

Project Planning Document

Real-Time IoT Data Pipeline for Cairo Traffic Monitoring

Digital Egypt Pioneers Initiative

Team Leader: Ali Elgatool | Duration: Sept 15 - Nov 28, 2025

1. Project Overview

What We're Building

A cloud-based traffic monitoring system that simulates real-time data from 7 Cairo locations, detects traffic problems automatically, and displays everything on an interactive dashboard with smart alerts.

Why It Matters

This project helps us learn real-world data engineering skills while building something practical—a system that could actually help manage Cairo's busy traffic by identifying congestion, speeding, and accidents in real-time.

Project Details

- Duration:** 75 days (September 15 - November 28, 2025)
 - Budget:** \$100 Azure Student Credit
 - Locations:** 7 Cairo traffic hotspots
 - Data Frequency:** Every 5 seconds
 - Final Output:** Live dashboard + automated alerts
-

2. Our Team

Name	What They Do	Why They're Perfect For It
Ali Elgatool (Leader)	Leads team & builds real-time analytics	Strong technical background, great organizer
Mohamed Elhaysam	Creates traffic simulator	Excellent Python skills, creative problem solver
Asmaa Nasr	Sets up data streaming	Detail-oriented, great with Azure services
Ibrahim Malki	Connects everything together	Big-picture thinker, ensures quality
Esraa Ahmed	Builds the dashboard	Design-minded, user experience focused
Kareem Mostafa	Develops alert system	Analytical, understands business logic

3. The Locations We're Monitoring

We chose 7 of Cairo's busiest and most important traffic points:

1. **Tahrir Square** - The heart of Cairo
2. **Ramses Square** - Major transport hub
3. **6th October Bridge** - Key highway route
4. **Nasr City** - Eastern business district
5. **Heliopolis** - Airport area
6. **Maadi Corniche** - Nile-side southern route
7. **Ahmed Orabi Square** - Western gateway

These locations represent different types of traffic patterns and give us a comprehensive view of Cairo's traffic flow.

4. What Problems We're Solving

Traffic Issues We Detect

1. Congestion

- When average speed drops below 20 mph for 5 minutes
- Helps identify traffic jams early
- Alerts authorities to manage flow

2. Speeding Violations

- When vehicles exceed 80 mph
- Safety concern that needs immediate attention
- Helps with traffic law enforcement

3. Accidents

- When speed suddenly drops by more than 35 mph in 60 seconds
 - Critical for emergency response
 - Enables quick assistance dispatch
-

5. Our Timeline

Milestone 1: Building the Foundation (Sept 15 - Oct 15)

What we accomplished:

- Created Python simulator that generates realistic Cairo traffic data
- Built in rush hour patterns, weather effects, and random incidents
- Set up Azure Event Hubs to receive streaming data
- Made sure data flows smoothly every 5 seconds

Key learning: Understanding how to simulate realistic data and work with cloud streaming services.

Milestone 2: Making Sense of the Data (Oct 15 - Nov 2)

What we're doing now:

- Building smart queries that automatically detect traffic problems
- Setting up databases to store all the information
- Making sure everything processes quickly (under 1 second)
- Testing accuracy to ensure we catch real problems

The challenge: Creating queries that are both fast and accurate.

Milestone 3: Bringing It All Together (Nov 3 - Nov 28)

What's coming next:

- Building a beautiful, easy-to-use dashboard
- Creating automatic alerts when problems occur
- Testing everything thoroughly
- Writing our final report and preparing the presentation

The goal: A polished system that works flawlessly and impresses everyone.

6. How We Work Together

Daily Check-ins (6:00 PM on WhatsApp)

Everyone shares quickly:

- What did I finish today?
- What am I working on tomorrow?
- Do I need help with anything?

Why it works: Keeps everyone aligned without taking too much time.

Weekly Team Meetings (Saturdays, 8:00 PM)

Longer session where we:

- Demo what we've built
- Solve problems together
- Plan the next week
- Celebrate our progress

Why it works: Face-to-face time builds team spirit and solves bigger issues.

Our Tools

- **WhatsApp:** Quick daily communication
 - **Google Meet:** Weekly video meetings
 - **Google Drive:** Sharing documents
 - **GitHub:** Managing our code
-

7. Our Technology Stack

What We're Using

Data Generation

- Python 3.9+ for creating realistic traffic data
- Libraries: Random, JSON, DateTime for data simulation

Cloud Services (All in Azure)

- **Event Hubs:** Receives streaming data (\$12/month)
- **Stream Analytics:** Detects problems in real-time (\$30/month)
- **SQL Database:** Stores processed data (\$5/month)
- **Data Lake:** Backs up raw data (\$2/month)

Visualization & Alerts

- **Streamlit:** Interactive dashboard (free)
- **Python Scripts:** Automated alert system (free)

Total Cost: ~\$50 (well within our \$100 budget)

8. How We Measure Success

Data Quality

- Every piece of data matches our schema perfectly
- Traffic patterns look realistic (at least 90%)
- Zero errors in data generation

System Performance

- Process each event in less than 0.05 seconds
- Dashboard loads in under 3 seconds
- Alerts arrive within 10 seconds of problems

Accuracy

- Catch 95% or more of actual traffic problems
- Very few false alarms (less than 5%)
- Reliable detection across all scenarios

Team Delivery

- Meet all deadlines
- Complete documentation
- Clear, professional presentation

9. Challenges We're Managing

Budget Concerns

The worry: Running out of Azure credit

Our solution: Monitor spending daily, use cheapest options, turn off resources when not testing

Team Availability

The worry: Someone gets busy with exams or personal stuff

Our solution: Everyone knows a bit about each other's work, detailed documentation, flexible task reassignment

Technical Difficulties

The worry: Something doesn't work as expected

Our solution: Start learning early, help each other, ask instructor when stuck, test frequently

Time Pressure

The worry: Not finishing on time

Our solution: Clear weekly goals, prioritize core features first, track progress closely

10. Quality Checks

Code Quality

- Follow Python best practices (PEP 8)
- Someone else reviews your code before merging
- Test your work with different scenarios
- Add comments so others understand your code

Testing Schedule

- **Every Friday:** Test what we built this week
- **Each Milestone:** Test the whole system together
- **Final Week:** Test everything multiple times

Documentation

- Write as we go (not at the end)
 - Include screenshots and examples
 - Explain things clearly for others
 - Keep it organized and professional
-

11. What Makes Us Succeed

By Milestone 1 (October 15)

We built a working simulator that generates realistic traffic data and streams it to Azure. Everything flows smoothly with zero errors.

By Milestone 2 (November 2)

Our system can automatically detect congestion, speeding, and accidents with 95%+ accuracy. All data is stored properly and queries run fast.

By Final Delivery (November 28)

We have a beautiful dashboard showing live traffic, automatic alerts working perfectly, complete documentation, and a successful presentation.

12. Important Dates

Date	Milestone	What Happens
Sept 15	Project Start	Team formation, kick-off meeting
Oct 15	Milestone 1 Due	Simulator & Event Hub complete
Nov 2	Milestone 2 Due	Analytics & Storage complete
Nov 28	Final Submission	Dashboard, alerts, & presentation

13. Lessons We're Learning

Technical Skills

- Python programming for data engineering
- Cloud services (Azure Event Hubs, Stream Analytics, SQL)
- Real-time data processing
- Dashboard development with Streamlit
- Database design and optimization

Professional Skills

- Working as a team on a complex project
- Managing time and meeting deadlines
- Solving problems collaboratively
- Presenting technical work clearly
- Writing professional documentation

Real-World Experience

- Building something that could actually be used
- Understanding how traffic monitoring systems work
- Learning cloud architecture and cost management
- Experiencing the full data engineering lifecycle

14. Our Approach

Start Simple, Then Improve

We're building the basic functionality first, making sure it works well, then adding enhancements. Core features before fancy features.

Test Early, Test Often

Rather than building everything then testing at the end, we test each piece as we build it. This catches problems early when they're easier to fix.

Help Each Other

When someone gets stuck, we jump in to help. We're a team, and we succeed together. No one works in isolation.

Document Everything

As we build, we document what we're doing and why. This helps when we need to remember later, and it makes our final report easier to write.

Stay Flexible

If something isn't working, we're ready to adjust our approach. The goal is a working system, not stubbornly sticking to one method.

15. Why This Project Matters

For Our Learning: This is hands-on experience with real data engineering tools and practices that companies actually use.

For Our Future: The skills we're gaining (Python, Azure, data processing, visualization) are exactly what employers look for.

For Cairo: While we're using simulated data, this type of system could really help manage traffic, reduce congestion, and improve safety.

For Our Portfolio: We'll have a complete, working project to show potential employers—proof that we can build real solutions.

Document Prepared By: Ali Elgatool, Team Leader

Last Updated: November 1, 2025

Status: Submitted to Instructor