# Smart Parking Dynamic Pricing System

================================================

## Overview :-

This project implements a dynamic pricing system for parking lots. The system uses real-time data and competitive analysis to optimize parking prices, considering factors such as occupancy, queue length, traffic conditions, vehicle types, and special events. Interactive visualizations help monitor and analyze pricing trends.

## Tech Stacks :-

| Technology | Purpose |
|---|---|
| Python | Core programming language |
| Pandas | Data processing and manipulation |
| NumPy | Numerical operations |
| Geopy | Geospatial distance calculations |
| Bokeh | Interactive data visualization |
| Jupyter Notebook | Development and demonstration |

## Architecture Diagram :-

### Flowchart :-

```
A[Data Input (CSV)] --> B[Preprocessing]
    B --> C[Demand Computation]
    C --> D[Pricing Algorithm]
    D --> E[Competitive Adjustment]
    E --> F[Visualization (Bokeh)]
    F --> G[Output (Notebook/Plots)]
```

# System Architecture & Workflow :-

## Data Input & Preprocessing

- Input: CSV file with parking lot data (timestamps, occupancy, queue length, traffic, special day, vehicle type).
- Preprocessing:
  - Merge date and time into a single timestamp.
  - Sort data chronologically by lot.

## Demand Computation

- For each lot and time point, compute a *demand score* using:
  - Occupancy ratio
  - Queue length
  - Traffic condition (numerical mapping)
  - Special day indicator
  - Vehicle type impact

## Pricing Algorithm

- Baseline Pricing: Sets an initial price, adjusted by occupancy.
- Demand-Based Pricing: Modifies the price based on the computed demand score, normalized for sensitivity.
- Competitive Adjustment:
  - Identifies nearby lots within a specified radius.
  - Compares prices and adjusts to remain competitive.

## Visualization

- Bokeh generates interactive plots for each lot:
  - Shows dynamic price over time.
  - Displays average competitor price for comparison.

## Output

- Streams real-time price updates for each lot.
- Visualizes pricing and competitor trends in the notebook environment.

## Working Code

- pip install pathway bokeh geopy --quiet
- Set up the virtual environment (env file ) for the code implementation.
- Also install all necessary libraries:-
  > pip install numpy
  > pip install panas
  > pip install geopy
- Code the code for the provided ipynb file.

# Customization & Extensibility

- **Adjustable Parameters:** All weights and sensitivity factors are configurable in the code.
- **Extensible Design:** Can be expanded to include features like weather, events, or historical trends.