

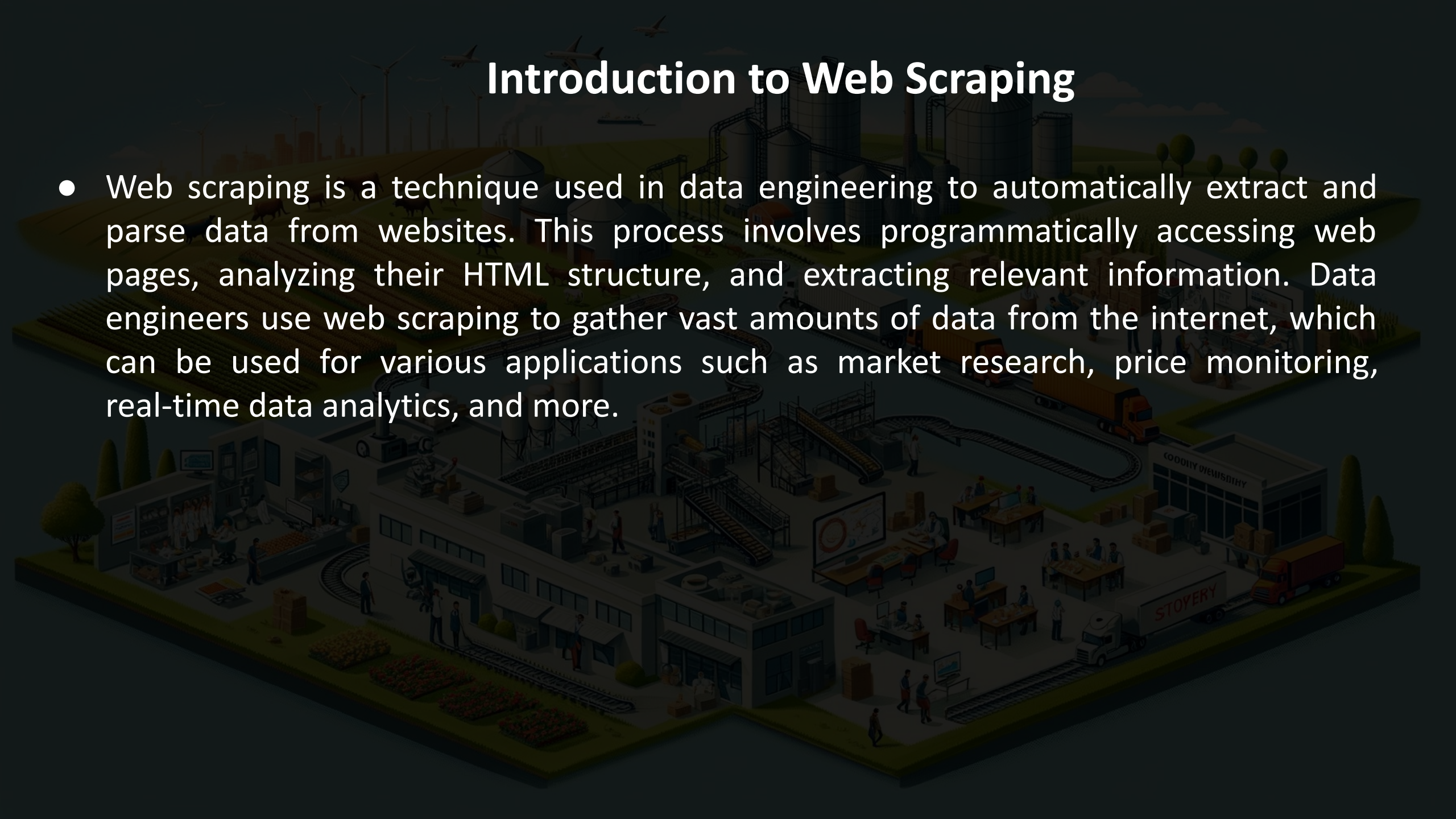
# Case Study– Web Scrapping on Dynamic Web Pages

(A Brief Introduction)



# Introduction to Web Scraping

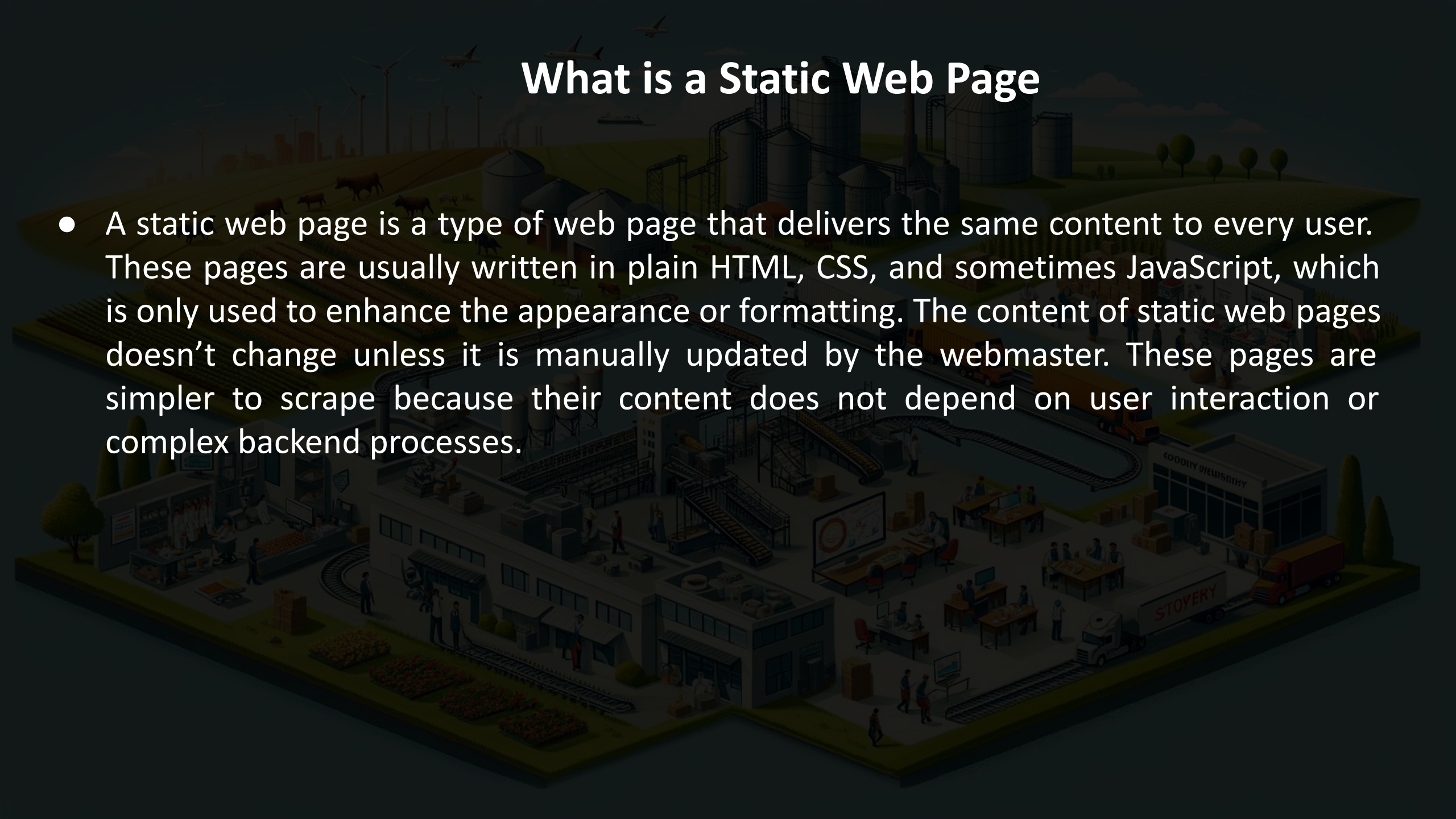
- Web scraping is a technique used in data engineering to automatically extract and parse data from websites. This process involves programmatically accessing web pages, analyzing their HTML structure, and extracting relevant information. Data engineers use web scraping to gather vast amounts of data from the internet, which can be used for various applications such as market research, price monitoring, real-time data analytics, and more.





# What is a Static Web Page

- A static web page is a type of web page that delivers the same content to every user. These pages are usually written in plain HTML, CSS, and sometimes JavaScript, which is only used to enhance the appearance or formatting. The content of static web pages doesn't change unless it is manually updated by the webmaster. These pages are simpler to scrape because their content does not depend on user interaction or complex backend processes.





# What is a Dynamic Web Page

- Contrary to static pages, dynamic web pages display content that can change based on user interaction, time of day, or other factors. They typically involve server-side processing languages like PHP, ASP.NET, or JavaScript frameworks like Angular or React. The content may be loaded asynchronously via APIs or AJAX calls. This makes them more challenging to scrape, as the data needs to be rendered or executed before it can be accessed.





# Differences Between Static and Dynamic Web Pages

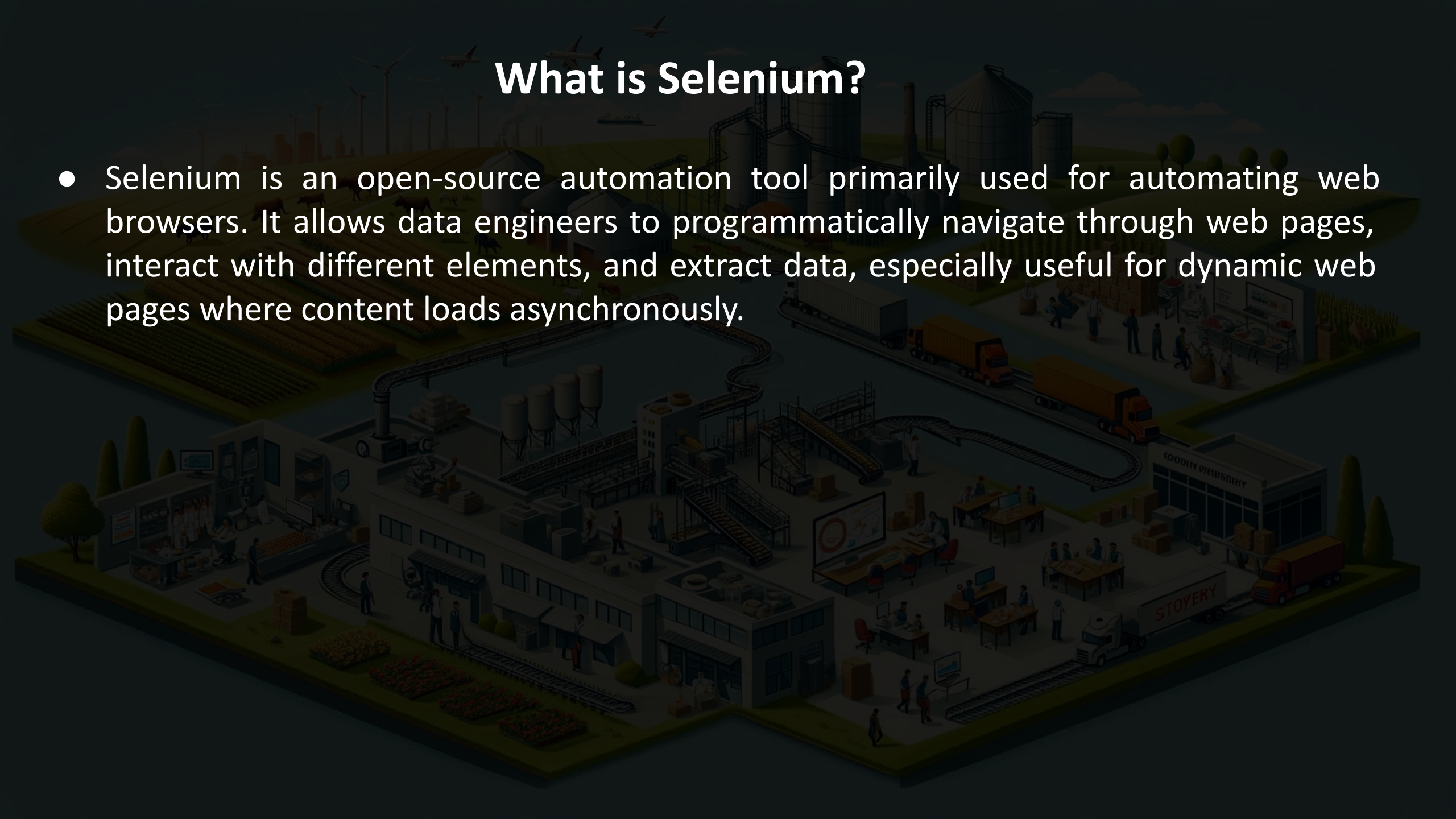
- Static Pages: The content does not change unless manually updated. They are easier to scrape as the data is readily available in the page's HTML.
- Dynamic Pages: Content changes dynamically and might depend on user inputs or other asynchronous events. Scraping these requires tools that can emulate a browser environment.

*For data engineers, understanding these differences is crucial as it determines the complexity of the scraping process and the tools needed for effective data extraction.*



# What is Selenium?

- Selenium is an open-source automation tool primarily used for automating web browsers. It allows data engineers to programmatically navigate through web pages, interact with different elements, and extract data, especially useful for dynamic web pages where content loads asynchronously.





# Importance of Selenium in Scraping Dynamic Pages

- Selenium is important for scraping dynamic web pages because it can interact with web elements that are loaded asynchronously or based on user interactions. It simulates real user behavior in a browser, enabling it to retrieve data that is not immediately available in the page's static HTML.
- This capability sets it apart from tools like Scrapy, which is highly efficient for scraping static content but does not handle JavaScript-driven dynamic content as effectively as Selenium.



# A Typical Workflow of a Web Scraping Project with Selenium

- A typical web scraping project using Selenium follows several key steps:
  - a. Planning and Requirement Analysis: Define the scope and objectives of the scraping project, including the data points to be extracted.
  - b. Tool Selection and Setup: Choose the appropriate tools (e.g., Selenium WebDriver, ChromeDriver) and set up the development environment.
  - c. Browser Automation: Use Selenium to programmatically control a web browser. This involves navigating to URLs, managing cookies, and possibly dealing with login forms.
  - d. Data Extraction: Interact with web page elements to extract data. This could involve locating elements by their HTML structure, handling dropdowns, checkboxes, and scrolling actions.
  - e. Data Processing: Clean and structure the extracted data, typically involving operations like string manipulation, date formatting, and the removal of unwanted data.
  - f. Storing Data: Save the processed data into a suitable format or database for further analysis or reporting.
  - g. Error Handling: Implement error handling mechanisms to manage issues like network failures, unexpected website changes, or data format changes.
  - h. Scheduling and Automation: Automate the scraping process to run at scheduled intervals, ensuring fresh data is always available.



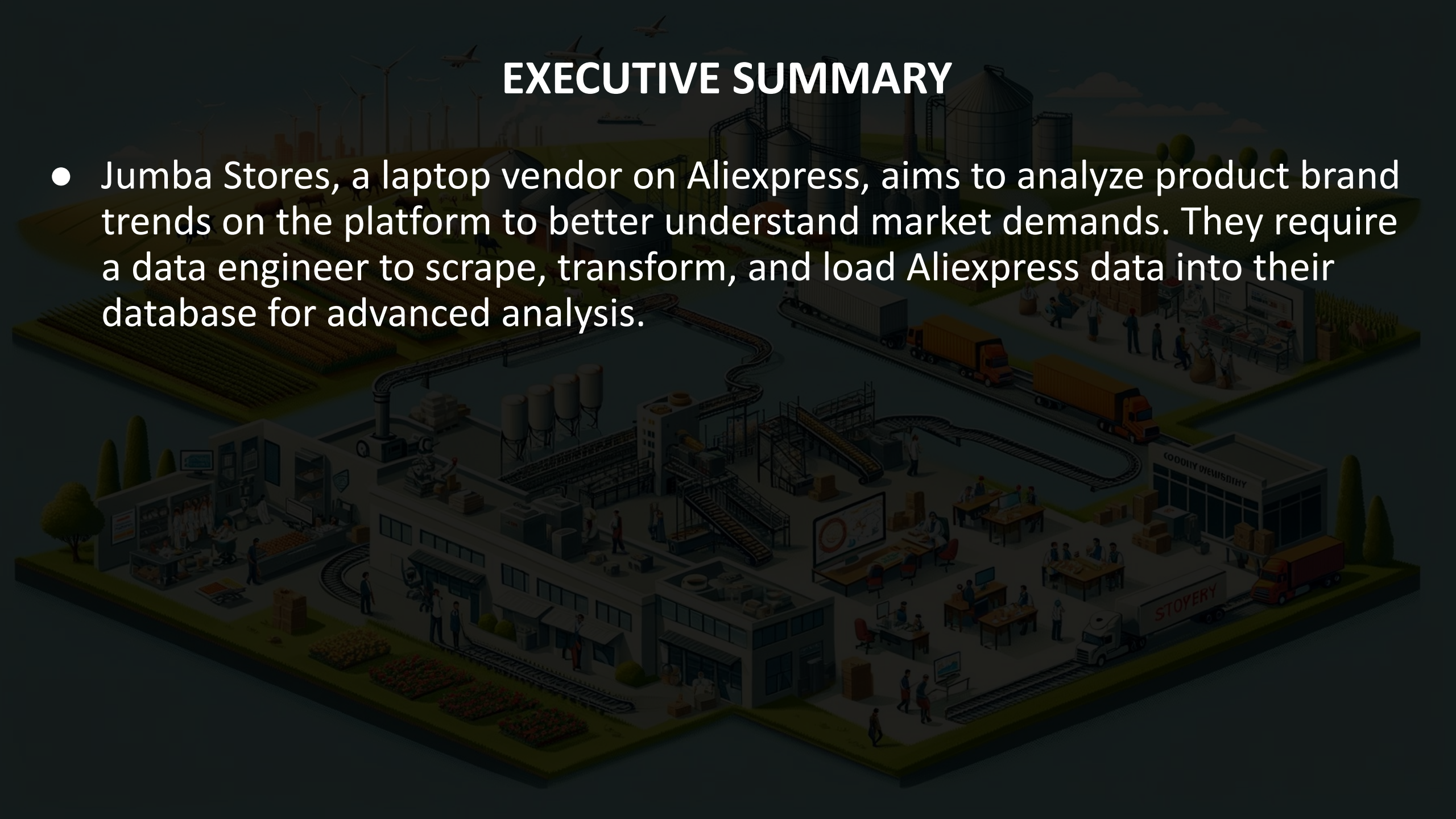


# Case Study For Jumba Stores



# EXECUTIVE SUMMARY

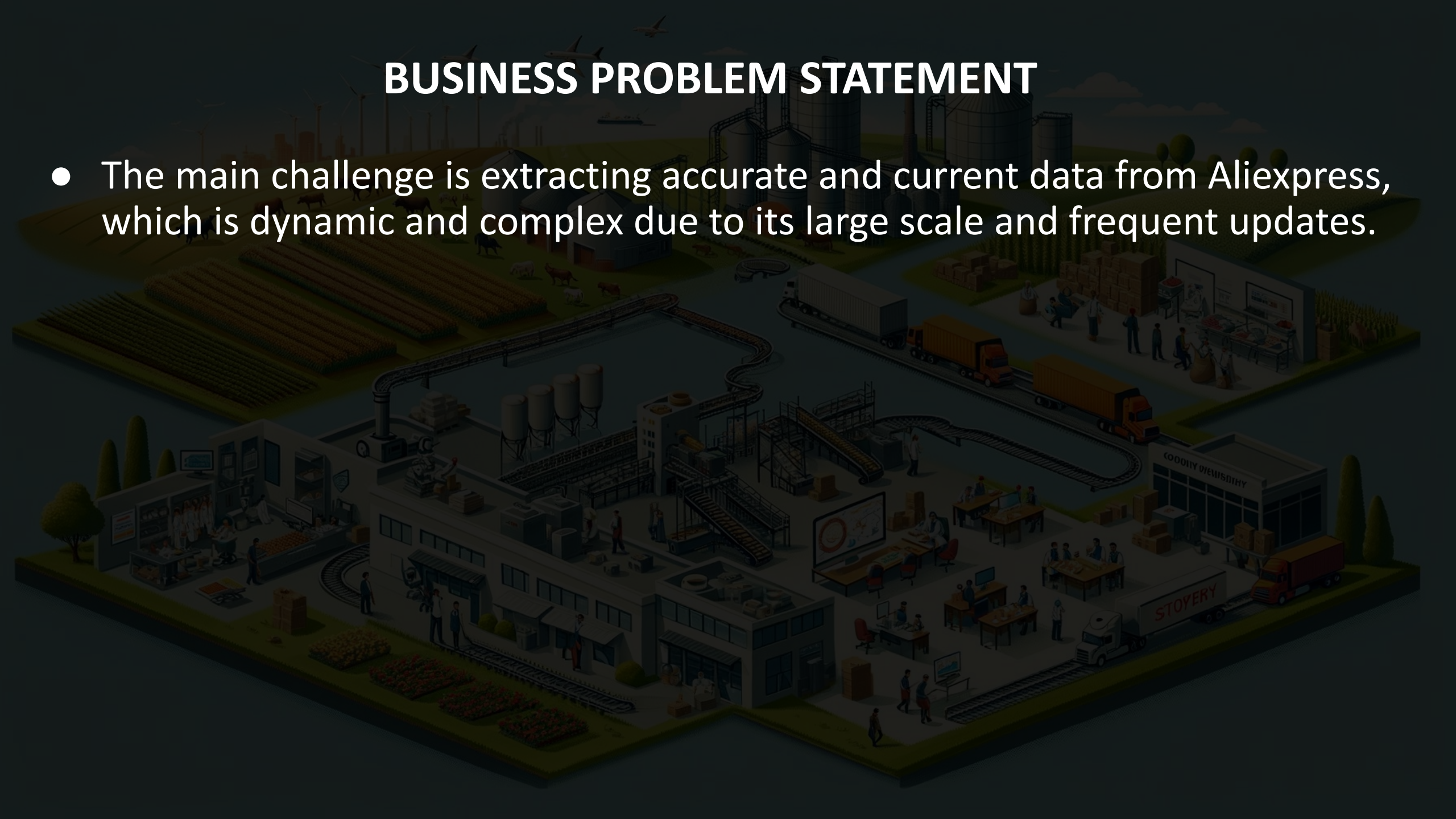
- Jumba Stores, a laptop vendor on Aliexpress, aims to analyze product brand trends on the platform to better understand market demands. They require a data engineer to scrape, transform, and load Aliexpress data into their database for advanced analysis.





# BUSINESS PROBLEM STATEMENT

- The main challenge is extracting accurate and current data from Aliexpress, which is dynamic and complex due to its large scale and frequent updates.







# Objectives

- To develop a scalable and robust system that periodically extracts data from Aliexpress, ensuring the data is clean, well-structured, and updated.






# Benefits

1. Market Insight: Understand real-time market trends and consumer preferences.
2. Strategic Decisions: Data-driven strategies for stock management and marketing.
3. Competitive Advantage: Staying ahead by leveraging up-to-date data.



# TECH STACK



**A. Python:**  
**We would leverage on libraries such as :**

- requests
- beautifulsoup
- pandas
- selenium
- sqlalchemy

**B. SQL**

**C. PostgreSQL**



# DATA SOURCE

A. Here is the link to the data source ⇒⇒⇒ LINK





# PROJECT SCOPE

- Data Extraction:  
Use Python with BeautifulSoup for static content and Selenium for dynamic content, extracting data from the laptop category into a pandas DataFrame.
- Data Cleaning and Transformation:  
Clean the data and normalize it to the Second Normal Form (2NF) or Third Normal Form (3NF) to ensure data integrity and efficiency.
- Data Loading:  
Employ sqlalchemy to load data into a PostgreSQL server, ensuring data is available for querying and analysis.

*This approach ensures Jumba Stores can effectively monitor market trends and make informed decisions based on comprehensive data analysis.*





**Now We Proceed To Coding!!!**

**Happy Coding!!!**



**GOODLUCK**