Akhil Bharatiya Maratha Shikshan Parishad's

Anantrao Pawar College of Engineering & Research

Record No.: ACA/D/003A **Revision: 00**	**DoI: 01/02/2025**

Synopsis and Project Approval sheet

Synopsis Report

1. Group No (Project Group Information)

Roll No	Student Name	Contact No	Email –Id	Sign
M2198	Ganesh Nivrutti Wagh	9270029991	ganesh.wagh3003@gmail.com	

2. Title of the Project(Problem Statement/ Title)

StoxPort: Investment Portfolio Tracker

StoxPort is designed as a web-based solution to empower individual investors to manage and monitor their investment portfolios. The problem it addresses is the need for an integrated, real-time system that not only tracks stock prices but also provides performance insights, smart alerts, and social engagement features.

3. Abstract(System Overview)

StoxPort is a FinTech web application built to streamline portfolio management. It allows users to add, edit, and delete stock holdings while tracking real-time portfolio values through live stock price integration. The system features a dynamic dashboard that displays key metrics such as total portfolio value, top-performing stocks, and portfolio distribution. Additionally, it offers smart alerts, watchlist recommendations, technical trends, future forecasts, news feed integration, and social sharing—coupled with a modern, responsive interface including dark mode for enhanced usability.

4. Project Purpose, Scope and Objectives

Purpose:

The project aims to develop a responsive and user-friendly web application that enables efficient management of investment portfolios with real-time data tracking and advanced analytical features.

Scope:

- **User Management:** Ability to add, edit, and delete stock holdings.
- **Data Integration:** Real-time stock price updates using free stock price APIs.
- **Analytics Dashboard:** Visual presentation of portfolio metrics, performance graphs, and news updates.
- **Enhanced Features:** Smart Alerts, Watchlist Recommendations, and social sharing options.
- **Security:** Implementation of Two-Factor Authentication (2FA) for robust user security.

Objectives:

- Develop a responsive frontend using React.
- Build a robust backend with Spring Boot and Java.
- Integrate a PostgreSQL database for reliable data storage.
- Leverage free stock price APIs for real-time data.
- Deploy the backend on Render and the frontend on Vercel/Netlify.

5. Software & Hardware Specification

Software Specifications:

- **Frontend:**
    - **Framework:** React
    - **Styling:** Tailwind CSS, Shadcn/UI
    - **Deployment:** Vercel or Netlify
- **Backend:**
    - **Framework:** Spring Boot (Java)
    - **API:** RESTful services for CRUD operations and portfolio calculations
    - **Deployment:** Render

- **Database:**
    - **Type:** PostgreSQL (relational database) on docker
    - **Design:** Structured schema for users, stock details, and portfolios
- **APIs:**
    - Integration with free stock price APIs such as Alpha Vantage, Finnhub, etc.
- **Development Tools:**
    - **IDE:** VS Code (frontend), IntelliJ (backend)
    - **Version Control:** Git, GitHub
    - **Project Management:** Jira
    - **Testing:** Postman (APIs), Selenium (UI)

Hardware Specifications:

- **Client Side:**
    - Any modern device (desktop, laptop) capable of running a web browser.

- **Server Side:**
    - Cloud-based hosting services (e.g., Render) will typically use scalable cloud infrastructure.
    - Standard requirements include multi-core processors, sufficient RAM (4GB or higher), and scalable storage solutions to handle real-time data processing and secure user transactions.

Brief description of technology used (Front End/ Back End)

Frontend:

- **Framework:** React is used to build a dynamic, component-based user interface.
- **Styling:** Tailwind CSS and Shadcn/UI ensure a responsive and modern look.
- **Features:** The UI includes a dashboard displaying portfolio metrics, interactive forms for managing stocks, and options like dark mode for user customization.

Backend:

- **Framework:** Java with Spring Boot forms the core of the backend, managing business logic and RESTful API development.

- **API Design:** REST APIs support CRUD operations, error handling with meaningful HTTP status codes, and portfolio value calculations.
- **Database Interaction:** Utilizes JPA/Hibernate for seamless communication with the PostgreSQL database.
- **Security:** Incorporates Two-Factor Authentication (2FA) for enhanced security.

6. Literature Survey

The literature review encompasses:

- **Existing FinTech Applications:**
  Comparative studies with popular platforms such as Yahoo Finance, Robinhood, and Personal Capital to identify strengths and gaps.
- **Real-Time Data Integration:**
  Analysis of free stock price APIs like Alpha Vantage and Finnhub, discussing their capabilities and limitations**.**
- **Security Measures:**
  Studies on Two-Factor Authentication (2FA) methods, particularly in microservice architectures, to ensure secure user authentication.
- **UI/UX Trends:**
  Exploration of modern design frameworks (e.g., Tailwind CSS, Material-UI) to ensure the application remains intuitive and responsive.
- **Additional Research:**
  Reference texts and research papers on full-stack development, Spring Boot, and real-time portfolio optimization provide theoretical backing and technical insights.

7. Project Requirement and Planning

Functional Requirements:

- **Stock Management:** Add, view, edit, and delete stock holdings.
- **Real-Time Calculations:** Dynamic portfolio value computation based on live data.
- **Dashboard:** Comprehensive view with performance graphs and key metrics.
- **Alerts & Recommendations:** Smart alerts and watchlist recommendations based on stock performance.
- **Additional Features:** News feed integration, social sharing, and dark mode.
- **Security:** Two-Factor Authentication (2FA).

Non-Functional Requirements:

- **Responsiveness:** A user-friendly, responsive design that works across devices.
- **Security & Scalability:** Secure backend with scalable architecture.
- **Database Efficiency:** Optimized schema design and efficient database interactions.
- **Reliability:** Robust exception handling and clear HTTP status codes.

Planning Phases:

1. **Phase 1:** Requirement Gathering & Technology Finalization – Define requirements, use cases, and select the tech stack (1 Week).
2. **Phase 2:** Backend & Database Development – Design database schema and develop RESTful APIs using Spring Boot (3–4 Weeks).
3. **Phase 3:** Frontend Development & API Integration – Build the React-based UI and integrate with backend services (2–3 Weeks)**.**
4. **Phase 4: Testing & Deployment** – Conduct thorough testing (API, UI, end-to-end) followed by deployment on Render (backend) and Vercel/Netlify (frontend) (2 Weeks).
5. **Finalization:** Bug fixes and optimization (1 Week).

8. Expected Outcome

Fully Functional Web Application: A seamless platform where users can manage their investment portfolios with live data integration.

Real-Time Tracking: Dynamic updates on portfolio valuation, performance graphs, and key financial metrics.

User Engagement: Features like smart alerts, watchlist recommendations, news updates, and social sharing to enhance decision-making.

Robust Security: Implementation of 2FA to ensure user data safety.

Comprehensive Documentation: Complete user guides, API documentation, and a detailed README for deployment and future enhancements.

9. References

React Documentation: https://react.dev/

Tailwind CSS Documentation: https://tailwindcss.com/

Spring Boot Documentation: https://spring.io/projects/spring-boot

MySQL Documentation: https://dev.mysql.com/doc/

Alpha Vantage API: https://www.alphavantage.co/

Finnhub API: https://finnhub.io/

Render Deployment Guide: https://render.com/docs

Vercel Deployment Guide: https://vercel.com/docs

Books:

  - Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker – Frank Zammetti
  - Spring Boot in Action – *Craig Walls*
  - Financial Technology: Case Studies in FinTech Innovation – *Niels Pedersen*

- **Research Papers:**
  - "Real-Time Portfolio Optimization Using Machine Learning" – *IEEE International Conference on FinTech*
  - "Two-Factor Authentication in Microservices: A Comparative Study" – *ACM Transactions on Cybersecurity*

Prof. Purvesh Wagh	Prof. S.A.Mutha	Dr. A.D.Newase
Project Guide	**Project Coordinator**	**HOD**