TRIBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A PROJECT

ON

Portfolio Management System

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE

CT 610 DATABASE MANAGEMENT SYSTEMS

SUBMITTED BY:

MADHU SUDAN OJHA	THA075BEI019
REWAN GAUTAM	THA075BEI032
SHOVIT NEPAL	THA075BEI040
SUMAN KARKI	THA075BEI042

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

DECLARATION

We, hereby declare that the project work report entitled “Portfolio Management System” submitted for the partial fulfillment of the requirements for the course of CT 610 Database Management Systems is our original work and the project work report has not been formed on the basis for the award of any degree, diploma, or other similar titles.

Signature: 

Name of Student: Rewan Gautam

Signature: 

Name of Student: Madhu Sudan Ojha

Signature: 

Name of Student: Shovit Nepal

Signature: 

Name of Student: Suman Karki

ACKNOWLEDGEMENTS

It is a matter of pleasure of acknowledgement by indebtedness to Mrs. Rama Bastola for providing us the opportunity to work on this project.

The experience of doing this project will surely enrich our technical and teamwork skills to a great extent.

ABSTRACT

In present scenario of our country, investors on the field of share market are increasing day by day. With hiking shares, people are investing their capital so that it profits them in the future reference. Likewise, people are crazily indulged in this field which may create messy circumstances for managing huge bundles of shares that they hold. Also, in this busy era people have no time to analyze fundamental, technical details and the news related to that scrip. Portfolio management system is in rescue for this task.

Key Words:

Share market, NEPSE, portfolio management, etc

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background

Share Market is the aggregation of buyers and sellers of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately, such as shares of private companies which are sold to investors through equity crowdfunding platforms. Investment in the stock market is most often done via stockbrokerages and electronic trading platforms which in TMS (Trade Management System) in Nepal. Investment is usually made with an investment strategy in mind.

A portfolio is a collection of financial investments like stocks, bonds, commodities, cash, and cash equivalents, as well as their fund counterparts. In our project, we will often use portfolio to define only investments in stocks. Portfolio management is the art and science of selecting and overseeing a group of stocks that meet the long-term financial objectives and risk tolerance of a client, a company, or an institution. Fundamental analysis determines a stock's fair value by examining the related company earning's with respect to investment. Fundamental analysis depends upon fundamental report. Fundamental analysts search for stocks that are currently trading at prices that are higher or lower than their real value. Technical analysis is an analysis methodology for forecasting the direction of prices through the study of market data.

1.2 Problem Statement

A trader must deal with several transactions and keep track of own profit or loss. The problem in current portfolio management system that exists is that they do not have many features. They do not have option to see fundamental and technical details of our current holdings easily. Also, news plays a vital role in stock market, and we have not seen any stock management system with options to view news based on holdings or even watchlist. So, portfolio management system can help overcome this flaw.

1.3 Objectives

The major objectives of the project are:

- To create a portfolio for each user
- To add transaction and update the portfolio accordingly.

- To display fundamental and technical report consisting of metrics like EPS, P/E Ratio, ROE, Volume, RSI, etc.
- To suggest the user to buy or sell according to the fundamental and technical report.
- To display net profit and loss for user

1.4 Applications

Anyone who is an investor or trader in share market can use this project. They can explore their share values, check if they are fundamentally and technically strong. They can also look at beautiful graphs to know more about their holdings.

1.5 Project Features

The proposed features of this project are:

- Company profile to see more about them.
- Current company price
- Transaction history
- Fundamental, Technical and Dividend history report
- News related to different company.
- Watchlist and holdings
- Easy user interface

1.6 Feasibility Analysis

1.6.1 Economic Feasibility

As it is only the concept of software, and the hardware requires is only the PC that is present everywhere nowadays. And it can be installed on custom designed computer for only the program above and can be installed on a very affordable price.

1.6.2 Technical Feasibility

As the database will be made in SQL, program is written in python, HTML and CSS language, any computer with minimum specifications can be used to develop this program.

1.6.3 Operational Feasibility

This project is user friendly, and all the instructions are carefully made so that no mistakes are made. Thus, this program can be operated by a person with general knowledge for the device.

1.7 System Requirement

1.7.1 Software Requirement

The whole system requires coding which will be written in various languages like SQL, Python, HTML and CSS. Under the hood, there will be sqlite3 library to connect the SQL query and database to the backend framework which is Flask. To write SQL queries, we will be using MySQL Workbench and for the frontend and backend we will be using Visual Studio Code. Any browser is also required as a software for frontend.

1.7.2 Hardware Requirement

Any normal PC is preferable for running the program.

.

CHAPTER 2: LITERATURE REVIEW

2.1 Database Management System

Database is an organized collection of related information so that it can easily be accessed, managed, and updated. It is a repository of logically linked and similar data. Database Management System (DBMS) is a collection of interrelated data and a set of programs to access those data. DBMS provides a way to store and retrieve database. It is designed to manage large bodies of information.

Management of data involves defining structure for storage of information, providing mechanism for manipulation of information, data security, data integrity and consistency.
Examples: MySQL, PostgreSQL, Microsoft Access, SQL Server, etc.

2.2 Entity Relationship Model

Entity Relationship (ER) modeling is a logical database design modeling technique. After the business requirements and data requirements are gathered, development of logical data model is started describing the data involved in the real word enterprise in terms of objects and their relationships. It is widely used to develop an initial database design. It was initially purposed by Dr Peter Chen in 1976.

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.

2.3 Structured Query Language

Structured Query Language (SQL) is a standard language for accessing and manipulating databases. [1] It includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI (American National Standards Institute) standard language, but there are many different versions of the SQL language. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres, and SQL Server use SQL as their standard database language. SQL is widely popular because it offers the following advantages −

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

2.4 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.[2] Python is easy and simple where it takes no time to make a program from pseudo-code. Python's popularity these days can be examined with its large number of modules and packages encouraging code reusability.

Python with it's easy to use syntax and high-level abstraction helps us to integrate our SQL program to front end. Python also provides several frameworks and libraries for quick production. Django and Flaks are two popular python backend web frameworks. We will be using flask for our project.

2.5 MySQL

MySQL is an open source relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. [3] The most common use for MySQL however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store. In association with a scripting language such as PHP or Perl (both offered on our hosting accounts) it is possible to create websites which will interact in real-time with a MySQL database to rapidly display categorised and searchable information to a website user.

2.6 HTML

HTML stands for Hyper Text Markup Language. It is the standard markup language for creating Web pages which describes the structure of a Web page. [4] It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. It consists of a series of elements. HTML elements tell the browser how to display the content. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

2.7 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and

variations in display for different devices and screen sizes as well as a variety of other effects. [5]

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

2.8 Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. [6] However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

2.9 Bootstrap

Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. [7] The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

CHAPTER 3: METHODOLOGY

For building the project, we planned with following methodology.

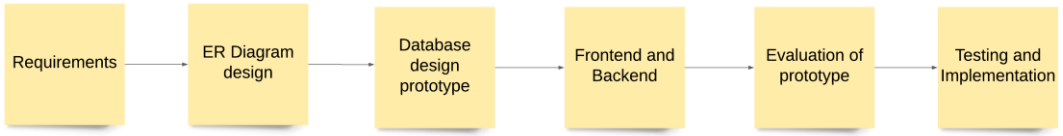


Fig: System design methodology

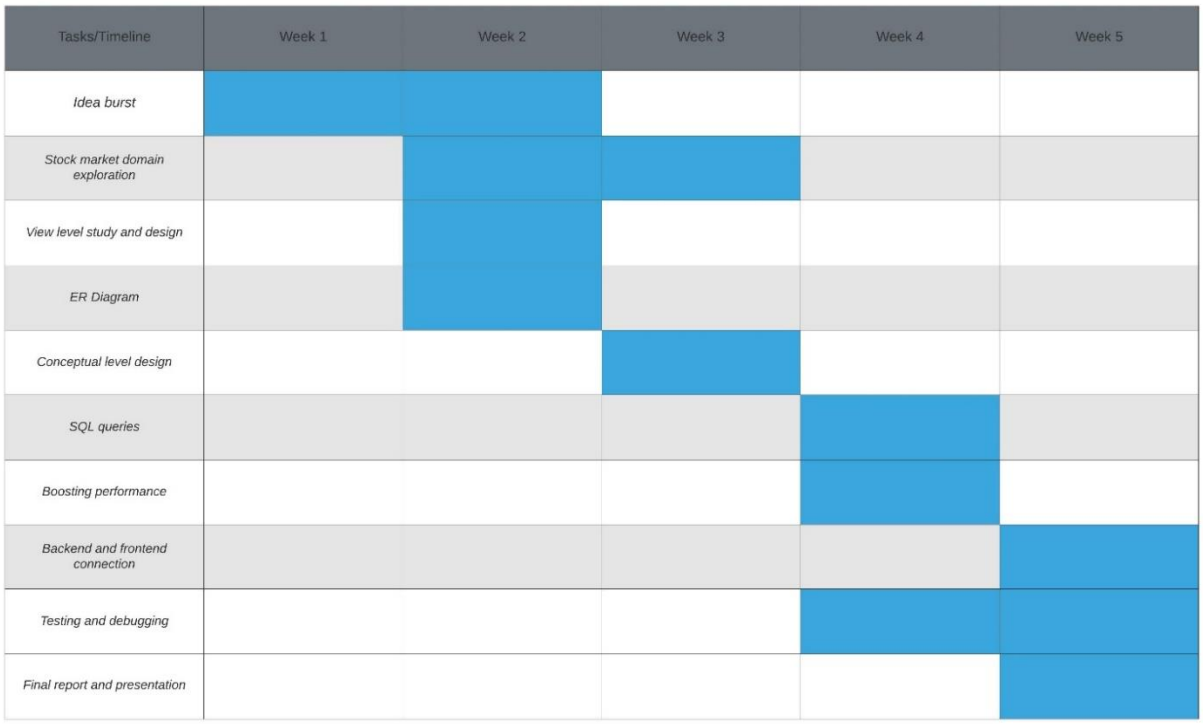


Fig: Gantt chart

The Gantt chart above gives the visualization of our task and the time to complete it. The plan was to complete the project within a month. In the initial stage, we planned to research what can we do and how we can do it. We also needed to grab domain knowledge of Stock market. Afterwards, we made a view level design with ER diagram and created conceptual level design. Afterwards, we created a simple database design prototype in MySQL using the details in the ER diagram and displayed the report in front end with the help of python flask framework. The prototype was evaluated to meet the final polished requirements. At the end, several testing and debugging were performed.

The ER diagram for this project is shown below:

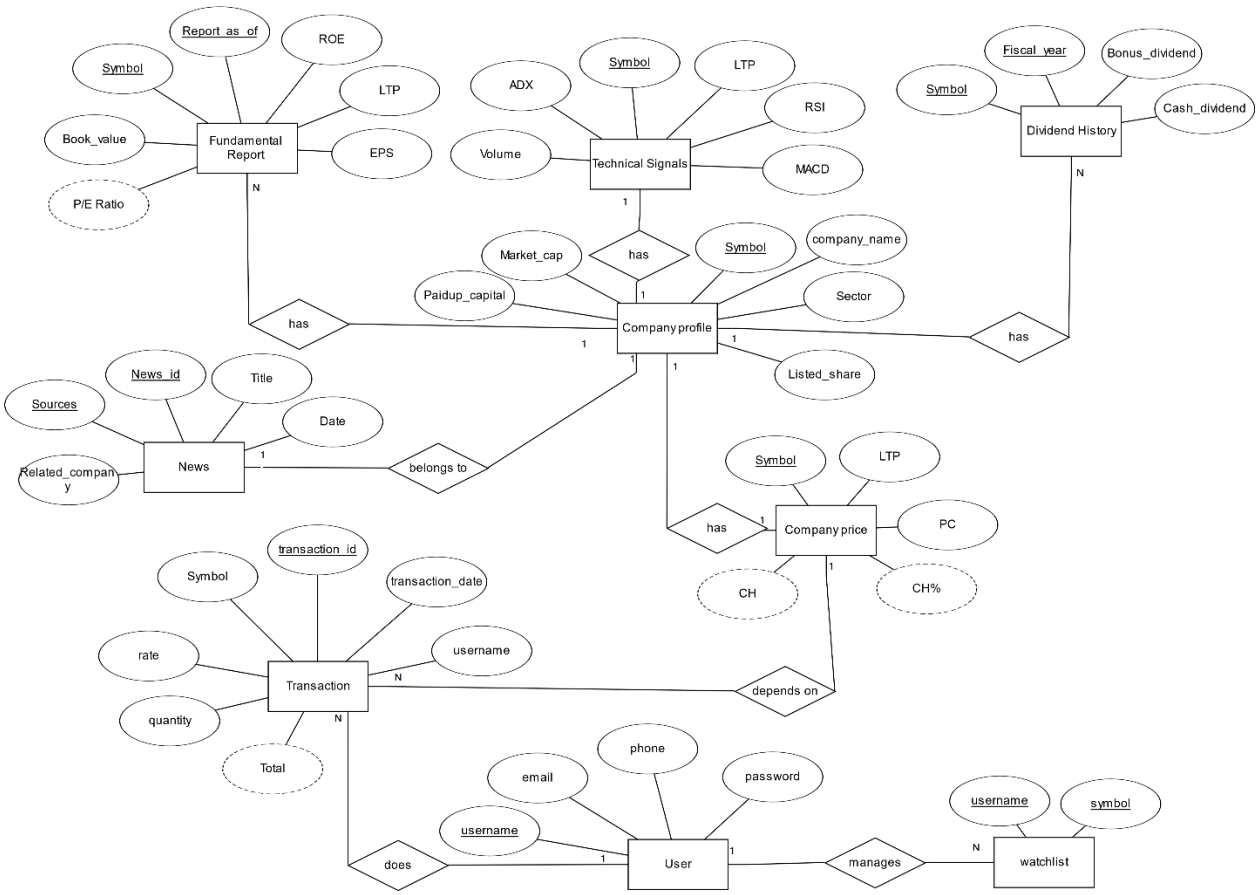


Figure: ER Diagram

We have created nine entities for this project. All the entities and their attributes are listed below:

- Fundamental Report: Symbol, Report_as_of, LTP, EPS, P/E Ratio, ROE, Book_value
- Technical Signals: Symbol, LTP, RSI, MACD, Volume, ADX
- Dividend History: Symbol, Fiscal_year, Bonus_dividend, Cash_dividend
- News: News_id, Title, Date, Sources, Related_company
- Transaction: Transaction_id,Transaction Date, username, Symbol, Quantity, Rate, Total
- Company price: Symbol, LTP, PC, CH, CH %
- Watchlist: username, symbol
- Company profile: Symbol, company_name, Sector, Listed_share, Paidup_capital, Market_cap
- User: username, username, email, phone

CHAPTER 4: IMPLEMENTATION AND RESULT

4.1 Implementation details

To build this project, we used HTML, CSS, JavaScript in the front end, MySQL in the backend for the database and python's flask framework to make an interface between front end and the database.

Some major code snippets are:

4.1.1 Function to find total amount (amount + broker commission + SEBON fee + DP charge) excluding capital gain tax

```
DELIMITER $$
CREATE FUNCTION getTotal(
    total float
)
RETURNS float
DETERMINISTIC
BEGIN
    DECLARE total_converted float;
    DECLARE comm float;
    DECLARE ptotal float;

    -- If sell, make the total positive to calculate commission later
    IF total<0 THEN
        SET ptotal = -total;
    ELSE
        SET ptotal = total;
    END IF;

    -- Commission is same for both buy and sell
    IF ptotal > 500000 THEN
        SET comm = (0.34/100)*ptotal;
    ELSEIF ptotal > 50000 THEN
        SET comm = (0.37/100)*ptotal;
    ELSEIF ptotal > 2500 THEN
        SET comm = (0.4/100)*ptotal;
    ELSEIF ptotal > 100 THEN
        SET comm =  10;
    END IF;

    -- If sell conditions
    IF total < 0 THEN
        set total = -total;
        set total_converted = total - (comm + total*(0.015/100) + 25);
        RETURN total_converted;
    END IF;

    SET total_converted = total + comm + 25 + (0.015/100)*total;
    RETURN (total_converted);
END$$
DELIMITER ;
```

4.1.2 Function to find profit loss with capital gain tax deducted

```
DELIMITER $$
CREATE FUNCTION capGain(
    total float,
    trans_date date
)
RETURNS float
DETERMINISTIC
BEGIN
    IF total<0 THEN
        RETURN total;
    ELSEIF datediff(Now(), trans_date)<365 THEN
        SET total = total - 0.075*total;
    ELSE
        SET total = total - 0.05*total;
    END IF;
    RETURN total;
END$$
DELIMITER ;
```

4.1.3 Stored procedure for portfolio (holdings with profit/loss)

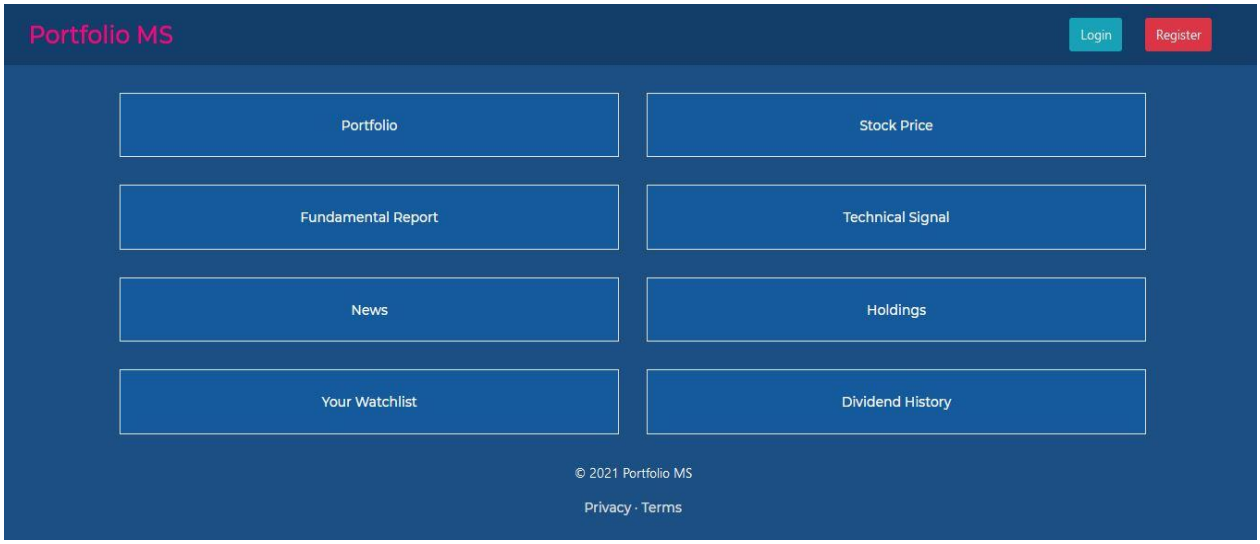
```
DELIMITER $$
CREATE PROCEDURE portfolio(in username varchar(30))
BEGIN
select symbol, sum(quantity) as quantity, LTP
,round((getTotal(-sum(quantity)*LTP)), 2) as current_value,
capGain(round((getTotal(-sum(quantity)*LTP)) - (getTotal(sum(quantity)*rate)), 2), transaction_date) as profit_loss
from transaction_history T
natural join company_price C
where username = username
group by symbol;
END$$
DELIMITER ;
```

4.1.4 Query for best companies to invest and it's implementation using flask

```
    query_suggestions = '''select symbol, EPS, ROE, book_value, rsi, adx, pe_ratio, macd from company_price
natural join fundamental_averaged
natural join technical_signals
natural join company_profile
where
EPS>25 and roe>13 and
book_value > 100 and
rsi>50 and adx >23 and
pe_ratio < 35 and
macd = 'bull'
order by symbol;
'''
    cur.execute(query_suggestions)
    suggestions = cur.fetchall()
```

4.2 Result analysis

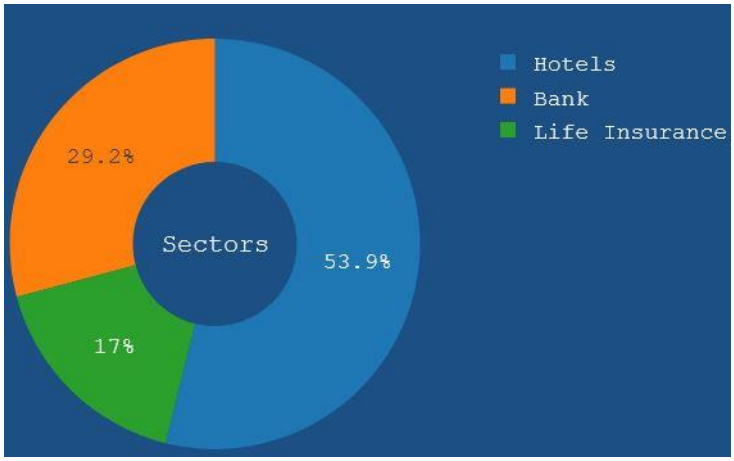
4.2.1 Front page



4.2.2 Best companies to invest: EPS>25, ROE>13, Book Value>100, RSI>50, ADX>23, P/E Ratio<35, MACD = bull

Symbol	EPS	ROE	Book Value	RSI	ADX	P/E Ratio	MACD
BIH	35.5	20.5	210.0	75.0	44.0	22.2	bull
ELEX	32.5	14.0	179.0	70.0	30.0	31.14	bull
NIL	208.0	52.5	305.0	50.5	40.0	27.89	bull

4.2.3 Sector wise total current value distribution



4.2.4 Holdings: Inclusion of Broker commission, Sebon fee, DP charge and Capital Gain Tax(with latest rule)

Symbol	Quantity	LTP	Total	Profit/Loss
HEX	100	1222.3	121734.42	1153.86
HIH	35	1500.5	52290.31	245.939
LEC	20	400.0	7941.8	-518.06
BIH	120	788.0	94170.95	-476.95
LSL	55	1000.0	54763.25	-528.71

4.2.5 Company Profile

Symbol	Company Name	Sector	Market Capitalization	Paidup Capital
BIH	Bank of Itahari	Bank	62323233232	623321321321
ELEX	Nepal Electronics Bank	Bank	32323233232	323321321321
HEX	Hotel Electronics	Hotels	82323233232	823321321321
HIH	Hotel Itahari	Hotels	12323233232	123321321321
KBL	Kumari Bank	Bank	1000000000	21212121221
LEC	Libery Energy	Hydropower	63233232	61321321
LSL	Life Saver Limited	Life Insurance	23233232	21321321
NBL	Nepal Bank Limited	Bank	532323233232	5323321321321
NEPP	Nepal Power	Hydropower	102323233232	10323321321321
NIL	Nepal Insurance Limited	Life Insurance	123232332	131321321

4.2.6 News

Date Posted	Title	Related Company	Related Sector	Sources
2021-07-10	CEO of Nepal Insurance Limited resigns immediately	NIL	Life Insurance	ekantipur,myRepublica
2021-07-05	Hotel itahari expands it's area	HIH	Hotels	itaharinews
2021-07-01	Kumari Bank announces right share of 1:1	KBL	Bank	myRepublica
2021-07-04	Liberty energy to test production soon	LEC	Hydropower	merokhabar

CHAPTER 5: CONCLUSION

To sum up, the project was completed with the coordination of our team and help from various sources and under the supervision of our great teacher. This project took us about 5 weeks to complete and be fully working.

CHAPTER 6: FURTHER WORKS

This project is not perfect as it has various aspects of improvement that can be done to meet the needs as required. Some of the areas might be as follows:

- Although there was a frontend interface for user registration, it was not linked with backend to make this feature work. So, it can be solved.
- Addition of admin panel
- We could add technical analysis graphs as technical analysis using graph has been very useful these days
- We can make the UI design better
- We can try using API to fetch real data because we tried with dummy data
- Storing database in local server is not always possible. We want to scale this project by storing database in cloud server

REFERENCES

[1] Tutorialspoint.com. 2021. *SQL Tutorial - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/sql/index.htm> [Accessed 10 July 2021].

[2] Python.org. 2020. *Welcome to Python.Org*. [online] Available at: <https://www.python.org/> [Accessed 11 July 2021].

[3] "MySQL - Wikipedia", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/MySQL. [Accessed: 12- Jul- 2021]

[4] "HTML - Wikipedia", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/HTML. [Accessed: 12- Jul- 2021]

[5] "CSS - Wikipedia", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/CSS. [Accessed: 12- Jul- 2021]

[6] "Flask (web framework) - Wikipedia", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/ Flask_(web_framework). [Accessed: 13- Jul- 2021]

[7] "Bootstrap 3 Tutorial", *W3schools.com*, 2021. [Online]. Available: https://www.w3schools.com/bootstrap/. [Accessed: 14- Jul- 2021]

APPENDIX

Project code link:

GitHub link:

https://github.com/gtmray/Portfolio-Management-System

or

Google Drive link:

https://drive.google.com/drive/folders/1Kr6E8-lpgY7Q-u-LJNUY2kbwIS-yLJLU?usp=sharing

Code for only MySQL file:

```
create database portfolio;
use portfolio;

-- For debugging
drop table company_profile;
drop table company_price;
drop table fundamental_report;
drop table technical_signals;
drop table dividend_history;
drop table news;
drop table user_profile;
drop table watchlist;
drop table transaction_history;

show tables;
desc company_profile;

create table company_profile
(
symbol varchar(6),
company_name varchar(100) NOT NULL,
sector varchar(20) NOT NULL,
market_cap bigint NOT NULL,
paidup_capital bigint NOT NULL,
primary key(symbol)
);

create table company_price
(
symbol varchar(6),
LTP float NOT NULL,
PC float NOT NULL,
primary key(symbol),
foreign key(symbol) references company_profile(symbol)
);

create table fundamental_report
(
symbol varchar(6),
report_as_of varchar(10),
EPS float NOT NULL,
ROE float NOT NULL,
book_value float NOT NULL,
primary key (symbol, report_as_of),
foreign key (symbol) references company_profile(symbol)
);
```

```
create table technical_signals
(
symbol varchar(6),
LTP float,
RSI float NOT NULL,
volume float NOT NULL,
ADX float NOT NULL,
MACD varchar(4) NOT NULL,
primary key (symbol),
foreign key (symbol) references company_profile(symbol)
);

create table dividend_history
(
symbol varchar(6),
fiscal_year varchar(6),
bonus_dividend float,
cash_dividend float,
primary key(symbol, fiscal_year),
foreign key(symbol) references company_profile(symbol)
);

create table news
(
news_id int auto_increment,
title varchar(200) NOT NULL,
date_of_news date NOT NULL,
related_company varchar(6),
sources varchar(20),
primary key(news_id, sources),
foreign key(related_company) references company_profile(symbol)
);

create table user_profile
(
username varchar(30),
email varchar(60) UNIQUE NOT NULL,
phone bigint UNIQUE NOT NULL,
user_password varchar(224),
primary key(username)
);

create table  watchlist
(
username varchar(30),
symbol varchar(6),
primary key(username, symbol),
foreign key(username) references user_profile(username),
foreign key(symbol) references company_profile(symbol)
```

```
);

create table transaction_history
(
transaction_id int auto_increment,
username varchar(30) NOT NULL,
symbol varchar(6) NOT NULL,
transaction_date datetime NOT NULL,
quantity int NOT NULL,
rate float NOT NULL,
primary key(transaction_id),
foreign key(symbol) references company_profile(symbol),
foreign key(username) references user_profile(username)
);

select * from transaction_history;

-- Data entry

insert into user_profile values
('rewan', 'uni.rayone@gmail.com', 9800000001, sha2('rewan123', 224)),
('mahesh', 'uni@gmail.com', 9800000002, sha2('arewan123', 224)),
('suman', 'uni1.rayone@gmail.com', 9800000003, sha2('rewan12345', 224)),
('madhu', 'uni2.rayone@gmail.com', 9800000004, sha2('arewan123', 224)),
('sobit', 'uni3.rayone@gmail.com', 9800000005, sha2('brewan123', 224)),
('ray', 'uni4.rayone@gmail.com', 9800000006, sha2('crewan123', 224)),
('rayone', 'uni5.rayone@gmail.com', 9800000007, sha2('drewan123', 224)),
('ravi', 'uni6.rayone@gmail.com', 9800000008, sha2('erewan123', 224)),
('michael', 'uni7.rayone@gmail.com', 9800000009, sha2('frewan123', 224)),
('hari', 'uni8.rayone@gmail.com', 9811111111, sha2('arewan123', 224)),
('madan', 'uni10.rayone@gmail.com', 9800000010, sha2('rfewan123', 224)),
('sandeep', 'uni11.rayone@gmail.com', 9800000011, sha2('frewan123', 224)),
('surya', 'tha0751@gmail.com', 9860000014, sha2('arewan123', 224)),
('vai', 'tha0752@gmail.com', 9860000013, sha2('wrewan123', 224)),
('gtm', 'tha075@gmail.com', 9860000012, sha2('erewan123', 224));

insert into company_profile values
('KBL', 'Kumari Bank', 'Bank', 1000000000, 21212121221),
('NIL', 'Nepal Insurance Limited', 'Life Insurance', 123232332, 131321321),
('LEC', 'Libery Energy', 'Hydropower', 63233232, 61321321),
('ELEX', 'Nepal Electronics Bank', 'Bank', 32323233232, 323321321321),
('NEPP', 'Nepal Power', 'Hydropower', 102323233232, 10323321321321),
('LSL', 'Life Saver Limited', 'Life Insurance', 23233232, 21321321),
('NBL', 'Nepal Bank Limited', 'Bank', 532323233232, 5323321321321),
('HEX', 'Hotel Electronics', 'Hotels', 82323233232, 823321321321),
('HIH', 'Hotel Itahari', 'Hotels', 12323233232, 123321321321),
('BIH', 'Bank of Itahari', 'Bank', 62323233232, 623321321321);

insert into company_price (symbol, LTP, PC) values
('KBL', 500, 470),
```

```
('NIL', 5800, 6000),
('LEC', 400, 410),
('ELEX', 1010, 1000),
('NEPP', 500, 480),
('LSL', 1000, 1040),
('NBL', 600, 580.5),
('HEX', 1222.3, 1220),
('HIH', 1500.5, 1499.4),
('BIH', 788, 777);

insert into fundamental_report(symbol, report_as_of, EPS, ROE, book_value) values
('KBL', '77/78_q3', 20.5, 11.97, 120),
('KBL', '77/78_q2', 19.5, 10, 110),
('NIL', '77/78_q3', 205, 50, 300),
('NIL', '77/78_q2', 211, 55, 310),
('LEC', '77/78_q3', 8, 4, 90),
('LEC', '77/78_q2', 7.5, 3.5, 88),
('ELEX', '77/78_q3', 34, 15, 180),
('ELEX', '77/78_q2', 31, 13, 178),
('NEPP', '77/78_q3', 21, 12, 119),
('NEPP', '77/78_q2', 20, 11, 118),
('LSL', '77/78_q3', 30, 12, 170),
('LSL', '77/78_q2', 35.4, 13, 180.5),
('NBL', '77/78_q3', 22, 13, 120),
('NBL', '77/78_q2', 21, 12, 117),
('HEX', '77/78_q3', 50, 15, 200),
('HEX', '77/78_q2', 48, 14, 199),
('HIH', '77/78_q3', 60, 20, 220),
('HIH', '77/78_q2', 55, 18, 200),
('BIH', '77/78_q3', 36, 20, 220),
('BIH', '77/78_q2', 35, 21, 200);

insert into technical_signals(symbol, RSI, volume, ADX, MACD) values
('KBL', 65.1, 451000, 33.3, 'bull'),
('NIL', 50.5, 100000, 40, 'bull'),
('LEC', 20, 12344, 15, 'bear'),
('ELEX', 70, 1200000, 30, 'bull'),
('NEPP', 45, 212000, 16.5, 'bull'),
('LSL', 53.4, 15312, 25.29, 'bull'),
('NBL', 66.41, 406121, 34.66, 'bull'),
('HEX', 40.2, 34000, 40, 'side'),
('HIH', 35, 120000, 30, 'side'),
('BIH', 75, 335000, 44, 'bull');

-- Updating LTP values in technical_signals
UPDATE technical_signals A
INNER JOIN company_price B ON A.symbol = B.symbol
SET A.LTP = B.LTP
WHERE A.symbol = B.symbol;
```

```
insert into dividend_history values
('KBL', '76/77', 5, 10),
('KBL', '75/76', 4, 11),
('NIL', '76/77', 10, 15),
('NIL', '75/76', 10, 13),
('LEC', '76/77', 0, 0),
('LEC', '75/76', 0, 0),
('ELEX', '76/77', 20, 10),
('ELEX', '75/76', 14, 10),
('NEPP', '76/77', 0, 0),
('NEPP', '75/76', 0, 0),
('LSL', '76/77', 5, 10),
('LSL', '75/76', 5, 10),
('NBL', '76/77', 11, 5),
('NBL', '75/76', 11, 0),
('HEX', '76/77', 0, 0),
('HEX', '75/76', 0, 0),
('HIH', '76/77', 0, 0),
('HIH', '75/76', 0, 0),
('BIH', '76/77', 20, 25),
('BIH', '75/76', 15, 20);

insert into watchlist values
('rewan', 'KBL'),
('rewan', 'HEX'),
('rewan', 'HIH'),
('rewan', 'BIH'),
('mahesh', 'HEX'),
('mahesh', 'ELEX'),
('mahesh', 'LEC'),
('suman', 'NEPP'),
('suman', 'LSL'),
('madhu', 'ELEX'),
('madhu', 'HEX'),
('madhu', 'NBL'),
('sobit', 'HEX'),
('sobit', 'LEC'),
('rayone','HIH');

insert into news(news_id, title, sources, date_of_news, related_company) values
(1, 'Kumari Bank announces right share of 1:1', 'myRepublica', '2021-07-01', 'KBL'),
(2, 'Liberty energy to test production soon', 'merokhabar', '2021-07-04', 'LEC'),
(3, "Hotel itahari expands it's area", 'itaharinews', '2021-07-05', 'HIH'),
(4, "CEO of Nepal Insurance Limited resigns immediately", 'ekantipur', '2021-07-10',
'NIL'),
(4, "CEO of Nepal Insurance Limited resigns immediately", 'myRepublica', '2021-07-
10', 'NIL');

insert into transaction_history(username, symbol, transaction_date, quantity, rate)
values
```

```
('rewan', 'HEX', '2021-07-01', 100, 1200),
('rewan', 'HIH', '2021-07-02', 55, 1480),
('rewan', 'HIH', '2021-07-06', -20, 1500),
('suman', 'LEC', '2021-07-10', 10, 420),
('suman', 'LEC', '2021-07-15', 10, 410),
('rewan', 'BIH', '2021-07-20', 120, 785.5),
('rewan', 'LSL', '2021-07-20', 55, 1001);

-- Holdings
Create view holdings_view as
select username, symbol, sum(quantity) as quantity  from transaction_history
group by username, symbol;

-- Holdings with LTP and current value for user rewan
select A.symbol, A.quantity, B.LTP, round(A.quantity*B.LTP, 2) as current_value
from holdings_view A
inner join company_price B
on A.symbol = B.symbol
where username = 'rewan';

-- Fundamental report
Create view fundamental_averaged as
SELECT F.symbol, LTP, round(avg(EPS), 2) as EPS, round(avg(ROE), 2) as ROE,
round(avg(book_value), 2) AS book_value, round(avg(LTP/EPS), 2) AS pe_ratio
FROM fundamental_report F
INNER JOIN company_price C
on F.symbol = C.symbol
group by(Symbol);

select * from  fundamental_averaged;

-- Fundamental report of certain company without averaging
select F.symbol, report_as_of, LTP, eps, roe, book_value, round(LTP/eps, 2) as
pe_ratio
from fundamental_report F
inner join company_price C
on F.symbol = C.symbol
where F.symbol = 'BIH';

-- Technical report
select A.symbol, sector, LTP, volume, RSI, ADX, MACD from technical_signals A
left join company_profile B
on A.symbol = B.symbol
order by (A.symbol);

-- Company profile
select * from company_profile
order by(symbol);

-- Company price
```

```
SELECT symbol, LTP, PC, round((LTP-PC), 2) as CH, round(((LTP-PC)/PC)*100,
2) AS CH_percent FROM company_price
order by symbol;

-- Dividend
select * from dividend_history;

-- Portfolio system

-- Certain user portfolio
select *
from holdings_view A
left outer join company_price B on A.symbol = B.symbol
left outer join fundamental_averaged F on A.symbol = F.symbol
left outer join technical_signals T on A.symbol = T.symbol
where username = 'rewan'
order by (A.symbol);

-- Fundamentally strong
select A.symbol from holdings_view A
left outer join fundamental_report F on A.symbol = F.symbol
where username = 'rewan'
group by(symbol);

-- Best companies to invest
select * from company_price
natural join fundamental_averaged
natural join technical_signals
natural join company_profile
where
EPS>25 and roe>13 and
book_value > 100 and
rsi>50 and adx >23 and
pe_ratio < 35 and
macd = 'bull'
order by symbol;

-- EPS more than 30
select * from fundamental_averaged
where eps > 30;

-- PE Ratio less than 30
select * from fundamental_averaged
where pe_ratio <30;

-- Technically
select * from technical_signals
where ADX > 23 and rsi>50 and rsi<70 and MACD = 'bull';

-- Total profit or loss
```

```
select * from transaction_history;
select * from holdings_view;

select username, A.symbol, sum(quantity) as quantity, sum(total) as total,
round(sum(total)/sum(quantity), 2) as updated_rate,
B.LTP, round((B.LTP * sum(quantity) - sum(total)), 2) as profit_loss
from transaction_history A left outer join company_price B
on A.symbol = B.symbol
group by A.username, A.symbol;

-- Watchlist
select symbol, LTP, PC, round((LTP-PC), 2) AS CH, round(((LTP-PC)/PC)*100, 2)
AS CH_percent from watchlist
natural join company_price
where username = 'suman'
order by (symbol);

-- For drop down watchlist addition
SELECT symbol from company_profile
where symbol not in
(select symbol from watchlist
where username = 'rewan');

-- News
select date_of_news, title, related_company, C.sector, group_concat(sources) as
sources
from news N
inner join company_profile C
on N.related_company = C.symbol
group by(title);

-- For pie chart (sector with total values)
select C.sector, sum(A.quantity*B.LTP) as current_value
from holdings_view A
inner join company_price B
on A.symbol = B.symbol
inner join company_profile C
on A.symbol = C.symbol
where username = 'rewan'
group by C.sector;


-- Profit loss in portfolio

-- Function to find total amount (amount + broker commission + sebon fee + DP
charge) excluding capital gain tax
DELIMITER $$
CREATE FUNCTION getTotal(
        total float
)
```

```
RETURNS float
DETERMINISTIC
BEGIN
        DECLARE total_converted float;
   DECLARE comm float;
   DECLARE ptotal float;

   -- If sell, make the total positive to calculate commission later
   IF total<0 THEN
                SET ptotal = -total;
        ELSE
                SET ptotal = total;
        END IF;

   -- Commission is same for both buy and sell
   IF ptotal > 500000 THEN
                SET comm = (0.34/100)*ptotal;
        ELSEIF ptotal > 50000 THEN
                SET comm = (0.37/100)*ptotal;
        ELSEIF ptotal > 2500 THEN
                SET comm = (0.4/100)*ptotal;
        ELSEIF ptotal > 100 THEN
                SET comm =  10;
        END IF;

   -- If sell conditions
        IF total < 0 THEN
                set total = -total;
                set total_converted = total - (comm + total*(0.015/100) + 25);
      RETURN total_converted;
   END IF;

        SET total_converted = total + comm + 25 + (0.015/100)*total;
        RETURN (total_converted);
END$$
DELIMITER ;


drop function getTotal;

-- Function to find profit loss with capital gain tax deducted
DELIMITER $$
CREATE FUNCTION capGain(
        total float,
   trans_date date
)
RETURNS float
DETERMINISTIC
BEGIN
   IF total<0 THEN
```

```
            RETURN total;
        ELSEIF datediff(Now(), trans_date)<365 THEN
            SET total = total - 0.075*total;
        ELSE
            SET total = total - 0.05*total;
        END IF;
        RETURN total;
END$$
DELIMITER ;

drop function capGain;


-- Profit loss summary
select symbol, sum(quantity) as quantity, LTP, rate,
capGain(round((getTotal(-sum(quantity)*LTP)) - (getTotal(sum(quantity)*rate)), 2),
transaction_date) as profit_loss
from transaction_history T
natural join company_price C
where username = 'rewan'
group by symbol;

select * from transaction_history
natural join company_price
where username='rewan'
group by symbol;

-- Stored procedure for portfolio (holdings with profit/loss)
DELIMITER $$
CREATE PROCEDURE portfolio(in username varchar(30))
BEGIN
select symbol, sum(quantity) as quantity, LTP
,round((getTotal(-sum(quantity)*LTP)), 2) as current_value,
capGain(round((getTotal(-sum(quantity)*LTP)) - (getTotal(sum(quantity)*rate)), 2),
transaction_date) as profit_loss
from transaction_history T
natural join company_price C
where username = username
group by symbol;
END$$
DELIMITER ;

drop procedure portfolio;
call portfolio('rewan')
```