**High-Level Design Document**

Mini Project

Contents

[1 Project Overview 2](#_Toc167321140)

[1.1 Brief introduction of the project 2](#_Toc167321141)

[1.2 Purpose of the project. 2](#_Toc167321142)

[2 Project Scope 2](#_Toc167321143)

[2.1 Streamlined Operations 2](#_Toc167321144)

[2.2 Enhanced Databases 3](#_Toc167321145)

[3 Technology Stack 3](#_Toc167321146)

[4 Architecture 4](#_Toc167321147)

[5 Data Scrapping 5](#_Toc167321148)

[6 User Data Transformation 5](#_Toc167321149)

[7 Data Visualization 6](#_Toc167321150)

# Project Overview

### 1.1 Brief introduction of the project

Our project aims to develop a pipeline to automate the process of data scraping, data transformation, and data visualization. The data collected is used in transforming Lenskart's operations by centralizing critical information.

### 1.2 Purpose of the project.

The project focuses on scraping data from Lenskart to gain insights into market trends and customer preferences for enhancing business strategies and personalized marketing efforts.

* **Market Analysis and Trend Identification:** Scraping data from Lenskart allows for the collection of comprehensive information about products, prices, customer reviews, and availability. This data can be analyzed to identify market trends, customer preferences, and competitive pricing strategies. Businesses and researchers can use this analysis to make informed decisions regarding product development, marketing strategies, and inventory management.
* **Personalized Customer Insights and Recommendations:** By gathering data on customer behavior, preferences, and purchase history from Lenskart, businesses can develop personalized marketing campaigns and product recommendations. This enables a more targeted approach to customer engagement, enhancing the customer experience and increasing the likelihood of repeat purchases and customer loyalty.

# Project Scope

Data engineering plays a vital role in transforming Lenskart’s operations by centralizing critical information and enhancing databases.

### 2.1 Streamlined Operations

* **Centralized Product Information:** Data engineering is used to design and implement a central product information management system. This system would house all the product details in a single location. This eliminates the need for scattered information across different departments or platforms, leading to:
  + Improved Accuracy: Consistent product information across all touchpoints (website, app, stores) reduces errors and confusion for both customers and employees.
  + Faster Updates: Centralized data allows for quick and efficient updates to product details, ensuring everyone has access to the latest information.
  + Enhanced Product Discovery: A well-organized product information management system facilitates better product categorization and search functionality, improving the customer experience.
* **Unified Pricing Details:** Unified Pricing Details: Data engineering can establish a centralized pricing engine. This engine would manage and distribute pricing information across all channels. This ensures:
* Real-time Price Consistency: Customers see the same prices regardless of whether they're browsing online or in a physical store.
* Promotional Efficiency: Managing promotions and discounts becomes easier with centralized pricing data, allowing for targeted campaigns and avoiding inconsistencies.
* Dynamic Pricing (Optional): The system could potentially support dynamic pricing strategies based on factors like competitor analysis or market trends (if aligned with Lenskart's business model).
  + - **Consolidated Customer Reviews:** Data engineers can build a system to collect and manage customer reviews from various sources (website, app, social media). This centralized system offers several benefits:
      * Improved Customer Insights: Analyzing consolidated reviews helps Lenskart understand customer sentiment towards specific products or services, guiding product development and improvement efforts.
      * Enhanced Customer Trust: Genuine customer reviews displayed prominently on product pages can build trust and credibility with potential buyers.
      * Better Customer Engagement: Lenskart can leverage reviews to personalize product recommendations or address customer concerns promptly.

### 2.2 Enhanced Databases

* + - * + **Customer Database:** Data engineers can design a robust Customer Relationship Management (CRM) system. This system would integrate data from various sources (purchase history, reviews, support interactions) to create a holistic customer profile. This allows Lenskart to:
        + Personalized Marketing: Targeted marketing campaigns based on customer preferences and purchase history can lead to increased sales and customer satisfaction.
        + Improved Customer Service: Support agents can access a complete customer history, enabling them to provide faster and more personalized service.
        + Loyalty Program Management: A centralized customer database facilitates the creation and management of loyalty programs, rewarding repeat customers.

# Technology Stack

The Mobile Application Platform for patient Instrument Tracking in Healthcare follows a robust and scalable architecture to ensure optimal performance, security, and usability. The architecture consists of the following components:

* + - * + **Python:** Python is chosen for its efficiency and extensive libraries, making it ideal for both web scraping and data transformation tasks:

**Web Scraping:** Lenskart APIs are utilized to extract data from Lenskart’s website.

**Data Transformation:** Library Pandas is employed to clean, manipulate, and transform the scraped data. Pandas provide flexible data structures and functions to manipulate structured data seamlessly.

* + - * + **Azure SQL Database, Azure Database Server:** These components ensure secure and scalable storage for the scraped data:
        + **Azure SQL Database:** A fully managed relational database service provided by Azure. It offers built-in high availability, automated backups, and security features. This service simplifies database management and scales according to demand, ensuring robust performance.
        + **Azure Database Server:** Provides the infrastructure for hosting the Azure SQL Database within the Azure cloud. This server ensures optimal performance and seamless integration with other Azure services, handling varying workloads efficiently.

**Azure Data Factory, Azure Function App:** These tools orchestrate and automate the data pipeline, ensuring a smooth data flow from scraping to storage and processing:

* + - * + **Azure Data Factory:** A cloud-based ETL (Extract, Transform, Load) service that orchestrates data movement and transformation. It allows the creation and scheduling of data-driven workflows, facilitating seamless data integration from various sources, and performing necessary transformations before loading the data into the database.
        + **Azure Function App:** A serverless compute service that enables running code in response to events or triggers, such as HTTP requests or scheduled timers. It is ideal for executing Python scripts for web scraping and data processing tasks, providing scalability and reducing the need for infrastructure management.

**SQL:** SQL is essential for querying and managing the database effectively.

**Power BI:** Power BI is used for data visualization, transforming data insights into an accessible and interactive format.

# Architecture

This stage involves designing a framework for storing, managing, and processing the scraped data. This stage defines how the data will be organized, accessed, and analyzed.

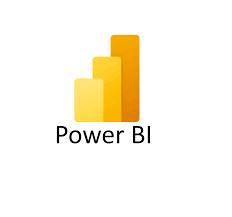
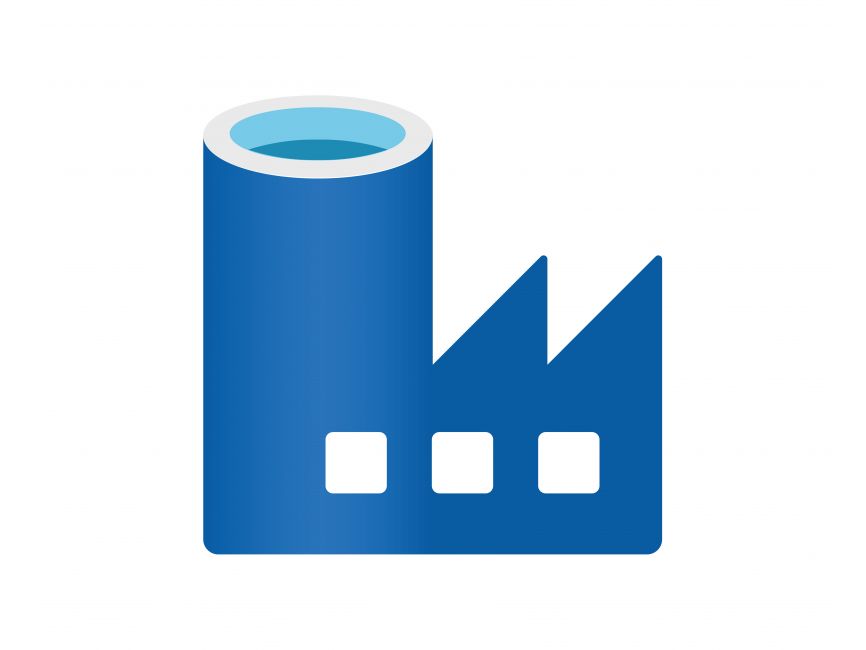
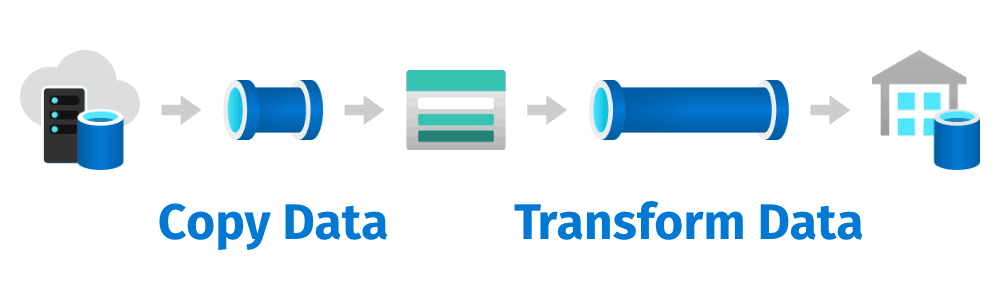
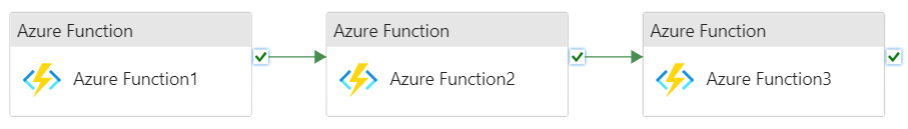
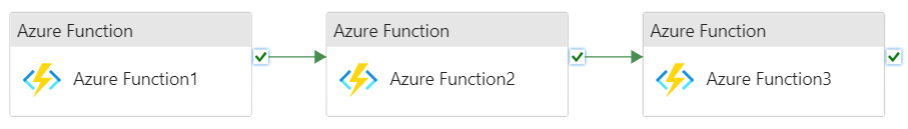
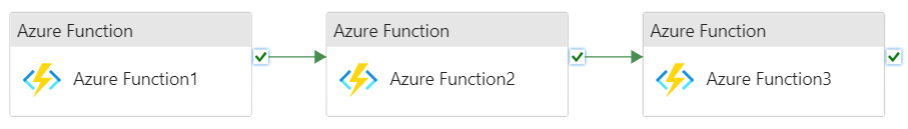
A well-designed architecture ensures efficient data management, simplifies access for analysis, and facilitates scalability as data volume grows.

In the architecture, Azure Data Factory (ADF) serves as the orchestration tool to create an automated pipeline that efficiently manages the end-to-end data workflow. The process begins with an Azure Function App, a serverless compute service, which is responsible for scraping data from the Lenskart website. This Function App is triggered within the ADF pipeline, either on a schedule or in response to specific events, ensuring that data extraction is automated and can scale based on demand.

Once the data is scraped, the Azure Function App also handles the transformation of this raw data. Using Python, the data is cleaned, structured, and formatted to ensure consistency and usability. These transformations might include filtering out irrelevant information, normalizing data formats, and performing calculations or aggregations as needed.

After the data is transformed, it is then stored in an Azure SQL Database. The Azure SQL Database, hosted on an Azure SQL Server, provides a reliable and scalable relational database solution. This setup ensures that the data is stored securely with high availability and automated backups, facilitating easy access and management. The integration between Azure Data Factory and the Azure SQL Database ensures that the entire process from data scraping to storage is streamlined and efficient.

Power BI can then be used to connect to the Azure SQL Database, allowing for real-time data visualization and analysis. By creating interactive dashboards and reports, Power BI helps stakeholders gain valuable insights from the data, driving informed business decisions and strategic planning.



# Data Scrapping

Data Scraping refers to the initial step of gathering data, often from online sources like websites. This is done using web scraping tools like selenium with python which automates the process of extracting specific information from webpages.

Ethical and legal aspects are crucial in data scraping, that includes:

Respect website terms of service and robots.txt restrictions.

Avoid overwhelming servers with excessive requests.

Only scrape publicly available data.

Data scraping using APIs and Python offers a powerful and efficient approach to data collection. Here's a breakdown of the key aspects:

**APIs (Application Programming Interfaces):**

* APIs act as intermediaries between your Python program and a web service that provides data.
* They offer a structured way to request and receive data in formats like JSON or XML.
* Many websites and services offer public APIs to access their data programmatically.

**Benefits of using APIs:**

* Structured Data: APIs deliver data in a well-defined format, eliminating the need for complex parsing from raw HTML (often required in web scraping).
* Efficiency: APIs are designed for programmatic access, making data retrieval faster and more reliable compared to scraping techniques.
* Reduced Legal Concerns: Scraping websites can sometimes raise legal issues. Using authorized APIs ensures compliance with the provider's terms of service.

**Data Scraping with Python and APIs:**

* Identify the API: Find a relevant API that provides the data you need. Many websites and services document their APIs publicly.
* Install Libraries: Use pip to install necessary libraries like requests for making API requests and potentially libraries for handling specific data formats (e.g., json for JSON data).
* API Authentication (if required): Some APIs require authentication to access data. This might involve obtaining an API key or using OAuth tokens.
* Parsing the Data: Depending on the format (JSON, XML), use appropriate libraries or methods to parse the data into a usable structure like Python dictionaries or lists.
* Data Storage and Processing: Store the extracted data in a structured format like CSV, JSON, or integrate it with data analysis tools like Pandas or NumPy for further processing and analysis.

# User Data Transformation

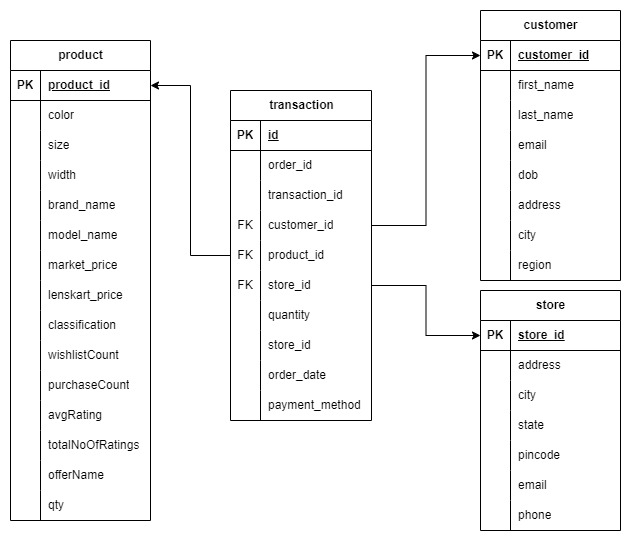
After scraping and storing the data, it's often necessary to transform it before analysis. Here's what data transformation involves:

* **Process:** This stage focuses on cleaning, manipulating, and preparing the data for meaningful insights. Common tasks include:
* Cleaning: Identifying and correcting errors, inconsistencies, or missing values.
* Formatting: Ensuring data is in a consistent format for analysis (e.g., standardizing date formats, converting units).
* Transformation: Deriving new data points or combining existing ones for deeper analysis (e.g., calculating average product ratings).
* **Importance:** Raw data might not be directly usable. Transformation ensures the data is accurate, complete, and ready for visualization and further analysis.

# Data Visualization

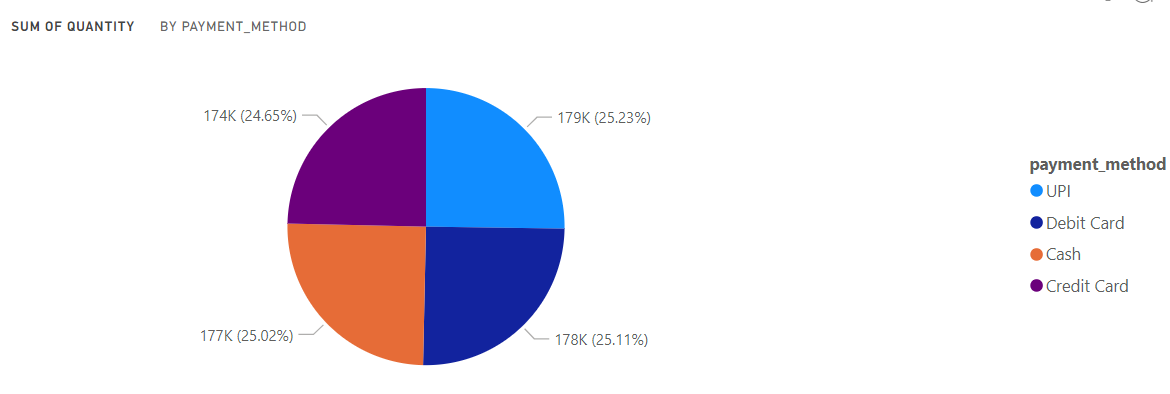
systematically examining and validating the software at various stages to identify defects, bugs, and any deviations from the expected behavior. By conducting thorough

This is the final stage where the transformed data is presented in a visual format for easier comprehension. Here's what data visualization involves:

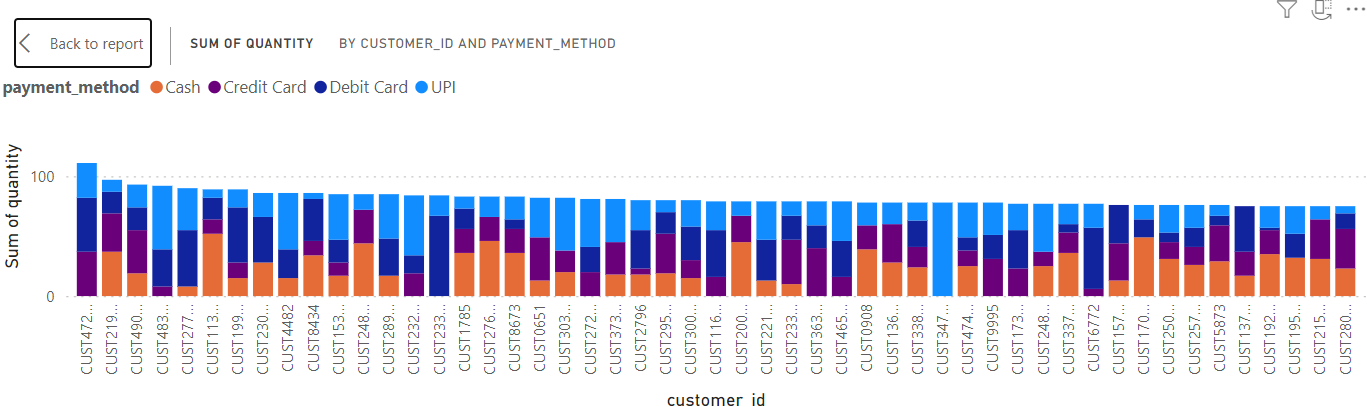
* Process: Creating charts, graphs, or other visual representations to communicate trends, patterns, and insights from the data. Common tools include:
  + Matplotlib (Python)
  + Tableau
  + Power BI
* Benefits: Data visualizations help to:
  + Identify trends and patterns that might be difficult to spot in raw data.
  + Communicate complex information clearly and concisely.
  + Make data analysis more engaging and impactful for audiences.

Schema of Database

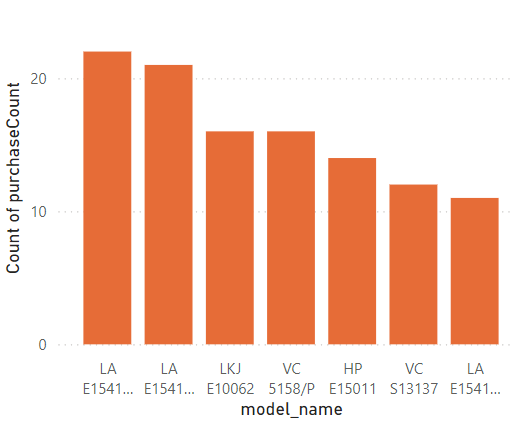
* In this project, Power BI tool is used for data visualization.

****

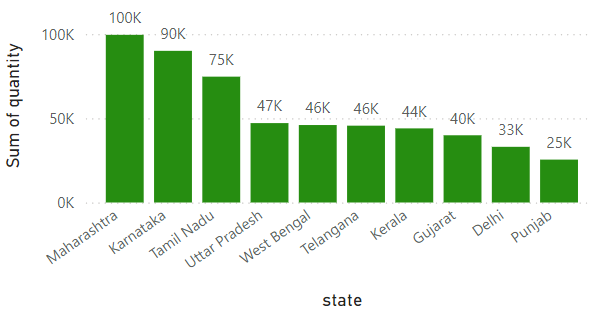
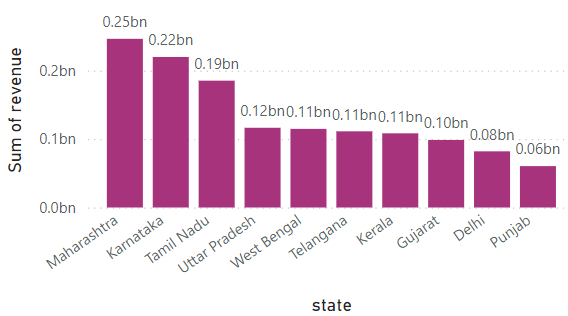
***Distribution of Payment Method***



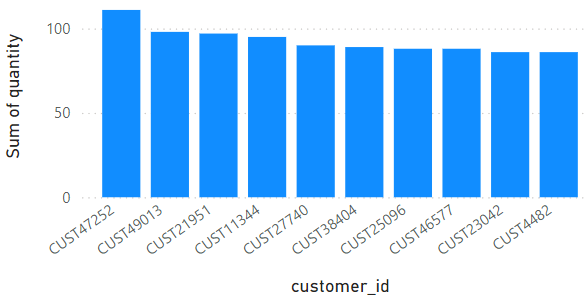
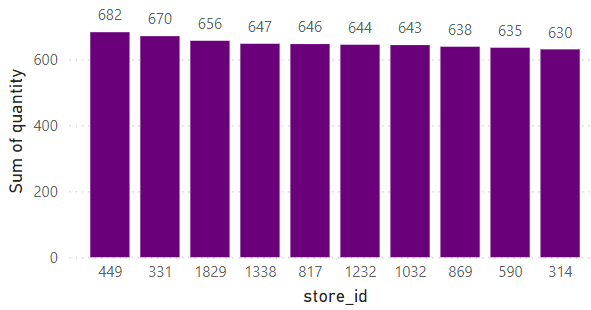
***Payment Method of Top 50 Customer***



***Top 7 selling models Heat Map of State-wise sell in India***



***Top 10 selling states based on item sold Top 10 selling states based on revenue***

******

***Top 10 Customers based on item purchased Top 10 Stores based on item sold***



***Top 15 Customer based on Revenue Top 10 Store based on Revenue***