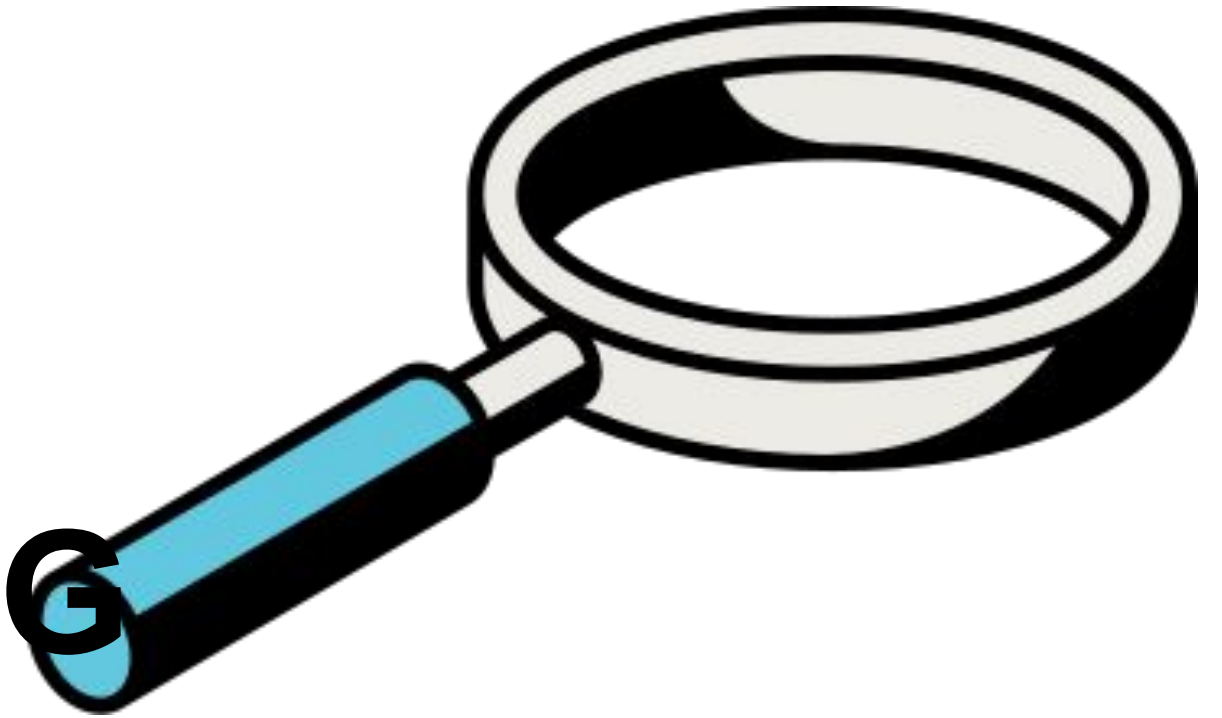
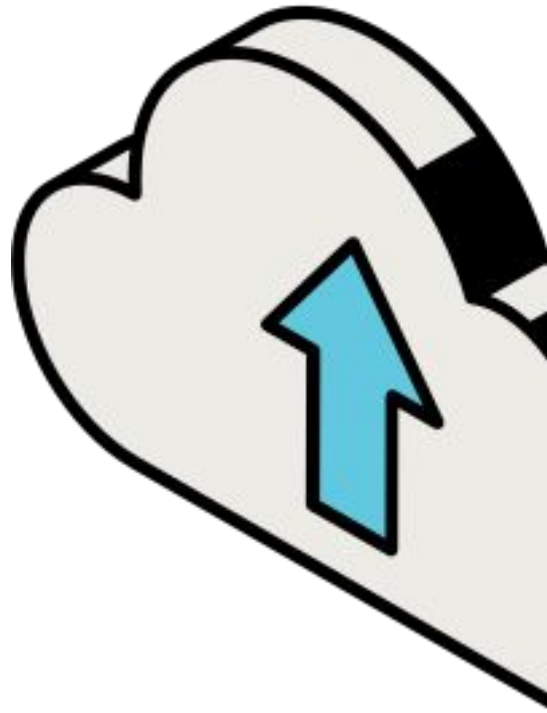


NAME : AAVANI RAJESH PERUMBESSI

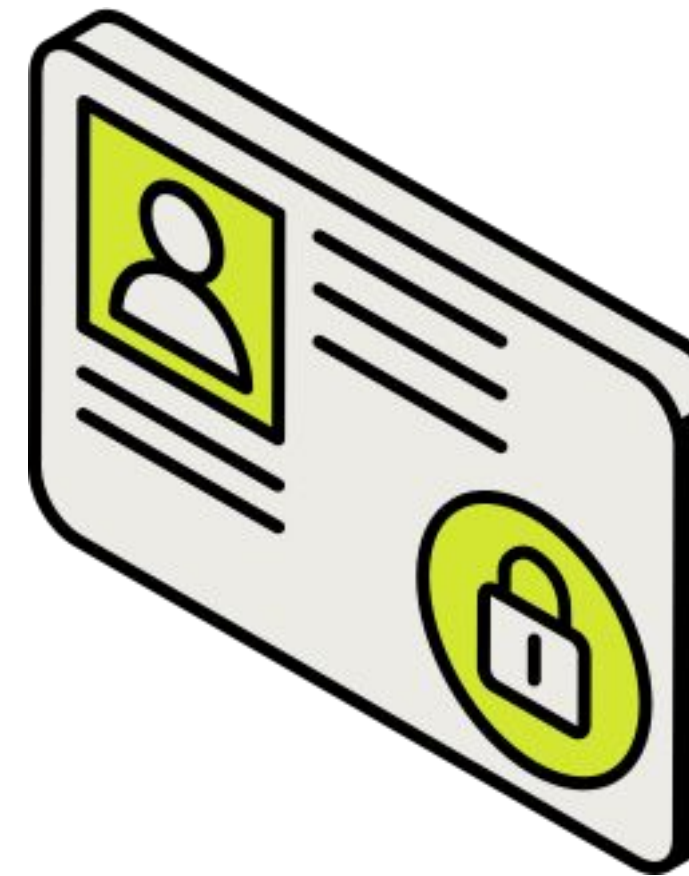
ROLL NO : 150096724059

COHORT : MARK ZUCKERBURG



WHAT IS DATA ?

Data are raw facts, figures, or information collected, observed, or measured or recorded, which serve as the basis for analysis and decisions. Data can take on many forms such as numbers, text, images, audio, videos, or symbols and are important in business, science, and technology fields. Data may be classified as structured, unstructured, or semi-structured, it may be qualitative or quantitative. The digital age is huge on data since huge amounts of data are collected in big data, analytics, and AI, all of which yield insights, predictions, and innovations that push the frontiers of progress and informed decisions.



TYPES OF DATA

Structured Data

Unstructured data

Semi Structured Data

Qualitative Data

Quantitative Data

STRUCTURED DATA

Structured data is highly organized and follows a predefined format, such as rows and columns in a table or database. It is easy to store, retrieve, and analyze using traditional tools like spreadsheets or relational databases. This type of data includes clearly defined fields and categories, making it suitable for applications like inventory management, customer databases, and financial records.

UNSTRUCTURED DATA

Unstructured data lacks a fixed format or organization, making it more challenging to process and analyze. It includes content like emails, social media posts, videos, images, and audio files. Since it doesn't fit neatly into rows and columns, specialized tools like artificial intelligence and machine learning algorithms are often required to extract meaningful insights from unstructured data.

SEMI - STRUCTURED DATA

Semi-structured data falls between structured and unstructured data, as it has some organization but doesn't conform to a strict structure. Examples include JSON and XML files or NoSQL databases. This type of data often contains metadata or tags, making it easier to parse and process than unstructured data while still offering flexibility in handling diverse types of information.

QUALITATIVE DATA

Qualitative data describes non-numerical characteristics or attributes. It provides detailed insights into qualities, behaviors, or perceptions that cannot be measured with numbers. Examples include customer feedback, survey responses, or descriptions of colors and textures.

This type of data is commonly used in social sciences, market research, and design studies to understand subjective experiences.

QUANTITATIVE DATA

Quantitative data consists of numerical values that can be measured, counted, or expressed mathematically. It is used for statistical analysis and helps identify trends, patterns, or correlations. Examples include sales figures, population statistics, and temperature readings. This type of data is essential in fields like economics, physics, and finance for precise and objective measurements.

WHAT IS INFORMATION ?

Information is processed, organized, or structured data that holds meaning and is useful for decision-making, analysis, or understanding. Unlike raw data, information is contextualized and interpreted, transforming it into a more valuable and actionable form. It is created by analyzing data, identifying patterns, or applying knowledge to extract insights. Information plays a vital role in various fields, such as business, science, and education, enabling individuals and organizations to solve problems, make informed choices, and achieve goals



TYPES OF INFORMATION

Conceptual information

Procedural information

Policy information

Stimulatory information

Empirical information

Quantitative Data

CONCEPTUAL INFORMATION

Conceptual information comes from ideas, theories, concepts, hypothesis and more. With conceptual information, an abstract idea is not always rooted in a scientific foundation and rather is the fundamental creation of beliefs, thoughts, philosophies and preferences. You can form or share conceptual information through comparison and reflection, creating philosophies that cannot be proven or seen.

PROCEDURAL INFORMATION

Procedural information, or imperative knowledge, is the method of how someone knows to do something and is used by performing a task.

You can refer to it as muscle memory since it is knowledge that is hard to explain and stored deeply in your mind.

POLICY INFORMATION

Policy information focuses on decision-making and the design, formation and selection of policies. It comprises laws, guidelines, regulations, rules and oversight for an organization, group of people or place. You can gain policy information through pictures, diagrams, descriptions and other visual, audio or written messages.

STIMULATORY INFORMATION

Stimulatory information is information that creates a response or stimulation amongst a person or group of people. Stimulation encourages the cause of activity and you can gain stimulatory information in a variety of ways, like in person through observation, through word-of-mouth communication or through outlets like the news.

EMPIRICAL INFORMATION

Empirical information means information gained through human senses, observation, experimentation and the testing of a hypothesis by establishing documentation of patterns or behavior.

DIRECTIVE INFORMATION

Directive and descriptive information is about providing directions to a person or group of people to achieve a particular result and outcome. You can use directive information with or without dictating the means to achieve the desired result. Directive information often comes in verbal or written form and can apply to leadership at work, in the military or government and with everyday experiences, like legal, life and safety matters.

N O C O D E

No-code is a development approach that enables users to create software applications without writing any code. It utilizes visual interfaces with drag-and-drop tools, pre-built templates, and customizable components, allowing non-technical users to build websites, mobile apps, or automation workflows. No-code platforms simplify the development process by abstracting the complexities of traditional programming, making it accessible to a wider audience, including business professionals and entrepreneurs.

N O C O D E

Few no code platforms are :

- Wix
- Webflow
- Bubble
- Airtable
- Zapier
- Adalo
- Glide
- OutSystems
- Carrd

L O W C O D E

Low-code is a software development approach that enables users to create applications with minimal hand-coding. It provides a visual development environment, where users can drag and drop pre-built components, and automate tasks with minimal technical skills.

However, unlike no-code, low-code platforms allow for some coding, offering greater customization and flexibility for more complex applications. Developers and business users can collaborate more effectively, as low-code platforms streamline the development process, enabling faster creation of web and mobile apps.

LOW CODE

Few low-code platforms are :

- OutSystems
- Mendix
- Microsoft PowerApps
- Appian
- Salesforce Lightning
- Betty Blocks
- Pega
- Zoho Creator
- Quick Base

LOW CODE

Platforms that allow developers to build applications with minimal hand-coding using drag-and-drop tools and reusable components.

Primarily developers, IT professionals, and tech-savvy users looking to accelerate development.

Suitable for building complex applications with advanced logic, integrations, and scalability.

NO CODE

Platforms enabling application creation without writing any code, relying entirely on visual interfaces.

Business users, citizen developers, or non-technical users aiming to create apps independently.

Best for simpler applications like forms, workflows, or basic internal tools.

LOW CODE

Platforms that allow developers to build applications with minimal hand-coding using drag-and-drop tools and reusable components.

Primarily developers, IT professionals, and tech-savvy users looking to accelerate development.

Suitable for building complex applications with advanced logic, integrations, and scalability.

NO CODE

Platforms enabling application creation without writing any code, relying entirely on visual interfaces.

Business users, citizen developers, or non-technical users aiming to create apps independently.

Best for simpler applications like forms, workflows, or basic internal tools.

LOW CODE

Offers flexibility for customization by enabling coding when required, making it adaptable to unique needs.

Requires a basic understanding of programming concepts and logic.

Supports integrations with various APIs, databases, and third-party tools for advanced functionalities.

NO CODE

Limited flexibility due to reliance on predefined templates and components.

Easy to learn and use, even for users with no technical background.

Limited integration options, often restricted to built-in connectors provided by the platform.

LOW CODE

Allows deeper customization through coding, enabling the creation of unique functionalities.

Ideal for enterprise-grade solutions, including complex web apps, mobile apps, and backend systems.

OutSystems, Mendix, Appian.

NO CODE

Customization is restricted to the tools and options provided by the platform.

Suitable for small-scale apps, prototypes, data collection forms, and dashboards.

Bubble, Airtable, Microsoft Power Apps.

ADVANTAGES OF LOW CODE

- **Faster Development:** Accelerates app creation with pre-built components and visual tools.
- **Reduced Coding Effort:** Minimizes manual coding, allowing developers to focus on logic.
- **Better Collaboration :** Enables business users and developers to work together seamlessly.
- **Cost Savings :** Lowers development costs by reducing the need for extensive coding.
- **Customization :** Allows custom coding for unique features while leveraging templates.
- **Scalability :** Supports enterprise-level applications with robust performance.

ADVANTAGES OF LOW CODE

- Simplified Maintenance : Eases updates and maintenance with reusable components and automation.
- Cross-Platform Support : Enables building apps for multiple platforms with minimal effort.
- Faster Time-to-Market : Reduces development time, helping businesses launch quickly.
- Empowers Non-Tech Users : Allows non-developers to create simple applications, reducing IT dependency.

DISADVANTAGES OF LOW CODE

- **Limited Customization:** While low-code platforms provide templates, they may lack the flexibility for complex custom features or advanced customization.
- **Scalability Issues:** Some low-code platforms may not handle large-scale applications effectively, especially when growing data or user demands.
- **Vendor Lock-In:** Applications built on low-code platforms may become dependent on that platform, making it difficult to migrate to another system.
- **Performance Limitations:** Pre-built components can sometimes lead to inefficient code, which may affect the performance of the application.
- **Security Concerns:** Low-code platforms may not offer the level of security and control needed for sensitive or highly regulated data.

DISADVANTAGES OF LOW CODE

- **Lack of Code Access:** Limited or no access to the underlying code can hinder troubleshooting and advanced customization.
- **Complexity in Large Projects:** While ideal for small projects, low-code platforms may become cumbersome for larger, enterprise-grade applications.
- **Integration Restrictions:** Some low-code platforms may have limited support for third-party integrations or APIs, restricting functionality.
- **Quality Control:** Rapid development may result in rushed solutions with potential bugs or technical debt.
- **Dependence on Platform Updates:** Features or changes in the platform might affect the application's functionality, and developers may not have control over updates.

BENIFITS OF LOW CODE

- Zero Coding Required: No need for technical skills, allowing non-developers to create applications easily.
- Rapid Development: Applications can be built quickly using drag-and-drop interfaces and pre-built templates.
- Cost-Effective: Reduces the need for hiring developers, making it affordable for small businesses or startups.
- Increased Accessibility: Empowers non-technical users, enabling them to create and modify apps without relying on IT teams.
- Faster Time-to-Market: Streamlined development process allows for quicker product launches and faster iterations.

BENIFITS OF LOW CODE

- Flexibility: Can build simple to moderately complex applications without coding barriers.
- Ease of Maintenance: Simple user interfaces allow for easy updates and modifications without technical expertise.
- No Technical Barriers: Reduces dependency on IT, making it easier for business teams to adapt and innovate.
- Collaboration: Enhances teamwork by allowing business users to directly contribute to app creation.
- Scalable for Small Projects: Ideal for small-scale projects, MVPs, or prototypes without extensive infrastructure needs.

ADVANTAGES OF NO CODE

- Ease of Use: No coding skills required, making it accessible to non-technical users.
- Faster Development: Enables rapid application creation with drag-and-drop interfaces.
- Cost-Effective: Reduces the need for expensive developers or custom coding.
- Empowers Business Users: Allows business professionals to build their own solutions.
- Reduced Dependency: Minimizes reliance on IT teams for simple applications.

ADVANTAGES OF NO CODE

- Quick Prototyping: Facilitates faster iteration and testing of ideas.
- Scalability: Some platforms support growing needs with additional features.
- Integration: Easy integration with third-party tools and services.
- Customization: Pre-built templates and features offer customization for various needs.
- Low Maintenance: Platforms manage hosting, security, and updates automatically.

DISADVANTAGES OF NO CODE

- Limited Customization: Limited flexibility for advanced or unique requirements.
- Scalability Issues: May not scale well for large or complex applications.
- Vendor Lock-in: You are often tied to the platform's ecosystem and limitations.
- Performance Limitations: No-code apps may not be as optimized as hand-coded solutions.
- Security Concerns: Potential risks due to lack of deep control over the underlying code.

DISADVANTAGES OF NO CODE

- Limited Functionality: Complex workflows or integrations may not be fully supported.
- Dependency on Platform: Any changes or outages by the platform provider affect your app.
- Learning Curve: Even no-code platforms require some time to master their tools.
- Limited Control Over Backend: Users can't fully control the backend infrastructure.
- Not Ideal for Large Teams: Not as suitable for large-scale enterprise-level applications.

FEATURES OF NO CODE

- Drag-and-Drop Interface: Easy-to-use visual builder for designing apps.
- Pre-Built Templates: Ready-made designs and workflows for quick deployment.
- Automations: Built-in automation tools for tasks like notifications and data processing.
- Integrations: Easy connections with external software, APIs, and services.
- Responsive Design: Automatically adapts apps for mobile and desktop views.

FEATURES OF NO CODE

- Cloud Hosting: No need for users to worry about infrastructure management.
- Collaboration Tools: Allows multiple users to work on the same project simultaneously.
- Version Control: Track changes and revert to previous versions of applications.
- Data Management: Built-in tools to manage databases and collections of information.
- User Authentication: Built-in security and user authentication systems.

5 V's of data

Volume

Variety

Vracity

Velocity

Value

VOLUME

- It refers to the massive amount of data generated every second.
- Includes data from sources like social media, IoT, e-commerce, and sensors.
- Measured in terabytes, petabytes, or even zettabytes.
- Requires scalable storage systems like Hadoop or cloud solutions.
- Challenges involve storing, accessing, and processing vast datasets.
- Data warehouses and data lakes are common solutions.
- Enables analysis on a larger scale for deeper insights.

VARIETY

- It represents the different forms and formats of data.
- Includes structured (databases), semi-structured (XML, JSON), and unstructured data (videos, emails).
- Sources range from social media posts to IoT devices and financial systems.
- Diverse formats make integration and analysis complex.
- Requires tools like NoSQL databases and data transformation platforms.
- Each data type provides unique value when analyzed correctly.

VELOCITY

- It refers to the speed at which data is created and processed.
- Examples include stock market transactions, streaming data, and real-time monitoring.
- Requires real-time or near real-time analytics for actionable decisions.
- Technologies like Apache Kafka and Spark Streaming manage high-speed data.
- Delay in processing velocity data can lead to missed opportunities.
- Critical for applications like fraud detection and live recommendations.

VERACITY

- It refers to the accuracy and trustworthiness of data.
- Data may be inconsistent, incomplete, or biased.
- Poor data quality can lead to incorrect conclusions or decisions.
- Requires processes like data cleaning, validation, and enrichment.
- Includes addressing noise, duplication, and outdated information.
- Confidence in data builds better predictive models and insights.

VALUE

- It represents the insights and benefits derived from analyzing data.
- Data alone is not useful unless it creates measurable impact.
- Helps optimize business processes, improve customer experiences, and drive innovation.
- Examples include cost reduction, revenue growth, and predictive analytics.
- Requires clear objectives and tools for meaningful analysis.

5 P's of data

Purpose

Programmability

Platform

Process

People

PURPOSE

- Data should serve a specific goal or objective.
- Clearly define why the data is being collected and analyzed.
- Aligns data initiatives with business outcomes.
- Examples: improving customer satisfaction, optimizing processes, or reducing costs.
- Ensures resources are focused on meaningful activities.
- Avoids collecting irrelevant or excessive data.
- Drives strategic decision-making through actionable insights

PROGRAMMABILITY

- It refers to the ability to automate and manipulate data workflows.
- Involves using programming languages like Python, R, or SQL.
- Enables custom data pipelines, analytics, and machine learning models.
- Facilitates real-time data processing and automation of repetitive tasks.
- Allows integration with APIs and other data systems for dynamic operations.
- Provides flexibility to adapt tools and algorithms to specific needs.

PLATFORM

- The infrastructure or ecosystem used to store, process, and analyze data.
- Examples include cloud platforms (AWS, Azure, Google Cloud) and on-premises systems.
- Supports data integration, analytics, and visualization.
- Must be scalable to handle growing data volumes and complexity.
- Ensures security, compliance, and accessibility of data.
- Provides tools for managing structured, semi-structured, and unstructured data.

PROCESS

- Refers to the steps taken to manage data from collection to insight generation.
- Includes data collection, cleaning, transformation, analysis, and reporting.
- Follows best practices like ETL (Extract, Transform, Load) for data pipelines.
- Ensures data quality and consistency through governance frameworks.
- Streamlines workflows to reduce inefficiencies and delays.
- Involves continuous monitoring and improvement of data processes.

PEOPLE

- It represents the human element in managing and utilizing data effectively.
- Includes data engineers, analysts, scientists, and business stakeholders.
- Requires a mix of technical expertise and domain knowledge.
- Fosters a data-driven culture within the organization.
- Encourages collaboration across departments for shared insights.
- Training and upskilling are essential to keep up with evolving technologies.

DATA

Data refers to raw facts, figures, or information that can be processed or analyzed to create insights

Data is used to store, analyze, or derive meaning from information.

Data is static in nature unless processed or manipulated.

CODE

Code is a set of instructions written in a programming language to instruct a computer on how to perform specific tasks or operations.

Code is used to create programs, websites, applications, or to automate tasks.

Code is dynamic, consisting of logic that is executed to perform actions.

DATA

Data can be structured (organized in a specific format), unstructured, or semi-structured.

Data can exist in numerical, textual, graphical, or multimedia forms such as numbers, words, images, or sounds.

Data can be modified through data processing or analysis, such as sorting or filtering.

CODE

Code has a well-defined syntax and structure based on the programming language.

Code is composed of human-readable text, written in programming languages such as Python, JavaScript, Java, or C++.

Code is modified to change the behavior of a program or fix issues by updating logic or functionality.

DATA

Data needs to be interpreted and processed to gain value. Raw data on its own does not convey meaning.

Data is used in analysis, research, business intelligence, and decision-making.

Example : A list of sales numbers, customer feedback, or survey responses

CODE

Code is interpreted by a computer or compiler to execute specific instructions.

Code is used to create software applications, websites, games, and automate processes.

Example : A script written to automate data collection, a program to calculate tax, or code for a website.

DATA

Data can be in various formats like CSV, JSON, XML, or databases.

Tools for working with data include Excel, databases, BI tools, or statistical software.

Data needs to be interpreted and processed to gain value. Raw data on its own does not convey meaning.

CODE

Code is written in text files (.py, .js, .java) and follows the specific syntax rules of a programming language.

Tools for working with code include IDEs (Integrated Development Environments), text editors, compilers, and debuggers.

Code is interpreted by a computer or compiler to execute specific instructions.

Thank you.

