INNOMATICS TECHNOLOGY HUB

Data Science Internship – February 2026

Internship Task Documentation

Task Instructions

1. Log in to your **LMS** and navigate to:
   Assessment & Task → Task 6: Function-Based Problem Solving
2. Open the **Google Form** provided in the task section to access your assigned Python problem.
3. Solve the problem using either **Jupyter Notebook** or **Google Colab**.
   Save your solution file in **.ipynb** format.
4. Upload (push) the `.ipynb` file to your **GitHub repository**.
   Ensure the repository link is in **HTTPS format** (e.g., `https://github.com/username/repository-name`).
5. Complete the **Google Form** by entering your required details and pasting your **GitHub repository HTTPS link**, then submit the form.

Submission Guidelines

- Your code must be **clean, well-structured, and properly organized**.
- Include **clear comments** explaining your logic wherever necessary.

Only submissions with a valid **GitHub HTTPS link submitted through the Google Form** will be considered for evaluation.

ASSIGNMENT 6

Problem Statement 1:Smart Parking Lot Management System

Design a function to manage a smart parking lot.

The system should:

1. Accept vehicle entry and exit logs
2. Calculate total parked vehicles
3. Identify peak parking usage
4. Alert if parking exceeds capacity

Real-Time Use

- Mall parking systems
- Smart city infrastructure

Hint

- Function + list
- Loop for counting
- Conditional alert

Sample Input

Parking Capacity: 50
Vehicle Logs: ["IN", "IN", "IN", "OUT", "IN", "IN", "OUT"]

Expected Output

Currently Parked Vehicles: 3
Parking Status: Available

Problem Statement 2:Online Food Delivery Time Estimator

Create a function that estimates **delivery time** based on:

- Distance (km)
- Weather condition
- Traffic level

Apply delays dynamically and display final ETA.

Real-Time Use

- Food delivery apps
- Logistics platforms

Hint

- Function + multiple conditions
- Mathematical adjustments

Sample Input

Distance (km): 8
Traffic Level: High
Weather: Rainy

Expected Output

Estimated Delivery Time: 55 minutes

Problem Statement 3:Movie Theatre Seat Occupancy Analyzer

Build a function that analyzes seat booking data and:

1. Calculates occupancy percentage
2. Determines if show is Housefull
3. Suggests opening additional shows

Real-Time Use

- Cinema ticketing systems
- Event management software

Hint

- Function + list
- Percentage calculation

Sample Input

Total Seats: 200
Booked Seats: [1,1,1,1,1,1,1,1,1,1,1] (150 entries)

Expected Output

Occupancy: 75%
Show Status: Almost Full

Problem Statement 4:Cloud Server Load Classification System

Create a function to classify **server load** based on CPU usage readings.

Rules:

- Average CPU < 50% → Normal
- 50%–80% → Warning
- 80% → Critical

Real-Time Use

- Cloud monitoring dashboards
- DevOps alerting systems

Hint

- Function + loop
- Average calculation

Sample Input

CPU Readings (%): [45, 60, 70, 85, 90]

Expected Output

Average CPU Load: 70%
Server Status: Warning

Problem Statement 5:Smart Classroom Resource Usage Monitor

Design a function that tracks usage of classroom resources (projector, AC, lights) and identifies **overuse patterns**.

Real-Time Use

- Smart classrooms
- Energy optimization systems

Hint

- Function + dictionary
- Conditional checks

Sample Input

Resource Usage (hours):

```
{
 "Projector": 6,
 "AC": 9,
 "Lights": 4
}
```

Expected Output

```
Overused Resources: AC
Energy Alert: Yes
```

Problem Statement 6: Online Event Registration Capacity Controller

Create a function that manages event registrations by:

1. Tracking registrations
2. Preventing overbooking
3. Triggering waitlist mode

Real-Time Use

- Webinar platforms
- Conference registration systems

Hint

- Function + loop
- Capacity validation

Sample Input

Event Capacity: 100
Registrations: 105

Expected Output

Confirmed Registrations: 100
Waitlisted Users: 5
Registration Status: Closed