**Project Synopsis: Finance Insights Dashboard**

# **Overview**

The "Finance Insights Dashboard" project aims to provide an automated, comprehensive solution for extracting, analyzing, and presenting financial data in a user-friendly web interface. By leveraging web scraping techniques, data analysis, and interactive visualizations, this project delivers timely financial insights tailored to client needs.

# **Objectives**

- Automate Data Extraction: Utilize web scraping to gather financial data from selected websites efficiently.

- Data Analysis and Visualization: Analyze the extracted data and create insightful visualizations to highlight key financial trends and metrics.

- User-Friendly Web Interface: Develop a responsive and interactive website for clients to easily access and interpret financial reports.

- Data Storage and Management: Implement a robust SQL database to store and manage the scraped financial data securely.

# **Scope**

## 1. User Management:

- User Registration and Authentication:

- Implement user registration for clients.

- Secure authentication process using username and password.

- User Profile Management:

- Allow users to update their profile information.

- Manage user settings and preferences.

- Role-Based Access:

- Define roles for admin and clients.

- Admins manage the system and client access.

- Clients access and view financial insights.

## 2. Data Extraction and Storage:

- Web Scraping:

- Develop scripts to extract financial data from one selected website.

- Data Storage:

- Design a SQL database schema to store the scraped data.

- Implement data insertion, updating, and retrieval mechanisms.

## 3. Data Processing and Analysis:

- Data Cleaning and Preprocessing:

- Clean and preprocess the scraped data using Pandas.

- Handle missing values, inconsistent formats, and duplicates.

- Data Analysis:

- Perform basic analysis to identify key financial metrics and trends.

- Calculate metrics such as averages, growth rates, and comparisons.

## 4. Visualization and Reporting:

- Interactive Visualizations:

- Create key interactive charts using D3.js or Chart.js.

- Types of visualizations may include line charts, bar charts, and pie charts.

- Reports Generation:

- Generate summary reports displaying key financial insights.

## 5. Web Application Development:

- Backend Development:

- Implement the backend using Flask to serve data and handle user requests.

- Develop API endpoints for data retrieval and user management.

- Frontend Development:

- Create a responsive web interface using HTML, CSS, and JavaScript.

- Design intuitive navigation and layouts for easy data interpretation.

- Integration of Visualizations:

- Embed interactive visualizations into the web interface.

- Ensure seamless interaction between frontend and backend components.

## 6. Notifications and Updates:

- Data Update Notifications:

- Notify users about the latest data updates and reports.

- System Alerts:

- Provide alerts for system issues or updates to users and admins.

# **Type of Project**

Data Science & Web Application

# **Technology Used**

- Python: Employed for web scraping and data analysis. Libraries such as BeautifulSoup and Scrapy will be used for scraping, while Pandas and Matplotlib/Seaborn will handle data analysis and visualization.

- HTML, CSS, and JavaScript: Used to create a responsive and interactive front-end for the web interface. JavaScript libraries like D3.js or Chart.js will be utilized for dynamic visualizations.

- SQL: A relational database will be used to store and manage the financial data securely. SQL queries will facilitate efficient data retrieval and manipulation.

# **Team Member**

Shreyash Bhardwaj