Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

B. Tech. CE Semester – V

Subject: Advance Technologies

Project title: Online Trading System

By :-

sr.	Name	Roll No.	ID
1	Devansh P. Maru	CE-075	19CEUBG055
2	Gaurav K. Mori	CE-084	19CEUBG019
3	Vedant V. Panchal	CE-095	19CEUBS090

Guided By :-

Prof. Prashant M. Jadav

Department Of ComputerEngineering

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

CERTIFICATE

This is to certify that the practical / term work carried out in the subject of

Advance Technologies and recorded in this journal is the

bonafide work of

Maru Devansh P. (CE-075) (19CEUBG055)
Mori Gaurav K. (CE-084) (19CEUBG019)
Panchal Vedant V. (CE-095) (19CEUBS090)

of B.Tech semester **V** in the branch of **Computer Engineering**

during the academic year **2021.**

Prof. Prashant M. Jadav
Assistant Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University
Nadiad

Dr. C. K. Bhensdadia,
Head,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University
Nadiad

Contents:-

1. Abstract :-

In previous days the stock exchange **'hall'** is called a **'floor'** is divided into a number of markets according to the security which is being dealt with. The authorized clerk goes to the particular part of the floor called the **'pit'** and makes his quotation for the purchase or sale according to the order. The dealer to whom the quotation is given quotes his own price, if it does not suit the clerk, he asks for a lower price to be quoted. When both the sides are satisfied, the price is settled and the **'bargain'** is made usually, those bargains are orally settled, there is no return contract between the two parties. The clerk usually note book which records all purchases on the debit side and sales on the credit side. This is called a **'Kuchaa Hissab'** for noting down details.

In this system of trading in traditional stock exchanges of the transactions between the dealer and the investor.

After invention of the **ONLINE TRADING** in stock exchange there is greater transparency of trading

2 Introduction :-

The internet's arrival and its subsequent popularity in India have made online trading in India, which is about the **online purchase and sales of shares,** one of the extremely popular means of trading. Both beginner and experienced traders and investors in India are milking this opportunity by trading online in futures and options, stocks and currencies worldwide. Such opportunities are in the form of reduced brokerage and commissions, better broking services etc.

Constructive uses of new technologies have always contributed positively towards improving human life standards and the economy of a country .Such as online trading, in equity markets it increased trade volumes and number of **investors trading in stock markets.** Online trading was started in India in the year 1995, where a new system is formed which allows the investor to trade through an internet site where the major financial products and services like equities, mutual funds, life insurance, general insurance, loans, share trading, commodities trading, portfolio management and financial planning etc. are directly available for the customer.

There is also stock option trading in India. An option is a **financial agreement,** with a predetermined maturity period and price, for the purchase or sales of the underlying products. Stock options enable the protection of dealers and control of their stocks, in addition to generation of higher earnings.

The investor has to register with an online trading portal and get into an agreement with the firm to trade in different securities following the terms and conditions listed down on the agreement. The order processing is done in correct timings as the servers of the online trading portal are connected to the **stock exchanges** and designated banks all around the clock. They can also get updates on the trading and check the current status of their orders either through e-mail or through the interface. Brokerages also provide research content on their websites, such that the clients can take their own decisions on stocks before investing.

Thanks to the ever-rising number of people owning computers along with a readily available internet access, **online stock trading** in India is simplified manifolds. This is because investments can now be easily controlled by traders themselves as a result of extensive availability of all types of information on the web.

3 Software Requirement Specification:-

Online Trading System

1. User Account Management:

R.1.1: Log-in Panel:

Description : Users are required to enter the necessary details in the log-in form to enter the Web. If Users does not have an account then they have to be registered first

.

Input : Enter email id and password by user

Output : confirmation message displayed.

R.1.2: Regestration Panel:

Description : Users are required to enter the necessary details in the Sign-in form.

Input : Enter Nesscery Details By user.

Output : confirmation message displayed And Redirect To Login page.

2. Stock Management:

R.2.1: Add Companies in Watchlist:

Description : User Can add any company in the watchlist threw Search Opreation. And added to the database.

Input : Click add Button to add a company.

Output : successfully added to the watchlist.

R.2.1.1: Search stock-graph of Companies:

Description : Users can search a any Company by their name and saw that Stock graph of that company. And analysis that in Which year company have highest and lowest stock prize.

Input : Enter the Company Name By user.

Output : Can see a Stock-graph of particular Company.

R.2.1.2: Company Details:

Description : User Can see a Share prize, Latest prize of share, Change In the Rate Of particular company.

Input : whatever user want to know about company.

Output : Details of Company.

R.2.3: Delete Companies from Watchlist:

Description : User Can delete a company from watchlist. And Deleted From database also.

Input : Click delete button to delete a company.

Output : successfully deleted from the watchlist.

3. Purchase Stock Management:

R.3.1: Buy stock:

Description : User can Buy a stock of any company from watchlist of Companies.

Input : Click Buy button to purchase a stock of company.

Output : successfully purchased from the watchlist.

R.3.1.1: Total of stocks:

Description : Total Of the number of stocks that user purchase of Different companies.

Input : Total share that user buy.

Output : Total prizes of the shares and Total required Amount.

R.3.2: Delete stock:

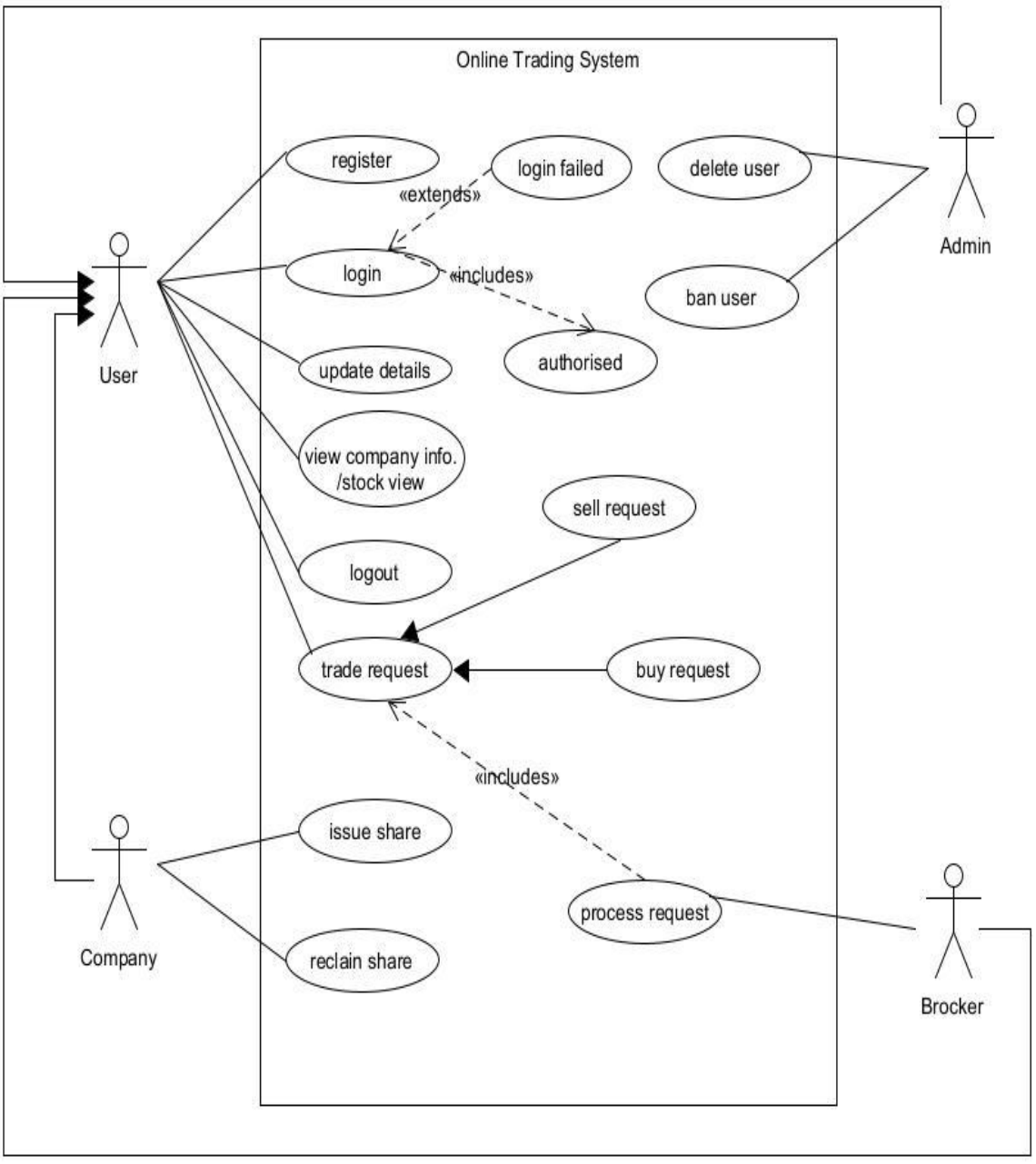
Description : Users can Delete a stock of any company after they Buy a stock

Input : Click delete button to delete a stock of company.

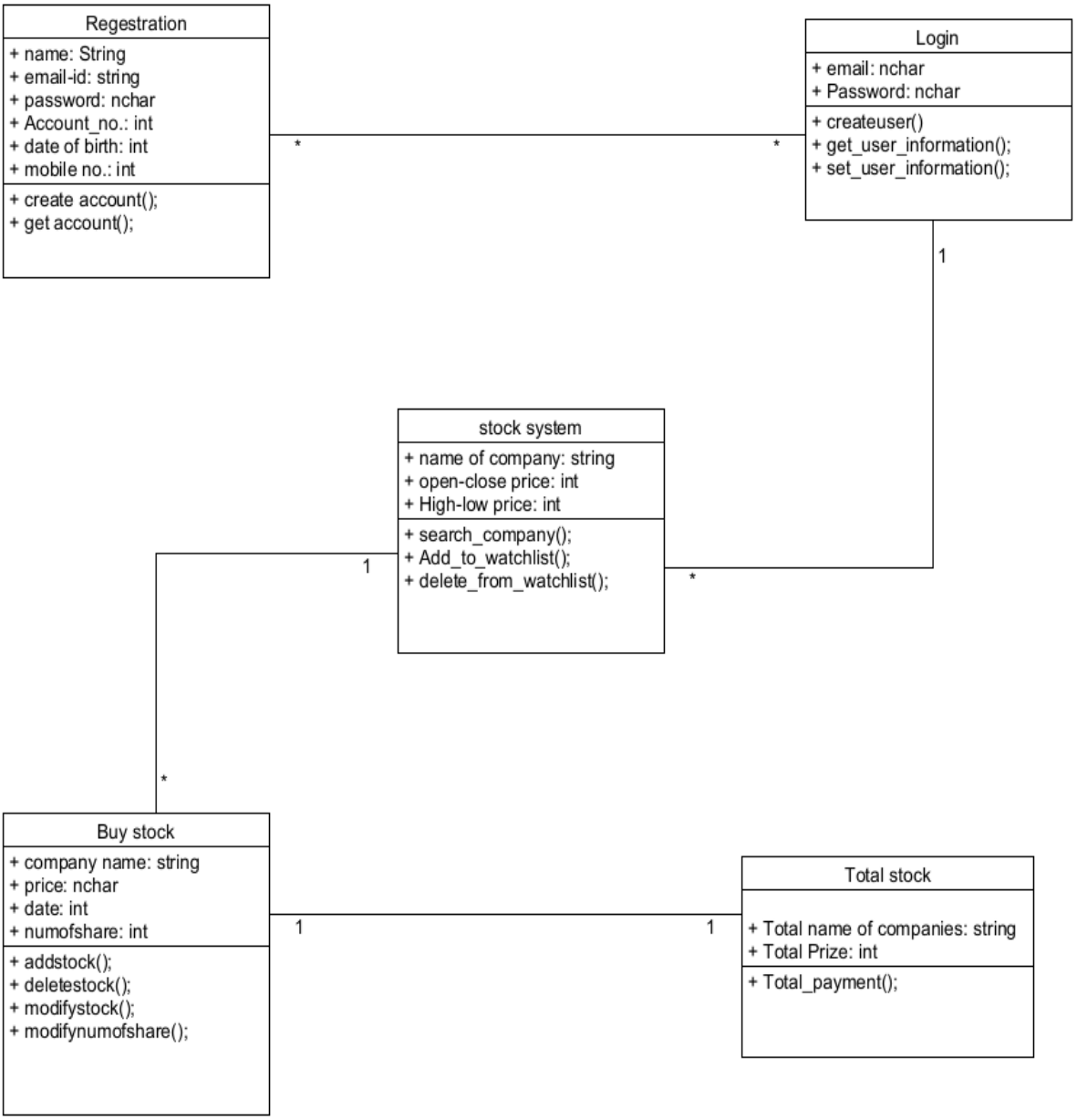
Output : successfully deleted from the purchased stocks.

4. Design :

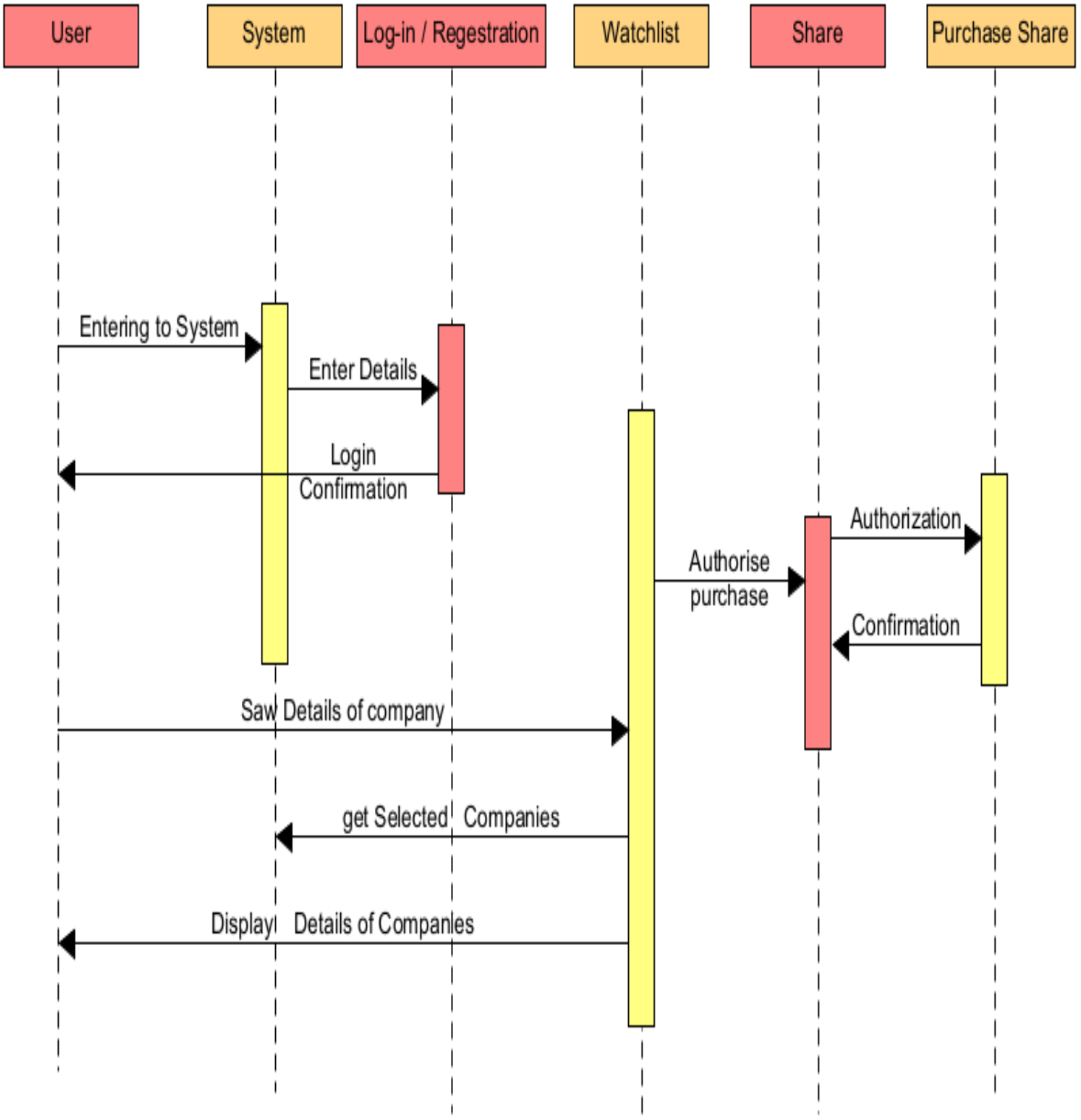
4.1 Use-case Diagram :



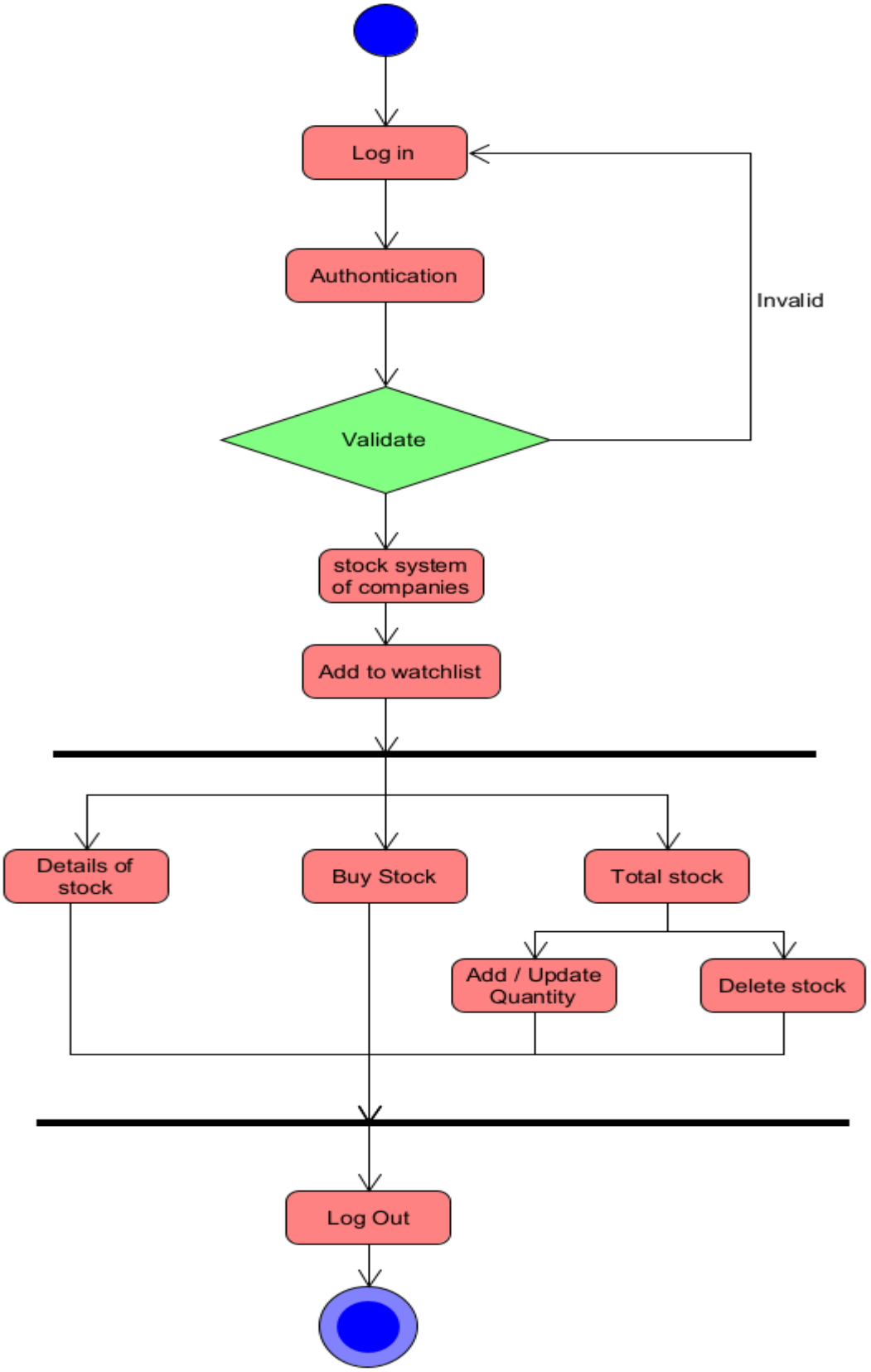
4.2 Class Diagram:



4.3 Sequence Diagram :

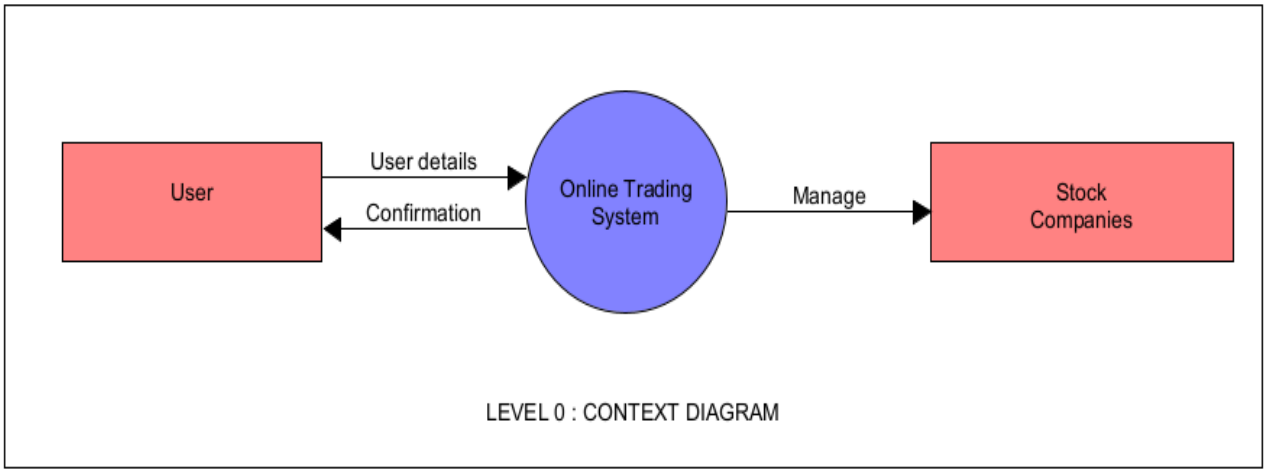


4.4 Activity Diagram :

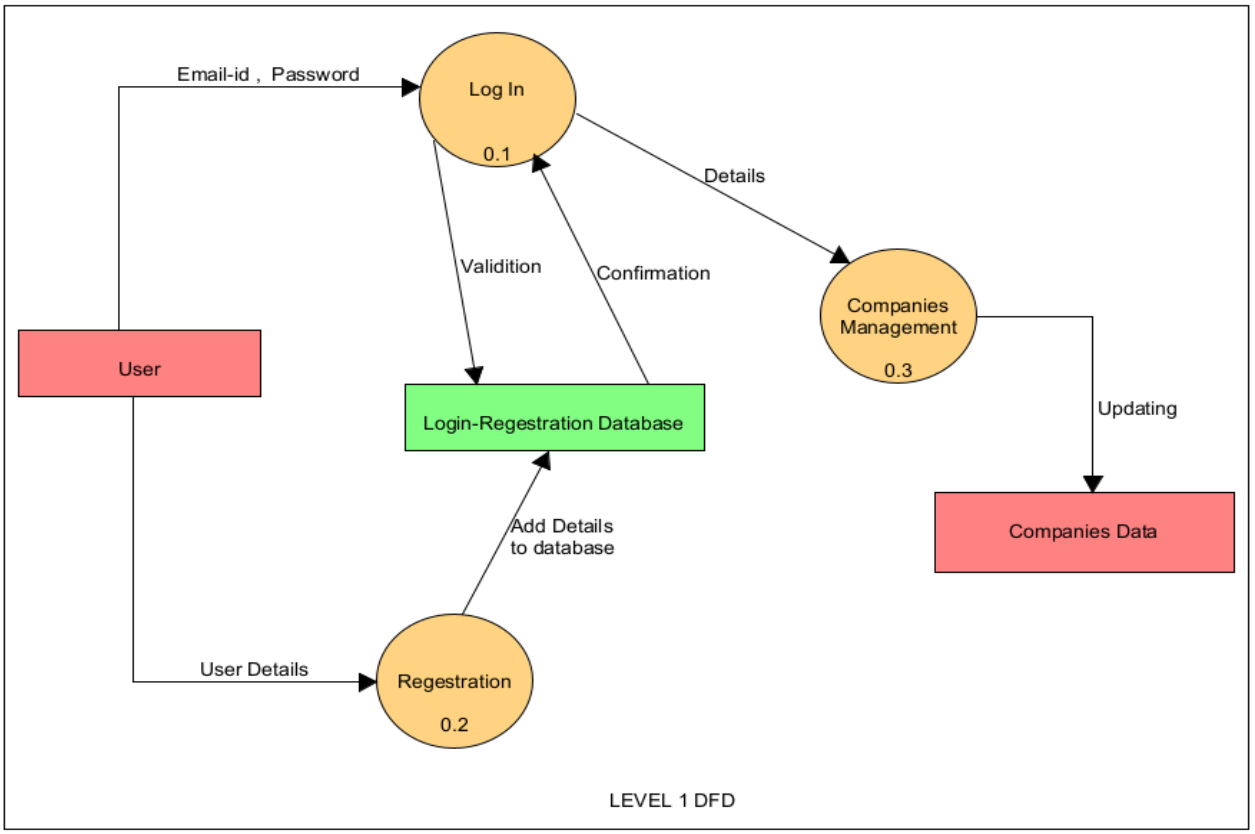


4.5 Data-flow Diagram :

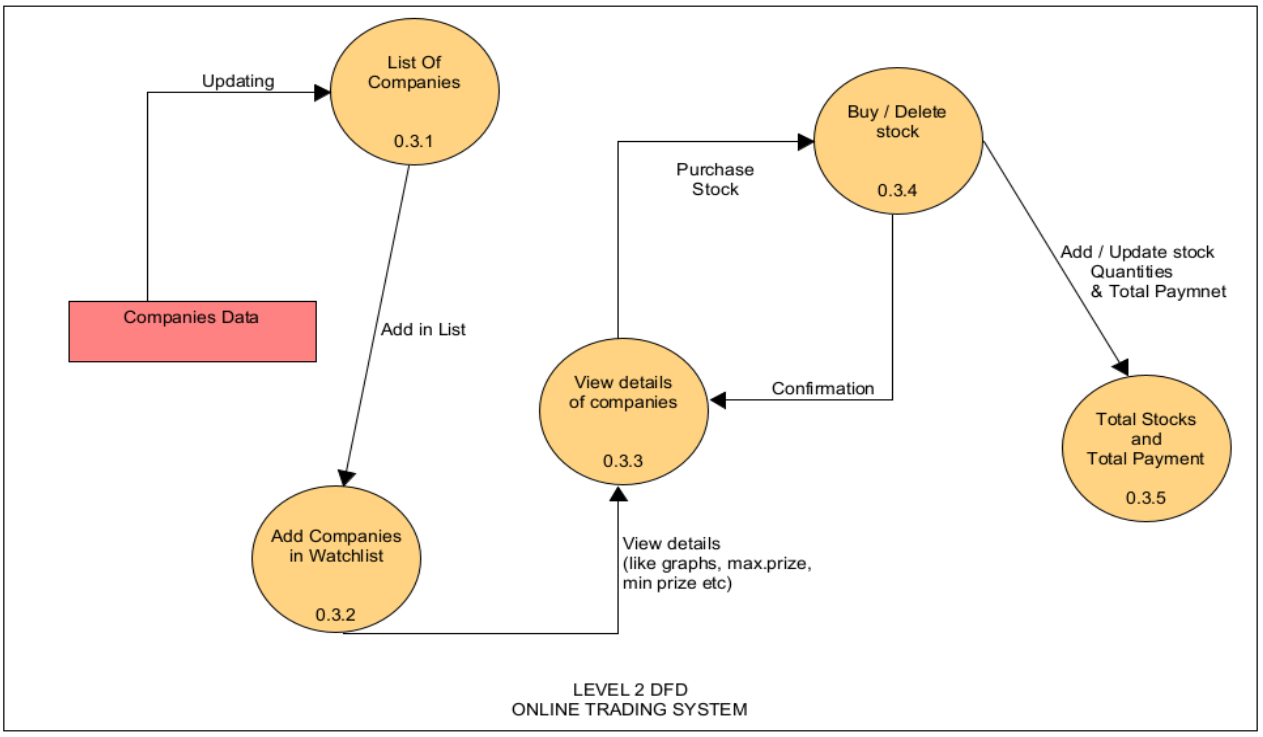
Level 0 Context DFD :



Level 1 DFD :



Level 2 DFD :


List Of Companies
0.3.1
Updating
Companies Data
Add in List
Add Companies in Watchlist
0.3.2
View details of companies
0.3.3
Purchase Stock
Buy / Delete stock
0.3.4
Confirmation
Add / Update stock Quantities & Total Paymnet
Total Stocks and Total Payment
0.3.5
View details (like graphs, max.prize, min prize etc)
LEVEL 2 DFD
ONLINE TRADING SYSTEM

4.6 Data-Dictionary :

Sr.	Name	Type	Size	PK/FK	Refered Table	Unique	Desc.
1	name	varchar	255	-	-	no	-
2	email	varchar	255	PK	-	yes	-
3	date of birth	decimal	10	-	-	no	-
4	mobile no.	decimal	10	-	-	yes	-
5	account no.	decimal	12	-	-	yes	-
6	password	varchar	05	-	-	no	-
7	confirm password	varchar	05	-	-	no	-

5. Implementation Details:-

5.1 Module:-

The system consists of 3 basic modules namely,

1. User Module
2. Stock Management Module
3. Purchase Stock Module

Each module consists of several methods to implement the required functionality. Implementation is done using React , Node js , Express , Mongoose. Database used in these modules is MongoDb.

1. User Module :

This module is the base for authentication and authorization to ensure the security aspect of the user. It also includes profile creation,login and logout using Mongodb database.

2. Stock Management Module :

This module is for Stocks details of different companies. Each company have their stock graphs , price , high-low ratio etc. user can add companies in watchlist and buy stocks

3. Purchase Stock Module :

This module is for Purchase a stock from watchlist. User can update quantities of stocks. And at the end user can see total stocks and total stock prize.

5.2 Major function prototype :

Login-register (back-end part) :

```
import express from "express"
import cors from "cors"
import mongoose from "mongoose"

const app = express()
app.use(express.json())
app.use(express.urlencoded())
app.use(cors())

mongoose.connect("mongodb://localhost:27017/trading_online", {
    useNewUrlParser: true,
    useUnifiedTopology: true
}, () => {
    console.log("DB connected")
})

const userSchema = new mongoose.Schema({
    name: String,
    email: String,
    dob: Date,
    phone: Number,
    acno: Number,
    pass: String
})

const User = new mongoose.model("User", userSchema)
```

```
app.post("/login", (req, res)=> {
    const { email, pass} = req.body
    User.findOne({ email: email}, (err,user) => {
        if(user){
            if(pass === user.pass ) {
                res.send({message: "Login Sucessfull", user: user})
            } else {
                res.send({message: "Password did not match"})
            }
        } else {
            res.send({message:"User not registered"})
        }
    })
})

app.post("/register", (req, res)=> {
    const {name, email, pass, phone, acno} = req.body
    User.findOne({email: email}, (err, user) => {
        if(user){
            res.send({message: "User already register"})
        } else {
            const user= new User({
                name,
                email,
                pass,
                phone,
                acno
            })
            user.save( err => {
                if(err) {
                    res.send(err)
```

Add Stock :

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import './AddStock.css';
import {Coin} from './Coin';
import { Link } from 'react-router-dom';
import Navbar from "./Navbar";
 export const AddStock = () => {
  const [coins, setCoins] = useState([]);
  const [search, setSearch] = useState('');
  const [result, setResult] = useState('');

  useEffect(() => {
    axios
      .get(
        ' http://localhost:8000/data'
      )
      .then(res => {
        setCoins(res.data);
        setResult(res.data);
        console.log(res.data);

      })
      .catch(error => console.log(error));
  }, []);

  const handleChange = e => {
    setSearch(e.target.value);
  };

  const filteredCoins = coins.filter(coin =>
    coin.T.toLowerCase().includes(search.toLowerCase())
  );
```

```
return (
  <div>
    <Navbar />
  <div className='coin-app'>
    <div className='coin-search'>
      <h1 className='coin-text' style={{color:"#843CF0"}} > Search a Company ...</h1>

      <form>
        <input
          className='coin-input'
          type='text'
          onChange={handleChange}
          placeholder='EX. TSLA/TTM/FB/TCS'
        />
      </form>
    </div>
```

Information of companies :

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import Chart from "./Chart";
import './watchlistdetails.css'
export const InfoC = (company) => {
    const [coins, setCoins] = useState([]);
    const [search, setSearch] = useState('');
    const [result, setResult] = useState('');


  let name= (company["company"]["name"]);
    useEffect(() => {
        axios
          .get(
            `https://www.alphavantage.co/query?function=OVERVIEW&symbol=${name}&apikey=QLC4LW522J6KY4JK`
          )
          .then(res => {
            setCoins(res.data);
            setResult(res.data);
            console.log(res.data);

          })
          .catch(error => console.log(error));
      }, []);
```

```
    return (
        <div>
            <div className="Details">
          <h1 style={{color: ' #B266FF'}}> {result.Name} </h1>
          <br></br>
          <label style={{color: '#CCFFCC'}}>  AssetType : {result.AssetType} </label>
           <br/>
           <br/>
           <label  style={{color: '#CCE5FF'}} >  {result.Description}</label>
           <br/>  <br/>
             <Chart company = { company }/>
            <div className="price">
            <br/>  <br/>
                <h2 style={{color: 'rgb(218, 198, 114)',fontSize:"30px"}}> Fundamentals </h2> <br/>
        <label>Price : ${company["company"]["price"]}</label>   <br/>  <br/>
        <label>High :  ${company["company"]["high"]}</label>     <br/>  <br/>
        <label>Low :  ${company["company"]["low"]}</label>  <br/>  <br/>
        <label>PriceChange :{Number( company["company"]["priceChange"]).toFixed(2)< 0 ? (
            <label className='coin-percent red'>{ Number(company["company"]["priceChange"]).toFixed(2)}%</label>
          ) : (
            <label className='coin-percent green'>{Number(company["company"]["priceChange"]).toFixed(2)}%</label>
          )}</label>  <br/>  <br/>
           <label>52 Week High :  ${result["52WeekHigh"]}</label>     <br/>  <br/>
        <label>52 Week low :  ${result["52WeekLow"]}</label>  <br/>  <br/>
        <label>LatestQuarter : {result["LatestQuarter"]}</label>   <br/>  <br/>
            </div>
            <div className="tech">
            <br/>  <br/>
            <h2 style={{color: 'rgb(218, 198, 114)',fontSize:"30px"}}> Technicals </h2> <br/>
        <label>Volume : ${company["company"]["volume"]}</label>   <br/>  <br/>
        <label>Marketcap : ${company["company"]["marketcap"]}</label>   <br/>  <br/>
```

Display Companies details :

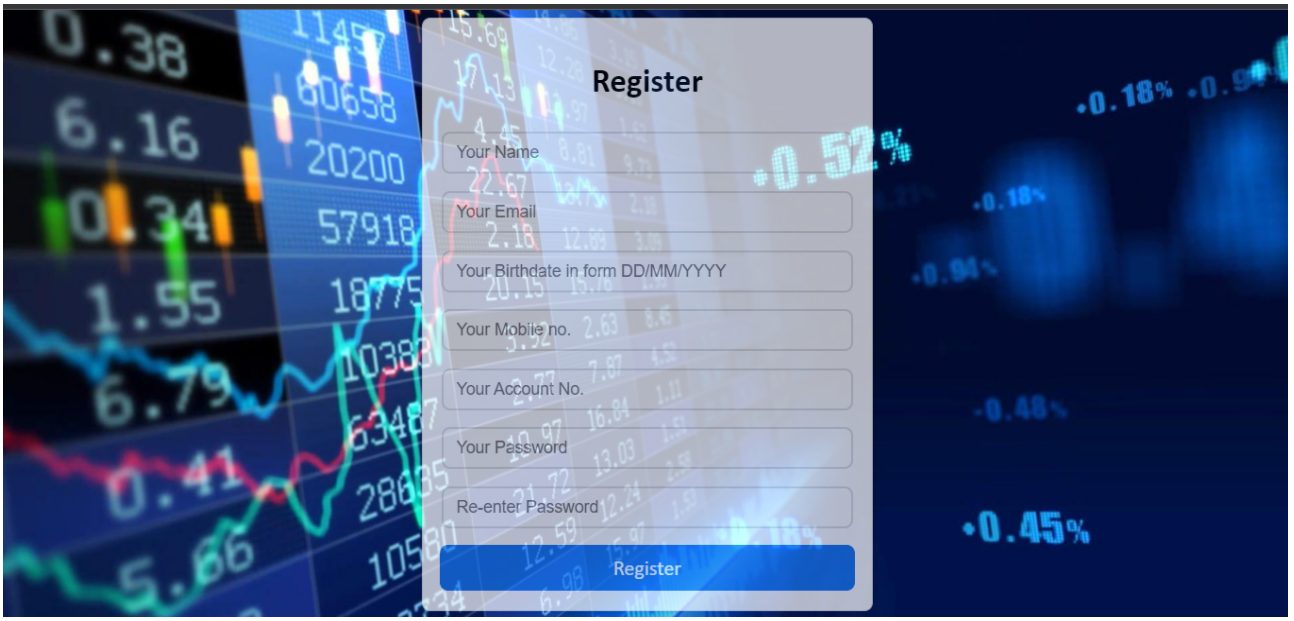
```
const PortfolioWatch = ({ company }) => {
    return (

        <div className="blog-list">

          {company.map(blog => (
              <div className='coin-data'>
            <div className="blog-preview" key={blog.id} >
              <Link to={`/PortfolioDetail/${blog.id}`}>
              <div className='coin-container'>
            <div className='coin-row'>
              <div className='coin'>

              <h1  style={{color: ' #E1A37F'}}>{blog.name} </h1>

              </div>
              <div className='coin-data'  style={{color: ' rgb(119, 255, 203)'}}>
              <p className='coin-price'>Open : ${blog.price}</p>
              <p className='coin-price'>QTY : {blog.count}</p>

              {/* <p className='coin-volume'>Volume : ${String(blog.volume)}</p> */}

              {Number(blog.priceChange).toFixed(2)< 0 ? (
                <p className='coin-percent red'>{Number(blog.priceChange).toFixed(2)}%</p>
              ) : (
                <p className='coin-percent green'>{Number(blog.priceChange).toFixed(2)}%</p>
              )}
```

Fetch the Stock-details :

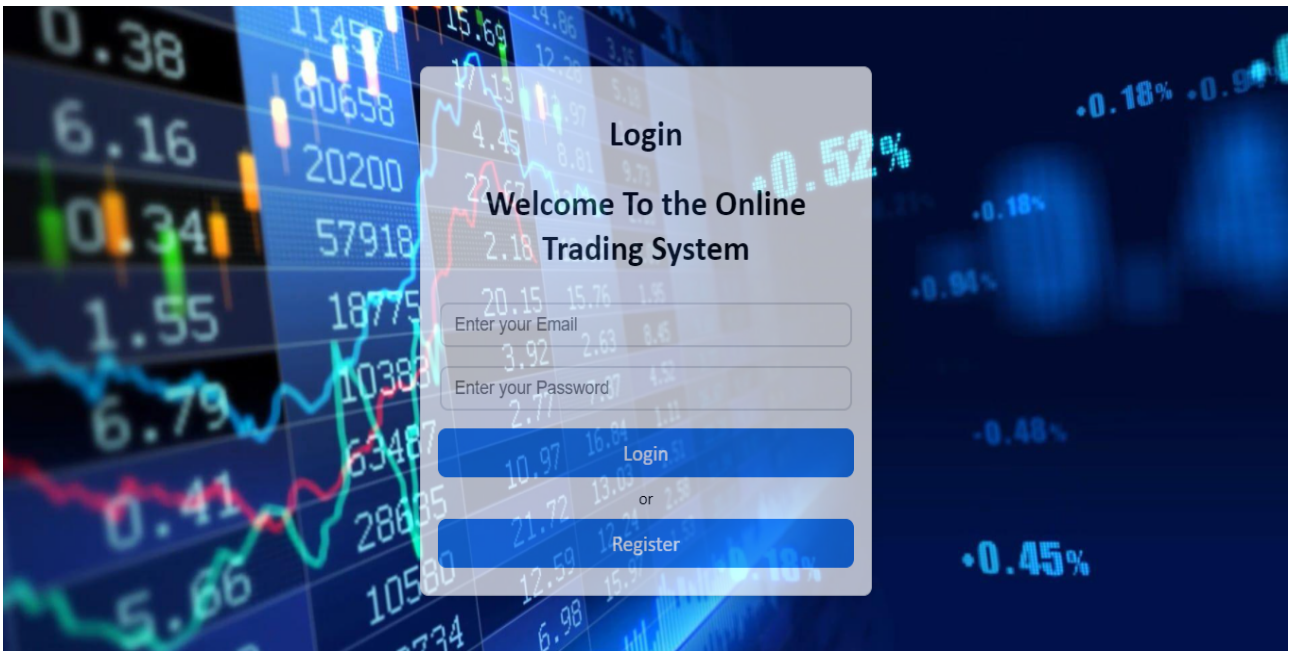
```
fetchStock() {
    const pointerToThis = this;
    //A0KA42Q8F2PXWPFB
    const API_KEY = 'QLC4LW522J6KY4JK';
    let StockSymbol = this.state.name;
    let API_Call2 =`https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=${StockSymbol}

    let API_Call = `https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=${StockSymbol}
    let stockChartXValuesFunction = [];
    let stockChartYValuesFunction = [];

    fetch(API_Call2)
      .then(
        function(response) {
          return response.json();
        }
      )
      .then(
        function(data) {
          console.log(data);
          for (var key in data['Time Series (Daily)']) {
            stockChartXValuesFunction.push(key);
            stockChartYValuesFunction.push(data['Time Series (Daily)'][key]['1. open']);

          }
          pointerToThis.setState({
            stockChartXValues: stockChartXValuesFunction,
            stockChartYValues: stockChartYValuesFunction
          });
        }
      )
  }
```

6. ScreenShots:-

Register-



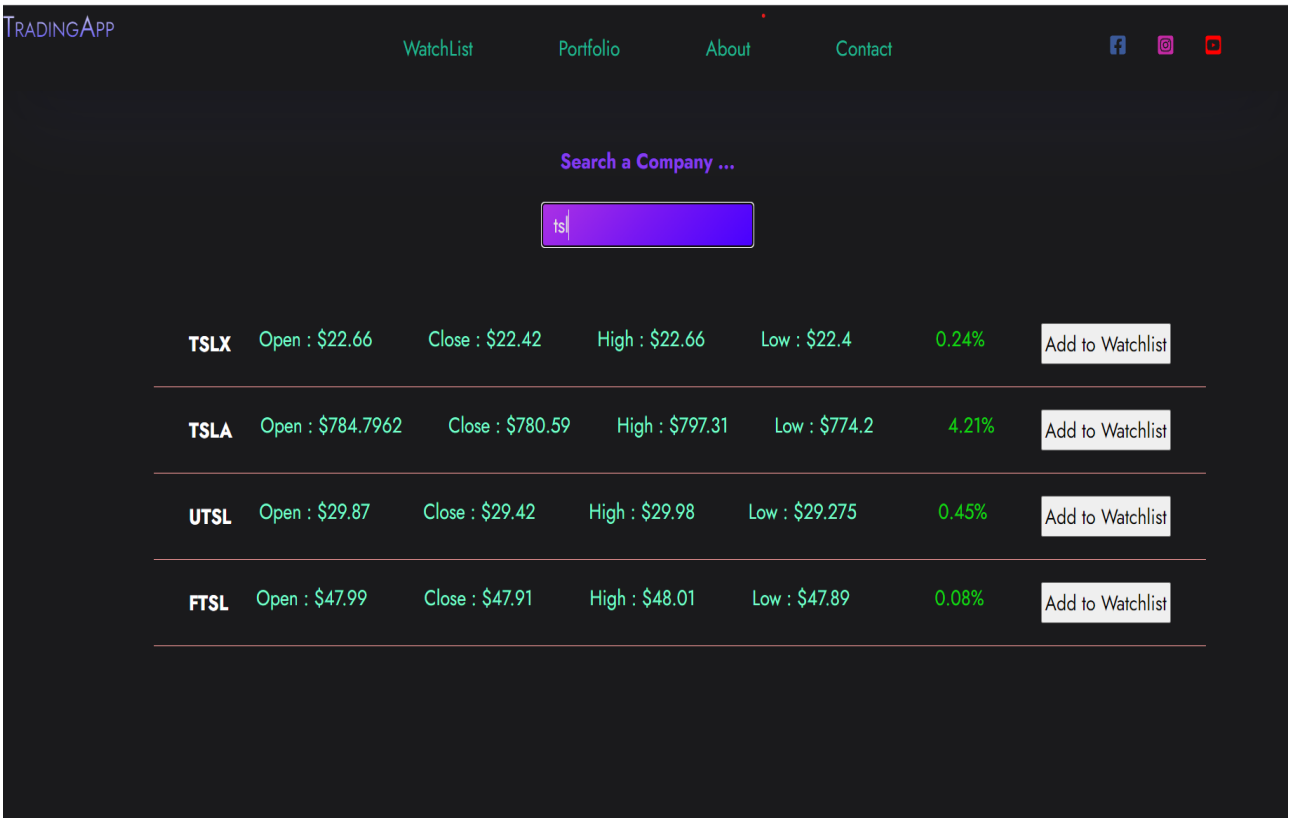
Log-in-



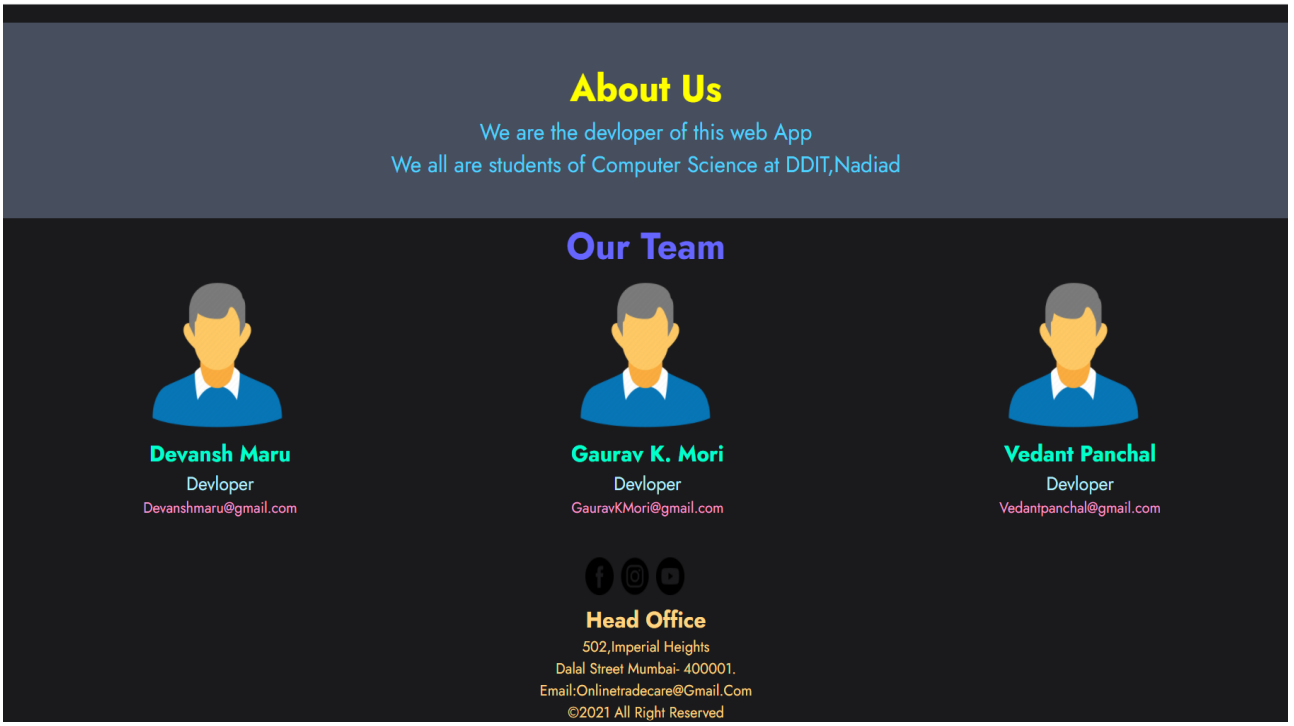
Homepage-

Companies list-

Seach & Add into watchlist companies-



Team Details-



View company details-

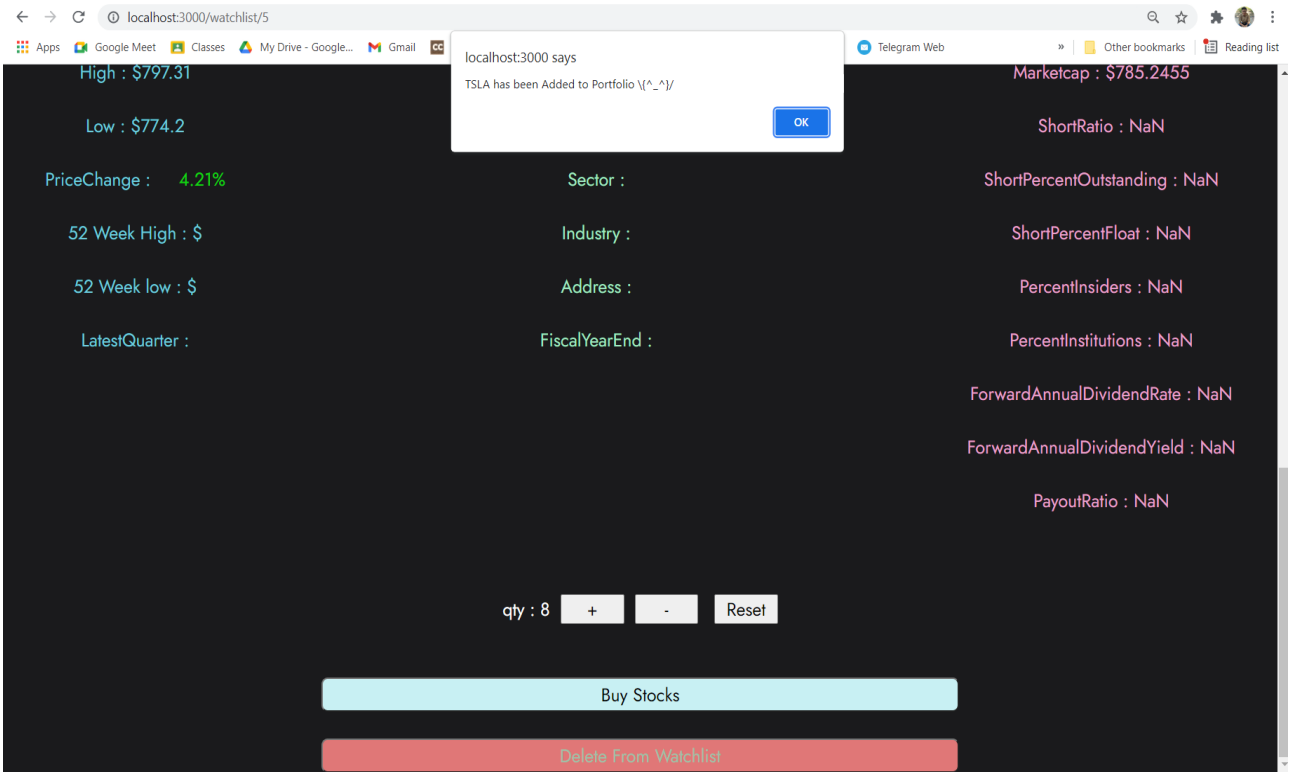
Tesla Inc

AssetType : Common Stock

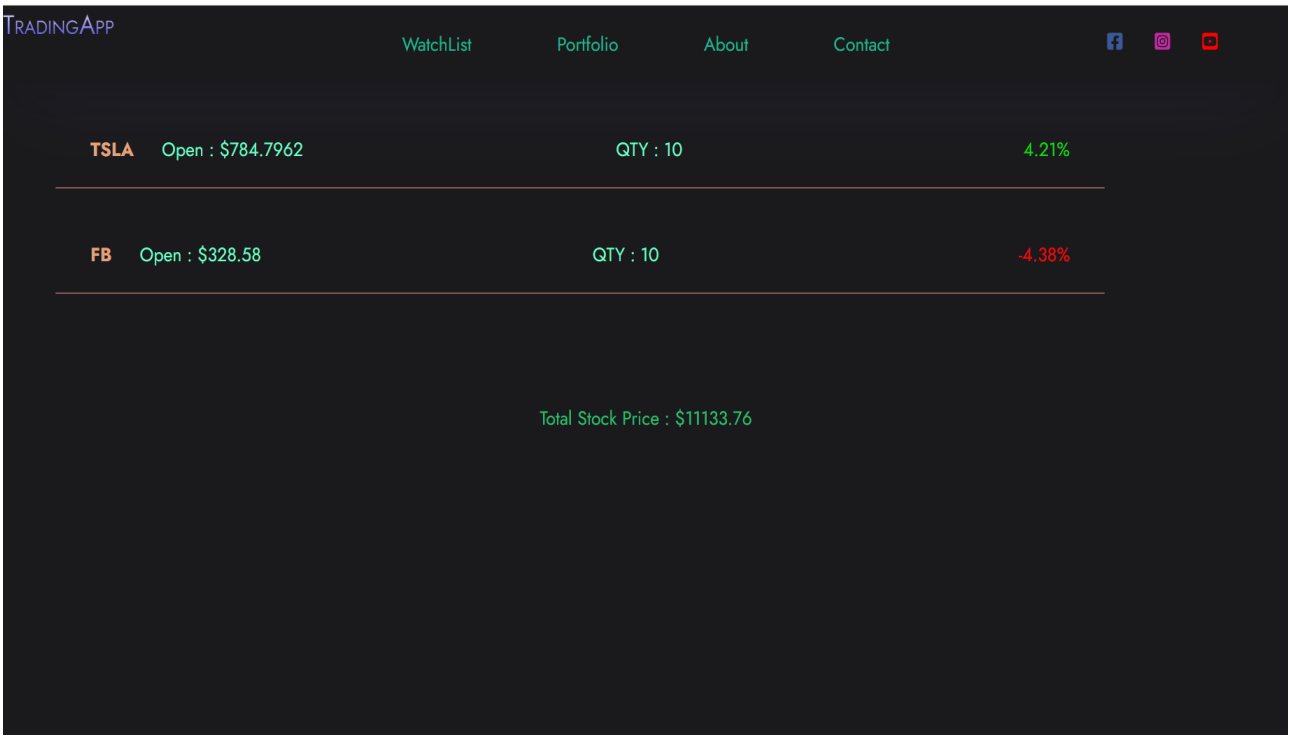
Tesla, Inc. is an American electric vehicle and clean energy company based in Palo Alto, California. Tesla's current products include electric cars, battery energy storage from home to grid-scale, solar panels and solar roof tiles, as well as other related products and services. In 2020, Tesla had the highest sales in the plug-in and battery electric passenger car segments, capturing 16% of the plug-in market (which includes plug-in hybrids) and 23% of the battery-electric (purely electric) market. Through its subsidiary Tesla Energy, the company develops and is a major installer of solar photovoltaic energy generation systems in the United States. Tesla Energy is also one of the largest global suppliers of battery energy storage systems, with 3 GWh of battery storage supplied in 2020.

Fundamentals	About	Technicals
Price : $784.7962	Exchange : NASDAQ	Volume : $18432625
High : $797.31	Currency : USD	Marketcap : $785.2455
Low : $774.2	Country : USA	ShortRatio : 1.56
PriceChange : 4.21%	Sector : MANUFACTURING	ShortPercentOutstanding : 0.03
52 Week High : $895.9	Industry : MOTOR VEHICLES & PASSENGER CAR BODIES	ShortPercentFloat : 0.04
52 Week low : $379.11	Address : 3500 DEER CREEK RD, PALO ALTO, CA, US	PercentInsiders : 18.96
LatestQuarter : 2021-06-30	FiscalYearEnd : December	PercentInstitutions : 41.43
		ForwardAnnualDividendRate : 0.00
		ForwardAnnualDividendYield : 0.00
		PayoutRatio : 0.00

Buy Stock-

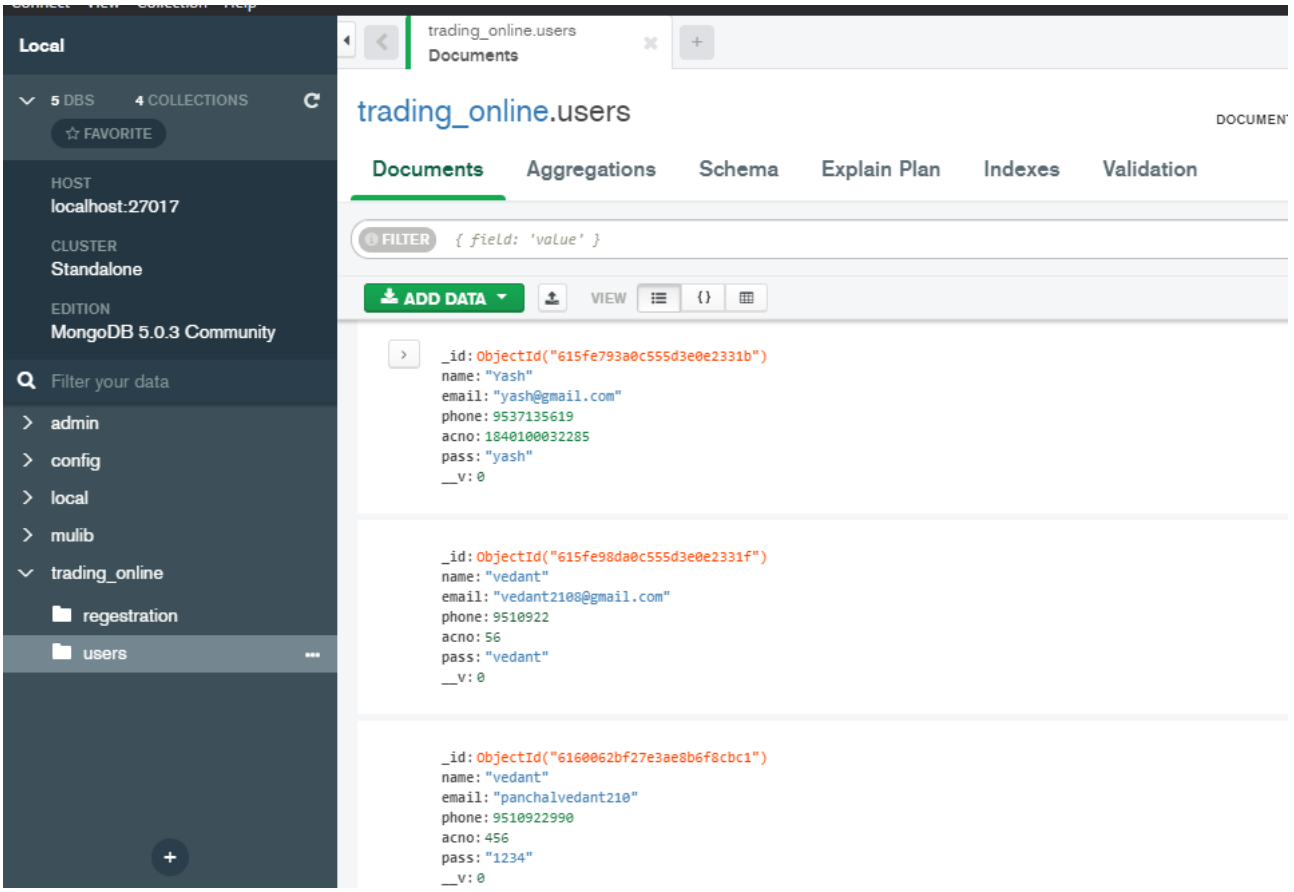


Total Stocks & Total Price-



Information-

Database (MongoDb)-



7. Conclusion:-

- The main features of this webapp includes flexibility, reduce manual work in an efficient manner, a quick, convenient, reliable and effective way to apply for their online trading market committee records. The project could very well be enhanced further as per the requirements.

- Online trading is the new concept in the stock market. In India, online trading is still at its infancy stage. Online trading has made it easy to trade in the stock market as now people can trade while sitting at their home. Now stock market is easily accessible by the people.

- There are some problems while doing the trade through the internet. Major problem faced by online trader is that the investors are loyal to their traditional brokers, they rely upon the suggestions given by their brokers. Another major problem is that the people don't have full knowledge regarding online trading. They find it difficult to trade themselves, as a wrong entry made by them, can bring them huge losses.

8. Limitations and Future Enhancements:-

Limitations:-

- We are unable to do Real Payment through bank account.
- Its Subject to Higher Risk. Its need tight security.
- Risk: user could lose their entire investment. If a company does poorly, investors will sell, sending the stock price plummeting. When user sell, user will lose your initial investment.

Future Enhancements:-

- User can do the real payment by bank account no.
- User can do Online payment.
- User can give feedback to the system.
- Manual admin panel.
- Email verification after signup.
- Tight authorization.
- UI designs.

9. Reference / Bibliography:-

Following links and websites were referred during the development of this project:

- https://stackoverflow.com
- https://nodejs.org/en/
- https://reactjs.org
- https://www.npmjs.com/package/json-server
- https://www.mongodb.com/
- https://expressjs.com/
- https://www.google.com/

Thank You