LAMINA
S T U D I O S

Data Science Internship

Individual Weekly Task Documentation

Week #2:
Foundations, Environment Setup,
Data Analytics & AI Microservice with Flask

Jimalyn B. Del Rosario

Table Contents

I.   My Summarized Log
Encompasses five weekdays from Monday to Friday with 8-hours spent per day

DAY 5
Week 2
Monday
07/21/2025

[08:00 AM - 11:00 AM]
- Read and studied the given task for Week 2 and created a plan for accomplishing all the tasks throughout the week
- Set up Notion for tracking and logging each accomplished task
- I watched the provided lecture videos about Big Data Analytics and Smart Supply Chain Technology

[11:00 AM - 12:00 PM]
- Break Time

[12:00 PM - 03:00 PM]
- Researched youtube video tutorials to be familiar with the following architectures and how to install and set up on my laptop
  - Laravel as backend
  - Next.js frontend
  - Flask for AI microservices
  - PostgreSQL as your database
  - Redis for job queueing
  - Metabase for analytics
  - Postman for testing

[12:00 PM - 03:00 PM]
- Continued with watching YouTube video tutorials

[03:00 PM - 05:00 PM]
- Started working on Individual Hands-on Tasks, that includes the installation, setting up and integration of each component of the architecture:
  1. Installation of PostgreSQL
  2. Creation of database
  3. Update of .env in Laravel
  4. Installation of PHP
  5. Installation of Laravel and creation of project
  6. Installation of Laravel Sanctum and Redis
  7. Creation of model and controller
  8. Migration
  9. API routing

DAY 6
Week 2
Tuesday
07/22/2025

[08:00 AM - 11:00 AM]
- Continued the Individual Hands-on Task:
    1. Installation of Flask dependencies
    2. Creation of app.py
    3. Installation of Postman
    4. Creation of a POST request
    5. Creation of a Job through Laravel Queueing with Redis
    6. Dispatch of a job from controller
    7. Running Laravel queue worker
    8. Setting up Next.js for frontend
    9. Installation of Axios and SWR

[11:00 AM - 12:00 PM]
- Break Time

[12:00 PM – 03:30 PM]
- Testing whether the entire architecture so far are well-integrated or has working connections.
- Worked through errors found in the architecture's connections

[03:30 PM – 05:00 PM]
- Continued with the architecture
    1. Download Metabase Analytics Dashboard
    2. Connection to PostgreSQL
    3. Embedding into Next.js via iframe

DAY 7
Week 2
Wednesday
07/23/2025

[08:00 AM - 10:00 AM]
- Conducted a self-review study on EDA processes and tools with Python Pandas, NumPy, Matplotlib

[10:00 AM - 11:00 AM]
- Searched for datasets online for EDA task

[11:00 AM - 12:00 PM]
- Break Time

[12:00 PM – 04:30 PM]
- Created a Jupyter Notebook for the EDA task
- Worked on the Data Cleaning and EDA of the Delhivery Dataset found in Kaggle
- Created simple analysis and visualizations
    - Route Type Distribution
    - Average segment_actual_time vs. segment_osrm_time
    - Actual vs Predicted Delivery Time
    - Top 10 Sources and Top 10 Destinations
    - Trip Created by Months

[04:30 PM – 05:00 PM]
- Created document format for Individual Weekly Task Documentation

DAY 8
Week 2
Thursday
07/24/2025

[08:00 AM - 10:00 AM]
- Selected a new dataset from the one recently provided by Ms. April
- Created a new Jupyter Notebook for Data Cleaning and EDA of new dataset

[10:00 AM - 11:00 PM]
- Conducted Data Cleaning on the Driver Behavior dataset

[11:00 PM – 12:00 PM]
- Break Time

[12:00 PM – 02:30 PM]
- Worked on EDA of the Driver Behavior Dataset provided by Ms. April

[02:30 PM – 05:30 PM]
- Conducted analysis and visualizations:
  - Distribution of Key Numeric Variables (Speed, Acceleration, RPM)
  - Distribution of Key Categorical Variables (Weather, Road Type, Traffic Condition)
  - Driver Comparison based on Average Speed, Fuel Usage, and Lane Deviation)
  - Trip Frequency per Hour
  - Trip Duration by Traffic Condition
  - Fuel Consumption by Road Type
  - Brake usage by Weather Condition
  - Anomaly Frequency

DAY 9
Week 2
Friday
07/25/2025

[08:00 AM - 09:00 AM]
- Started on the Individual Weekly Task Documentation

[09:00 AM - 10:00 PM]
- Early break

[10:00 PM – 02:00 PM]
- Collected all the screensnaps showing how I worked on the Individual Hands-on task and created a report for how I conducted each step.

[02:00 PM – 02:30 PM]
- Afternoon break

[02:30 PM – 05:00 PM]
- Continued working on the Individual Weekly Task Documentation
- Worked on creating a report for the Data Cleaning and EDA Task
- Collected all the links to lecture videos I watched
- Collected all the links to the deliverables
- Organized all the deliverable files
- Submit all the deliverables to the drive folder.

II.   Lecture Report and Team Progress Report

Lectures Watched
- Big Data Analytics for Smart Logistics,
- What Data Do Smart Supply Chain Technology Systems Use?
- Laravel in 100 Seconds
- Next.js in 100 Seconds // Plus Full Beginner's Tutorial
- A brilliant introduction to Flask
- Build a Rest API using Flask
- PostgreSQL in 100 Seconds
- What is Metabase?
- Installing Metabase
- Postman API Testing Tutorial
- Redis in 100 Seconds

Watching these videos helped me understand every part of the system I'm building. The logistics videos made it clear to me what kind of data I need to collect and why it matters for things like delivery time predictions. The Laravel, Next.js, Flask, and PostgreSQL videos showed me how to build and connect each part of my app from the backend to the frontend to AI services. Metabase helped me understand the importance of seeing the data clearly through dashboards, and Postman makes it easier to test my APIs. Although brief, these videos give me the knowledge I need to build the architecture

Team Progress

- The team conducted a team meeting prior to the meeting with the other team.
- Sections and sub teams were formed to distribute the tasks.

Sections and Subteams

Task	Assigned
Admin View	James, Nathaniel
Insights	Jima, Lloyd
Logistics Provider View	Hans, Daniel, Kief
Driver Performance Analysis	Glay, Evan, Kharl
Predictive Route Estimation	Syke, Paul, Kurt
Smart Logistics Recommendation	Keith, Catherine, Angela

- Tasks and Deliverables were established:
  1. Analytics Module (Thumbworx 2.0)
      - Phase 1 (Operational Analytics)
          - Admin View
          - Logistic Provider View
      - Phase 2 (Insights and Trends)
          - Driver Performance
          - Support Issues
          - Active Delivery Days/Times
          - Client Drop Off Locatiion Frequency
          - Average Delivery Time Per Route
  2. AI/ML Upload
      - Predictive Route Estimation
      - Driver Performance Analysis
      - Smart Logistics Recommendations

III.   Individual Hands-on Activity

- Architecture
- Laravel as backend
- Next.js frontend
- Flask for AI microservices
- PostgreSQL as your database
- Redis for job queueing
- Metabase for analytics
- Postman for testing
- A sample Machine Learning model for delivery time prediction

PART 1: Setup PostgreSQL

- Install PostgreSQL
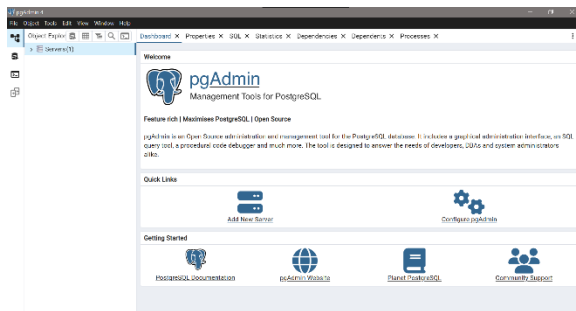
- Create database
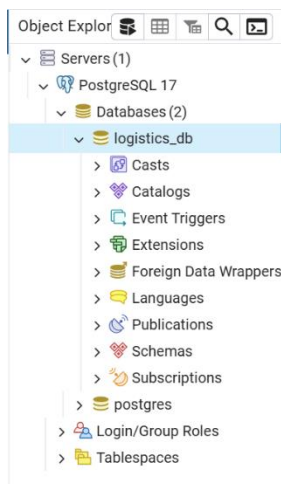
- Update .env in Laravel
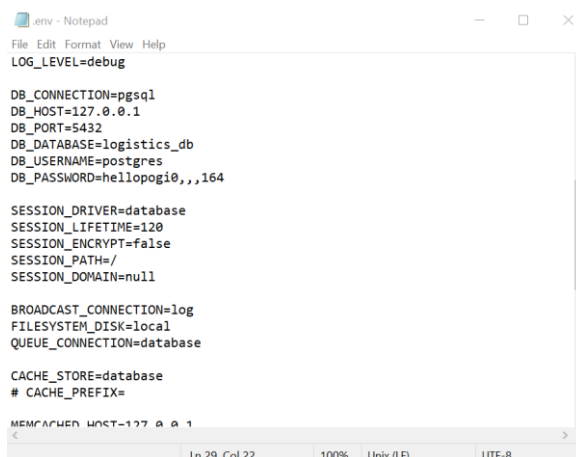
PART 2: Laravel Backend Setup

- Install PHP

- Install Composer

- Install Laravel and Create Laravel Project

- Install Laravel Sanctum for API auth and Redis for queues

- Create a model & controller for delivery

- Migration

- Run migration

- DeliveryController

- API route

PART 3: Flask Microservice with ML Model

- Install dependencies

- app.py

PART 4: Test flask API with Postman

- POST Request

PART 5: Laravel Queueing with Redis

- Create a Job

- Dispatch job from controller


- Run Laravel queue worker

PART 6: Next.js Frontend Integration

- Install Axios & SWR

- Check Integration

PART 7: Metabase Analytics Dashboard

- Download Metabase: https://www.metabase.com/start/oss/
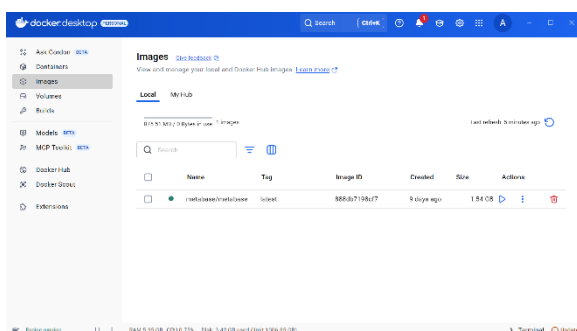
- Connect to PostgreSQL:
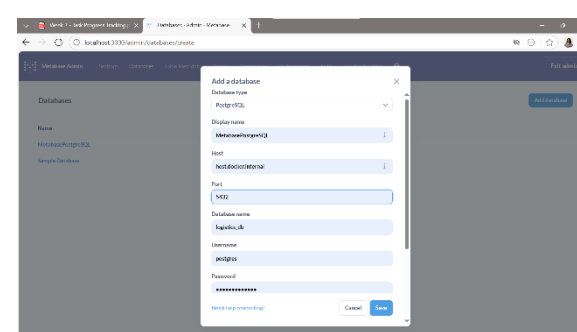
Observations and Errors Encountered:

1. The frontend would run on npm run dev but webpage shows error
2. laravel works on php artisan serve
3. flask would run on python app.py but webpage shows error
4. testing POST for localhost/predict works on POSTMAN
5. testing POST for localhost/predict-delivery does not work despite controllers, app.py, api routing are propoerly configured

Because of this, I still have yet to build the dashboards and embed the dashboards into Next.js via iframe.

IV.   Data Cleaning and EDA on Delhivery Dataset

I started on this activity earlier assuming that the search for a dataset is our responsibility. Later on during the week, we were provided with links of datasets to select from. I have two outputs for the activity because I wanted to show the process on the Delhivery Dataset too as I believe it has relevance to the study we are conducting, and I learned a lot of things from it too.

The Data Cleaning and EDA process of both datasets were conducted in Google Colab. I used the same group of python libraries for both analyses. Pandas for handling dataframes and transforming the dataset. NumPy for statistical functions. And, MatplotLib and Seaborn or the different visualizations.

First dataset:
https://colab.research.google.com/drive/1HvvmXJMhKtaG0MIQILwDMWFYz5MCVVqq?usp=drive_link

The basic data cleaning processes I conducted are:

- Dropping null rows
- Dropping unnecessary and redundant columns
- Converting to appropriate data types

The basic data analysis and visualization tasks conducted are:

- Statistical description of the numerical columns (count, mean, ave, min, max, etc)
- Route Type Distribution
- Average segment_actual_time vs. segment_osrm_time
- Actual vs Predicted Delivery Time
- Top 10 Sources and Top 10 Destinations
- Trip Created by Months
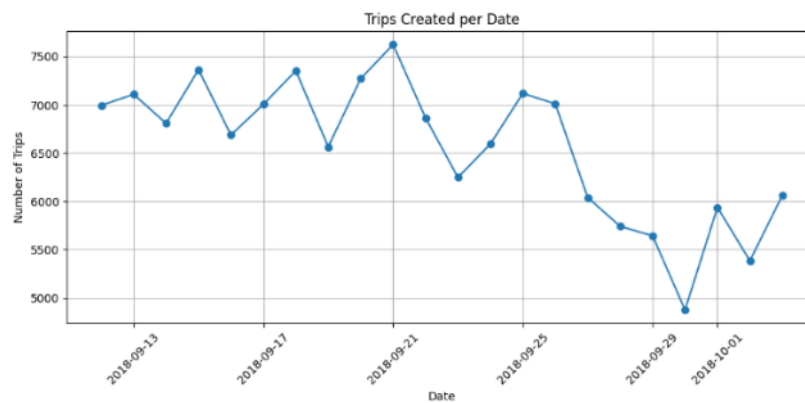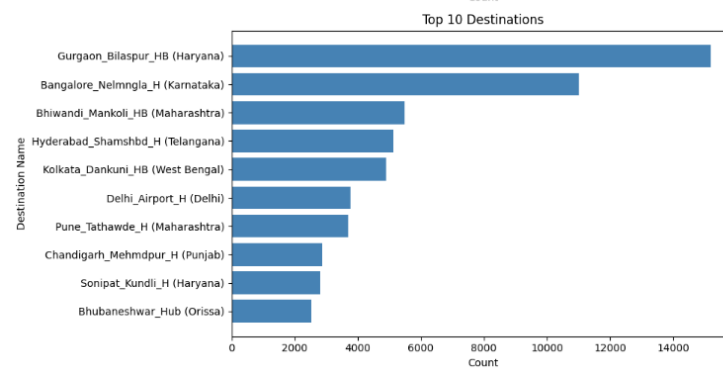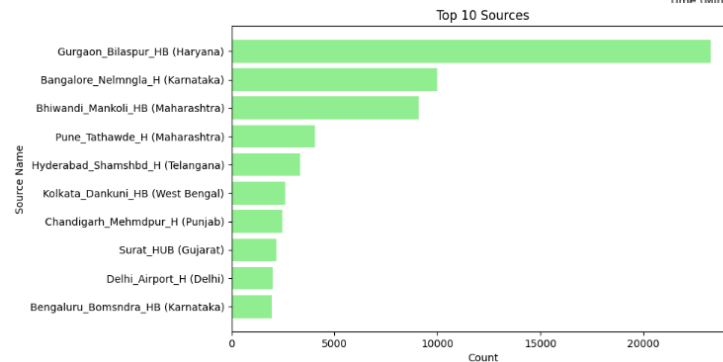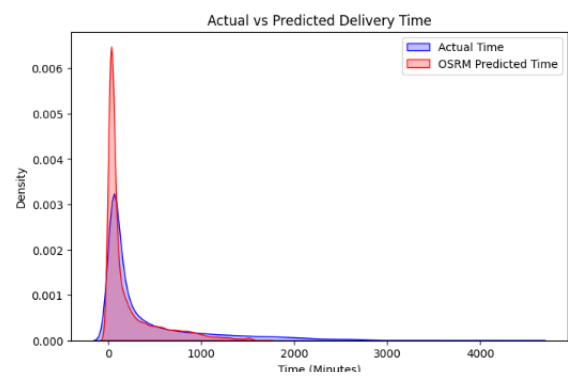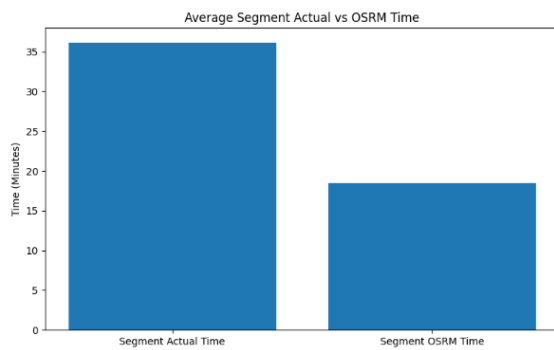
V.   Data Cleaning and EDA on Driver Behavior Dataset

Second dataset:

https://colab.research.google.com/drive/1nJlVVogWuLsqsccGeQt24irP09ifbaYn?usp=drive_link

The dataset is already in good formatting. The only data cleaning step it required was the conversion of columns to appropriate data types for analysis.

Data analysis and visualizations:

- Distribution of Key Numeric Variables (Speed, Acceleration, RPM)
- Distribution of Key Categorical Variables (Weather, Road Type, Traffic Condition)
- Driver Comparison based on Average Speed, Fuel Usage, and Lane Deviation)
- Trip Frequency per Hour
- Trip Duration by Traffic Condition
- Fuel Consumption by Road Type
- Brake usage by Weather Condition
- Anomaly Frequency

Fuel Consumption by Road Type

Anomaly Frequency

```
df[['anomalous_event', 'route_anomaly', 'geofencing_violation']].astype(int).sum()
```

	0
anomalous_event	12020
route_anomaly	11965
geofencing_violation	6029

dtype: int64