h3')
107 for a in date:
108     print(" ")
109     print(a.text)
110     div = soup.find('h3', text=a.text).find_next_siblings('ul')[0]
111     title = div.find_all('a')
112     for b in title:
113         time_raw = b.find_next_siblings('span', {'class': 'graydate'})[0].text
114         time = time_raw[3:].strip()
115         dateInsert = str(datetime.datetime.now())
116         tarikh = a.text
117         tajuk = b.text
118         penulis = None
119         category = "news"
120         print(b.text)
121         print(time)
```

aisha_datamining_crawl.py

```
        07:55

        Tue, 5 February 2019
        Germania airline says filed for bankruptcy, cancels all flights
        23:43
        Merck KGaA wins GSK for immunotherapy deal worth up to $4.2 bln
        20:48
        Volkswagen courting Swedish investors to anchor Traton truck IPO
        19:50
```

[ ]:

ENG 8:36 PM 7/3/2019

JupyterLab

localhost:8892/lab

File Edit View Run Kernel Tabs Settings Help

aisha_datamining_crawl.py

```
#soup = BeautifulSoup(page.text, 'html.parser')
innerHTML = browser.execute_script('return document.body.innerHTML')
soup = BeautifulSoup(innerHTML, 'lxml')
date = soup.find("div", {"id": "maincontent730"}).find_all('h3')
print(date)
for a in date:
    print(" ")
    data_ul = soup.find('h3', text=a.text).find_next_siblings('ul')[0]
    #data_li = data_ul.select('ul > li')
    data_li = data_ul.findAll('li')
    for b in data_li:
        title = b.find('a')
        author = b.find('span', {'class': 'comuid'})
        all_text = b.find('span', {'class': 'graydate'}).text
        child_text = b.find('span', {'class': 'comuid'}).text
        parent_text = all_text.replace(child_text, '')
        print(" ")
        dateInsert = str(datetime.datetime.now())
        tarikh = a.text
        print(tarikh)
        tajuk = title.text
        print(tajuk)
        penulis = author.text
        print(penulis)
        category = "blog"
        time = parent_text[5:].strip()
```

aisha_datamining_crawl.py

```
Thu, 28 February 2019
YTL Hosp REIT - New Assets Cushioned Weaker AUD Earnings
kltrader

Thu, 28 February 2019
Evening Market Summary - 27 Feb 2019
mplus313
```

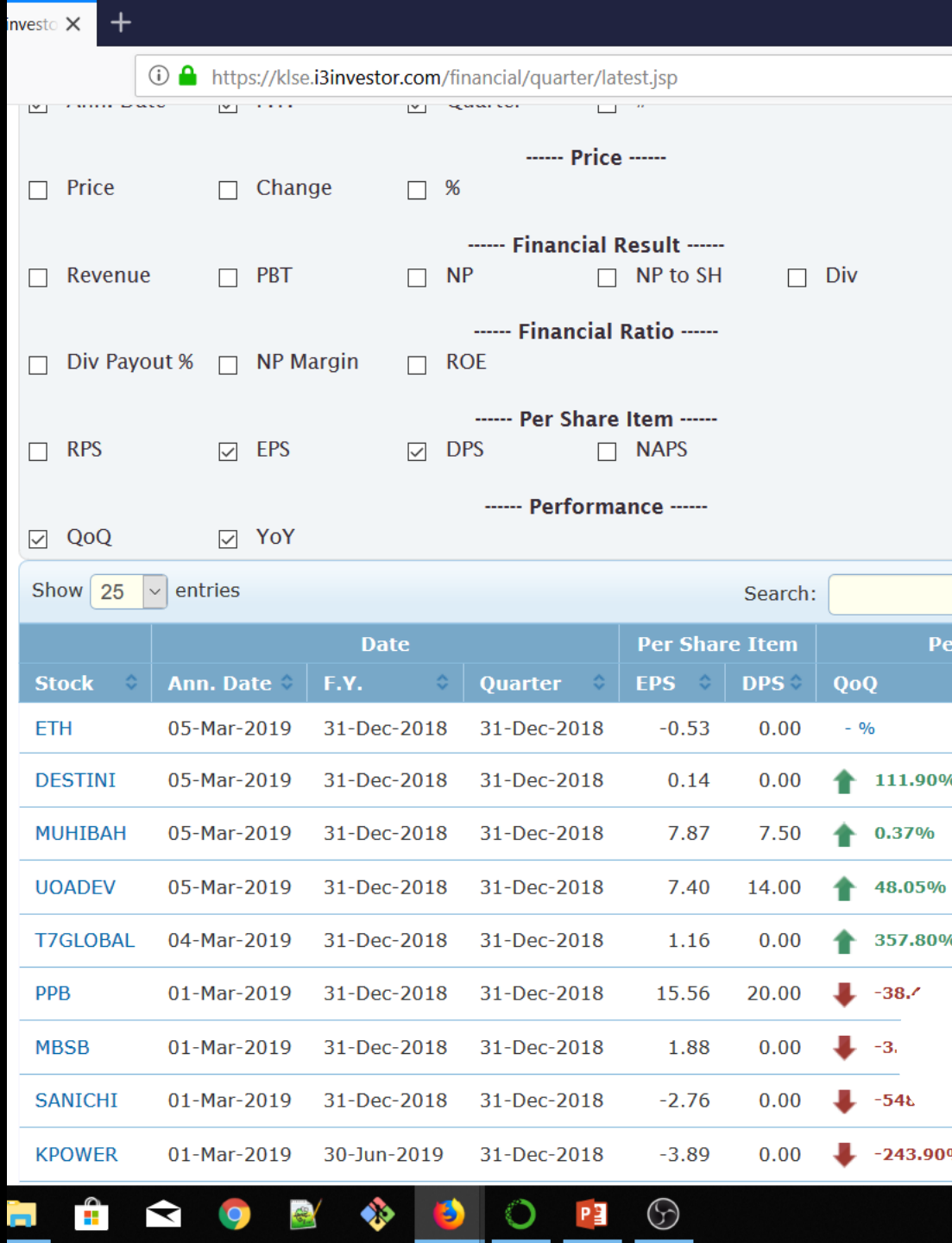
[ ]:

ENG 8:37 PM 7/3/2019

Financial Performance

source: Investor.com

- Mainly Selenium module
- For Loop within While statement function
- Element.click() – to automate clicking of check boxes and table page index



JupyterLab

localhost:8892/lab

File Edit View Run Kernel Tabs Settings Help

aisha_datamining_crawl.py

```
#   CRAWLING Financial Info
############################
print("!!! START CRAWLING FINANCIAL DATA")

browser = webdriver.Firefox(executable_path=r'C:\geckodriver.exe')
browser.implicitly_wait(40)

Financialurl = 'https://klse.i3investor.com/financial/quarter/latest.jsp'
browser.get(Financialurl)

#to expand the "modify the visible columns i.e. checkboxes"
WebElementexpanded = browser.find_element_by_xpath("//*[@id='ui-accordion-financialResultTableColumnsDiv-header-0']/span")
WebElementexpanded.click()

# to ensure all checkboxes are checked
allLinks = browser.find_elements_by_xpath('//input[@type="checkbox"]')
for link in allLinks:
    if link.is_selected():
        print('Checkbox already selected');
    else:
        link.click();
        print('Checkbox selected');
```

aisha_datamining_crawl.py

```
Checkbox selected
Checkbox selected
Checkbox selected
Checkbox already selected
Checkbox already selected
Checkbox selected
Checkbox already selected
Checkbox already selected
```

JupyterLab

localhost:8892/lab

File Edit View Run Kernel Tabs Settings Help

aisha_datamining_crawl.py

```
elm = browser.find_element_by_class_name('next')
tbl = browser.find_element_by_xpath('//*[@id="tablebody"]')
while True:
    element = WebDriverWait(browser, 100).until(lambda x: x.find_element_by_id('tablebody'))
    for row in tbl.find_elements_by_tag_name('tr'):
        data = row.find_elements_by_tag_name('td')
        file_row = []
        for datum in data:
            datum_text = datum.text
            file_row.append(datum_text)
        dateInsert = str(datetime.datetime.now())
        print(file_row)
        tpl = tuple(file_row)
        tpl = (dateInsert,)+tpl
        #val = tpl
        #mycursor = mydb.cursor()
        #sql = "INSERT INTO test_financial (date_crawl, no, stock, annDate, fy, quarter, h, price, ch, percentage, revenue, pbt, np, NptoSh, dividend, networth, divpayout, npmargin, roe, rps, eps, dps, naps, QoY, YoY) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
        #mycursor.execute(sql,val)
        #mydb.commit()
    elm = browser.find_element_by_class_name('next')
    if 'ui-state-disabled' in elm.get_attribute('class'):
        break;
    elm.click()

```

aisha_datamining_crawl.py

```
['2', 'DESTINI', '05-Mar-2019', '31-Dec-2018', '31-Dec-2018', '4', '0.235  ', '+0.015', '6.82%', '102,770', '9,816', '2,156', '1,563', '-', '509,225', '- %', '2.10%', '0.31%', '8.90', '0.14', '0.00', '0.4408', '  111.90%', '  -72.34%']
['3', 'MUHIBAH', '05-Mar-2019', '31-Dec-2018', '31-Dec-2018', '4', '2.97  ', '0.00', '0.00%', '451,871', '70,985', '64,831', '37,882', '36,118', '1,122,076', '95.34%', '14.35%', '3.38%', '93.83', '7.87', '7.50', '2.3300', '  0.37%', '  3.76%']
['4', 'UOADEV', '05-Mar-2019', '31-Dec-2018', '31-Dec-2018', '4', '2.18  ', '-0.04', '1.80%', '486,352', '175,466', '150,716', '136,447', '258,123', '4,683,094', '189.17%', '30.99%', '2.91%', '26.38', '7.40', '14.00', '2.5400', '  48.05%', '  -28.86%']
['5', 'T7GLOBAL', '04-Mar-2019', '31-Dec-2018', '31-Dec-2018', '4', '0.535  ', '-0.015', '2.73%', '44,874', '2,958', '4,783', '3,374', '-', '140,272', '- %', '10.66%', '2.41%', '10.88', '1.16', '0.00', '0.3400', '  357.80%', '  1,932.53%']
```

[ ]:

ENG 8:40 PM 7/3/2019