
Sir Parcel - Logistics & Courier Management System

Sir Parcel is a comprehensive, feature-rich web application designed to simulate the operations of a modern courier and logistics company. Built with Streamlit, it offers a user-friendly interface for customers to track parcels, manage accounts, estimate shipping costs, and interact with an AI-powered assistant.

Table of Contents

- [Key Features](#)
 - [Technology Stack](#)
 - [Project Structure](#)
 - [Setup and Installation](#)
 - [How It Works](#)
 - [Meet the Team](#)
 - [Future Enhancements](#)
-

Key Features

- **User Authentication:** Secure login, signup, and password reset functionality for registered users.
- **Account Management:** Logged-in users can update their profile details, view a complete history of their orders, and download invoices.
- **Public Parcel Tracking:** Anyone can track a package using a Waybill number without needing to log in, viewing a simplified timeline and proof of delivery.
- **Package Claiming:** Users can claim a publicly tracked package and add it to their personal account dashboard to view more detailed information.
- **Dynamic Price Calculator:** A quote tool that estimates shipping costs based on origin, destination, and weight. It uses a primary direct-rate system with a fallback to zone-based pricing.
- **Location Finder:** An interactive tool to find "Sir Parcel" offices by state and city, displaying addresses and contact information.
- **Pickup Scheduling:** A form for users to schedule a parcel pickup, providing shipper, recipient, and package details.
-  **AI Assistant:** A conversational chatbot, powered by the Gemini API, that can answer user questions about public package tracking data.

- **PDF Invoice Generation:** Automatically generates downloadable PDF invoices and proofs of delivery for every order.
 - **Resourceful Tools:** Includes helpful information such as a list of prohibited items, detailed packing guidelines, and a volumetric weight calculator.
-

Technology Stack

- **Framework:** Streamlit
 - **Language:** Python
 - **Core Libraries:**
 - pandas for data handling.
 - httpx and asyncio for asynchronous API requests to the Gemini API.
 - fpdf for generating PDF documents.
 - **API:** Google Gemini for the AI Assistant.
 - **Data Storage:** JSON files for storing user, order, and application data.
-

Project Structure

The project is organized into a main application script and several JSON files that act as a database.

```
.  
|—— buledart.py      # The main Streamlit application logic.  
|—— users.json       # Stores user account data and credentials.  
|—— orders.json      # Stores detailed order information for logged-in users.  
|—— non_login.json   # Stores public tracking data for unclaimed parcels.  
|—— claim_package.json # Logs which parcels have been claimed by users.  
|—— locations.json    # Contains data on all office locations by state and city.  
|—— prices.json       # Defines the zone-based pricing structure (fallback).  
|—— price_estimates.json # Defines direct city-to-city shipping rates (primary).  
|—— f51f4e152830793.Y...jpg # Background image for the UI.  
└—— requirements.txt   # List of Python dependencies.
```

Setup and Installation

To run this project locally, follow these steps:

1. Prerequisites:

- Python 3.8 or higher installed.

4. Install Dependencies:

Bash

```
pip install -r requirements.txt
```

5. Get a Gemini API Key:

- Visit the [Google AI for Developers](#) website to create an API key.
- Open the buledart.py file and replace the placeholder API key in the get_gemini_response function with your own key:

Python

```
# Inside the get_gemini_response function  
  
apiKey = "YOUR_GEMINI_API_KEY_HERE"
```

6. Run the Application:

Bash

```
streamlit run buledart.py
```

The application will open in your default web browser.

How It Works

Parcel Tracking and Claiming

The application manages two types of parcel data:

- **Public Data (non_login.json):** This file holds tracking information for all parcels not yet associated with a user account. It's accessible via the public "Track Parcel" tab and is the knowledge base for the AI Assistant.
- **Private Data (orders.json):** This file contains more detailed order information, including recipient names and product prices. When a user logs in and "claims" a package using its Waybill number, the system creates a detailed record in orders.json linked to their username. The claim_package.json file prevents a package from being claimed multiple times.

Price Calculation

The "Get a Quote" feature uses a two-tiered logic for accuracy:

1. **Direct Rate Check:** It first checks price_estimates.json for a pre-defined, direct shipping rate between the selected origin and destination cities.
2. **Zone-Based Fallback:** If a direct rate isn't found, it uses locations.json and prices.json to determine the shipping zones of the cities and calculates the cost based on whether the shipment is intra-zone, to an adjacent zone, or national.

AI Assistant

The AI Assistant provides an interactive way for users to get information.

- When a user sends a message, the application sends the prompt along with the entire public tracking data from non_login.json to the Gemini API.
 - The system prompt instructs the AI to act as "Sir Parcel AI" and answer questions based only on the provided JSON data.
 - This allows users to ask questions like, "What is the status of my package FMPP1003?" or "Where is the package coming from?" in natural language.
-

🤝 Meet the Team

As per the "About Us" section in the application:

- **Joshuah Vijay:** Team Lead
 - **Arthi D:** Lead Developer
 - **Arthi CA:** UI/UX Designer
 - **Viswanthan SS:** Database Architect
-

🔮 Future Enhancements

- **Database Integration:** Replace the JSON files with a relational (like PostgreSQL) or NoSQL (like MongoDB) database for better scalability, data integrity, and performance.
- **Real-time Tracking:** Implement a mock real-time update mechanism using WebSockets to show location changes on a map.
- **Administrator Dashboard:** Create a separate interface for administrators to manage users, add new parcels, and update tracking statuses.
- **API for Third Parties:** Develop a REST API to allow e-commerce platforms to integrate with the Sir Parcel system for booking and tracking.
- **Test Coverage:** Write unit and integration tests to ensure the reliability and stability of the application.