

# Daily Project Management Log

---

Link for video of my project: <https://www.youtube.com/watch?v=f4wnIortIs4>

Name: Sibomana Yvette

REG N: 24RP05289

Project Title: Trustaurant – Blockchain Payment-Based Project

Project Start Date: April 30, 2025

## Day 1

Started the project. Selected the project topic and wrote the proposal document outlining the goals, features, and technologies to be used.

## Day 2

Began development by creating the smart contract for the project. Set up the development environment with necessary tools and configurations.

## Day 3

Designed the user interface for both the Administrator Dashboard and Client Dashboard using a design tool.

## Day 4

Started implementing the frontend design and Backend code. Integrated the UI designs into the actual application using a frontend framework.

## Day 5

Added a new feature to the Administrator Dashboard to enhance functionality.

## Day 6

Identified and fixed bugs in some features that were not working properly. Spent the day debugging and testing functionalities.

## Day 7 and beyond

Deployed the smart contract successfully. Continued making improvements and ensured full integration between frontend and Backend and also smart contract.

Here it is

## **Output structure**

### **Trustaurant - Blockchain-Based Restaurant Management System Presentation**

#### **Slide 1: Introduction**

**Title: Trustaurant - Blockchain-Based Restaurant Management System**

**Tagline: A decentralized application bridging restaurant owners and customers through blockchain technology**

**Key Message: Transparent, secure, and efficient restaurant management powered by smart contracts**

#### **Slide 2: Problem Statement**

**Traditional restaurant management systems face challenges:**

**Lack of transparency in transactions**

**Centralized control of data**

**Limited trust between restaurant owners and customers**

**Inefficient payment processing**

#### **Slide 3: Our Solution**

**Trustaurant leverages blockchain technology to:**

**Create a transparent ordering system**

**Establish trust through immutable transactions**

**Enable secure, direct payments**

**Streamline request management**

**Provide real-time status updates**

#### **Slide 4: Project Overview**

**Trustaurant allows:**

**Restaurant Owners: Manage meals, process customer requests, handle funds**

**Customers: Browse meals, deposit funds, make requests, track order statuses**

**All interactions secured by blockchain technology**

**Slide 5: Technology Stack**

**Layer Tech Used**

**Frontend     React.js + Vite**

**Smart Contracts     Solidity (v0.8.0+)**

**Development Tools Hardhat**

**Blockchain API     ethers.js**

**Wallet Integration     MetaMask**

**Slide 6: Smart Contract Architecture**

**Core logic defined in Trustaurant.sol:**

**Meal management (add, remove, availability toggle)**

**Request lifecycle (submit, approve, track)**

**Role-based access control (Owner vs. Customer)**

**Balance tracking and fund withdrawal logic**

**Slide 7: Features for Restaurant Owners**

**Admin Dashboard Capabilities:**

**Add, delete, and toggle meal availability**

**Review and approve/deny customer requests**

**Monitor restaurant balance**

**Withdraw accumulated funds securely**

**Demo: Show the Admin Dashboard interface**

**Slide 8: Features for Customers**

### **Client Dashboard Capabilities:**

**Browse current meal offerings**

**Deposit ETH into the system**

**Request meals directly from the dApp**

**View real-time status of their requests**

**Demo: Show the Client Dashboard interface**

### **Slide 9: User Flow Demonstration**

#### **Restaurant Owner:**

**Adding a new meal**

**Setting availability**

**Approving/denying requests**

**Withdrawing funds**

#### **Customer:**

**Connecting wallet**

**Depositing funds**

**Requesting a meal**

**Checking request status**

### **Slide 10: Security Considerations**

**Role-based access control restricts admin actions**

**Balance logic prevents unauthorized withdrawals**

**All transactions are validated on-chain**

**Smart contract security best practices implemented**

### **Slide 11: Project Structure**

**plaintext**

```
trustainable/ |—— contracts/      # Smart contract files |  |——  
Trustainable.sol # Main contract |—— frontend/      # React frontend |  
|—— src/      # React components & pages |  |—— public/      # Static  
assets |—— scripts/      # Contract deployment scripts |—— test/  
# Smart contract tests |—— hardhat.config.js  # Hardhat config |——  
README.md      # Project documentation
```

## **Slide 12: Getting Started**

**Prerequisites:**

**Node.js & npm**

**MetaMask browser extension**

**Installation steps:**

**Clone repository**

**Install dependencies**

**Start local blockchain**

**Deploy smart contract**

**Launch frontend**

## **Slide 13: Live Demo**

**Connect MetaMask wallet**

**Show both admin and customer interfaces**

**Demonstrate key features:**

**Adding a meal (admin)**

**Depositing funds (customer)**

**Requesting a meal (customer)**

**Approving a request (admin)**

**Slide 14: Future Enhancements**

**Mobile application development**

**Integration with delivery services**

**Loyalty program using tokens**

**Multi-restaurant support**

**Enhanced analytics for restaurant owners**

**Slide 15: Conclusion**

**Trustaurant demonstrates blockchain's practical application beyond finance**

**Creates a transparent, efficient system for restaurant management**

**Builds trust between restaurant owners and customers**

**Open-source project with MIT License**

**Slide 16: Q&A**

**Thank you for your attention!**

**Questions?**

**Contact: [help@trustaurant.com](mailto:help@trustaurant.com)**