Practical No. 07

Aim: To write test cases to validate the requirements of the AI Stock Analyzer project based on its problem statement and System Requirements Specification (SRS) document.

Software testing is a process for validating and verifying the working of a software application. It ensures that the software is working without any errors or bugs and provides the expected output to the user. The software testing process isn't limited to finding faults but also includes finding measures to upgrade the software in factors such as efficiency, usability, and accuracy. To test software effectively, a structured format called a **Test Case** is used.

What is a Test Case?

A test case is a defined format for software testing required to check if a particular feature of an application is working as intended. A test case consists of a set of conditions, steps, and inputs used to verify the expected outcome against the actual outcome. It is a fundamental unit of testing that helps ensure quality and correctness.

A test case consists of various parameters, which are outlined below.

Parameters of a Test Case

- **Module Name**: The name of the module or feature being tested (e.g., User Authentication, Dashboard).
- **Test Case ID**: A unique identifier for the test case (e.g., TC_AUTH_01).
- **Test Case Description**: A brief description of what is being tested.
- **Test Priority**: The importance of the test case (e.g., High, Medium, Low).
- **Prerequisite**: Any conditions that must be met before executing the test steps.
- **Test Steps**: The specific sequence of actions to be performed to execute the test.
- **Test Data**: The input values used for the test.
- **Expected Result**: The anticipated outcome of the test.
- **Actual Result**: The real outcome observed after executing the test steps.
- **Status**: The result of the test (Pass, Fail, Not Executed).
- **Comments**: Any additional notes or remarks.

Conclusion

The design documents and test cases created for the **AI Stock Analyzer** project are fundamental tools for its successful development and deployment. The **E-R Diagram** provides a clear blueprint for the database structure, while the **Data Flow and Sequence Diagrams** map out the system's logic and the interactions between its components. Together, these diagrams ensure that the system's architecture is well-planned and efficient.