

COURIER TRACKING SYSTEM



A PROJECT REPORT

Submitted by

THILIPKUMAR S (2303811724321117)

in partial fulfillment of requirements for the award of the course

CGB1221-DATABASE MANAGEMENT SYSTEMS

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE- 2025

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**COURIER TRACKING SYSTEM**” is the bonafide work of **THILIPKUMAR S (2303811724321117)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.T. AVUDAIAPPAN, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

SIGNATURE

Mrs.S. GEETHA, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on ...04.06.2025.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**COURIER TRACKING SYSTEM**” is the result of original work done by me and best of my knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS**.

Signature

THILIPKUMAR S

Place: Samayapuram

Date: 04.06.2025

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E.,Ph.D.**, Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mrs.S.GEETHA, M.E.**, Department of **ARTIFICIAL INTELLIGENCE**, for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

INSTITUTE

Vision:

- To serve the society by offering top-notch technical education on par with global standards.

Mission:

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all – round personalities respecting moral and ethical values.

DEPARTMENT

Vision:

- To excel in education, innovation, and research in Artificial Intelligence and Data Science to fulfil industrial demands and societal expectations.

Mission

- To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.
- To collaborate with industry and offer top-notch facilities in a conducive learning environment.
- To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.
- To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

- **PEO1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.
- **PEO2:** Provide industry-specific solutions for the society with effective communication and ethics.
- **PEO3** Enhance their professional skills through research and lifelong learning initiatives.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Capable of finding the important factors in large datasets, simplify the data, and improve predictive model accuracy.
- **PSO2:** Capable of analyzing and providing a solution to a given real-world problem by designing an effective program.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Courier Tracking System is a web-based application designed to streamline the process of managing courier services through real-time tracking and automated data handling. The system is built on fundamental DBMS concepts such as relational database design, normalization, and secure SQL operations. It features modules for customer management, shipment tracking, delivery updates, and admin controls. Users can track their shipments in real-time, while administrators manage customer records, assign tracking numbers, and monitor delivery progress. This project demonstrates the effective use of database management techniques in building a reliable and user-friendly courier service platform.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD DATABASES FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>The Courier Tracking System is a web-based application designed to streamline the process of managing courier services through real-time tracking and automated data handling. The system is built on fundamental DBMS concepts such as relational database design, normalization, and secure SQL operations. It features modules for customer management, shipment tracking, delivery updates, and admin controls. Users can track their shipments in real-time, while administrators manage customer records, assign tracking numbers, and monitor delivery progress. This project demonstrates the effective use of database management techniques in building a reliable and user-friendly courier service platform.</p>	<p>PO1 -2 PO2 -3 PO3 -3 PO5 -3 PO6-1 PO9 -2 PO10 -2 PO11-3</p>	<p>PSO1 -3 PSO2 -2</p>

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
	1.3 SQL AND DATABASE CONCEPTS	2
2	PROJECT METHODOLOGY	3
	2.1 PROPOSED WORK	3
	2.2 BLOCK DIAGRAM	4
3	MODULE DESCRIPTION	5
	3.1 COURIER BOOKING MODULE	5
	3.2 TRACKING MODULE	5
	3.3 ADMIN AUTHENTICATION & DASHBOARD MODULE	6
	3.4 COURIER STATUS UPDATE MODULE	6
	3.5 REPORTING & RECORD VIEWING MODULE	7
4	CONCLUSION & FUTURE SCOPE	8
	APPENDIX A SOURCE CODE	9
	APPENDIX B SCREENSHOTS	17
	REFERENCES	20

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The objective of this project is to build an efficient courier tracking system using a relational database to manage key operations such as booking, shipment tracking, and status updates. The system is designed to reduce manual errors, enhance transparency, and provide real-time tracking information to users. It offers role-based access for customers and administrators, allowing secure login, courier management, and data updates. By applying SQL and effective database design, the system ensures fast, accurate, and scalable handling of courier-related data.

1.2 OVERVIEW

This project delivers a functional courier tracking system centered around a well-structured relational database that supports core logistics operations such as booking, tracking, and status management. Customers can book couriers and track their shipments using tracking IDs, while administrators can manage courier details, update shipment status, and generate booking reports. The system's architecture is built on normalized database tables that represent key entities like users, shipments, locations, and tracking records, connected through primary and foreign key relationships. SQL is used for accurate and efficient data operations including retrieval, updates, and insertions. The backend ensures transactional integrity and data consistency, while the user interface interacts securely with the database to provide real-time tracking and management features.

1.3 SQL AND DATABASE CONCEPTS

Structured Query Language (SQL) is central to the development of the courier tracking system, enabling robust interaction with the underlying relational database. The project uses normalized tables to reduce data redundancy and ensure consistency, with primary and foreign key constraints maintaining clear relationships between entities such as customers, bookings, shipment statuses, and tracking records. SQL is used for core operations like SELECT, INSERT, UPDATE, and DELETE, as well as for filtering and joining data across multiple tables. Foreign keys link bookings to users and shipment updates, reflecting real-world logistics workflows. Indexes and constraints are applied to improve performance and maintain data integrity. This approach illustrates how foundational database concepts support a scalable, real-time courier management system.

CHAPTER 2

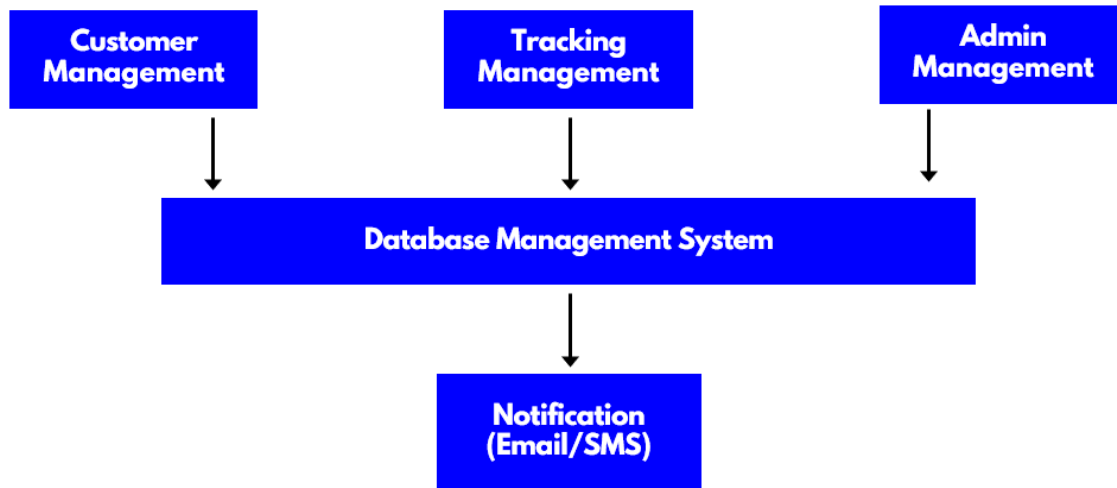
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The Courier Tracking System is developed to offer a centralized and automated platform for managing courier operations, with an emphasis on efficient database-driven functionality. The system is divided into modular components, including the user interface, business logic, authentication layer, and a relational database that manages all critical data such as user details, courier bookings, shipment statuses, and delivery routes. The database schema leverages relational design principles, using primary keys, foreign keys, indexes, and constraints to maintain data integrity and support optimized performance.

Core entities include users (customers and admins), courier records with detailed tracking information, and booking transactions. The system supports full CRUD operations, allowing real-time updates and retrievals from both frontend and backend interfaces. Secure authentication ensures role-based access, where customers can track shipments and admins can manage courier statuses and review booking logs. Through structured schema design and transaction handling, the system supports concurrent user actions while maintaining consistency and reliability. This project demonstrates the practical use of database management concepts such as normalization, indexing, referential integrity, and transactional control within a dynamic logistics environment.

2.2 BLOCK DIAGRAM



CHAPTER 3

MODULE DESCRIPTION

3.1 COURIER BOOKING MODULE

This module allows users to book new courier services by entering details such as sender name, source and destination cities, parcel weight, and delivery speed. It automatically calculates the price and estimated delivery time based on the weight and type of service selected. All booking data is timestamped and securely stored in the database for future tracking.

Key Functions:

- `createBooking()`: Records new booking details including user input and system-generated time.
- `calculatePrice()`: Computes shipping cost based on weight and delivery type.
- `storeBooking()`: Saves the entire booking info into the database securely.

3.2 TRACKING MODULE

This module enables customers to track their couriers using a unique tracking ID. Upon entering a valid ID, the system retrieves and displays key details such as sender info, route, current location, and estimated delivery days. It handles incorrect or missing tracking IDs gracefully and presents tracking data in a clear, user-friendly format.

Key Functions:

- `trackCourier()`: Retrieves courier details based on the tracking ID.
- `displayTrackingInfo()`: Shows sender, route, location, and ETA.
- `handleInvalidID()`: Detects and alerts users on invalid or missing IDs.
- `updateAvailability()`: Adjusts availability status after booking.

3.3 COURIER STATUS UPDATE MODULE

This module allows the admin to update the courier's current location and revise the estimated delivery days. Inputs are validated to ensure accuracy, and changes are written back to the database with confirmation messages. It helps keep tracking information current and reliable for users.

Key Functions:

- `updateLocation()`: Modifies the courier's present location.
- `updateETA()`: Adjusts the estimated delivery days.
- `validateAndSave()`: Ensures input correctness and updates the database.

3.4 USER AUTHENTICATION & ADMIN DASHBOARD MODULE

This module handles secure login functionality for the admin user. It verifies credentials against stored database records and restricts access to the dashboard unless valid login details are provided. Once authenticated, the admin gains access to courier records, status updates, and reports. This module acts as the central control point for administrative tasks

Key Functions:

- `loginAdmin()`: Authenticates the admin using stored username and password.
- `validateCredentials()`: Confirms login details and grants dashboard access.
- `loadDashboard()`: Loads shipment records and tools for quick access.

3.5 REPORTING & RECORD VIEWING MODULE

This module provides the admin with a complete list of all courier bookings, sorted by recent activity. It shows key information like tracking ID, sender details, route, location, and ETA. It also handles scenarios with no records and offers a snapshot view for monitoring operations and generating manual reports.

Key Functions:

- `viewAllBookings()`: Displays all courier booking records in a tabular format.
- `sortByRecent()`: Organizes records by most recent bookings.
- `handleEmptyData()`: Manages views when no data is available.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

CONCLUSION

The **Courier Booking Module** plays a crucial role in automating and streamlining logistics and parcel management operations. By integrating database management systems (DBMS), the module ensures efficient storage, retrieval, and management of user information, parcel details, booking status, and tracking records. A well-structured backend database using technologies like MySQL or SQLite enables seamless interaction between users, administrators, and delivery personnel. Through features such as real-time tracking, secure data handling, and automated booking ID generation, the system not only improves operational accuracy but also enhances customer satisfaction and transparency.

FUTURE SCOPE

1. **Integration with GPS and IoT Devices:** Real-time tracking of courier movement through GPS-enabled devices and Internet of Things (IoT) sensors for live updates.
2. **Mobile App Support:** Developing an Android/iOS mobile application for booking, tracking, and managing courier services on-the-go.
3. **AI-Based Route Optimization:** Implementing machine learning algorithms to optimize delivery routes based on traffic and distance.
4. **Cloud-Based DBMS:** Migrating to cloud platforms such as Firebase or AWS RDS for better scalability, data availability, and remote access.
5. **Advanced Security Features:** Incorporating user authentication via OTP/email verification and encryption protocols for safeguarding customer data.
6. **Multi-Language & Global Support:** Expanding system usability across regions by supporting multiple languages and international shipping logic.

APPENDIX A – SOURCE CODE

```
import os
import sqlite3
import random
import string
from datetime import datetime
import gradio as gr

DB = "courier_system.db"
TN_CITIES = [
    "Chennai", "Coimbatore", "Madurai", "Tiruchirappalli",
    "Salem", "Erode", "Tirunelveli", "Vellore",
    "Thoothukudi", "Dindigul"
]

# Setup DB (reset every time you run)
if os.path.exists(DB):
    os.remove(DB)

conn = sqlite3.connect(DB)
c = conn.cursor()
c.execute("""
CREATE TABLE courier(
    tracking_id TEXT PRIMARY KEY,
    user_name TEXT,
    from_loc TEXT,
    to_loc TEXT,
    weight_kg REAL,
    price_inr REAL,
```

```

    speed TEXT,
    booked_on TEXT,
    current_loc TEXT,
    exp_days INTEGER
)
"""
c.execute("""
CREATE TABLE admin(
    username TEXT PRIMARY KEY,
    password TEXT
)
""")
c.execute("INSERT INTO admin VALUES ('admin', 'admin123')")
conn.commit()
conn.close()

def gen_tid():
    conn = sqlite3.connect(DB)
    cur = conn.cursor()
    while True:
        tid = ".join(random.choices(string.ascii_uppercase + string.digits, k=8))
        cur.execute("SELECT 1 FROM courier WHERE tracking_id=?", (tid,))
        if not cur.fetchone():
            conn.close()
            return tid

def get_all_ids():
    conn = sqlite3.connect(DB)
    cur = conn.cursor()
    cur.execute("SELECT tracking_id FROM courier ORDER BY booked_on

```

DESC")

```
ids = [r[0] for r in cur.fetchall()]
```

```
conn.close()
```

```
return ids
```

```
def book_courier(name, frm, to, weight, speed):
```

```
    if not name.strip() or frm not in TN_CITIES or to not in TN_CITIES or weight  
<= 0:
```

```
        return "✗ Fill all fields correctly.", ""
```

```
    if frm == to:
```

```
        return "✗ 'From' and 'To' locations must be different.", ""
```

```
    base_price = weight * 15
```

```
    exp_days = 5
```

```
    if speed == "Fast Delivery":
```

```
        base_price *= 1.5
```

```
        exp_days = 2
```

```
    tid = gen_tid()
```

```
    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

```
    conn = sqlite3.connect(DB)
```

```
    cur = conn.cursor()
```

```
    cur.execute("""
```

```
        INSERT INTO courier(tracking_id, user_name, from_loc, to_loc, weight_kg,  
price_inr, speed, booked_on, current_loc, exp_days)
```

```
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
```

```
    """, (tid, name, frm, to, weight, base_price, speed, now, frm, exp_days))
```

```
    conn.commit()
```

```
    conn.close()
```

```
    return (
```

```
        f"✔ Booked! Tracking ID: {tid}\nPrice: ₹{base_price:.2f}\nEstimated  
delivery: {exp_days} days",
```

```
    tid
)
```

```
def track_courier(tid):
```

```
    tid = tid.strip()
```

```
    if not tid:
```

```
        return " ✕ Please enter Tracking ID."
```

```
    conn = sqlite3.connect(DB)
```

```
    cur = conn.cursor()
```

```
    cur.execute("""
```

```
        SELECT user_name, from_loc, to_loc, current_loc, exp_days
```

```
        FROM courier WHERE tracking_id=?
```

```
    """, (tid,))
```

```
    row = cur.fetchone()
```

```
    conn.close()
```

```
    if not row:
```

```
        return " ✕ Tracking ID not found."
```

```
    user, frm, to, cur_loc, days = row
```

```
    return (f"Tracking ID: {tid}\nUser: {user}\nFrom: {frm}\nTo: {to}\nCurrent  
location: {cur_loc}\nEstimated days left: {days}")
```

```
def admin_login(username, password):
```

```
    username = username.strip()
```

```
    password = password.strip()
```

```
    conn = sqlite3.connect(DB)
```

```
    cur = conn.cursor()
```

```
    cur.execute("SELECT 1 FROM admin WHERE username=? AND password=?",  
(username, password))
```

```
    result = cur.fetchone()
```

```
    conn.close()
```

```

if result:
    all_ids = get_all_ids()
    return (
        "✔ Login successful.",
        gr.update(choices=all_ids), # Update the dropdown with choices
        gr.update(visible=True)    # Show the admin panel
    )
else:
    return (
        "✘ Invalid username or password.",
        gr.update(choices=[]),
        gr.update(visible=False)
    )

def admin_update(tracking_id, new_location, new_days):
    if not tracking_id:
        return "✘ Select a Tracking ID."
    if new_location not in TN_CITIES:
        return "✘ Invalid location."
    if new_days < 1:
        return "✘ Expected days must be >= 1."
    conn = sqlite3.connect(DB)
    cur = conn.cursor()
    cur.execute("UPDATE courier SET current_loc=?, exp_days=? WHERE
tracking_id=?", (new_location, int(new_days), tracking_id))
    conn.commit()
    updated = cur.rowcount
    conn.close()
    if updated:

```

```

    return "✅ Update successful."
else:
    return "❌ Tracking ID not found."

```

```

def admin_view_all():
    conn = sqlite3.connect(DB)
    cur = conn.cursor()
    cur.execute("""
        SELECT tracking_id, user_name, from_loc, to_loc, current_loc, exp_days
        FROM courier ORDER BY booked_on DESC
    """)
    rows = cur.fetchall()
    conn.close()
    if not rows:
        return "No bookings found."
    header = "Tracking ID | User | From → To | Current Location | ETA (days)"
    lines = [header, "-" * len(header)]
    for tid, user, frm, to, cur_loc, days in rows:
        lines.append(f"{tid} | {user} | {frm} → {to} | {cur_loc} | {days}")
    return "\n".join(lines)

```

Gradio UI

with gr.Blocks() as app:

```

    gr.Markdown("# 📦 Courier Tracking System")

```

with gr.Tab("Book Courier"):

```

    name = gr.Textbox(label="Your Name")
    frm = gr.Dropdown(choices=TN_CITIES, label="From (Tamil Nadu only)")
    to = gr.Dropdown(choices=TN_CITIES, label="To (Tamil Nadu only)")
    weight = gr.Number(label="Weight (kg)", value=1, minimum=0.1)

```



```

speed = gr.Radio(["Normal Delivery", "Fast Delivery"], label="Delivery
Speed", value="Normal Delivery")

btn_book = gr.Button("Book Courier")

out_book = gr.Textbox(label="Booking Status", interactive=False, lines=3)

tid_out = gr.Textbox(label="Tracking ID", interactive=False)


btn_book.click(book_courier, inputs=[name, frm, to, weight, speed],
outputs=[out_book, tid_out])


with gr.Tab("Track Courier"):

    tid_in = gr.Textbox(label="Enter Tracking ID")

    btn_track = gr.Button("Track Courier")

    out_track = gr.Textbox(label="Tracking Information", interactive=False,
lines=6)


btn_track.click(track_courier, inputs=tid_in, outputs=out_track)


with gr.Tab("Admin"):

    admin_user = gr.Textbox(label="Username")

    admin_pass = gr.Textbox(label="Password", type="password")

    btn_login = gr.Button("Login")

    login_status = gr.Textbox(label="Login Status", interactive=False)


with gr.Column(visible=False) as admin_panel:

    select_tid = gr.Dropdown(label="Select Tracking ID", choices=[])

    new_loc = gr.Dropdown(choices=TN_CITIES, label="Update Current
Location")

    new_days = gr.Number(label="Update Expected Days", value=5,
minimum=1, precision=0)

    btn_update = gr.Button("Update")

```

```
update_status = gr.Textbox(label="Update Status", interactive=False)
btn_view_all = gr.Button("View All Bookings")
view_all_out = gr.Textbox(label="All Courier Bookings", interactive=False,
lines=10)
```

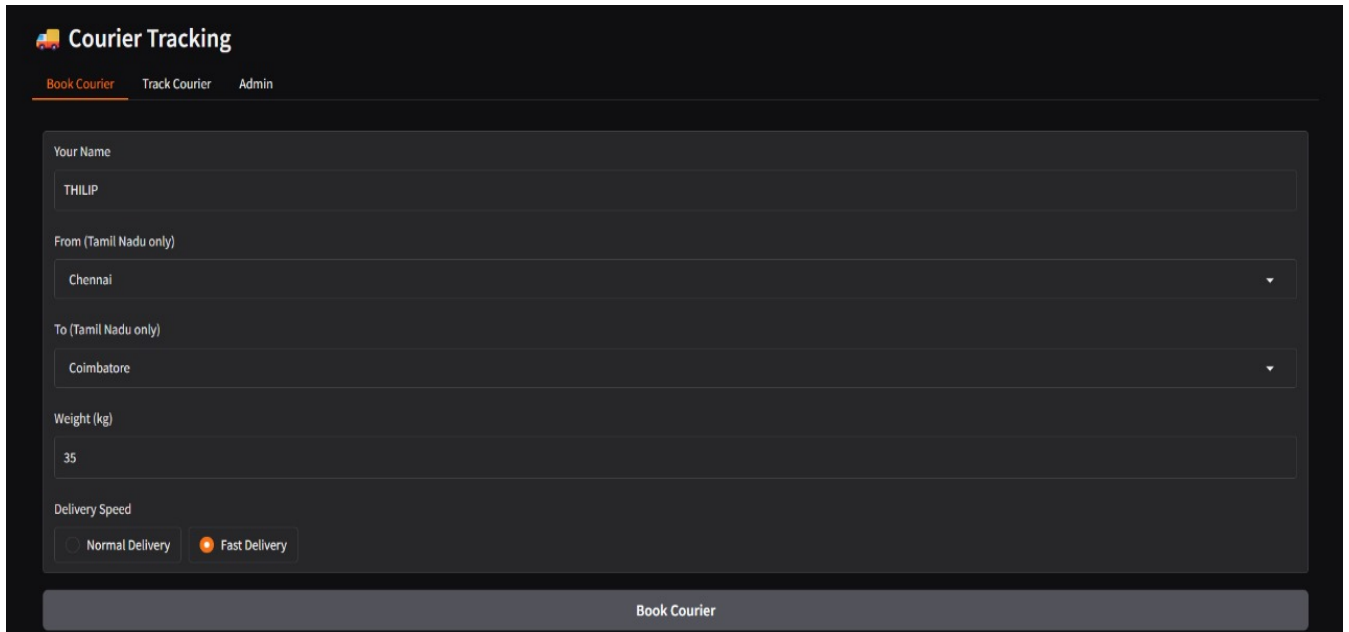
```
btn_login.click(
    admin_login,
    inputs=[admin_user, admin_pass],
    outputs=[login_status, select_tid, admin_panel]
)
```

```
btn_update.click(
    admin_update,
    inputs=[select_tid, new_loc, new_days],
    outputs=update_status
)
```

```
btn_view_all.click(
    admin_view_all,
    inputs=None,
    outputs=view_all_out
)
```

```
app.launch()
```

APPENDIX B – SCREENSHOTS



Courier Tracking

[Book Courier](#) [Track Courier](#) [Admin](#)

Your Name
THILIP

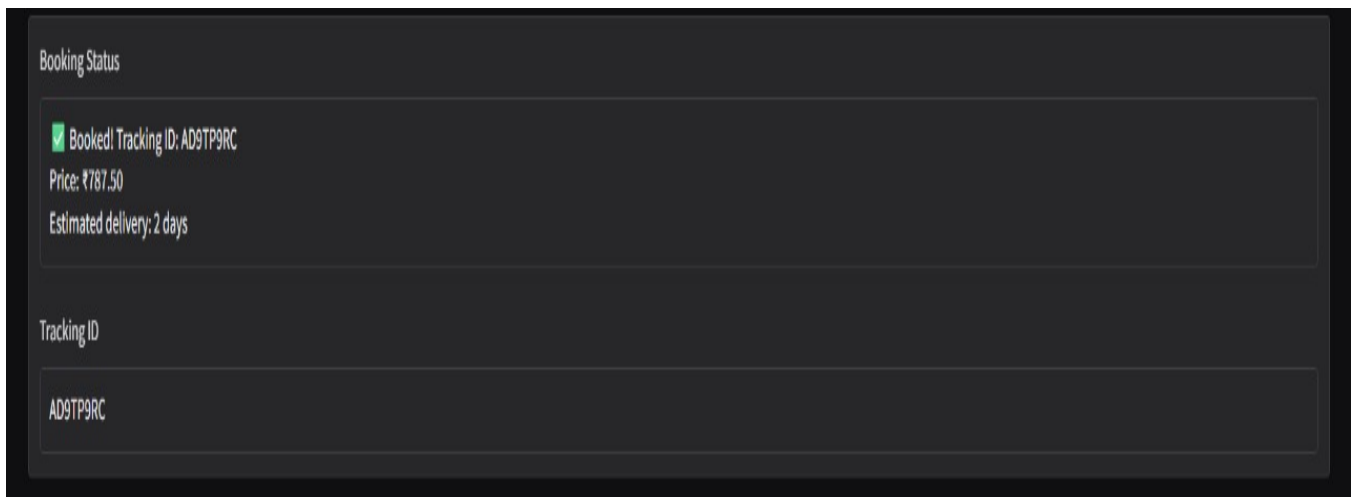
From (Tamil Nadu only)
Chennai

To (Tamil Nadu only)
Coimbatore

Weight (kg)
35

Delivery Speed
☐ Normal Delivery ☒ Fast Delivery

Book Courier




Booking Status

☒ Booked! Tracking ID: AD9TP9RC
Price: ₹787.50
Estimated delivery: 2 days

Tracking ID

AD9TP9RC

 **Courier Tracking**

[Book Courier](#) [Track Courier](#) [Admin](#)

Enter Tracking ID

AD9TP9RC

Track Courier

Tracking Information

Tracking ID: AD9TP9RC

User: THILIP

From: Chennai

To: Coimbatore

Current location: Chennai

Estimated days left: 2

[Book Courier](#) [Track Courier](#) [Admin](#)

Username

admin

Password

Login

Login Status

☒ Login successful.

Select Tracking ID

O4CYJ4UZ

Update Current Location

Coimbatore

Update Expected Days

1

Update

18

Select Tracking ID

04CYJ4UZ

Update Current Location

Coimbatore

Update Expected Days

1

Update

Update Status

☒ Update successful.

View All Bookings

All Courier Bookings

Tracking ID | User | From → To | Current Location | ETA (days)

04CYJ4UZ | tikjh | Chennai → Coimbatore | Coimbatore | 1

REFERENCES

1. Mozilla Developer Network: HTML Reference. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTML>
2. SQLite Consortium: SQLite Official Documentation. Retrieved from <https://www.sqlite.org/docs.html>
3. Stack Overflow Community: SQL Queries and Database Design Discussions. Retrieved from <https://stackoverflow.com/>
4. W3Schools: SQL Tutorial. Retrieved from <https://www.w3schools.com/sql/>
5. GeeksforGeeks: Courier Management System Project using DBMS. Retrieved from <https://www.geeksforgeeks.org/courier-management-system-project/>