STOCK SELECTION TOOL REPORT

PREPARED FOR: DR. AAMIR ADEEB
PREPARED BY: WAN NUR FATIHAH BINTI WAN HALIM

19 DECEMBER 2024

1.0 INTRODUCTION

The Stock Selection Tool is a Python-based application designed to help users examine historical stock closing prices from the Malaysian market. Given the speed at which technology is developing and the growing significance of making decisions based on data, this tool gives users an easy and efficient way to learn about the performance of stocks over a given time frame.

The tool allows users to safely register and log in, retrieve past closing price information for certain stocks, carry out necessary analyses, and save the outcomes for later use. The program guarantees precise data retrieval and smooth user interactions by utilising powerful Python modules like pandas and yfinance.

For people or companies looking for a simple method of analysing stock trends without the need for sophisticated financial tools, the Stock Selection Tool is very beneficial. Users can learn more about market behaviour and improve their decision-making skills through this application.

2.0 OBJECTIVE

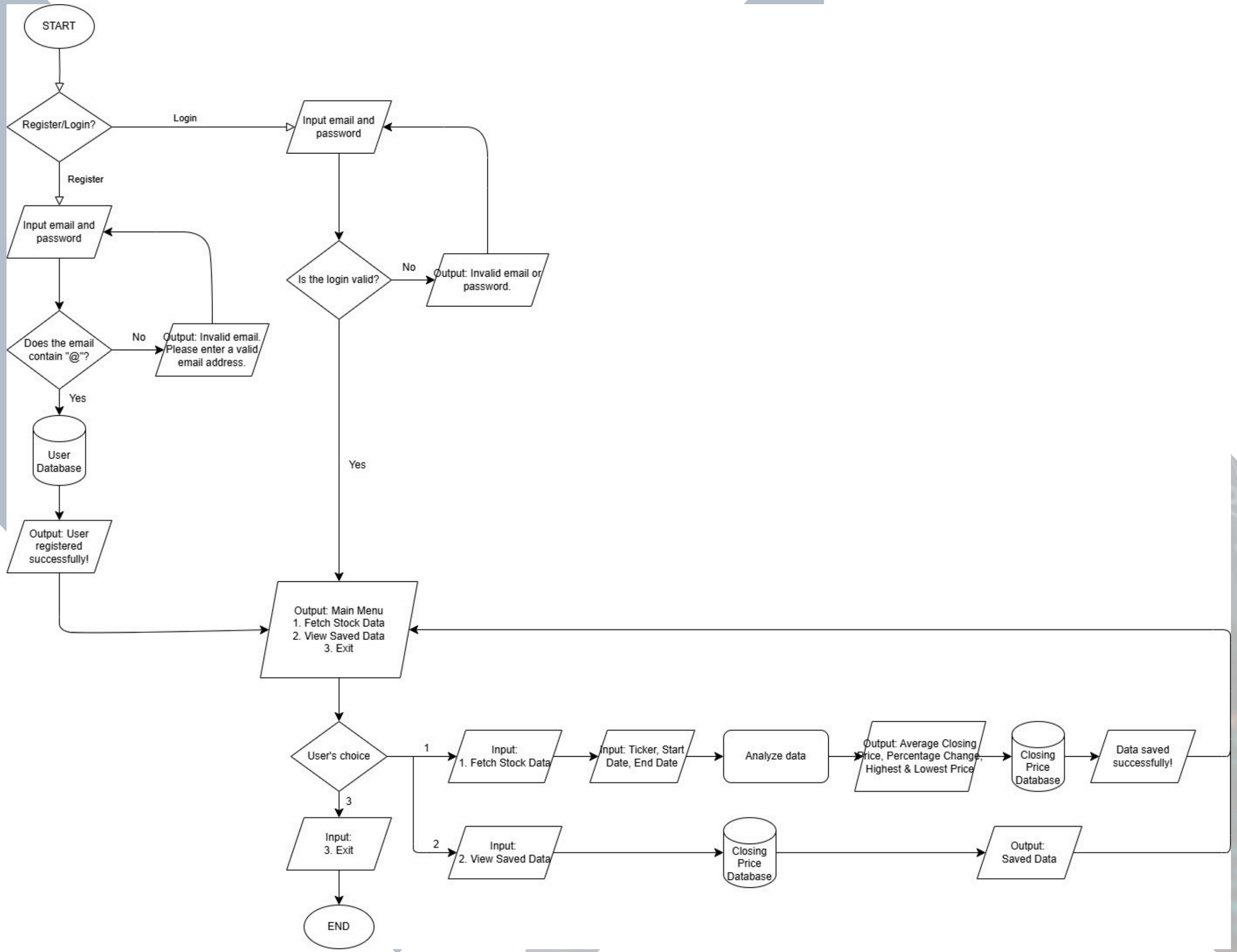
1. To streamline stock data retrieval for specific stock tickers and time frame

2. Enable data storage and retrieval of saved data

3. Perform essential stock data analysis with basic metrics, such as average closing price, percentage change, and identification of the highest and lowest closing prices.

3.0 METHODOLOGY

FLOWCHART



3.0 METHODOLOGY

FLOWCHART EXPLANATION

1. When the tool is ran, the user is prompted to choose whether to register or login.
2. If the user chooses to register, users enter an email and password, where the email is validated to ensure it includes the “@” symbol. The information are stored if the registration is successful.
3. If the user chooses to login, existing users input their email and password. If valid, the user is taken to the main menu, otherwise, an error message is displayed.
4. After authentication, the user is taken to the main menu, where three options are presented:

- 1. Fetch Stock Data: The user has to enter stock ticker and specify the start and end dates for the data retrieval. The system then fetches historical data for the specified criterias using the YFinance library. If successful, basic analysis is performed, including showing the average closing price, percentage change, and identifying the highest and lowest closing prices. If not, an error message is shown, and the user is prompted to retry.
- 2. View Saved Data: The system checks for the existence of the CSV file storing user interactions and analysis results. The data is shown to the user if the file is present. The system notifies the user that no data is accessible if the file is not found.
- 3. Exit: The program is terminated when this option is selected. Before it ends, the tool makes sure that all of the data is safely preserved.

4.0 SETUP & CONFIGURATION

- **Pandas**

Pandas is a Python library for data manipulation and analysis, used to calculate stock metrics and manage CSV files.

- **YFinance**

It is a tool to fetch historical stock data from Yahoo Finance, used to retrieve closing prices for Malaysian stocks.

- **CSV**

CSV is a file format for storing tabular data, used to save user interactions like email, stock tickers, and analysis results.

- **JSON**

JSON is a lightweight data format, used to securely store and retrieve user credentials for registration and login.

5.0 CODING EXPLANATION (MAIN.PY)

main.py > main

```python
import functions

def is_valid_email(email):
    """Check if the provided email is in a valid format."""
    if "@" in email and email.count("@") == 1 and email.split("@")[1]:
        return True
    return False
```

This is to validate the format of the email address entered, where it must contain “@” symbol.

```python
def main():
    print("Welcome to the Stock Selection Tool!")

    while True:
        action = input("Do you want to register or login? (register/login): ").strip().lower()
        if action in ["register", "login"]:
            break
        print("Invalid choice. Please type 'register' or 'login'.")
```

This displays the welcome message at the main interface, and asks the user to decide whether to register or login.

```python
    while True:
        email = input("Enter email: ").strip()
        if is_valid_email(email):
            break
        print("Invalid email. Please enter a valid email address.")

    password = input("Enter password: ").strip()
```

This will ask the user to enter the email address and password, and show the error message if the email address entered is invalid, which is without “@” symbol.

```python
    if action == "register":
        success, msg = functions.register_user(email, password)
        print(msg)
        if not success:
            return
    elif action == "login":
        success, msg = functions.authenticate_user(email, password)
        print(msg)
        if not success:
            return
```

If user chooses to register, the functions allow the tool to create and store new users, and if user chooses login, it will take the user to enter their existing email and password.

5.0 CODING EXPLANATION (MAIN.PY)

```python
def main():
    while True:
        print("\n1. Fetch Stock Data")
        print("2. View Saved Data")
        print("3. Exit")
        choice = input("Enter your choice: ").strip()
```

Output, or the display of the Main Menu

```python
def main():
    if choice == "1":
        ticker = input("Enter stock ticker (e.g., 1155.KL): ").strip()
        start_date = input("Enter start date (YYYY-MM-DD): ").strip()
        end_date = input("Enter end date (YYYY-MM-DD): ").strip()

        closing_prices = functions.get_closing_prices(ticker, start_date, end_date)
        if closing_prices.empty:
            print("No data available for the given ticker and date range.")
            continue

        analysis = functions.analyze_closing_prices(closing_prices)
        print(f"\nAnalysis Results for {ticker}:")
        for key, value in analysis.items():
            print(f"{key}: {value}")

        data_to_save = {"email": email, "ticker": ticker, **analysis}
        functions.save_to_csv(data_to_save, "data.csv")
        print("Data saved successfully!")
```

If the user chooses option 1, which is to Fetch Stock Data, they will be required to enter stock ticker, start date and end date of the data that is being examined.

```python
    elif choice == "2":
        saved_data = functions.read_from_csv("data.csv")
        if saved_data is not None:
            print("\nSaved Data:")
            print(saved_data)

    elif choice == "3":
        print("Goodbye!")
        break

    else:
        print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

If the user chooses option 2, the program will take the information from data.csv of searches made before. If user chooses option 3, the program will close and end.

5.0 CODING EXPLANATION

(FUNCTIONS.PY)

```python
def get_closing_prices(ticker, start_date, end_date):
    """Fetch historical closing prices for the given stock ticker."""
    try:
        stock_data = yf.download(ticker, start=start_date, end=end_date)
        return stock_data["Close"]
    except Exception as e:
        print(f"Error fetching data: {e}")
        return pd.Series()

def analyze_closing_prices(data):
    """Perform basic analysis on closing prices."""
    if data.empty:
        return {}

    average_price = data.mean()
    percentage_change = ((data.iloc[-1] - data.iloc[0]) / data.iloc[0]) * 100
    highest_price = data.max()
    lowest_price = data.min()

    return {
        "Average Closing Price": round(average_price, 2),
        "Percentage Change": round(percentage_change, 2),
        "Highest Price": round(highest_price, 2),
        "Lowest Price": round(lowest_price, 2),
    }
```

This fetches the historical closing prices for the stock ticker entered, and analyzes the closing prices, performing basic calculations on the closing prices, finding the average, calculating the percentage change, finding the maximum, or highest closing price, and also the minimum, or the lowest closing price.

6.0 CASE STUDY

```
Welcome to the Stock Selection Tool!
Do you want to register or login? (register/login): register
Enter email: teha
Invalid email. Please enter a valid email address.
Enter email: tehaa@gmail.com
Enter password: tehaaa
User registered successfully!
```

- The tool displays a welcome message to the user, and prompts the user to choose to register or login.
- User entered an invalid email (does not contain “@”, so the tool showed an error message and prompted the user to retry.
- The user is registered successfully after entering valid credentials.

```
Welcome to the Stock Selection Tool!
Do you want to register or login? (register/login): login
Enter email: tehaa@gmail.com
Enter password: tehaaa
Login successful!

1. Fetch Stock Data
2. View Saved Data
3. Exit
Enter your choice: 3
Goodbye!
```

- The user chooses to login.
- If the credentials are valid, it shows Login successful and takes the user to the Main Menu.

6.0 CASE STUDY

```
1. Fetch Stock Data
2. View Saved Data
3. Exit
Enter your choice: 1
Enter stock ticker (e.g., 1155.KL): 1155.KL
Enter start date (YYYY-MM-DD): 2021-10-10
Enter end date (YYYY-MM-DD): 2022-10-10
[**********************100%**********************]  1 of 1 completed

Analysis Results for 1155.KL:
Average Closing Price: Ticker
1155.KL    8.61
dtype: float64
Percentage Change: Ticker
1155.KL    6.0
dtype: float64
Highest Price: Ticker
1155.KL    9.12
dtype: float64
Lowest Price: Ticker
1155.KL    7.95
dtype: float64
Data saved successfully!
```

User chooses between the 3 options, user chooses option 1, user has to enter stock ticker, start date, and end date to start the analysis.

```
1. Fetch Stock Data
2. View Saved Data
3. Exit
Enter your choice: 2

Saved Data:
                email   ticker  ...                                Highest Price                                Lowest Price
0  wnurfatihah@gmail.com  1155.KL  ...  Ticker\n1155.KL    10.34\ndtype: float64  Ticker\n1155.KL    10.12\ndtype: float64
1    lynazynal@gmail.com  1155.KL  ...  Ticker\n1155.KL    10.94\ndtype: float64    Ticker\n1155.KL     9.0\ndtype: float64
2       aisyah@gmail.com  1155.KL  ...   Ticker\n1155.KL     9.83\ndtype: float64   Ticker\n1155.KL    9.83\ndtype: float64
3       aisyah@gmail.com  1155.KL  ...   Ticker\n1155.KL     9.83\ndtype: float64   Ticker\n1155.KL    9.83\ndtype: float64
4       aisyah@gmail.com  1155.kl  ...  Ticker\n1155.KL    10.06\ndtype: float64   Ticker\n1155.KL    9.98\ndtype: float64
5        tehaa@gmail.com  1155.KL  ...   Ticker\n1155.KL     9.12\ndtype: float64   Ticker\n1155.KL    7.95\ndtype: float64

[6 rows x 6 columns]

1. Fetch Stock Data
2. View Saved Data
3. Exit
Enter your choice: 3
Goodbye!
```

- User chooses option 2, the tool displays saved data from past analyses.
- User chooses option 3, the program closes.