ACHARYA INSTITUTE OF TECHNOLOGY

Affiliated to Visvesvaraya Technological University, Belagavi, Govt. of Karnataka.
Approved by AICTE, New Delhi

Department of Computer Science & Engineering
(Data Science)

PROJECT REPORT
ON
From BI to Big Data: Explain, Design & Defend

Subject Name: Big Data Analytics
Subject Code: BAD601

Submitted By:	Submitted To:
(Sanjana H – 1AY23CD052)	Ms. Surbhi

TABLE OF CONTENTS

About this project

1.Project Objective:

The primary goal of this project is to analyze the technical infrastructure required to power a global, data-heavy platform like Instagram. It explores the transition from Old School data management to the Modern Big Data era, specifically focusing on how massive amounts of user-generated content are transformed into a personalized experience.

2.Solving the "Three V's" Challenge

The project explores how Instagram's architecture was redesigned to solve the three core challenges of Big Data:

- **Volume:** Handling petabytes of data from billions of users without the system crashing.
- **Velocity:** Ensuring that when you "Like" a photo, the recommendation engine updates your feed in milliseconds, not hours.
- **Variety:** Managing a messy mix of structured data (phone numbers), semi-structured data (hashtags), and unstructured data (the actual pixels in a video).

3. Detailed Layer Breakdown

The analysis is categorized into four distinct functional tiers to show the journey of a single Like or Video View:

- **The Ingestion Layer (Data Sources):** We examine the move from manual entries to high-speed stream ingestion via Apache Kafka, which acts as a "buffer" for the millions of events happening every second.
- **The Computational Layer (Processing):** This section compares Batch Processing (the legacy way of doing things once a day) against Stream Processing (the modern way). We highlight Apache Spark as the engine that allows Instagram to be live.

- **The Persistence Layer (Storage):** We describe the shift from RDBMS (Relational Databases) to Distributed Storage. By using HDFS (Hadoop Distributed File System) and **S3**, data is spread across thousands of cheap computers. If one computer breaks, the data isn't lost this is known as Fault Tolerance.
- **The Intelligence Layer (Analytics):** This is the final stage where raw data becomes Gold. We explain how Predictive Models use historical data to guess what you want to see next, creating the addictive quality of the Explore Page.

4. Technical Impact on Business

Finally, the project concludes that the Hadoop-based Architecture is not just a technical choice, but a Business Strategy. It allows for:

- **Hyper-Targeted Advertising:** Making money by showing the right ad to the right person.
- **User Retention:** Keeping people on the app longer by predicting their interests.
- **Scalability:** Allowing Instagram to grow from 1 million to 2 billion users without needing to rewrite its entire code.

About the Tools and Technologies

1. The Data Ingestion Layer (Getting Data In)

- Apache Kafka: This is the Post Office. It acts as a message broker that catches every scroll, like, and video view in real-time, holding them safely until the system is ready to process them.
- RabbitMQ: Often used for managing background tasks, such as sending out push notifications or handling simple message queues.

2. The Processing Layer (The Brain)

- Apache Spark : The industry standard for real-time speed. It processes data in-memory , making it up to 100x faster than older methods. This is how Instagram knows what to show you the second you open the app.
- Apache Flink: A high-performance alternative to Spark for continuous, millisecond-level data streaming.
- Python (Django Framework): The core programming language for Instagram's backend. It is flexible, powerful, and allows engineers to build complex features quickly.

3. The Storage Layer (The Memory)

- HDFS (Hadoop Distributed File System): A Data Lake that spreads files across thousands of cheap servers. If one server dies, the data stays safe on the others (Fault Tolerance).
- Amazon S3: A cloud-based storage system used to store the trillions of actual photos and videos uploaded to the platform.
- PostgreSQL: Used for the Legacy side or for highly structured data like user login info and account settings.
- Apache Cassandra (NoSQL): A decentralized database that handles massive amounts of data across multiple regions (like keeping your feed fast whether you are in London or Tokyo).

4. The Analytics & AI Layer (The Intelligence)

- TensorFlow / PyTorch: These are the AI frameworks used to build the Recommendation Engines. They learn your taste profile to decide which Reels go viral.
- Presto / Hive: Tools that allow analysts to run SQL-like queries on the massive Data Lake without needing to move the data.
- Tableau / Power BI: Used by the business team to create the Descriptive Analytics dashboards.

Detailed description of my contribution

Task 1: Infographic Development & Conceptualization

I led the creative direction for the project's visual aids. I chose a high-contrast, neon-tech aesthetic to represent the modern nature of Instagram's infrastructure. I designed the infographic to show a side-by-side evolution,.

Task 2: Role Play & Communication Strategy

I developed two distinct personas to represent the classic Tech vs. Business conflict. I framed the Business Manager's perspective around the reliability of SQL/Excel and my own persona around Future-Proofing through Big Data. I successfully argued that Big Data isn't just a cost it's a revenue driver through real-time Predictive Analytics. To simulate a high-level consultation that justifies the migration from traditional RDBMS (SQL) to a Hadoop-based Big Data ecosystem for a hyper-growth platform like Instagram.

Task 3: Architectural System Design

I mapped out the four critical layers of data flow: Data Sources, Processing, Storage, and Analytics. I conducted a deep-dive analysis into the differences between Schema-on-Write (Traditional) and Schema-on-Read (Big Data). I researched and implemented the logic for Horizontal Scaling, explaining how Instagram handles billions of users by adding cheap nodes instead of one expensive server.

Task 4 : Tools and Analytics

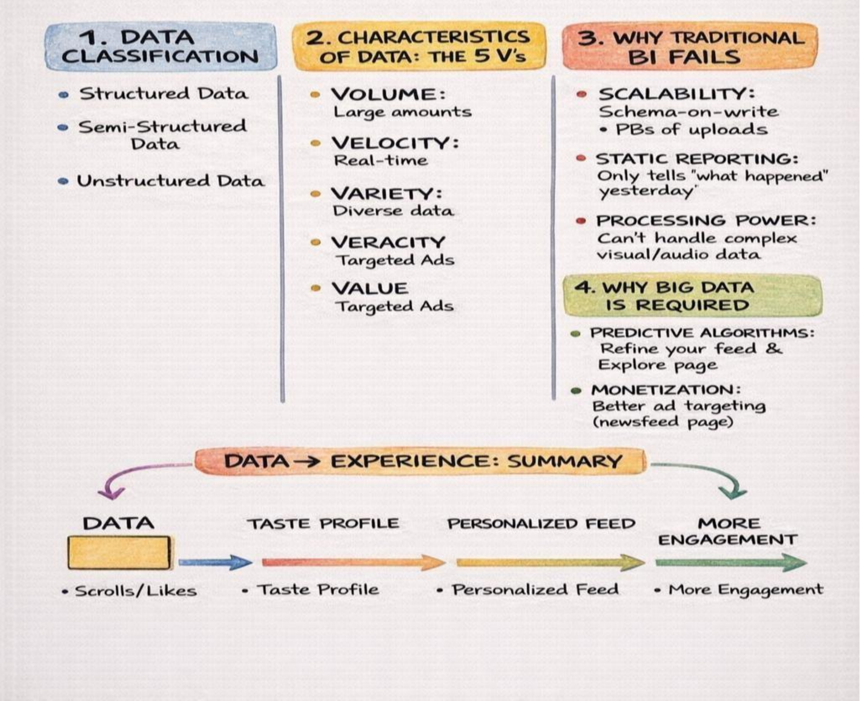
I identified and justified the use of specific industry-standard tools including Apache Kafka (Ingestion), Apache Spark (Processing), HDFS (Storage), and TensorFlow (AI). I created a comparative table showing how traditional tools (like SQL Server) are replaced by Big Data tools (like Cassandra and S3) to meet the "3 V's" (Volume, Velocity, Variety). I ensured that each tool mentioned was correctly placed within the architecture to reflect a real-world production environment.

Implementation

- **Workflow Mapping:** I defined the 4-phase roadmap to transition Instagram from a reactive SQL environment to a proactive Big Data environment.
- **Risk Mitigation:** I specifically chose HDFS for the implementation phase to solve the risk of Single Point of Failure, ensuring the app stays online even if a server rack fails.
- **Real-time Logic:** I advocated for an In-Memory processing strategy (Spark) over traditional disk-based processing to meet the Velocity requirements of a modern social feed.
- **Cost-Benefit Analysis:** I structured the implementation to use Commodity Hardware, proving that Big Data can be implemented cost-effectively by using many small computers rather than one giant super-server.

Results

Task 1:



1. DATA CLASSIFICATION

- Structured Data
- Semi-Structured Data
- Unstructured Data

2. CHARACTERISTICS OF DATA: THE 5 V's

- VOLUME: Large amounts
- VELOCITY: Real-time
- VARIETY: Diverse data
- VERACITY Targeted Ads
- VALUE Targeted Ads

3. WHY TRADITIONAL BI FAILS

- SCALABILITY: Schema-on-write • PBs of uploads
- STATIC REPORTING: Only tells "what happened" yesterday"
- PROCESSING POWER: Can't handle complex visual/audio data

4. WHY BIG DATA IS REQUIRED

- PREDICTIVE ALGORITHMS: Refine your feed & Explore page
- MONETIZATION: Better ad targeting (newsfeed page)

DATA → EXPERIENCE: SUMMARY

DATA → TASTE PROFILE → PERSONALIZED FEED → MORE ENGAGEMENT

• Scrolls/Likes • Taste Profile • Personalized Feed • More Engagement

Fig 1.1

Task 2 : Role play

Manager: Consultant, I just don't see the point. My Excel macros function, my SQL queries are clear, and my team is aware of the precise location of each data row. When something isn't broken, why fix it?

Consultant: It's constricted rather than broken. In a suit and tie, you're attempting to complete a marathon. You're moving, of course, but the Three V's Volume, Velocity, and variety are going to cause you to overheat.

Manager: My SQL server handles my volume just fine.

Consultant: For now. But your Traditional BI relies on Vertical Scaling buying a bigger, more expensive server every time you grow. Eventually, you hit a ceiling. Big Data uses Horizontal Scaling, adding hundreds of cheap nodes to a cluster instead.

Manager: Okay, but what about the Variety part? Data is data.

Consultant: Not in the modern world. Your SQL setup is built for Structured Data neat little tables. But 80% of your customer insights are in Unstructured Data: voice notes, images, and sensor logs. Traditional BI tools literally can't see that content.

Manager: So I just need a better import tool?

Consultant: No, you need a different philosophy. You're currently using Schema-on-Write, where you must define exactly what the data is before you save it. Big Data uses Schema-on-Read we dump everything into a Data Lake and figure out the structure only when we need to ask a question.

Manager: Dumping everything in a lake sounds like a mess. How do you even find anything?

Consultant: That's where Hadoop comes in. It uses a Distributed File System (HDFS). It breaks your data into chunks and replicates them across the cluster. It's Fault Tolerant, meaning if one computer dies, the system doesn't even skip a beat.

Manager: My SQL database has relationships for a reason. Everything is connected.

Consultant: And that's exactly why it's slow. For massive scale, we use NoSQL databases. They abandon the rigid table structure for high-speed, flexible storage that can handle millions of hits per second without locking up.

Manager: Is that why my real-time dashboard usually has a 24-hour delay?

Consultant: You're stuck in Batch Processing (ETL). By the time you see a trend, it's already historical. We need Stream Processing tools like Apache Spark to analyze data the second it's created.

Manager: So what's the Executive Summary? Why should I care?

Consultant: Because right now you're doing Descriptive Analytics . Big Data gives you Predictive Analytics. It uses Machine Learning to tell you what will happen predicting Customer Churn before the customer even knows they're unhappy.

Manager: Alright. You've convinced me that my spreadsheets are a safety net I've outgrown. Where do we start?

Consultant: We start by identifying your most valuable dark data the stuff you're collecting but not using. Once we unlock that with a modern stack, you'll stop looking at the past and start engineering the future. Let's grab a coffee and map out your first Hadoop cluster.

Task 3 :

Traditional Data ware house

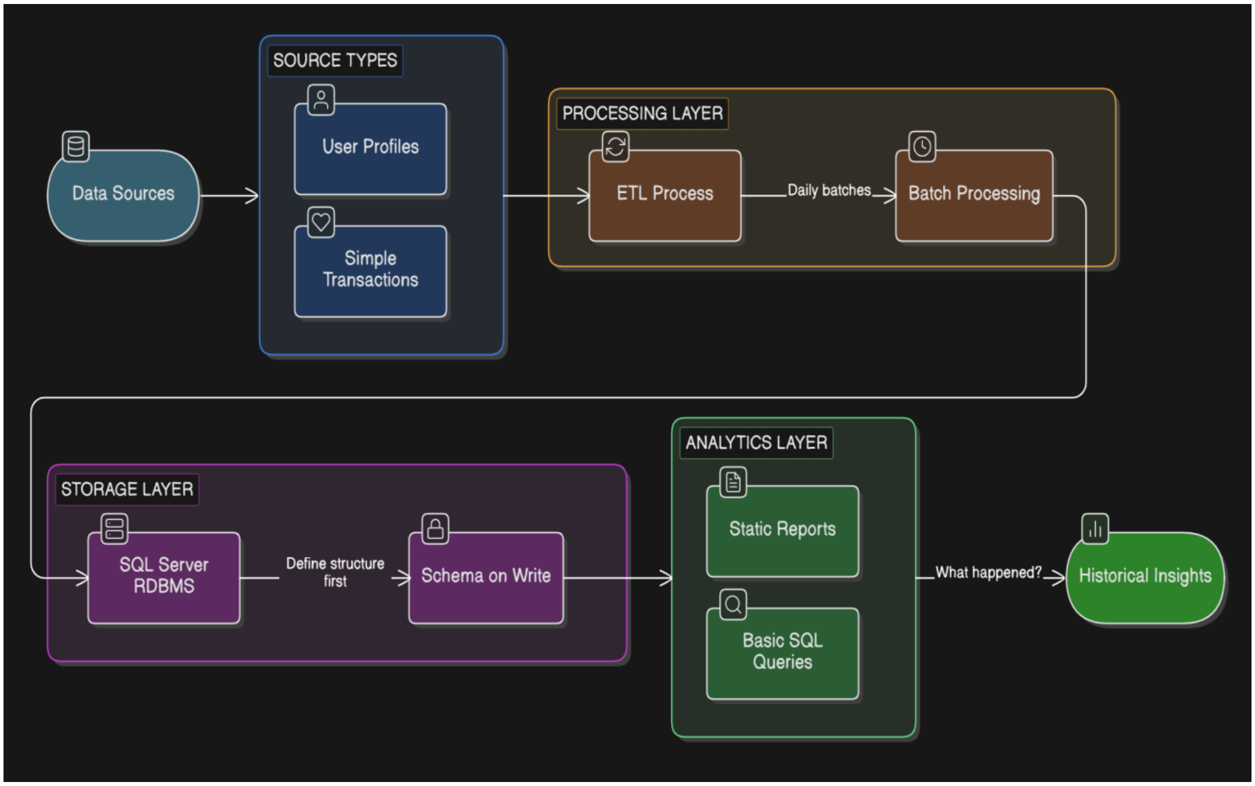


Fig 1.2

Hadoop based big data architecture

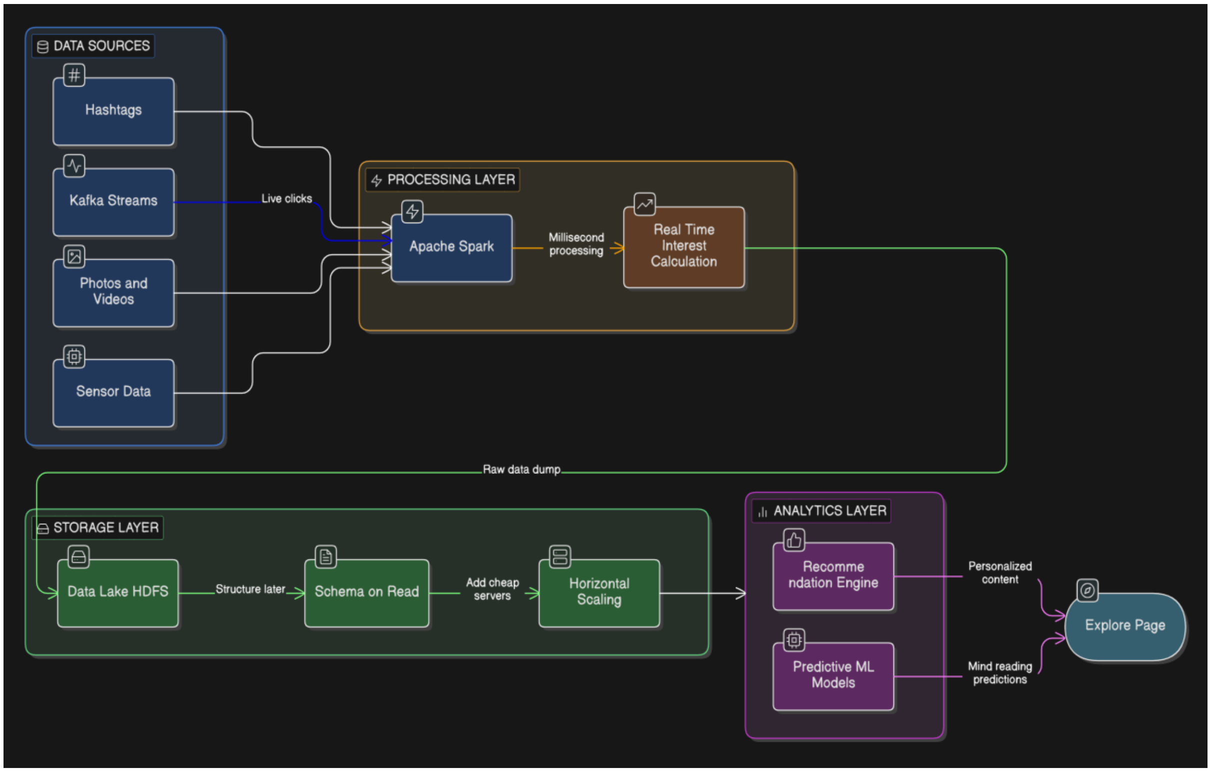


Fig 1.3

Task 4:

Business Question	Analytics Type	Tool
What happened?	Descriptive Analytics	SQL, Power BI, Tableau, Excel
Why did it happen?	Diagnostic Analytics	Google Analytics, Drill-down Dashboards, Data Mining
What will happen next?	Predictive Analytics	Machine Learning (Python/R), Apache Spark, SAS
What action should be taken?	Prescriptive Analytics	Recommendation Engines, AI Optimization, Simulation Tools

Table 1.1