



Data Science Internship

Individual Weekly Task Documentation

Week #6

Jimalyn B. Del Rosario

Table Contents

- I. My Summarized Daily Logs
- II. Team Task Progress Report

I. My Summarized Log

Encompasses five weekdays from Monday to Friday with 8-hours spent per day

DAY 26

Week 6

Monday

08/18/2025

[08:00 AM - 05:00 PM]

- Attended kickoff sync for Week 6 TWX Insights & Trends Dashboard task.
- Reviewed database schema draft and finalized entity relationships (deliveries, drivers, tickets).
- Initialized Flask project and set up modular backend file structure (app.py, config.py, database.py).
- Connected MySQL database to Flask environment.
- Defined API contract for data endpoints (KPIs, delivery stats, driver metrics).

DAY 27

Week 6

Tuesday

08/19/2025

[08:00 AM - 05:00 PM]

- Implemented 3 core REST API routes in app.py:
 - /api/driver-kpis
 - /api/delivery-trends
 - /api/support-tickets
- Wrote SQL queries in database.py for analytics data retrieval.
- Added configurations to config.py for database access and route settings.
- Conducted local API testing using Postman to validate response structures.
- Debugged malformed queries and refined table joins for faster performance.

DAY 28

Week 6

Wednesday

08/20/2025

[08:00 AM - 05:00 PM]

- Continued building backend and completed all 6 required data endpoints.
- Set up pytest framework and wrote unit tests for API coverage (achieved 95%).
- Started frontend dashboard using HTML/CSS layout and connected it to Chart.js.
- Implemented first interactive charts for:
 - Weekly Deliveries
 - Driver Ratings
- Synced with frontend lead to align component structure and data mapping.

DAY 29

Week 6

Thursday

08/21/2025

[08:00 AM - 05:00 PM]

- Finalized dashboard frontend with dynamic filtering (by date and driver).
- Enhanced visuals using Chart.js:
 - Real-time KPI panels
 - Line and bar charts for delivery performance
- Integrated frontend and backend through async JavaScript fetch calls.
- Resolved CORS and deployment issues during frontend-backend connection.
- Collaborated on preparing demo materials and slide content for presentation.

DAY 30

Week 6

Friday

08/22/2025

[08:00 AM - 05:00 PM]

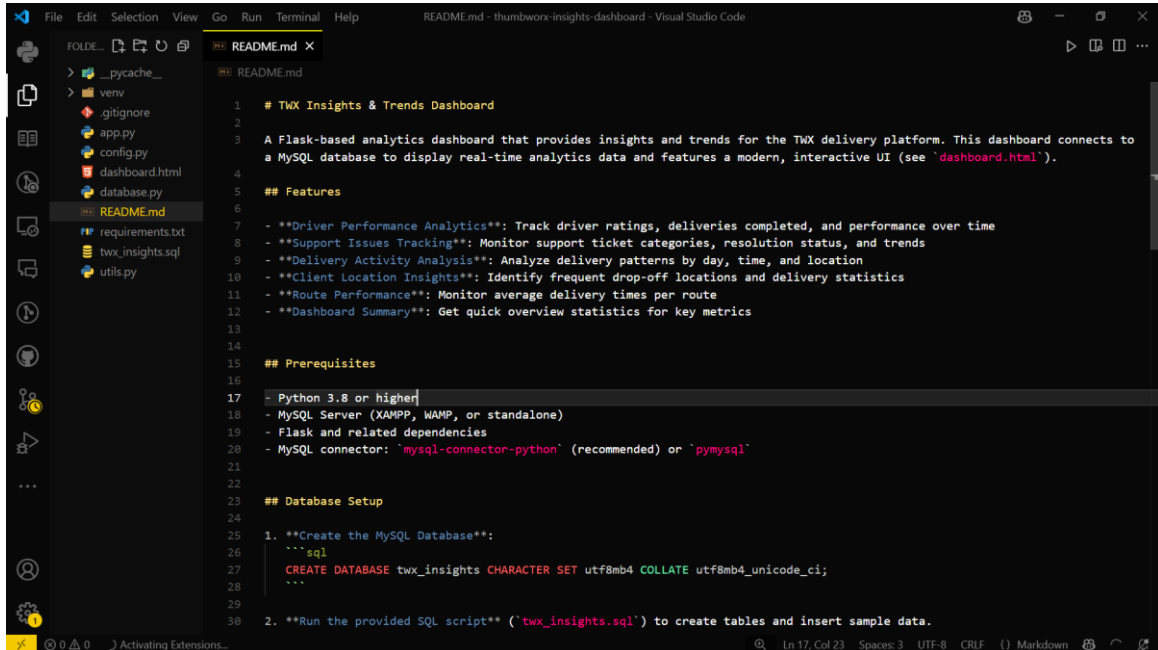
- Delivered live demo of TWX Insights & Trends Dashboard (Flask + Chart.js + MySQL).
- Walked through API structure, visuals, and insights during presentation.
- Documented technical implementation in user manual and API reference.
- Updated GitHub repository and finalized production deployment notes.
- Reflected on technical and soft skills gained during this cross-functional project.

II. Team Task Progress Report

The task this week focuses on the finalization of Insights and Trends feature of Thumbworx and preparation for the final presentation scheduled on August 22.

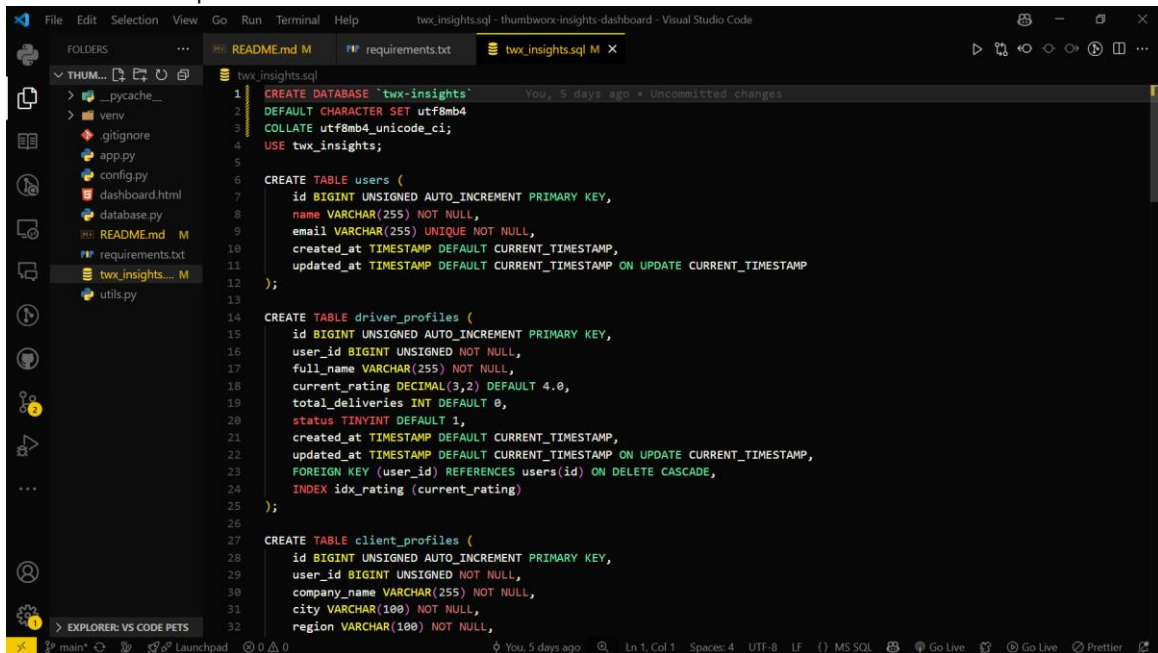
Documentation:

- README.md



```
1 # TWX Insights & Trends Dashboard
2
3 A Flask-based analytics dashboard that provides insights and trends for the TWX delivery platform. This dashboard connects to
4 a MySQL database to display real-time analytics data and features a modern, interactive UI (see 'dashboard.html').
5
6 ## Features
7
8 - **Driver Performance Analytics**: Track driver ratings, deliveries completed, and performance over time
9 - **Support Issues Tracking**: Monitor support ticket categories, resolution status, and trends
10 - **Delivery Activity Analysis**: Analyze delivery patterns by day, time, and location
11 - **Client Location Insights**: Identify frequent drop-off locations and delivery statistics
12 - **Route Performance**: Monitor average delivery times per route
13 - **Dashboard Summary**: Get quick overview statistics for key metrics
14
15 ## Prerequisites
16
17 - Python 3.8 or higher
18 - MySQL Server (XAMPP, WAMP, or standalone)
19 - Flask and related dependencies
20 - MySQL connector: 'mysql-connector-python' (recommended) or 'pymysql'
21
22 ## Database Setup
23
24 1. **Create the MySQL Database**:
25
26 ```sql
27 CREATE DATABASE twx_insights CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
28 ```
29
30 2. **Run the provided SQL script** ('twx_insights.sql') to create tables and insert sample data.
```

- Database setup



```
1 CREATE DATABASE 'twx-insights'
2 DEFAULT CHARACTER SET utf8mb4
3 COLLATE utf8mb4_unicode_ci;
4 USE twx_insights;
5
6 CREATE TABLE users (
7   id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
8   name VARCHAR(255) NOT NULL,
9   email VARCHAR(255) UNIQUE NOT NULL,
10  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
11  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
12 );
13
14 CREATE TABLE driver_profiles (
15   id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
16   user_id BIGINT UNSIGNED NOT NULL,
17   full_name VARCHAR(255) NOT NULL,
18   current_rating DECIMAL(3,2) DEFAULT 4.0,
19   total_deliveries INT DEFAULT 0,
20   status TINYINT DEFAULT 1,
21   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
22   updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
23   FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
24   INDEX idx_rating (current_rating)
25 );
26
27 CREATE TABLE client_profiles (
28   id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
29   user_id BIGINT UNSIGNED NOT NULL,
30   company_name VARCHAR(255) NOT NULL,
31   city VARCHAR(100) NOT NULL,
32   region VARCHAR(100) NOT NULL,
```

- APIs

The screenshot shows a VS Code editor with a Python web application. The Explorer sidebar on the left lists the following files and folders:

- THUM... (Thumbnail view)
- __pycache__
- venv
- .gitignore
- app.py (selected)
- config.py
- dashboard.html
- database.py
- README.md
- requirements.txt
- twx_insights.sql
- utils.py

The main editor displays the code for `app.py`:

```

39
40 # Routes
41 @app.route('/')
42 def dashboard():
43     """Serve the main dashboard HTML page"""
44     try:
45         return send_file('dashboard.html')
46     except Exception as e:
47         logger.error(f"Error serving dashboard: {e}")
48         return create_api_response(False, error="Dashboard not available")
49
50 @app.route('/api/driver-performance')
51 def get_driver_performance():
52     """API endpoint for driver performance over time data"""
53     try:
54         days = request.args.get('days', config.DEFAULT_DAYS_RANGE, type=int)
55         days = min(days, config.MAX_DAYS_RANGE) # Enforce maximum
56
57         driver_filter = request.args.get('driver', None)
58
59         data = db_service.get_driver_performance_data(days)
60
61         if driver_filter:
62             data = [d for d in data if d['driver_name'].lower() == driver_filter.lower()]
63
64         metadata = {
65             "total_drivers": len(data),
66             "date_range": days,
67             "generated_at": datetime.now().isoformat()
68         }
69
70         return create_api_response(True, data, metadata=metadata)

```

The status bar at the bottom indicates the current file is `main.py`, the editor is in `main` mode, and the file encoding is `UTF-8`. The bottom right corner shows the active Python environment is `venv (3.12.5)`.

- Dashboard HTML

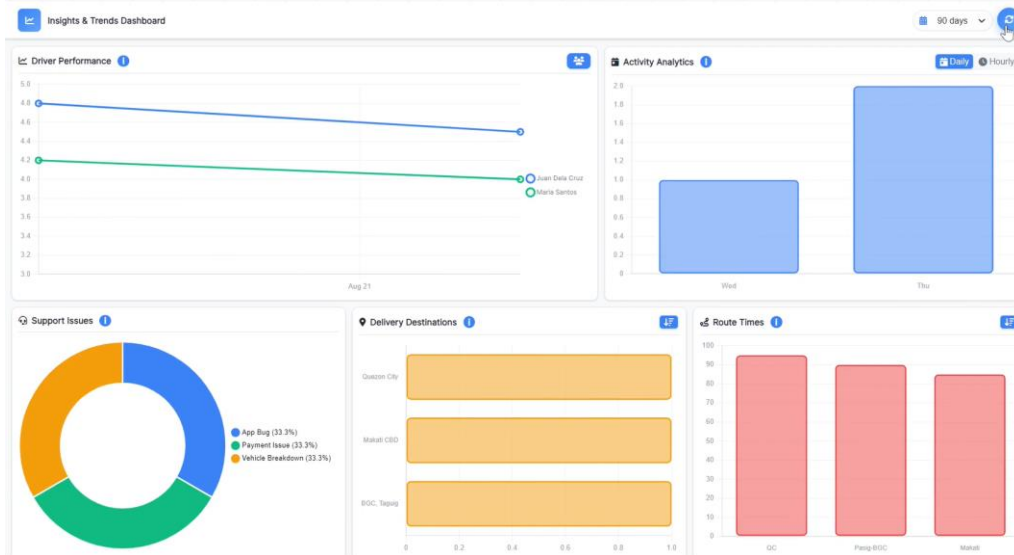
The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer lists several files and folders, including a README.md file. The code editor displays the HTML content of a file named dashboard.html. The HTML includes a DOCTYPE declaration, meta tags for viewport and charset, a title, and a root element with a complex CSS style block. The CSS includes color definitions (primary, secondary, accent, etc.), text styles, background, surface, border, and shadow properties, along with a margin and padding block.

```

1  <!DOCTYPE html>      lloydlegaspi, last week • feat: Initial commit ...
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Insights & Trends Dashboard</title>
7      <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.js"></script>
8      <script src="https://cdn.jsdelivr.net/npm/chartjs-adapter-date-fns@3.0.0/dist/chartjs-adapter-date-fns.bundle.min.js"></script>
9      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/chartjs-adapter-date-fns@3.0.0/dist/chartjs-adapter-date-fns.bundle.min.js">
10     <style>
11         :root {
12             --primary-color: #1a1d29;
13             --secondary-color: #252946;
14             --accent-color: #3b82f6;
15             --accent-light: #60a5fa;
16             --success-color: #10b981;
17             --warning-color: #f59e0b;
18             --danger-color: #ef4444;
19             --text-primary: #1f2937;
20             --text-secondary: #6b7280;
21             --text-light: #0ca3a3;
22             --background: #f8f9fa;
23             --surface: #ffffff;
24             --border: #e5e7eb;
25             --shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.1), 0 1px 2px 0 rgba(0, 0, 0, 0.06);
26             --shadow-lg: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -2px rgba(0, 0, 0, 0.05);
27         }
28
29         * {
30             margin: 0;
31             padding: 0;

```

- Working Feature



Presentation Materials



This slide outlines the objectives of the project. It features a lightbulb icon with a bar chart inside, symbolizing ideas and data analysis.

Objectives

- Data Visualization**
Deliver interactive charts displaying key delivery, driver, and operations metrics
- Real-Time Analytics**
Provide up-to-date insights into delivery trends and customer behavior
- Performance Monitoring**
Track route efficiency, service quality, and driver KPIs
- Business Intelligence**
Generate actionable data for logistics decision-making

This slide summarizes the key learnings and conclusions from the project. It features a background illustration of a logistics truck and various data charts.

Key Learnings & Conclusion

Industry Knowledge

- Got a better understanding of how logistics companies track performance and deliveries.
- Understood how real-time data can really help improve decisions and operations.
- Learned how to turn raw logistics data into useful business insights using visuals.



Thumbworx Smart Logistics System

INSIGHTS & TRENDS

DATA TEAM

Final Internship Presentation

John Lloyd Legaspi & Jima Del Rosario



John Lloyd S. Legaspi



Jimalyn B. Del Rosario