

SIX WEEKS INTERNSHIP/ TRAINING REPORT

On

SOFTWARE ENGINEERING

COMPLETED AT

BHARAT HEAVY ELECTRICALS LIMITED, VARANASI



SUBMITTED BY:

Ridhi Singh

221630

(DEPARTMENT OF COMPUTER
SCIENCE AND ENGINEERING)

Submitted To :

Mr. Ram Ashish Singh Kushwaha

CENTRAL UNIVERSITY OF HARYANA

[2022-26]



TABLE OF CONTENTS

- 1. Abstract**
- 2. Acknowledgement**
- 3. Executive Summary**
- 4. Introduction**
- 5. Sectors Targeted by BHEL HERP Varanasi**
- 6. Policies Implemented by BHEL HERP Varanasi**
- 7. Project**
 - 7.1. Task – 1**
 - 7.2. Task – 2**
- 8. Skills and Learning**
- 9. References**

ACKNOWLEDGMENT

I would like to express my deepest gratitude to BHEL HERP, Varanasi, for providing me with the opportunity to undertake my summer internship at their esteemed organization. This experience has been incredibly enriching and has significantly contributed to my professional growth and understanding of the industry.

I am profoundly grateful to my mentor, Mr. Ram Ashish Singh Kushwaha, for his invaluable guidance, support, and encouragement throughout the training period. His expertise and insights have been instrumental in shaping my understanding of the various aspects of vendor analysis. His patience and willingness to share his vast knowledge have been a great source of inspiration for me.

I would also like to extend my sincere thanks to the entire team at BHEL HERP, Varanasi, for their cooperation and support. The knowledge and skills I have acquired during this training have provided me with a solid foundation for my future career endeavours.

Finally, I am thankful to my family and friends for their continuous support and encouragement during the training period.

This report on "Analysis of Vendors in BHEL HERP, Varanasi" is the culmination of my efforts and learning during the training, and I hope it will be a valuable contribution to the organization.

Thank you.

Ridhi Singh

Central University of Haryana

ABSTRACT

I would like to express my sincere gratitude to the entire unit of Bharat Heavy Electricals Limited, HERP, Varanasi, for providing me with an exceptional summer internship experience. I would like to extend my special thanks to Mr. Ram Ashish Singh Kushwaha (Sr. Manager - Digital Transformation Group) and Mr. Abhishek Kumar Srivastava (Dy. Manager - Digital Transformation Group), for their invaluable support and guidance throughout the training period.

The vocational training at BHEL HERP has been an incredible learning opportunity for me, allowing me to gain practical insights into industry practices and hands-on experience in the field of IT.

I would like to specifically acknowledge the contributions and guidance of Mr. Ram Ashish Singh Kushwaha. His mentorship and support have been instrumental in shaping my project and helping me overcome challenges along the way.

Furthermore, I want to express my heartfelt gratitude to Mr. Abhishek Srivastava for his exceptional guidance and support throughout the vocational training. His patience, encouragement, and continuous feedback have played a vital role in my project's successful completion.

I am honoured to have been a part of such an esteemed organization, and I am confident that the skills and knowledge I have acquired will serve me well in my future endeavours.

EXECUTIVE SUMMARY

The "Analysis of Vendors in BHEL (HERP), Varanasi" paper offers a complete assessment of the vendor ecosystem at Bharat Heavy Electricals Limited's Heavy Equipment Repair Plant (HERP) in Varanasi. This research intends to provide insights into the vendor selection process, evaluation criteria, and overall vendor influence on BHEL HERP Varanasi's operational efficiency and strategic goals.

BHEL HERP Varanasi, a key unit within BHEL, manufactures and refurbishes heavy electrical equipment for a variety of industries, including power generation, transmission and distribution, industrial machinery, transportation, renewable energy, defence, oil and gas, and mining and metallurgy. The plant's performance is significantly dependent on a strong vendor network.

This report opens with an overview of BHEL HERP Varanasi, including its historical background, infrastructure, manpower, and contributions to several industries. It provides context for understanding the crucial role that suppliers play in supporting the plant's operations.

The following sections discuss the vendor selection and assessment procedure. BHEL HERP Varanasi uses tough criteria to guarantee that only the most dependable and skilled vendors are hired. These factors include quality standards, delivery schedules, cost-effectiveness, technical capabilities, and regulatory adherence.

Furthermore, the report examines the difficulties encountered by vendors and BHEL HERP Varanasi in ensuring seamless collaboration. These hurdles include volatile market circumstances, legislative changes, technology developments, and logistical concerns.

The report finishes with emphasizing the necessity of a robust vendor network in attaining operational excellence, satisfying customer requests, and maintaining a competitive edge in the marketplace.

In summary, this report presents a complete examination of the vendor ecosystem at BHEL HERP Varanasi, providing significant insights for stakeholders looking to increase cooperation, supply chain efficiency, and promote the plant's long-term success.

INTRODUCTION

Bharat Heavy Electricals Limited (BHEL) is one of India's leading engineering and manufacturing companies in the energy and infrastructure industries. Among its many sections, the Heavy Equipment Repair Plant (HERP) in Varanasi is particularly important due to its specialized emphasis and strategic location. It has since evolved to be a cornerstone of BHEL's activities, contributing to a variety of national initiatives and industrial achievements.

Historical Background

The origins of BHEL may be traced back to the early 1960s, when it was founded to attain self-sufficiency in the production of heavy electrical equipment. HERP Varanasi was established as part of this larger objective. Its growth has matched India's economic prosperity, with constant modernization and capacity expansion to keep up with technical breakthroughs and rising demand.

Location and Area

HERP is conveniently placed in Varanasi, one of the world's oldest towns and a significant cultural and economic hub in the state of Uttar Pradesh. Its geographical location provides logistical benefits, such as closeness to important industrial zones and access to a well-established transportation network, which facilitates the efficient transfer of raw materials and completed goods.

Infrastructure and Capabilities

HERP Varanasi boasts cutting-edge infrastructure that enables a wide range of businesses. The complex has modern machining centers, fabrication shops, assembly units, and testing laboratories. These characteristics allow HERP to perform difficult repair and production jobs with great accuracy and quality.

The plant's primary emphasis areas are the refurbishing of turbines, generators, and other heavy equipment, which are critical for maintaining and improving the operating efficiency of power plants around the country.

Workforce

The strength of HERP Varanasi is its devoted and professional team. The unit employs more than 1,000 engineers, technicians, and support workers. This staff is critical to maintaining HERP's reputation for competence and dependability. Continuous training and development initiatives are in place to keep the personnel current on technical changes and industry best practices.

Contribution to BHEL and the Nation

HERP Varanasi is an important part of BHEL's overall strategy. It not only increases the company's income but also helps to achieve the larger aim of national self-reliance in heavy electrical equipment manufacture. Through its high-quality goods and services, HERP contributes to the efficient operation of power plants, which is vital for India's energy security and industrial growth.

To summarize, BHEL HERP Varanasi demonstrates India's engineering brilliance and industrial capacity. Its historical history, strong infrastructure, experienced staff, and strategic contributions highlight its significance within BHEL and in the country's industrial scene.

SECTORS TARGETED BY BHEL HERP VARANASI

Bharat Heavy Electricals Limited (BHEL), notably through its Heavy Equipment Repair Plant (HERP) in Varanasi, plays an important role in supplying numerous sectors with its manufacturing expertise. BHEL HERP Varanasi targets the following key sectors:

Power Generation

The power generating sector is a major beneficiary of BHEL HERP Varanasi's manufacturing capability. The factory manufactures and refurbishes essential components for thermal, hydroelectric, and nuclear power plants. HERP Varanasi plays an important role in satisfying the country's expanding energy demands by assisting the power sector.

Transmission and Distribution

BHEL HERP Varanasi expands its experience into the transmission and distribution (T&D) sector, which is critical for efficiently delivering electricity from power plants to end customers. Transformers, switchgear, and other substation components are among the T&D equipment manufactured at the site. HERP's contributions to the technology and development sector help to expand and modernize India's electricity infrastructure.

Industrial Machinery

The industrial machinery industry greatly benefits from HERP Varanasi's superior production capabilities. Pumps, compressors, heat exchangers, and other mechanical components are commonly utilized in sectors such as steel, cement, petrochemicals, and refineries. HERP's high-quality products ensure that these sectors run effectively and sustainably, helping to boost overall industrial growth and productivity.

Transportation

Transportation comprises traction motors, alternators, and other vital components for locomotives and rolling equipment. HERP's participation in the transportation sector promotes the development of dependable and efficient transportation networks, which are critical to economic growth and connectivity.

Renewable Energy

As the globe moves toward more sustainable energy options, BHEL HERP Varanasi is heavily involved in the renewable energy industry. The facility makes wind turbine and solar power system components, which help to construct green energy projects. HERP aims to minimize carbon emissions and promote ecologically responsible energy options.

Defence

BHEL HERP Varanasi also provides important contributions in the defence industry. The company manufactures numerous components and equipment for defence purposes, assuring the dependability and performance of military gear. HERP's cooperation in the defence industry strengthens national security and serves the country's strategic objectives.

Oil and Gas

The oil and gas industry depends on HERP Varanasi for crucial equipment and components used in exploration, extraction, and processing. The company produces pumps, compressors, and other mechanical components required for the proper operation of oil and gas installations.

Mining and Metallurgy

The mining and metallurgical sectors also benefit from HERP Varanasi's production capabilities. The plant manufactures heavy-duty equipment and components for mining and metal processing industries. Mineral extraction and processing need the use of crushers, conveyors, and grinding mills.

In summary, BHEL HERP Varanasi manufactures goods for a wide range of industries, including power production, transmission and distribution, industrial machinery, transportation, renewable energy, military, oil and gas, and mining and metallurgy. The plant's sophisticated skills and dedication to quality guarantee that it continues to play an important role in supporting and advancing these crucial industries, helping to drive India's industrial and economic growth.

POLICIES IMPLEMENTED BY BHEL HERP VARANASI

BHEL HERP Varanasi follows a number of regulations designed to ensure efficient operations, quality assurance, environmental sustainability, and employee welfare. These policies serve as the foundation for the organization's strategic goals and day-to-day operations, allowing it to preserve its reputation for quality and reliability.

1. Quality Assurance Policy

- ***Commitment to Excellence:*** BHEL HERP Varanasi is devoted to providing high-quality goods and services.
- ***Continuous Improvement:*** The policy stresses continuous improvement through frequent audits, feedback systems.

2. Environmental Policy

- ***Sustainability Initiatives:*** This involves eliminating trash, saving energy, and utilizing environmentally friendly products.
- ***Compliance with Regulations:*** BHEL HERP Varanasi complies with all environmental laws and regulations, guaranteeing national and international standards.

3. Healthy and Safety Policy

- ***Workplace Safety:*** Creating a safe work environment is a major concern.
- ***Health Monitoring:*** Employees' health is monitored and improved through regular check-ups and wellness programs.

4. Vendor Management Policy

- ***Vendor Selection:*** BHEL HERP. Varanasi has a comprehensive vendor selection procedure to guarantee that all vendors fulfil the company's quality and ethical requirements.
- ***Vendor Performance Evaluation:*** Regular reviews and audits are performed to analyse vendor performance, maintaining consistency and dependability in the supply chain.

5. Human Resource Policy

- **Employee growth:** The policy emphasizes ongoing learning and growth via training programs, workshops, and educational opportunities.
- **Equal Opportunity:** BHEL HERP Varanasi is committed to offering equal employment opportunities and creating a diverse and inclusive workplace.

6. Corporate Social Responsibility (CSR) Policy

- **Community Engagement:** To benefit the local community, the plant participates in a variety of community development activities such as education, healthcare, and infrastructure development.
- **Sustainable Development Goals:** BHEL HERP Varanasi's CSR operations are aligned with the United Nations' Sustainable Development Goals (SDGs), with an emphasis on poverty reduction, excellent education, and renewable energy.

7. Energy Management Policy

- **Energy Efficiency:** The policy emphasizes the importance of energy efficiency in all activities and encourages the use of energy-saving technology and practices.
- **Energy Audits:** Regular energy audits are carried out to discover and implement energy-saving strategies.

8. Ethical Conduct and Compliance Policy

- **Code of Conduct:** A strong code of conduct exists, detailing ethical standards and business procedures that all workers must follow.
- **Anti-Corruption:** The policy contains steps to prevent corruption and encourage ethical behaviour throughout the company.

In summary, BHEL HERP Varanasi's policies address a wide variety of issues, including quality assurance, environmental sustainability, employee welfare, and vendor management. These principles are intended to safeguard the plant's efficient and ethical operation, which contributes to its overall performance and reputation as a leader in the heavy electrical equipment manufacturing sector.

INTERNSHIP-TRAINING STRUCTURE

WEEK 1: Python Basics and Streamlit Introduction

WEEK 2: Project 1

WEEK 3: What is data analytics? Introduction to Python libraries like pandas, numpy.

WEEK 4: What is data visualisation? Working with Power BI and python libraries like Seaborn and matplotlib.

WEEK 5: Project 2.1 (group)

Base: Data Analytics

WEEK 6: Project 2.2 (group)

Base: Data Visualization

Project 1

Task:

The Visitor Management System is designed to streamline the process of managing visitors entering and exiting a facility. It captures essential details about the visitors, such as their name, contact number, purpose of visit, and photograph, along with their check-in and check-out times. The system ensures efficient data management and security through a user-friendly interface, which is divided into two main sections: Visitor and Admin.

Objectives

- To create an intuitive interface for visitors to check in and check out of the facility.
- To securely store visitor data, including personal information and visit details.
- To provide an admin interface for monitoring and managing visitor records.
- To ensure data integrity and security throughout the system.

Technologies Used

```
import streamlit as st
import sqlite3
import datetime
import random
import string
import base64
import pandas as pd
import io
from PIL import Image
```

- **Streamlit:** For creating the interactive web interface.
- **SQLite:** As the database for storing visitor information.
- **Pandas:** For data manipulation and display.
- **PIL (Python Imaging Library):** For handling image data.
- **Base64:** For encoding and decoding images.
- **Datetime:** For handling date and time operations.
- **Random & String:** For generating random gate pass numbers

System Components

Database Management

- **SQLite Connection:** Establishes a connection to the SQLite database.
- **Table Creation:** Ensures the visitor table is created if it doesn't exist.
- **Data Insertion:** Adds new visitor records to the database.
- **Data Update:** Updates check-out times for visitors.
- **Data Retrieval:** Fetches all visitor records and specific records by gate pass number.

```
def connection():
    conn = sqlite3.connect('gateKeeper.db')
    return conn

def create_table(conn):
    c = conn.cursor()
    c.execute('CREATE TABLE IF NOT EXISTS visitor
              (
                name TEXT, contact_number TEXT, purpose_of_visit TEXT, additional_info TEXT, checkIn_time TEXT,
                gate_pass TEXT)')
    conn.commit()
```

```
def add_visitor(conn, name, contact_number, purpose_of_visit, additional_info, checkIn_time, gate_pass, picture):
    c = conn.cursor()
    c.execute('INSERT INTO visitor (name, contact_number, purpose_of_visit, additional_info, checkIn_time, gate_pass, picture)
              VALUES (name, contact_number, purpose_of_visit, additional_info, checkIn_time, gate_pass, picture)')
    conn.commit()

def update_checkout_time(conn, gate_pass, checkout_time):
    c = conn.cursor()
    c.execute('UPDATE Visitor SET checkout_time = ? WHERE gate_pass = ?', (checkout_time, gate_pass))
    conn.commit()
```

```
def generate_GatePassNo():
    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=8))

def get_allVisitors(conn):
    c = conn.cursor()
    c.execute('SELECT * FROM visitor')
    data = c.fetchall()
    return data

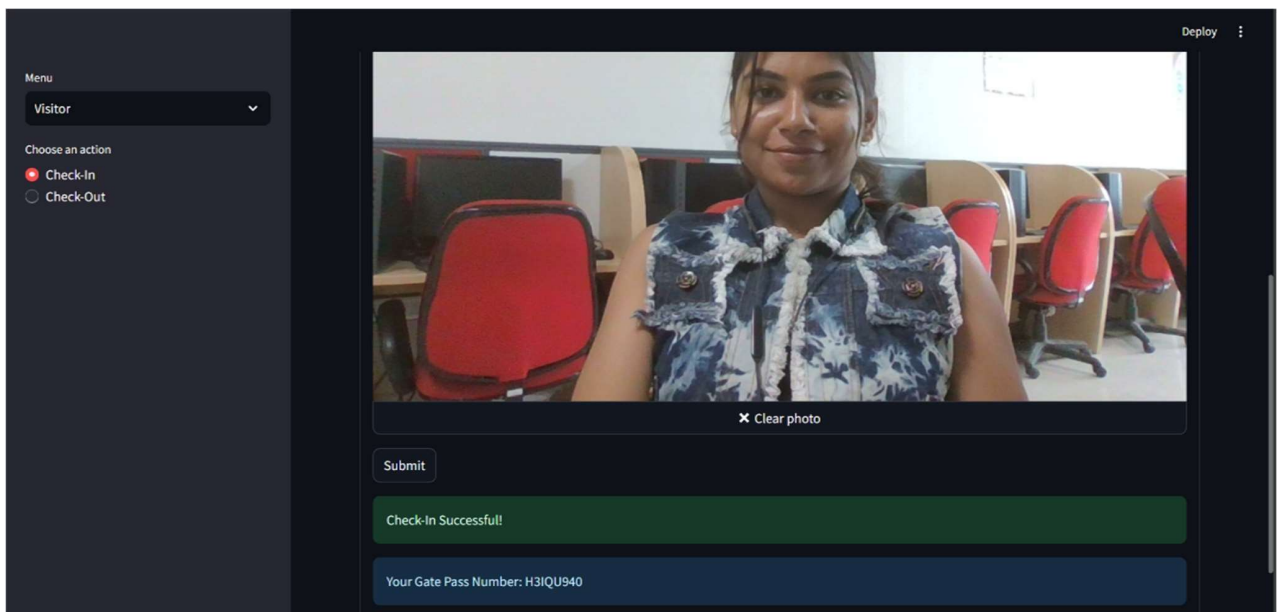
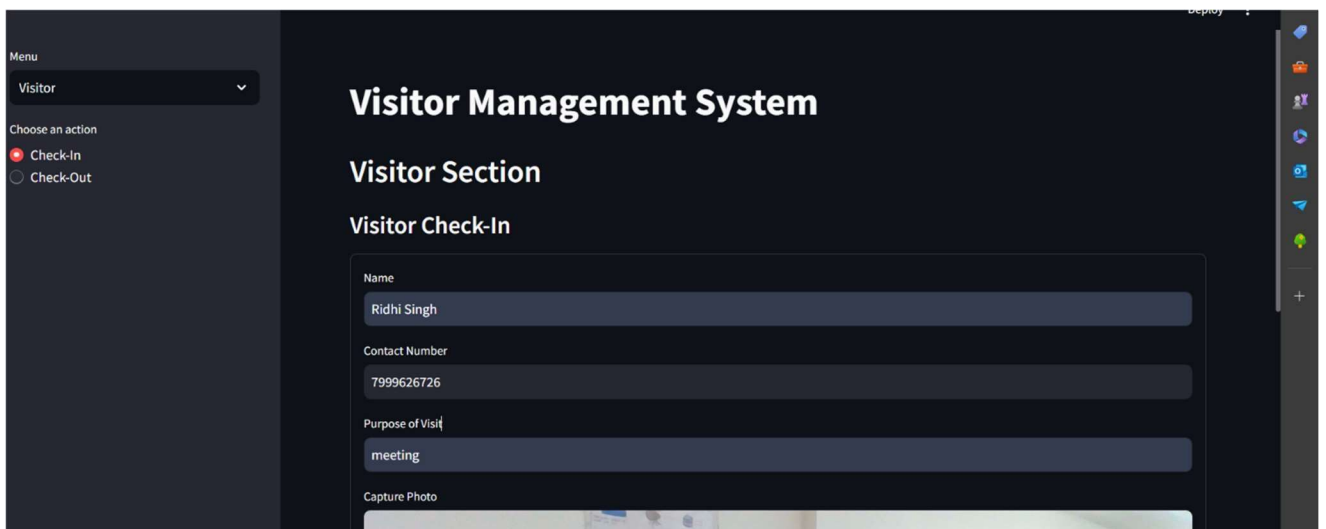
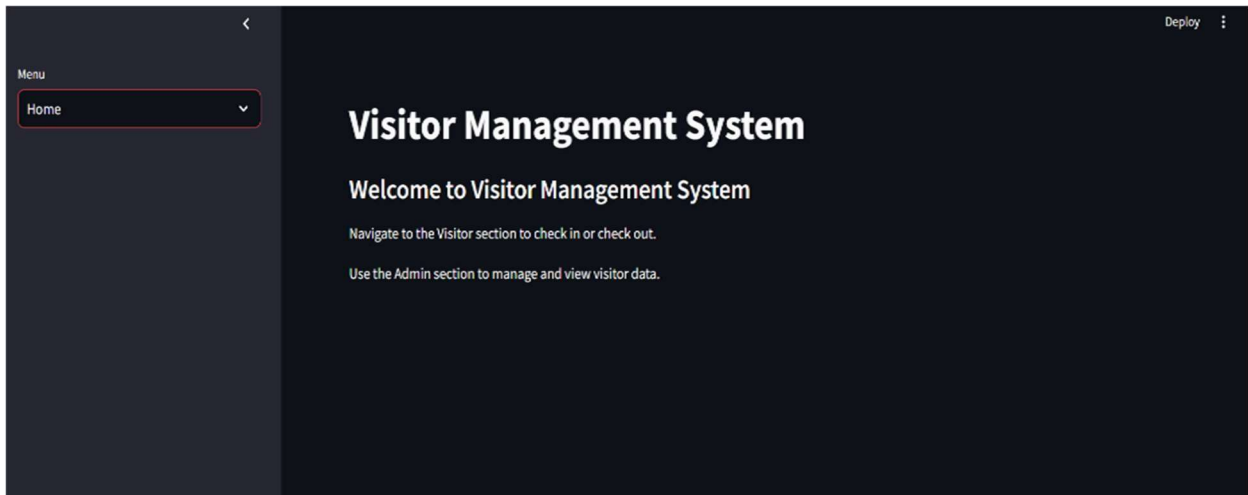
def get_visitor_by_name_and_contact(conn, name, contact_number):
    c = conn.cursor()
    c.execute('SELECT * FROM visitor WHERE name=? AND contact_number=?', (name, contact_number))
    data = c.fetchone()
    return data

def get_gate_pass_by_details(conn, name, contact_number):
    c = conn.cursor()
    c.execute('SELECT gate_pass FROM visitor WHERE name=? AND contact_number=?', (name, contact_number))
    data = c.fetchone()
    return data
```

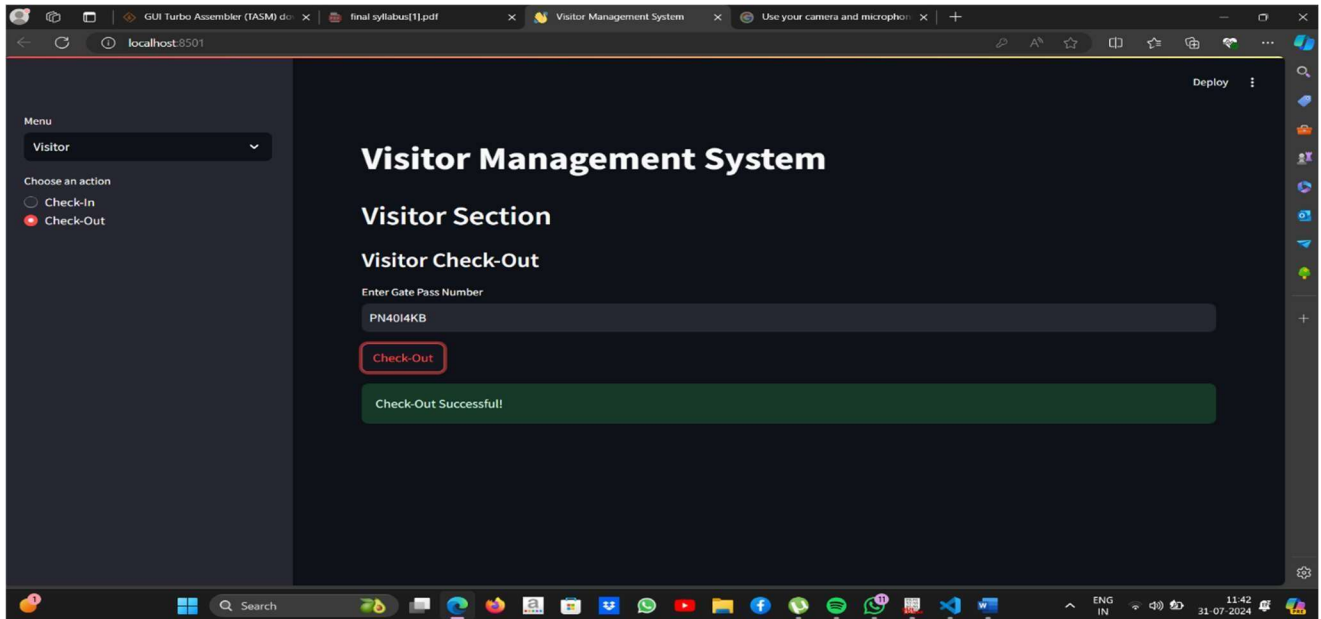
Visitor Section

- **Check-In Process:** Captures the visitor's name, contact number, purpose of visit, and picture. Generates a gate pass number. Stores

the check-in details in the database. Displays the generated gate pass number to the visitor.

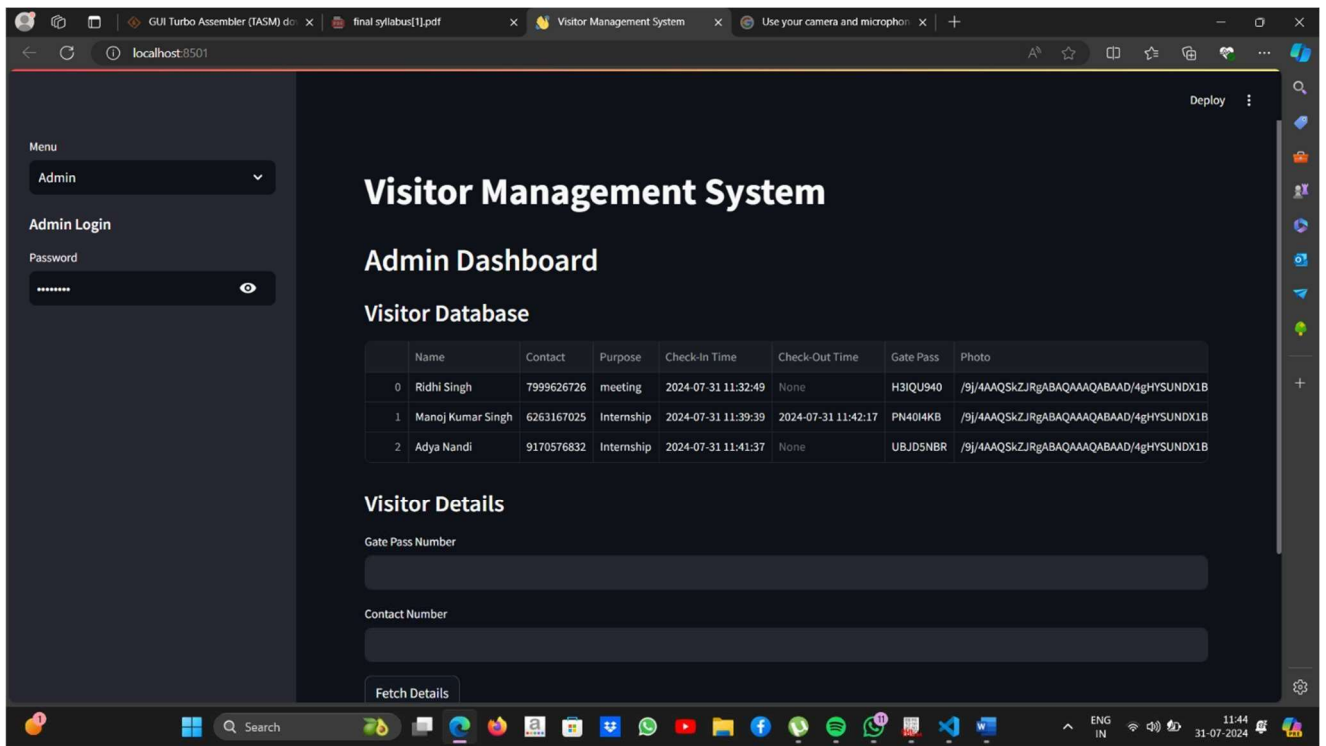


- **Check-Out Process:** Allows visitors to check out by entering their gate pass number. Updates the check-out time in the database. Also retrieves the gate pass number and provides it to the user in case user forgets. Confirms successful check-out to the visitor.



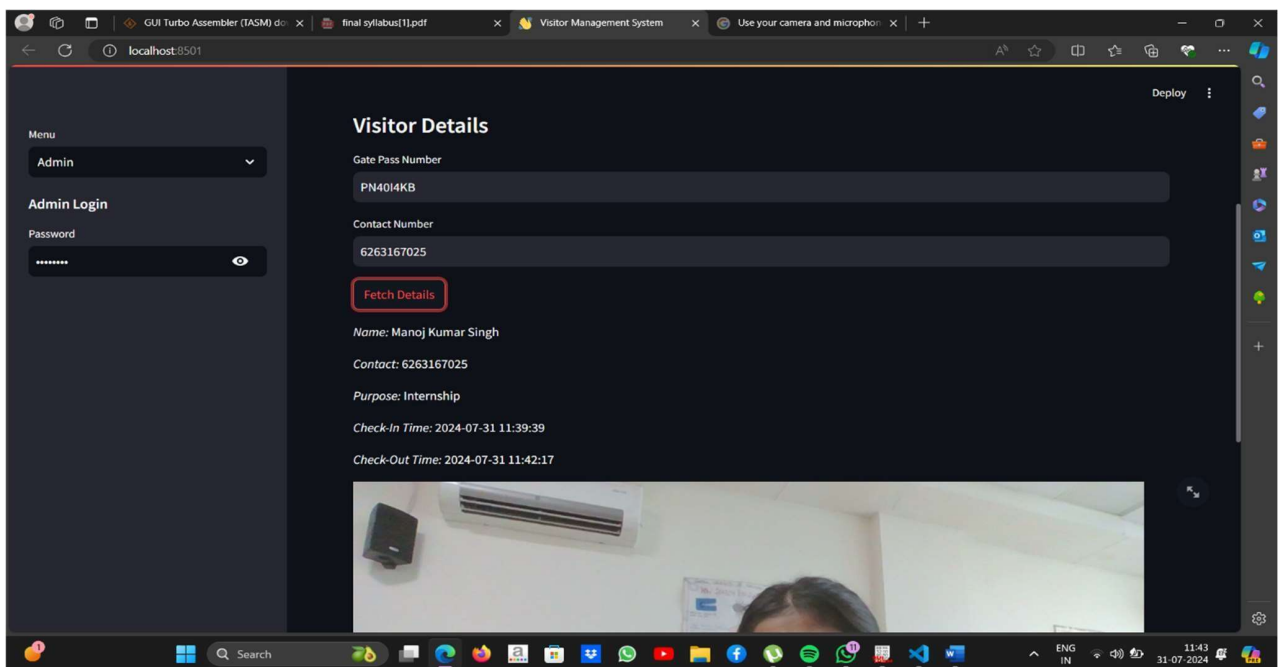
Admin Section

- **Password Protected Access:** Ensures only authorized personnel can access the admin section.

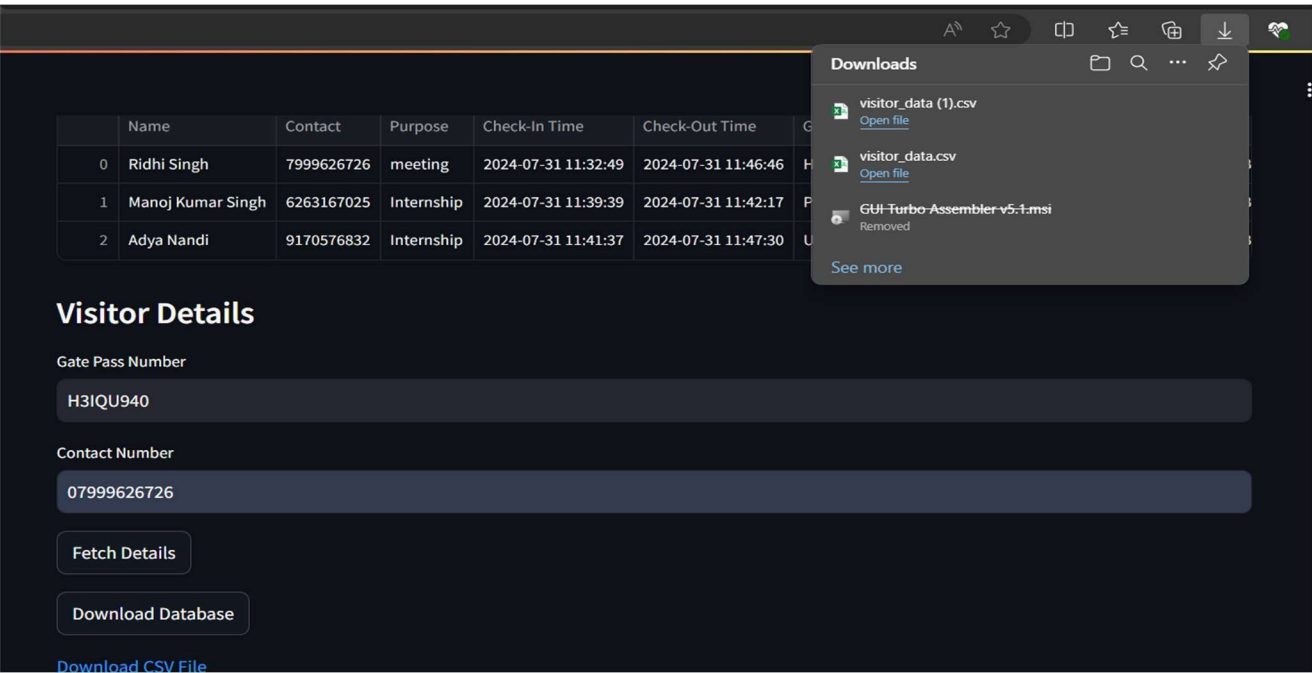


Visitor Data Display: Shows a table of all visitor records.

View Specific Visitor Details: Allows the admin to view details of a visitor by entering their gate pass number.



Download Data: Provides an option to download the visitor data as a CSV file.



Key Features

- **Data Integrity:** Ensures all required fields are filled out and validates the contact number format.
- **Security:** Admin access is protected by a password.
- **User-Friendly Interface:** Provides a simple and intuitive interface for both visitors and administrators.
- **Efficient Data Management:** Allows for easy addition, update, retrieval, and download of visitor data.

Challenges and Solutions

Challenge: Ensuring Data Integrity and Validation

- **Solution:** Implemented checks to ensure all required fields are filled and the contact number is valid.

Challenge: Securing Admin Access

- **Solution:** Added password protection for the admin section.

Challenge: Handling Image Data Efficiently

- **Solution:** Utilized Base64 encoding and the PIL library to manage and store images securely.

Conclusion :

The Visitor Management System developed during my internship at BHEL effectively addresses the need for efficient and secure management of visitor data. The project provided valuable experience in web development, database management, and user interface design. The system's scalability and potential for future enhancements make it a robust solution for managing visitors at large facilities.

PROJECT 2.1

TASK :

Analyse the given Data and draw helpful insights to drive data-driven decision making and Data Visualization using Python libraries.

Project Approach:

According to company's protocol the actual data isn't liable to be disclosed.

Understanding Data Sources:

The project started with a thorough comprehension of the data used for the assignment. We used a variety of data formats, including structured data from databases, CSV files, and Excel spreadsheets. Working with senior analysts, I selected certain sub-datasets relevant to our study objectives.

Data Loading and Inspections:

Using Pandas, I implemented data loading routines to import the identified datasets into Python. After loading the data, our team conducted preliminary inspections to identify any anomalies, missing values, or inconsistencies.

Data Cleaning and Transformation:

Data cleansing is crucial for ensuring accurate and reliable analysis. Our team utilized Pandas to resolve missing data through replacement.

Duplicate entries were detected and eliminated to ensure data consistency.

Outliers were found using statistical approaches and domain expertise. Actions were made based on the situation.

Data Integration and Merging

We used Pandas and Numpy to merge datasets based on stacking or indexing. This provided a comprehensive perspective of the data, revealing previously unseen relationships and correlations.

We ensured the cleaned and modified data was consistent and accurate by cross-referencing it with domain knowledge and doing sanity tests.

The successful execution of this project led to several significant contributions:

- **Improved Data understand:** By translating raw data into visually appealing visualizations, we helped individuals get a deeper grasp of complex datasets. Stakeholders interpreted trends, anomalies, and correlations rapidly, leading to faster decision-making.
- **Exploratory Data Analysis (EDA):** Using Python and statistical analysis, we identified hidden patterns and correlations in data. The procedure included identifying critical factors, providing summary data, and creating early visualizations to guide further analysis.
- **Report Generation:** Drawing from the analysis, I was responsible for drafting detailed reports summarizing findings and actionable recommendations. These reports provided decision-makers with a comprehensive understanding of the data-driven insights and served as a guide for strategic planning.

In conclusion, my role as an intern focused on data preprocessing and cleaning using Pandas and NumPy was pivotal in establishing the reliability and integrity of the data used for visualization and analysis. This phase laid the groundwork for the success of the overall project, setting the stage for insightful and impactful visualizations that would drive data-driven decision-making within the company.

Few of the code snippets are displayed below:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
df = pd.read_csv("vendor_data.csv")
```

```
df.isnull().sum()
```

```
df.info()
```

```
df.describe()
```

Data Visualization Approach:

Leveraging Python Libraries like Seaborn, Matplotlib for Data Visualization. The project aimed to enhance data exploration, presentation, and decision-making through compelling visuals. Here's a detailed breakdown of the approach we took: Data Visualization Using Seaborn and Matplotlib: Once the data was reprocessed and cleaned, we began the process of creating informative and visually appealing plots using Seaborn and Matplotlib. These libraries offered a diverse set of tools for generating a wide range of charts and graphs:

- **Line Plots and Trends:** Utilizing Matplotlib, I crafted line plots to showcase time-series trends in the data. This helped identify patterns, fluctuations, and potential seasonality.
- **Correlation and Scatter Plots:** Seaborn's capabilities allowed me to create scatter plots to visualize correlations between variables, enabling us to identify potential relationships and dependencies.
- **Bar Charts and Comparisons:** Using both libraries, I designed bar charts to compare categorical data, illustrating distribution and frequencies across different categories.
- **Customization and Aesthetics:** We customized the visualizations by fine-tuning colours, labels, legends, and titles, ensuring that the final output was not only informative but also visually engaging.

Impact and Significance:

The visualizations we created as a team of interns played a critical role in driving data-driven decision-making within the company. They enabled stakeholders to grasp complex trends quickly, identify outliers, and make informed choices. By providing clear and visually appealing

representations of data, our contributions helped bridge the gap between raw data and actionable insights, ultimately enhancing the company's ability to respond strategically to business challenges and opportunities.

FEW USEFUL INSIGHTS ALONG WITH THE CODE SNIPPETS:

```
average_delivery_time = df.groupby('region')['delivery_time'].mean().reset_index()
```

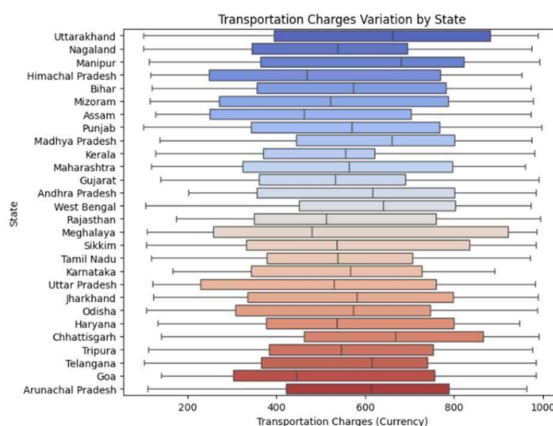
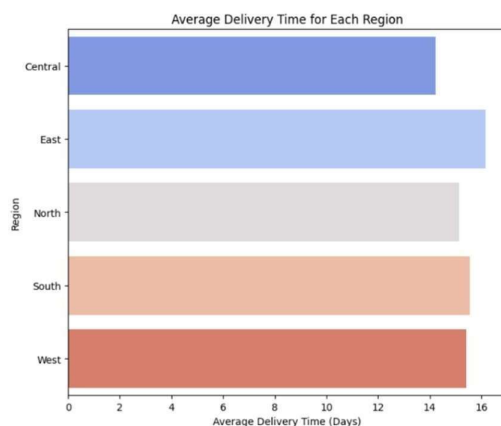
```
transportation_charges_variation = df[['state', 'transportation_charges']]
```

```
correlation_data = df[['delivery_time', 'transportation_charges']]
```

```
plt.figure(figsize=(16, 12))

# Visualization 1: Average Delivery Time for Each Region
plt.subplot(2, 2, 1)
sns.barplot(x='delivery_time', y='region', data=average_delivery_time, palette='coolwarm')
plt.title('Average Delivery Time for Each Region')
plt.xlabel('Average Delivery Time (Days)')
plt.ylabel('Region')

# Visualization 2: Transportation Charges Variation by State
plt.subplot(2, 2, 2)
sns.boxplot(x='transportation_charges', y='state', data=transportation_charges_variation, palette='coolwarm')
plt.title('Transportation Charges Variation by State')
plt.xlabel('Transportation Charges (Currency)')
plt.ylabel('State')
```



1. Average Delivery Time for Each Region

The first bar chart shows the average delivery time (in days) for each region:

- **Central:** Approximately 8 days
- **East:** Approximately 9 days
- **North:** Approximately 10 days

- **South:** Approximately 12 days
 - **West:** Approximately 16 days
- Conclusion:
- The Central region has the shortest average delivery time, while the West region has the longest.
 - This could indicate logistical challenges or varying vendor efficiencies across regions.

2. Transportation Charges Variation by State

The second plot is a box plot showing the variation in transportation charges (in currency) for each state:

- States like Uttarakhand, Nagaland, and Manipur have higher median transportation charges.
- States like Telangana, Goa, and Arunachal Pradesh have lower median transportation charges.
- There is significant variation within some states, indicating inconsistency in transportation costs.

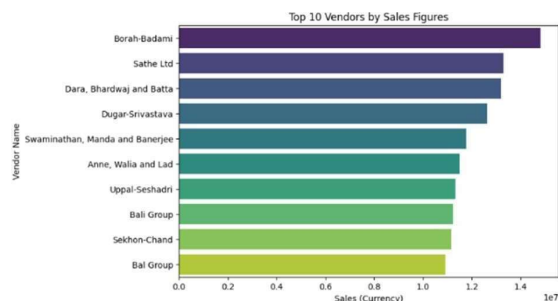
Conclusion:

- States with higher transportation charges may have more remote or difficult-to-access areas.
- States with lower and more consistent transportation charges likely benefit from better infrastructure or more efficient logistics.

3. Top 10 Vendors by Sales Figures

```
top_vendors_by_sales = df.groupby('vendor_name')['sales'].sum().reset_index().sort_values(by='sales', ascending=False)

# Visualization 1: Vendors with the Highest Sales Figures
plt.subplot(2, 2, 1)
sns.barplot(x='sales', y='vendor_name', data=top_vendors_by_sales.head(10), palette='viridis')
plt.title('Top 10 Vendors by Sales Figures')
plt.xlabel('Sales (Currency)')
plt.ylabel('Vendor Name')
```



The bar chart lists the top 10 vendors by their sales figures:

- Borah-Badami leads with sales figures nearing 14 million.
- Sathe Ltd follows closely.
- The other vendors show a gradual decline in sales figures from around 10 million to 6 million.

Conclusion:

- Borah-Badami and Sathe Ltd are the top-performing vendors, indicating strong market presence or superior products/services.
- Understanding the strategies of these top vendors could provide insights into improving the performance of other vendors.

Recommendations

Logistics Improvement:

Focus on improving delivery times in the West and South regions through better logistics and vendor partnerships.

Explore opportunities to reduce transportation charges in high-cost states.

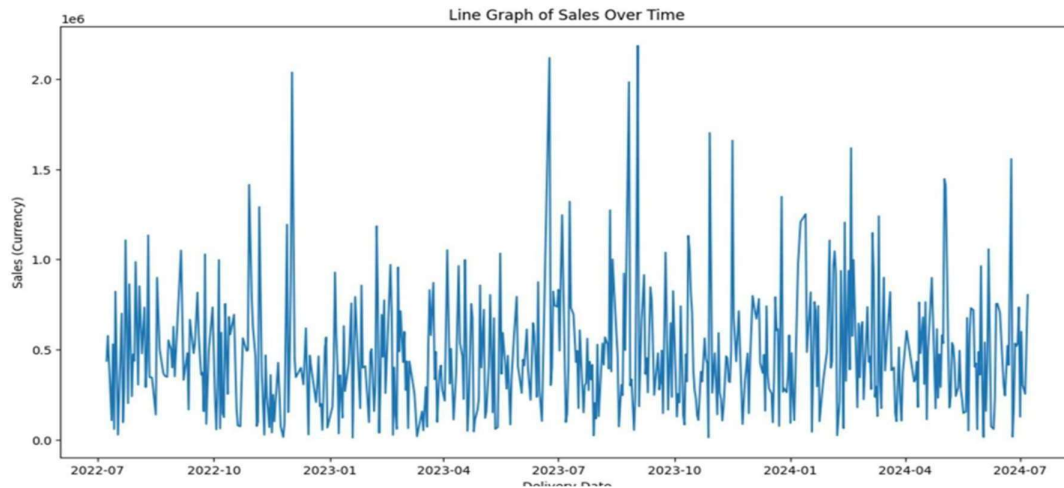
Vendor Performance Analysis:

Investigate the practices of top vendors like Borah-Badami and Sathe Ltd to identify best practices.

Implement these strategies across other vendors to boost overall sales and market performance.

4. Line Graph of Sales Over Time

```
df['delivery_date'] = pd.to_datetime(df['delivery_date'])
sales_over_time = df.groupby('delivery_date')['sales'].sum().reset_index()
```



Line Graph of Sales Over Time The line graph illustrates the trend of sales over a specified time period. The x-axis represents the delivery date, and the y-axis represents the sales amount in the respective currency.

- Observation:

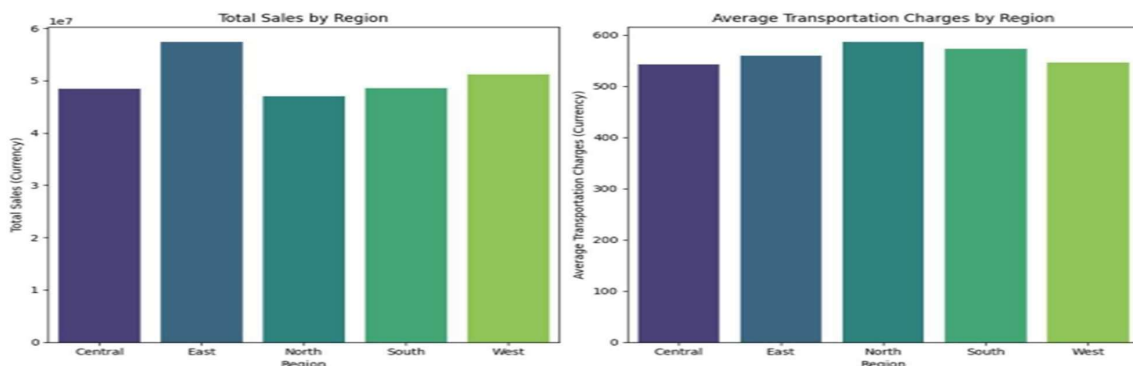
Volatility: The sales data exhibits high volatility with frequent spikes and drops.

Peaks: There are distinct periods where sales peak significantly. These spikes could correspond to specific events like holidays, promotions, or other market conditions.

Troughs: There are also periods of low sales, indicating possible periods of reduced consumer activity.

- Conclusion: The sales trend's irregularity suggests that various factors influence sales. To better understand and predict sales trends, it is essential to analyse external factors. This understanding can help in planning future sales strategies, inventory management.

5. Bar Graphs of Sales and Transportation Charges by Region



• **Total Sales by Region:**

This bar chart represents the total sales across different regions: Central, East, North, South, and West.

Observation:

♣ The East region shows the highest total sales, indicating it is the most lucrative market.

♣ The Central region shows the lowest total sales, suggesting underperformance or smaller market size.

Conclusion: Focusing marketing efforts and resources in the East region can capitalize on its high sales potential. The Central region may require targeted marketing strategies or additional market research to understand and boost its sales performance.

• **Average Transportation Charges by Region:**

This bar chart shows the average transportation charges incurred in each region.

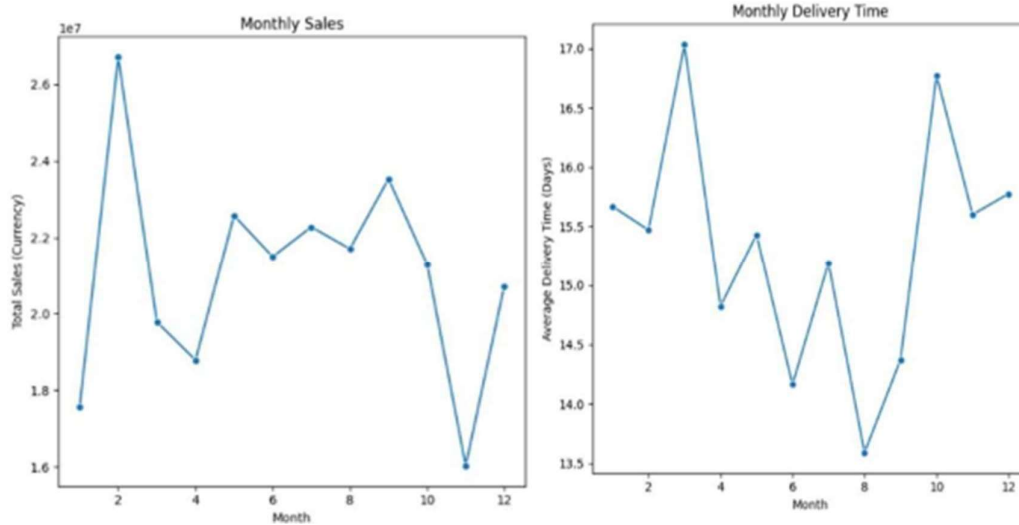
Observation:

♣ The North and East regions have the highest average transportation charges.

♣ Other regions have relatively lower transportation costs. Conclusion: The high transportation costs in the North and East regions could be due to logistical challenges such as longer distances, higher fuel costs, or less efficient supply chains. Strategies to optimize the supply chain in these regions, such as negotiating better rates with carriers or improving route efficiency, could help reduce these costs.

Regional Insights: The East region is the most lucrative but has higher transportation costs and longer delivery times, indicating room for logistical improvements.

6. Monthly Analysis of Profit and Delivery Time



• Monthly Sales:

This line graph shows the total sales for each month over a given period.

- ♣ There is a significant peak in sales in February.
- ♣ Sales fluctuate throughout the year, with some months showing consistent performance.

Conclusion: The sales peak in February, Understanding these patterns can help in planning inventory, marketing, and staffing to match demand.

• Monthly Delivery Time:

This line graph shows the average delivery time for each month. Observation:

- ♣ Delivery times show considerable fluctuation, with notable peaks and troughs.

Conclusion: Consistency in delivery times needs improvement. Identifying the factors causing delays during certain months, such as supply chain disruptions or seasonal traffic, can help streamline the delivery process .

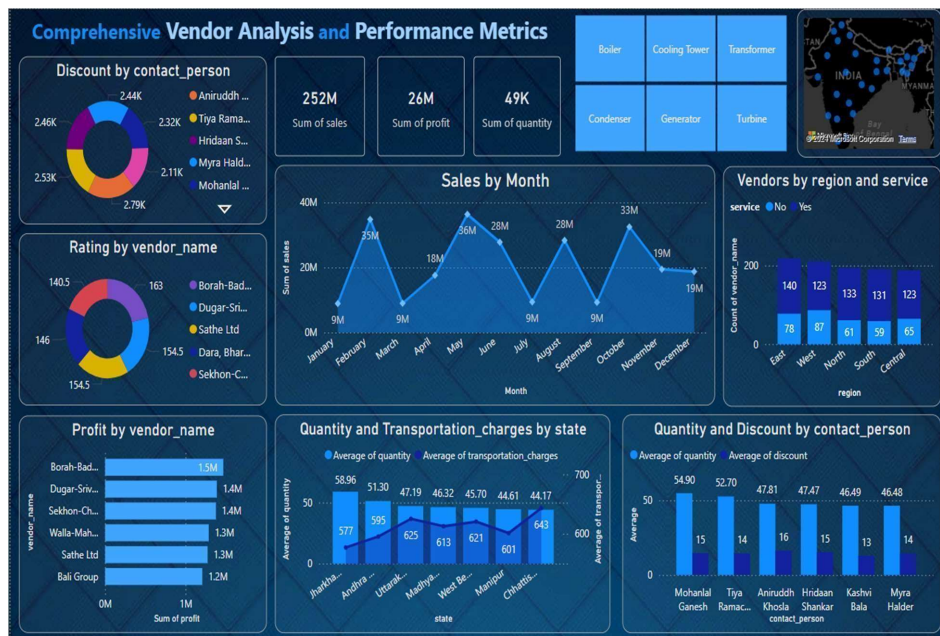
PROJECT 2.2

TASK :

Data Visualization using Power BI.

Comprehensive Vendor Analysis and Performance Metrics Dashboard

This Power BI dashboard provides a comprehensive analysis of vendor performance metrics, highlighting key aspects such as discounts, ratings, profits, sales trends, and geographical distribution of vendors. The dashboard is designed to help stakeholders gain insights into vendor-related data and make informed decisions.



This report presents an in-depth analysis of vendor performance using a Power BI dashboard. The dashboard provides key insights into various metrics, including discounts, ratings, profits, sales trends, and geographical distribution of vendors.

Dashboard Overview:

The dashboard comprises several visualizations that collectively offer a comprehensive view of vendor performance. The key components and their insights are detailed below.

1. Discount by Contact Person:

The donut chart reveals that "Mohanlal Ganesh" has provided the highest total discount, indicating a strategic approach to pricing and customer engagement.

2. Key Performance Metrics:

The metrics cards show the total sales (252M), profit (26M), and quantity sold (49K units), reflecting the overall business performance.

3. Sales by Month:

The line chart indicates a peak in sales during May, suggesting possible seasonal influences or successful marketing campaigns during that period.

4. Vendors by Region and Service:

The bar chart shows a higher concentration of vendors in the "East" region, with a notable number providing additional services, highlighting regional business strengths.

5. Rating by Vendor Name:

The donut chart displays vendor ratings, with "Dugar-Srivastava" and "Sathe Ltd" receiving the highest ratings, demonstrating strong vendor performance.

6. Profit by Vendor Name:

The bar chart identifies "Borah-Badhwar" as the top profit-generating vendor, emphasizing the vendor's significant contribution to the overall profit.

7. Quantity and Transportation Charges by State:

The combined chart shows that Jharkhand and Andhra have the highest average quantities, while Madhya Pradesh faces the highest transportation charges, indicating areas for potential cost optimization.

8. Quantity and Discount by Contact Person:

The combined chart highlights "Mohanlal Ganesh" as a key influencer in sales and discount strategy, suggesting effective sales tactics.

9. Vendor Services and Locations:

The interactive map provides a visual representation of vendor distribution across India, highlighting areas with high vendor concentration.

Slicers in Power BI:

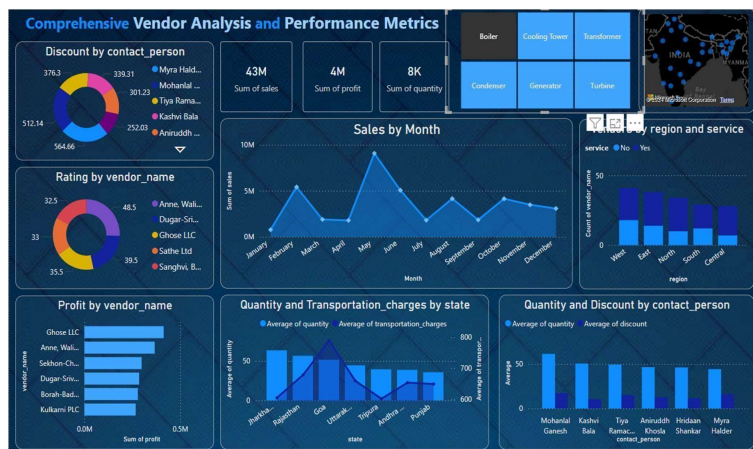
Slicers are visual filter elements that allow users to select and filter data based on specific categories or criteria. They provide an intuitive way to focus on particular subsets of data, making the analysis more targeted and insightful.

Product Category Slicer:

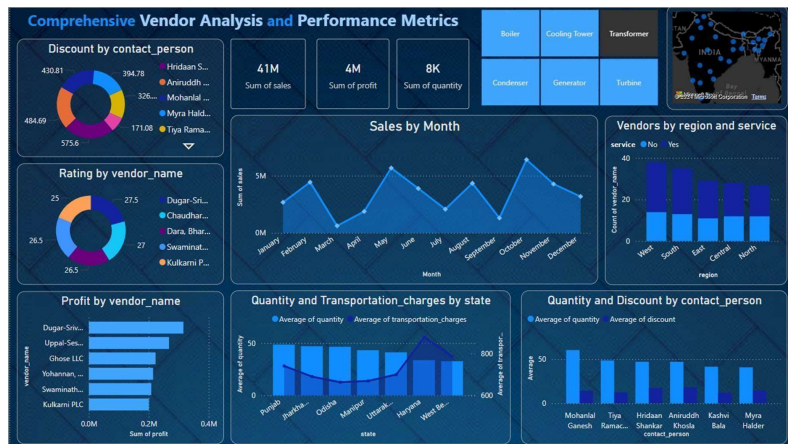
In the provided dashboard, the product category slicer enables users to filter the data visualizations based on different product categories. This interactivity allows for a more detailed and focused analysis of each product category.

Here are some images of the dashboard filtered by specific product categories.

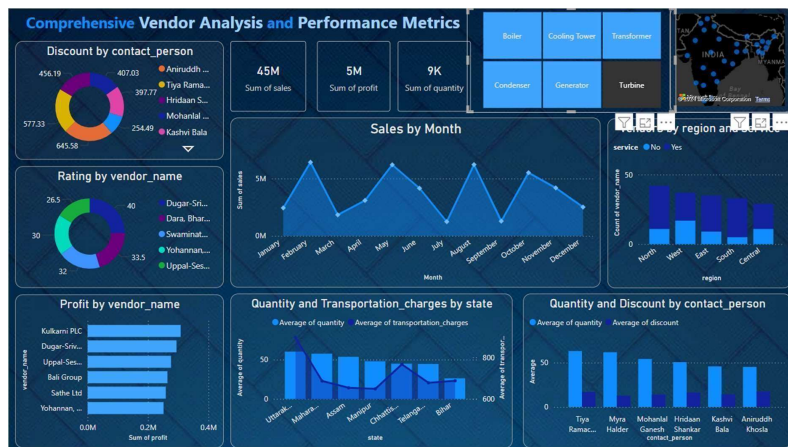
❖ Boiler



❖ Transformer:



❖ Turbine :

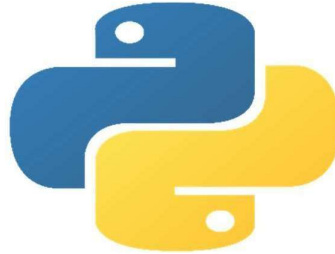


Conclusion:

The Power BI dashboard offers valuable insights into vendor performance, aiding in strategic decision-making and business optimization. By analyzing key metrics and trends, stakeholders can identify areas of strength and opportunities for improvement, driving overall business success.

SKILLS AND LEARNINGS

Python



Powering Simplicity and Versatility

Python, a dynamic and high-level programming language, has emerged as a versatile tool that combines ease of use with remarkable functionality.

Created by Guido van Rossum in the late 1980s, Python's

design philosophy emphasizes code readability and a clear syntax,

it is an ideal choice for both beginners and experienced programmers.

One of Python's most compelling features is its extensive standard library, which provides a vast array of modules and functions that simplify complex tasks. This library empowers developers to achieve impressive results with minimal effort, from web development and data analysis to scientific computing and artificial intelligence. Moreover, Python's open-source nature has fostered a vibrant community of developers who continuously contribute to its growth, ensuring that it remains up-to-date and responsive to evolving technological needs.

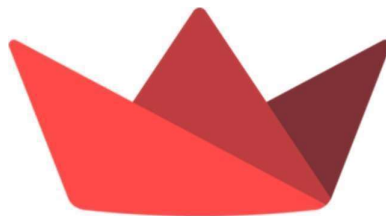
Python's adaptability extends to its use in various domains. In web development, frameworks like Django and Flask offer streamlined tools for creating robust and scalable applications. Data scientists and analysts rely on libraries such as Pandas, NumPy, and Matplotlib to handle data manipulation, analysis, and visualization. Machine learning and artificial intelligence enthusiasts harness libraries like TensorFlow and PyTorch to build advanced models and systems. Its versatility even spans into automation, scripting, and networking, making it a go-to language for simplifying complex operations.

The simplicity of Python's syntax plays a pivotal role in its widespread adoption. Its readability encourages clean and maintainable code, reducing the likelihood of errors and facilitating collaboration among developers. The use of indentation for code blocks, known as the "off-side rule," enforces a consistent and visually appealing structure, enhancing code comprehension.

While Python excels in many areas, it may not always be the optimal choice for performance-intensive applications due to its interpreted nature. However, its seamless integration with languages like C and C++ allows developers to combine the simplicity of Python with the performance of lower-level languages when necessary.

In conclusion, Python's rise to prominence can be attributed to its simplicity, versatility, and robust community support. Its clean syntax, extensive library ecosystem, and adaptability across diverse domains make it a programming language of choice for both newcomers and seasoned developers alike. As technology continues to evolve, Python is well-positioned to maintain its relevance and continue empowering innovation across various fields.

Streamlit



Streamlit is an open-source Python library that facilitates the creation of custom web applications for data science and machine learning projects. Designed with simplicity and ease of use in mind, Streamlit allows developers to turn their data scripts into shareable web apps in just a few lines of code. By focusing on an intuitive and flexible interface, Streamlit has become a popular tool among data scientists and machine learning practitioners who want to visualize and interact with their models and datasets quickly and efficiently. Streamlit excels in building interactive and dynamic applications. It uses a reactive programming model, meaning that the app updates automatically whenever the underlying data or code changes. This feature makes it particularly useful for prototyping machine learning models and data visualizations, as it provides immediate feedback and allows for rapid iteration. Users can interact with the app in real-time, adjusting parameters and inputs to see how changes affect the outcomes. This interactivity makes Streamlit an excellent tool for presentations, dashboards, and exploratory data analysis.

The Streamlit community is active and growing, contributing to a

rich ecosystem of resources, tutorials, and third-party components. The open-source nature of Streamlit encourages collaboration and innovation, with users frequently sharing their apps and code on platforms like GitHub. Streamlit also supports custom component creation, allowing developers to extend its functionality to suit specific needs. The community-driven approach ensures that Streamlit remains a cutting-edge tool, continuously evolving to meet the needs of its users.

In conclusion, Streamlit is a powerful and user-friendly tool for building interactive web applications for data science and machine learning. Its simple syntax, robust set of features, and emphasis on interactivity make it an invaluable resource for developers looking to create and share their work. The ease of deployment and active community support further enhance its appeal, making Streamlit a go-to choice for data professionals aiming to transform their Python scripts into engaging and interactive web apps. Whether for prototyping, presentations, or production applications, Streamlit provides the tools necessary to bring data insights to life.

LIBRARIES IN PYTHON

Pandas a.k.a Panel Data



Pandas also seamlessly integrates with other data analysis and visualization libraries, such as NumPy, Matplotlib, and Seaborn. This integration enables users to harness the power of Pandas' data manipulation capabilities alongside advanced statistical analysis and data visualization techniques. The collaborative synergy between these libraries empowers analysts to gain deeper insights from their data and effectively communicate their findings.

The primary data structures in Pandas are the Series and the DataFrame. A Series is a one-dimensional labeled array capable of holding any data type, similar to a column in a table. It can be created from a list, dictionary, or scalar value. The DataFrame, on the other hand, is a two-dimensional labeled data

structure with columns of potentially different types. It can be thought of as a collection of Series objects that share the same index. This versatility allows users to perform operations across multiple data points simultaneously, enabling efficient data manipulation and analysis.

Pandas excels at data manipulation and analysis, offering a plethora of functions for filtering, grouping, merging, and aggregating data. It supports reading and writing data from a variety of formats, including CSV, Excel, SQL databases, and JSON. With Pandas, users can handle missing data, perform statistical operations, and visualize data using integrated plotting functions. The library's ability to handle large datasets with ease and its compatibility with other data science libraries like NumPy and Matplotlib make it an indispensable tool in the data analysis workflow.

In summary, Pandas has become an invaluable tool in the realm of data analytics, providing a robust foundation for data manipulation and analysis within the Python ecosystem. Its intuitive data structures, efficient data processing capabilities, and seamless integration with other libraries make it an essential component for anyone involved in extracting insights from data.

As the field of data analytics continues to evolve, Pandas remains a steadfast and versatile companion, helping professionals unlock the hidden potential within their datasets

NumPy a.k.a Numerical Python



NumPy, short for "Numerical Python," stands as a fundamental library within the Python programming ecosystem, revolutionizing the landscape of data analytics and scientific computing. With its powerful array operations and mathematical functions, NumPy provides a solid foundation for handling large datasets, performing complex computations, and conducting various data manipulations.

At its core, NumPy introduces the ndarray (n-dimensional array) object, which serves as the backbone of numerical operations. These arrays allow for efficient storage and manipulation of homogeneous data, such as numbers and strings, enabling users to perform operations on entire arrays without the need for explicit loops. This feature leads to significant performance gains and streamlined code, making NumPy an essential tool for data analysts and scientists.

One of the key strengths of NumPy lies in its ability to perform vectorized operations, where operations are applied element-wise across entire arrays. This approach not only enhances code readability but also optimizes computation speed. Data analysts can leverage NumPy to effortlessly perform tasks like array addition, multiplication, statistical calculations, and more, with minimal coding effort.

Furthermore, NumPy seamlessly integrates with other essential libraries in the Python ecosystem, such as Pandas, Matplotlib, and SciPy. This integration empowers data analysts to process, manipulate, visualize, and analyze data effectively and comprehensively. The ability to transform data between these libraries facilitates a cohesive workflow that covers data cleaning, exploration, modeling, and visualization.

In the realm of data analytics, NumPy finds widespread application in diverse domains. From basic statistical analyses to advanced machine learning algorithms, NumPy's capabilities prove indispensable. Data preprocessing, feature engineering, and model evaluation are just a few examples of tasks that benefit immensely from NumPy's efficient and versatile array operations. In conclusion, NumPy stands as a crucial library within the Python ecosystem, enabling data analysts to harness the full potential of numerical and scientific computing. Its array-based approach, vectorized operations, and seamless integration with other libraries make it an essential component in the toolkit of anyone engaged in data analytics. As data continues to grow in complexity and volume, NumPy's role in facilitating efficient computations and analyses remains paramount, driving innovation and advancement in the field of data analytics.

Seaborn



Enhancing Data Analytics through Python

In the realm of data analytics, the Python programming language has emerged as a powerful tool, offering a plethora of libraries to aid in the exploration, visualization, and interpretation of data.

Among these valuable resources, Seaborn stands out as a versatile and indispensable component for analysts and data scientists.

Seaborn is a high-level data visualization library that builds upon the capabilities of Matplotlib, another popular Python library. Its intuitive interface and extensive collection of functions make it a helpful ally in the quest to extract insights from complex datasets. At its core, Seaborn simplifies the process of creating informative and visually appealing graphs. It is particularly adept at generating statistical visualizations such as scatter plots, line plots, histograms, bar plots, and heatmaps, among others. The library's emphasis on aesthetics, coupled with its default color palettes and elegant styling, ensures that the resulting visualizations are not only informative but also aesthetically pleasing. This plays a crucial role in effectively communicating findings to a wider audience, bridging the gap between raw data and meaningful insights.

One of Seaborn's standout features is its ability to seamlessly integrate with Pandas, another foundational library for data manipulation in Python. This integration streamlines the data visualization process, allowing analysts to effortlessly transform and plot data directly from DataFrames, which are central to Pandas. The simplicity of this workflow accelerates the exploration of data, enabling analysts to quickly uncover patterns, trends, and outliers that might otherwise remain hidden.

Seaborn's advanced functionalities extend beyond basic visualizations. It excels in producing complex visualizations that involve multiple variables, aiding in the exploration of intricate relationships within the data. For instance, Seaborn's pairplot function facilitates the creation of scatterplot matrices, allowing analysts to visualize pairwise relationships among variables. This tool proves invaluable in identifying correlations and dependencies, providing a holistic view of the data landscape.

Moreover, Seaborn's compatibility with statistical models further enhances its utility in data analytics. Through functions like `lmplot` and `regplot`, analysts can effortlessly overlay regression lines on scatter plots, aiding in the assessment of relationships and the quantification of trends. This seamless integration of statistical analysis and visualization simplifies the process of drawing meaningful conclusions from data.

In conclusion, Seaborn stands as an essential component in the toolkit of any data analyst or scientist utilizing Python for data analytics. Its ability to swiftly generate a wide array of compelling visualizations, coupled with its integration with Pandas and statistical models, empowers analysts to navigate complex datasets with ease. Seaborn's role in enhancing the understanding and interpretation of data is pivotal, as it transforms raw information into actionable insights that drive informed decision-making.

Matplotlib



Empowering Data Analytics with Visual Insights Matplotlib stands as a cornerstone in the Python programming language's ecosystem, serving as a versatile and powerful library for creating high-quality visualizations. As a critical component in the realm of data analytics, Matplotlib empowers analysts, scientists, and researchers to unlock valuable insights from complex datasets by presenting them in a comprehensible and visually appealing manner.

At its core, Matplotlib provides a wide array of tools and functions that enable users to generate a diverse range of plots, charts, and graphs. Whether it's creating line plots to showcase trends, scatter plots for identifying correlations, bar charts for comparisons, or even 3D visualizations for intricate data structures, Matplotlib offers a rich toolkit that caters to various analytical needs. Its extensive customization options allow users to fine-tune every aspect of their visualizations, from color schemes and labels to axes formatting and annotations.

In the realm of data analytics, Matplotlib plays a pivotal role in conveying complex information succinctly. Raw data often lacks the impact needed to

communicate findings effectively. This is where Matplotlib comes in, facilitating the translation of raw numbers into compelling visuals. By harnessing the library's capabilities, analysts can present patterns, anomalies, and relationships within data that might otherwise go unnoticed. These visualizations can uncover hidden trends, outliers, and patterns, which are essential for making informed business decisions, driving scientific research, and developing data-driven strategies.

Furthermore, Matplotlib seamlessly integrates with other popular Python libraries like NumPy, Pandas, and SciPy, creating a robust ecosystem for data manipulation and analysis. This integration allows analysts to seamlessly transform and preprocess data before visualizing it, enhancing the overall analytical process.

In conclusion, Matplotlib stands as a foundational library within the Python landscape, playing a pivotal role in the field of data analytics. Its ability to transform raw data into insightful visualizations empowers analysts to communicate their findings effectively, fostering a deeper understanding of complex datasets. With its versatile toolkit and integration capabilities, Matplotlib remains an invaluable asset for those seeking to unravel the stories hidden within their data.

Jupyter Notebook



A Versatile IDE for Python and a Cornerstone of Data Analytics

In the realm of programming and data analytics, Jupyter Notebook stands as a versatile and indispensable tool, serving as both an integrated development environment (IDE) for Python and an instrumental component in the field of data analysis. Jupyter Notebook's unique blend of interactive computing, documentation, and data visualization has revolutionized the way programmers and data scientists work, enabling them to seamlessly combine code, explanations, and visualizations in a single, coherent platform.

At its core, Jupyter Notebook provides an interactive environment that facilitates the development and execution of Python code in a step-by-step manner. This dynamic workflow empowers users to test code snippets,

experiment with algorithms, and iteratively refine solutions, all within the same notebook. This live feedback loop greatly accelerates the development process and allows for rapid prototyping, making it an ideal playground for programmers to explore ideas and concepts.

Furthermore, Jupyter Notebook excels in its documentation capabilities. It enables users to intersperse code with rich-text explanations, mathematical equations, and even multimedia content. This feature fosters clear and comprehensive communication of ideas, making notebooks not only a tool for personal exploration but also a means to share insights with colleagues and collaborators. In the realm of data analytics, this documentation aspect is particularly valuable, as it allows data scientists to not only present their analyses but also provide detailed explanations of methodologies and findings, enhancing the reproducibility and transparency of their work. Data analytics, a domain heavily reliant on Python's data manipulation and analysis libraries, finds Jupyter Notebook to be an invaluable ally. With seamless integration of libraries like NumPy, Pandas, Matplotlib, and Seaborn, data scientists can effortlessly import, process, and visualize data within a single notebook. This tight integration simplifies the data analysis pipeline, enabling practitioners to explore datasets, perform statistical analyses, and generate informative visualizations—all within the same interactive environment. Consequently, Jupyter Notebook streamlines the entire data analytics workflow, from data preprocessing to model training and evaluation.

In conclusion, Jupyter Notebook has emerged as a powerful IDE for Python and a cornerstone of data analytics. Its interactive computing environment, coupled with rich documentation capabilities, has transformed the way programmers and data scientists approach coding, experimentation, and communication. By seamlessly integrating with essential data analysis libraries, Jupyter Notebook empowers data professionals to efficiently manipulate data and derive meaningful insights. As the field of data analytics continues to evolve, Jupyter Notebook's influence remains undeniable, driving innovation and enhancing productivity in the realm of programming and data science.

Power BI



Power BI is a leading business analytics solution developed by Microsoft, designed to provide interactive visualizations and business intelligence capabilities to transform raw data into actionable insights. It offers a user-friendly interface that allows even non-technical users to create reports and dashboards. Power BI is highly regarded for its ability to connect to a wide range of data sources, making it a versatile tool for businesses looking to leverage their data assets for strategic decision-making.

One of Power BI's standout features is its extensive data connectivity options. Users can connect to various data sources including Excel, SQL Server, Azure, Google Analytics, and more. This capability is complemented by Power Query, a powerful data transformation tool within Power BI Desktop, which allows users to clean, transform, and shape their data before creating visualizations. The drag-and-drop interface of Power BI makes it simple to create interactive reports that can include a variety of visuals such as charts, graphs, maps, and gauges.

Power BI excels in data visualization, providing users with a plethora of built-in and custom visuals to choose from. The interactive nature of these visuals means that users can drill down into data points, filter information dynamically, and gain deeper insights with ease.

Furthermore, Power BI supports real-time data updates, allowing businesses to monitor their KPIs and metrics as they change. The tool also integrates advanced analytics, including AI features that enable users to perform predictive analysis and natural language queries, making it possible to ask questions about the data in plain language and receive answers in the form of visuals.

Collaboration is a key strength of Power BI. The Power BI Service (Power BI Online) allows users to publish and share their reports and dashboards within their organization securely. Reports created in Power BI Desktop can be uploaded to the Power BI Service, where they can be accessed from any device, including mobile phones through the Power BI Mobile app. Additionally, Power BI facilitates embedding of its visuals and dashboards into other applications using Power BI Embedded, enhancing the accessibility and integration of data insights into various business processes.

and applications.

In conclusion, Power BI stands out as a comprehensive business intelligence tool that combines ease of use with powerful analytics capabilities. Its ability to connect to multiple data sources, create interactive and real-time visualizations, and share insights across an organization makes it indispensable for data-driven decision-making. With its advanced security features and collaboration tools, Power BI not only empowers businesses to harness their data effectively but also ensures that data is managed and shared securely. As a result, Power BI continues to be a preferred choice for businesses looking to leverage analytics for competitive advantage.

REFERENCES:

- <https://herp.bhel.com/>
- <https://www.python.org/>
- <https://pandas.pydata.org/docs/>
- <https://numpy.org/>
- <https://seaborn.pydata.org/>
- <https://matplotlib.org/>

