TASK 4 OPTIMIZATION MODEL

SOLVE A BUSINESS PROBLEM USING OPTIMIZATION TECHNIQUES (E.G., LINEAR PROGRAMMING) AND PYTHON LIBRARIES LIKE PULP

✅ Project Title: Production Planning Optimization using Linear Programming

🔧 Problem Statement: A factory produces 2 products:

Product A

Product B

The factory has limited resources:

Resource — Available Hours

Machine — 1 40 hours

Machine — 2 50 hours

Product A requires:

1 hour on Machine 1

2 hours on Machine 2

Product B requires:

2 hours on Machine 1

1 hour on Machine 2

Profit per unit:

Product A = $30

Product B = $20

Goal: Maximize total profit while not exceeding machine capacities.

```
pip install pulp
```

```
Collecting pulp
  Downloading pulp-3.2.1-py3-none-any.whl.metadata (6.9 kB)
Downloading pulp-3.2.1-py3-none-any.whl (16.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.4/16.4 MB 91.7 MB/s eta 0:00:00
Installing collected packages: pulp
Successfully installed pulp-3.2.1
```

```
# ELiteTech Internship - Task 4: Optimization Model
# Production Planning using Linear Programming

import pulp

# Initialize LP problem: Maximize profit
model = pulp.LpProblem("Maximize_Profit", pulp.LpMaximize)

# Decision variables
A = pulp.LpVariable("Product_A", lowBound=0, cat='Integer')
B = pulp.LpVariable("Product_B", lowBound=0, cat='Integer')

# Objective function (maximize profit)
model += 30*A + 20*B

# Constraints
model += 1*A + 2*B <= 40   # Machine 1 constraint
model += 2*A + 1*B <= 50   # Machine 2 constraint

# Solve the problem
model.solve()

# Results
print(f"Status: {pulp.LpStatus[model.status]}")
print(f"Produce {A.varValue} units of Product A")
print(f"Produce {B.varValue} units of Product B")
print(f"Maximum Profit: ${pulp.value(model.objective)}")
```

```
Status: Optimal
Produce 20.0 units of Product A
Produce 10.0 units of Product B
Maximum Profit: $800.0
```

+ Code + Text