**XJTLU Entrepreneur College (Taicang) Cover Sheet**

|  |  |  |
| --- | --- | --- |
| Module Code and Title | DTS207TC Database Development and Design | |
| School Title | AI and Advance Computing (AIAC) | |
| Assignment Title | Coursework 2 | |
| Submission Deadline | **23.59 pm** China time (UTC+8 Beijing) on Tuesday **24ST December 2024** | |
| Final Word Count |  | |
| If you agree to let the university use your work anonymously for teaching and learning purposes, please type **“yes”** here. | | **yes** |

I certify that I have read and understood the University’s Policy for dealing with Plagiarism, Collusion, and the Fabrication of Data (available on Learning Mall Online). With reference to this policy, I certify that:

* My work does not contain any instances of plagiarism and/or collusion.  
  My work does not contain any fabricated data.

**By uploading my assignment onto Learning Mall Online, I formally declare that all of the above information is true to the best of my knowledge and belief.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Scoring – For Tutor Use** | | | | | | |
| **Student ID** | | | | **2254164** | | |
|  | | | | | | |
| **Stage of Marking** | | **Marker**  **Code** | **Learning Outcomes Achieved （F/P/M/D）**  **(please modify as appropriate)** | | | **Final**  **Score** |
| **A** | **B** | **C** |
| 1st Marker – red pen | | **FA** |  |  |  |  |
| Moderation  – green pen | | **IM**  **Initials** | The original mark has been accepted by the moderator (please circle as appropriate): | | | Y / N |
|  | Data entry and score calculation have been checked by another tutor (please circle): | | | Y |
| 2nd Marker if needed – green pen | |  |  |  |  |  |
| **For Academic Office Use** | | | **Possible Academic Infringement (please tick as appropriate)** | | | |
| **Date**  **Received** | **Days late** | **Late Penalty** | **☐ Category A** | | Total Academic Infringement Penalty (A, B, C, D, E, please modify where necessary) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  |  |  | **☐ Category B** | |
| **☐ Category C** | |
| **☐ Category D** | |
| **☐ Category E** | |

**Project Report**

Before continuing, here is some important information:

* In this coursework, most files and codes are edited using VS Studio Code. However, other similar programs may be used.
* The program is structured as follows:

web\_basicSortingProgram

| xml-csv # folder that contain materials for xml-to-csv conversion

| web # folder that includes source material for webpage implementation

1. **XML Validation and Conversion**

Both process is conducted under a single Python script titled ***task\_1.py***

* Validation: Perform formatting consistency check of all the 3 XML files using a schema file titled s*chema.XSD*. The process reveals several formatting mistakes in the files:
* Info.xml = direct usage of ‘&’ (solution: change ‘&’ to ‘&amp’)
* Price.xml = the tag symbol is not closed properly (solution: automated correction made using *price\_corrector.py*)
* Time.xml = A root tag is non-existent (solution: add a root tag <time>)
* Conversion: extracting and writing down table information to its CSV file counterpart.



Figure 1.1 – Snippets of *info.csv*

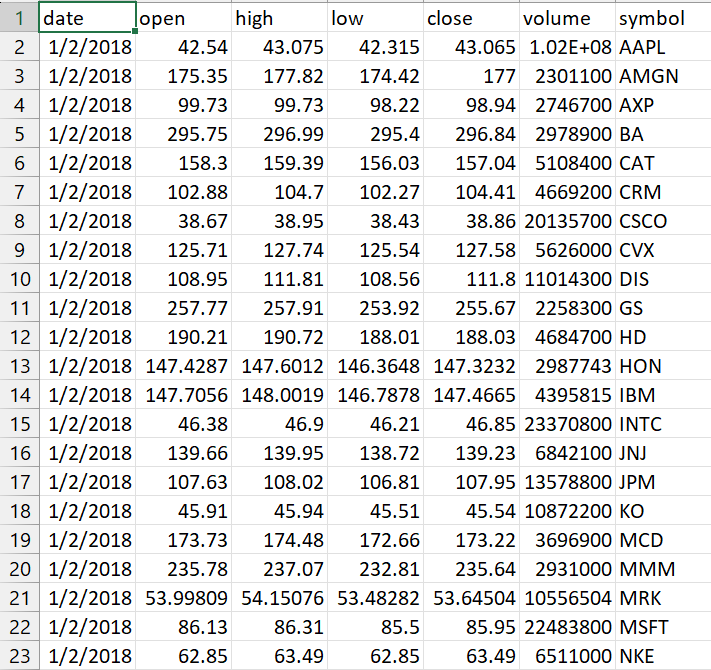
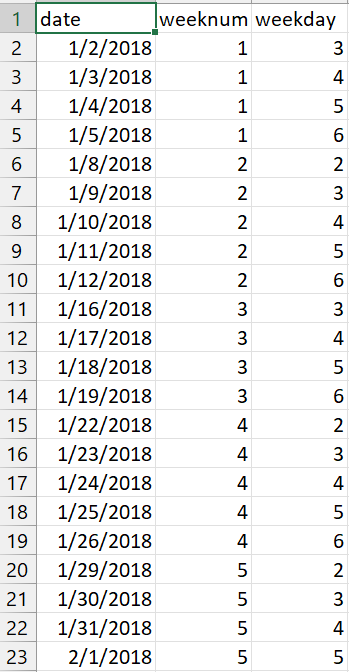
****

Figure 1.2 – Snippets of *price.csv* and *time.csv*

1. **ORM and ERD Design**

In this step, we will mount all data from CSV files created in the previous step inside a table set in the Postgres database named: dts207cw2

* The mounting process is conducted with *app.py* (specified in line 41, called in lines 108-140)

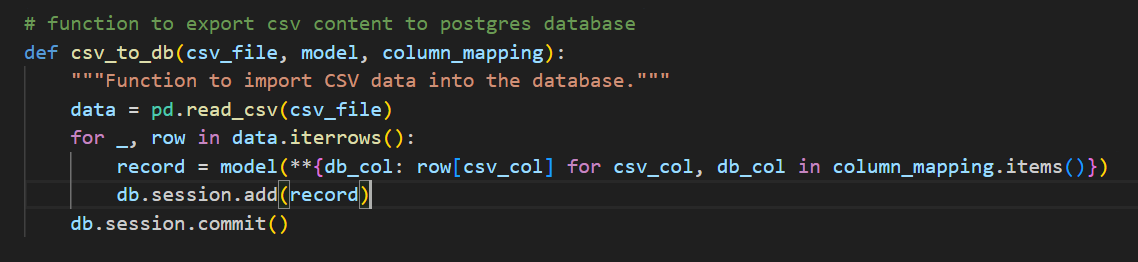


Figure 2.1 – Function to export CSV data to the database

* The database is constructed using ERD as described in the image below. The table elements ‘symbol’ and ‘date’ are to be made to reference each other.

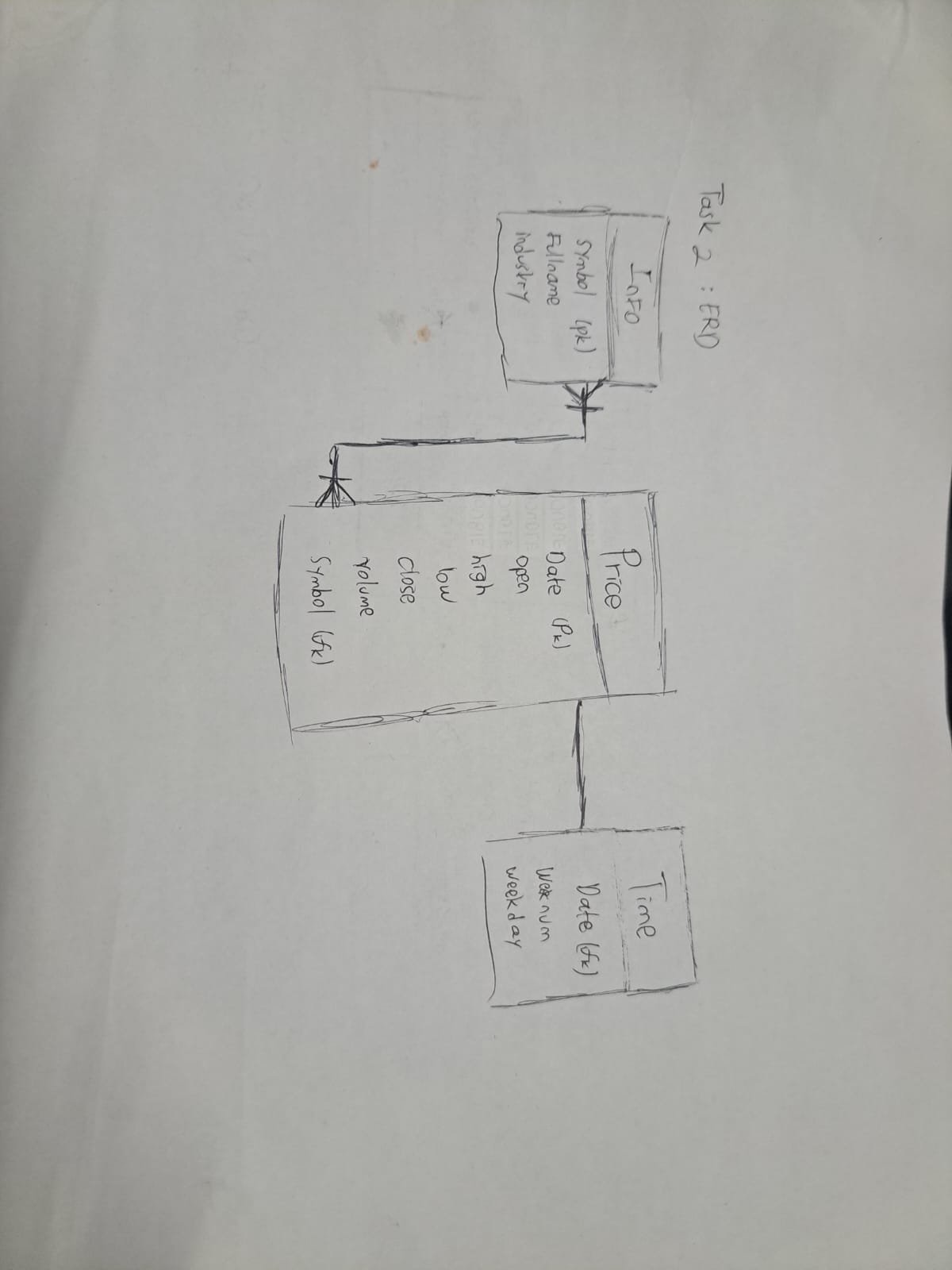


Figure 2.2 – ERD Design

* Based of the ERD design above, the ORM implementation should look like this:

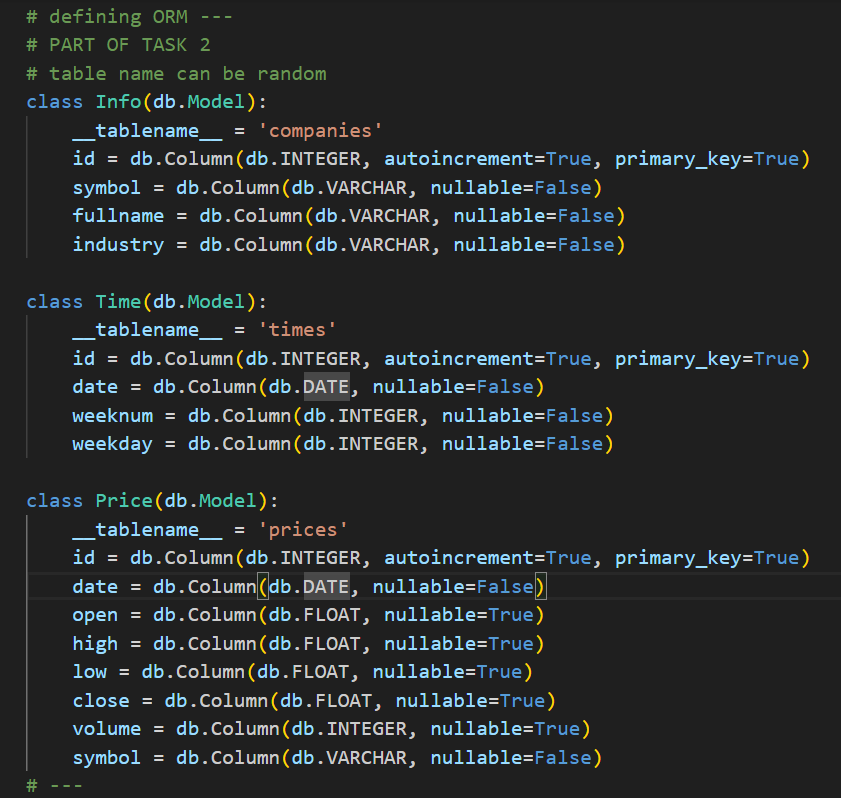


Figure 2.3 – ORM Implementation

1. **Web Page Implementation**

In summary, line 52 of *app.py* routes the table's contents from the dts207cw database into an HTML file (table.html) so that Flask can display the intended content at the specified local IP address. Running the command `app.run` in the 'main' module will provide a link redirecting us to this webpage.

* Pagination of the webpage is also being conducted in this step. For my case, I set the 30 rows of the maximum viewable content before moving on to the next page.

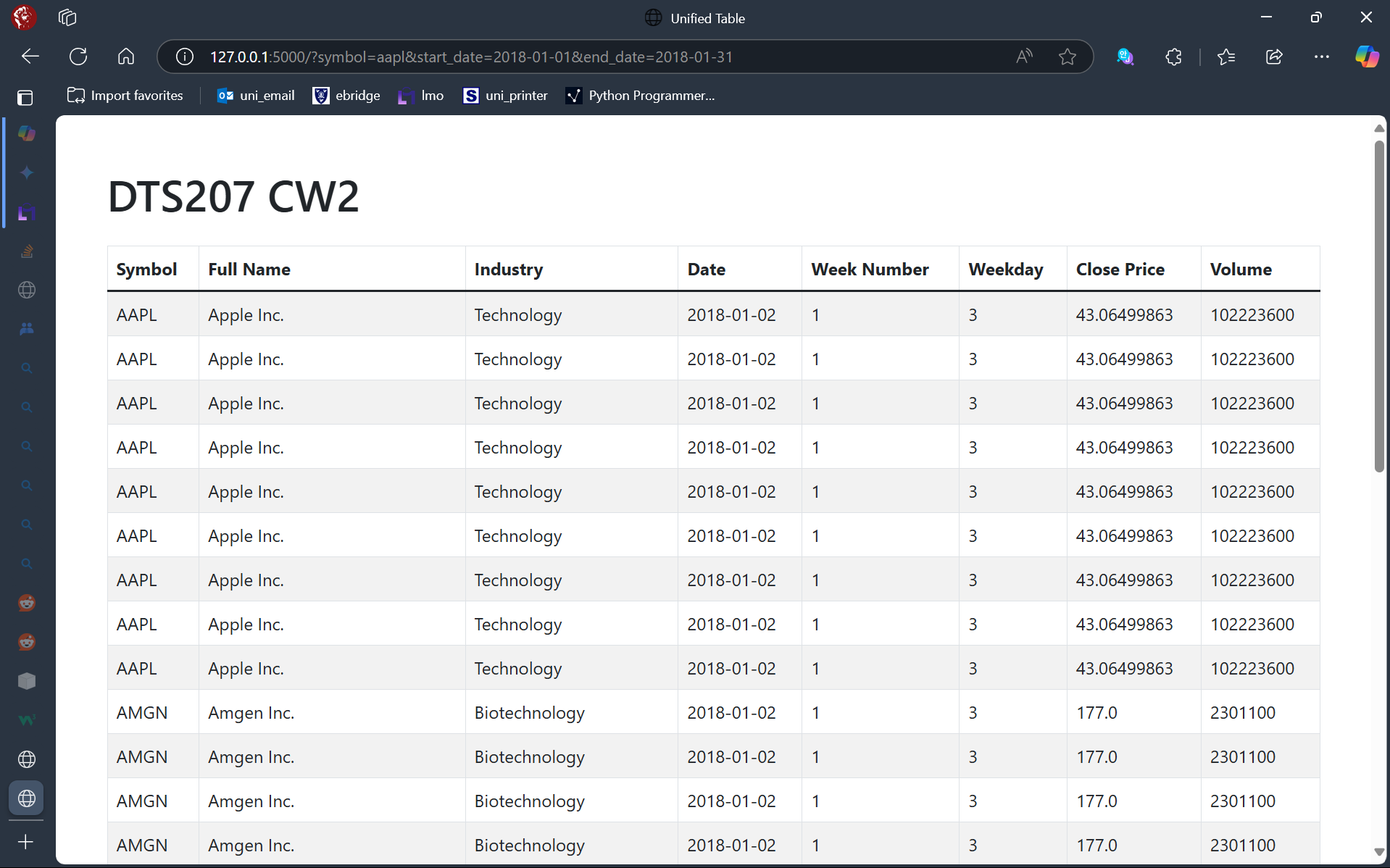


Figure 3.1 – Preview of the table on web--browser

1. **Adding Filtering Functionality**

In this step, I slightly modified the routing section of *app.py* and the webpage *table.html* to include a filter for ‘symbol’ and ‘date’.

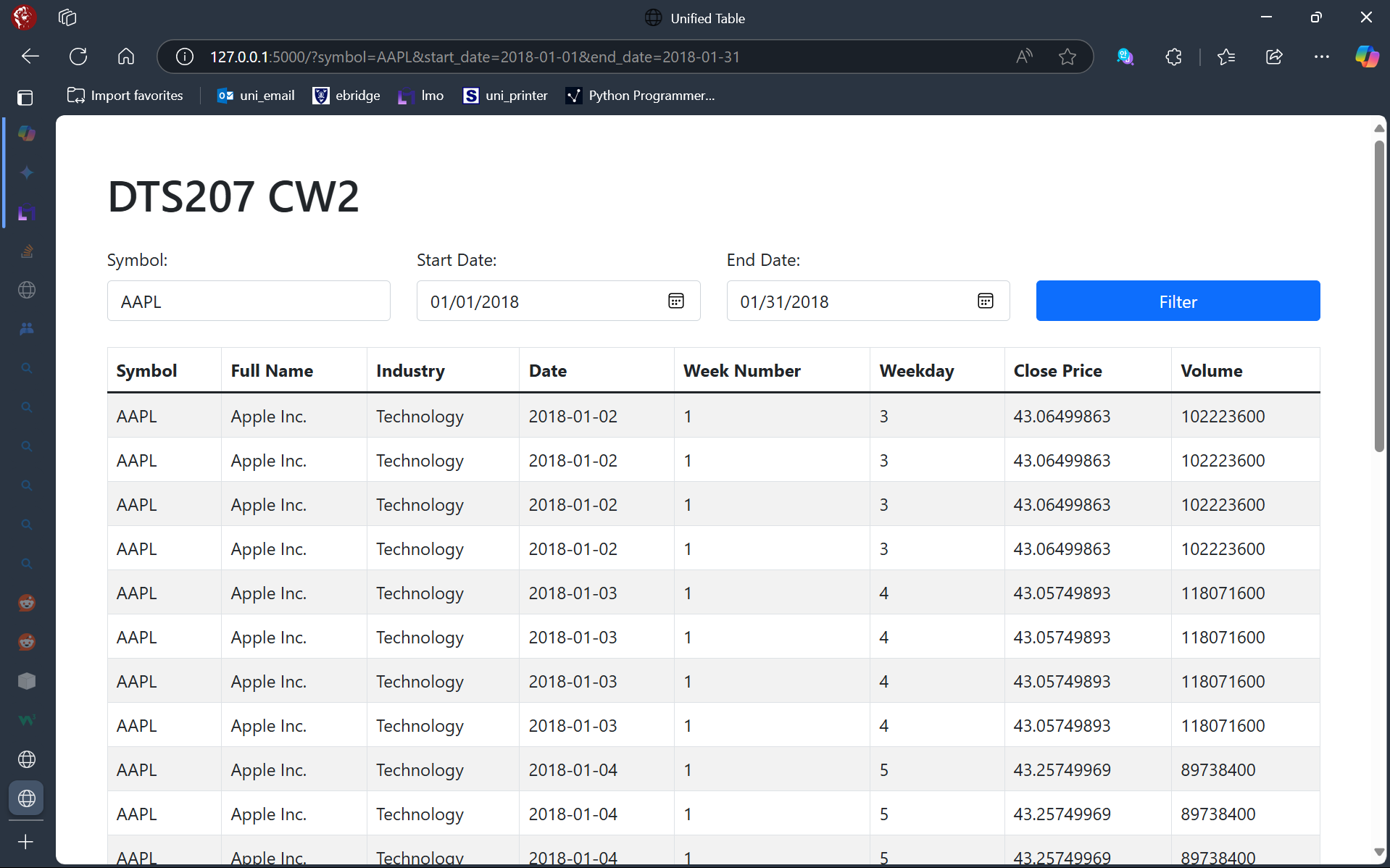
****

Figure 4.1 – Preview of the filter

1. **Runtime Testing**

Essentially, we use the provided test script along with other CW materials to demonstrate the query running time of the created website.

* *app.py* must be active during the testing phase

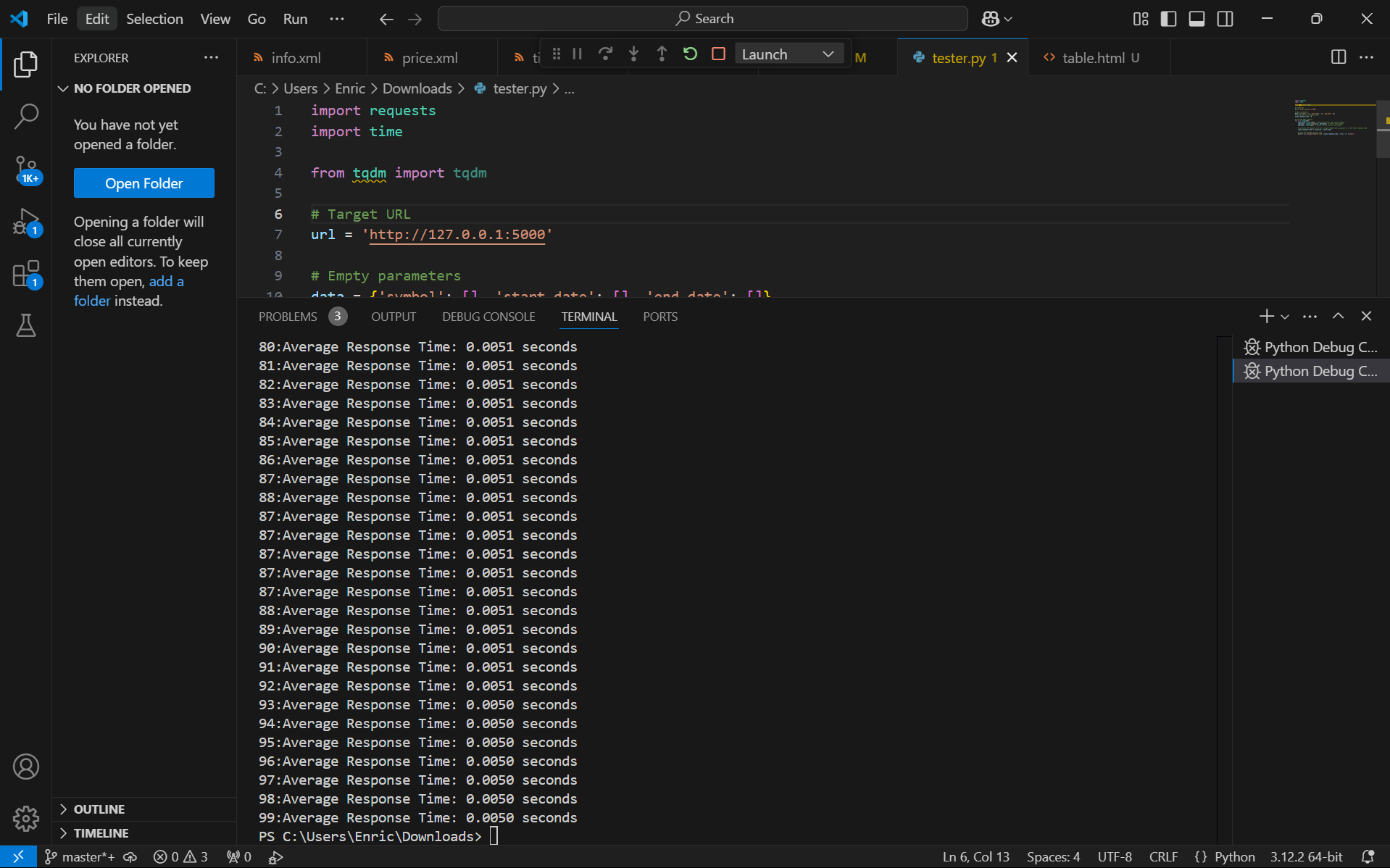
****

Figure 5.1 – Utilization of test program showing the last 20 calls