COSC 612/ AIT 624, SPRING 2025

PROJECT TITLE: FABRIC INVENTORY MANAGEMENT SYSTEM

TEAM NAME: CODE CRAFTERS

TEAM MEMBERS:

HALEEMAH AMISU - hamisu1@students.towson.edu

JOSEPH FRISHKORN - jfrish1@students.towson.edu

AASTHA BHATT - abhatt6@students.towson.edu

BHUVAN ANGIRAA - bangira1@students.towson.edu

SAI MANASWINI UTLA – sulta1@students.towson.edu

March 16th, 2025

SECTION 2: TEAM MEMBER RESUMES

Haleemah T. Amisu
tayoamisu68@gmail.com | (443)-370-3259

Summary

Results-driven software professional with a strong foundation in software development, quality assurance, and test planning. Experienced in collaborating with cross-functional teams to ensure the delivery of high-quality software solutions. Proficient in Python programming, with familiarity in C++ and software development best practices. Committed to continuous learning and tackling new challenges.

Experience

Experis Baltimore, MD
Data Quality Analyst July 2022 – August 2024

* Implemented data profiling and cleansing processes, identifying and resolving data anomalies, leading to a 15% reduction in data errors.

● Frequently moderate training and alignment sessions with teams throughout the ecosystem to improve collection quality and manage communication
● Validated data integrity and maintained a 98% accuracy rate.
● Conducted root cause analysis for data quality issues, implementing corrective actions

Experis Baltimore, MD
Media Quality Tester Nov 2021 – June 2022

● Conducted comprehensive quality assurance tests for over 1000 research participants on media content, including videos, audio files, and interactive applications
● Developed and executed test plans and test cases, ensuring thorough coverage and reducing testing time by 25% through effective test execution strategies
● Documented and tracked defects, following up with the development team to ensure timely resolution

Projects

Attendance Management System June 2020

● Revolutionized Alcorn State University Math Center Attendance Tracking with PHP, MySQL, JavaScript, jQuery, and Ajax to create a multifunctional attendance management web application
● Implemented Fingerprint authentication and automatic syncing features to the web application

Skills

● Programming Languages: Python, C++, Java
● Database Management: SQL
● Web Development: HTML, CSS, JavaScript
● Test Planning and Execution
● Data Analysis
● AGILE Methodologies

Education

Towson University Towson, MD
Master of Science in Computer Science

Alcorn State University Lorman, MS
Bachelor of Science in Computer Science

Additional Information

● Proficient in Linux/Darwin command line environments
● Experience with software development best practices, version control, and continuous integration
● Strong analytical and problem-solving abilities
● Excellent communication and interpersonal skills

Aastha Bhatt

Bachelor of Computer Information Systems| Concentration: Data Analytics
Accelerated Bachelor's to Master's Program
Masters of Computer Science| Concentration: Data Science

SKILLS

- **Computer:** Microsoft Office Suite (Excel, PowerPoint), macOS, Tableau, Google Suite, Salesforce, Form Assembly, Weka
- **Coding Languages:** R
- **Languages:** English (Native)

PROJECT

Goal: Improve the Churn Rate of a Credit Card Company

- Scrub the Bank Churn dataset with 10,000 records using R and Excel.
- Run 5 tests among the characteristics of customers: Multiple Linear Regression, Independent Sample T-test, and One-Way ANOVA.
- Visualize the information gained from the data cleaning using R and Tableau.
- Found that the company should market more toward married couples and young individuals, as they are more willing to stay long-term in the company.

- **Goal: Identify if Buying Musical Instruments from Amazon is Worth it (Text Analytics)**
- Scrub over 500 Amazon Reviews by removing stop words, emojis, whitespaces, etc.
  - Run Sentiment Analysis to understand the 8 overall emotions of customers.
  - Perform topic modeling to understand the reasoning behind each of the 5 rating categories.
  - Found that buying musical instruments from Amazon might not be beneficial; however, it is worth buying instrument accessories such as the 2 most commonly bought: amPlug and washers.

Goal: Identify the Majors that Yield the Highest Salary vs. Non-STEM Majors

- Scrub the Salary of each major dataset with 500 records using R and Excel.
- Run classification models: Naive Bayes, K Nearest Neighbor, Decision Tree, Random Forest, Bagging, Boosting.
- Cluster data and find similarities between the 25th and 90th percentile for STEM and Non-STEM majors.
- Found that one of the most significant differences in the salary difference between STEM and Non-STEM majors is right at the beginning of the careers, as STEM majors are more likely to make more than $50,000.

EXPERIENCE

Towson University Office of Technology - Salesforce Support Technician |Towson, MD
Aug. 2021 – Aug. 2023

- Configured the Salesforce platform to meet over 150 client needs by creating custom objects, fields, events, and reports.
- Feed data into Salesforce efficiently by making 50+ forms with well-thought-out Form Assembly mapping.
- Clean over 1,000 records of data in Salesforce using Demand Tools.
- Recommend the best solution to clients based on needs by holding 1:1 meetings.

SAI MANASWINI UTLA
Email: sutla1@students.towson.edu | Mobile: +1 (667) 308-1413

TECHNICAL SKILLS

- **Languages**: Python, Java
- **Cloud**: Google Cloud Platform (GCP)
- **Frameworks**: Flask, Django
- **Databases**: MySQL, LDAP
- **Front-end**: HTML, CSS, JavaScript, jQuery, Bootstrap

PROFESSIONAL EXPERIENCE

Infosys Ltd
DEC 2021 - DEC 2024
Project: Contact Center Systems | **Client**: Bright speed

- Developed cloud functions, cloud run services, and automated data fetching into GCP buckets using Python.
- Created triggers, schedules, and cloud-based processes to streamline workflows.

Project: TQSG Substrate | **Client**: Cisco Systems Inc

- Implemented LDAP operations (CRUD) and administrative tasks using Python scripts.
- Automated processes using Ansible and Jenkins, enhancing workflow efficiency.
- Provided production support and code fixes for Cisco applications.

Ojas Innovative Technologies
DEC 2020 - NOV 2021
Project: Steris | **Client**: Steris

- Developed Flask-based solutions to manage vendor credentialing processes for healthcare compliance.
- Integrated database queries and managed CSV data manipulation.

MedXperts
OCT 2019 - DEC 2020

- Built REST web services for a telemedicine app using Flask and MySQL.
- Developed modules for appointment booking, medical records, and medicine notifications.

ETG Global Services
JUNE 2018 - SEPT 2019
Project: Funimation | **Client**: Sony Pictures Entertainment

- Developed new features and customized Django-based applications.
- Implemented front-end UI enhancements using jQuery and tested modules with Protractor.

Joseph W. Frishkorn

Frederick, MD 21701 | (858) 275-4190 | jwf2127@columbia.edu

Education

MASTER OF SCIENCE (M.S.), COMPUTER SCIENCE
Towson University, Towson, MD | Expected May 2026 | GPA: 3.67

TRAINEE | NPOWER | FEB 2024 – JULY 2024
Instructional program designed to launch tech careers through hands on training and projects.

BACHELOR OF ARTS (B.A.), AMERICAN STUDIES
Columbia University, New York, NY | May 2022 | GPA: 3.68

ASSOCIATE OF ARTS (A.A), GENERAL STUDIES
Frederick Community College, Frederick, MD | Dec 2017

Certifications

- **CompTIA IT Fundamentals (ITF+), CompTIA | Apr 2024**
- **Google IT Support, Google | May 2024**

Experience

PLATOON SERGEANT | U.S. MARINE CORPS | MAY 2016 – SEPT 2017
- Developed and executed monthly maintenance plans and technical training schedules.
- Mentored and supervised 38 Marines to ensure their well-being and achievement of professional goals.

QUALITY ASSURANCE REPRESENTATIVE | U.S. MARINE CORPS | MAY 2016 – SEPT 2017
- Enforced maintenance practices for a safe working environment under strict Naval regulations.
- Oversaw and maintained inventory of Air Traffic Control landing instruments valued at $2.8M.

WORK CENTER SUPERVISOR | U.S. MARINE CORPS | MAY 2016 – SEPT 2017
- Liaised with customers, employees, and stakeholders to resolve inquiries and manage order fulfillment.

HAZARDOUS MATERIAL MANAGEMENT | U.S. MARINE CORPS | MAY 2015 – SEPT 2016
- Optimized Maintenance Department HAZMAT storage facilities and HAZWASTE collection points.

ELECTRICAL MAINTENANCE TECHNICIAN | U.S. MARINE CORPS | SEPT 2012 – SEPT 2017
- Diagnosed and repaired faulty equipment using test equipment and technical manuals.

Awards

- Navy and Marine Corps Achievement Medal, ATC Det B, MACS-1 | Dec 2016
- Certificate of Commendation (4), ATC Det B, MACS-1 | April 2017
- Good Conduct Medal, ATC Det B, MACS-1 | Sept 2015

Bhuvan Angiraa

+1 610-342-5505 - bhuvanarr007@gmail.com
1773 Amuskai Road, Parkville, Maryland, 21234

PROFILE

I am a dedicated MS in Computer Science student passionate about expanding my skills in data science. I am eager to learn and apply advanced analytical techniques to solve real-world problems. My goal is to leverage my expertise to create innovative solutions and drive business success in the data science field.

PROFESSIONAL EXPERIENCE

Acruex India Private Limited **Nov 1st, 2023 - March 31st, 2024**
Database Administrator Intern

My responsibilities were to manage and maintain database, ensuring their performance, security, and availability, while working alongside with the team. I was responsible for database installation, configuration, backup, and recovery. I also monitored system performance and implement security measures to protect sensitive information.

EDUCATION & CERTIFICATIONS

Masters of Science in Computer Science
Towson University Present-2026

Bachelors of Technology in Information Technology
IIMT College of Engineering 2019-2023

AWS Solution Architect Udemy, 2022

DevOps Fundamentals Udemy, 2022

Digital Marketing Google, 2021

LANGUAGES

Fluent with C++ and Python

Good with MS-SUITE

Good knowledge of Amazon Web Services and Microsoft Azure

Firewall Protections From The 3rd Party Applications
ACADEMIC PROJECT **2023**

I researched factors contributing to data leaks, emphasizing financial losses, and proposed strategies to enhance security and minimize risks. I developed the concept of a user-friendly, mobile firewall for continuous protection of user credentials and data. This work was published in the International Journal of Scientific Research in Science and Technology in 2023.

KEY COMPETENCIES

Data-driven strategic planning
Ability to work independently.
Team Player

Report writing and presenting
Critical thinking skills.

Proactive and self-motivated
Exceptional organizational skills
Excellent communication skills

SECTION 3: PLANNING AND SCHEDULING

The team, Code Crafters, held its initial meeting on February 21, 2025, to select a project, assign roles, and review Sprint 1 requirements. The team also established a regular meeting schedule and agreed upon primary communication channels to ensure efficient collaboration.

Meeting Schedule

- **Sprint 2 Kickoff:** March 2, 2025
- **Weekly After-Class Check-ins:** Mondays at 7:00 PM (after class, brief sync-up)
- **Primary Weekly Meeting:** Sundays at 7:00 PM (longer planning and work session)
- **Platform:** Discord (voice and text channels)

Communication Plan

- **Primary Communication:** Group text message (for quick updates and scheduling)
- **Secondary Communication:** Discord (for meetings and real-time collaboration)
- **Document Collaboration:** Microsoft Word for drafting written content
- **Code Management:** Shared GitHub repository (SWE- Code Crafters)

Work Breakdown Structure (WBS) for Sprint 2

The Work Breakdown Structure table below outlines each team member's responsibilities, estimated effort, dependencies, and due dates for Sprint 2.

Assignee Name	Task	Duration (hours)	Dependency	Due Date	Status
Haleemah Amisu	Update Project board, compile report	3	Team members complete assigned task for report compilation.	03/16/2025	complete
Joseph Frishkorn	Write out use cases and user requirements	3	none	03/16/2025	Complete
Aastha Bhatt	Create class Diagrams	3	User Requirements, Use Cases	03/16/2025	Complete
Sai Manaswini (Coordinator)	Update problem statement (Add functional and nonfunctional requirements), write User requirements, create use case diagram	3	None	03/16/2025	Complete
Bhuvan Angiraa	Database specification and Analysis	3	Database Tables (Attributes, relations, primary key and foreign key)	03/16/2025	Complete

SECTION 4: TEAMWORKS BASICS SUMMARY

Teamwork Basics Summary

1. **Work Norms:** In a group project, it's important to divide tasks efficiently. Effective work distribution ensures the project progresses smoothly. Assigning tasks based on individual strengths and workloads will help to guarantee the team meets its deadlines while also satisfying its goals. For example, if your team is working on a project and one member is more technically proficient while another is better suited for administrative roles, it may be best to assign the first team member to handle troubleshooting while the second member manages documentation. Deadlines will be collectively agreed upon, and the team coordinator will track progress. If someone misses a deadline, we will first check in and redistribute tasks if necessary. Quality checks involve peer review and a two-person integrity system. In a past group coding project, one teammate wrote the code while another reviewed it for errors. If disagreements arise over quality, the team will discuss standards upfront to avoid misalignment, with the team coordinator having final say. Recognizing different work habits is crucial. I prefer starting early, while others might work best under deadline pressure. We will compromise by setting incremental deadlines to keep everyone on track and committing to an open communication within the team to avoid confusion concerning deadlines.

2. **Facilitator Norms:** Having a facilitator will help the team to maintain its direction and focus, especially during weekly meetings. For this project, the facilitator will rotate for each sprint. Their role will include setting agendas, guiding discussions, and keeping meetings efficient. For example, in a previous course, our project facilitator ensured that all voices were heard, and discussions stayed on topic, preventing dominant personalities from overshadowing others and keeping meetings on track and substantive.

3. **Communication Norms:** Clear and consistent communication is essential. The team has established Discord as our preferred communication method for the weekly virtual meetings but is subject to change if necessary. The group will meet Friday evenings at 7:00 p.m. unless otherwise planned. Additional meetings will be scheduled on an as-needed basis. Email is preferred for detailed updates, while text or instant messaging in the Discord server is acceptable for quick questions. In my experience with previous projects, platforms like Discord, Slack and WhatsApp, were our primary tools for daily check-ins, while emails were used for formal submissions. This system was simple and effective, and so it will be adopted unless it proves to be unsuccessful. Regular updates prevent confusion and ensure accountability. If a message goes unanswered for over 24 hours, a follow-up will be standard practice.

4. **Meeting Norms:** Scheduling meetings can be tricky with varying availabilities. For the time-being, the group will meet virtually every week on Fridays at 7:00 p.m. The Team Coordinator

will be responsible for confirming the team's availability, and if necessary, will reschedule the weekly meeting so that the majority of the group may meet. Weekly meetings will be scheduled for 1.5 hours, unless all goals of the meeting are accomplished before the allotted time. Additional meetings will be scheduled at the group's discretion, with the Team Coordinator having final say. In one of my courses, we had a designated coordinator who set up meetings. Meetings should be held in a distraction-free environment. Punctuality is important; if someone is late, they should catch up without disrupting the discussion, but every effort should be made to arrive on time or to notify the team beforehand about a late arrival. If a member misses multiple meetings, we address the issue privately to find a solution, such as adjusting their responsibilities or our meeting schedule.

5. **Consideration Norms:** Creating a respectful and inclusive environment is key. Basic courtesies like avoiding loud eating, keeping discussions balanced, and being mindful of others' comfort levels make teamwork smoother. Members should feel comfortable offering their thoughts and opinions while at the same time being mindful to maintain a respectful, professional, and academic atmosphere. In a recent internship, one teammate often dominated discussions. To address this, we implemented a speaking time limit so everyone could contribute equally. If norms need to change, an open conversation should take place. The goal is to ensure every team member feels valued and respected while maintaining productivity.

Hints for Handling Difficult Behavior

Sometimes, team members may exhibit difficult behaviors that challenge group dynamics. Below are a few common difficult behaviors and possible solutions:

1. **Dominating The Conversation**: This person takes over discussions and does not let others contribute.

Solution: Set guidelines for balanced participation, such as time limits for speaking. Encourage quieter members to share their thoughts. If they are not as comfortable in a group setting, they may contribute via email, though it is still important to speak during meetings.

2. **Procrastination**: A team member constantly delays completing their work, affecting project timelines.

Solution: Set incremental deadlines and check in regularly. Assign accountability partners to encourage progress.

3. **Perfectionism**: Insists on redoing work multiple times, delaying progress.

Solution: Set clear quality expectations upfront. Agree on a review process with reasonable limits on revisions.

4. **Negative Think**: Constantly complains and discourages team efforts.

Solution: Redirect negativity by focusing on solutions rather than problems. Acknowledge concerns but encourage a proactive approach.

5. **Poor Attendance**: Consistently misses meetings and does not contribute.

Solution: Address the issue directly and understand their challenges. If needed, redistribute tasks or adjust their role.

6. **Resistant to Feedback**: Takes suggestions personally and refuses to adapt.

Solution: Provide feedback constructively, emphasizing team goals rather than individual faults. Encourage a culture of open communication.

Hints for Handling Group Problems

Groups sometimes struggle with productivity, especially when members are unfamiliar with each other's working styles. In a past group project, we used a shared document to track progress and identify roadblocks.

1. **Floundering:** If the team struggles to stay productive, identifying obstacles will be crucial. A good approach is to identify the most important tasks at hand and complete tasks that stand as obstacles to completing next steps. This helped my team in a past internship when we were stuck on troubleshooting an IT issue. By breaking down tasks and assigning roles, we regained momentum. Listing tasks and clarifying objectives in the Work Breakdown Structure will aid in keeping the group on track with tasks.
2. **Going Off on Tangents:** Casual conversation can strengthen team relationships, but excessive distractions hurt productivity. A polite reminder, such as "Let's get back to where we were," helps refocus discussions. In a previous group project, we had a rule to pause side conversations after five minutes.
3. **Making a Decision Too Quickly:** Some team members may push for rapid decisions. To ensure all voices are heard, asking, "Are we ready to decide?" allows for discussion. In my previous projects, we prevented hasty decisions by double-checking team consensus before finalizing plans.
4. **Not Making a Decision:** Consensus-building is key to decision-making. If the team is stuck, we will implement tools like multi-voting, where members vote for their top choices to help narrow options. For small decisions, we will use Plan A where each member will distribute points to rank choices.
5. **Feuding Between Members:** Conflicts can derail progress. The best approach is addressing the issue directly. In my Marine Corps experience, resolving disagreements involved structured discussions where each person shared their perspective before finding common ground.
6. **Ignoring or Ridiculing Others:** Excluding or ridiculing teammates creates tension. Encouraging inclusive collaboration is essential. In a class project, when a member felt sidelined, we started rotating responsibilities to ensure equal participation.
7. **Members Not Doing Their Share:** Addressing non-participation requires direct communication. In one project, a team member repeatedly missed deadlines. We had a private discussion, explaining how it affected the group, and adjusted their tasks to better fit their strengths.

SECTION 5: PROBLEM STATEMENT

What is the Fabric Inventory Management System?
It is a smart tool that helps businesses in textiles, fashion, upholstery, and manufacturing keep track of their fabric stock. It ensures they always know what's available, how much is being used, and when to restock, helping to prevent shortages and reduce waste.

Whom is it for?
It is designed for businesses and professionals involved in fabric-related industries. It is particularly useful for textile manufacturers, fashion designers, wholesale fabric distributors and for the businesses which work or deal with fabric.

What problem does it solve?
It solves problems and challenges faced by businesses handling fabric such as stock mismanagement, manual errors, slow & inefficient working environment, preventing wastage & loss of resources and many more.

What alternatives are available?
There are many alternatives available to this such as manual tracking methods which include spreadsheets and physical logbooks. They also depend on the scale/size of the business and the needs like **Zoho**, which is a cloud-based inventory system used in various industries as well as in the fabric industries.

Why is this project compelling and worth developing?
Solves Real Industry Challenges - stock mismanagement, wastage, and inefficiencies, Reduces Costs & Increases Efficiency, High Demand Across Multiple Industries, Potential for Business Growth & Monetization, Integration with Emerging Technologies **–** use of AI & ML to predict the demand of fabrics.
Primary objectives – Automating fabric tracking, reduce wastage, increasing efficiency, optimizing inventory, enhance decision-making (data driven/demand driven)
Target Customers – Textile manufacturers, retailers, wholesale distributors,
Scope - Includes inventory management, Order & supplier management, multi-user access.

Who are the competitors, and what is novel about your approach?
Fabric Inventory Competitors:
1.Fabric Manager
2.Stitch Lab
3.BlueCherry ERP
4.Fishbowl Inventory
The websites mentioned above follow a similar approach to fabric inventory management, where fabric is entered into the system based on key attributes such as color, type, and quantity. They also track fabric usage and maintain up-to-date information regarding stock levels. These features are designed to ensure efficient inventory management and seamless operations within fabric-related businesses.
However, our website offers several unique features that set it apart:

1. **Chatbot Assistance**: We provide a chatbot that offers immediate assistance to users, helping them address common queries in real time. This feature improves user experience by providing quick, automated support for frequent questions, making it easier for users to navigate the system and resolve issues without needing to wait for manual help.
2. **Restock Requests**: When stock reaches a predefined threshold, our platform automatically sends restock requests to suppliers. This proactive feature ensures that fabric inventory is always replenished on time, reducing the chances of stockouts and improving overall operational efficiency.
3. **Role-Based Profiles and Access Control**: Our system allows users to create profiles based on their roles within the organization, providing customized access and permissions depending on their responsibilities. This feature enhances security and streamlines workflows, as different users can access only the parts of the system that are relevant to their tasks.
4. **Common Features**: Like other platforms, we offer essential features such as:
    a. **Easy Search and Filter**: Users can quickly search, and filter fabrics based on various attributes like type, color, and other criteria, making it easy to find specific materials in large inventories.
    b. **Detailed Fabric Catalog**: A comprehensive catalog displays all the available fabrics, with detailed information about each one, allowing users to view their options at a glance.
    c. **Simple Stock Management**: Adding, editing, or deleting fabric stock is a simple process. Users can update inventory by filling out a straightforward form, ensuring that the system stays current with minimal effort.

These unique and common features together enhance user convenience, improve operational efficiency, and provide better control over fabric inventory management, making our platform a more comprehensive and user-friendly solution.

Clearly demonstrate that the system can be built effectively using available resources and technology:

This website can be built using a combination of available tools and technologies, ensuring a robust, scalable, and responsive platform for managing fabric inventory.

Frontend:

- **HTML, CSS, JavaScript**: These core technologies will form the foundation of the website's user interface, providing a responsive and user-friendly design for staff to easily navigate and manage the fabric inventory. The front end will be optimized for both desktop and mobile devices, ensuring accessibility for all users.

Backend:

- **Python**: The backend is built using Python for its versatility and power. Python's extensive libraries and frameworks are beneficial for integrating machine learning models and handling complex backend processes. The language is widely recognized for its

scalability and maintainability, making it an excellent choice for building the website's core functionalities.

- **Machine Learning**: Python is used to integrate machine learning algorithms, especially for the chatbot feature. These algorithms enable the chatbot to improve its responses over time by learning from the data it processes.
- **Libraries**: We use popular libraries like **TensorFlow**, **PyTorch**, and **scikit-learn** to build and train machine learning models for the chatbot. Supervised learning and reinforcement learning are common approaches employed to improve chatbot performance through continuous training and data analysis.

Database:

- **MySQL**: For efficient database management, we rely on MySQL, which will store and manage all the fabric inventory data. It enables structured queries, ensuring that the system can easily handle large volumes of data and allow users to access, update, and delete fabric stock as needed.

Cloud Hosting and Services:

- **AWS (Amazon Web Services)**: To ensure high performance, scalability, and security, we use AWS for hosting and managing the website. AWS offers cloud-based services that are well-suited for websites with growing demands, ensuring that the platform can scale seamlessly as the user base and inventory size increase.
    - AWS provides essential infrastructure such as virtual servers (EC2), managed databases (RDS), and storage (S3), all of which contribute to maintaining smooth website performance.

Artificial Intelligence or Machine Learning Component:

AI/ML models analyze sales data to forecast demand, with key components including the AI/ML prediction engine. Additionally, conversational AI can interpret user queries, and this capability can also be integrated into a chatbot.

Machine Learning for Chatbot:

- **Chatbot Functionality**: The chatbot, powered by Machine Learning, continuously learns from user interactions. With supervised learning, the chatbot is initially trained on labeled data, allowing it to handle basic queries. Over time, through reinforcement learning, the chatbot adapts and improves its ability to address more complex or context-dependent queries.
    - **Libraries**: TensorFlow, PyTorch, and scikit-learn are used to build the models that allow the chatbot to learn and provide increasingly accurate responses.

By combining these technologies, the website will not only offer seamless fabric inventory management but also provide an interactive, intelligent user experience through the chatbot. The integration of machine learning ensures that the platform remains innovative, efficient, and adaptive to the needs of users.

What makes this project interesting from a technical perspective?

This project is technically fascinating due to its seamless integration of modern web technologies, advanced machine learning, cloud computing, and database management. The frontend, built with HTML, CSS, and JavaScript, offers a responsive and user-friendly interface, while the backend leverages Python for scalability, machine learning integration, and efficient data handling with SQL databases. AWS cloud hosting ensures optimal performance, scalability, and security. Unique features like an AI-powered chatbot, automatic restock requests, and role-based access control enhance user experience and streamline fabric inventory management. The chatbot continuously improves through machine learning algorithms using libraries like TensorFlow and PyTorch, while real-time updates and proactive inventory management further elevate the platform's efficiency. Combining traditional web development with cutting-edge AI techniques, this project creates a highly adaptive, intelligent, and scalable system, making it both technically challenging and innovative.

- **Functional Requirements**: Features and functions that describe what the system should do.
- **Non-Functional Requirements**: Qualities and performance attributes of the system that define how the system should perform.

Functional Requirements	**Non-Functional Requirements**
1. The system should ensure secure user authentication with role-based access.	The system should be secure and support role-based access for different user levels.
2. The system should allow authorized users to add new fabric entries.	The system should have an intuitive user interface for easy navigation.
3. Users should be able to update fabric stock when receiving shipments.	The system should provide quick responses for inventory queries and updates.
4. The system should allow authorized users to remove fabric records if no dependencies exist.	The system should ensure high availability with minimal downtime.
5. Users should be able to quickly search and filter fabric inventory.	The system should scale easily to handle increasing fabric inventory and transactions.
6. The system should generate alerts when stock levels fall below a predefined threshold.	The system should process data securely and comply with relevant data protection regulations.
7. The system should facilitate placing restock orders with suppliers.	The system should be able to handle large datasets for analytical reports without significant performance degradation.
8. The system should generate analytical reports for inventory and sales.	
9. User access should be role-based, limiting actions based on predefined permissions.	

10. The chatbot should assist users with inventory-related queries.	
11. The system should evaluate supplier performance based on historical transactions.	

SECTION 6: SYSTEM REQUIREMENTS

Use Cases

Use Case no: 1	**Use Case Name:** User Login & Authentication
Actors: Staff Member (User)	
Description: A staff member logs into the system using their credentials. Upon successful authentication, they gain access to relevant features based on their role.	
Alternate Path: If login fails, the system prompts the user to retry or reset the password.	
Pre-Condition: The user must be registered in the system.	

Use Case no: 2	**Use Case Name:** Add New Fabric to Inventory
Actors: Inventory Manager	
Description: The inventory manager adds a new fabric item to the system, entering details such as type, color, quantity, supplier, and price.	
Alternate Path: If data is incomplete, the system prompts the user to fill in missing fields.	
Pre-Condition: The user must have proper access to modify inventory.	

Use Case no: 3	**Use Case Name:** Update Fabric Stock
Actors: Inventory Manager	
Description: A user updates the stock level of an existing fabric item after receiving a new shipment.	
Alternate Path: If the fabric does not exist, the system prompts the user to add it as a new entry.	
Pre-Condition: The fabric item must exist in the database.	

Use Case no: 4	**Use Case Name:** Remove Fabric from Inventory
Actors: Inventory Manager	
Description: A user deletes a fabric entry from the system if it is discontinued or no longer available.	
Alternate Path: If the fabric is linked to pending orders, the system restricts deletion and prompts an alternative action.	
Pre-Condition: The user must have permission to remove inventory items.	

Use Case no: 5	**Use Case Name:** Search & Filter Fabric Inventory
Actors: Staff Member	
Description: A user searches for fabrics by various filters such as type, color, supplier, or stock level.	
Alternate Path: If no results are found, the system suggests modifying search criteria.	
Pre-Condition: The database must contain fabric entries.	

Use Case no: 6	**Use Case Name:** Generate Restock Alerts & Requests
Actors: System, Inventory Manager	

Description: The system automatically generates a restock alert when a fabric's stock level falls below the defined threshold. A restock request is then sent to the supplier.
Alternate Path: The inventory manager can manually override the system's suggestion if stock levels are sufficient.
Pre-Condition: A restock threshold must be set for each fabric.

Use Case no: 7	**Use Case Name:** Process Supplier Orders
Actors: Supplier, Inventory Manager	
Description: The system allows users to place orders with suppliers when restocking is needed.	
Alternate Path: If the supplier is out of stock, the system suggests alternative suppliers.	
Pre-Condition: The supplier must be registered in the system.	

Use Case no: 8	**Use Case Name:** View Sales & Inventory Reports
Actors: Business Owner, Inventory Manager	
Description: The system generates reports on fabric sales, inventory status, and usage trends.	
Alternate Path: If no data is available for the selected timeframe, the system informs the user.	
Pre-Condition: The system must have recorded transactions.	

Use Case no: 9	**Use Case Name:** Role-Based Access Control
Actors: Admin, Staff Member	
Description: The system grants different access levels based on the user's role, restricting or allowing actions such as adding/deleting fabrics.	
Alternate Path: If an unauthorized user tries to perform restricted actions, an error message is displayed.	
Pre-Condition: User roles must be predefined.	

Use Case no: 10	**Use Case Name:** Chatbot Assistance
Actors: User, AI Chatbot	
Description: A user interacts with the chatbot to ask questions about fabric availability, restock alerts, and order processing.	
Alternate Path: If the chatbot cannot answer a query, it redirects the user to human support.	
Pre-Condition: The chatbot must have trained responses or access to relevant system data.	

Use Case no: 11	**Use Case Name:** Track Supplier Performance
Actors: Business Owner, Inventory Manager	
Description: The system evaluates supplier performance based on factors like delivery speed, order accuracy, and quality.	
Alternate Path: If data is incomplete, the system prompts users to manually enter performance records.	

Pre-Condition: The system must have stored data on past supplier interactions.

System Requirements

Requirement number: 1	**Use Case Number:** 1
Introduction: The system should ensure secure user authentication with role-based access.	
Inputs: Username, password	
Requirements Description: The system shall validate user credentials against a database and restrict access based on predefined roles. If login fails, the system shall prompt for retry or password reset.	
Outputs: Access granted to user dashboard or error message	

Requirement number: 2	**Use Case Number:** 2
Introduction: The system should allow authorized users to add new fabric entries.	
Inputs: Fabric type, color, supplier details, quantity, price	
Requirements Description: The system shall validate and store fabric details in the database. If incomplete data is detected, a prompt should request missing fields.	
Outputs: Confirmation message and updated fabric inventory	

Requirement number: 3	**Use Case Number:** 3
Introduction: Users should be able to update fabric stock when receiving shipments.	
Inputs: Fabric ID, updated quantity.	
Requirements Description: The system shall allow inventory managers to modify stock levels and maintain logs of changes.	
Outputs: Updated stock levels and confirmation message	

Requirement number: 4	**Use Case Number:** 4
Introduction: The system should allow authorized users to remove fabric records if no dependencies exist.	
Inputs: Fabric ID	
Requirements Description: If the fabric is linked to pending orders, deletion should be restricted with an error message. Otherwise, the record should be removed from the database.	
Outputs: Confirmation of deletion or restriction warning	

Requirement number: 5	**Use Case Number:** 5
Introduction: Users should be able to quickly search and filter fabric inventory.	
Inputs: Search criteria (type, color, supplier, stock level)	
Requirements Description: The system shall retrieve matching fabric entries and display them. If no matches exist, it shall prompt users to refine search criteria.	
Outputs: List of matching fabrics or "no matches found" message	

Requirement number: 6	**Use Case Number:** 6
Introduction: The system should generate alerts when stock levels fall below a predefined threshold.	
Inputs: Fabric stock levels	
Requirements Description: The system shall automatically notify users when restocking is required.	
Outputs: Restock alert notification	

Requirement number: 7	**Use Case Number:** 7
Introduction: The system should facilitate placing restock orders with suppliers.	
Inputs: Fabric ID, quantity, supplier details	
Requirements Description: The system shall generate and send order requests to suppliers. If a supplier is unavailable, alternative suppliers should be suggested.	
Outputs: Order confirmation message	

Requirement number: 8	**Use Case Number:** 8
Introduction: The system should generate analytical reports for inventory and sales.	
Inputs: Report type, date range	
Requirements Description: The system shall compile sales and inventory data into structured reports.	
Outputs: Graphs, tables, or downloadable reports	

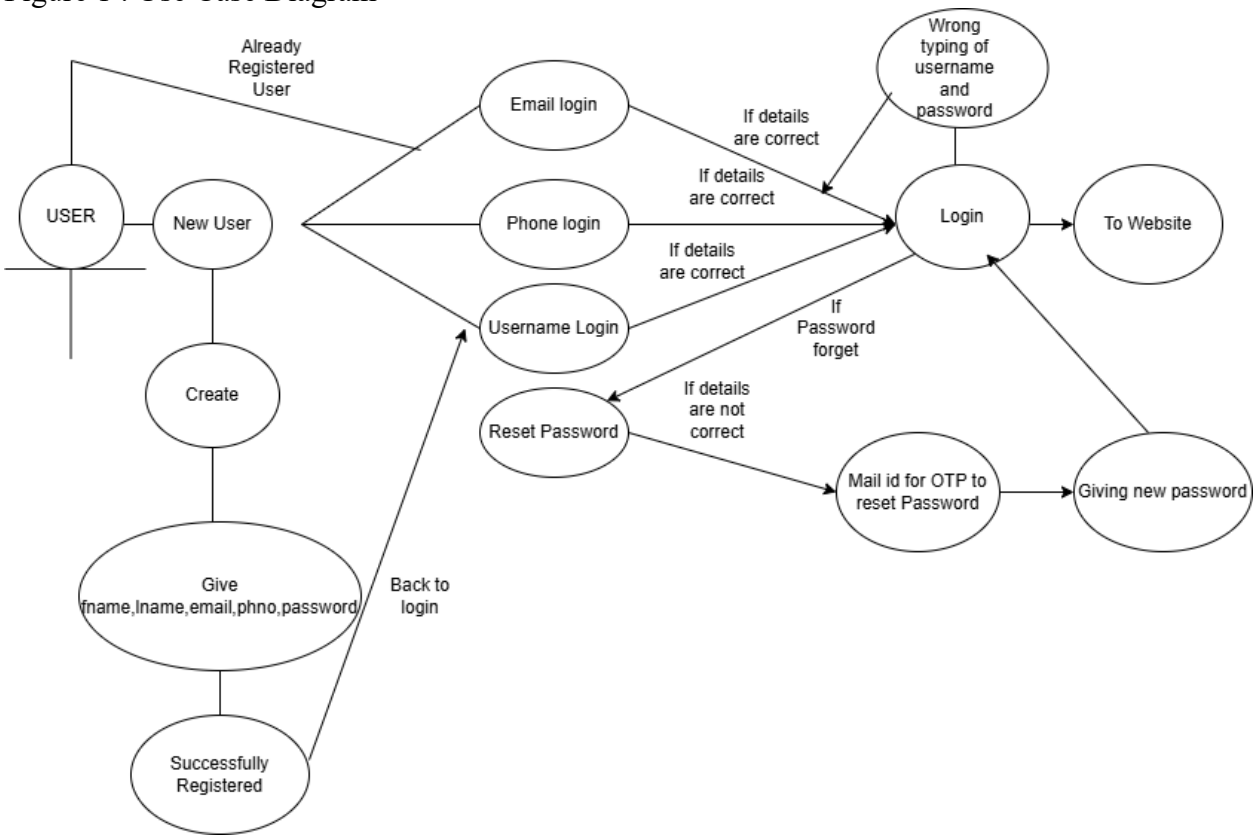
Requirement number: 9	**Use Case Number:** 9
Introduction: User access should be role-based, limiting actions based on predefined permissions.	
Inputs: User role, authentication credentials	
Requirements Description: The system shall grant or restrict access to functions based on user roles. Unauthorized actions shall trigger a warning message.	
Outputs: Access granted or restricted message	

Requirement number: 10	**Use Case Number:** 10
Introduction: The chatbot should assist users with inventory-related queries.	
Inputs: User query	
Requirements Description: The chatbot shall process input and provide relevant inventory data or suggest escalation to human support.	
Outputs: Chatbot response or escalation suggestion	

Requirement number: 11	**Use Case Number:** 11
Introduction: The system should evaluate supplier performance based on historical transactions.	

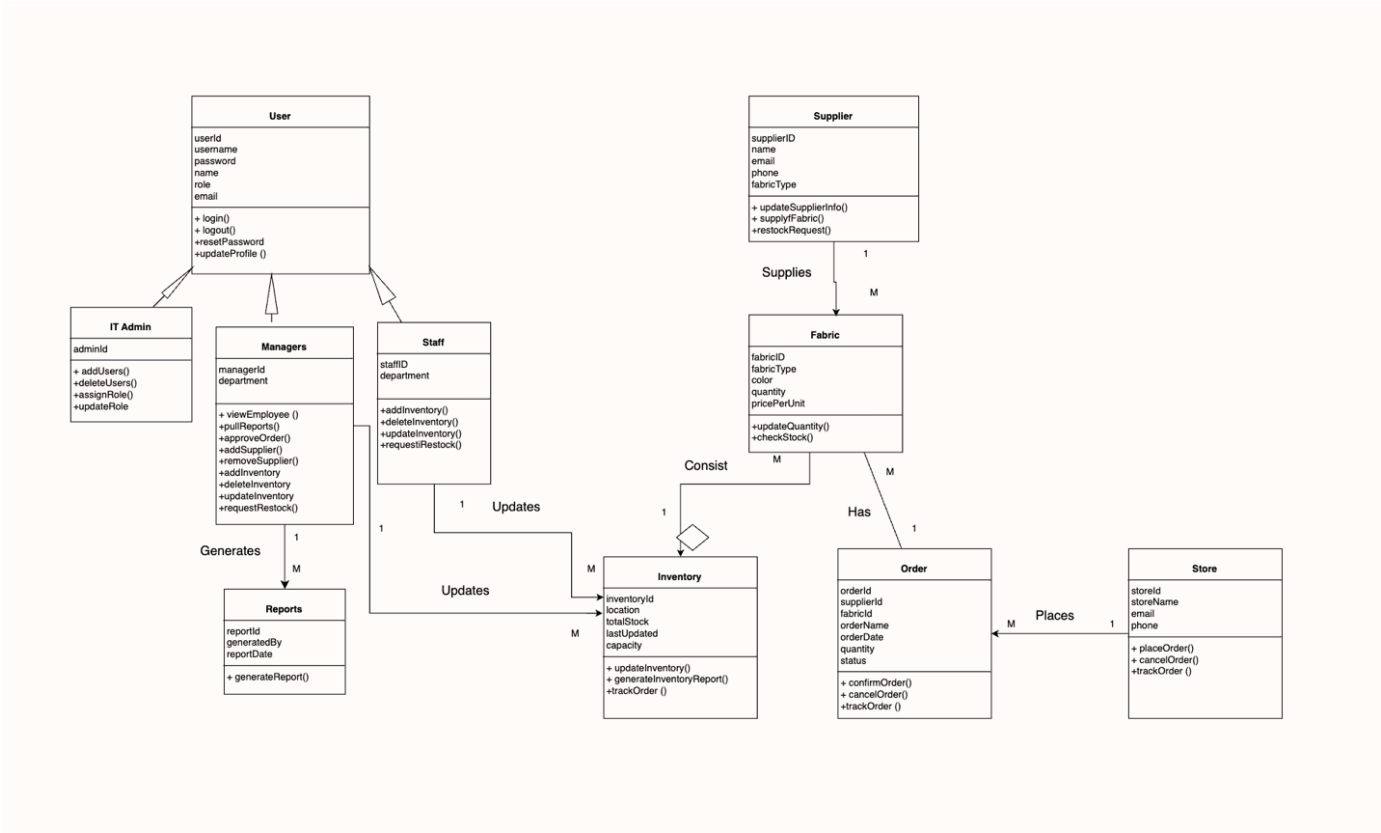
Inputs: Supplier transaction history, delivery speed, order accuracy
Requirements Description: The system shall generate reports on supplier reliability and performance.
Outputs: Supplier rating and performance analytics

Figure 1 : Use Case Diagram



SECTION 7: SYSTEM MODELING

Figure 2: Class Diagram for Inventory Management System



System Architecture:

The context model below, Figure 3, visualizes at a high level the system architecture of the Fabric Inventory Tool. This shows the interaction of the tool with other components. Each component has a unique job/purpose, however, there is overlap as together all of them make the efficient and complete Fabric Inventory Tool.

Below are the various components/systems:

1. Fabric Inventory Tool:
    a. This is the center piece in the context model as it is the primary system. It tracks incoming and outgoing inventory, manages stock levels, and generates restock alerts. The database and sales tracking system connect/interact with this.

2. User Interface:
    a. This component helps staff members of the fabric supply interact with the system and perform tasks such as tracking and updating inventory. The basic inventory update features include adding, editing, and deleting items. This component

interacts with all other system components.

3. Database:
    a. This component stores the data in an organized manner. Such as data regarding the type of fabric, the quantity, various suppliers' details, the user data, etc. It interacts with all other components.

4. Restock Ordering System:
    a. When inventory levels reach a set threshold, this system alerts the user about which fabric needs restocking and identifies the supplier that provided it.

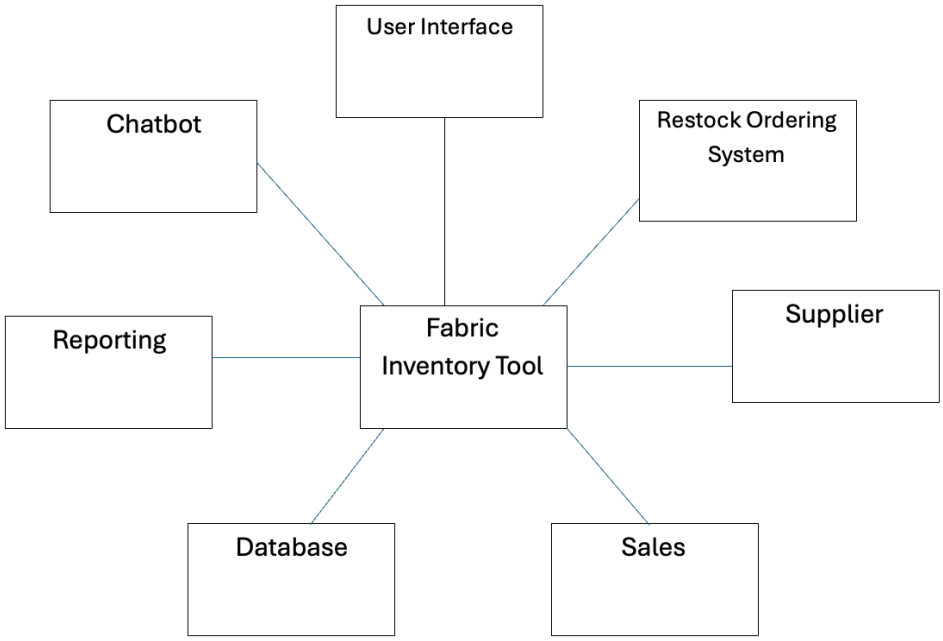
5. Supplier
    a. This is the third-party supplier that produces the various fabrics and sells it to the user. This is whom we look towards to purchase supplies, thus making it important that the inventory interacts with it.

6. Sales System
    a. This system keeps track of the financial aspect of the user business. How many stocks are being purchased and at what price. As well as the rate at which the stock the user ordered and the need to restock. The profit or loss the user is making.

7. Reporting System
    a. This system interacts with the inventory making reports that display the stocks, supplying, ordering and purchase transactions.

8. Chatbot
    a. Helps the user gain information about the inventory quickly and accurately through the help of AI.

Figure 3: Context Model



Database Tables

#Note- PK stands for Primary key & FK stands for Foreign Key.

1. Users Table

- user_id (INT, PK, AUTO_INCREMENT)
- username (VARCHAR(50), UNIQUE, NOT NULL)
- password_hash (VARCHAR(255), NOT NULL)
- role (ENUM('Admin', 'Inventory Manager', 'Staff Member', 'Business Owner'), NOT NULL)

2. **Fabric Inventory Table**

- fabric_id (INT, PK, AUTO_INCREMENT)
- fabric_type (VARCHAR(100), NOT NULL)
- color (VARCHAR(50), NOT NULL)
- quantity (INT, NOT NULL)
- price_per_unit (DECIMAL(10,2), NOT NULL)
- supplier_id (INT, FK → Suppliers.supplier_id)
- restock_threshold (INT, NOT NULL)

3. Suppliers Table

- supplier_id (INT, PK, AUTO_INCREMENT)

- supplier_name (VARCHAR(100), NOT NULL)
- contact_info (VARCHAR(255), NOT NULL)
- address (TEXT, NOT NULL)
- rating (DECIMAL(3,2), DEFAULT NULL)

4. **Stock Transactions Table**
   - transaction_id (INT, PK, AUTO_INCREMENT)
   - fabric_id (INT, FK → Fabric Inventory.fabric_id)
   - transaction_type (ENUM('Addition', 'Removal', 'Update'), NOT NULL)
   - quantity_changed (INT, NOT NULL)
   - user_id (INT, FK → Users.user_id)
   - transaction_date (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)

5. **Restock Alerts Table**
   - alert_id (INT, PK, AUTO_INCREMENT)
   - fabric_id (INT, FK → Fabric Inventory.fabric_id)
   - current_stock (INT, NOT NULL)
   - alert_date (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
   - status (ENUM('Pending', 'Resolved'), DEFAULT 'Pending')

6. **Orders Table**
   - order_id (INT, PK, AUTO_INCREMENT)
   - supplier_id (INT, FK → Suppliers.supplier_id)
   - fabric_id (INT, FK → Fabric Inventory.fabric_id)
   - quantity (INT, NOT NULL)
   - order_date (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
   - status (ENUM('Pending', 'Completed', 'Cancelled'), DEFAULT 'Pending')

7. **Sales Reports Table**
   - report_id (INT, PK, AUTO_INCREMENT)
   - generated_by (INT, FK → Users.user_id)
   - report_date (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP)
   - report_data (TEXT, NOT NULL) -- (Can store JSON or structured data)

8. **Role-Based Access Table**
   - role_id (INT, PK, AUTO_INCREMENT)
   - role_name (VARCHAR(50), UNIQUE, NOT NULL)
   - permissions (TEXT, NOT NULL) -- (Stores role-based access rules)

Relationships:

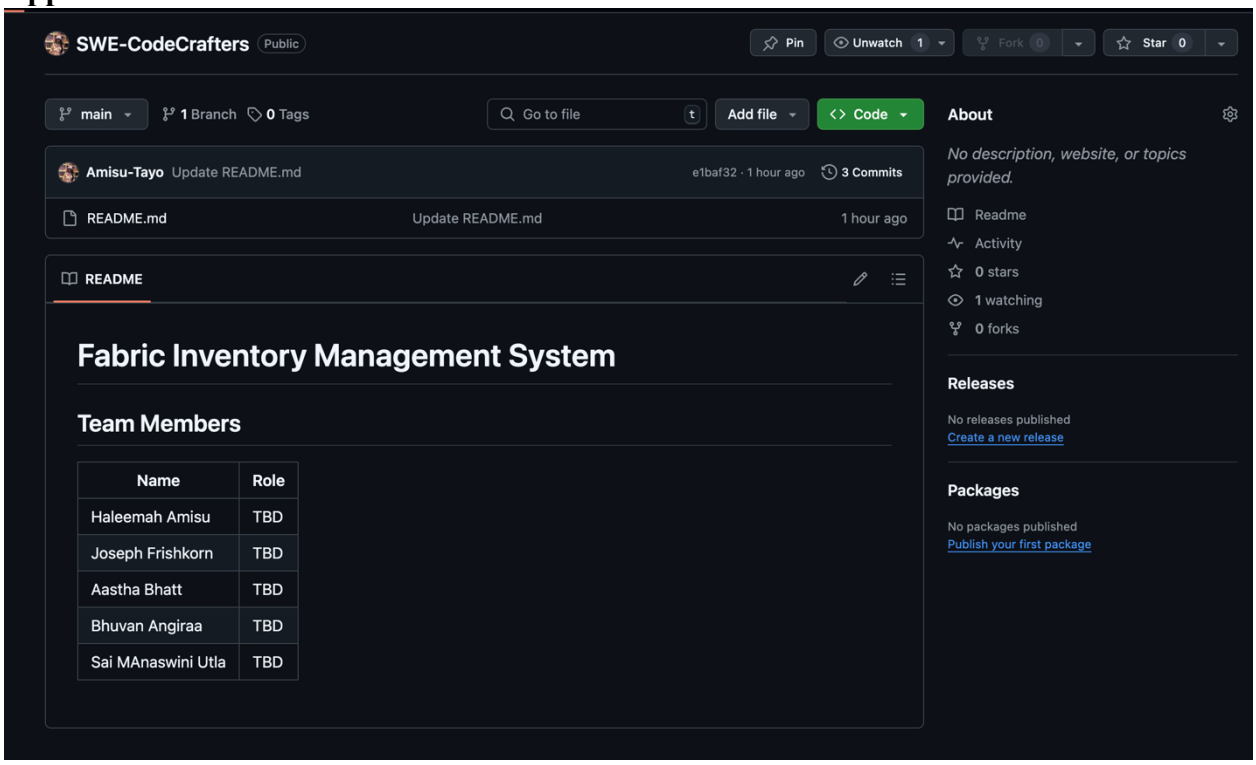
1. **Users** manage **Fabric Inventory** and process transactions.
2. **Fabric Inventory** belongs to **Suppliers**.
3. **Stock Transactions** track fabric changes by **Users**.
4. **Restock Alerts** are generated based on **Fabric Inventory** stock levels.

5. **Orders** are placed with **Suppliers** for **Fabric Inventory** restocking.
6. **Sales Reports** are generated by **Users**.
7. **Role-Based Access** defines access for **Users**.

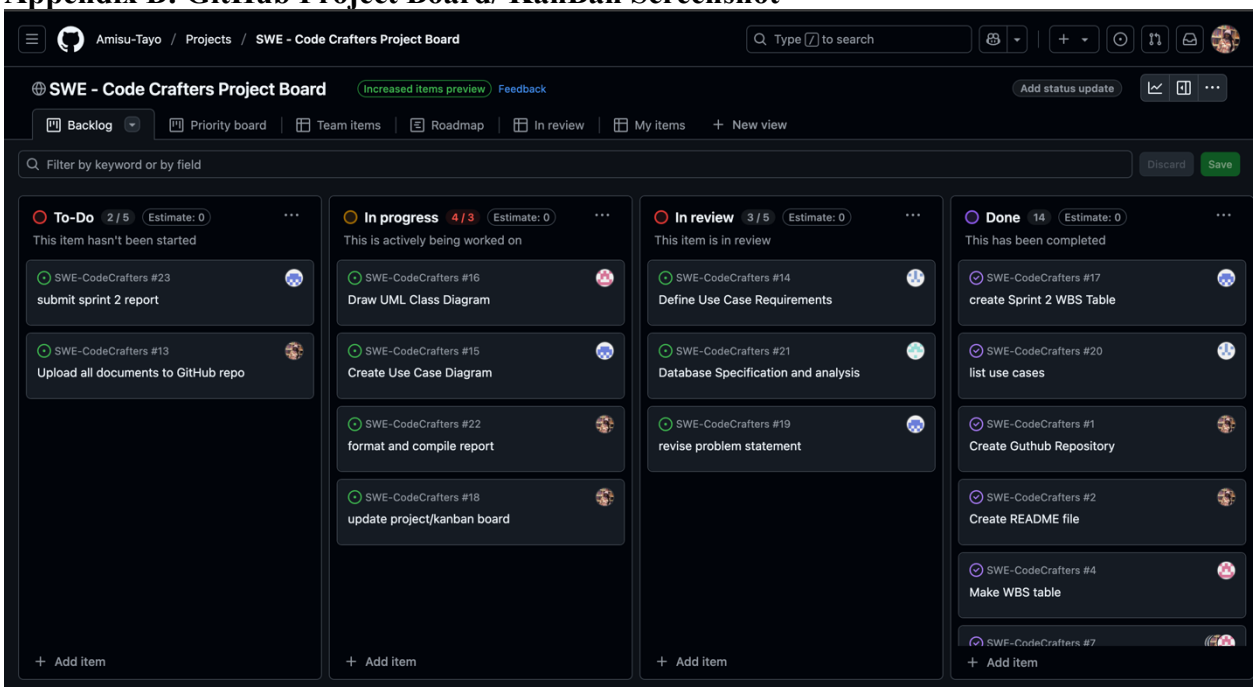
Database Management System,
We will be using MySQL as our DBMS and will be using MySQL Workbench to host the system.

SECTION 7: APPENDIX

Appendix A: GitHub README Screenshot



Appendix B: GitHub Project Board/ KanBan Screenshot



Appendix C: Direct Links
GitHub Repository: https://github.com/Amisu-Tayo/SWE-CodeCrafters
Project Board/KanBan Link: https://github.com/Amisu-Tayo/SWE-CodeCrafters/projects?query=is%3Aopen