Stock Assistant

Marcin Krawiec Konrad Syrnik

Program:

W ramach demonstracji naszego projektu zostaną zrealizowane następujące punkty:

- Analiza danych giełdowych - zastosowanie yfinance
- Tworzenie GUI - zastosowanie PyQt5
- Daza danych - zastosowanie MySQL
- Prezentacja aplikacji *Stock Assistant*
- Test i zadania praktyczne

yfinance

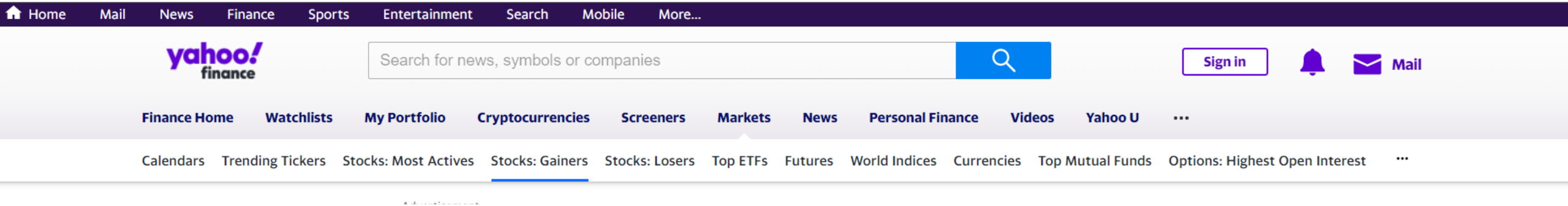
=

+

yahoo!
finance

Yahoo Finance

- Oferuje doskonały zakres danych rynkowych dotyczących **akcji**,**walut** i **kryptowalut**.
- Oferuje również **wiadomości rynkowe**, **raporty** i **analizy**, a także różne opcje i fundamentalne dane.



yfinance

- Pakiet Pythona,
- Umożliwia pobieranie historycznych danych rynkowych z Yahoo Finance API,
- Yfinance jest całkowicie otwarte i bezpłatne,
- Wysoka ziarnistość danych (dane 1min/2min/5min),
- Zwraca dane bezpośrednio w pandas dataframes/series.

yfinance - układ biblioteki

Yfinance zawiera tylko trzy moduły:

- yf.**Ticker(s)**
- yf.**download**
- yf.**pandas_datareader**

Moduł **download** służy do szybkiego pobierania danych historycznych wielu tickerów jednocześnie.

A **pandas_datareader** jest dla wstecznej kompatybilności ze starszym kodem.

yfinance - pobieranie danych

Pobierzmy najnowsze dane dzienne z 30 dni dla Google.

```
import yfinance as yf


goog = yf.Ticker('goog')
data = goog.history()
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2022-03-30	2857.399902	2869.610107	2843.360107	2852.889893	1052300	0	0
2022-03-31	2848.969971	2852.889893	2792.379883	2792.989990	1475800	0	0
2022-04-01	2800.199951	2819.000000	2775.939941	2814.000000	1173600	0	0
2022-04-04	2816.489990	2880.875000	2816.489990	2872.850098	953800	0	0
2022-04-05	2867.989990	2871.800049	2818.870117	2821.260010	962800	0	0
2022-04-06	2783.229980	2796.969971	2728.362061	2743.520020	1178700	0	0
2022-04-07	2732.360107	2754.030029	2697.145020	2729.300049	972400	0	0
2022-04-08	2725.000000	2725.000000	2675.050049	2680.209961	821000	0	0
2022-04-11	2658.000000	2658.783936	2592.350098	2595.929932	1209400	0	0

yfinance - pobieranie danych

Teraz pobierzmy dane minutowe z ostatniego tygodnia; tylko tym razem użyjemy daty rozpoczęcia i zakończenia zamiast okresu.

- **Okres musi przypadać w ciągu ostatnich 30 dni**
- **Na każde żądanie dozwolonych jest tylko siedem dni z dokładnością do 1 minuty**

```
goog = yf.Ticker('goog')
data = goog.history(interval='1m', start='2022-04-23', end='2022-04-30')
```

Datetime	Open	High	Low	Close	Volume	Dividends	Stock Splits
2022-04-25 09:30:00-04:00	2388.590088	2388.590088	2375.385010	2381.459961	66760	0	0
2022-04-25 09:31:00-04:00	2382.530029	2392.280029	2380.995117	2387.524170	9926	0	0
2022-04-25 09:32:00-04:00	2387.080078	2397.449951	2385.739990	2392.270020	10321	0	0
2022-04-25 09:33:00-04:00	2390.965088	2395.370117	2389.000000	2392.030029	15981	0	0
2022-04-25 09:34:00-04:00	2394.205078	2396.469971	2392.159912	2396.469971	5293	0	0
...	...	...	...	...	...	...	...

yfinance - pobieranie wielu tickerów

Pobierzmy najnowsze dane miesięczne dla Google i Facebooka (META).

```
data = yf.download(['GOOG','META'], period='1mo')
```

Poprawne wartości parametru **period**:
„1d”, „5d”, „1mo”, „3mo”, „6mo”, „1y”, „2y”, „ 5y”, „10y”, „ytd”, „max”.

	Adj Close		Close		High		Low		Open		Volume	
	GOOG	META	GOOG	META	GOOG	META	GOOG	META	GOOG	META	GOOG	META
Date												
2022-03-30	2852.889893	12.26	2852.889893	12.26	2869.610107	12.6100	2843.360107	12.1900	2857.399902	12.48	1052300	792705.0
2022-03-31	2792.989990	11.96	2792.989990	11.96	2852.889893	12.2700	2792.379883	11.9400	2848.969971	12.25	1475800	326389.0
2022-04-01	2814.000000	11.95	2814.000000	11.95	2819.000000	12.0600	2775.939941	11.8100	2800.199951	12.03	1173600	290255.0
2022-04-04	2872.850098	12.44	2872.850098	12.44	2880.875000	12.4700	2816.489990	12.0500	2816.489990	12.07	953800	382919.0
2022-04-05	2821.260010	12.05	2821.260010	12.05	2871.800049	12.4200	2818.870117	12.0050	2867.989990	12.40	962800	521028.0
2022-04-06	2743.520020	11.57	2743.520020	11.57	2796.969971	11.8200	2728.362061	11.4200	2783.229980	11.80	1178700	515777.0
2022-04-07	2729.300049	11.47	2729.300049	11.47	2754.030029	11.7000	2697.145020	11.2100	2732.360107	11.56	972400	356492.0
2022-04-08	2680.209961	11.27	2680.209961	11.27	2725.000000	11.4600	2675.050049	11.2300	2725.000000	11.46	821000	536573.0
2022-04-11	2595.929932	11.04	2595.929932	11.04	2658.783936	11.1950	2592.350098	10.9200	2658.000000	11.04	1209400	597443.0
2022-04-12	2567.489990	10.92	2567.489990	10.92	2648.469971	11.3700	2551.520020	10.8800	2648.469971	11.18	1150200	455259.0
2022-04-13	2605.719971	11.21	2605.719971	11.21	2613.114990	11.2500	2568.771973	10.8900	2572.530029	11.01	977100	362959.0
2022-04-14	2545.060059	10.85	2545.060059	10.85	2614.205078	11.2300	2542.229980	10.8400	2612.989990	11.20	1171400	401526.0
2022-04-18	2559.219971	10.76	2559.219971	10.76	2574.239990	10.8469	2531.569092	10.6000	2548.199951	10.73	745900	366416.0
2022-04-19	2610.620117	11.02	2610.620117	11.02	2618.074951	11.0650	2549.030029	10.6700	2561.540039	10.77	1136000	277844.0

Analiza danych - przydatne funkcje

- Funkcja pct_change() oblicza procentową zmianę między bieżącym elementem a poprzednim.
- cumprod() oblicza skumulowane zwroty
- corr() oblicza wartość korelacji między 2 elementami
  Wartości mieszczą się w zakresie od -1 do 1.

	AAPL	MSFT	TSLA
AAPL	1.000000	0.715340	0.519194
MSFT	0.715340	1.000000	0.466133
TSLA	0.519194	0.466133	1.000000

Analiza danych - weights

```
stocks = ['GOOG', 'TSLA', 'MSFT', 'LMT']
weights = [0.7035, 0.185, 0.0858, 0.0257]

data = yf.download(stocks, start='2021-01-01')
x = data['Close'].pct_change()

ret = (x * weights).sum(axis=1)
cumulative = (ret + 1).cumprod()
```

Analiza danych - volatility

```
print(np.std(ret))
annual_std = np.std(ret) * np.sqrt(252)
```

- Wykorzystywana do pomiaru ryzyka, niestabilności cen akcji.
- Zmienność jest obliczana przy użyciu odchylenia standardowego zwrotu portfela.
- Możemy również obliczyć roczną zmienność, wyciągając pierwiastek kwadratowy z liczby dni handlowych w roku (252) i mnożąc go przez dzienną zmienność.

Analiza danych - sharpe ratio

```
sharpe = (np.mean(ret)/np.std(ret)) * np.sqrt(252)
print('Sharpe: %f' % sharpe)
```

- Wskaźnik Sharpe'a jest miarą zwrotu z portfela skorygowanego o ryzyko.
- Portfel o wyższym współczynniku Sharpe'a jest uważany za lepszy.
- Aby obliczyć współczynnik Sharpe'a, musimy wziąć średni zwrot i podzielić go przez zmienność.
- Wskaźniki Sharpe'a większe niż 1 są uważane za atrakcyjne

PyQt5 =  + 

Qt - co to to takiego?

Qt to zestaw wieloplatformowych bibliotek C++, które implementują interfejsy API wysokiego poziomu w celu uzyskania dostępu do wielu aspektów nowoczesnych systemów stacjonarnych i mobilnych.

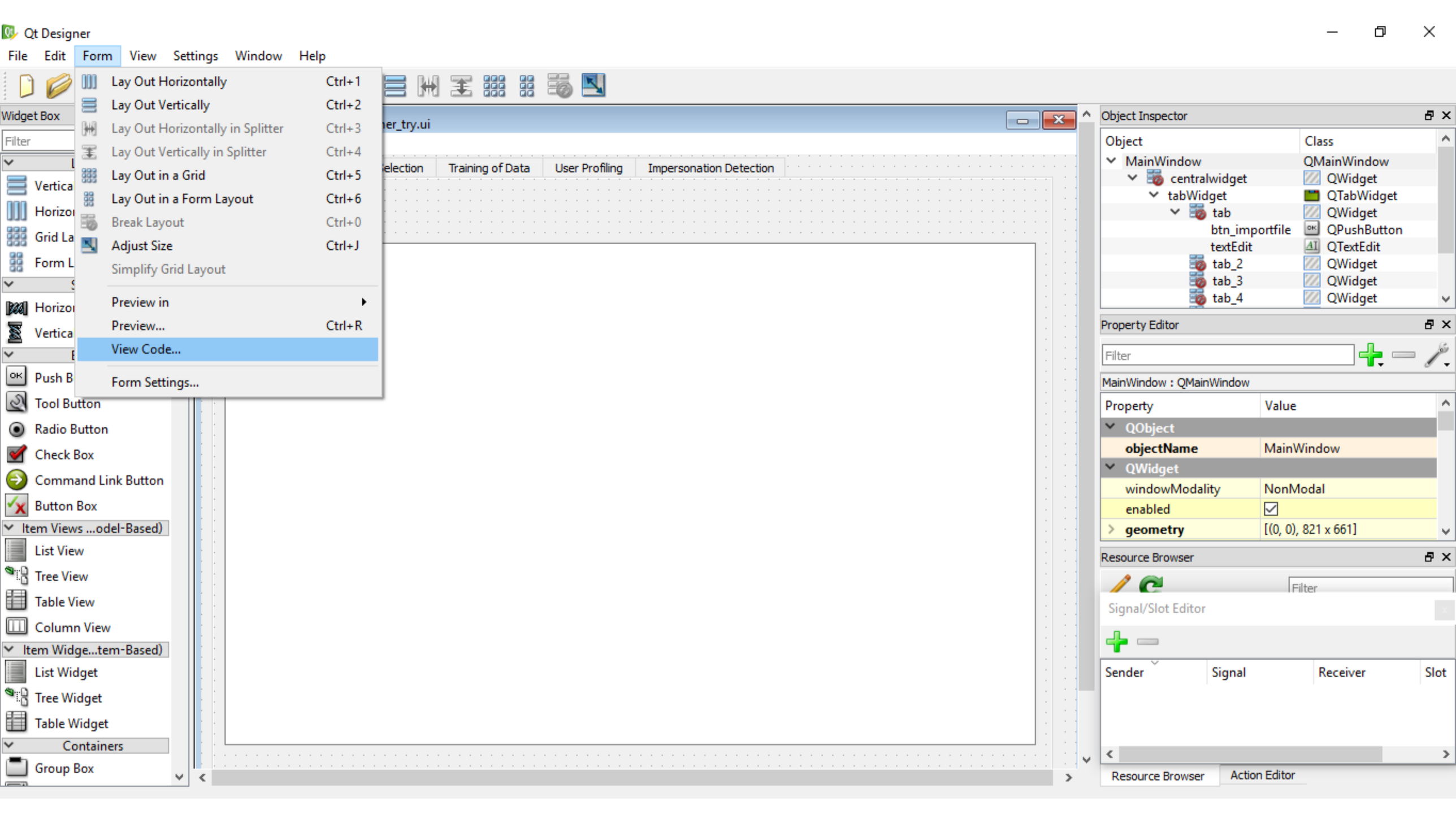
Obejmują one usługi:

- lokalizacji i pozycjonowania,
- multimedia,
- łączność NFC i Bluetooth,
- przeglądarkę internetową opartą na Chromium,
- tradycyjne tworzenie interfejsu użytkownika

PyQt5

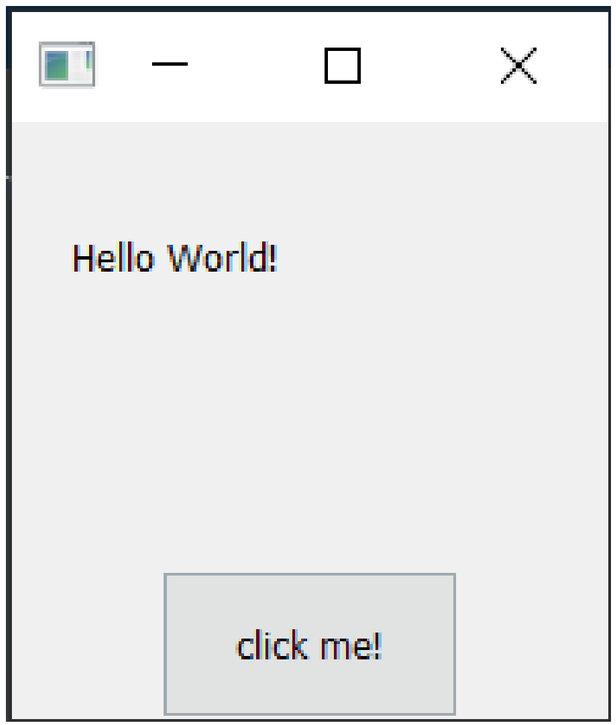
- PyQt5 to zestaw rozwiązań Pythona dla frameworka aplikacji Qt v5.
- Biblioteka Qt jest jedną z najpotężniejszych bibliotek GUI.
- Obecnie rozwijany przez *Riverbank Computing*.
- Zaimplementowany jako zestaw modułów Pythona.
- Posiada ponad 620 klas oraz 6000 funkcji i metod.
- Jest to wieloplatformowy zestaw narzędzi (Unix, Windows i Mac OS).
- Do najczęściej wykorzystywnaych modułów należą:

  Qt, **QtCore**, **QtWidgets**, **QtGui**, QtNetwork, QtMultimedia, QtSql


Qt Designer
File
Edit
Form
View
Settings
Window
Help
Lay Out Horizontally Ctrl+1
Lay Out Vertically Ctrl+2
Lay Out Horizontally in Splitter Ctrl+3
Lay Out Vertically in Splitter Ctrl+4
Lay Out in a Grid Ctrl+5
Lay Out in a Form Layout Ctrl+6
Break Layout Ctrl+0
Adjust Size Ctrl+J
Simplify Grid Layout
Preview in
Preview... Ctrl+R
View Code...
Form Settings...
Widget Box
Filter
Push Button
Tool Button
Radio Button
Check Box
Command Link Button
Button Box
Item Views ...odel-Based)
List View
Tree View
Table View
Column View
Item Widge...tem-Based)
List Widget
Tree Widget
Table Widget
Containers
Group Box
Training of Data
User Profiling
Impersonation Detection
Object Inspector
Object
Class
MainWindow
QMainWindow
centralwidget
QWidget
tabWidget
QTabWidget
tab
QWidget
btn_importfile
QPushButton
textEdit
QTextEdit
tab_2
QWidget
tab_3
QWidget
tab_4
QWidget
Property Editor
Filter
MainWindow : QMainWindow
Property
Value
QObject
objectName
MainWindow
QWidget
windowModality
NonModal
enabled
geometry
[(0, 0), 821 x 661]
Resource Browser
Filter
Signal/Slot Editor
Sender
Signal
Receiver
Slot
Resource Browser
Action Editor

PyQt5

hello_world



hello_world.py

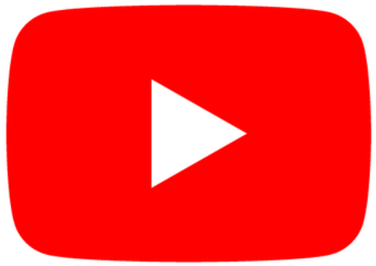
```python
import sys
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *


class window(QWidget):
    def __init__(self):
        super().__init__()
        self.setGeometry(200, 200, 200, 200)
        self.setWindowTitle("PyQt5")

        self.button = QPushButton(self)
        self.button.setText("click me!")
        self.button.setGeometry(50, 150, 100, 50)

        self.label = QLabel(self)
        self.label.setGeometry(20, 20, 160, 50)

        self.button.clicked.connect(lambda: self.label.setText("Hello World!"))


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = window()
    ex.show()
    sys.exit(app.exec_())

```

MySQL

MySQL

- System baz danych używany do tworzenia aplikacji internetowych.
- Używany zarówno w małych, jak i dużych aplikacjach.
- System zarządzania relacyjnymi bazami danych (RDBMS).
- Szybki, niezawodny, elastyczny i łatwy w użyciu.
- Obsługuje standardowy język SQL (Structured Query Language).
- Jest darmowy.
- Został opracowany przez Michaela Wideniusa i Davida Axmarka w 1994 roku.
- Jest obecnie rozwijany, dystrybuowany i wspierany przez Oracle Corporation.
- Napisany w C, C++ i oparty na wielowątkowości

MySQL - kto używa?



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- sakila
- stock_assistant
  - Tables
    - hehe
    - jojo
    - tere
    - tescior1
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
    - test1
  - Views
  - Stored Procedures
  - Functions

Administration Schemas

Information

No object selected

Object Info Session

Query 1 stock_assistant.tescior1

Limit to 1000 rows

```
1  drop table test
2
3
4
5
```

SQLAdditions

Jump to

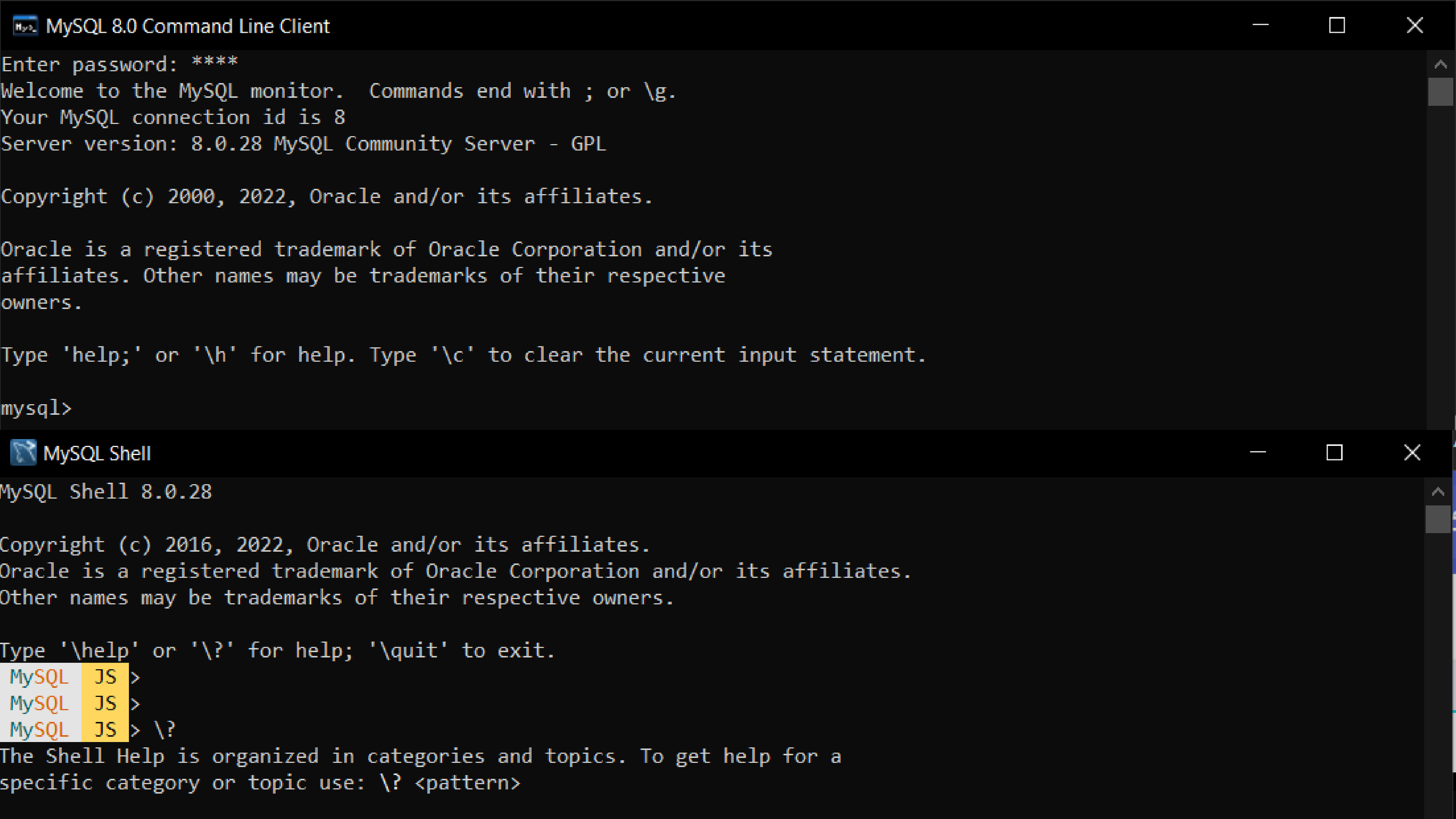
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch


MySQL 8.0 Command Line Client
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.28 MySQL Community Server - GPL
Copyright (c) 2000, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
MySQL Shell
MySQL Shell 8.0.28
Copyright (c) 2016, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.
Type '\help' or '\?' for help; '\quit' to exit.
MySQL JS >
MySQL JS >
MySQL JS > \?
The Shell Help is organized in categories and topics. To get help for a
specific category or topic use: \? <pattern>

MySQL - tworzenie bazy danych

```
mysql> create database test;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| stock_assistant    |
| sys                |
| test               |
| world              |
+--------------------+
8 rows in set (0.00 sec)
```

```
mysql> drop database test;
Query OK, 0 rows affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| stock_assistant    |
| sys                |
| world              |
+--------------------+
7 rows in set (0.00 sec)
```

MySQL - połączenie z bazą

```python
import mysql.connector
import logging
import sys


class DatabaseConnector:

    # establishing connection to database
    try:
        database = mysql.connector.connect(host='127.0.0.1', user='root', password='root', database='stock_assistant',
                                           auth_plugin='mysql_native_password')
    except mysql.connector.Error as e:
        logging.critical('Connection to database has not been established: ' + str(e))
        sys.exit()
```

MySQL - CREATE

```
CREATE TABLE IF NOT EXISTS nazwa_tabeli (nazwa_kolumny_1 typ inne_parametry,
                                         nazwa_kolumny_2 typ inne_parametry, ...)
```

```python
@staticmethod
def create_table(name):

    query = 'CREATE TABLE IF NOT EXISTS %s (id INT AUTO_INCREMENT PRIMARY KEY, stock VARCHAR(250) NOT NULL,' \
            ' amount FLOAT NOT NULL, value FLOAT NOT NULL, date VARCHAR(250) NOT NULL )' % name

    try:
        cursor = DatabaseConnector.database.cursor()
        cursor.execute(query)
    except (mysql.connector.Error, AttributeError) as e:
        logging.error('Query has not been executed: ' + str(e))
```

MySQL - INSERT

```
INSERT INTO nazwa_tabeli (kolumna_1, kolumna_1, ...) VALUES (wartość_1, wartość_2, ...)
```

```python
@staticmethod
def insert_into(name, stock, amount, value, date):

    query = 'INSERT INTO %s (stock, amount, value, date) VALUES (%s, %s, %s, %s)' % (name, stock, amount, value, date)


    try:
        cursor = DatabaseConnector.database.cursor()
        cursor.execute(query)
        DatabaseConnector.database.commit()
    except (mysql.connector.Error, AttributeError) as e:
        logging.error('Query has not been executed: ' + str(e))
```

MySQL - SELECT

```
SELECT wartość_1, wartość_2, ... FROM nazwa_tabeli
SELECT * FROM nazwa_tabeli
```

```python
@staticmethod
def select_from(name):

    query = 'SELECT stock, amount, value, date FROM %s' % name

    try:
        cursor = DatabaseConnector.database.cursor()
        cursor.execute(query)
        data = []
        for element in cursor:
            # tuple unpacking
            (stock, amount, value, date) = element
            line = [stock, amount, value, date]
            data.append(line)
        return data
    except (mysql.connector.Error, AttributeError) as e:
        logging.error('Query has not been executed: ' + str(e))
```

MySQL - DELETE

```
DELETE FROM nazwa_tabeli WHERE wartość_kolumny = x
```

```python
@staticmethod
def delete_from(name, stock, date):

    query = 'DELETE FROM %s WHERE stock=%s and date=%s' % (name, stock, date)

    try:
        cursor = DatabaseConnector.database.cursor()
        cursor.execute(query)
        DatabaseConnector.database.commit()
    except (mysql.connector.Error, AttributeError) as e:
        logging.error('Query has not been executed: ' + str(e))
```

MySQL - DROP

```
DROP TABLE nazwa_tabeli
```

```python
@staticmethod
def drop_table(name):

    query = 'DROP TABLE %s' % name

    try:
        cursor = DatabaseConnector.database.cursor()
        cursor.execute(query)
        DatabaseConnector.database.commit()
    except (mysql.connector.Error, AttributeError) as e:
        logging.error('Query has not been executed: ' + str(e))
```

MySQL - SHOW

SHOW TABLES

```
@staticmethod
def show_tables():

    query = 'SHOW TABLES'

    try:
        cursor = DatabaseConnector.database.cursor()
        cursor.execute(query)
        names = []
        for name in cursor:
            # tuple unpacking
            (n,) = name
            names.append(n)
        return names
    except (mysql.connector.Error, AttributeError) as e:
        logging.error('Query has not been executed: ' + str(e))
```

Demonstracja aplikacji

Stock Assistant

Test

Instalacja pakietów

```
pip install PyQt5

pip install yfinance
```

Zadania praktyczne

Pomocne linki

yfinance

https://pypi.org/project/yfinance/
https://analyzingalpha.com/yfinance-python
https://algotrading101.com/learn/yfinance-guide/

PyQt5

https://www.tutorialspoint.com/pyqt5/pyqt5_hello_world.html
https://doc.qt.io/qt-5/classes.html
https://www.riverbankcomputing.com/static/Docs/PyQt5/

Dziękujemy

za uwagę