Regular Recording of Market Prices with Web Scraping Methods and Monitoring of Price Inflation

Submitted to the Graduate School of Natural and Applied Sciences for master's w/o thesis term project in Software Engineering

Writer's Name: Berkcan Teber

ORCID 0000-0003-4489-7166

Thesis Advisor: Prof. Dr. Femin Yalçın Küçükbayrak

January, 2023

Declaration of Authorship

I, Berkcan Teber, declare that this thesis titled Regular Recording of Market Prices with Web Scraping Methods and Monitoring of Price Inflation and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for the Master's / Doctoral degree at this university.

- Where any part of this thesis has previously been submitted for a degree or anyother qualification at this university or any other institution, this has been clearly stated.

- Where I have consulted the published work of others, this is always clearly attributed.

- Where I have quoted from the work of others, the source is always given. Thisthesis is entirely my own work, with the exception of such quotations.

- I have acknowledged all major sources of assistance.

- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 15.01.2023

Regular Recording of Market Prices with Web Scraping Methods and Monitoring of Price Inflation

Abstract

Due to the political and economic problems experienced today, almost all countries are experiencing inflation problems and the purchasing power of the people is decreasing due to this problem. States are taking very drastic measures to solve this problem. Food inflation is a fundamental component of total inflation. This study was written in Java in order to automate the regular recording of the product prices of some chain supermarkets in a spreadsheet by using web scraping methods through Selenium, thus contributing to a more accurate determination of price inflation. can be controlled and recorded.

Thanks to this study, it is planned to contribute to the studies done or to be done on food inflation. In addition, not only the relevant institutions, but also anyone who wants will be able to easily control the product prices and, if desired, calculate the change between the prices.

Keywords: Selenium, web scraping, inflation, supermarket, food price

Veri Kazıma Yöntemleri ile Market Fiyatlarının Düzenli Olarak Kaydedilmesi ve Fiyat Enflasyonunun Takibi

Öz

Günümüzde yaşanan siyasi ve ekonomik problemlerden ötürü neredeyse tüm ülkeler enflasyon sorunu yaşıyorlar ve bu sorun yüzünden halkların alım gücü gittikçe düşüyor. Devletler bu sorunu çözmek amacıyla çok sert önlemler alıyorlar. Toplam enflasyonu oluşturan temel bir parça da gıda enflasyonu. Bu çalışma Java dilinde, Selenium aracılığıyla veri kazıma yöntemleri kullanılarak bazı zincir süpermarketlerin ürün fiyatlarının bir elektronik tabloya düzenli olarak kaydedilmesinin otomatikleştirilmesi ve bu sayede fiyat enflasyonunun daha doğru tespitine katkıda bulunulması amacıyla yazılmıştır. Bu sayede sadece enflasyon sepetindeki ürünlerin değil, supermarket raflarında bulunan tüm ürünlerin fiyatları düzenli olarak kontrol edilebilecek ve kayıt altına alınabilecek.

Bu çalışma sayesinde gıda enflasyonu konusunda yapılan ya da yapılacak çalışmalara katkı sağlanması planlanmaktadır. Ayrıca sadece ilgili kurumlar değil, isteyen herkes kolaylıkla ürün fiyatlarını kontrol edebilecek ve istenirse fiyatlar arasındaki değişimi hesaplayabilecek.

Anahtar Kelimeler: Selenium, veri kazıma, enflasyon, süpermarket, gıda fiyatı

Table of Contents

Declaration of Authorship i

Abstract ii

Öz iii

Table of Contents iv

List of Figures vi

List of Abbreviations vii

1 Introduction 1

2 Selenium and Its Application 3

2.1 Selenium WebDriver 3

2.2 Selenium Grid 3

2.3 Selenium IDE 4

2.4 Selenium Remote Control 4

3 Java and Its Distinguishing Features from Other Programming Languages 5

4 Integrated Development Environment (IDE) 8

5 Web Scraping 9

6 Maven 10

7 Locators of Elements 11

7.1 Using of Xpath 11

7.1.1 Contains () 12

7.1.2 OR & AND 12

7.1.3 Starts-with() 13

7.1.4 Text() 13

7.1.5 Following 13

7.1.6 Ancestor and Parent 13

7.2 Using of CSS Selectors 14

8 The Inflation and The Importance of Analyzing It at Short Intervals 16

9 Web Scraping of Prices Using Selenium 18

9.1 Creating The Project on Intellij and File Structure in The Project 18

9.2 Maven Structure and Pom.xml 19

9.3 Basic Codes in The Project 19

9.4 Codes Working Before and After The Application 20

9.5 Chromedriver Related Settings 21

9.6 Codes Running Before Main Class Codes 23

9.7 Constants on The Pages 25

9.8 Saving The Found Texts to Excel 26

9.8.1 Structure on Migros Website 26

9.8.2 Codes Related to Saving Data and Exporting to Excel 27

9.8.3 Creation of The Excel File and Saving 28

10 Possible Contributions of The Project 29

References 30

Curriculum Vitae 32

List of Figures

Figure 9.1 Shows the classification of the project by folders ……..19
Figure 9.2 Some code snippets in the BasePage class ……20
Figure 9.3 Some code snippets in the BaseTest ……..21
Figure 9.4 Code snippets showing browser settings in the project ……22
Figure 9.5 Code snippets showing test settings in the project ……24
Figure 9.6 Code snippets showing some constants of Migros page ……25
Figure 9.7 Image of the first category on Migros online shopping site ……...26
Figure 9.8 Code snippets showing the way of recording found values from Migros online shopping website on excel ……27
Figure 9.9 Sample data saved in excel after running Migros application ……28

List of Abbreviations

İKÇÜ	İzmir Kâtip Çelebi University
M.Eng.	Master of Engineering
B.Eng.	Bachelor of Engineering
B.B.A.	Bachelor of Business Administration
TÜİK	Türkiye İstatistik Kurumu

Introduction

Web scraping is a computer program technique of extracting information from websites. The simple copying technique made by humans can actually be considered as a web scraping technique. Apart from this, the process of automatically extracting information from a website at regular intervals with some tools is one of the web scraping techniques. In this study, web scraping was done in Java language with Selenium, which is mostly used for testing.

Selenium is basically a tool that automatically performs user actions such as clicking on a website screen or typing in a field. In this study, selenium was used to reach different products by going to the online shopping site of supermarkets, such as the user. Then, the names and prices of these products were recorded sequentially in an excel file with Java and these processes were automated. In other words, product prices of some of Turkey's largest chain markets such as Migros, A101 and Şok can be scraped with regular planning, daily or weekly thanks to this project.

While in similar studies, web scraping is mostly used for price tracking, obtaining outsourcing data for finance, sentiment analysis or news and content tracking, in this study the main aim is to scrap product and price information from online websites of supermarkets. This brings this work to a different point than the others. In addition, selenium was used for web scraping in this study, but other studies generally use ParseHub, Scrapy, OctoParse, Scraper API, Webhose.io tools. This is one of the differences between this study and similar studies. More detailed information about these web scraping tools will be given separately.

Today, inflation is the main cause of economic problems in many countries and it is rapidly melting people's savings and salaries. States are trying to take action against it. The main reason for this study is to capture and record product prices from different supermarkets with Selenium. In this way, it is planned to contribute to the calculation of food inflation. In addition, if it is desired to be used in a different study later, the prices of many products can be archived. Finally, by taking this study as a reference,

product prices can be obtained from different supermarkets by web scraping in a similar way.

Selenium and Its Application

Selenium is a free and open source automated testing framework and it is used to validate web applications on different browsers and platforms. Selenium can supports different programming languages such as Java, C#, Python, etc. Contrary to popular belief, selenium is not a single tool, but a library of packages. For this reason, it is called Selenium Suite. Selenium is divided into 4 main parts according to the features which they provide.

2.1 Selenium WebDriver

Selenium WebDriver is a framework designed for creating and executing test cases. These test cases are created and executed via element locators in WebDriver. It is important to remind that Selenium WebDriver is a programming library, not an IDE.

2.2 Selenium Grid

Selenium-Grid, which has been developed by Selenium, runs different browsers in parallel on different servers. The main purpose here is to see the test results for different operating systems, hardwares and devices and to run test processes in parallel in a distributed environment and to obtain test results quickly. When these tests run in parallel, a significant time saving is achieved.

Selenium-Grid performs selenium tests using Selenium Hub and Node structure.

Hub: This structure, which acts as a server, hosts many processes on itself, and because of this, we can test the same code on different platforms and browsers by responding to requests from different clients.

Node: The structure consisting of one or more clients connected to the hub is called a node. We can perform selenium tests by making requests from many Nodes to a single Hub using selenium-grid strcuture.

2.3 Selenium IDE

Selenium IDE is a browser extension for creating and running tests. Selenium IDE has some key advantages. For example, it can act like a normal user to perform and save operations accordingly. It has support for many languages such as Java, .Net and Python. Therefore, it is preferred more than other test tools thanks to its multi-language and platform support. [1]

2.4 Selenium Remote Control

Selenium RC, also known as Selenium 1, was previously the main Selenium project. It supports mainly JavaScript for automation, but it also supports Ruby, PHP, Python, Perl, C#, Java. In addition, Selenium RC works almost on every browser. Unfortunatelly, Selenium RC is officially deprecated today [2].

Java and Its Distinguishing Features from Other Programming Languages

When the first computers came into existence, the high-level languages we now use such as FORTRAN, COBOL, Pascal, C/C++, Java did not exist. Since the hardware structures of different machines are different, the machine languages of different brands and models of computers were different from each other and each machine could only understand that language. For this reason, the first programmers could only make the computer work with the programming language of the machine they were using. In addition to being difficult to learn machine language, the machine language learned for a machine could not be used for different brands and models. To overcome this difficulty, first assembly language and then different high-level languages were created. High-level languages brought great comfort to programmers. Because the programmer could write the source program in the language he/she wanted without thinking about the operating system and the machine.

The first programming languages, such as COBOL, served programmers successfully for many years and they still do. But in that time there was a problem. A source program compiled on a particular type of machine running under a particular operating system could only be run on that type of machine running under that operating system. When the operating system and/or machine type changes; so when the platform changed, the program couldn't run there; it had to be recompiled with a compiler suitable for the new platform. This problem is called platform dependency shortly. What we mean by platform dependency is the fact that a source program is compiled with a specific compiler running under a specific operating system and can only be run on certain types of computers. For example, it is not possible to run a computer program compiled on a PC which is running Windows operating system on a machine which is running Macintosh or Linux operating system.

To solve this problem, it is necessary to create a language that can work independently of the platform. Sun company solved this problem with the Java language it developed. In fact, Sun's main goal was not to solve this big problem, but to develop a language

that would enable easy use of electrical appliances. And that's why Sun's programmer, James Gosling, created the Java language that works on every platform in 1995.

Gosling made a simple but wonderful invention. He designed a common virtual machine that can be installed on different operating systems and different hardware. This virtual machine, called JVM (Java Virtual Machine), was distributed free of charge. Today, JVM can be easily installed on every platform and source programs written in Java language are converted into a kind of machine language that can run in JVM with java compiler.

Anyone who wants to run Java applications can install the JVM on their machine. For this, it is sufficient to download and install the program called JRE (Java Runtime Environment) from the internet. The JRE is installed on the computer once and after that, all Java applications can run on this machine. When a java application is running on the machine, the JRE automatically creates the JVM virtual machine. The JVM is a program that runs when needed; Like any program, it's deleted from main memory when it's done. Today, most browsers that use java applications are capable of automatically downloading and installing the JRE.

Java is a simple, modern, object-oriented, type-protected language that inherits all the good features of C and C++. Moreover, it has the ability to work on any platform. Due of this capability, it may be used in a variety of areas, including computers, internet applications, mobile phones, game consoles, and home appliances. Java may be considered as both a programming language and an environment, for this reason. The operating system, networks, internet programming, databases, and other middleware technologies are all included in this ecosystem.

Java has been choosen for this project as the programming language. Compared to other languages, Java offers some benefits. For example, we can give Java as an example of object-oriented programming language. It offers all of the benefits of object-oriented programming. Important programming features including inheritance, polymorphism, modular programming, debugging, and code reuse are available to the programmer. Due to its OS and hardware independence, Java Bytecode can be easily migrated from one computer system to another. Java detects errors that other languages can only detect at run time, at compile time. It has strong debugging capability. The

Java language, compiler, and interpreter are designed with security as a priority. It is the first language that emphasizes security in its design [3].

Integrated Development Environment (IDE)

An integrated development environment (IDE for short) is a type of software that aims to enable computer programmers to develop software quickly and comfortably, includes many tools that can organize the development process and all of the tools that contribute to the efficient use of the development process [4].

Most famous IDE's are Eclipse, Microsoft Visual Studio, Code::Blocks and Intellij IDEA. In this project, I have chosen to use Intellij IDEA because it is especially suitable for Java. IntelliJ is a popular and widely used Java Integrated Development Environment (IDE) produced by JetBrains. It is an IDE with support for Windows, Mac and Linux environments [5].

Web Scraping

Web scraping is the process of automatically pulling the data we want from a website. We can use and interpret this data as we want. There are various uses for web scraping. For example, in price comparison transactions, coupon discount code withdrawal transactions, banks' exchange rates. We can use webscraping to pull the data we want from the frontend screens like these. In this study, web scrapin was made with selenium in order to attract product prices from the markets.

There are many different tools or frameworks that are used as ready-made packages for webscraping. For example, it can be done with the Scrapy framework written in Python. It can be also done with the Google add-on Chrome Scraper. Apart from these, paid tools such as Scrape.do and ScrapingBee can also be used for webscraping [6].

Selenium was used for webscraping in this project. The most important reason for this was that selenium could work on every screen, even if the screens vary according to the markets, since selenium can do every operation that a normal user can do. For example, if javascript was used on the screen where webscraping was requested, Scrapy could cause a problem, but such a problem would not occur in selenium. However, the biggest disadvantage of selenium is that webscraping with selenium takes longer time compared to other tools [7].

Maven

Maven is a powerful project management tool based on POM (Project Object Model). It is used for project creation, dependency and documentation. Like ANT it simplifies the compilation process. However, it is a much more advanced technology than ANT. In short, it would not be wrong to say that Maven is a tool that can be used to create and manage any Java-based project. Maven simplifies the day-to-day work of Java developers and is generally helpful in understanding any Java-based project.

Maven becomes much more important and functional especially in some cases. For example, if there are too many dependencies in the project and these dependencies need to be updated frequently, Maven becomes much more functional. In this project, many different dependencies were used.

The most important file in a maven project is the POM file. Now let's talk about this file. The Project Object Model(POM) file is a file that contains both information about the project and its dependencies, source, plugins used, commands required to compile the project, etc [8].

Maven uses the following fields in the pom.xml file to distinguish dependencies from each other in special storage areas called repository;

-groupId : Shows the organization to which the application belongs, in order to avoid groupId conflicts between different applications, the reverse order of the organization's web address is used as the groupId.

- artifactId : The name of the application and it must be unique within the organization

- version : The version of the application

- packaging : The method of packaging the application (jar/war/zip)

If the dependency is not in the local repo, Maven recursively downloads the dependency library and other dependencies from the central repo to the local repo and makes them available to the applications that need it [9].

Locators of Elements

There are some attributes or formats we can use to enable us to access elements to click on the screen, write text or get text. Examples of these are id, className, linkText, name, cssSelector, xpath. Between them, there is a hierarchy of importance. If our element has an id in the first place, we need to use it because usually the ids are created specifically for the element and there is no possibility of conflict with another element. However, if the id is just numbers, it shouldn't be used because these numbers usually change every time the screen is updated. Therefore, it is inevitable to get an error at this point when the application is run again. However, if the element has a unique id, the same element can always be successfully accessed and the probability of the application failing can be reduced. If there is no unique id then attributes such as className or name can be used. If we cannot reach the element we want with these, xpath or cssSelector can be used because some elements may not have className or name. To access these classes or ids belonging to the element, after opening the browser, right click on the text or button and click on the inspect button. In this way, the attributes and properties of the element can be accessed.

7.1 Using of Xpath

Xpath is simply defined as an xml path. We can say the syntax or language used to find any item on web pages. It uses HTML and the DOM framework to find any element on a web page. Absolute XPath is the direct way to find the item, but the downside of Absolute XPath is that XPath fails if any change is made to the item's path. The Absolute XPath example is shown below.

Absolute xpath:

```
html/body/div[2]/section/div[2]/div/div/div/div[1]/div/div/div/div/div[1]/div[3]/div/h4[1]/a
```

When using Relative Xpath, the path begins in the center of the HTML DOM hierarchy. It starts with a double slash (//), meaning it can search for the item anywhere on the webpage. It can be started from the middle of the HTML DOM structure and does not

require long xpath typing. Relative is the common format used to find the item via XPath. Relative XPath has the advantage of being easier to use and requiring less upkeep and expense.

Relative xpath:
//*[@class='featur-box']//span[text()='Testing']

When using Xpath, some dynamic elements can be benefited so that the searched element can be found much faster and more precisely.

7.1.1 Contains ()

Contains() is used when the value of any property changes dynamically, for example login information. The Contains property is capable of finding the element containing partial text. For example, the element can be found this way by including a name or a text. Below are examples with contains().

```
Xpath=//span[contains(@type,'sub')]

Xpath=//td[contains(@name,'btn')]
```

7.1.2 OR & AND

In the OR expression, 2 conditions are used for either condition 1 or condition 2 to be true. It applies if either condition is true or both conditions are true. It means that any condition must be true to find the item. In the AND expression, two conditions are used, both conditions must be true to find the element. If any condition is false, the item cannot be found. Below are examples with 'or' and 'and'.

```
Xpath=//tr[@type='submit' or @name='btnReset']

Xpath=//input[@type='text' and @name='btnLogin']
```

7.1.3 Starts-with()

The starts-with function finds the element whose attribute value is changed in any action on the web page. In this expression, it matches the property's start text to find the element whose attribute changes dynamically. Of course, the item whose attribute value is static can also be found with this function. For example, if the last part of the id of a certain element consists of numbers and changes every time the page is refreshed, the unchanged part of this id can be found with the help of this function. Below you can see an example of using this function.

```
Xpath=//span[starts-with(@name,’message’)]
```

7.1.4 Text()

With the Text() function, the exact matching text element of the element is found. In the example below, the element with the text "UserID" is found with the text function.

```
Xpath=//input[text()=’UserID’]
```

7.1.5 Following

When using Xpath, there is sometimes a need to reach the previous or next sibling element. In such cases, the desired element can be found using 'following-sibling' or 'preceding-sibling'. Below are some examples of this.

```
Xpath=//button[contains(.,'Reader')]/preceding-sibling::button[@name='settings']
```

```
Xpath=//a[@href='/accounting.html'][i[@class='icon-usd']]/following-sibling::h4
```

7.1.6 Ancestor and Parent

If the element you want to reach does not have an id and you have difficulty in reaching it, but its grandchild element has a unique value, you can first reach the grandchild element and then the ancestor element can be reached. In such a case, 'ancestor' is used. Similarly, if the element to be reached does not have an id and you have difficulty in

reaching it, but if its child element has a unique value, you can reach the child element first and then reach the parent element in a similar way [10].

```
Xpath= //*[@id='rt-feature']//parent::div

Xpath= //div[.//a[text()='SELENIUM']]/ancestor::div[@class='rt-grid-2 rt-omega']
```

7.2 Using of CSS Selectors

CSS Selector or xpath is usually used if an element has no information about ID or name, or if they are variables. Compared to xpath, CSS Selector works faster. Therefore, if there is no obstacle, CSS Selector is preferred first. However, some elements may be impossible to reach with CSS Selector, in such cases the use of xpath is inevitable [11].

The use of CSS Selector with different attributes can be summarized as follows.

ID → #

```
Example: #toc3
```

Class → .

```
Example: .guide-toc
```

Attribute → [attribute=value]

```
Example: input[type=’submit’]
```

Sub-String → [attribute^=value]

Contains (*)

```
Example: input[id*=’logi’]
```

Starts with (^)

Example: input[name^=’first’]

Ends with ($)

Example: input[class$=’fi’]

The Inflation and The Importance of Analyzing It at Short Intervals

The constant rise in the average price of goods and services is referred to as inflation. Inflation covers not only the price change of one or a few goods and services, but also of all goods and services used by an average consumer during the year. In other words, in a country, while the inflation rate increases, the prices of some goods and services may decrease, or similarly, while the inflation rate decreases, the prices of some goods and services may increase. In addition, for the increase in prices to be defined as inflation, it must be continuous, not just for a certain period.

In addition to the prices of goods and services, salaries and wages also change over time. However, when the increase in salaries and wages is less than the increase in the prices of goods and services, inflation reduces the purchasing power of consumers. In short, inflation causes people to buy less goods and services with the money they have than in the past.

Inflation, which is generally expressed as an annual percentage change compared to the same period of the previous year, is calculated by looking at the change in the average prices of the items in the goods and services basket during one year. Goods and services basket is the name given to the sum of goods and services whose prices are followed during a certain period in order to calculate inflation. The weights of the goods and services in the basket are determined on the basis of the expenditures made by a large sample of households subject to the household budget survey throughout the year.

There are 4 main reasons for inflation. First of all, in demand inflation, the supply of goods and services not keeping pace with the increase in aggregate demand causes an increase in the prices of goods and services. Another reason is cost inflation. This inflation is caused by the decrease in the total supply as a result of the increase in the production costs due to the increase in the prices of commodities such as oil and food or natural disasters in an economy, and the subsequent increase in the general level of

prices. Another reason for inflation is the money supply. An increase in the money supply in an economy can lead to an upward pressure on prices by increasing investment and consumption expenditures. Finally, inflation expectations are one of the factors that play a key role in the formation of inflation. If consumers and producers expect that prices will continue to rise in the future, these expectations are reflected in the prices of goods and services through future wage demands.

The costs of inflation are important not only for individuals or companies, but also for society and the economy as a whole. The most obvious cost of inflation is seen in the decision-making phase of consumers and investors. The high inflation environment creates uncertainty for economic units, affects the decisions to be taken and pushes individuals to indecision about consumption and investment. The continuous and variable increase in prices makes it difficult for consumers to compare different goods and services with each other and therefore to choose the product to buy. This may cause consumers to limit their consumption. In addition, firms are less willing to invest because they cannot foresee their future costs and profits. Investments are generally made in the medium and long term. In the inflation environment, prices are constantly increasing, and even the rate of increase in prices may vary in different periods. Considering that the expenses of the investments will be paid in the medium and long term and the profits to be obtained from the investments cannot be obtained in a short time, it becomes difficult for the companies to calculate whether the investments in question will be successful or not [12].

Web Scraping of Prices Using Selenium

9.1 Creating The Project on Intellij and File Structure in The Project

First of all, when starting the project, an empty maven project was created on intellij. This project was named "Migros" because it was created to scrap data from the Migros supermarket. For ease of management, the basic structures of the project were divided into folders. First, folder named "properties" was created and then the second folder named "driver" were created under it. In this project, Chrome was selected as the browser and chromedriver was downloaded for this browser to work locally, then this chromedriver.exe file was put under the driver folder. The important point here is that the downloaded chromedriver.exe should have the same or very close versions of the local chrome browser version. If there are serious differences between the two versions, an error may occur when running chromedriver.exe.

Then a new folder was created under the Java folder and named "com.webScraping.migros". Other folders containing the codes of the project were placed under this general package folder. The project was divided into 5 basic areas and folders named "base", "constants", "page", "scrap" and "util" were created for each of these 5 areas. In the figure below, you can see the general folder grouping of the project.

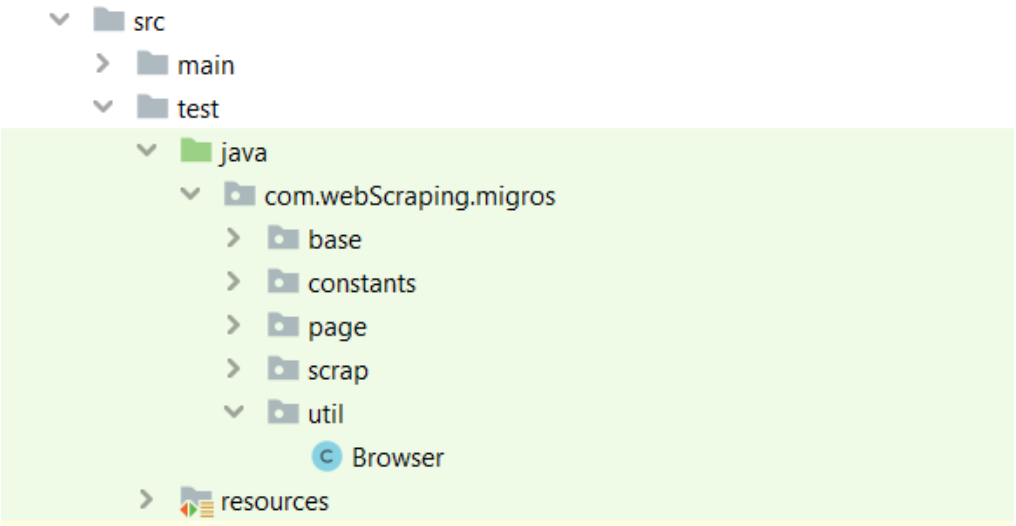

Figure 9.1: Shows the classification of the project by folders

9.2 Maven Structure and Pom.xml

In addition, this project was created as a maven project for easy management of libraries. There is a file called pom.xml for easy management of these libraries in Maven projects. Libraries such as "selenium-java", "junit", "log4j-core" and "poi" have been added as dependencies to this pom.xml file that can be used in the project in the future.

9.3 Basic Codes in The Project

Two Java classes named "BasePage" and "BaseTest" were created under the base folder. Functions such as "waitSeconds", "findElement", "getText", "waitUntilElementAppear" and "waitUntilElementClickable" have been created in the BasePage class that can make it easier to use in the project. The purpose of creating these functions in this class is to avoid cluttering the code in the project. Below you can see sample code pieces of this class.

```
public static void waitSeconds(int seconds) {
    try {
        TimeUnit.SECONDS.sleep(seconds);
    } catch (InterruptedException e) {
        log.error(e.getMessage(), e);
    }
}

public WebElement findElement(By by) {
    return driver.findElement(by);
}

public void waitUntilPresence(By by) { wait.until(ExpectedConditi-
ons.presenceOfAllElementsLocatedBy(by)); }

public void waitUntilElementAppear(By by) {
    wait.until(ExpectedConditions.visibilityOfElementLocated(by));
}

public void waitUntilElementClickable(By by) {
    //log.info("Elementin tıklanabilir olması bekleniyor.");
    wait.until(ExpectedConditions.elementToBeClickable(by));
}
```

Figure 9.2: Some code snippets in the BasePage class

9.4 Codes Working Before and After The Application

Another class in the Base folder is the "BaseTest" class. In this class, functions are defined that will run before the main class of the project is run and after the main class is run. In addition, the url containing the web address of Migros that will be opened when the browser is setting for the first time is also defined here.

```
@Before
public void setUp() {
    log.info("SetUp işlemi yapılıyor...");
    browser.setBrowser("https://www.migros.com.tr/");
}


@After
public void tearDown() {
    getDriver().close();
}

public static RemoteWebDriver getDriver() {
    return driver;
}

public static void setDriver(RemoteWebDriver driver) {
    BaseTest.driver = driver;
}
```

Figure 9.3: Some code snippets in the BaseTest

9.5 Chromedriver Related Settings

In addition, the "Browser" class, which contains additional settings and extensions of the Chrome browser, has been created under the "util" folder. First, we showed the browser the location of the chromedriver.exe file, which was added to the project before. Later, we have closed pop-ups, language translation windows and information screens that may appear on the screen automatically. Because these pop ups can affect web scraping automation and may cause errors. Then chrome settings changed to selenium settings and timeout was added.

```
public void setBrowser(String url) {

    //Chrome driver'ın dizinini belirtir.
    System.setProperty("webdriver.chrome.driver", "properties/dri-
ver/chromedriver.exe");

    // Browser ayarları
    DesiredCapabilities capabilities = new DesiredCapabilities();

    // Chrome ayarları
    ChromeOptions option = new ChromeOptions();

    // Popup'ları otomatik kapatır.
    option.addArguments("disable-popup-blocking");

    // Güvenlik sertifakası hatası varsa es geçer.
    option.addArguments("ignore-certificate-errors");

    // Dil çevirme penceresini kapatır.
    option.addArguments("disable-translate");

    // Browser'ı tam ekran çalıştırır.
    option.addArguments("start-maximized");

    // Otomatik şifre kaydet seçeneğini kapatır.
    option.addArguments("disable-automatic-password-saving");
    option.addArguments("allow-silent-push");

    // Bilgi ekranını kapatır.
    option.addArguments("disable-infobars");
    option.addArguments("--disable-notifications");
    capabilities.setCapability("browserName", "chrome");

    // Chrome ayarlarını selenium ayarlarına dönüştürür.
    capabilities.setCapability(ChromeOptions.CAPABILITY, option);

    // Driver'ı setliyoruz.
    BaseTest.setDriver(new ChromeDriver(option));

    BaseTest.getDriver().navigate().to(url);
    BaseTest.getDriver().manage().timeouts().implicitlyWait(5, Time-
Unit.SECONDS);

}
```

Figure 9.4: Code snippets showing browser settings in the project

9.6 Codes Running Before Main Class Codes

Then, a class named "MigrosTest" was created under the scrap folder and the desired operations were defined with the @before, @test and @after tags. For example, by taking the current date under the @before tag, an excel file is created according to the current date. In addition, column names such as "date", "category", "product name" and "product price" are added to this excel file. Under the @test tag, the main function is called and the project is run. Some codes of this class can be seen in the figure below.

```
@Before
public void before() throws IOException {

    SimpleDateFormat klasor_tarih = new SimpleDateFormat("MMMM-
yyyy", new Locale("tr"));
    SimpleDateFormat sekil = new SimpleDateFormat("dd.MM.yyyy");
    Date tarih = new Date();
    String gununtarihi = sekil.format(tarih) + ".xlsx";
    Workbook wb1 = new XSSFWorkbook();
    Sheet sheet1 = wb1.createSheet("MİGROS");
    String dosyatarihi = klasor_tarih.format(tarih);
    File dosyaYeri = new
File(String.format("C:\\Users\\berkcan\\Desktop\\DATABASE\\" +
dosyatarihi + "\\"));

dosyaYeri.mkdir(); // Migros klasörü oluşturur.
System.out.println(dosyaYeri.getName() + " adlı dosya
Oluşturuldu..");


FileOutputStream fileOut1;
fileOut1 = new
FileOutputStream(String.format("C:\\Users\\berkcan\\Desktop\\DATABAS
E\\" + dosyatarihi + "\\%s", gununtarihi));
System.out.println("Yeni excel oluşturuldu.");


    Row row2 = sheet1.createRow(0);
    Cell cell1 = row2.createCell(0);
    Cell cell2 = row2.createCell(1);
    Cell cell3 = row2.createCell(2);
    Cell cell4 = row2.createCell(3);
    Cell cell5 = row2.createCell(4);
    Cell cell6 = row2.createCell(5);

    cell1.setCellValue("Mağaza Adı:");
    cell2.setCellValue("Tarih:");
    cell3.setCellValue("Ana Kategori:");
    cell4.setCellValue("Alt Kategori:");
    cell5.setCellValue("Ürün Adı:");
    cell6.setCellValue("Ürün Fiyatı:");
    wb1.write(fileOut1);
    fileOut1.close();
    migros1 = new MigrosPage(getDriver());
}

@Test
public void test() throws IOException {
    System.out.println("Migrostan veri çekme işlemi başlıyor.");
    migros1.migros();
}

@After
public void after() {
    log.info("Migrostan veri çekim işlemi sonlandı.");
}
```

Figure 9.5: Code snippets showing test settings in the project

9.7 Constants on The Pages

After the necessary preparations were made for the running of the project, coding was started for the fields to be clicked on the pages of Migros and the texts to be saved from the page. In order to avoid the coding complexity in the Page class, locators such as id and xpath belonging to the elements on the screen have been saved in a separate class under constans.

```
public class ConstantsMigros {

    public static final By urunAdi = By.xpath("//a[@class='mat-
caption text-color-black product-name']");

    public static final By urunFiyati =
By.xpath("//div[@class='price-new subtitle-
1']//span[@class='amount']");

    public static final By altKategoriSayisi =
By.xpath("//div[@class='filter__subcategories ng-star-
inserted']//div[@class='item ng-star-inserted']//a");

    public static final By altKategoriTiklama = By.xpath("(//fe-
breadcrumb[@class='breadcrumb desktop-only-
v2']//a[@class='breadcrumbs__link' and contains (text(),
'Anasayfa')]//parent::li/following-sibling::li)[1]//a");

    public static final By anaSayfaTiklama = By.xpath("//fe-
breadcrumb[@class='breadcrumb desktop-only-
v2']//a[@class='breadcrumbs__link' and contains (text(),
'Anasayfa')]");

}
```

Figure 9.6: Code snippets showing some constants of Migros page

As seen in the figure above, xpaths are often used to locate elements in this project. The reason for this is that the programmers who design Migros' website usually do not give a unique id or class to the elements.

9.8 Saving The Found Texts to Excel

9.8.1 Structure on Migros Website

After the appropriate elements for clicking and receiving text are determined for all pages, the Migros application is started and the browser first stands up. Then the Migros home page opens and the first main category to be opened is determined. Then click on the first main category. It is checked whether there are sub-categories under the main category that opens. If there is a sub-category, the first main category is clicked and the product list of this sub-category is obtained. If there are more sub-categories belonging to this sub-category, they are not clicked and all these products are registered in the common sub-category.

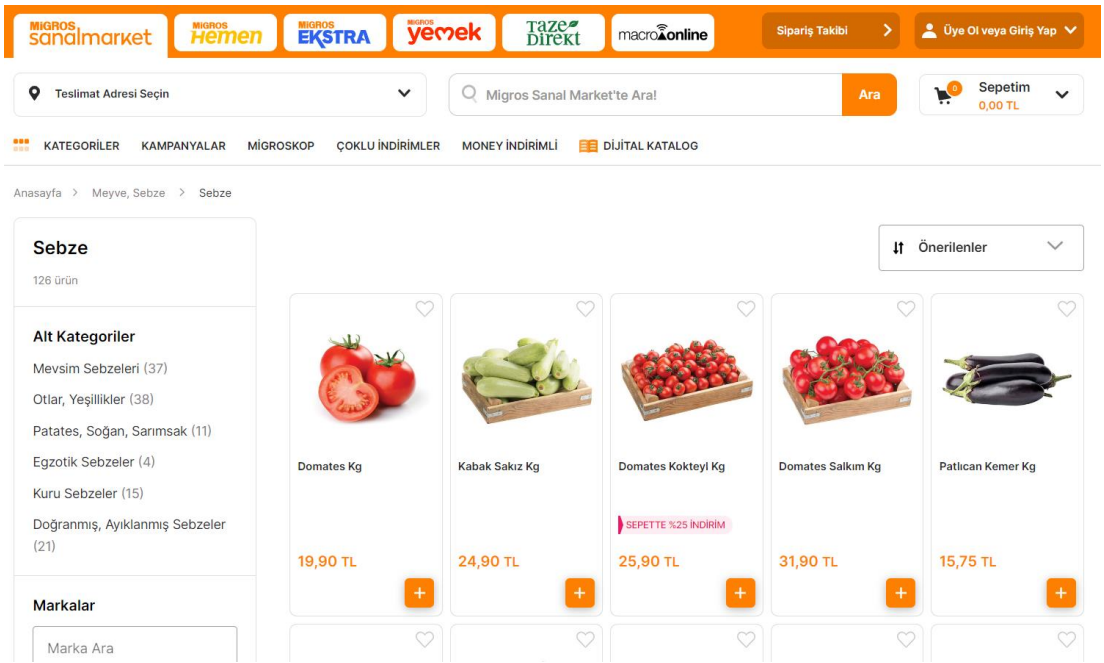

Figure 9.7: Image of the first category on Migros online shopping site

9.8.2 Codes Related to Saving Data and Exporting to Excel

In the MigrosPage class, the number of main categories is found and their titles are recorded. Accordingly, the for loop is used to click on these main headings. Then, with the same technique, subheadings are found and the names of these subheadings are also recorded. Then, the product lists under these sub-headings are reached. Here, first of all, how many pages the product lists consist of is recorded by assigning the last page number on the screen to a variable, and with a new for loop where this variable is used, the product name and price in the number are saved in a list one after the other. When all the products on the screen are finished, this list is saved in excel and the next page button is clicked to see the products on the new page. Proceeding with this way, all products and prices on the Migros online shopping site are recorded in Excel on a daily basis.

```
int rowNum = 1;
int cellNum = 0;
boolean hasRun = false;
Row row2 = null;
while (i.hasNext()) {
    List<String> templist = (List<String>) i.next();
    Iterator<String> tempIterator = templist.iterator();
    rowNum = 1;
    while (tempIterator.hasNext()) {
        String temp = (String) tempIterator.next();
        if (!hasRun) {
            row2 = sheet1.createRow(rowNum++);
        }
        if (hasRun) {
            row2 = sheet1.getRow(rowNum++);
        }
        Cell cell = row2.createCell(cellNum);
        cell.setCellValue(temp);
    }
    hasRun = true;
    cellNum++;
}
```

Figure 9.8: Code snippets showing the way of recording found values from Migros online shopping website on excel

9.8.3 Creation of The Excel File and Saving

Before starting the application, it creates an excel with the name of the day's date and creates a Migros sheet in this excel. After the data extraction from each sub-category is completed, these data are saved in Excel and the lists in the application are cleared, and then data extraction from a new lat category is continued. The data scrapped from the first subcategory can be seen in the figure below.

	A	B	C	D	E	F
1	Mağaza Adı:	Tarih:	Ana Kategori:	Alt Kategori:	Ürün Adı:	Ürün Fiyatı:
2	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Domates Kg	8,95 TL
3	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Migros Havuç Beypazarı Paket Kg	7,95 TL
4	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Kabak Sakız Kg	18,90 TL
5	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Domates Kokteyl Kg	21,90 TL
6	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Patlıcan Kemer Kg	32,90 TL
7	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Hıyar Kg	24,90 TL
8	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Karnabahar Kg	9,95 TL
9	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Domates Salkım Kg	19,90 TL
10	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Kereviz Kg	9,50 TL
11	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Biber Köy Usulü Kg	38,90 TL
12	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Biber Çarliston Kg	26,30 TL
13	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Biber Sivri Kg	29,90 TL
14	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Lahana Beyaz Kg	6,35 TL
15	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Biber Dolmalık Kg	29,90 TL
16	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Pancar Kg	8,75 TL
17	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Domates Pembe Kg	22,90 TL
18	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Hıyar Badem Paket Kg	24,90 TL
19	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Domates Agrocan Salkım Paket	24,90 TL
20	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Bakla Sakız Kg	21,90 TL
21	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Patlıcan Bostan Kg	32,90 TL
22	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Havuç Siyah Kg	6,95 TL
23	MİGROS	18.03.2022	Meyve, Sebze	Sebze	Biber Acı Sili Kırmızı Kg	39,90 TL

Figure 9.9: Sample data saved in excel after running Migros application

Possible Contributions of The Project

The main purpose of this project is to make it easier to examine food inflation at frequent intervals. For this, a selenium and java based web scraping software has been written that can work on most online supermarket websites. At the end of this web scraping process, the results based on the relevant categories are saved in an excel file according to the current date. As time progresses, an archive will be created according to the data obtained and the historical price of the desired product can be viewed retrospectively. In addition, if desired, the prices of some products can be selected according to the products in the inflation basket of TÜİK and their average can be calculated. In this way, food inflation can be examined more transparently, more accurately, openly and at shorter intervals. Similarly, different inflation baskets can be created retrospectively or currently, and food inflations of different products can be examined. In addition, according to the archive that will be formed over time, the causes and consequences of food inflation can be examined in more detail according to events, so that food inflation can be better understood and different measures can be taken.

Today, most products are sold online. In addition, based on this project, the prices of technological products, furniture products or textile products can be scrapped using selenium with the web scraping method, and the prices of these products can be archived. If desired, the inflation change of these products can also be calculated. When a wider product range archive begins to form, more detailed information about the general inflation of the country can be obtained.

References

[1] Başalak, İ. (2022, January 8). Selenium Nedir ?. Medium. [Internet]. [date accessed 15.11.2022] https://medium.com/@ilkebasalak/selenium-nedir-8c7d908c93e6

[2] Doğan, Ö. (2020, September 7). Selenium Kütüphanesi Nedir? Nasıl Kullanılır? Teknoloji.org. [Internet]. [date accessed 15.11.2022] https://teknoloji.org/selenium-kutuphanesi-nedir-nasil-kullanilir/

[3] Karaçay, D. (2022, December 20). Java Nedir? Etudio. [Internet]. [date accessed 18.11.2022] http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/java/ch02/JavaNedir.htm

[4] Kimmig, Markus; Monperrus, Martin; Mezini, Mira (1 November 2011). "Querying source code with natural language". 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011). Lawrence, KS, USA: IEEE: 376-379

[5] Bilgin, B. E. (2021, December 15). IntelliJ IDEA ve Selenium Webdriver Kullanımı—Örnek Bir Uygulama |. Medium. [Internet]. [date accessed 20.11.2022] https://medium.com/@bebilgin/intellij-idea-ve-selenium-webdriver-kullan%C4%B1m%C4%B1-%C3%B6rnek-bir-uygulama-43889cef6b25

[6] Demirel, A. (2021, December 10). Selenium ile nasıl web scraping (veri kazıma) yapılır ? Medium. [Internet]. [date accessed 01.12.2022] https://medium.com/ahmetdemirel-blog/selenium-ile-nas%C4%B1l-web-scraping-veri-kaz%C4%B1ma-yap%C4%B1l%C4%B1r-697c07511064

[7] Azram, K. (2021, October 4). Selenium vs Scrapy: Which One Should You Choose for Web Scraping? Blazemeter. [Internet]. [date accessed 18.11.2022] https://www.blazemeter.com/blog/scrapy-vs-selenium#selenium

[8] Şen, M. A. E. (2021, December 10). Maven Nedir? - Yazilim VIP. Medium. [Internet]. [date accessed 18.11.2022] https://medium.com/yazilim-vip/bu-yaz%C4%B1n%C4%B1n-amac%C4%B1-maven-ile-siz-okurlar%C4%B1-tan%C4%B1%C5%9Ft%C4%B1rmakt%C4%B1r-aab0f6ff91f4

[9] Karabakla, H. (2021, December 27). Maven nedir ve neden kullanılmalı ? - Sıfırdan İleri Düzeye Java Eğitim Serisi. Medium. [Internet]. [date accessed 21.11.2022] https://medium.com/s%C4%B1f%C4%B1rdan-i%CC%87leri-d%C3%BCzeye-java-e%C4%9Fitim-serisi/maven-nedir-ve-neden-kullan%C4%B1lmal%C4%B1-454fe9cec87c

[10] Ekici, B. (2020, February 11). Xpath Kullanımı. [Internet]. [date accessed 29.12.2022] http://www.barisekici.com/2020/01/19/xpath-kullanimi/

[11] Gulec, G. (2020, May 20). CSS Selector ve Xpath Kullanımları – Yazılım Test Mühendisi Kişisel Blog. [Internet]. [date accessed 29.12.2022] http://gizemgulec.com/2020/05/css-selector-ve-xpath-kullanimlari/

[12] Türkiye Cumhuriyeti Merkez Bankası (2013). ENFLASYON ve FİYAT İSTİKRARI. [Internet]. [date accessed 15.12.2022] https://www.tcmb.gov.tr/wps/wcm/connect/06084069-3751-44a3-ba98-fc5a65b908ba/Enflasyon_FiyatIstikrari.pdf?MOD=AJPERES

Curriculum Vitae

Name Surname : Berkcan Teber

E-mail (1) : berkcanteber@gmail.com

E-mail (2) : teber92@hotmail..com

Education:

2011–2016 Yeditepe University, Dept. of Genetics and Bioengineering, B.Eng.

2013–2016 Anadolu University, Dept. of Business Management, B.B.A.

2021–2023 İzmir Kâtip Çelebi University, Dept. of Software Engineering, M.Eng.

Work Experience:

2015 – 2015 Microbiology Laboratory Technician, Ibrahim Etem MENARINI

2021 – Today Software Tester, Testinium