**Requirement Gathering and Analysis Phase**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 10 April 2025 |
| Team ID | SWITD1743598785 |
| Project Name | Stock Trading Web App |
| Maximum Marks | 4 marks |

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form Registration through Gmail Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | User Login | Login via Form Login via Social Media Accounts |
| FR-4 | User Portfolio Management | View Portfolio Edit Portfolio |
| FR-5 | Stock Market Information | Real-time Stock Quotes Historical Data |
| FR-6 | Trading Functionality | Buy Stocks Sell Stocks |
| FR-7 | Alerts and Notifications | Price Alerts News Updates |
| FR-8 | Reporting and Analytics | Performance: The application must load quickly and handle a high number of simultaneous requests. Implement performance optimizations such as caching (e.g., Redis), content delivery networks (CDN), and efficient database queries. Aim for load times under 2 seconds. |

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | Usability | The interface must be intuitive and easy to navigate for users of all levels. This includes clear navigation menus, accessible help resources, and a clean design layout. |
| NFR-2 | Security | Implement robust security measures to protect user data and transactions. This includes encryption (SSL/TLS), secure authentication mechanisms (e.g., OAuth2, JWT), regular security audits, and compliance with industry standards like OWASP. |
| NFR-3 | Reliability | The system should consistently perform as expected without failures. Implement failover ensure data integrity and application stability. |
| NFR-4 | Performance | The application must load quickly and handle a high number of simultaneous requests. Implement performance optimizations such as caching (e.g., Redis), content delivery networks (CDN), and efficient database queries. Aim for load times under 2 seconds. |
| NFR-5 | Availability | Ensure the system is available 99.9% of the time. Use cloud services with multi-region deployments, load balancers, and regular maintenance schedules to minimize downtime. |
| NFR-6 | Scalability | The system should be able to handle increased loads without performance degradation. Implement horizontal scaling for web servers and databases, use microservices architecture where applicable, and utilize cloud infrastructure to dynamically adjust resources based on demand. |