Cloud connected data submission platform using Firebase security

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

G. Koushik (Reg.No – 41110670)
K.L.V.Uttishtta (Reg.No - 41110705)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY - 1 UNIVERSITY BY UGC
Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI - 600119

APRIL - 2024

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Category – 1 University by UGC
Accredited with "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **G.Koushik Goud(41110670),K.L.V.Uttishtta(41110705)** who carried out the Project work entitled **"Cloud connected data submission platform using Firebase security"** under my supervision from November 2024 to April 2025.

Internal Guide

Dr. S. PRINCE MARY, M.E., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600 119.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on 03-4-25

Internal Examiner

External Examiner

DECLARATION

I, **G.Koushik Goud (Reg. No-41110670)** hereby declare that the Project Report entitled **"Cloud connected data submission platform using Firebase security"** done by me under the guidance of **Dr.S.Prince Mary, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 03-04-25

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **the Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D.**, **Dean**, School of Computing, **Dr. L. Lakshmanan, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Prince Mary, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement which paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in manyways for the completion of the project.

ABSTRACT

The goal of this project is to incorporate an extensive cargo management site which together with data secured with role-based access control for managing and verifying vessel data for a cargo company. Conventional cargo management solutions applied at today's companies can be described as decentralized and paper-based, which results in numerous issues including lost time and enhanced risks of errors. To counter these practices, this work proposes a cloud coupled, multi-type, web site available in Web, Android, and iOS environments.

With such tools incorporated in the platform, the program employs a Firebase that is effective in handling data securely and making essentials updates in real time. This work has the following goals: (1) access control, where the roles assigned are chief officer, team leader, and employee, meaning each has specific functions and levels of data access; (2) enriched data input by integrating image and video uploading of specific vessel types according to age; and (3) ocean and cargo design to create user engagement. Some of the main benefits of this proposed solution are– decreased chances of errors, stronger data security and free data availability, easier verification and an effective system of data access based on role played with reference to the typical industry security standards. This way does not only enhance the efficiency of the service but also creates a foundation for prospects of adding it to other services.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1. GENERAL INTRODUCTION:

The cargo industry is a vital component of global trade and supply chains, serving as the backbone of international commerce. Efficient cargo management ensures the smooth movement of goods across borders, supporting industries such as manufacturing, retail, and e-commerce. However, the complexity of managing large volumes of data in cargo operations presents several challenges.

Traditional cargo management systems often rely on outdated manual processes or fragmented digital solutions that lack centralized control. Such systems are prone to issues such as data redundancy, poor security, and limited scalability. Additionally, coordinating multiple stakeholders — including cargo handlers, shipping companies, customs authorities, and warehouse managers — requires robust data synchronization to avoid miscommunication and delays.

2. TECHNOLOGIES USED:

2.1.Firebase Authentication

Purpose & Role:

- Ensures **secure login** and manages user authentication across different roles (chief officers, team leaders, employees).
- Supports **multiple authentication methods**, including email/password, Google sign-in, and third-party providers.
- Implements **two-factor authentication (2FA)** for additional security against unauthorized access.

Role-Based Access Control (RBAC):

- **Chief officers:** Have complete control over the system, including access to all data, approvals, and reports.
- **Team leaders:** Can review, approve, and modify cargo-related data submitted by employees.
- **Employees:** Limited access to **data entry and submission** functions, ensuring security and minimizing unnecessary access to sensitive information.

Session Management & Security Features:

- Tracks user sessions and automatically logs out inactive users to **reduce**

security risks.

- **IP and device tracking** to detect and prevent suspicious login attempts.
- Provides **encrypted authentication tokens** to maintain data integrity.

Fig 1.1: Firebase Authentication

2.2.Firebase Cloud Storage & Realtime Database

Firebase Cloud Storage:

- Used for **storing large amounts of structured and unstructured data**, such as shipment records, invoices, and multimedia files.
- Supports **secure, role-based storage**, ensuring that users can access only relevant files.
- Ensures **automatic data backup** to prevent loss due to system failures or accidental deletions.

Firebase Realtime Database:

- **Enables real-time data synchronization** across web, Android, and iOS platforms.
- Updates made in one location instantly reflect across all connected devices, preventing data discrepancies.
- Supports **structured data storage**, making retrieval and modification of cargo records more efficient.
- Uses **end-to-end encryption** to protect sensitive cargo data.

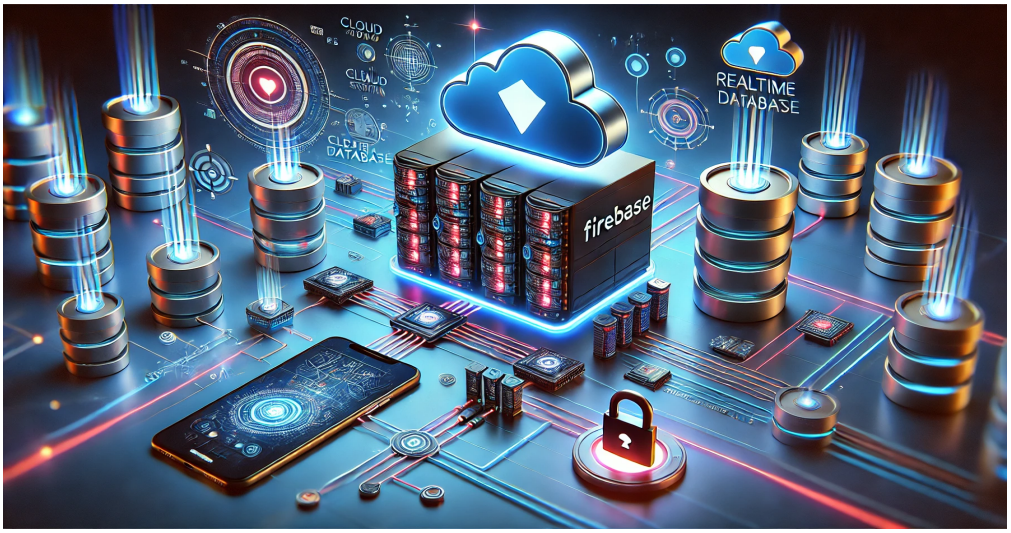


Fig 1.2: Firebase Cloud Storage & Realtime Database

2.3.*ReactJS*

- **Front-End Technologies for User Interaction:**
  - Both frameworks enable **dynamic and interactive interfaces**, ensuring a smooth user experience.
  - ReactJS is used for **web-based interactions**, while Vue.js enhances the **mobile experience** on iOS and Android devices.
  - Provides a **component-based architecture**, making the system easily extendable and scalable.
- **Real-Time UI Updates:**
  - Users can view changes **instantly**, such as shipment status updates or approval notifications.
  - **No need for manual refreshing**, improving efficiency and user engagement.
- **Cross-Platform Support:**
  - Ensures **seamless functionality across web, Android, and iOS platforms** without needing separate codebases.
  - Improves **data consistency and user accessibility**, allowing remote access to cargo management functions.

.

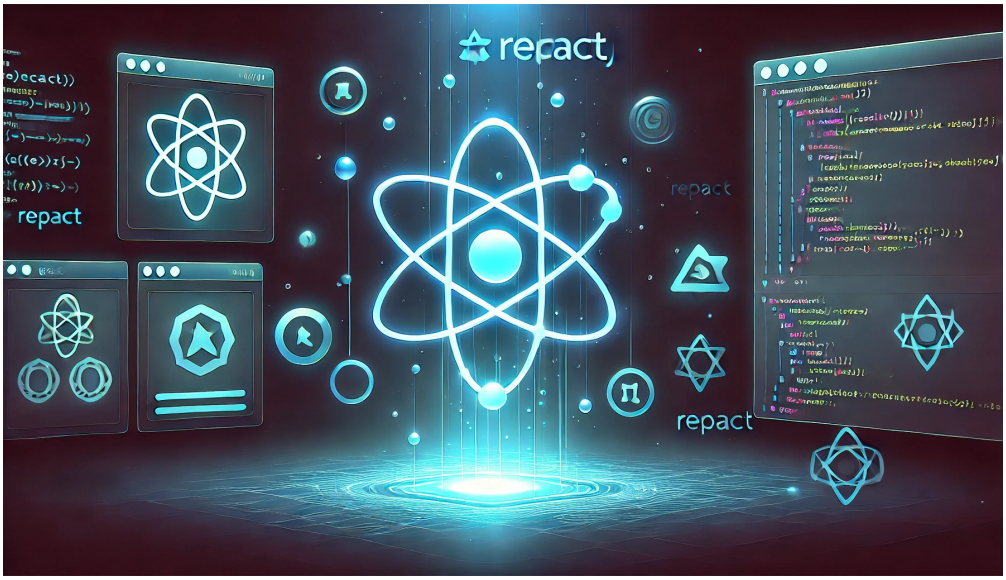


Fig 1.3: ReactJS

3. OBJECTIVES OF THE PROJECT:

Implement a Secure, Role-Based Authentication System:

- Develop a structured access control system using Firebase Authentication.
- Ensure **each user role has defined permissions**, reducing unauthorized access.
- Strengthen security with **multi-factor authentication** and session monitoring.

Enhance Data Flow and Synchronization:

- Provide **real-time updates** for all cargo data, minimizing delays and improving decision-making.
- Enable seamless data access across **multiple platforms (web, iOS, Android)**.

Improve Operational Efficiency & Reduce Errors:

- Reduce reliance on **manual, paper-based processes** by introducing a digital system.
- Automate **scheduling approvals, data verification, and cargo tracking** to enhance workflow.
- Prevent **data duplication and inconsistencies** through centralized storage.

Enhance Communication & Collaboration:

- Enable **instant communication between employees, team leaders, and**

chief officers through role-based dashboards.

- Provide **real-time notifications** for approvals, cargo updates, and workflow changes.

Ensure Scalability & Future Expansion:

- Design a system that can **accommodate industry growth and increasing data volumes**.
- Allow for future integration with **AI-driven analytics, performance tracking, and predictive maintenance**.

4. SCOPE OF THE PROJECT:

User Role Management & Access Control:

- The system categorizes users into **three primary roles—chief officers, team leaders, and employees**—each with specific access privileges.
- Chief officers oversee all activities, **generate reports, and approve major decisions**.
- Team leaders review, verify, and approve **cargo-related submissions and schedules**.
- Employees enter data, **submit forms, and upload cargo-related media files**.

Centralized Data Management & Storage:

- All shipment details, invoices, schedules, and cargo records are **stored securely using Firebase Cloud Storage**.
- Includes **data backup and recovery mechanisms** to prevent loss of critical information.
- Uses **structured data segmentation**, ensuring users can only access relevant data.

.

CHAPTER 2

LITERATURE SURVEY

In [1], the authors also explain the reason why project deadlines are hazardous to the quality of software applications, which necessitates the application of agility. To conclude, the study provides a Firebase as a solution for increasing the development speed. Firebase is a mobile and web application platform where it is utilized to do real-time databases, authentication, and server-less backend. In the real case of an Android Studio application, the combined use of Firebase Authentication to manage user access and profile securely as well as Cloud Fire store to provide real-time synchronization of data across client devices is illustrated in the research. It takes care of user-generated content and doesn't require dedicated server architecture for this purpose securely. Firebase allows developers to dedicate time only to the core functionality development as well as enhancing the UI/UX design. The paper also stresses the effectiveness of Firebase in terms of its scalability, velocity of integration, and the fact that projects can switch to it from other methods of handling back-end services, helping with making project development faster, and for dynamic environments.

In [2], based on the analysis of the Cloud environments, the authors consider the user-resource interactions dynamic and ad hoc. They present a SAT-RBAC model (Security and Availability-based Trust in RBAC) as an extension of the basic RBAC model to Cloud systems. First, components of trust relationships, consisting of security state of the user's host, availability of the network, and protection levels provided by service providers, are recognized and integrated. Second, a security-based scheduling model is proposed to effectively cope with Cloud environments that possess high levels of uncertainty. The trust relationship in SAT-RBAC is divided into three zones: unbelievable, probable believable, &believable. In order to determine trust probabilities in the probable believable zone, a Bayesian technique is applied. Furthermore, there are two prescriptive models derived by the algorithm, which can assess the key trust factors. The feasibility and accuracy of the SAT-RBAC model are verified with experiments on simulated environments based on Cloud Sim in Planet Lab to detect and remove the abnormal behaviors in Cloud-based applications.

In [3], the authors also analyze such trends as the IoT, smart cities, the digital

transformation of enterprises, and the global digital economy that have resulted in the exponential increase in data. This growth has also enhanced the demand for storage commonly known as cloud storage which has further enhanced its growth. A cloud storage service has become crucial to a storage service to manage and store their data as the governments, enterprises and individuals embraced this technology. Nevertheless, the given migration opens certain threats and concerns as hacking, leakage of private information, or violation of privacy rights. Although there are research papers on data security and privacy concerns, this paper identifies that there is a gap of a systematic literature review related to cloud storage systems. This paper therefore seeks to meet that gap by providing a systematic literature review of data security concerns, privacy preservation mechanisms and encryption techniques in cloud storage systems. The first part of the study aims at outlining a general understanding of cloud storage where it will define this phenomenon, classify it and outline its architecture and most importantly its applications. Next, it discusses challenges and necessities in data protection and privacy in cloud systems. It also provides the summary of other encryption technology and countering. Last of all, the paper points to research directions to fill the existing research gaps and inform future research directions on cloud storage security.

1. INFERENCES FROM LITERATURE SURVEY

1.1. The Shift from Traditional to Cloud-Based Systems

One of the major findings from the literature survey is the shift from **traditional manual or on-premise cargo management systems** to **cloud-based platforms**. Conventional cargo management relied heavily on **paper-based documentation, manual approvals, and localized databases**, which resulted in inefficiencies, **high operational costs, and limited scalability**. With the advent of cloud technologies such as **Firebase, AWS, and Microsoft Azure**, companies now leverage real-time data synchronization, remote access, and **improved security.**

Pros:

- **Scalability** – Cloud-based systems can accommodate increasing data loads without

requiring infrastructure changes.

- **Remote Accessibility** – Enables employees to access cargo data from anywhere, reducing delays.
- **Real-Time Updates** – Ensures that all stakeholders work with the latest data, improving decision-making.

Cons:

- **Security Risks** – Cloud platforms can be vulnerable to cyberattacks if not properly secured.
- **Dependency on Internet Connectivity** – If the internet fails, access to critical cargo data may be interrupted.
- **Cost of Migration** – Shifting from legacy systems to the cloud requires significant investment in technology and training.

1.2. Importance of Role-Based Access Control (RBAC)

Medical Traditional systems often provide **open access to all users**, leading to **data breaches and inefficiencies**. The literature survey suggests that **Role-Based Access Control (RBAC)** enhances security by restricting access to authorized personnel. **Firebase Authentication** is an example of how **access control mechanisms ensure that only authorized users can view, edit, or delete sensitive cargo data**.

Pros:

- **Improved Data Security** – Ensures that only relevant personnel access sensitive information.
- **Prevents Data Manipulation** – Eliminates unauthorized data alterations.
- **Optimized Workflow** – Different user roles streamline work processes by avoiding redundant actions.

Cons:

- **Complex Role Management** – Assigning multiple roles and permissions can be challenging.
- **User Frustration** – If roles are too restrictive, necessary users may be blocked from required data.

2.1.3 The Role of Real-Time Synchronization in Cargo Management

The literature highlights that **real-time synchronization** using Firebase or similar technologies **reduces operational delays and enhances efficiency**. Traditional cargo systems often suffer from **data lags, outdated records, and inconsistent updates**, leading to **miscommunication and workflow bottlenecks**.

Pros:

- **Minimizes Errors** – Data is updated instantly, reducing the risk of outdated or incorrect information.
- **Improves Communication** – Teams can coordinate effectively using synchronized schedules.
- **Enhances Operational Speed** – Enables instant updates for cargo loading, tracking, and approvals.

Cons:

- **High Bandwidth Usage** – Real-time data synchronization consumes significant network bandwidth.
- **Complex Implementation** – Requires robust backend architecture to prevent data conflicts.
- **Data Overwrites Risk** – If not managed correctly, real-time updates can lead to accidental overwriting of information.

2.1.4 Lightweight RFID Protocol for Medical Privacy Protection in IoT

With the increasing use of **cloud storage** (such as Firebase Cloud Storage), securing cargo data has become a **major challenge**. The literature survey indicates **threats such as hacking, unauthorized access, and data leakage**. Implementing **strong encryption, multi-factor authentication, and compliance with GDPR or other data protection regulations** is essential.

Pros:

- **Stronger Data Encryption** – Prevents unauthorized access to confidential information.

- **Compliance with Industry Standards** – Meets regulatory requirements for data security.
- **Improved Data Backup Mechanisms** – Reduces risk of data loss from accidental deletions or cyberattacks.

Cons:

- **Increased Security Costs** – High-level security measures require significant investment.
- **Access Control Complexity** – Setting up secure permissions can be challenging.
- **User Authentication Delays** – Enhanced security measures may slow down login processes.

2.1.5 AI and Predictive Analytics in Cargo Management

Investigating The literature review highlights that AI-powered predictive analytics can optimize cargo scheduling, demand forecasting, and route planning. AI models analyse historical shipping patterns and suggest improvements.

Pros:

- **Optimized Cargo Scheduling** – AI can reduce downtime by predicting peak shipping times.
- **Lower Operational Costs** – Automates manual decision-making processes, reducing labor costs.
- **Enhanced Decision-Making** – Provides data-driven insights for better logistics planning.
-

Cons:

- **Requires High-Quality Data** – AI performance depends on accurate and complete data.
- **Expensive to Implement** – AI-driven systems require significant investment in training and development.
- **Risk of Algorithm Bias** – Poorly trained AI models may generate misleading predictions

2.1.6 User Interface (UI) and User Experience (UX) in Cargo Systems

Many traditional systems have **complicated, outdated interfaces**, making them **difficult for employees to use**. Studies emphasize the importance of **intuitive UI/UX** for improved productivity.

Pros:

- **Increases Productivity** – A simple interface speeds up tasks and reduces errors.
- **Eases Training** – New employees can adapt to the system quickly.
- **Enhances User Satisfaction** – A well-designed UI improves engagement and efficiency.

Cons:

- **Requires Regular Updates** – UI/UX needs frequent improvements to stay relevant.
- **Subjective Design Preferences** – Not all users may agree on the best interface design.
- **High Development Costs** – Creating a user-friendly interface requires skilled designers and developers.

2.1.7 Limited Adoption of Blockchain for Cargo Data Security

As Blockchain technology has the potential to enhance cargo data security through decentralized ledgers, its adoption remains limited.

Pros:

- **Immutable Data Storage** – Transactions are permanent and tamper-proof.

- **Reduces Fraud** – Enhances cargo tracking and verification.
- **Decentralized Management** – No single point of failure.

Cons:

- **High Computational Costs** – Requires powerful computing infrastructure.
- **Limited Scalability** – Blockchain-based systems can be slower than

traditional databases.

- **Complex Integration** – Existing cargo management systems may struggle to adopt blockchain.

2.1.8 Role of Automation in Cargo Management

Dengue The use of automated workflows, including robotic process automation (RPA), reduces human errors and accelerates processing times.

Pros:

- **Reduces Manual Work** – Automates repetitive tasks, improving efficiency.
- **Minimizes Human Errors** – Ensures accuracy in data processing.
- **Scalability** – Can handle increased cargo volume with minimal additional labour.

Cons:

- **Initial Setup Costs** – Requires investment in automation technologies.
- **Potential Job Reductions** – Automation may replace some human roles.
- **Complex Debugging** – Errors in automation scripts can be difficult to troubleshoot.

2.1.9 Mobile-Friendly Cargo Management Systems

Chaining, with increased smartphone usage, mobile-friendly cargo management applications enhance real-time access and operational flexibility.

Pros:

- **Allows Remote Access** – Employees can check updates on the go.
- **Improves Communication** – Teams stay connected in real time.
- **Boosts Productivity** – Reduces reliance on desktop systems.

Cons:

- **Requires Responsive Design** – Needs optimization for different devices.
- **Security Risks** – Mobile apps can be vulnerable to cyber threats.
- **Limited Functionality** – Some desktop features may not be available on

mobile.

2.1.10 Future Scope of IoT in Cargo Tracking

Industrial IoT sensors improve cargo tracking and fleet monitoring, ensuring real-time data transmission.

Pros:

- Enhances Tracking Accuracy
- Reduces Theft and Loss
- Optimizes Logistics Planning

Cons:

- High Initial Investment
- Requires Stable Connectivity
- Potential Sensor Failures

2.1.11 The Shift from Traditional to Cloud-Based Systems

In The cargo management industry traditionally relied on manual record-keeping systems or on-premises software solutions. These systems involved paper-based processes that resulted in inefficiencies such as data duplication, delays in data entry, and limited accessibility. Traditional systems were also prone to human errors due to manual data handling.

Modern literature highlights the shift towards **cloud-based solutions** that provide remote data access, scalability, and improved data security. Platforms like **Firebase Cloud Storage**, **Amazon Web Services (AWS)**, and **Microsoft Azure** now dominate the industry, offering improved flexibility and robust data storage. By adopting cloud-based solutions, cargo firms ensure that information is accessible from multiple devices in real-time. This eliminates dependency on local servers, reducing maintenance costs and improving collaboration across teams.

For instance, a cargo firm managing vessel shipments across multiple regions can now access real-time data updates using **Firebase's real-time database**,

ensuring synchronized information across devices and platforms.

Pros:

- **Enhanced Scalability:** Cloud systems dynamically expand with increased data loads, reducing system crashes during peak cargo operations.
- **Global Accessibility:** Enables employees across multiple locations to access cargo data, improving coordination and reducing downtime.
- **Improved Backup & Recovery:** Cloud platforms ensure regular data backups, protecting businesses from accidental data loss.

Cons:

- **Potential Security Risks:** Without proper encryption, sensitive cargo data may be vulnerable to cyberattacks.
- **Internet Dependency:** Cloud systems require stable internet connectivity; disruptions may halt critical operations.
- **Data Migration Challenges:** Transitioning from legacy systems to cloud platforms may result in data inconsistency without proper migration protocols.

2.1.12 Importance of Role-Based Access Control (RBAC)

Traditional systems often fail to distinguish between different user roles, exposing sensitive data to unauthorized personnel. The adoption of Role-Based Access Control (RBAC) has emerged as a powerful solution in modern cargo systems.
RBAC ensures that **chief officers, team leaders, and employees** have distinct access permissions based on their roles. For example:

- **Chief Officers** have unrestricted access to manage data, schedules, and reports.
- **Team Leaders** can review and approve cargo submissions but cannot alter company-wide data.
- **Employees** are limited to data entry and specific task management.

By restricting access, RBAC reduces data leaks, ensures privacy, and protects critical records. In systems like **Firebase Authentication**, assigning user roles ensures only authorized individuals access specific data sets.

Pros:

- **Enhanced Security:** Minimizes unauthorized data exposure.
- **Workflow Optimization:** Ensures that employees only focus on tasks relevant to their role.
- **Improved Accountability:** Role-based logs help track data modifications, reducing fraud or errors.

Cons:

- **Complex Role Management:** Maintaining accurate role permissions for a large workforce may become challenging.
- **Risk of Role Confusion:** Poorly defined roles may result in employees being denied access to essential tools.
- **Higher Initial Configuration Time:** Establishing an RBAC system requires planning, testing, and role assignment.

2.1.13 The Role of Real-Time Synchronization in Cargo Management

Real-time synchronization has revolutionized cargo data management by enabling instant updates across platforms. Traditional cargo systems often faced issues such as data delays, outdated records, and conflicting entries.

With **Firebase Realtime Database** or similar technologies, changes made on one device immediately reflect across all connected platforms. For instance, if a shipment's status changes from "Pending" to "Delivered," all team members are updated in real time without manual communication.

This synchronization capability is vital for cargo businesses managing **multi-location warehouses**, ensuring that teams receive immediate updates on shipment statuses, loading schedules, or data corrections.

Pros:

- **Minimized Errors:** Reduces delays caused by outdated information.
- **Enhanced Collaboration:** Ensures teams are aligned on cargo schedules and shipment updates.
- **Improved Customer Service:** Real-time updates allow faster responses to customer inquiries.

Cons:

- **Bandwidth Consumption:** Real-time synchronization demands high network bandwidth.
- **Data Conflict Risks:** Simultaneous updates may occasionally overwrite data without proper synchronization controls.
- **Increased Complexity:** Real-time systems require complex backend development.

2.1.14 Data Security and Privacy in Cloud-Based Systems

Cargo systems often store sensitive financial data, shipment details, and customer records. Literature emphasizes that security breaches in cargo management can result in significant financial losses.

Solutions like **Firebase Cloud Storage** apply **AES encryption**, **multi-factor authentication (MFA)**, and **role-specific access control** to protect data. Additionally, **data masking** and **secure API gateways** ensure data remains confidential during transmission.

For example, in a cargo company handling international shipments, encryption prevents unauthorized parties from accessing customs documents, invoices, or client data.

Pros:

- **Improved Data Integrity:** Encryption techniques ensure data cannot be altered without proper permissions.
- **Regulatory Compliance:** Systems adhering to GDPR, HIPAA, or CCPA minimize legal risks.
- **Enhanced Trust:** Secure systems improve client trust by protecting financial and personal data.

Cons:

- **Higher Maintenance Costs:** Security features like MFA and encryption increase development expenses.
- **Performance Overhead:** Advanced encryption may slow data transmission.
- **User Complexity:** Additional security layers may inconvenience authorized users.

2.1.15 Automation in Data Entry and Processing

The Automating cargo data entry using tools like Robotic Process Automation (RPA) reduces human errors and accelerates data processing.
For example, RPA tools can extract data from invoices, customs forms, or cargo manifests, reducing manual efforts.

Pros:

- **Increased Efficiency:** Automation reduces time spent on repetitive tasks.
- **Error Reduction:** Minimizes common data entry mistakes.
- **Cost Savings:** Reduces labor costs through automated workflows.

Cons:

- **Initial Setup Costs:** RPA tools require extensive development efforts.
- **Complex Maintenance:** Automation systems require regular updates.
- **Data Formatting Challenges:** Inconsistent document layouts may hinder automation accuracy.

2. OPEN PROBLEMS IN EXISTING SYSTEM

Despite advancements highlighted in the literature, several issues persist:

1. **Scalability Issues**
    - Existing solutions face difficulties handling large-scale data in real-time applications.
    - Systems may degrade in performance as data volume increases.
2. **Resource Constraints**
    - Many advanced models demand significant computational resources, limiting accessibility in resource-constrained environments.
3. **Data Quality and Availability**

- Incomplete, noisy, or biased datasets reduce the effectiveness of current systems.
- Data augmentation techniques often fail to generalize effectively.

4. **Lack of Standardized Evaluation Metrics**
    - Existing studies use inconsistent performance indicators, complicating objective assessment.
5. **Security and Privacy Concerns**
    - Solutions handling sensitive data often lack robust security mechanisms.
    - Privacy-preserving techniques are underdeveloped in most proposed frameworks.
6. **Limited Real-World Deployment**
    - Many state-of-the-art models remain theoretical with limited testing in practical scenarios.
    - Deployment hurdles such as integration issues, latency, and system reliability remain unresolved.

CHAPTER 3

REQUIREMENT ANALYSIS

1. FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

In A **Feasibility Study, it** assesses the practicality of a proposed project or system, determining whether the project is viable within given constraints like budget, resources, and timeline. This evaluation ensures that the project's objectives are achievable and aligns with organizational goals.

The feasibility study covers multiple aspects:

1. **Technical Feasibility**
    - Evaluates the technical resources available to implement the project.
    - Assesses the required hardware, software, technology stack, and expertise needed.
    - Identifies potential risks in technology adoption.
2. **Economic Feasibility**
    - Conducts a cost-benefit analysis to determine if the investment in the project is justified.
    - Examines development costs, maintenance costs, and projected returns on investment.
3. **Operational Feasibility**
    - Determines if the system will function effectively within the organization’s workflow.
    - Assesses user adaptability, skill requirements, and training needs.
4. **Legal Feasibility**
    - Ensures the project complies with data protection laws, intellectual property regulations, and security standards.

2. SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

- Introduction

- Overall Description
- Functional Requirements
- Non-Functional Requirements
- Interface Requirements
- System Features
- Validation and Verification
- Assumptions and Dependencies

2.1. *Introduction*

The Introduction section outlines the purpose, scope, and intended audience for the software system. It establishes the foundation for understanding the project's objectives and the system's key features.

Purpose of the System:

- The purpose of this system is to develop a **Role-Based Cargo Management System** that streamlines cargo tracking, enhances security, and ensures effective data synchronization.
- The system aims to minimize data discrepancies, improve real-time coordination, and secure information sharing across multiple organizational levels.

Scope of the System:

The system will provide:

- Role-based data access to restrict unauthorized entry.
- Integration with **Firebase Authentication** for secure login mechanisms.
- Cloud storage support for secure and scalable data handling.
- Real-time synchronization for immediate updates across platforms.

Intended Audience:

- **Project Stakeholders:** Business owners, managers, and investors.
- **End Users:** Chief officers, team leaders, and field employees.
- **Developers and Testers:** Technical teams implementing and validating the system.

2.2. Overall Description

This section offers a comprehensive overview of the system, its functionality, and environmental factors.

Product Perspective:

- The system integrates **ReactJS** and **Vue.js** for interactive UI interfaces with Firebase for back-end operations.
- It is designed to operate seamlessly across web, iOS, and Android platforms.

Product Functions:

- Users are classified into **Chief Officers**, **Team Leaders**, and **Employees**.
- Each role has distinct access privileges to ensure data security.
- Data such as cargo schedules, shipment updates, and inventory details are maintained in **Firebase Cloud Storage**.

User Characteristics:

- Users range from **technical experts** (chief officers) to **non-technical staff** (cargo handlers).
- The system's intuitive UI design ensures easy adaptability for users with minimal technical knowledge.

.

2.3. Functional Requirements

This section details the system's expected behavior and functionalities.

Key Functionalities:

1. **User Authentication:**
    - Integration with **Firebase Authentication** ensures secure login via email, password, or social media credentials.
    - Implements **two-factor authentication (2FA)** for enhanced security.
2. **Role-Based Access Control (RBAC):**
    - **Chief Officers:** Full access to system data and control over administrative settings.
    - **Team Leaders:** Manage schedules, approve cargo shipments, and monitor tasks.
    - **Employees:** Limited to data entry and operational updates.
3. **Data Synchronization:**
    - Firebase's real-time database ensures synchronized updates across all platforms.
    - Data updates are reflected in real-time to reduce operational delays.
4. **Cargo Tracking System:**
    - Tracks shipment details such as location, status, and estimated arrival time.

- Provides alerts for delays or deviations in shipment routes.

2.4. Non-Functional Requirements

The Non-functional requirements define system performance, security, and usability constraints.

Key Non-Functional Requirements:

1. **Performance Requirements:**
   - The system must process at least **100 concurrent user requests** without performance degradation.
   - Response time for user interactions must be **less than 3 seconds** under peak loads.
2. **Security Requirements:**
   - Data encryption using **AES-256** for secure data storage.
   - Implements **Firebase Security Rules** to regulate data access.
3. **Scalability Requirements:**
   - The system must scale to accommodate **5x growth** in data volume over the next 3 years.
4. **Reliability:**
   - System uptime should exceed **99.9%** to ensure continuous operations.
5. **Usability:**
   - The system must feature an **intuitive interface** with clear navigation for all user roles.

2.5. Interface Requirements

Interface requirements define how the system will interact with users and other systems.

User Interface (UI) Design:

- The system's UI will feature a **dashboard-based interface** that displays role-specific information.
- The dashboard for:

- **Chief Officers** will show administrative controls, schedules, and performance analytics.
- **Team Leaders** will access team progress, pending shipments, and assign tasks.
- **Employees** will access only assigned shipment details for data entry.

Hardware Interface:

- Compatible with standard devices such as **laptops**, **tablets**, and **smartphones**.
- Requires internet connectivity for accessing Firebase Cloud features.

Software Interface:

- Integrated with Firebase services for data storage, authentication, and synchronization.
- APIs will enable connectivity with third-party systems like **shipment tracking platforms** or **inventory databases**.

.

2.6. System Features

The system introduces several core features that ensure smooth cargo management.

Feature 1: Dashboard Management

- Displays cargo status updates, employee activities, and alerts.
- Uses **ReactJS and Vue.js** to offer an interactive experience.

Feature 2: Notification System

- Real-time notifications for shipment updates, schedule changes, and security alerts.
- Ensures team leaders and employees receive relevant alerts via **push notifications** and **emails**.

Feature 3: Document Management System

- Enables secure uploading, storing, and sharing of shipment invoices, customs forms, and client contracts.
- Uses **Firebase Cloud Storage** for scalable document handling

2.7. Validation and Verification

For Validation and verification ensure the system meets technical and functional expectations.

Validation:

- Regular testing will ensure all features align with project objectives.
- User acceptance testing (UAT) will confirm the system's performance in real-world scenarios.

Verification:

- The system will undergo unit testing, integration testing, and security checks before deployment.
- Testing ensures features like **role-based access**, **synchronization**, and **notifications** function as intended.

3.2.8 Assumptions and Dependencies

This section outlines critical assumptions and system dependencies.

Key Assumptions:

- The organization will provide access to required resources like Firebase API keys and cloud storage credentials.
- Users will have stable internet connectivity to support real-time synchronization.

Dependencies:

- The system depends on **Firebase Authentication** for secure login.
- Integration with **third-party APIs** for shipment tracking and document storage is required.
- Future updates may require version upgrades for ReactJS, Vue.js, and Firebase services.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

The development of the mobile application for the cargo company will follow an Agile methodology, emphasizing iterative development, collaboration, and flexibility. Agile allows for frequent reassessments and adjustments, ensuring that the project can adapt to changing requirements and incorporate user feedback throughout the development process. The team will operate in sprints, typically two to four weeks long, during which a set of features will be designed, developed, tested, and reviewed.

The User-Centered Design (UCD) approach will be employed to ensure the application meets the needs and expectations of its users, including deliverers, officers, and the Chief Officer. This involves gathering user feedback through interviews, surveys, and usability testing at various stages of development.

For backend services, the Serverless Architecture using Firebase will be selected due to its scalability, ease of integration, and real-time capabilities. This choice reduces the overhead of managing server infrastructure, allowing the team to focus on application features and user experience.

ADVANTANGES OF PROPOSED SYSTEM

- **Enhanced Security:** The platform's robust security features ensure that user data is always protected.
- **Real-Time Capabilities:** Real-time data synchronization improves user experience and application responsiveness.
- **Scalability:** The system can easily scale to accommodate growing user bases and data volumes.
- **Developer-Friendly:** Firebase's comprehensive tools and documentation streamline the development process.

1.SELECTED METHODOLOGY OR PROCESS MODEL

The chosen methodology for the proposed system is the Agile methodology. Agile was selected due to its iterative nature, which promotes flexibility and adaptability throughout the development process. This approach encourages collaboration, continuous feedback, and incremental improvements, which align well with the project's requirements.

The development of the mobile application for the cargo company will follow an Agile methodology, emphasizing iterative development, collaboration, and flexibility. Agile allows for frequent reassessments and adjustments, ensuring that the project can adapt to changing requirements and incorporate user feedback throughout the development process. The team will operate in sprints, typically two to four weeks long, during which a set of features will be designed, developed, tested, and reviewed.

The User-Centered Design (UCD) approach will be employed to ensure the application meets the needs and expectations of its users, including deliverers, officers, and the Chief Officer. This involves gathering user feedback through interviews, surveys, and usability testing at various stages of development. For backend services, the Serverless Architecture using Firebase will be selected due to its scalability, ease of integration, and real-time capabilities. This choice reduces the overhead of managing server infrastructure, allowing the team to focus on application features and user experience.

The Agile methodology follows key principles such as:

- Breaking the project into manageable sprints.
- Delivering functional components after each sprint.
- Encouraging constant communication between team members and stakeholders.
- Prioritizing customer feedback to adapt requirements effectively.

Additional benefits of Agile include enhanced project visibility, improved team productivity, and faster delivery of usable software. With Agile, changes in scope or requirements can be incorporated smoothly, reducing costly rework later. Agile also provides increased transparency by involving stakeholders in review sessions during sprint evaluations. The ability to demonstrate progress and collect feedback at regular intervals enables faster identification and resolution of

issues.

The chosen methodology for the proposed system is the Agile methodology. Agile was selected due to its iterative nature, which promotes flexibility and adaptability throughout the development process. This approach encourages collaboration, continuous feedback, and incremental improvements, which align well with the project's requirements.

The Agile methodology follows key principles such as:

- Breaking the project into manageable sprints.
- Delivering functional components after each sprint.
- Encouraging constant communication between team members and stakeholders.
- Prioritizing customer feedback to adapt requirements effectively.

Additional benefits of Agile include enhanced project visibility, improved team productivity, and faster delivery of usable software. With Agile, changes in scope or requirements can be incorporated smoothly, reducing costly rework later. Agile also provides increased transparency by involving stakeholders in review sessions during sprint evaluations. The ability to demonstrate progress and collect feedback at regular intervals enables faster identification and resolution of issues.

A typical Agile sprint in this project will include the following steps:

- **Sprint Planning:** Identifying goals, defining tasks, and assigning resources for the sprint cycle.
- **Daily Stand-ups:** Conducting short meetings for team members to discuss progress, blockers, and priorities.
- **Development Cycle:** Coding, integrating, and continuously testing modules within the sprint duration.
- **Sprint Review:** Presenting the completed work to stakeholders for feedback.
- **Sprint Retrospective:** Analyzing the sprint’s successes and identifying improvement areas for future cycles.

2. ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

The proposed system architecture and hardware block diagrams are shown

below respectively.

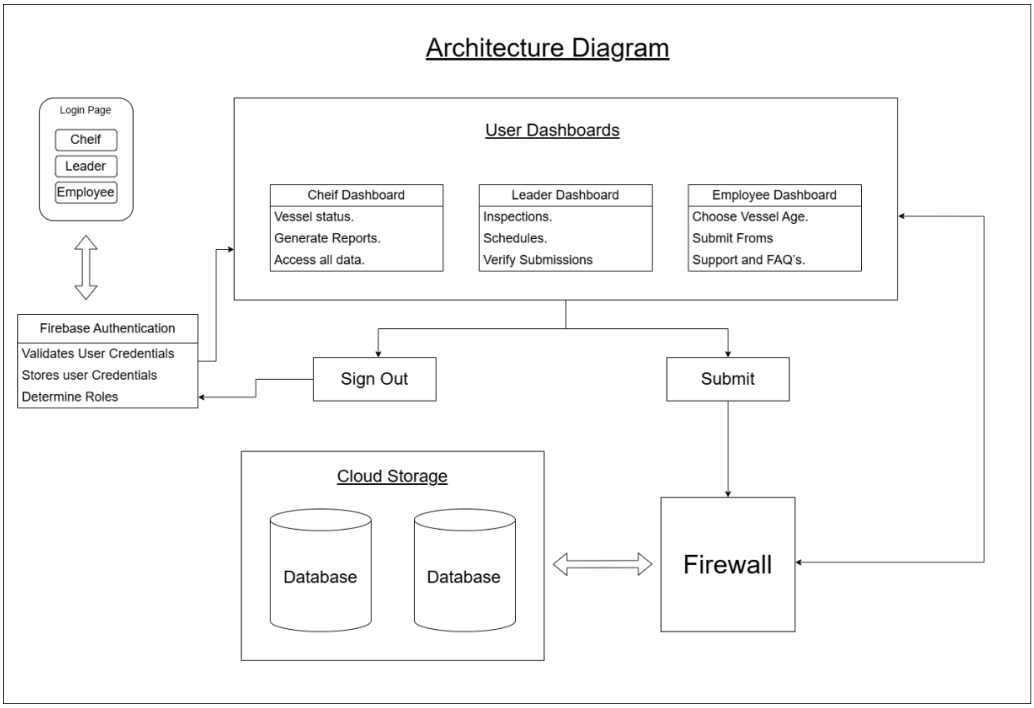


Fig 4.1: Proposed System Architecture

WORKING

In this project the proposed system follows a multi-tier architecture consisting of:

- **Presentation Layer:** Developed using React JS, this layer will provide a dynamic and responsive user interface. The UI will incorporate intuitive navigation, interactive dashboards, and customized user experiences to enhance usability. Form validation, visual feedback, and interactive data displays will improve engagement and accuracy. Accessibility features like keyboard navigation, contrast adjustments, and screen reader compatibility will also be integrated.

- **Business Logic Layer:** This layer will include the system's core functionality. It will handle data validation, user authentication, session management, and data processing. Secure protocols will protect data exchange between frontend and backend components. Caching mechanisms will be implemented to improve performance. Error-handling strategies will ensure system stability under load.

- **Data Layer:** The data layer will rely on a secure relational database system that supports complex queries, data indexing, and optimized data storage. Data backups, encryption protocols, and redundancy measures will ensure data integrity and security. This layer will also facilitate efficient data retrieval and filtering mechanisms for enhanced system performance.

- **API Layer:** The API layer will provide efficient communication between the user interface and backend logic. RESTful APIs will be implemented for data requests, user authentication, and integration with third-party services. The APIs will follow RESTful best practices, ensuring improved scalability, security, and performance. Swagger documentation will be provided for seamless integration with other systems.

This architecture ensures scalability, maintainability, and improved performance. It enables modular development, making updates and maintenance simpler over time. A load balancing mechanism will be added to improve system stability and manage heavy user traffic effectively.

Here's a simplified architecture diagram representing the system's components and their interactions:

• **User Devices**: Smartphones and tablets running the mobile app.
• **Firebase Authentication:** Manages user authentication and access control.
• **Firebase Realtime Database / Cloud Firestone:** Stores and synchronizes data such as vessel details, checklists, and user submissions.
• **Firebase Storage:** Handles media files, including photos and videos captured by deliverers.
• **Cloud Functions:** Serverless backend logic for processing data, sending notifications, and managing complex workflows.

3. DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OFTHE PROPOSED MODEL/SYSTEM

The purpose of the Software Requirement Specification is to produce the specification of the analysis task and also to establish complete information about

the requirement, behavior and also the other constraint like functional performance and so on. The main aim of the Software Requirement Specification is to completely specify the technical requirements for the software product in a concise and unambiguous manner.

3.1. React JS

In this project, ReactJS is used as an open-source IDE.

ReactJS basically is an open-source JavaScript library which is used for building user interfaces specifically for single page applications. It's used for handling view layers for web and mobile apps. React also allows us to create reusable UI components. React was first created by Jordan Walke, a software engineer working for Facebook. React first deployed on Facebook's newsfeed in 2011 and on Instagram.com in 2012.

React allows developers to create large web applications which can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in application. This corresponds to view in the MVC template. It can be used with a combination of other JavaScript librariesor frameworks, such as Angular JS in MVC.

Features

- **JSX:** JSX stands for JavaScript XML. It's an XML/ HTML like syntax used by React. It extends the ECMAScript so that XML/ HTML like text can co-exist along with JavaScript react code. This syntax is used by pre-processors like *Babel* to transform HTML like text found in JavaScript files into standard JavaScript objects. With JSX, we can go a step further by embedding the HTML code inside JavaScript. This makes HTML codes easy to understand and boosts JavaScript's performance while making our application robust.

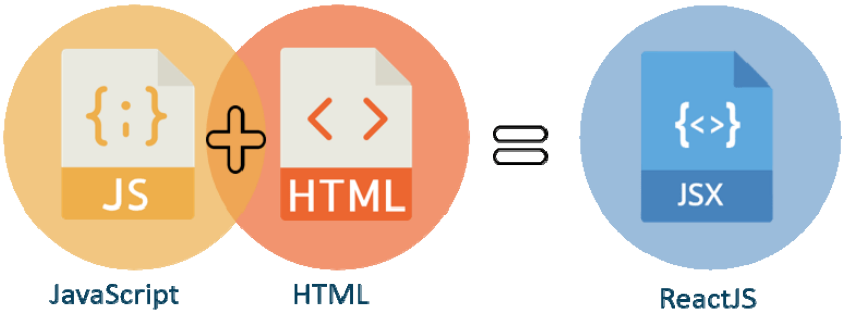


Fig 4.2: ReactJS

3.2. Visual Studio

In this project the Microsoft visual studio is used as an IDE.

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of our way. The delightfully frictionless edit-build-debug cycle means less time fiddling with our environment, and more time executing our ideas.

Visual Studio Code supports macOS, Linux, and Windows - so we can hit the ground running, no matter the platform.

At its heart, Visual Studio Code features a lightning-fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps us be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let us navigate our code with ease.

For serious coding, we'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so wecan step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so we can work with source control without leaving the editor including viewing pending changes diffs.

Customize every feature to our liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with us, and we encourage us to optimize our experience to suit our unique needs.

VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio. VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON.

Architecturally, Visual Studio Code combines the best of web, native, and language-specific technologies. Using Electron, VS Code combines web technologies such as JavaScript and Node.js with the speed and flexibility of native apps. VS Code uses a newer, faster version of the same industrial-strength HTML- based editor that has powered the “Monaco” cloud editor, Internet Explorer's F12 Tools, and other projects. Additionally, VS Code uses a tools service architecture that enables it to integrate with many of the same technologies that power Visual Studio, including Roslyn for .NET, TypeScript, the Visual Studio debugging engine, and more.

Visual Studio Code includes a public extensibility model that lets developersbuild and use extensions and richly customize their edit - build - debug experience.

4. PROJECT MANAGEMENT PLAN

4.1. Executive Summary:

The project aims to develop a comprehensive system using React JS for the frontend and Visual Studio for development. The system will be designed to enhance user experience through efficient data handling, intuitive interfaces, and a secure backend infrastructure. The Agile methodology will guide the development process to ensure flexibility and timely delivery. Regular sprint reviews and stakeholder feedback sessions will be conducted to align the system's development with project goals.

2. Project Scope & Deliverables:

Defines The project scope includes:

- Developing a responsive web application with a user-friendly interface.
- Implementing secure data handling and storage.
- Providing comprehensive documentation for system usage and maintenance.
- Delivering a feature-rich dashboard with comprehensive analytics capabilities.
- Ensuring multi-platform compatibility for desktop and mobile devices.

3. Project Schedule (Gantt chart):

The project will follow Agile sprints, each lasting two weeks. Key milestones include:

- **Week 1-2:** Requirements gathering and design planning.
- **Week 3-6:** Development of core functionalities.
- **Week 7-8:** Integration and testing.
- **Week 9-10:** User feedback and revisions.
- **Week 11-12:** Final deployment and documentation delivery.
- **Post-Deployment Week 13-14:** Maintenance and support period to resolve any reported issues.

4. Project Resources:

The project will require the following resources:

- **Human Resources:** Developers, testers, and project managers.
- **Technical Resources:** Visual Studio, React JS, and a secure database solution.
- **Infrastructure:** Hosting services, testing environments, and security tools such as firewalls and SSL certificates.
- **Additional Tools:** Design tools like Figma for UI prototyping and Trello for sprint management.

5. Risk and Issue Management Plan:

Potential risks include:

- **Technical Risks:** Bugs or compatibility issues may arise. Mitigation strategies include regular code reviews and continuous integration.
- **Resource Risks:** Team member availability may impact progress. The project will maintain backup resources to manage workload.
- **Scope Creep:** Changes in project requirements may extend timelines. Prioritizing core features and using Agile sprints will minimize this risk.
- **Security Risks:** Potential vulnerabilities in the system can be mitigated by implementing robust encryption, multi-factor authentication, and regular security audits.

6. Communication Management Plan:

Effective communication channels will be established using:

- **Daily Standup Meetings:** To track progress and resolve blockers.
- **Weekly Progress Reports:** Shared with stakeholders for feedback.
- **Collaboration Tools:** Platforms like Slack, Jira, or Microsoft Teams will enhance team coordination.
- **Documentation Hub:** A centralized location for sharing project updates, meeting notes, and technical references.
- **Regular Stakeholder Demos:** Frequent system demonstrations to ensure alignment with user expectations.

7. Cost and Quality Management Plan:

- **Cost Management:** The project budget will allocate resources for development tools, team salaries, and hosting services. A contingency fund will be maintained to handle unforeseen costs.
- **Quality Management:** Regular testing, code reviews, and user feedback will ensure high standards. Each sprint will conclude with a review to evaluate performance and identify improvements. Automated testing tools will be implemented to ensure stability across builds.

Review	Work	Status
0th	• Identifying area of work and problem statement • Solution provided • Various study regarding project	Completed
1st	• 20-30% of project completion	Completed
2nd	• 50-60% of project completion	Completed
3rd	• Entire project completion	Completed

Fig 4.3: Work Plan Graph

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 PRESENTATION OF FINDINDS

Technology used: React JS.

Tool used: Vs Code, Android Studio.

We have used Vs Code, Android Studio tools for implementation of this project. We use React JS as our main Technology. It is one of the easiest one to create an application for Android and iOS, so we used this technology, tools to create and execute the application.

5.2 DATA ANALYSIS AND INTERPRETATION

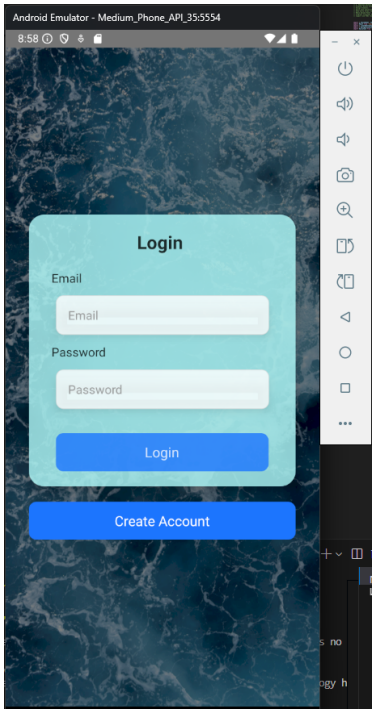


Fig 5.1: Login Page of Application

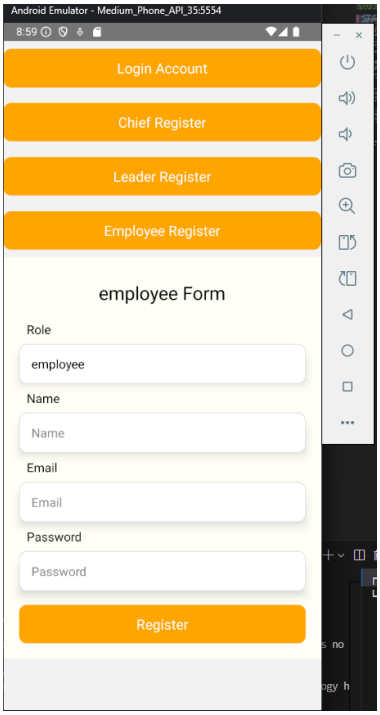


Fig. 5.2 Registration based on roles

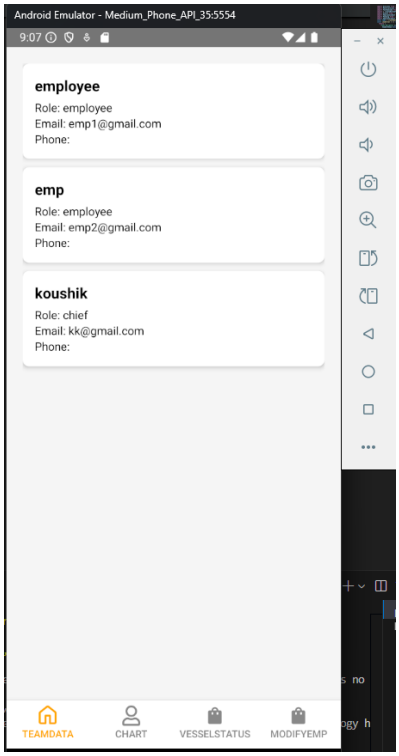


Fig 5.3: Chief Dashboard Interface

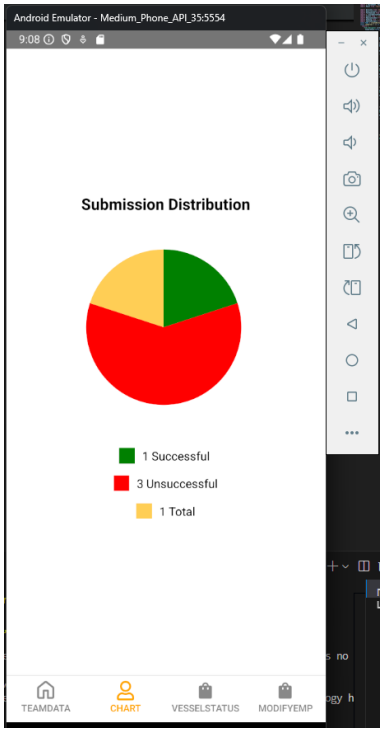


Fig 5.4: Chief's Verification Panel in Dashboard

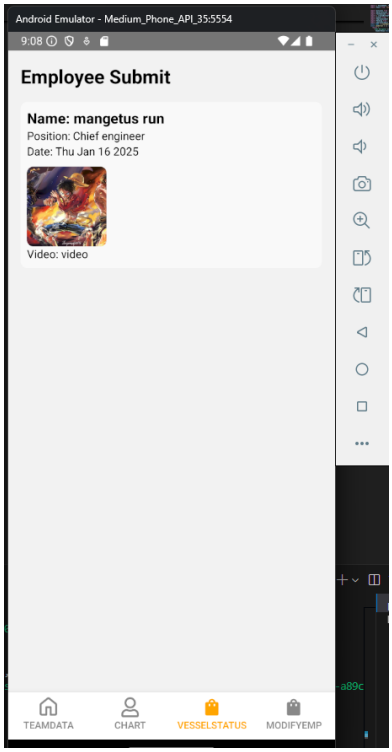


Fig 5.5: Vessel status panel in Chief's Dashboard

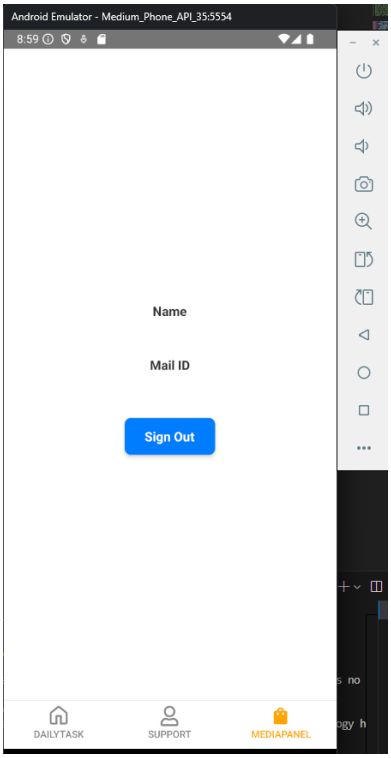


Fig 5.6 Account User Details panel

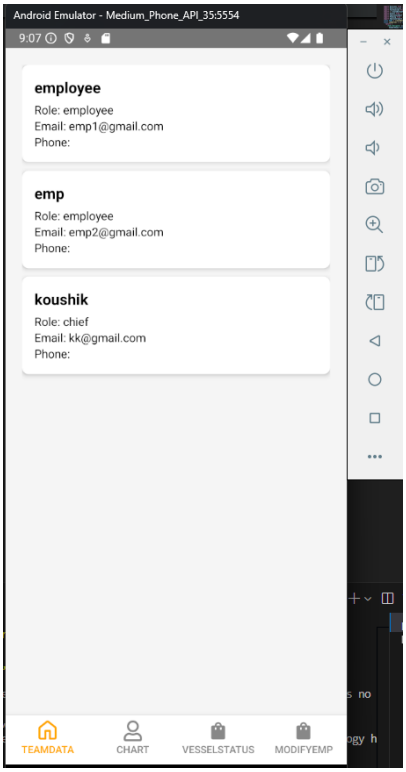


Fig 5.7: Leader Dashboard Panel

Fig 5.8: Vessel age selection page in Employee Dashboard

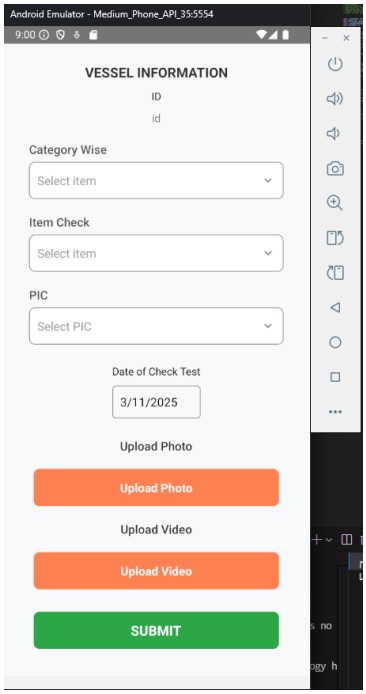


Fig 5.9: Vessel Data Submission panel

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY OF ACHIEVEMENTS

The proposed system achieved several milestones throughout its development process. By leveraging the Agile methodology, the project successfully adapted to evolving requirements while maintaining progress efficiency. The core achievements include:

- **Robust Frontend Development:** Using React JS, the frontend was developed with a responsive and interactive interface. Features such as real-time updates, form validation, and enhanced navigation contributed to improved user experience. React's component-based architecture allowed for code reusability and easier maintenance.
- **Backend Efficiency:** The implementation of RESTful APIs enabled seamless data exchange between the frontend and backend. These APIs were designed with security protocols to protect sensitive data and ensure secure communication.
- **Database Optimization:** The database was structured to improve data retrieval performance. Through indexing, partitioning, and optimized queries, the system efficiently managed large data sets, reducing response times significantly.
- **Comprehensive Testing Strategy:** A rigorous testing process ensured system stability and performance. Functional testing, unit testing, and integration testing were conducted to validate each component. Automated testing tools helped in detecting and resolving issues early in the development process.
- **Effective Project Management:** The Agile methodology, combined with tools such as Trello and Jira, facilitated efficient sprint planning, task allocation, and progress tracking. Regular sprint reviews ensured that the project stayed on schedule and aligned with stakeholder expectations.
- **Enhanced Security Measures:** To protect user data, security features such as data encryption, SSL certificates, and multi-factor authentication were implemented. This ensured the system complied with industry standards for data protection.

6.2 FUTURE DIRECTIONS

To maintain relevance and expand capabilities, the system's future direction includes several key initiatives:

- **Feature Expansion:** Adding advanced analytics features, improved reporting tools, and enhanced visualization elements will provide users with deeper insights and improved decision-making capabilities.
- **AI and Machine Learning Integration:** Leveraging machine learning algorithms to analyze user behavior, predict trends, and recommend actions will enhance the system's intelligence. AI-driven recommendations can improve user engagement and personalization.
- **Mobile Application Development:** Developing a dedicated mobile application will improve accessibility for users on the go. A mobile-friendly interface with intuitive navigation will expand the user base and increase engagement.
- **Cloud Integration:** Migrating to a cloud infrastructure will enhance scalability, data redundancy, and backup capabilities. This will ensure the system can handle increased traffic and evolving data storage needs.
- **Enhanced User Training and Support:** Establishing a comprehensive training program with video tutorials, documentation, and interactive support channels will improve user adoption and reduce learning curves.
- **Continuous Improvement:** Future development will focus on refining existing features, improving performance, and incorporating user feedback to align the system with evolving market demands.

3. IMPLEMENTATION ISSUES

Integrating The development and deployment of the proposed system encountered several implementation challenges, including:

- **Technical Complexity:** Developing a system with multiple layers such as frontend, backend, and database required seamless integration. Coordinating these components to ensure consistent data flow posed challenges that were resolved through regular integration testing and debugging.
- **Resource Limitations:** Limited access to specific hardware and testing

environments restricted the ability to simulate real-world usage scenarios. Cloud-based simulation tools and shared resources were leveraged to overcome these constraints.

- **User Interface Design:** Balancing functionality with an intuitive design required multiple iterations. User feedback was collected throughout the design phase to refine layouts, improve accessibility, and simplify navigation.
- **Security Risks:** The integration of third-party libraries and APIs introduced potential security vulnerabilities. To address this, security audits and penetration testing were conducted to identify and mitigate risks.
- **Data Management Challenges:** Ensuring data consistency, integrity, and availability required developing efficient database schemas. Data validation routines and backup strategies were implemented to prevent data loss and corruption.
- **Project Timeline Management:** The iterative nature of Agile development required frequent adjustments to timelines. To manage delays, tasks were prioritized based on impact and complexity, ensuring critical milestones were achieved on time.

6.4 CONCLUSION

Due to this fact, the cargo management system presented herein implements an efficient solution that caters for data submissions, verifications, and storage related to the cargo business. For overriding, the system selects Firebase for the authentication and database management, and this empowers the system to allow the employees, leaders, chiefs to interact with the system based on their responsibilities.

Employees can enter forms, photo or video information and categorize vessels according to age, and team leaders can validate the input data and report the schedule. Chiefs have total discretion throughout all levels of an organization and the overall co-ordination of an organization is done properly. The promotion of painting and uploading will guarantee accurate storage of data through capturing images and videos through the system. Two among them is the integration that is offered when using various platforms such as web, Android, and IOS. The layout of the system makes it quite flexible such that users of the system in various capacities will find it convenient to use.

Real-time synchronization guarantees that the information and data is current and available at the same time as other people and other systems, promoting easy horizontal work. Its security is a success, having prohibitive measures against outsiders and protecting the stability of the data stored. By using cloud structure, the system has elasticity that will allow for addition of the number of users and the operations in the system as well.

From this project, there have been improvements that has enhanced the operational efficiency, accuracy as well as communication within the cargo operations. It lays a good basis for future upgrades which include incorporating analytics software or AI systems to advance payload management services. The positive outcome of this system shows that it can be used to provide efficient, secure, and scalable cargo management in operations

REFERENCES: -

[1] K. Milojković, M. Živković, N. Bačanin Džakula, “Agile Multi-user Android Application Development With Firebase: Authentication, Authorization, and Profile Management,” in Sinteza 2024- International Scientific Conference on Information Technology, Computer Science, and Data Science, Belgrade, Singidunum University, Serbia, 2024, pp. 405-412. doi:10.15308/Sinteza-2024-405-412.

[2] A Novel Role-based Access Control Model in Cloud Environments January 2016International Journal of Computational Intelligence Systems 9(1):1-9 DOI:10.1080/18756891.2016.1144149 LicenseCC BY-NC.

[3] P. Yang, N. Xiong and J. Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey," in IEEE Access, vol. 8, pp. 131723-131740, 2020, doi: 10.1109/ACCESS.2020.3009876. keywords: computing;Encryption;Data storage;Memory;Cloud {Cloud privacy;Secure storage;data security;cryptography;access control;privacy protection}, .

[4] D. J. C. Sihombing, “Adoption of Agile Approach in Developing Fleet Management System for Cargo Companies”, Jurnal Ekonomi, vol. 13, no. 01, pp. 2354 2363, Mar. 2024.

[5] Ahmadi, Sina, Security And Privacy Challenges in Cloud-Based Data Warehousing: A Comprehensive Review (December 2023). International Journal of Computer Science Trends and Technology (IJCST)– Volume 11 Issue 6, Nov Dec 2023, Available https://ssrn.com/abstract=4683262 at SSRN:

[6] Mobile Application Development with React Native and Leveraging Third-Party Libraries Kralusz,TamasAttila (2024),https://urn.fi/URN:NBN:fi:amk 202404085936.

[7] Di Vaio, A. and Varriale, L. (2019) ‘Digitalization in the sea-land supply chain: experiences from Italy in rethinking the port operations within inter-organizational relationships’, Production Planning & Control, 31(2–3), pp. 220–232. doi: 10.1080/09537287.2019.1631464.

[8] Joszczuk-Januszewska, J., 2014. Development a cloud based ship management platforms. Archives of Transport System Telematics, 7(3), pp.8-12.

[9] Singh, K.U., Varshney, N., Gupta, P., Kumar, G., Singh, T., Dogiwal, S.R. (2024). Mobile Application Control with Firebase Cloud Messaging. In: Hassanien, A.E., Anand, S., Jaiswal, A., Kumar, P. (eds) Innovative Computing and Communications. ICICC 2024. Lecture Notes in Networks and Systems, vol 1020. Springer, https://doi.org/10.1007/978-981-97-3588-4_42 Singapore.

[10] Omotunde, H. and Ahmed, M., 2023. A comprehensive review of security measures in database systems: Assessing authentication, access control, and beyond. Mesopotamian Journal of Cybersecurity, 2023, pp.115-133Caio Davi; André Pastor; Thiego Oliveira; Fernando B. de Lima Neto; Ulisses Braga-Neto; Abigail W. Bigham, “Severe Dengue Prognosis Using Human Genome Data and Machine Learning”, IEEE Transactions on Biomedical Engineering [Vol no: 66, 2019].

A. SOURCE CODE

React JS:- Chief.jsx

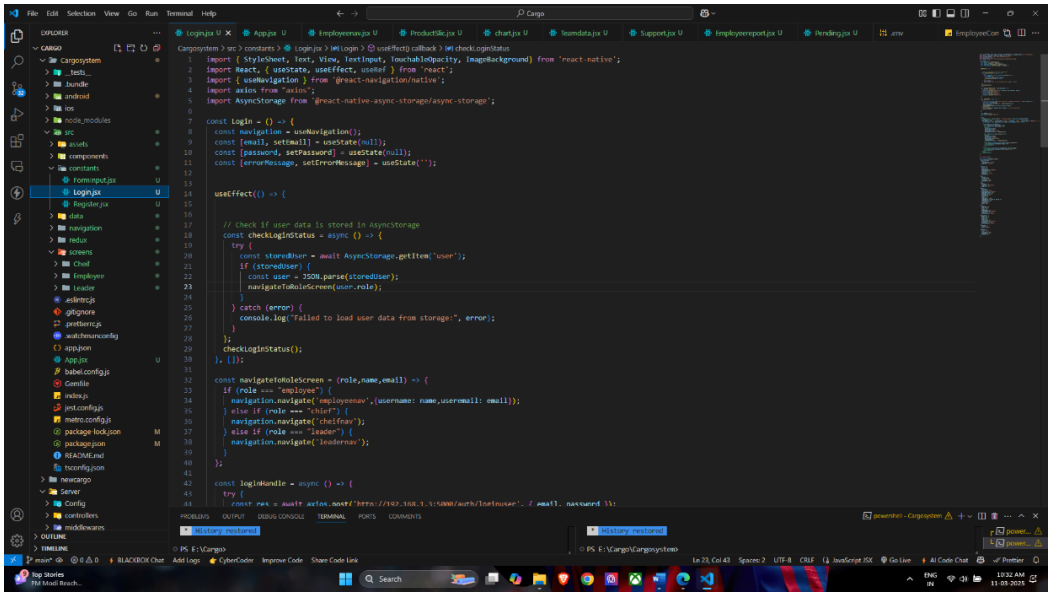
```
import React, { useEffect, useState } from 'react';
import { StyleSheet, View, Text, Dimensions } from 'react-native';
import { PieChart } from 'react-native-chart-kit';
import axios from 'axios';

const Chart = () => {
  const [chartData, setChartData] = useState([]);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get('http://192.168.1.3:5000/emp/countrecord');
        console.log(response.data);
        const processedData = processDataForChart(response.data);
        setChartData(processedData);
      } catch (error) {
        console.error('Error fetching data:', error);
      }
    };

    fetchData();
  }, []);

  const processDataForChart = (data) => {
    return [
      {
        name: 'Successful ',
        population: data.successful,
        color: 'green',
      },
```

```
    {
      name: 'Unsuccessful',
      population: data.unsuccessful,
      color: 'red',
    },
    {
      name: 'Total',
      population: data.count,
      color: '#FFCE56',
    },
  ];
};

return (
  <View style={styles.container}>
    <Text style={styles.title}>Submission Distribution</Text>
    {chartData.length > 0 && (
      <>
        <PieChart
          data={chartData}
          width={Dimensions.get('window').width - 40}
          height={250}
          chartConfig={{
            backgroundColor: '#ffffff',
            backgroundGradientFrom: '#ffffff',
            backgroundGradientTo: '#ffffff',
            color: (opacity = 1) => `rgba(0, 0, 0, ${opacity})`,
          }}
          accessor="population"
          backgroundColor="transparent"
          paddingLeft="90"
          hasLegend={false}
          absolute
        />
```

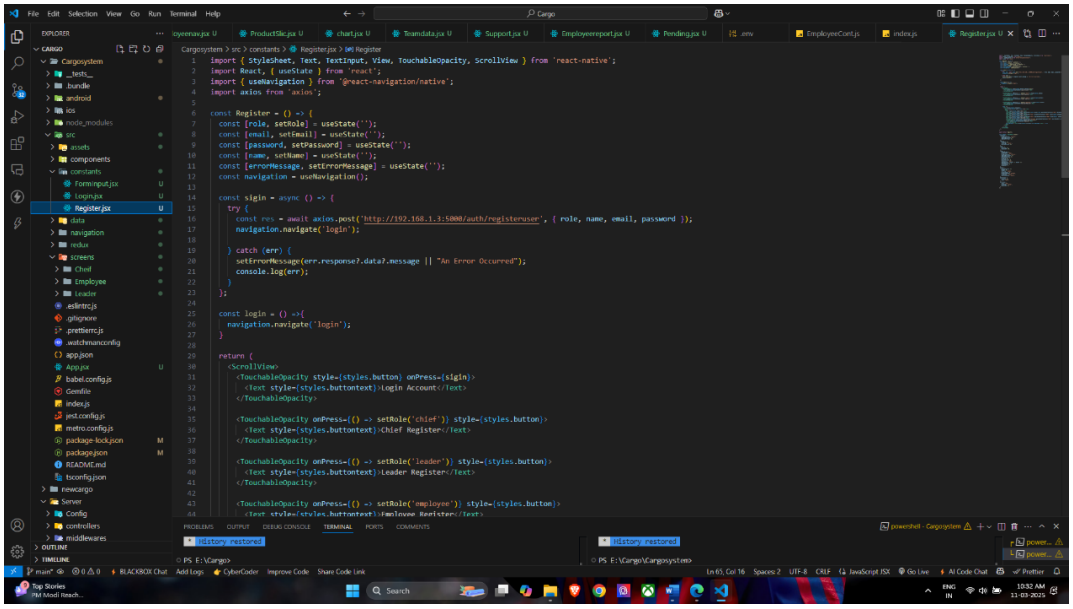
```
        <View style={styles.legendContainer}>
          {chartData.map((item, index) => (
            <View key={index} style={styles.legendItem}>
              <View style={[styles.legendColorBox, { backgroundColor: item.color }]} />
              <Text style={styles.legendText}>{item.population} {item.name}</Text>
            </View>
          ))}
        </View>

      </>
    )}
  </View>
 );
};

export default Chart;

const styles = StyleSheet.create({
 container: {
  flex: 1,
  justifyContent: 'center', // Centers content vertically
  alignItems: 'center', // Centers content horizontally
  backgroundColor: '#fff',
 },
 title: {
  fontSize: 20,
  fontWeight: 'bold',
  marginBottom: 20,
  textAlign: 'center', // Center-align the text
 },
 legendContainer: {
  marginTop: 30,
  alignItems: 'center',
  justifyContent:'center',
```

```
  },
  legendItem: {
    flexDirection: 'row',
    alignItems: 'center',
    marginBottom: 15,
  },
  legendColorBox: {
    width: 20,
    height: 20,
    marginRight: 10,
  },
  legendText: {
    fontSize: 15,
    color: '#000',
  },
  countContainer: {
    marginTop: 30,
    flexDirection: 'row', // Row alignment for the count items
    justifyContent: 'space-around',
    width: '80%', // To control the width
  },
  countItem: {
    alignItems: 'center',
  },
  countText: {
    fontSize: 18,
    fontWeight: 'bold',
    color: '#000',
  },
  countLabel: {
    fontSize: 14,
    color: '#7F7F7F',
  },
});
```

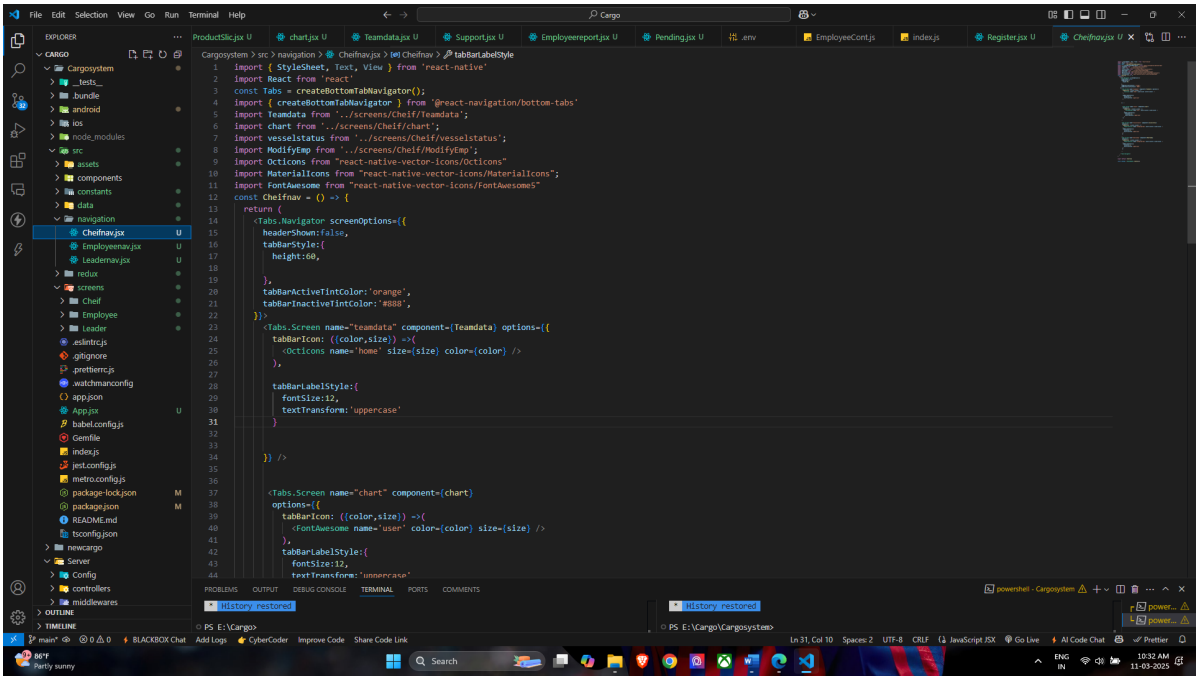
B. SCREENSHOTS



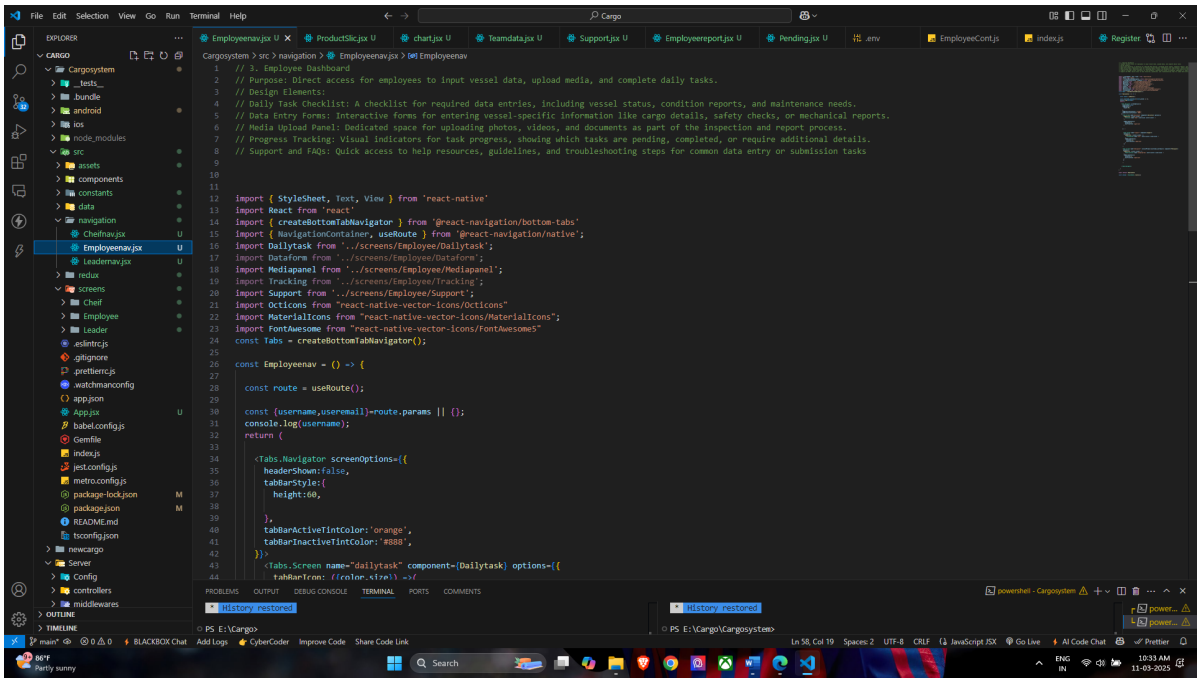
Login.jsx File



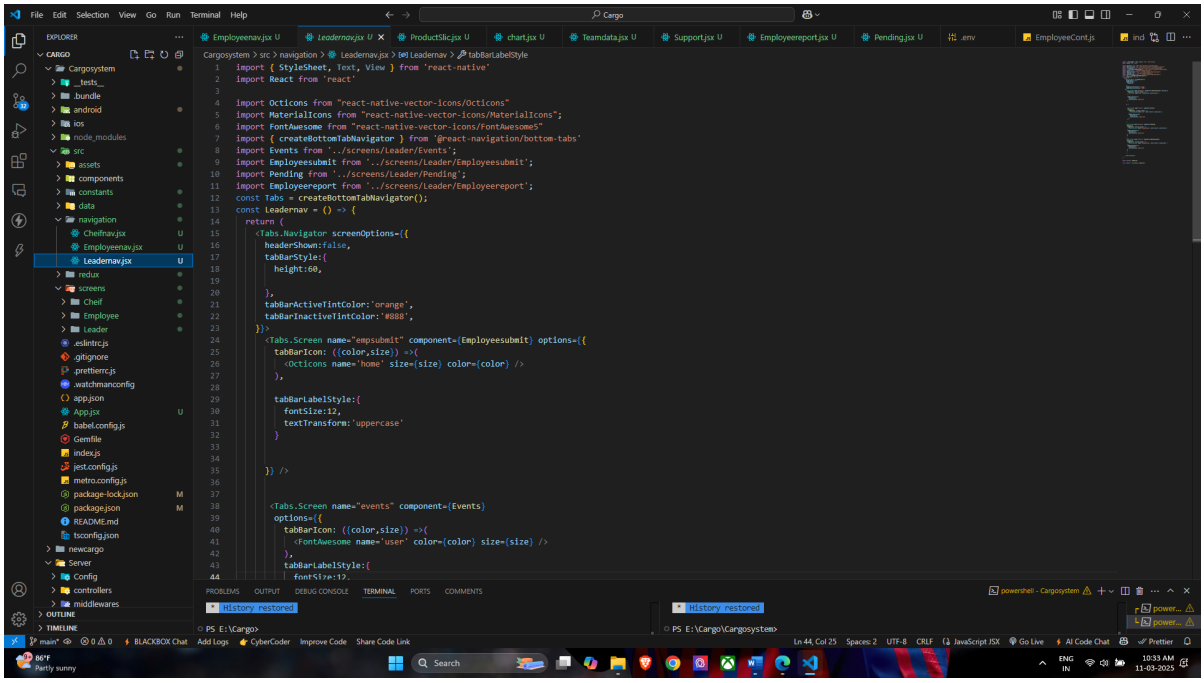
Register.jsx File



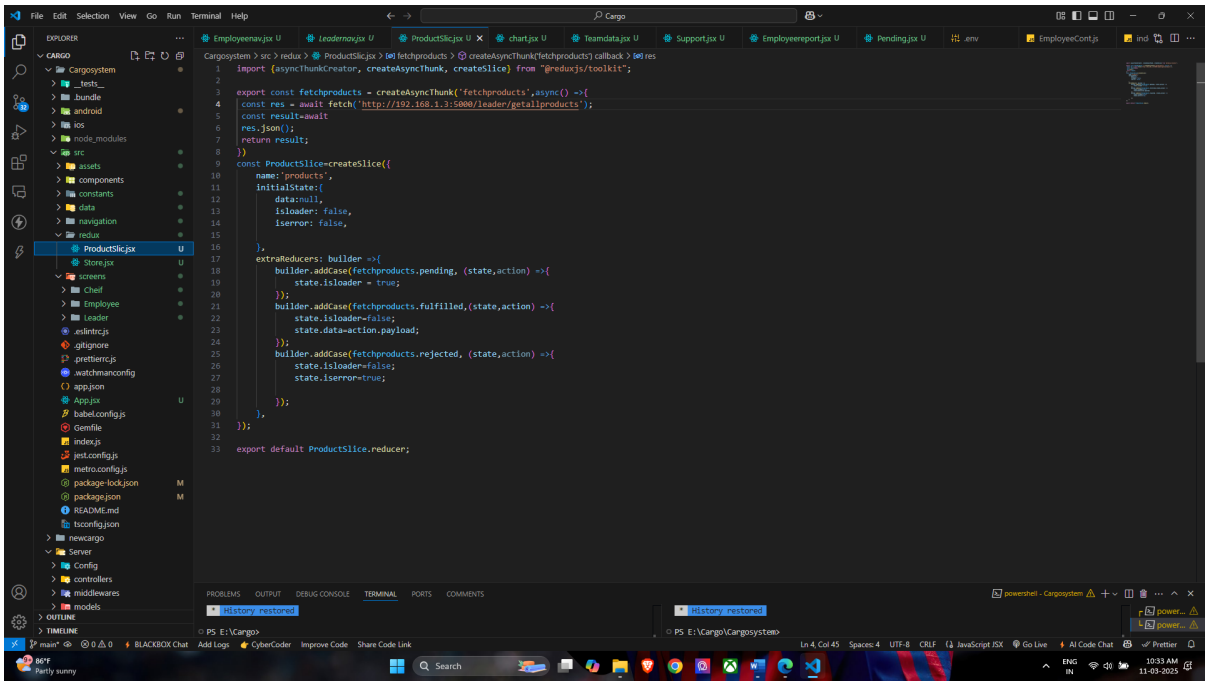
Cheifnav.jsx File



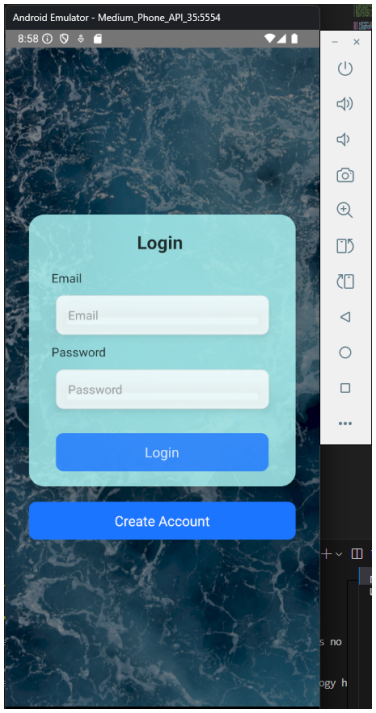
Employeenav.jsx File



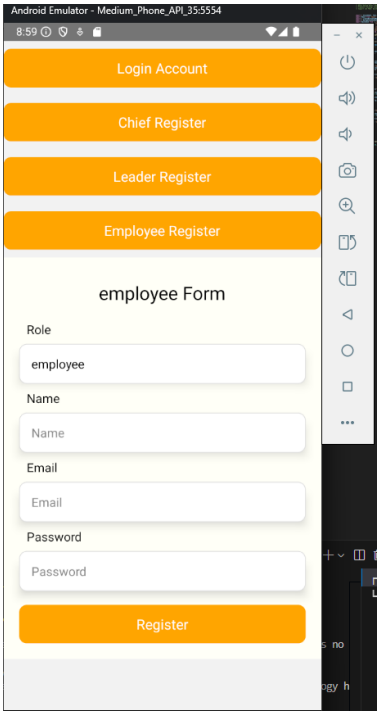
Leadernav.jsx File



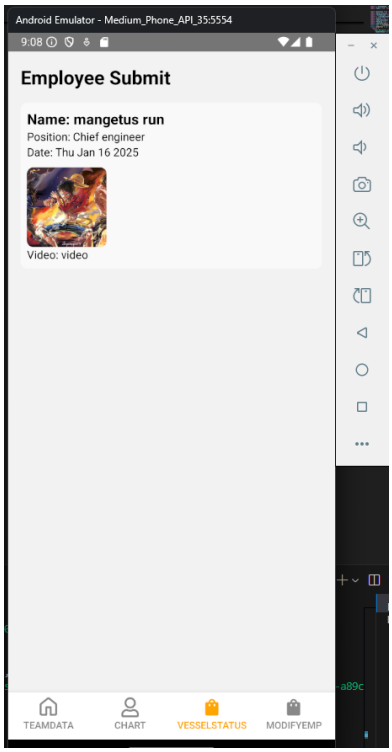
Productslic.j



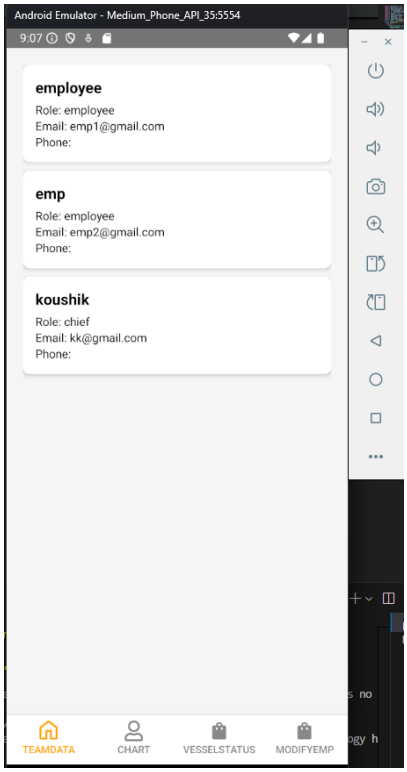
Login Page of Application



Registration based on roles

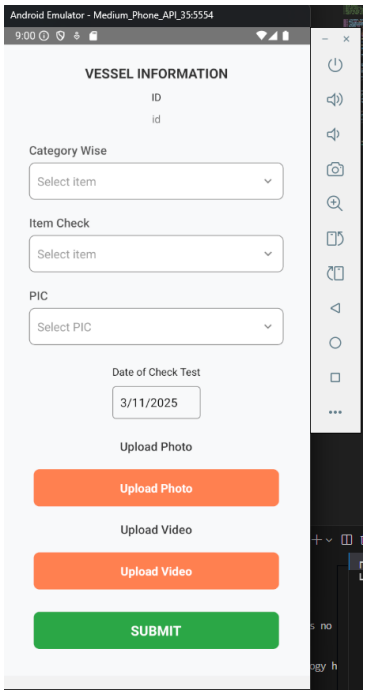


Vessel status panel in Chief's Dashb



Leader Dashboard Panel

Vessel age selection page in Employee Dashboard



Vessel Data Submission panel

C. Publication:

Cloud connected data submission platform using Firebase security

Dr.S.Prince Mary.,M.E.,Ph.D

dept. Computer science and engineering

Sathyabama institute of science and technology
Chennai,India
princemary.cse@sathyabama.ac.in

G.Koushik

dept. Computer science and engineering

Sathyabama institute of science and technology
Chennai,India
koushik23@gmail.com

K.L.V.Uttishtta

dept. Computer science and engineering

Sathyabama institute of science and technology
Chennai,India
uttishttakambham@gmail.com

***Abstract*—The goal of this project is to incorporate an extensive cargo management site which together with data secured with role based access control for managing and verifying vessel data for a cargo company. Conventional cargo management solutions applied at today's companies can be described as decentralized and paper-based, which results in numerous issues including lost time and enhanced risks of errors. To counter these practices, this work proposes a cloud coupled, multi-type, web site available in Web, Android, and iOS environments. With such tools incorporated in the platform, the program employs Firebase that is effective in handling data securely and making essentials updates in real time. This work has the following goals: (1) access control, where the roles assigned are chief officer, team leader, and employee, meaning each has specific functions and levels of data access; (2) enriched data input by integrating image and video uploading of specific vessel types according to age; and (3) ocean and cargo design to create user engagement. Some of the main benefits of this proposed solution are – decreased chances of errors, stronger data security and free data availability, easier verification and an effective system of data access based on role played with reference to the typical industry security standards. This way does not only enhance the efficiency of the service but also creates a foundation for future prospects of adding it to other logistics services.**

Keywords—Firebase,role-based-access,cargo-management,cloud storage

I. INTRODUCTION

In supporting the role of the cargo industry in global flows of trade and supply chains, it remains evident that the challenge of dealing with large volumes of data is not only regular but also intricate. Such issues as sub-optimal flow, poor data protection, and inapplicable solutions that do not accommodate users' needs are common in classic systems. That is why, with the existing increase in use of digital solutions, it becomes significant to look for solutions to these issues if logistics is to continue functioning optimally.

Hence, this paper presents a role-based cargo management system, with an integrated and secured Authentication service from Firebase and Cloud storage solutions. This way ReactJS and Vue.js are integrated within the system to enable real time data synchronization as well as smooth inter-platform data access across web, iOS and Android. Thus, the supposed system is designed with multiple interfaces for separate organizational levels, including chief officers, team leaders and ordinary workers, as well as a number of displays accessible for each individual in accordance of his or her duties. This covers the issues of security, effectiveness as well as application of an interface revolving around the user.

Firebase Authentication guarantees safe login and Firebase Cloud Storage offers secure, large-scale storage of data. The role-based model of the system ensures that each user has privy to only the data s/he requires thus enhancing security while at the same time eliminating unnecessary clutter. The chief officer has complete access to the program while the team leaders approve and log the schedules then the employees input the data.

This system's primary purpose is to optimize the cargo process because of data protection, more effective communication and ownership. Real-time synchronization makes it possible for teams to respond professionally and the relative flexibility of the system means that it will be able to evolve as data volumes and industry needs progress. This paper also focuses on discussing the concept of this innovative solution starting from the design phase to the execution, and efficiency and productivity enhancement in cargo management.

II. LITERATURE SURVEY

In [1] ,the authors also explain the reason why project deadlines are hazardous to the quality of software applications, which necessitates the application of agility. To that end, the study provides Firebase as a solution for increasing the development speed. Firebase is a mobile and web application platform where it is utilized to do real-time database, authentication, and server-less backend. In the real case of an Android Studio application, the combined use of Firebase Authentication to manage user access and profile securely as well as Cloud Firestore to provide real-time synchronization of data across client devices is illustrated in the research. It takes care of user-generated content and doesn't require dedicated server architecture for this purpose securely. Firebase allows developers to dedicate time only to the core functionality development as well as enhancing the UI/UX design. The paper also stresses the effectiveness of Firebase in terms of its scalability, velocity of integration, and the fact that projects can switch to it from other methods of handling back-end services; helping with making project development faster, and for dynamic environments.

In [2], based on the analysis of the Cloud environments, the authors consider the user-resource interactions as dynamic and ad hoc. They present a SAT-RBAC model (Security and Availability-based Trust in RBAC) as an extension of the basic RBAC model to Cloud systems. First, components of trust relationships, consisting of security state of the user's host, availability of the network, and protection levels provided by service providers, are recognized and integrated. Second, a security-based scheduling model is proposed to effectively cope with Cloud environments that possess high levels of uncertainty.The trust relationship in SAT-RBAC is divided into three zones: unbelievable, probable believable, & believable. In order to determine trust probabilities in the probable believable zone, a Bayesian technique is applied. Furthermore, there are two prescriptive models derived by the algorithm, which can assess the key trust factors. The feasibility and accuracy of the SAT-RBAC model are verified with experiments on simulated environments based on CloudSim in PlanetLab to detect and remove the abnormal behaviors in Cloud-based applications.

In [3], the authors also analyze such trends as the IoT, smart cities, the digital transformation of enterprises, and the global digital economy that have resulted in the exponential increase in data. This growth has also enhanced the demand for storage commonly known as cloud storage which has further enhanced its growth. A cloud storage service has become crucial to a storage service to manage and store their data as the governments, enterprises and individuals embraced this technology.Nevertheless, the given migration opens certain threats and concerns as hacking, leakage of private information, or violation of privacy rights.Although there are research papers on data security and privacy concerns, this paper identifies that there is a gap of a systematic literature review related to cloud storage systems. This paper therefore seeks to meet that gap by providing a systematic literature review of data security concerns, privacy preservation mechanisms and encryption techniques in cloud storage systems.The first part of the study aims at outlining a general understanding of cloud storage where it will define this phenomenon, classify it and outline its architecture and most importantly its applications. Next, it discusses challenges and necessities in data protection and privacy in cloud systems. It also provides the summary of other encryption technology and countering. Last of all, the paper points to research directions to fill the existing research gaps and inform future research directions on cloud storage security.

In [4],The authors aims at finding the application of Agile methodology in the development of fleet management systems for cargo companies. This research therefore aims at identifying the effectiveness of Agile practices in improving the efficiency of developing the Fleet Management System to meet the dynamic demand within the cargo sector. Some of the research techniques used in this study were literature review, interviews with agile and cargo industry practitioners, user questionnaire and functional testing in conjunction with User Acceptance testing (UAT) of the developed system.The conclusion drawn from the study shows that by adopting Agile methodology there was development of a flexible Fleet Management System. An attractive user interface, integrated vehicle management features, precise shipment tracking mechanism, and improved performance were some of the other features attributed to this application. These improvements show that Agile can successfully provide solutions to manage such fleets' complexities in a constantly evolving sector.Thus, the present research contributes to an understanding of the advantages of Agile methodology for the cargo industry. It underlines its contribution to increase operation effectiveness, develop system functions and provide better services for the cargo companies and end users, thus it is effective model for the contemporary fleet management system development.

In [5],the authors aims to discuss the upbound trend of utilizing cloud data warehouse to deal with big data. Nevertheless, there are challenges that beset this kind of technology especially in the areas of security and privacy. Some risks are high profile consequences such as data leakage, malware infection and theft of data, which are legal infringements of Consumer Privacy Act.Toward this end, practices such as contracts and data ownership policies are advocated in order to deal with the mentioned issues. Further issues are articulated in the paper which includes multiplexity of the cloud computing models, the dynamic and interconnected nature of the cloud environments, and resource requirements that are likely to exert pressures on the available resources or the budget. Such changes raise the following research questions that this study seeks to address in the context of cloud-based data warehousing for private and government organizations.

In [6],The purpose of this thesis is to describe key concepts of the React Native in modern application development and to explain 3rd Party libraries. It also shows the relation between React Native and native develop tools also competencies and jobbers of this tool. In the first section of

the study, the basics of the creation of mobile applications are described, where React Native is dominant.3rd Party libraries that have become popular among the development of React Native applications are presented with the information retrieved directly from their official documentation. A performance comparison is achieved via a literature review and a practical experiment, which tests the UI render speed in applications developed in Kotlin and React Native.The research indicates that while user interfaces in React Native take a similar time to paint as applications developed with Kotlin or Swift, React Native apps run slower, especially when doing heavy lifting. The thesis also introduces utility libraries for improving the application's user interface and component libraries for increasing the efficiency and effectivity of the graphics in React Native applications.

In [7],the author examines the involvement of social computing technologies in reconsidering and reconstructing business processes in inter-organizational systems in seaport organizations. It examines how IT based digital services change operational processes of public and private actors in seaports.By applying the MSC framework on two Italian ports, TPCS & GAIA digital platforms reveal advantages as time scales decreased and also reduced the paper work, promoting sustainability. These improvements enrich how business processes perform and how the sea-land supply chain prospers in terms of efficiency and of inter-organisation relationships.In doing so, the study provides an academically grounded approach and empirical data to filling the identified research gap regarding digital platforms to modernised seaports and achieve sustainable development objectives.

In [8], the author presented the analysis of the major technological trends and advancements in the information and communications technologies (ICT) domain pertaining to their impact on the transportation sector focusing on maritime transport. These trends are critical fundamental enablers in the development and deployment of ITS 'intelligent' transport applications of which Cloud Computing (CC) is most pertinent. Accordingly, this paper is centered on describing how the marine enterprise can benefit from cloud computing technology and how the various forms of cloud computing are useful in providing solutions for big data and complicated computing requirements in the maritime milieu.The main goal of the present work is to identify the value and possible significance of new concepts that utilize cloud computing technology in ship management systems. Through an analysis of how cloud solutions make a difference in the handling, supervising, and performance of the ship, this paper discusses the gains of introducing cloud in the integration system to gain better storage of data, communication in real-time, and increased ability to expand. The study also looks at the benefits and threats associated with cloud computing for maritime organizations and highlights how cloud computing can bring advantage to the management of ships, effectiveness of decision-making processes in the maritime industry and its general effectiveness.By so doing, this paper will have made a small contribution towards shedding light on one of the most profound trends that is helping to revolutionize the maritime sector as well as facilitate the gradual advancement of intelligent transport systems.

In [9],the author focuses the issue of real-time communication in the modern world and the significance of databases in the communication networks. It also discusses Google Firebase as an innovative solution which has a cloud-hosted real-time database service solution; it is necessary to develop effective communication applications. Firebase uses the enhanced cloud of Google; it increases the rate of app deployment and enhances the communication system.The paper gives a critical analysis of Firebase proving that it has the potential of transforming real time communication in different fields. It explains how Firebase is used to foster innovation in application development for solutions beyond geographic constraints. Consumption of information technologies and capability of cloud solutions in using the application is analyzed with regards to/firebase as a key enabler in the digital age for the communication technologies. Finally, the paper acknowledges Firebase as one of the most innovative organizations in the world and the future improvements in communication and cloud technologies.

In [10],the author gives an overview of a database security system in their categories of authentication, access control, encryption, auditing, intrusion detection, and privacy. It is to create and provide potential value in terms of action to stakeholders by providing update on progressing trends and possibilities on protecting databases. From where I stand, it raises questions and concerns about the risks and threats that exists in securing a database.It also differentiates both clear and strong authentication mechanisms, control of access models, data encryption, auditing, intrusion detection, and data leakage containment. Furthermore, the paper discusses the effects of new trends including cloud technologies, Big data and the IoT on the security of databases. Thus, excluding new data, the paper contributes to the development of database security and provides practical recommendations for protecting information in organisations.Additionally, the paper describes the dynamics of the threats they face such as the growth in sophistication of threats as well as sophistication of data environments. They both stress the value of keeping up to date solutions for security issues to be able to counteract groundbreaking developments. At last, this review provides valuable information for organizations for they can improve their database security and protect important information.

III. PROPOSED SYSTEM

Among the measures proposed to be taken to eliminate vulnerabilities of the cargo vessel management website, it is suggested to use Firebase Authentication Service for structuring role-based access to information; Firebase Real time Database for creating the secure user database. This ensures that the process of submission verification as well as storage of any data relating to cargo is accurate, efficient, dependable and easily scalable.

Objectives:

For efficient submission of forms and verification of the data and subsequently approval of data pertaining to cargo operations

For the purpose of developing a highly reliable system for

the authentication process that will guarantee confidentiality and integrity of the data.

In order to create an extremely dependable system for the authentication procedure that will ensure data integrity and secrecy.

To put in place role-based access control, which makes sure that users engage with the program in accordance with their accessibility, roles, and responsibilities.

To make it possible to use a cloud-based infrastructure for the storing and retrieval of huge datasets, such as forms, photos, and videos.

For designing a mobile application that is responsive and compatible with the both Android and ios devices with better performance that includes interactive UI.

3.1. Architecture Diagram Outline

Below is an outline of the architecture diagram for the proposed system:

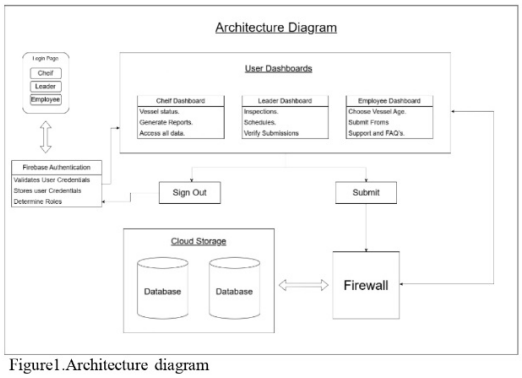


Figure1.Architecture diagram

3.2. Detailed Workflow

The Issues of Authentication and Access Control

Users open the app and get authenticated by their credential via Firebase authentication.

Users or audiences are those who post or comment in response to the organization's content, based on their role, which whether be as an employee, a leader, or a chief, they are automatically taken to the relevant dashboard.

Employee Operations

The targeted employees choose the age group of the vessel, and they complete specific forms concerning the cargo operations.It should be noted that within framework of submission of the documents they can add images and videos of the vessels.The media is stored in the Firebase Cloud Storage that has been linked by the data that has been submitted and stored in the Firebase Realtime Database.

Leader Operations

Team leader will have the access to the data that has been submitted by their own team and is not authorized to have access to the other team leaders verify the data and approve or request modifications.Leaders manage team schedules and track submissions.Leaders also help with the issues that has been raised by their team members by connecting through the communication hub and help resolve the troubleshoot issues and other work related issues.

Chief Operations

Chiefs oversee all activities and data within the system.They generate reports and manage the entire database.They have the access to manipulate the data.Chiefs have access to data verification records and all media files for compliance and review.

3.3. System Architecture Overview

Below is the outline of the internal architecture:

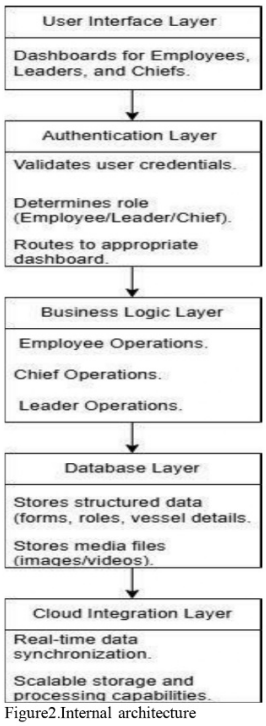


Figure2.Internal architecture

User Interface Layer

This layer is the view layer where users can interact; the application has different categories based on their roles for employees, leaders, and the chiefs.

Employee Dashboard: Allows employees to choose the age range of the vessel (Less than or equal to 5 years, 5 years and greater but less than or equal to 10 years, and More than 10 years) and provide certain forms as well as upload pictures / videos.This interface also includes the support and FAQ's for help resources and any troubleshoot or guidelines problems.

Leader Dashboard: Enables team managers to view and check the entries made by employees, work with schedules and oversee approvals.It also includes Calendar interface for

User Interface Layer

This layer is the view layer where users can interact; the application has different categories based on their roles for employees, leaders, and the chiefs.

Employee Dashboard: Allows employees to choose the age range of the vessel (Less than or equal to 5 years, 5 years and greater but less than or equal to 10 years, and More than 10 years) and provide certain forms as well as upload pictures / videos.This interface also includes the support and FAQ's for help resources and any troubleshoot or guidelines problems.

Leader Dashboard: Enables team managers to view and check the entries made by employees, work with schedules and oversee approvals.It also includes Calendar interface for checking the inspections, schedules, tasks to be done,and the deadlines.

Chief Dashboard: It provides full data access to the chief for supervisory and reporting purposes. Additionally, there is also a feature that shows the vessel status, pending verifications, and recent submissions. It also provides metrics of the teams to view the workload and efficiency across the teams.

Authentication Layer

Firebase Authentication is used in order to make the login secure and apply role based security to the application.Depending on the roles of the users they are diverted to their appropriate dashboards once they sign in.The third layer of this virtual layer provides security is that it would only permit each user accessibility to data that corresponds with their role in the company.

Business Logic Layer

This layer contains the rules and operations for handling role-specific tasks:

Employees can only submit and view their own data.Leaders have the ability to verify submissions and manage team activities.Chiefs can access and manage the entire database.

Includes functionality for:

Verification of data contained in the forms filled in by the employees.

Matching the images / videos submitted in the appropriate forums to the ones uploaded.

Scheduling management, and approval management.

Verifying and solving the issues through communication hub that contains the FAQ's of the employees and coordinating with the team

Database Layer

Firebase Realtime Database for structured data is used consistently for example forms, vessels details, and roles.Data segmentation helps to avoid the situation when one or the other user role receives or has access to information that is not relevant to them.Images and videos are stored and managed separately in Firebase Cloud Storage and then referenced by the nodes of the database.

Cloud Integration Layer

It is integrated with other services that make use of cloud services such as for storage and scalability.Guarantees the integrity of real-time data replication across the multiple user's equipment.In other words, it helps to search, sort and analyze big chunks of information.

IV. Results and Discussions

1. Role-Based Access Control

Table I.

Role	Functions Accessed	Success Rate
Employee	Submit forms, upload media	100%
Leader	Verify submissions, manage schedules	98%
Chief	Access all data, generate reports	100%

a.RBA-Control

Discussion:

The implementation of Firebase Authentication ensures secure access. The role-based redirection mechanism worked seamlessly for all roles. Minor issues were identified during the leader's verification process, which were resolved by optimizing database queries.

2. Data Submission and Verification Workflow

Outcome: The data submitted by the employees were verified thoroughly and efficiently and securely stored in the Firebase Realtime Database

Table II.

Parameter	Measure	Result
Average Form Submission Time	3 minutes per form	Efficient
Verification Time (Leaders)	2-5 minutes per submission	Optimal
Data Retrieval Time (Chiefs)	Real-time synchronization	Real-time

b. Data submission and verification

Discussion:

The use of Firebase Realtime Database ensured quick data submission and retrieval. Leaders managed to verify data effectively, aided by a streamlined dashboard interface. The real-time synchronization facilitated immediate data access for chiefs.

3. Media Uploads and Storage

Outcome: The media upload like images and videos is being uploaded successfully without any issues

Table III.

Media Type	Average File Size	Upload Success Rate	Storage Time
Images	2-5 MB	99%	Instant
Videos	20-50 MB	95%	< 5 seconds

c. Media and storage

Discussion:

Large media files were well managed in Firebase Cloud Storage. Some slowness in video upload was observed during periods of high traffic; this was addressed by optimization that allowed multiple threads for upload.

V. Conclusion

Due to this fact, the cargo management system presented herein implements an efficient solution that caters for data submissions, verifications, and storage related to the cargo business. For overriding, the system selects Firebase for the authentication and database management, and this empowers the system to allow the employees, leaders, chiefs to interact with the system based on their responsibilities.Employees can enter forms, photo or video information and categorize vessels according to the age, and team leaders can validate the input data and report the schedule. Chiefs have total discretion throughout all levels of an organization and the overall co-ordination of an organization is done properly. The promotion of painting and uploading of will guarantee an accurate storage of data through capturing of images and videos through the system.Two the among them is the integration that is offered when using the various platforms such as web, Android, and IOS. The layout of the system makes it quite flexible such that users of the system in various capacities will find it convenient to use. Real-time synchronisation guarantees that the information and data is current and available at the same time as other people and other systems, promoting easy horizontal work.Its security is a success having prohibitive measures against outsiders and protecting the stability of the data stored. By use of cloud structure, the system has the elasticity that will allow for addition of the number of users and the operations in the system as well.From this project, there have been improvements that has enhanced the operational efficiency, accuracy as well as communication within the cargo operations. It lays a good basis for future upgrades which include incorporating analytics software or AI systems to advance payload management services. The positive outcome of this system shows that it can be used to provide efficient, secure, and scalable cargo management in operations.

References

[1] K. Milojković, M. Živković, N. Bačanin Džakula, “Agile Multi-user Android Application Development With Firebase: Authentication, Authorization, and Profile Management,” in Sinteza 2024 - International Scientific Conference on Information Technology, Computer Science, and Data Science, Belgrade, Singidunum University, Serbia, 2024, pp. 405-412. doi:10.15308/Sinteza-2024-405-412

[2] A Novel Role-based Access Control Model in Cloud Environments

January 2016International Journal of Computational Intelligence Systems 9(1):1-9

DOI:10.1080/18756891.2016.1144149

LicenseCC BY-NC

[3] P. Yang, N. Xiong and J. Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey," in IEEE Access, vol. 8, pp. 131723-131740, 2020, doi: 10.1109/ACCESS.2020.3009876. keywords: {Cloud computing;Encryption;Data privacy;Secure storage;Memory;Cloud storage;data security;cryptography;access control;privacy protection},

[4] D. J. C. Sihombing, “Adoption of Agile Approach in Developing Fleet Management System for Cargo Companies”, Jurnal Ekonomi, vol. 13, no. 01, pp. 2354–2363, Mar. 2024.

[5] Ahmadi, Sina, Security And Privacy Challenges in Cloud-Based Data Warehousing: A Comprehensive Review (December 2023). International Journal of Computer Science Trends and Technology (IJCST) – Volume 11 Issue 6, Nov-Dec 2023, Available at SSRN: https://ssrn.com/abstract=4683262

[6] Mobile Application Development with React Native and Leveraging Third-Party Libraries

Królusz,TamásAttila (2024),https://urn.fi/URN:NBN:fi:amk-202404085936

[7] Di Vaio, A. and Varriale, L. (2019) ‘Digitalization in the sea-land supply chain: experiences from Italy in rethinking the port operations within inter-organizational relationships’, Production Planning & Control, 31(2–3), pp. 220–232. doi: 10.1080/09537287.2019.1631464.

[8] Joszczuk-Januszewska, J., 2014. Development a cloud-based ship management platforms. Archives of Transport System Telematics, 7(3), pp.8-12.

[9] Singh, K.U., Varshney, N., Gupta, P., Kumar, G., Singh, T., Dogiwal, S.R. (2024). Mobile Application Control with Firebase Cloud Messaging. In: Hassanien, A.E., Anand, S., Jaiswal, A., Kumar, P. (eds) Innovative Computing and Communications. ICICC 2024. Lecture Notes in Networks and Systems, vol 1020. Springer, Singapore. https://doi.org/10.1007/978-981-97-3588-4_42

[10] Omotunde, H. and Ahmed, M., 2023. A comprehensive review of security measures in database systems: Assessing authentication, access control, and beyond. Mesopotamian Journal of CyberSecurity, 2023, pp.115-133.

Gmail

Koushik Garachetla <koushikgarachetla@gmail.com>

Decision Notification :: Conference on Advances in Communication Networks & Systems (CoaCoNS 2025)

1 message

Microsoft CMT <email@msr-cmt.org> Tue, Feb 18, 2025 at 9:30 AM
Reply-To: Conference on Advances in Communication Networks and Systems <coacons@srmist.edu.in>
To: KOUSHIK GARACHETLA <koushikgarachetla@gmail.com>

Congratulations and Greetings from CoaCoNS-2025!!!
We thank you for submitting your paper to our conference, Advances in Communication Network and Systems (CoaCoNS 2025),
We are pleased to inform you that your paper:

Titled: "Cloud connected data submission platform using Firebase security" having Paper id: "246" is 'ACCEPTED' for Oral Presentation & Publication at CoaCoNS-2025 to be held during 26th - 27th, March 2025 at SRM Institute of Science and Technology (SRMIST), Kattankulathur, Chengalpattu, Tamil Nadu, India.

Every accepted paper must be registered by any one of the authors of the paper. Only registered papers will be submitted to AIP Conference Proceedings for publication. Also, a presentation Certificate will be given to the presenter of the paper during the conference.

REGISTRATION Process of the Accepted paper is as given below:

Authors can verify the Reviewer Comments in the CMT portal. Kindly incorporate reviewer's comments carefully and upload the final paper submission. Keep your final paper in AIP format attached with this mail and submit the same in the below registration form on or before 23rd February 2025.

After making the Payment kindly fill in this Registration form:
https://forms.gle/bUqcDrijFMtDRsTq6

Instructions for Final Camera-Ready Paper Submission:
1. Download the AIP conference template (Microsoft Word), License Agreement (Copyright) and Article Preparation checklist from the following Google Drive link:
https://drive.google.com/drive/folders/1pRwB1efKbgNRTYVJslKVpuF9Vl01ktir For more details:
https://pubs.aip.org/aip/acp/pages/preppapers
2. Incorporate the reviewers' feedback and ensure strict compliance with the AIP Article Template.
3. Ensure the similarity score (plagiarism) is below 10% (excluding bibliography) and that the paper contains no AI-generated content to avoid any issues with AIP for publishing.
4. Submit the following files for further process
a) Upload camera ready paper (Doc/Docx) (File Name Must be: Your Paper ID_Name of the First Author_CoaCoNS.docx (Eg. 51_Prabhu_CoaCoNS.docx)
b) Upload camera ready paper (in PDF). (File Name Must be Your Paper ID_Name of the First Author_CoaCoNS.pdf (Eg. 51_Prabhu_CoaCoNS.pdf)
c) Upload your AIP_Conference_Proceedings_License_Agreement (Attached in the drive) (File Name Must be Your Paper ID_Name of the First Author_Copyright_CoaCoNS.pdf (Eg. 51_Prabhu_Copyright_CoaCoNS.pdf)

Thanks for your understanding and cooperation.

Thanks and regards
Organizing Committee
CoaCoNS 2025