SYSTEM REQUIREMENTS

03 : AUTONOMOUS TRADING BOT

<TEAM MEMBER NAMES & IDS>

STUDENT ID	NAME
23100197	AHMED TAHIR SHEKHANI
23100011	SULEMAN MAHMOOD
23100198	ALI ASGHAR
23100176	SYED TALAL HASAN

TABLE OF CONTENTS

1. Introduction

A web application with an autonomous trading bot instance running at the backend that can generate profitable returns with the least amount of human intervention using Mathematical models, concepts of Game Theory, Financial Techniques and Artificial Intelligence to predict the price pattern and automatically trade in order to gain first-mover advantage for sudden peaks in the price movement. Users will use the frontend to run bot instances with required configuration and display the records of the trade.

The overall objective for the application would be to achieve the return target provided by the analyst while configuring the bot and minimize loss according to the risk factor provided. The potential users of this application would be trade analysts or managers who will use the bot to run its instances according to their requirements.

Technical details:
The project's tech stack would be Next JS for web application UI particularly frontend, Flask for backend server, and Postgresql for persistent storage. The application would follow three-tier architecture with a repository pattern for the persistent layer, models, and command layer for modifying the state.

2. System Actors

Actor Name	Description
Analysts	Analysts will configure and start the execution of the bot, analyze it's statistics and assign investors to their bots.
Investors	Investors will use the bot for investment purposes and can view a summary report of the performance of the bot for the time period

3. Functional Requirements

Requirements	
Sr#	**Requirement**
1	As an analyst, I want to be able to login with my credentials.
2	As an analyst, I want to be able to view my dashboard.
3	As an analyst, I want to be able to configure the parameters (such as duration of execution, balance, maximum drawdown, target return, list of stocks) on which the bot will execute.
4	As an analyst, I want to be able to initiate the execution of the bot.
5	As an analyst, I want to be able to stop the execution of the bot.
6	As an analyst, I want to be able to instantiate new instances of the bot based on different configurations.
7	As an analyst, I want to be able to register an investor who can then view their bot's performance
8	As an analyst, I want to be able to assign one or more bots to an investor
9	As an investor, I want to be able to login to my dashboard with the credentials provided by my analyst.
10	As an investor, I want to be able to see reports / statistics of my bot's performance.

4. Non-functional Requirements / Quality Attributes

Sr#	Requirements
1	The system should complete execution of any request within 1 minute.
2	The system should completely and accurately respond to requests at least 90% of the time it is invoked.
3	The time to get current stock prices from pakistan stock exchange should not be more than 5 seconds
4	The system should be available from Monday to Friday 9 am to 5 pm.

5. Security Requirements

Sr #	Security Risks	Potential Losses	Controls
1	Broken Access Control	- Unauthorized accessed users can use the amount from the main user's wallet for investment with high risk and might result in loss. - Access of API with missing access controls for Post, Put, and Delete result in loss of data for reporting and misconfiguration of the bot as unauthorized users can manipulate the saved configuration of the bot in the application.	- Allow users to access the records associated with their unique ID. - Log access failures and alert admin if failed multiple times. - Use stateless JWT with short duration
2	Cryptographic Failures	- Unauthorized users can steal the session cookie by intercepting requests. He can then replay this session cookie to access or modify the bot by hijacking the session. - Unauthorized users can gain access to the password database through a	- Enforce TLS for all pages and require strong encryption. - Use complicated hash methods which should then further be salted so even if someone gains access to the password database, they can't make use of it.

		file upload flaw and can gain access to unsalted hashes, which can be exposed with pre-calculated hashes from a rainbow table.	
3	Security Misconfiguration	- Unauthorized users may access the machine learning model and make the same predictions as the hedge fund manager, resulting in lower financial profits. - A misconfigured database of user credentials can lead to malicious users to gain access to the main application and its prediction model which can be misconfigured to return false predictions, resulting in huge losses for the hedge fund.	- Segmented Application with isolation of the model from the main components that requires special security access. - Remove unnecessary features and port access.

Sr#	Security Tool Name	Brief description (why the tool is suitable for your project)
1	Mend	- Can implement using CI/CD - It provides security over 70 CWE types including OWASP top 10 security issues. - It is easy to use, setup and administer - Since, our application is based on real time data management so fast scanning will be required for testing and for that mend is suitable as it is typically 10 times faster than other static security scanning tools.
2	Indusface WAS	- Uses globally accepted best practices. - Can be used for fully managed risk detection for the application.

6. Who Did What?

Name of the Team Member	Tasks done
Suleman Mahmood	Functional requirements (3), Actors (2)
Ahmed Tahir Shekhani	Security requirements (5), Introduction (1)
Talal	Security requirements (5), Introduction (1)
Ali	Non-functional requirements (4), Security requirements (5)

Note: Numbers in (x) means the section

7. Review checklist

Section Title	Reviewer Name(s)
Introduction	Ali Asghar
Actors	Ahmed Tahir Shekhani
Functional Requirements	Ali Asghar, Ahmed Tahir Shekhani
Non-functional requirements	Talal
Security Requirements	Suleman Mahmood