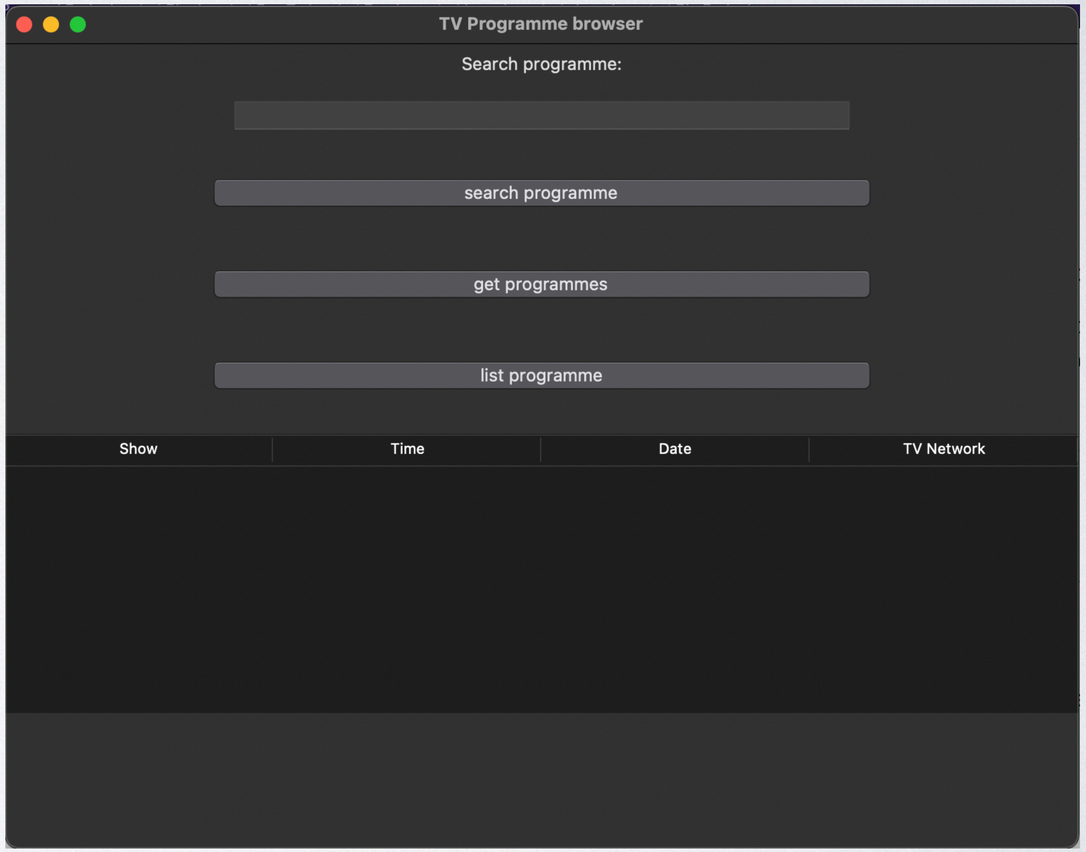
TV PROGRAMME SCRAPER

Designed and coded by Maxime Kiriakov

PYTHON GUI APPLICATION

- Support for BTV and NOVA tv networks
- Actions include:
  - Searching for a show by name
  - Scraping for the programme one week ahead
  - Listing all shows in the programme



LIBRARIES USED

- Sqlite3 - database of choice
- Tkinter - GUI support
- Scrapy - web-crawler/scraper
- BeautifulSoup - HTML parser

CODE STRUCTURE

scrapy.Spider —> the web-crawler:

- cookies and headers needed to bypass “accept cookies” on some websites (BTV in our case)
- start_requests - entry point of the crawler
- get_urls_from_week_nova - specific callback for the given website
- get_urls_from_week_btv - specific callback for the given website

```
# Crawler
class ProgramSpider(scrapy.Spider):
    name = "programs"

    # cookies and headers added to be able to access btv website
    # or else it blocks the crawler
    # NOTE: change luckynumber from time to time or you will
    # be blocked from btv
    cookies = {
        'CookieOptIn': 'true',
        'luckynumber': '1326712231',
        'MpSession': '9ff31f05-36fd-4570-9cdc-e1800bf682fe',
    }

    headers = {
        'Pragma': 'no-cache',
        'DNT': '1',
        'Accept-Encoding': 'gzip, deflate, sdch, br',
        'Accept-Language': 'en-US,en;q=0.8',
        'Upgrade-Insecure-Requests': '1',
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537
        'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',
        'Cache-Control': 'no-cache',
        'Referer': 'https://www.marktplaats.nl/cookiewall/?target=https%3A%2F%2Fwww.marktplaats.nl%2F',
        'Connection': 'keep-alive',
    }
```

```
def start_requests(self):
    urls = [
        'https://nova.bg/schedule',
        'https://www.btv.bg/programata/'
    ]

    # delete and create table anew to prevent duplicate entries
    # NOTE: could use other methods to avoid duplication
    # but are more complex (outside of this course)
    cur.execute('DROP TABLE IF EXISTS programs')
    cur.execute('CREATE TABLE IF NOT EXISTS\
        programs (name text, time text, date text, tv_network text)')

    for url in urls:
        if url.find("nova") != -1:
            yield scrapy.Request(
                url=url,
                callback=self.get_urls_from_week_nova
            )
        elif url.find("btv") != -1:
            yield scrapy.Request(
                url=url,
                callback=self.get_urls_from_week_btv,
                headers=self.headers,
                cookies=self.cookies
            )
```

- Following the scraper implementation we have:
  - Connection to DB
  - Initialisation of the Crawler
  - Tkinter GUI window setup
  - Definition of the functions for each button

```
# DB setup
con = sqlite3.connect('db.sqlite')
cur = con.cursor()

# Crawler setup
process = CrawlerProcess()
process.crawl(ProgramSpider)
```

```
# Tkinter window setup
root = tk.Tk()
root.title('TV Programme browser')
root.geometry('800x600+50+50')
```

```
# Button functions
def search_programme():
    tree.delete(*tree.get_children())

    for row in cur.execute(f'SELECT * FROM programs WHERE\
            name LIKE "%{command.get()}%"'):
        tree.insert('', tk.END, values=row)


def get_programmes():
    process.start()
    con.commit()


def list_programme():
    tree.delete(*tree.get_children())

    for row in cur.execute('SELECT * FROM programs'):
        tree.insert('', tk.END, values=row)
```

TKINTER WINDOW ELEMENTS

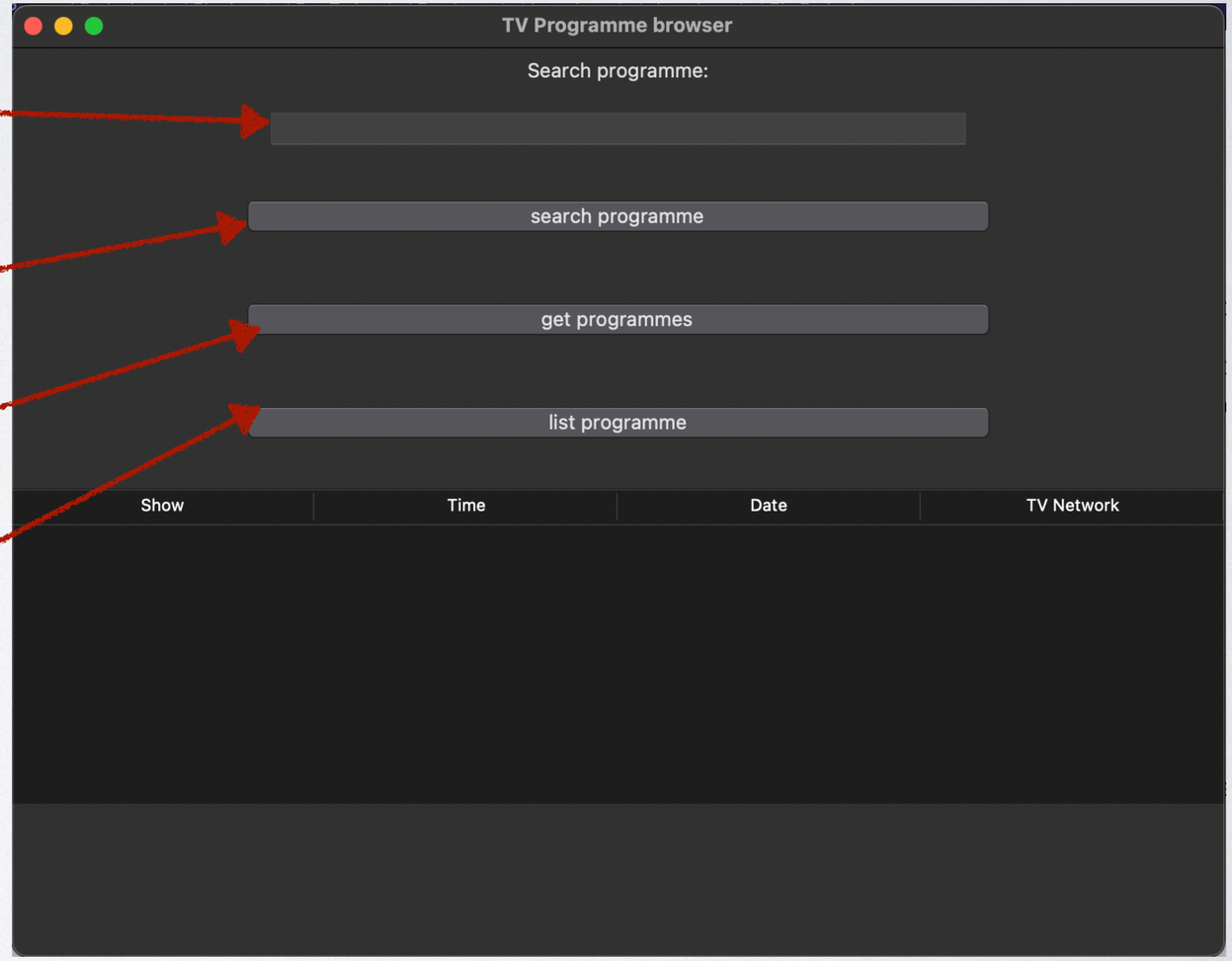
```
command_label = ttk.Label(root, text='Search programme:')
command_label.pack(ipady=5)

command = tk.StringVar()
command_entry = ttk.Entry(root, textvariable=command, width=50)
command_entry.pack(pady=10)
```

```
search_button = ttk.Button(
    root,
    text="search programme",
    command=search_programme,
    width=50
)
search_button.pack(pady=20)

get_button = ttk.Button(
    root,
    text="get programmes",
    command=get_programmes,
    width=50
)
get_button.pack(pady=20)

list_button = ttk.Button(
    root,
    text="list programme",
    command=list_programme,
    width=50
)
list_button.pack(pady=20)
```



TKINTER WINDOW ELEMENTS

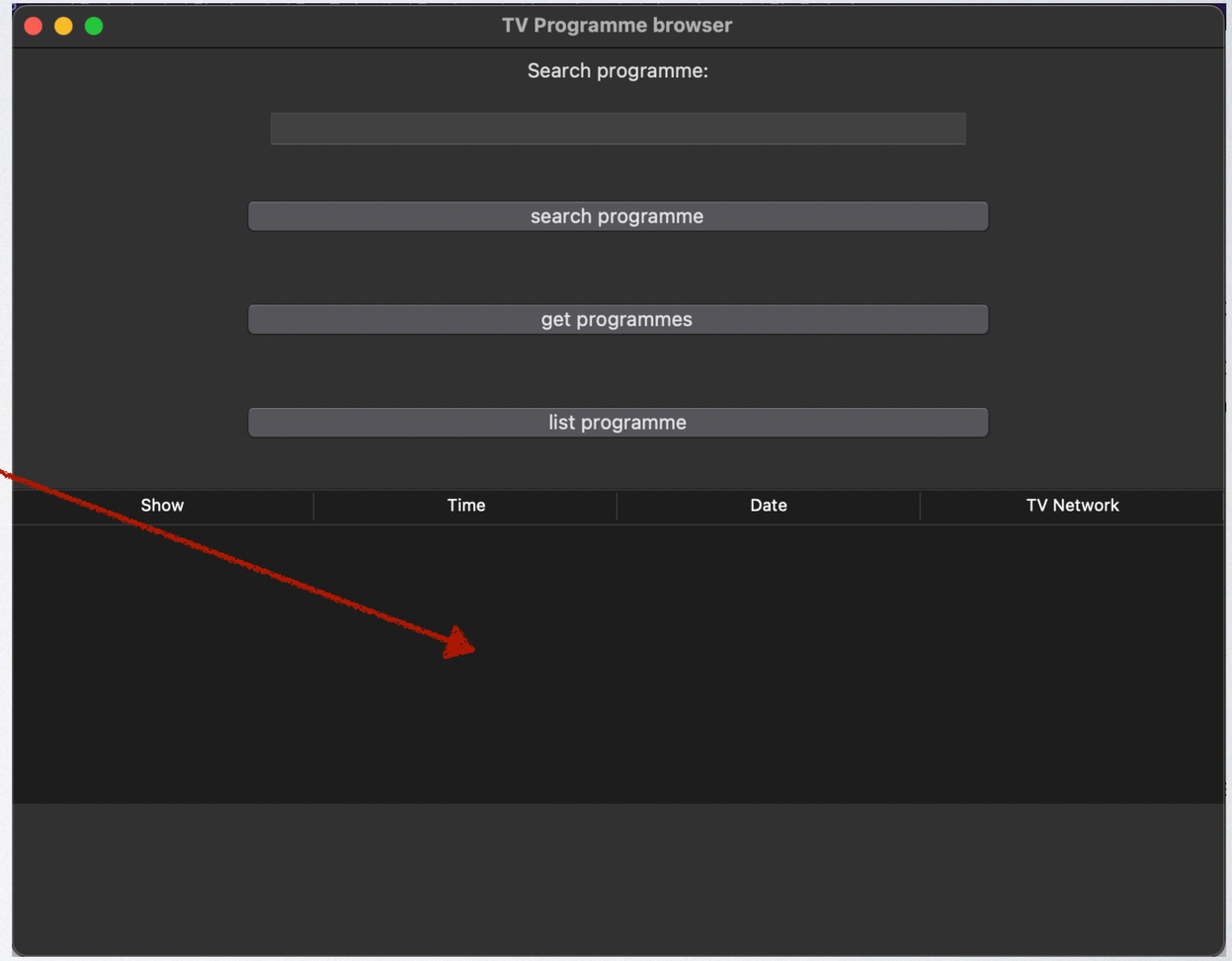
```
columns = ('show', 'time', 'date', 'tv_network')

tree = ttk.Treeview(root, columns=columns, show='headings')

tree.heading('show', text='Show')
tree.heading('time', text='Time')
tree.heading('date', text='Date')
tree.heading('tv_network', text='TV Network')

tree.pack(pady=10)

scrollbar = ttk.Scrollbar(root, orient=tk.VERTICAL, command=tree.yview)
tree.configure(yscroll=scrollbar.set)
```



```
# Start application
root.mainloop()
```

THANK YOU