Automatic Managing Of Courier

Geddam Savithri
PG scholar, Department of MCA, DNR College, Bhimavaram, Andhra Pradesh.
V.Sarala
(Assistant Professor), Master of Computer Applications, DNR college, Bhimavaram, Andhra Pradesh.

Abstract:

This Django-based application aims to streamline courier management by integrating functionalities such as parcel tracking, feedback collection, employee management, and courier status updates. The system uses MySQL as the backend database and enables users to track the status of their couriers in real-time by interacting with the system through web interfaces. The feedback mechanism provides valuable insights into the service quality through a user-friendly display of feedback data, including a pie chart visualization of feedback rankings. Employees and admins can update parcel status and track courier movement on a dynamic map. The system also supports file uploads for images associated with parcels, leveraging Django's file handling capabilities. Key features include parcel tracking with dynamic updates, interactive map integration, courier feedback collection, and an employee login system for secure access to administrative actions. The application provides a comprehensive approach to courier service management, offering both user-friendly interfaces and efficient backend processes for tracking, status updates, and feedback analysis.

I. Introduction

In today's fast-paced and technology-driven world, logistics and courier services play a crucial role in global connectivity. With the exponential growth of e-commerce, the demand for efficient and real-time courier tracking systems has become paramount. Customers expect transparent, reliable, and prompt delivery services, and businesses seek ways to streamline their operations to meet these expectations. In this context, an automated **Courier Management System** built using Django, a high-level Python web framework, proves to be an essential solution

1

consumers.

The project titled **"Courier Management System using Django"** is designed to enhance the efficiency of courier service tracking, status management, and customer interaction. This system not only helps customers to track their parcels but also enables employees to update parcel statuses in real-time, submit feedback, and collect shipment data in a structured manner. The system provides a user-friendly web interface that supports multiple roles such as employees, administrators, and customers, each with tailored functionalities to meet their needs.

The core idea of this project revolves around the integration of modern web development techniques with robust backend logic to ensure real-time data processing and visualization. Developed using Django, the system incorporates powerful Python libraries such as pymysql for database connectivity, matplotlib for dynamic feedback chart generation, and numpy for statistical data handling. The use of Google Maps embedding further enhances the usability by allowing users to visualize current parcel locations directly on a map interface.

II. Literature Survey

A **Courier Management System** is a digital solution that automates the processes involved in managing, tracking, and delivering parcels. Over the years, various courier tracking systems and logistics management platforms have been developed, each addressing specific aspects of the delivery pipeline. This literature survey explores existing systems, their technological frameworks, strengths, limitations, and how the proposed Django-based system builds upon them.

1. Traditional Courier Systems

Before digital transformation, courier services primarily relied on manual record-keeping and telephone-based status updates. Delivery slips and logbooks were used to track parcel movements. However, such systems had significant drawbacks:

Prone to human errors and data loss.

Delayed communication between the delivery team and customers.

No real-time tracking or feedback collection mechanism.

The inefficiencies in traditional courier management prompted the need for digital solutions that automate parcel tracking and status updates.

2. Web-Based Courier Systems

Several web-based courier tracking systems emerged to address the inefficiencies of manual systems. For instance:

Blue Dart and **DTDC** implemented customer-centric web platforms for basic tracking using consignment numbers.

FedEx and **DHL** introduced real-time global tracking features using barcodes and GPS.

While these platforms offered essential tracking functionality, they often had limitations:

Minimal user feedback mechanisms.

Limited visualization of tracking data.

Lack of transparency in location updates and service issues.

Restricted accessibility for internal employees to manage status updates through a unified portal.

3. Android-Based Tracking Applications

With the rise of smartphones, courier companies also introduced mobile apps that allowed users to:

Track shipments via GPS.

Get notifications on delivery status.

Access delivery history.

However, studies such as those by *Kumar et al. (2020)* showed that while mobile apps improved user engagement, they often failed to integrate seamlessly with internal systems used by courier staff, leading to data silos and inconsistencies.

4. Cloud-Based Logistics Platforms

Cloud-based systems such as **Zoho Logistics**, **ShipRocket**, and **ClickPost** provided advanced features:

Centralized databases with multi-user access.

AI-driven delivery optimization.

Integrated customer support ticketing.

Although highly efficient, these systems:

Are mostly subscription-based and costly for small businesses.

Require technical expertise for customization.

May not offer transparency into the backend for educational or open-source learning purposes.

Courier Management System

This online application allows both courier management and users to track and update courier status so both will be aware of current location of parcel. This application can be managed by 3 different users describe below

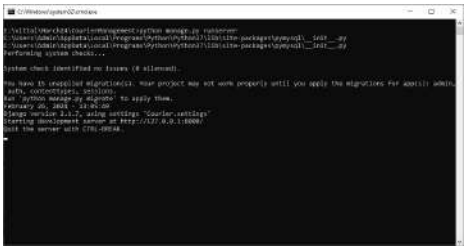
1) Admin: admin can login to system using username and password as ‘admin’. After login admin will add employee details who are responsible to collect courier parcel and update location till deliver. Admin can view list of all couriers and working employees
2) Employee: employee can login to system using username and password given by admin. After login employee will add new courier details and then update its current

location details. Employee can view current location of courier on maps.

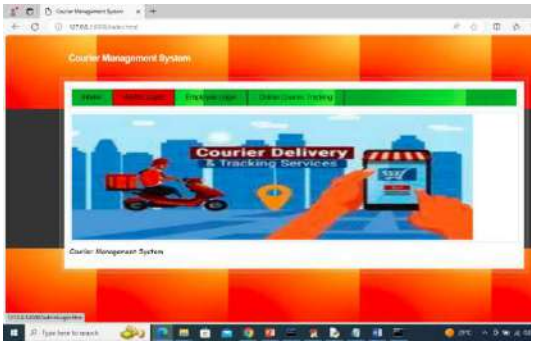
3) User: employee will give unique courier id to each user and using this ID user can track courier status.

To run project install MYSQL and then open MYSQL console and then copy content from 'Database.txt' file and paste in MYSQL console to create database.

Now install python 3.7 and then install all packages given in 'requirements.txt' file and then double click on 'run.bat' to start python web server and get below page



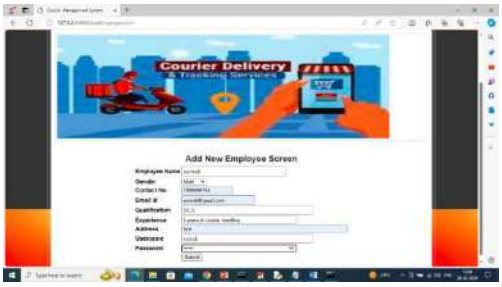
In above screen python server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and press enter key to get below page



In above screen click on 'Admin Login' link to get below login page

In above screen admin is login and after login will get below page

In above screen admin can click on 'Add New Employee' link to add employe details and get below page



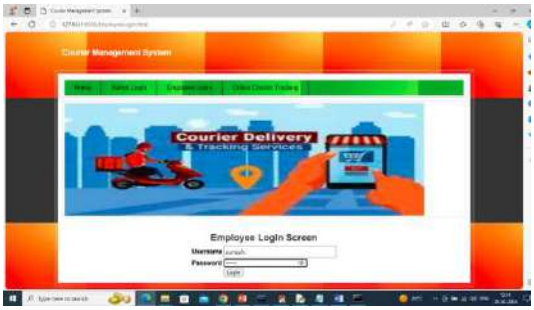
In above screen admin will add employee details and then press button to save details

In above screen employee details added to database and by using this login details employee can login to application and now

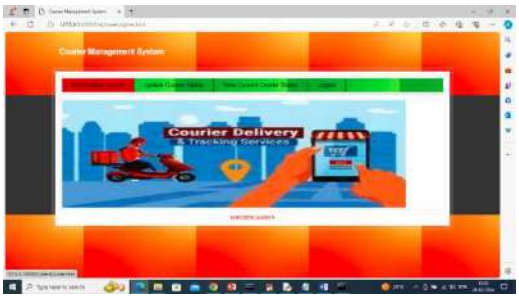
click on 'View Employee' link to get list of all available employees

In above screen admin can details of all employees and now click on 'View Courier List' to view details of all booked couriers

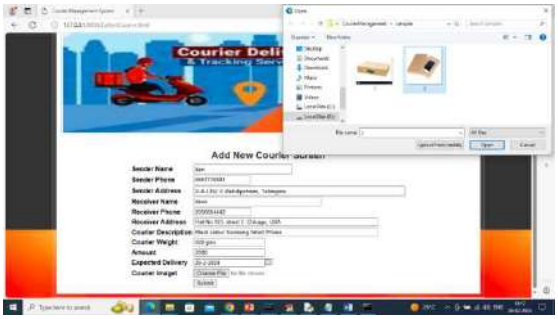
In above screen admin can view list of all booked couriers list and now logout and login as employee



In above screen employee is login and after login will get below page

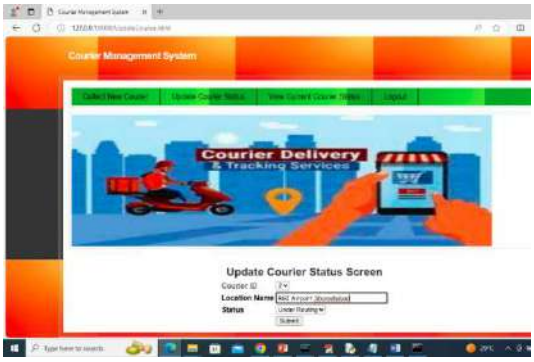


In above screen employee will click on 'Collect New Courier' link to add new courier details



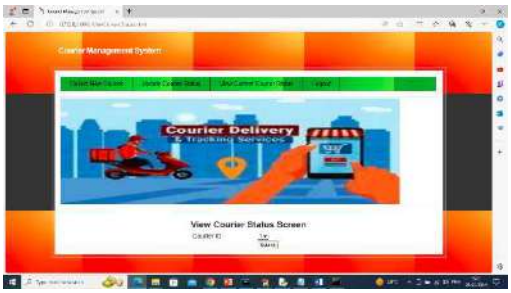
In above screen employee will collect all courier details and then upload courier item image and then press 'Submit' button to add new courier details and get below page

In above screen courier details added and this courier can be tracked using ID as 2 and upon courier progress then employee will click on 'Update Courier Status' link to update location and get below page



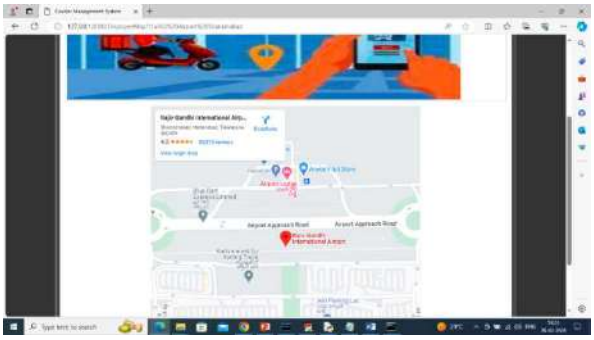
In above screen employee will select courier ID and update current location and this process continues till courier delivered and once delivered then he will update status as 'Delivered'

In above screen one courier delivered then employee will choose status as 'Delivered' and update its status and now click on 'View Courier Current Status' link to get below page



In above screen select courier ID and then click on 'Submit' button to get below page

In above screen for selected courier ID employee can see current location and once delivered then status will be shown as 'Delivered' and can click on 'View on Map' link to get below page



In above screen employee can see 'courier current location was RGI Airport'. Now logout and track same with user

In above screen user also can track courier just by entering ID and then press button to get below page

In above screen user can also see all possible location of his courier. Similarly by following above screens you can manage and run all courier system.

V. **Conclusion**

The **Courier Management System** developed using **Python Django** provides an efficient, user-friendly, and scalable solution to automate and streamline courier operations. The system successfully meets the core objectives of the project, which include efficient tracking of couriers, secure user authentication, dynamic status updates, feedback submission, and image upload capabilities.

Through this platform, administrators can manage users and gain real-time insights into all courier activities, while employees are equipped with tools to handle bookings, manage proof of delivery, and keep status updates accurate and current. Customers benefit from an intuitive interface where they can track their packages and provide feedback, enhancing overall satisfaction and transparency.

References

1. **Books and Journals**
   i. Sommerville, Ian. *Software Engineering* (10th Edition). Pearson Education, 2015.
   ii. Pressman, Roger S. *Software Engineering: A Practitioner's Approach* (8th Edition). McGraw-Hill Education, 2014.
   iii. Dennis, Alan, Barbara Haley Wixom, and Roberta M. Roth. *Systems Analysis and Design*. Wiley, 2015.
b. **Research Papers**
   i. Gupta, R., & Sharma, P. (2021). *Logistics Management System: A Review on Digital Transformation and Future Trends*. International Journal of Advanced Research in Computer Science, 12(3), 56-61.
   ii. Kumar, S., & Raj, R. (2020). *Real-Time Tracking and Route Optimization in Courier Systems using IoT and AI*. International Journal of Computer Applications, 177(41), 24-29.
c. **Web Resources**
   i. GeeksforGeeks – Tutorials and code references for backend and frontend development.
   ii. Stack Overflow – Community-based troubleshooting and discussions related to web development issues.
   iii. W3Schools – Reference for HTML, CSS, JavaScript, and PHP implementation.
   iv. GitHub – Code repositories and version control resources used for project collaboration.
d. **Technologies & Frameworks**
   i. PHP Official Documentation – https://www.php.net/docs.php
   ii. MySQL Reference Manual – https://dev.mysql.com/doc/
   iii. Bootstrap Framework – https://getbootstrap.com/docs/