Sri Lanka Institute of Information Technology

Project Proposal Report
Information Technology Project (IT2080)
2024
ITP24J_b03_01

Lanka Glass House –

Glass House Store Management System

Submitted by:

	Name with Initials (Surname first)	Registration Number	Contact Phone Number	Email
1.	Chithrasena D.C.M.A	IT22114112	0713969545	abhishekchitresena0707@gmail.com
2.	Wickramarathne H.I.B	IT22239198	0740056156	haritheb@gmail.com
3.	Wickrmasinghe T.B.B	IT22201614	0777958497	thidaswickramasinghe8@gmail.com
4.	Herath H.M.M.G.E.H.B.	IT22198136	0766981768	herathharshaka@gmail.com
5.	Praveesha R.G.R	IT22299970	0776306246	rashmipraveesha@gmail.com
6.	Priyashantha D G M D	IT22295392	0773226037	manojdulanjana7678@gmail.com
7.	Dias K.T.D	IT22125002	0717915534	tharindudeshan48@gmail.com
8.	Tennakoon T.M.S.S	IT22197832	0772254473	sapurnat@gmail.com

Contents

Contents

Background

Company Background

Lanka Glass House located in Kandy, Sri Lanka, is a renowned provider of high-quality glass products and custom solutions. Serving residential, commercial, and industrial clients, they specialize in delivering exceptional service and craftsmanship. It consist of a team of skilled professionals who are dedicated to meet the unique needs of each project, ensuring superior quality and customer satisfaction. Through their extensive product range and bespoke solutions, they have built a local reputation as a trusted leader in the glass industry.

What is an E-Commerce platform?

An e-commerce system, often known simply as an online marketplace, is a digital platform designed for commercial transactions conducted over the internet. The term "e-commerce" originates from electronic commerce, highlighting the electronic nature of buying and selling products and services in the virtual realm. These platforms serve as dynamic hubs connecting businesses and consumers, offering a wide array of products and services. Educational institutions, enterprises, and individuals alike leverage e-commerce systems for seamless online transactions and interactions. In the contemporary digital landscape, "e-commerce" has become synonymous with online shopping, capturing the essence of the widespread use of these platforms for purchasing goods and services. These systems cater to diverse needs, ranging from electronics and clothing to various household items, providing users with a convenient and accessible avenue for transactions that transcends physical boundaries.

Our Website

Our website for Lanka Glass House is a comprehensive solution designed to streamline business operations and enhance efficiency across all facets of the company. This application integrates eight essential functions:

- Customer Management: Efficiently track and manage customer information and interactions, ensuring personalized service and improved customer satisfaction.

- Employee Management: Simplify employee record-keeping, scheduling, and performance evaluations, facilitating smooth HR operations and boosting productivity.

- Inventory Management: Maintain real-time tracking of glass products and supplies, optimizing stock levels and minimizing shortages.

- Financial Management: Oversee financial transactions, generate detailed reports, and monitor budgets to ensure accurate and efficient financial operations.

- Order Management: Streamline the entire order processing workflow, from creation to fulfillment, ensuring timely and accurate delivery of customer orders.

- Delivery Management: Coordinate and monitor delivery schedules and logistics, providing real-time updates to ensure prompt and reliable service.

- Maintenance and Replacement: Schedule and track maintenance and replacement tasks, ensuring the longevity and quality of glass products and equipment.

- Supplier Management: Manage supplier information, contracts, and purchase orders, optimizing supplier relationships and ensuring a steady supply of quality materials.

This integrated web application is designed to support Lanka Glass House in delivering exceptional service, improving operational efficiency, and driving business growth.

Problems and Motivations

Problems

In developing a comprehensive web application for Lanka Glass House, we have identified several key issues that need to be addressed to enhance the business's operational efficiency and customer satisfaction:

1. Inefficient Customer Management:

- The current system lacks a centralized way to track and manage customer information, leading to potential lapses in service and missed opportunities for personalized engagement.

2. Employee Management Challenges:

- Manual record-keeping and scheduling for employees are time-consuming and prone to errors, resulting in inefficiencies and difficulties in performance tracking.

3. Inventory Management Issues:

- Without real-time tracking of glass products and supplies, the shop faces challenges in maintaining optimal stock levels, leading to either overstocking or stockouts.

4. Financial Management Complications:

- Managing financial transactions, generating reports, and monitoring budgets manually can be cumbersome and prone to inaccuracies, impacting the financial health of the business.

5. Order Processing Inefficiencies:

- The absence of an integrated order management system leads to delays and inaccuracies in order processing, affecting customer satisfaction and operational efficiency.

6. Delivery Coordination Problems:

- Coordinating delivery schedules and logistics manually can result in delays, miscommunications, and reduced customer satisfaction.

7. Maintenance and Replacement Oversights:

- Keeping track of maintenance and replacement tasks for glass products and equipment is challenging without a systematic approach, leading to potential quality issues and customer dissatisfaction.

8. Supplier Management Difficulties:

- Managing supplier information and purchase orders manually can lead to inefficiencies and disruptions in the supply chain, affecting the availability of quality materials.

By addressing these identified problems through our integrated web application, we aim to enhance operational efficiency, improve customer and employee satisfaction, and support the growth and success of [Your Glass Shop's Name].

<u>Motivation</u>

The motivation behind developing a comprehensive web application for Lanka Glass House comes from the need to enhance operational efficiency and customer satisfaction while supporting the business's growth in a competitive market. By streamlining processes such as customer management, inventory tracking, and order fulfillment, we aim to reduce manual errors and save time, allowing the staff to focus on delivering exceptional service.

Additionally, a robust financial management system will ensure accurate tracking of transactions and budgets, providing a clear picture of the business's financial health. The project is also driven by the goal of empowering employees through better management tools, creating a more organized and productive work environment. Ultimately, the web application seeks to maintain high product quality, foster strong supplier relationships, and position the Business as a leader in the glass industry, ensuring long-term success and sustainability.

Aims and Objectives

Aims

Enhance Customer Satisfaction:

- Achieve a significant increase in customer satisfaction scores through improved service delivery and personalized interactions.

Increase Operational Efficiency:

- Reduce the time spent on manual processes by a considerable margin through automation and streamlined workflows.

Ensure Accurate Financial Management:

- Implement a financial management system that enables accurate reporting and monitoring of budgets, with a goal of reducing financial discrepancies.

Boost Employee Productivity:

- Improve employee productivity through effective management and streamlined HR processes.

Optimize Inventory Control:

- Maintain inventory levels that minimize stockouts and overstock situations.

Streamline Order Fulfillment:

- Achieve a high on-time delivery rate through improved order and delivery management processes.

Maintain Product Quality:

- Establish a maintenance schedule that ensures compliance with quality standards for products and equipment.

Strengthen Supplier Relations:

- Improve supplier communication and efficiency, leading to a increase in timely deliveries of materials.

Objectives

The objectives of the web application for Lanka Glass House focus on enhancing operational efficiency and improving service quality. Each objective targets key areas as follows

1. Develop a Customer Management System: Implement a centralized system to efficiently track customer information and interactions.

2. Create an Employee Management Portal: Establish a module for managing employee records and scheduling.

3. Implement Inventory Tracking: Introduce real-time inventory tracking capabilities to optimize stock levels.

4. Build a Financial Management Dashboard: Develop a dashboard for tracking financial transactions and generating reports.

5. Design an Order Management System: Streamline order processing by creating a system that tracks orders from creation to fulfillment.

6. Establish Delivery Management Features: Integrate delivery management capabilities, including assigning drivers.

7. Set Up Maintenance Tracking: Create a module for scheduling and assigning technicians.

8. Implement Supplier Management Tools: Develop a supplier management system to track supplier information and purchase orders.

System Overview

The web application for Lanka Glass House is a comprehensive management solution designed to enhance business operations and improve user experience. The system consists of four main components: frontend, backend, database, and web API, each of which supports various functions essential for the business.

Frontend:

- The frontend is the user interface, designed using modern web technologies such as React or Angular. It offers an intuitive and responsive experience for users, enabling seamless interaction with the application. Key functionalities include:

- Customer Management: Interfaces for adding, updating, and viewing customer profiles and interactions.
- Employee Management: Tools for managing employee records and personal details.
- Inventory Management: Dashboards for monitoring stock levels, adding new products, and managing inventory data.
- Financial Management: Financial dashboards that display sales data, expenses, and profit margins.
- Order Management: Interfaces for processing orders, tracking order status, and managing customer requests.
- Delivery Management: Features for scheduling deliveries, tracking shipments, and updating customers on delivery status.
- Maintenance Tracking: Modules for scheduling and tracking maintenance tasks for products and equipment.
- Supplier Management: Tools for managing supplier information and purchase orders.

Backend:

- The backend serves as the application’s server-side logic, built with Node.js and Express. It handles business logic, data processing, and user authentication. Key functions include:

- User Authentication: Secure login and registration processes to ensure user data protection.
- Data Validation: Ensures that all input data meets specific criteria before being processed.
- Business Logic: Implements rules for customer management, order processing, inventory updates, and financial calculations.
- Data Management: Manages operations for all data entities in the application.

Database:

- The database is the central repository for storing all application data, utilizing MongoDB as a NoSQL database. Key data structures include:

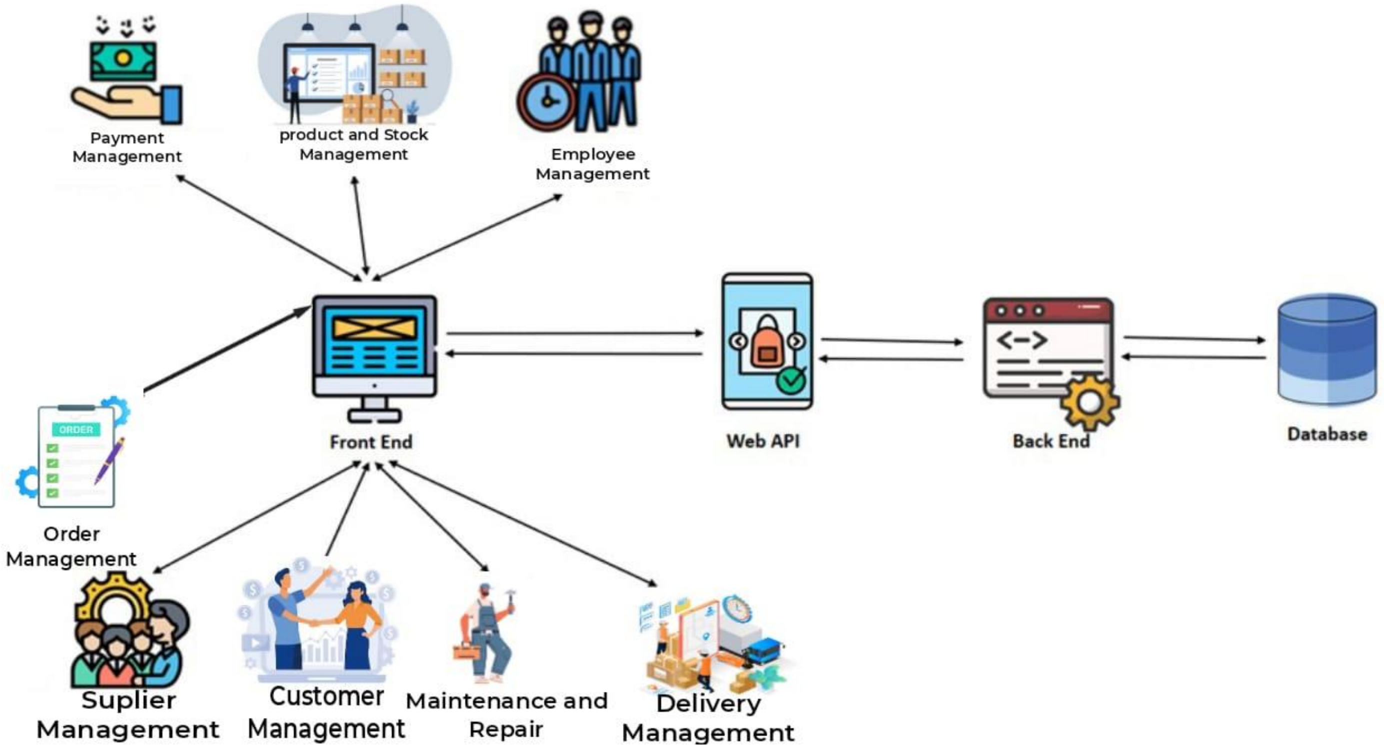
- Customer Data: Information about customers, including contact details, purchase history, and feedback.
- Employee Records: Details about employees, including roles, schedules, and performance metrics.
- Inventory Details: Information about products, including stock levels, descriptions, and supplier details.
- Financial Transactions: Records of sales, expenses, and financial reports for analysis.
- Order Data: Information related to customer orders, including status, dates, and fulfillment details.
- Supplier Information: Details about suppliers, including contact information and contracts.

Web API:

- The web API (Application Programming Interface) facilitates communication between the frontend and backend, allowing for seamless data exchange. Key features include:
- RESTful Endpoints
- Data Security: Implements authentication and authorization protocols to protect sensitive data and ensure that only authorized users can access specific functionalities.

Together, these components and functions create a cohesive and efficient web site that addresses the operational needs of Lanka Glass House, providing a reliable solution for managing customer relations, employee oversight, inventory control, financial tracking, order processing, delivery management, maintenance, and supplier relationships.

System Diagram



1) Handle Customer Function

The Customer Management function is crucial for maintaining a comprehensive database of customer profiles. This function enables users to create, edit, and delete customer records, ensuring that accurate information is readily available. By allowing for efficient search and filtering capabilities, this function facilitates personalized customer interactions and enhances service delivery.

Functional requirements

- **Add Customer Profiles:** Users should be able to enter customer details, including name, contact information, and address. The system must validate the data to ensure completeness (e.g., mandatory fields must be filled).

- **Edit and Delete Customer Profiles:** Users should have the ability to modify existing customer information and remove profiles when necessary, with confirmation prompts to prevent accidental deletions.

- **Search and Filter Customers:** The system must allow users to search for customers by name, contact number, or purchase history. Filters should be available to narrow down results based on specific criteria.

- **View Interaction History:** Users should be able to access a log of past interactions with each customer, including purchases, complaints, and feedback, to facilitate personalized service.

Non-Functional Requirements:

- **Response Time:** The system must return search results within 2 seconds to ensure a smooth user experience.

- **Data Security:** Customer data must be encrypted in transit and at rest to comply with data protection regulations.

- **User Interface:** The UI should be intuitive, with clear labels and an easy-to-navigate layout to minimize training requirements for staff.

.

Technical Requirements

- Development Environment
  o Node.js
- Front-End
  o React.js
- Back-End
  o Express.js
  o Node.js As the server-side runtime environment.
    - o Middleware
- Database
  o MongoDB
    - o Mongoose to access the database
- Authentication and Authorization
  o JSON Web Tokens
- Version Control
  o Git
- Development Tools
  o Visual Studio Code
- Integration and APIs

2) Employee Management Function

The Employee Management function plays a vital role in overseeing the workforce within the organization. It enables users to add, edit, and delete employee records while tracking performance metrics. By assigning roles and permissions, this function ensures that staff have appropriate access to the system, contributing to efficient operations and effective personnel management.

Functional Requirements:

- **Add Employee Records:** Users should be able to input employee information such as name, role, contact details, and employment status. The system should validate this information to ensure accuracy.
- **Edit and Delete Employee Records:** Users should have the capability to update or remove employee information, with the system prompting for confirmation before deletion.
- **Assign Roles and Permissions:** Administrators should be able to set user roles (e.g., admin, employee) to control access to specific functionalities within the application.

Non-Functional Requirements:

- **Concurrency:** The system should support at least 100 concurrent users without performance degradation.
- **Data Access Control:** Access to employee records must be restricted based on user roles, ensuring that sensitive information is protected.
- **Usability:** The interface should be designed for ease of use, allowing employees to quickly find and manage information.

Technical Requirements

- Development Environment
  o Node.js
- Front-End
  o React.js
- Back-End
  o Express.js
  o Node.js As the server-side runtime environment.
    - Middleware
- Database
  o MongoDB
    - Mongoose to access the database
- Authentication and Authorization
  o JSON Web Tokens
- Version Control
  o Git
- Development Tools
  o Visual Studio Code
- Integration and APIs

3)Inventory Management Function

The Inventory Management function is essential for tracking and managing stock levels of products. It allows users to add new items, edit existing product details, and delete items no longer in inventory. This function automatically monitors stock levels, providing alerts for low inventory, and generates insightful reports to optimize stock management and ensure product availability.

Functional Requirements:

- **Add Inventory Items:** Users should be able to input details for new inventory items, including product name, description, price, and quantity. The system must validate that all required fields are filled out.
- **Edit and Delete Inventory Items:** Users should be able to modify or remove items from the inventory, with confirmation prompts to prevent accidental deletions.
- **Alerts:** The system notify when stocks are low.
- **Generate Inventory Reports:** Users should be able to create reports that provide insights into inventory performance, such as turnover rates and stock levels.
- **Search:** The system be able to display products when searched through a search bar.

Non-Functional Requirements:

- **Data Accuracy:** The system must ensure a data accuracy rate of 99.9% for inventory records to prevent discrepancies.
- **Real-Time Updates:** Inventory updates should occur in real-time to reflect current stock levels immediately.
- **Usability:** The system should be intuitive and interactive.

Technical Requirements

- Development Environment
  o Node.js
- Front-End
  o React.js
- Back-End
  o Express.js
  o Node.js As the server-side runtime environment.
    - Middleware
- Database
  o MongoDB
    - Mongoose to access the database
- Authentication and Authorization
  o JSON Web Tokens
- Version Control
  o Git
- Development Tools
  o Visual Studio Code
- Integration and APIs

4) Financial Management Function

The Financial Management function is critical for maintaining the financial health of the organization. It enables users to track all financial transactions, generate detailed reports, and manage budgets effectively. By providing real-time insights into financial performance, this function supports informed decision-making and ensures that the business remains financially viable.

Functional Requirements:

- **Track Financial Transactions:** The system should record all sales and expenses, allowing users to categorize transactions for easy tracking and reporting.
- **Generate Financial Reports:** Users should be able to create detailed financial reports, including profit and loss statements, cash flow reports, and balance sheets.
- **Bill Generation:** The system should Generate bills for customers after a purchase.
- **Provide Real-Time Financial Insights:** Users should have access to dashboards that display key financial metrics and trends in real time.

Non-Functional Requirements:

- **Data Encryption:** Financial data must be encrypted both during transmission and storage to ensure confidentiality.
- **Report Generation Speed:** Reports should be generated quickly to maintain user productivity.
- **Clarity of Information:** The interface for financial reports must be clear and easy to understand, allowing users to quickly interpret the data.

Technical Requirements

- Development Environment
  o Node.js
- Front-End
  o React.js
- Back-End
  o Express.js
  o Node.js As the server-side runtime environment.
    - Middleware
- Database
  o MongoDB
    - Mongoose to access the database
- Authentication and Authorization
  o JSON Web Tokens
- Version Control
  o Git
- Development Tools
  o Visual Studio Code
- Integration and APIs

5) Order Management Function

The Order Management function serves as a central hub for processing customer orders. It allows users to create, edit, and delete orders while providing real-time status updates on order fulfillment. By generating invoices and receipts, this function ensures that all transactions are documented, enhancing customer satisfaction and streamlining the sales process.

Functional Requirements:

- **Create Customer Orders:** Users should be able to input customer orders, including product details, quantities, and payment information, with real-time validation of product availability.
- **Edit and Delete Orders:** Users should be able to modify or cancel orders as needed, with appropriate prompts for confirmation.
- **Generate Invoices and Receipts:** Users should be able to automatically generate invoices and receipts for completed orders, which can be displayed to customers.

Non-Functional Requirements:

- **Processing Speed:** Order processing must be completed quickly to ensure timely service.
- **System Load Handling:** The application should handle peak loads (e.g., during sales events) without crashing or slowing down.
- **Scalability:** System should be able to handle numerous orders at the same time.

Technical Requirements

- Development Environment
  o Node.js
- Front-End
  o React.js
- Back-End
  o Express.js
  o Node.js As the server-side runtime environment.
    - Middleware
- Database
  o MongoDB
    - Mongoose to access the database
- Authentication and Authorization
  o JSON Web Tokens
- Version Control
  o Git
- Development Tools
  o Visual Studio Code
- Integration and APIs

6) Delivery Management Function

The Delivery Management function is integral to ensuring timely and efficient order fulfillment. It enables users to request and track deliveries, providing real-time updates on their status. By notifying customers about their delivery progress, this function enhances communication and improves the overall customer experience.

Functional Requirements:

- **Request Deliveries:** Customers should have the ability to request deliveries by filling out information.
- **Assign Drivers:** The system should provide the admins the feature to assign drivers to different deliveries considering logistics and availability.
- **Generate Delivery Reports:** Users should be able to create reports analyzing delivery performance, including on-time delivery rates and customer feedback.

Non-Functional Requirements:

- **Real-Time Updates:** The system should provide real-time tracking updates with minimal latency.
- **Data Security:** Delivery information must be securely stored and accessible only to authorized personnel.
- **Interface Efficiency:** The delivery management interface should be designed for quick access to delivery details and updates.

Technical Requirements

- Development Environment
  - o Node.js
- Front-End
  - o React.js
- Back-End
  - o Express.js
  - o Node.js As the server-side runtime environment.
    - Middleware
- Database
  - o MongoDB
    - Mongoose to access the database
- Authentication and Authorization
  - o JSON Web Tokens
- Version Control
  - o Git
- Development Tools
  - o Visual Studio Code

Integration and APIs

7) Maintenance Function

The Maintenance Tracking function is essential for ensuring the upkeep of products and equipment. It allows users to schedule maintenance tasks, track their history, and send notifications to relevant personnel. By generating maintenance reports, this function helps the organization stay on top of upkeep needs and optimize maintenance efforts.

- **Functional Requirements:**

  - **Schedule Maintenance Tasks:** Customers should be able to request and schedule maintenance tasks for products and equipment, specifying due dates and responsible personnel.
  - **Track Maintenance History:** The system should maintain a log of all maintenance activities performed, including dates, descriptions, and results.
  - **Generate Maintenance Reports:** Users should be able to create reports detailing maintenance activities, frequency, and costs.

- **Non-Functional Requirements:**

  - **Notification Timeliness:** Notifications about maintenance tasks should be sent according to the scheduled time.
  - **Data Integrity:** The system must maintain the integrity of maintenance records, ensuring they are accurate and tamper-proof.
  - **Usability:** The interface for scheduling and tracking maintenance tasks should be user-friendly to facilitate quick task management.

Technical Requirements

- Development Environment
  o Node.js
- Front-End
  o React.js
- Back-End
  o Express.js
  o Node.js As the server-side runtime environment.
    - Middleware
- Database
  o MongoDB
    - Mongoose to access the database
- Authentication and Authorization
  o JSON Web Tokens
- Version Control
  o Git
- Development Tools
  o Visual Studio Code

Integration and APIs

8) Supplier Handling Function

The Supplier Management function is important for maintaining relationships with vendors and suppliers. It enables users to add, edit, and delete supplier records while tracking purchase orders. By generating performance reports, this function provides insights into supplier reliability and product quality, aiding in informed sourcing decisions.

Functional Requirements:

- **Add Supplier Records**: Admins should be able to input supplier information, including name, contact details, and product offerings, with validation for accuracy.
- **Edit and Delete Supplier Records:** Admins should be able to modify or remove supplier records as necessary, with confirmation prompts for deletions**.**
- **Generate Supplier Reports**: Admins should be able to generate reports on supplier performance, reliability, and product quality.

Non-Functional Requirements:

- **Access Control:** Supplier data must be accessible only to authorized users to protect sensitive information.
- **Retrieval Speed:** Supplier records should be retrievable immediately to maintain efficiency.
- **Data Entry Efficiency:** The interface for managing supplier information should facilitate quick and accurate data entry.

Technical Requirements

- Development Environment
  o Node.js
- Front-End
  o React.js
- Back-End
  o Express.js
  o Node.js As the server-side runtime environment.
    - Middleware
- Database
  o MongoDB
    - Mongoose to access the database
- Authentication and Authorization
  o JSON Web Tokens
- Version Control
  o Git
- Development Tools
  o Visual Studio Code

Integration and APIs

Literature Review

For the literature review, we researched existing websites like Royal Glass House, Kandy Glass House, Central Picture Palace (Pvt) Ltd., and Kings Glass Emporium. We compared their functions with ours to identify areas needing improvement and more attention.

Customer Management

Examining customer management across these websites, it is clear that most of the websites prioritize basic customer information and interaction. For instance, Royal Glass House maintains customer records but lacks advanced CRM functionalities such as detailed interaction histories and personalized service. In contrast, our website offers a robust CRM system that tracks customer interactions, preferences, and provides personalized experiences.

Employee Management

None of the competitor websites offer dedicated employee management functions. For example, Kandy Glass House and Central Picture Palace (Pvt) Ltd. rely on manual HR processes, which can be inefficient and prone to errors. Our website, however, integrates an advanced employee management system, including attendance tracking, performance evaluations, and role-based access control, ensuring efficient management and employee satisfaction.

Inventory Management

Inventory management is a critical function that is partially addressed by competitors. For instance, Royal Glass House and Kings Glass Emporium have basic inventory tracking systems but lack real-time updates and automated reordering. Our web application provides comprehensive inventory management with real-time tracking, automated reordering, and detailed reporting, reducing the risk of stockouts and overstock situations.

Financial Management

Financial management features are also limited in competitor websites. While Kings Glass Emporium and Kandy Glass House manage basic financial records, they do not provide real-time financial insights or detailed reporting. Our solution offers a full suite of financial management tools, including real-time tracking, budget management, and detailed financial reporting, enabling better financial decision-making and planning.

Order Management

Order management functions are commonly found on competitor websites, as they are essential for customer service. Royal Glass House and Central Picture Palace (Pvt) Ltd. have basic order tracking systems, but they lack automation and real-time updates. Our system streamlines the order lifecycle, from creation to fulfillment, with real-time status updates, reducing errors and improving customer satisfaction.

Delivery Management

Delivery management is another area where competitors fall short. Royal Glass House relies on manual scheduling and tracking, which can lead to delays. Our website offers advanced delivery management with real-time tracking, automated scheduling, and customer notifications, ensuring timely and accurate deliveries.

Maintenance Tracking

Maintenance tracking is not addressed by any competitor websites. Our web application includes a structured maintenance tracking system, scheduling and reporting maintenance tasks to ensure optimal performance and prevent costly repairs.

Supplier Management

Supplier management is minimally addressed by competitors. Kings Glass Emporium maintains basic supplier records but lacks comprehensive tracking and performance analysis. Our system provides detailed supplier management, tracking purchase orders, and generating performance reports, enhancing procurement efficiency and supplier relationships.

our literature review highlights the limitations of existing solutions from Royal Glass House, Kandy Glass House, Central Picture Palace (Pvt) Ltd., and Kings Glass Emporium. These competitors lack comprehensive integration, automation, and advanced functionalities that our web application offers. By addressing these gaps, our solution ensures streamlined operations, improved efficiency, and enhanced customer satisfaction, making it a superior choice for [Your Glass Shop's Name]. This review confirms the necessity of developing a tailored system that comprehensively addresses the specific challenges faced by the client.

Functions	Lanka Glass House Kandy	Royal Glass House	Kandy Glass House	Central Picture Palace (Pvt) Ltd.	Kings glass emporium
Financial Handling	☑	☒	☑	☒	☑
Inventory Handling	☑	☑	☒	☒	☑
Customer Handling	☑	☒	☑	☑	☑
Maintenance and Repair	☑	☒	☒	☒	☒
Delivery Handling	☑	☑	☒	☒	☒
Order Handling	☑	☑	☑	☑	☑
Supplier Handling	☑	☒	☒	☒	☑
Employee Handling	☑	☒	☒	☒	☒

Methodology

Methods

Agile Software Engineering Methodology

By using the Agile methodology , Lanka Glass House has demonstrated its dedication to an agile and iterative approach to the development of its e-commerce system. Based on the Agile Manifesto, which emphasizes adaptability, teamwork, and the timely delivery of working software, the approach was developed. Lanka Glass House's development methodology is based on the fundamental principles of the Agile Manifesto, which include an emphasis on people and interactions, usable software, customer collaboration, and adaptability to change.

The Agile methodology is chosen for its adaptability, crucial in the ever-evolving landscape of software development. Lanka Glass House's Agile teams operate in short, focused iterations or sprints, typically spanning one to four weeks. Each sprint is a structured cycle involving meticulous planning, execution, and comprehensive review of development tasks, ultimately leading to the delivery of a tangible software increment.

In the Agile spirit, Lanka Glass House places a strong emphasis on frequent communication and collaboration. Team members actively engage with each other and stakeholders, fostering a shared understanding of goals and ensuring alignment with customer needs throughout the development process. This collaborative approach is integral to the success of the project.

The selected Agile methodology for this project is Kanban, a system known for its adaptability and emphasis on continuous delivery. Kanban's visual board and flow-based approach allow for real-time monitoring of tasks, providing transparency and enabling teams to make informed decisions promptly.

Reasons for Selecting Agile Methodology at Lanka Glass House:

Customer Satisfaction: The Agile focus on continuous and quick delivery is strategically aligned with ensuring prompt satisfaction of customer needs and expectations.

Flexibility: Agile's inherent adaptability allows store's teams to respond swiftly to changing project scopes, market conditions, or emerging opportunities, fostering a responsive and resilient development process.

Collaboration: Agile's strong emphasis on teamwork, communication, and collaboration among team members ensures a collective ownership mindset, leading to the development of optimal solutions.

Transparency: Agile's commitment to open communication fosters transparency throughout the development process. This transparency builds trust among team members and stakeholders, ensuring a shared vision and common goals.

Continuous Improvement: Agile's focus on continuous learning and improvement aligns with Lanka Glass House's commitment to refinement. The iterative nature of Agile processes allows for constant enhancement, leading to the delivery of better results over time.

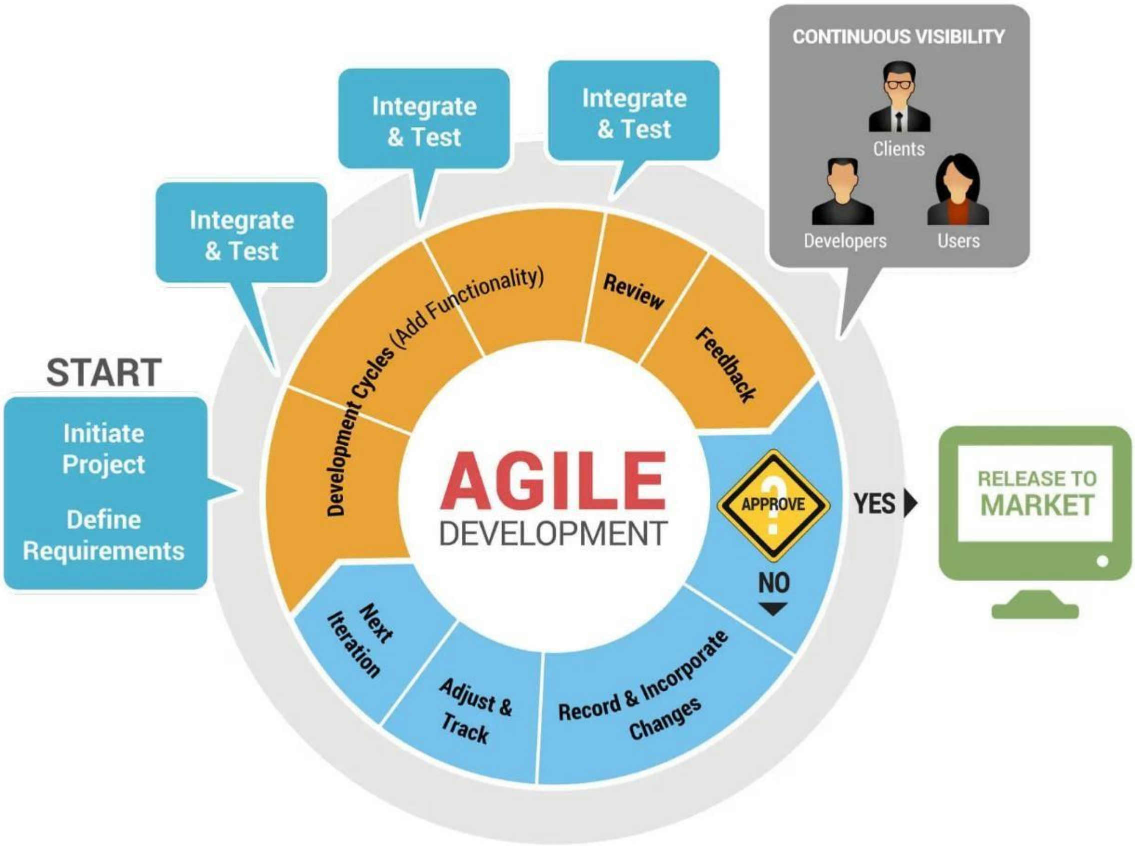
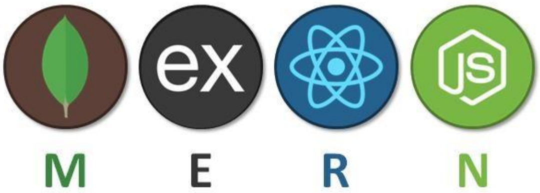


Figure 3 - Agile Methodology

Tools and Technologies

We use MERN stack for developing our web application. MERN is a combination of 4 technologies: MongoDB, Express JS, React JS, and Node JS.



MongoDB is a document database used to build highly available and scalable internet applications. With its flexible schema approach, it's popular with development teams using agile methodologies.

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

React is a declarative, efficient, and flexible JavaScript library for building user interfaces.

Node JS is a single-threaded, open-source, cross-platform runtime environment for building fast and scalable server-side and networking applications.

Apart from MERN Stack we use following tools and technologies in our project.

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.

GitHub is a code hosting platform for version control and collaboration. It lets us to work together on projects from anywhere.

Project plan (Gantt chart)

PROPERTIES	12th June		2nd July	10th July	1st August	8th August		25th June		5th Sept		26th Sept	12th Oct
	W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
Preparation for Project proposal	■	■											
Requirements Engineering			■	■									
planning				■	■								
System UI Design					■	■	■						
Database Design						■	■	■	■				
Coding the Structure							■	■	■	■			
Development							■	■	■	■	■		
Testing											■	■	
Launching the Web Application													■

Figure 2 – Gantt Chart

1]Planning and Discovery (1-2 weeks):

□ Purpose: Understand the project requirements, goals, and objectives.

Activities:

- Initial meetings with stakeholders to gather requirements.
- Conduct competitor analysis to identify strengths and weaknesses.
- Outline the website's goals, features, and functionalities.
- Develop a project plan, including timelines and resource allocation.

2]Design (2-4 weeks):

□ Purpose: Create a visually appealing and user-friendly design.

Activities:

- Develop wireframes to outline the website's layout and structure.
- Create mockups to visualize the design elements and overall aesthetic.
- Finalize the website's visual identity, including colors, fonts, and imagery.
- Gather feedback from stakeholders and make necessary adjustments.

3]Development (4-8 weeks):

□ Purpose: Convert the design into a functional website with coding.

Activities:

- Write code for the website's features and functionalities.
- Implement responsive design for compatibility across devices.
- Integrate databases and other necessary back-end elements.
- Regularly test and debug during the development process.

4]Testing (1-2 weeks):

□ Purpose: Ensure the website functions properly and is free of issues.

Activities:

Conduct various tests, including functionality, compatibility, and performance testing.

- Test the website on different browsers and devices.
- Identify and fix any bugs or issues discovered during testing.
- Seek feedback from users or stakeholders for final validation.

5]Launch (1-2 weeks):

□ Purpose: Move the website from development to a live server.

Activities:

- Migrate the website to a production server.
- Perform final checks to ensure everything is functioning as expected.
- Set up any necessary monitoring tools for ongoing performance.
- Announce and promote the website's launch.

6]Maintenance (ongoing):

***Purpose: Keep the website running smoothly and up to date.**

***Activities:**

- Address and fix any post-launch issues or bugs.
- Implement security updates and patches regularly.
- Provide ongoing support for users and stakeholders.
- Update content and features as needed to meet changing requirements**.**

Evaluation Methods

The evaluation of our web application for Lanka Glass House will be conducted through a comprehensive and structured approach to ensure that the system meets all functional and non-functional requirements, operates efficiently, and provides value to the users. The evaluation will consist of several key components:

1. Functional Testing

Objective: To verify that each function of the system operates as intended.

Method:
- Unit Testing: Each individual component and function (e.g., customer management, employee management, inventory management) will be tested in isolation to ensure correctness.
- Integration Testing: Combined components will be tested together to verify that they work as expected when integrated.
- System Testing: The entire system will be tested to ensure all components work together seamlessly.
- User Acceptance Testing (UAT): End-users (e.g., shop employees, managers) will test the system to ensure it meets their needs and expectations.

Metrics:
- Number of test cases executed.
- Number of test cases passed/failed.
- Bug/issue tracking and resolution rate.

2. Performance Testing

Objective: To ensure the system performs well under expected load conditions.

Method:
- Load Testing: The system will be subjected to a high load to determine how it performs under stress.
- Stress Testing: The system will be tested beyond its normal operational capacity to identify its breaking point.
- Scalability Testing: The system’s ability to scale up or down based on demand will be evaluated.

Metrics:
- Response time under various load conditions.
- System throughput (transactions per second).
- Resource utilization (CPU, memory).

3. Usability Testing

Objective: To ensure the system is user-friendly and meets user expectations.

Method:
- Heuristic Evaluation: Experts will review the system against established usability principles.
- User Surveys and Feedback: End-users will provide feedback on their experience using the system.
- Usability Testing Sessions: Real users will perform tasks while being observed to identify usability issues.

Metrics:
- User satisfaction scores.
- Task completion rates.
- Time taken to complete tasks.

4.Reliability Testing

Objective: To ensure the system is reliable and operates without failure under normal conditions.

Method:
- Failure Rate Analysis: The system's failure rates will be monitored and analyzed.
- Recovery Testing: The system's ability to recover from failures (e.g., power outages, system crashes) will be evaluated.

Metrics:
- Mean time between failures (MTBF).
- Mean time to repair (MTTR).
- System uptime percentage.

5. Beta Testing

Objective: To identify any remaining issues in a real-world environment before full-scale deployment.

Method:
- Controlled Release: The system will be released to a limited number of users in a real-world setting.
- Feedback Collection: Continuous feedback from beta users will be collected and analyzed.

Metrics:
- Number of issues reported by beta users.
- User feedback ratings.
- Time taken to resolve reported issues.

Appendix

System Diagram :

This contains detailed diagrams illustrating the architecture and components of the computer parts and hardware system. These diagrams provide a visual representation of how different elements of the system interact with each other.

Literature Review :

Find an annotated bibliography of relevant literature sources consulted during the research process. Each entry includes a brief summary of the key findings and insights obtained from the respective source

Grant Chart :

Includes a Gantt chart outlining the timeline and milestones for the project related to the development or implementation of the computer parts and hardware system. The chart visually represents the planned schedule for different tasks and activities involved in the project

Methodology :

Detailed description of the methodology employed in the research or development process of the computer parts and hardware system. This appendix outlines the specific techniques, procedures, and tools utilized to achieve the objectives of the project.

Work breakdown structure (work distribution)

	Student ID and Name with initials	**Tasks**
1	Chithrasena D C M A – IT22114112	Implementing Sales & financial Function
2	Wickramarathne H.I.B – IT22239198	Implementing inventory management Function
3	Wickrmasinghe T.B.B – IT22239198	Implementing Customer management Function
4	Herath H.M.M.G.E.H.B. - IT22198136	Implementing maintenance & repair handling Function
5	Praveesha R.G.R – IT22299970	Implementing delivery handling Function
6	Priyashantha D G M D – IT22295392	Implementing Order Handling Function
7	Dias K.T.D – IT22125002	Implementing employee Management Function
8	Tennakoon T.M.S.S – IT22197832	Implementing Supplier Management Function

References

- a. j.watson, "What is an employee management system?," 2024.
- Coursera Staff , "What Is E-commerce? Types, Benefits, and More," 2023.
- "What is agile methodology? Modern software development explained," 06 Apr 2022. [Online]. Available: https://www.infoworld.com/article/3237508/what-is-agilemethodology-modern-software-development-explained.html.
- "What is Mongo DB?," MongoDB, [Online]. Available: https://www.mongodb.com/whatis-mongodb.
- Express JS, [Online]. Available: https://expressjs.com/.
- React JS, [Online]. Available: https://reactjs.org/.
- Node JS, [Online]. Available: https://nodejs.org/en/about/.
- Visual Studio Code, [Online]. Available: https://code.visualstudio.com/docs.
- [3] GitHub, "GitHub," [Online]. Available: https://github.com/chaoocharles/e-shop-assets. [Accessed 25 February 2024].