**Citi Bridge Project**

**Project 2**

**Market Cap based Trade Recommendation System**

Group 6

Mentor : Soumya Mohapatra

Team members :

Sanyukta Bapte

Shweta Barge

Zayeema Massom

Vrushali Lad

Gauri Nandolkar

# Introduction

## Purpose

Market cap based trade recommendation system recommends top 5 stocks of the selected market cap by the user. The user may then decide to buy these stocks or not. Each user will have an account and has to login the system. The user can see his saved stocks the next time he logs in.

## Product Scope

This product is only of NIFTY\_50 Stocks of NSE.

The user cannot register into the system only authentication service is provided.

Our product can only provide login service at this stage

# Overall Description

## Product Perspective

A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

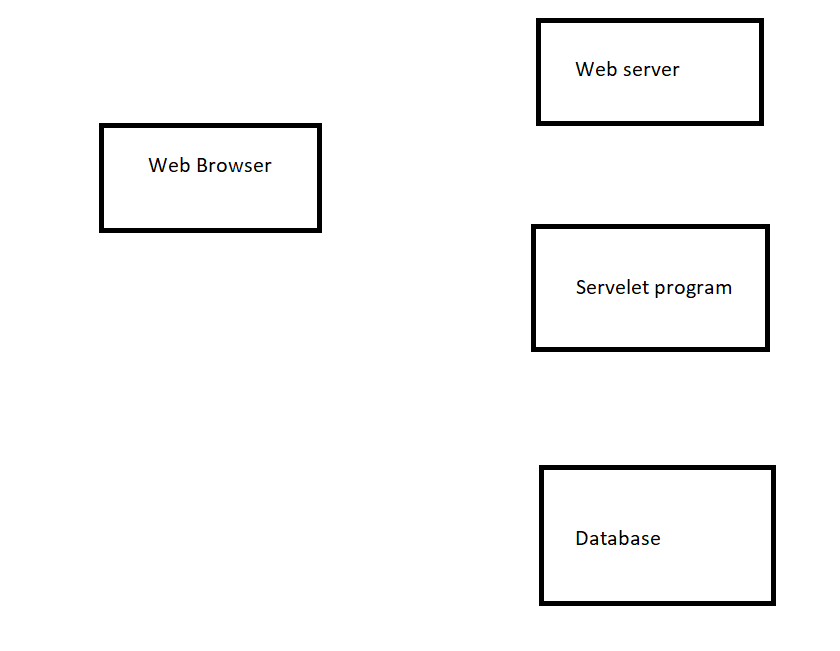
HTTP Response

HTTP Request

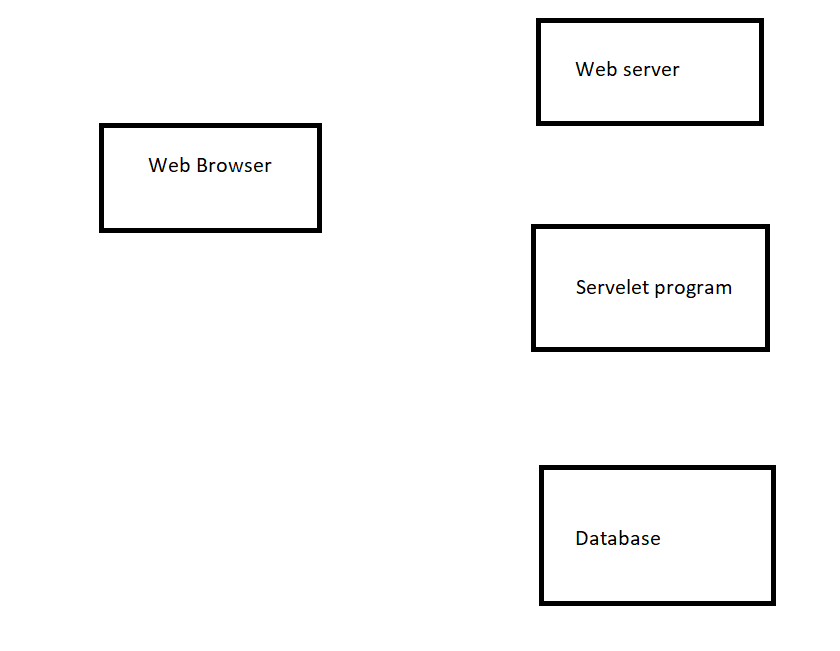
Web Server

Web Browser

Servlet Program



Database



1. Proper JDBC Environment should set-up along with database creation.
2. To do so, download the mysql-connector.jar file from the internet,
3. As it is downloaded, move the jar file to the apache-tomcat server folder,
4. Place the file in **lib** folder present in the apache-tomcat directory.
5. **To start with the basic concept of interfacing:**
   * **Step 1: Creation of Database and Table in MySQL**

As soon as jar file is placed in the folder, create a database and table in MySQL,

use db;

create table user\_data(

user\_id int(11) primary key,

username varchar(20),

password varchar(20)

);

create table user\_saved\_stock(

user\_id int(11),

stocksymbol varchar(10),

stockprice decimal(10,2),

stockquantity int(5),

FOREIGN KEY(user\_id) REFERENCES user\_data(user\_id),

FOREIGN KEY(stocksymbol) REFERENCES stock\_for\_cap(stocksymbol)

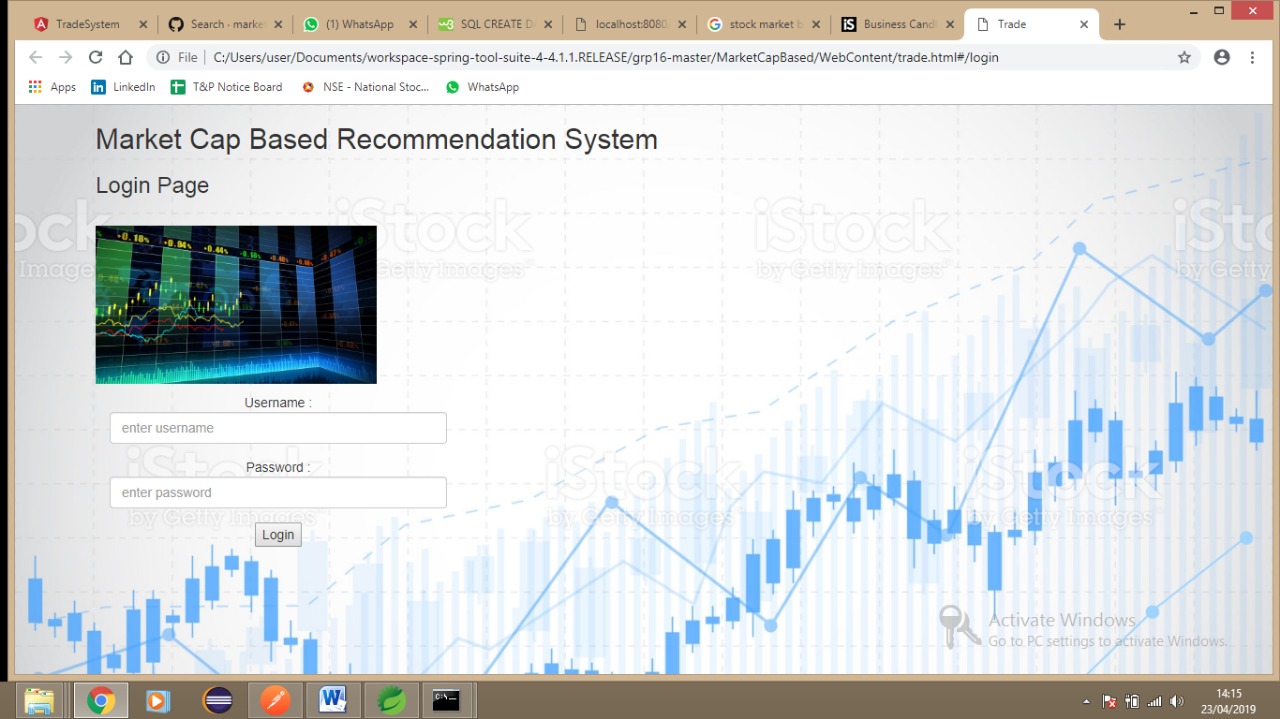
);

create table stock\_for\_Cap(

stocksymbol varchar(10) primary key,

marketcap varchar(20)

);

* + **Step 2: Implementation of required Web-pages**  
    Create a form in HTML file, where take all the inputs required to insert data into the database. Specify the servlet name in it, with the POST method as security is important aspects in database connectivity.
  + **Output:**  
    

Submit the data (with validation) as all the required data are inserted.

* + **Step 3: Creation of Java Servlet program with JDBC Connection**

To create a JDBC Connection steps are

* + 1. Import all the packages
    2. Register the JDBC Driver
    3. Open a connection
    4. Execute the query, and retrieve the result
    5. Clean up the JDBC Environment

Create a separate class to create a connection of database, as it is a lame process to writing the same code snippet in all the program. Create a .java file which returns a Connection object.

package com.citi.trade.dao;

import java.sql.Connection;

import java.sql.DriverManager;

public class DBConnection {

private static final String MYSQL\_JDBC\_DRIVER = "com.mysql.jdbc.Driver";

private static final String dbURL = "jdbc:mysql://localhost:3306/db";

private static final String username = "root";

private static final String password = "root";

/\*\*

\* This method creates mysql connection object

\* @return connection object

\*/

public static Connection createConnection() {

Connection connection = null;

try {

Class.forName(MYSQL\_JDBC\_DRIVER);

connection = DriverManager.getConnection(dbURL, username, password);

} catch (Exception e) {

e.printStackTrace();

}

return connection;

}

}

|  |
| --- |
|  |

* + **Step 4: To use this class method, create an object in Java Servlet program**

(Authentication Servlet code)

* + **Step 5: Get the data from the HTML file**

To get the data from the HTML file, the request object is used which calls [getParameter()](https://www.geeksforgeeks.org/getparameter-passing-data-from-client-to-jsp/) Method to fetch the data from the channel. After successful insertion, the writer object is created to display a success message.

After insertion operation (for stocks) from Servlet, data will be reflected in MySQL Database

## Product Functions

1. The user can login through the website

2. Authentication is done

3. User’s data is saved in MySql database

## User Classes and Characteristics

Traders, Investors, are user classes of this product.

## Operating Environment

JDK 8.0 or above, Java IDE, Bootstrap, MySql database server must exist on the system

# References

<https://www.geeksforgeeks.org/introduction-java-servlets/>

<https://www.geeksforgeeks.org/java-servlet-and-jdbc-example-insert-data-in-mysql/>

<https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/>

<https://youtu.be/47xNBNd-LLI>

<https://spring.io/guides/gs/rest-service/>

<https://spring.io/guides/gs/accessing-data-mysql/>

<https://spring.io/projects/spring-data-mongodb>

<https://www.nseindia.com/live_market/dynaContent/live_watch/equities_stock_watch.htm>

<https://github.com/>