MBEYA UNIVERSITY OF SCIENCE AND TECHNOLOGY
COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEM AND TECHNOLOGY
COURSE OF INFORMATION AND COMMUNICATION TECHNOLOGY

FINAL PROJECT REPORT: RESTAURANT MANAGEMENT SYSTEM

Course: Mobile Application Development

Instructor: Mr. Amani Rukoijo

MEMBERS:

NAME:	REG NO:
ALLY M. ALLY	24101133370036 (Lead backend Developer)
RAHEL N. MAILA	24101133370045 (Frontend & UI/UX Designer)
JUMANNE M. GIRIMUYI	24101133370075 (security specialist)
ERICA J. MKAMBA	24101133370082 (Project manager & QA Lead)

1. PROJECT OVERVIEW

The Restaurant Management System is a full-stack web application developed to streamline the ordering process in the hospitality industry. The system provides a seamless interface for customers to browse menu items and place orders, while offering a robust administrative dashboard for business owners to manage inventory and monitor sales in real-time.

2. PROJECT TEAM AND ROLES

To ensure the successful implementation of the Restaurant Management System, our group divided the technical tasks based on individual specializations. Below is the breakdown of responsibilities:

2.1 **ALLY M. ALLY** – Backend Developer & Data Logic

Responsibility: Logic Implementation and Data Persistence.

Key Contributions: Developed the core JavaScript functions to handle the application's business logic. This included implementing the Web Storage API (LocalStorage) to ensure that menu items and orders remain persistent. He also engineered the CRUD (Create, Read, Update, Delete) operations within the Admin Dashboard.

2.2 **RAHEL N. MAILA** – Frontend Developer & UI/UX Designer

Responsibility: Interface Design and Styling.

Key Contributions: Focused on the structural design using HTML5 and aesthetic styling using CSS3. He implemented a Mobile-First approach using Flexbox and Grid systems to ensure the application is fully responsive across all device screen sizes as per the project requirements.

2.3 **ERICA J. MKAMBA**– Security Engineer & Session Manager

Responsibility: System Security and Authentication.

Key Contributions: Implemented security protocols using SessionStorage to protect administrative routes. He ensured that the admin.html page is inaccessible to unauthorized users and developed the validation logic for user registration and login credentials.

2.4 **JUMANNE M. GIRIMUYI**– QA Engineer & Documentation Lead

Responsibility: Quality Assurance and Technical Documentation.Key Contributions: Performed rigorous system testing to identify and resolve bugs. He was also responsible for the project’s

technical documentation, including the final Report.pdf, the README.md file for GitHub, and coordinating the 10-minute video presentation.

3. TECHNICAL SPECIFICATIONS (TECH STACK)

The project follows professional development standards by utilizing the following technologies:

Frontend: HTML5 (Structure) and CSS3 (Styling). Used Flexbox and Grid for a modern, responsive layout.

Backend Logic: Vanilla JavaScript (ES6+). Handles all functional logic including authentication and CRUD operations.

Database Integration: Web Storage API (LocalStorage). This ensures data persistence, allowing orders and menu items to remain saved even after a browser refresh or restart.

Version Control: Hosted on GitHub to track development history and facilitate deployment.

4. CORE FEATURES

4.1 Security and Authentication

Role-Based Access Control (RBAC): The system distinguishes between "Admin" and "Customer" roles.

Session Management: Uses sessionStorage to protect administrative routes, ensuring that only authenticated admins can access the management panel.

Input Validation: Ensures usernames are unique and passwords meet the 6-character minimum requirement.

4.2 CRUD Operations (Create, Read, Update, Delete)

Create: Admins can add new food items with specific prices to the database.

Read: The menu is dynamically fetched from LocalStorage and displayed to customers in real-time.

Update: Admins can update their credentials (password management).

Delete: Admins can remove items from the menu when they are out of stock.

4.3 Live Order Monitoring

The Admin panel features an auto-refreshing table that polls the database every 4 seconds to display incoming customer orders without requiring a manual page reload.

5. SYSTEM WORKFLOW

Registration/Login: User selects a role and enters credentials.

Admin Actions: The admin populates the activeMenu key in LocalStorage.

Customer Actions: The customer browses the menu and places an order, which is stored in the orderFeed key.

Order Processing: The admin views the order details (customer name, dish, price, and timestamp) and clears them once fulfilled.

6. MOBILE-FIRST DESIGN (RESPONSIVENESS)

The application is fully optimized for mobile devices. Using media queries and a flexible grid system, the interface adjusts seamlessly from desktop monitors to smartphone screens, fulfilling the modern requirements of mobile-first application development.

7. CONCLUSION: This project successfully demonstrates the integration of frontend design with client-side database management. By utilizing LocalStorage and GitHub for deployment, the system provides a scalable and efficient solution for small-to-medium restaurant businesses