**Capstone Project Guide: Python**

# Summary of the Capstone Project Requirements

Project has two main parts:  
  
1. \*\*Customer Data Cleanup\*\*  
 - Take a messy customer list, clean it, and then output the cleaned data in a `.CSV` file.  
 - Steps include:  
 - Removing any stray characters from names.  
 - Formatting phone numbers as `NNN-NNN-NNNN`.  
 - Handling missing values in the `sms-opt-out` field (a choice based on research or logical assumptions).  
  
2. \*\*Transaction Data Analysis\*\*  
 - Analyze a dataset of 2024 transactions to find insights about the business.  
 - Need following done:  
 - Load the data into a DataFrame, show the first 10 rows, and get a statistical summary.  
 - Generate at least three additional insights using aggregation or statistics.  
 - Create a data visualization, such as a graph or chart, with proper titles and axis labels.

# Project Deliverables

1. \*\*Code and Cleaned Data\*\*: Write a Python program (script or Jupyter Notebook) to clean the customer data.  
2. \*\*Data Analysis Notebook\*\*: A Jupyter Notebook with data analysis and visualization of the transactions.  
3. \*\*Documentation and Repository\*\*: A well-documented GitHub repository with daily commits and a `README` file.

# Suggested Guide to Complete This Project

## Part 1: Customer Data Cleanup

1. \*\*Examine the Data\*\*  
 - Load the customer data into a DataFrame using `pandas`.  
 - Review the data to understand what kind of cleaning is needed.  
  
2. \*\*Cleaning Steps\*\*  
  
 \*\*Step-by-Step Instructions:\*\*  
 - \*\*Names\*\*: Use regex to keep only letters, hyphens, and periods.  
 ```python  
 df['name'] = df['name'].str.replace(r"[^a-zA-Z\.\- ]", "", regex=True)  
 ```  
 - \*\*Phone Numbers\*\*: Ensure each number follows the `NNN-NNN-NNNN` format. Use a function to format and correct phone numbers.  
 ```python  
 def format\_phone(phone):  
 # Apply phone formatting here  
 return formatted\_phone  
 df['phone'] = df['phone'].apply(format\_phone)  
 ```  
 - \*\*`sms-opt-out` Field\*\*: Decide how to handle blanks based on research or a logical assumption (e.g., assume blanks mean opt-out).  
  
3. \*\*Export the Cleaned Data\*\*  
 - Save the cleaned DataFrame as a `.CSV` file.  
 - Confirm it saved correctly by loading and printing the first few rows.  
  
4. \*\*Document the Process\*\*  
 - Add comments to each step in your code explaining what it does.  
 - Add commit messages like “Cleaned name field” or “Formatted phone numbers” to your GitHub repository.

## Part 2: Transaction Data Analysis

1. \*\*Initial Setup\*\*  
 - Load the transaction data into a new DataFrame in a Jupyter Notebook.  
 - Display the first 10 rows and use `.describe()` to get a basic summary.  
  
2. \*\*Generate Insights\*\*  
 - Use `groupby()`, `sum()`, or `mean()` to analyze sales performance by different metrics like product category, month, or region.  
 - Examples of insights:  
 - Total sales per month  
 - Average sale amount by category  
 - Most popular product types  
 - Document each insight with Markdown text.  
  
3. \*\*Create a Plot\*\*  
 - Use `Matplotlib` or `Seaborn` to visualize one or more of these insights.  
 - Ensure that your plot has a title and labeled axes.  
  
4. \*\*Document and Save\*\*  
 - Add Markdown cells to explain each part of analysis.  
 - Once the analysis is complete, save notebook and make regular commits.

## Optional Add-ons

1. \*\*Combine Customer and Transaction Data\*\*: Merge the datasets to analyze customer purchasing behavior.  
2. \*\*Customer Data Patterns\*\*: Trying finding patterns, such as area codes for geographic insights, which could guide future marketing.

# Presentation Tips

- \*\*Practice\*\*: Do a quick run-through to stay within the 10-minute limit.  
- \*\*Prepare Notes\*\*: Use notes to keep your presentation organized.  
- \*\*Share Insights\*\*: Emphasize what was learned from the data and share any interesting patterns found.