Department of Systemics

School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

Project Title

Uncovering Market Insights: A Web Scraping Analysis of Stock Performance

Submitted By

Name: Rohit Sharma
SAP ID: 500088022
Roll Number: R2142201937
Batch: B4 (CCVT)

Submitted To

Saurabh Shanu

Project Title

Uncovering Market Insights: A Web Scraping Analysis of Stock Performance

1. Introduction/ Concept Note

Stock market analysis is essential for investors and financial analysts. This helps them in providing insights into performance of various stocks and also look at the market trends. Web scraping is a strong technology that can be used to collect stock market data from various web sources. This project analyse and compare stocks market that uses web scraping to collect and present stock market data. The project focuses on stock performance, stock comparison and market trends using Java and appropriate libraries to develop the web scraping tool. The application will be then deployed on a cloud environment for the reasons mentioned in the document below.

Understanding the trends and patterns of stock market are essential for investors and financial experts as stocks are considered as a very critical part of the economy. Web scraping is a very strong technology that can be used to collect and evaluate a lot of stock market data from multiple sources. With an emphasis on stock performance, market trends, the aim of this project is to create a web scraping tool that can be used to collect and analyse stock market data.

The last stage of the project is to deploy the application on cloud for various reasons. Cloud computing is an essential component of a stock comparison web scraping application because it provides various functionalities like scalability, cost-effectiveness, data storage, accessibility, and security purposes.

The objectives of this project are to:

- Develop a web scraping tool that collects stock market data from various web sources.
- Analyses of stock performance data, including stock prices and volume and their comparison.
- Deployment of the application on cloud.

2. Literature Review

Web scraping is a technique used to retrieve data from various websites. It involves making HTTP requests to the server of a website then downloading the HTML content of a web page, and finally extract the data that we want. The extracted data can be stored in the local system or in a database if the database functionality is provided in the code.

The process of web scraping can be broken down into the following steps:

1. Send an HTTP request to the URL of the web page you want to access.
2. The server responds to the HTTP request by returning the HTML content of the web page.
3. Once we have accessed the HTML content, we have to parse the data.
4. Once we have collected the data, we can store it in a file or database for later use.

- **Applications of Web Scraping**

Web scraping can be used in numerous fields such as data analysis, business intelligence, market research, and data journalism. Web scraping can be used in data analysis to collect massive amounts of data from websites for further analysis of data. Web scraping can be used in business intelligence to track market trends and get competitive intelligence. Web scraping can be used in market research to collect information on consumer behaviour and preferences. Web scraping can be used in data journalism to collect information for investigative reporting.

- **Methodology**:

The methodology for the project would involve the following steps:

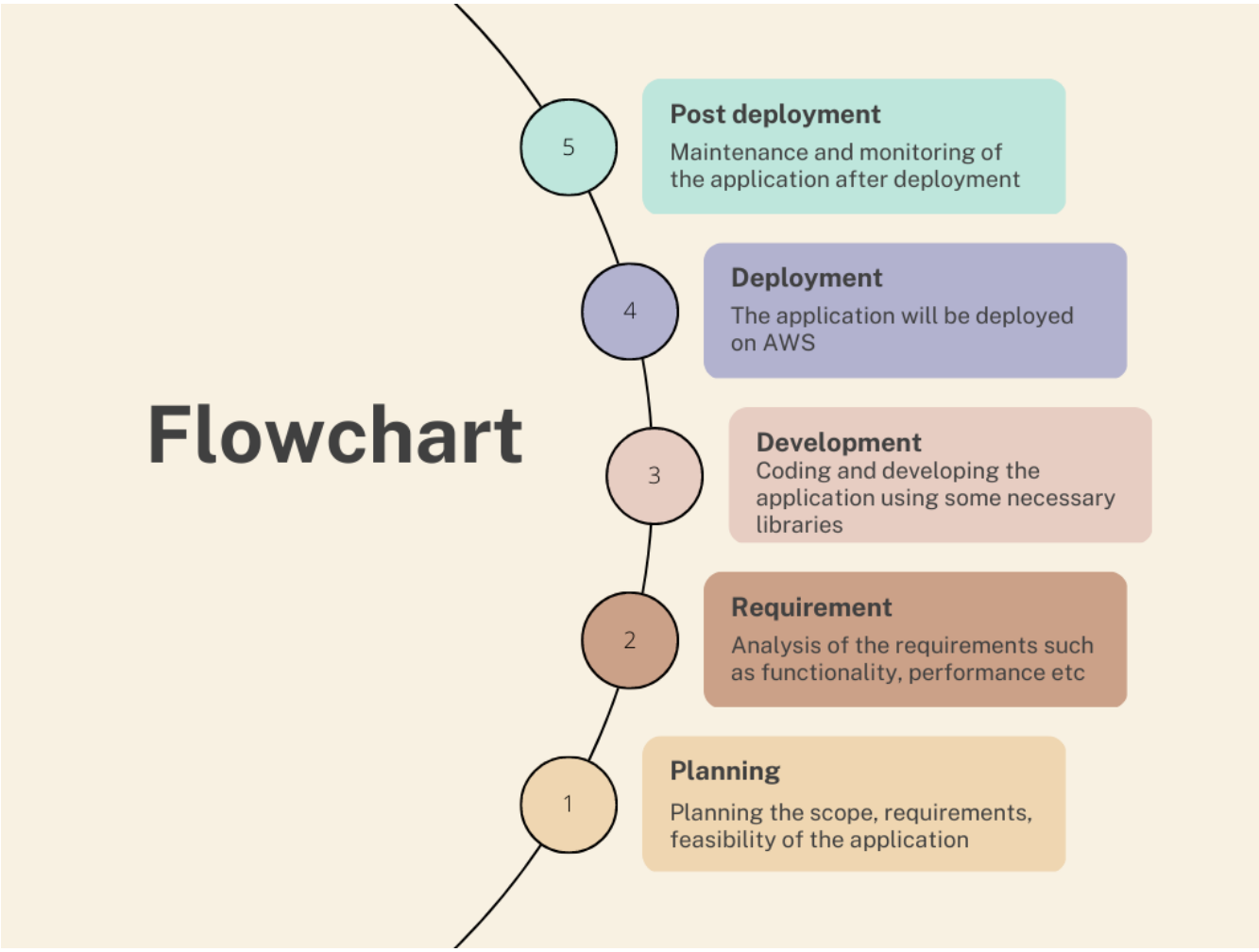
1. Identify the source of data: The first step is to identify the websites that will provide the data related stock market. This could include financial news websites, stock market databases, or other relevant sources.
2. Selecting a Java library for web scraping: There are several Java libraries available for web scraping, such as Jsoup.
3. Develop a web scraper: Use the selected Java library to develop a web scraper that can extract the required data from the identified sources. This involves writing code to navigate to the source (website), locate the required data and extract the data in a suitable format.
4. Analysis and comparison of the data: The Java GUI will reflect the extracted data and other operations on the data such as comparison.

5. Deployment on cloud: The source code files including the JAR file will be bundled up together and the application will be deployed on the cloud to provide the benefits mentioned earlier.

Detailed review on why this application requires cloud support is given below. There are several reasons to deploy this application on cloud like-

- **Scalability**: Cloud computing allows for easy scaling of resources as the need arises, which can be important for real-time web scraping as it may require large amount of processing power and storage capacity.
- **Reliability**: Cloud-based systems are often more reliable than on-premise systems, as they have built-in redundancy and are fault tolerant. This can be important for real-time it web scraping as it requires continuous, reliable access to data.
- **Flexibility**: Cloud computing allows the use of different types of resources, such as virtual machines, containers, and serverless functions, which can be useful for real-time web scraping as it may require different types of resources at different times.
- **Cost-effective:** Cloud computing can be more cost-effective than on-premise systems, as it allows for the use of resources on a pay-as-you-go basis, without the need for large upfront investments.
- **Global accessibility:** Cloud providers like AWS, Azure, GCP have data centres all over the world which means that the web scraping solution can be accessible globally with minimal latency.
- **Security**: Cloud providers offer a variety of security options that can be used to protect data. This is important for real-time web scraping as it may involve sensitive data in some cases.
- **Real-time data availability:** The application requires collection of data from we sources at real-time, for which the cloud platform will be much beneficial to collect the real-time data from the sources.

3. Flowchart


Flowchart
5
Post deployment
Maintenance and monitoring of the application after deployment
4
Deployment
The application will be deployed on AWS
3
Development
Coding and developing the application using some necessary libraries
2
Requirement
Analysis of the requirements such as functionality, performance etc
1
Planning
Planning the scope, requirements, feasibility of the application

4. Architecture Style

Multiple Instruction Multiple Data (MIMD) is a sort of parallel computing architecture in which multiple processors or cores simultaneously execute various instructions on various types of data.
Depending on the needs of the application and the amount of data being processed, a MIMD architecture may be used to construct a stock analysis web scraping application that makes use of parallel computing.
For instance, in a MIMD design, various processors or cores will be responsible for gathering, processing and analysing data from various websites.
So, we can conclude that it is feasible to use the MIMD architectural design to implement the above application where each core/processor has its own function to perform whether to collect data from web sources, for processing or for analysing the data. Here, every processor will communicate with each other to make the entire application perform at an optimum levels.

5. Cloud Platform and Why

To deploy the application on cloud, I will be using the **Microsoft Azure** Platform as it provides several options for deploying a web scraping application:

- Azure Functions: It is a serverless computing platform that allows you to run code snippets in response to HTTP requests. We can easily deploy our code on Azure Functions.
- Azure App Service: It is a fully managed platform for building, deploying and scaling web applications.

Both Azure and AWS are popular cloud computing platforms. They both provide a variety of options and services for deploying and maintaining apps.
Here are a few ways in which I found Azure may be considered better than AWS for cloud deployment:

1. **Strong focus on security and compliance:** We are seeing that most of the world's top organisations are migrating their data to Azure. This is because Azure focuses on security and compliance and includes features such as Azure Active Directory, Azure Key Vault and Azure Security Center to meet the security and compliance requirements.
2. **Global network of data centers:** Azure has a global network of data centers, which ensures low latency and high availability for the applications hosted over it.

3. **Other Microsoft products:** Azure integrates well with other Microsoft products such as Office 365 and Visual Studio Code which are the basic platforms we need to develop the application. This will be very helpful in making the process more streamlined and easier.
4. **Scalability**: Azure provides various services to scale our application with the help of Load Balancer and other functionalities.
5. **Reliability**: As mentioned above, Azure have a strong network of data centers globally and also Azure provides disaster recovery capabilities such as Azure Site Recovery which helps to quickly recover from a disaster. These feature make Azure more reliable.