Python Project Report

E-Commerce Cart System

E-NexaMart

Developed By:

- Kruthik BT
- Gowtham Gowda C B
- Akash B V
- Rohith S J

Date: January 05, 2026

1. Introduction

The E-Commerce Cart System is a Python-based command-line interface (CLI) application designed to simulate a virtual shopping experience. It provides a platform for users to browse products, add items to a cart, manage quantities, and generate professional receipts. The project emphasizes a user-friendly experience through the use of color-coded terminal outputs and sound effects, making the interaction engaging and intuitive. Unlike simple scripts, this system features data persistence, ensuring that cart history is saved and retrieving user preferences across sessions.

2. Objectives

- To simulate a real-world shopping cart logic in a CLI environment.
- To implement data persistence using JSON for cart history.
- To enhance user experience with visual (colors) and audio (sound effects) feedback.
- To generate persistent records of transactions via text-based receipts.

3. Key Features

User Profiles: Supports personalized user greetings and guest access modes.

Dynamic Cart Management: Allows adding items with custom names, quantities, and prices dynamically.

Visual Interface: Utilizes the `termcolor` library to present a color-coded, clean table format for the cart.

Audio Feedback: Integrates `playsound` to provide instantaneous audio feedback for success or error states.

Receipt Generation: Automatically generates detailed text receipts in a dedicated 'Saved Files' directory.

Data Persistence: Auto-saves the cart state to `history.json` (handled by `json_savefile.py`) to prevent data loss.

4. Technical Architecture

4.1 Technology Stack

- Language: Python 3.x
- Core Modules: os, datetime, json, pathlib
- External Libraries: termcolor (UI styling), playsound (Audio effects)

4.2 File Structure

File/Directory	Description
main.py	The entry point of the application. Handles the main loop, user menu, and input validation.
models.py	Contains the class definitions: `user` (profile management), `CartSys` (cart logic), and `Sound` (audio handler).
json_savefile.py	Manages File I/O operations for saving cart data to JSON files.
audio/	Directory containing .mp3 files for system sounds.
Saved Files/	Output directory for storing generated receipts and history.

5. Usage Workflow

1. Start the Application: Run `python main.py`.

2. User Login: Enter a name or press Enter for Guest mode.

3. Main Menu: Choose from Add Item, View Cart, Print Receipt, or Exit.

4. Transaction: Upon choosing 'Print Cart Details', a receipt is generated with a timestamp.

5. Exit: The system saves the current session before closing.

6. Conclusion

The E-Commerce Cart System successfully demonstrates the application of object-oriented programming principles in Python to build a functional tool. By integrating external libraries for UI and audio, it bridges the gap between simple script execution and interactive application development. The modular code structure ensures maintainability and scalability for future enhancements.