Page No.
Date

04/10/2023

Name : Kreena Shah
Sapid : 60004210243
Batch : C'32
Subject : ADBMS

Experiment 1

Aim : Case Study on professional and commercial database : Summary and comparison.

Theory :

MakeMyTrip, one of India's leading online travel companies, has long relied on MongoDB as its database system. MongoDB, a NO SQL database, offers advantages like Developer UX., Scalability, Transactionality, platform maturity & an extensive ecosystem. These features have been vital for MakeMyTrip's operations. However as the company grows & the demand for realtime data processing increases, it's crucial to explore more efficient solutions.

Requirements that are crucial for MakeMyTrip's operations.

(a) Real Time Data Processing
(b) Scalability
(c) Reliability
(d) Performance
(e) Cost Effectiveness

Page No.
Date

What MongoDB provided us ?

(a) Developer UX
MongoDB's flexible schema allows developer to work with semi-structured & structured data intuitively, accomodating changing data requirements.

(b) Scalability & Transactionality
MongoDB's horizontal scalability supports growing data volumes & transactional support, crucial for applications requiring ACID properties.

(c) Platform & Ecosystem Maturity
With a mature ecosystem & extensive community support. MongoDB has proven reliable. It integrates well with various languages, frameworks & cloud platforms

Limitations of MongoDB

(a) Joins not supported
MongoDB lacks traditional SQL join support, making complex queries with multiple collections challenging & potentially leading to slower query performance.

(b) Limited Data Size & Nesting
MongoDB may not handle extremely large datasets efficiently or complex, nested data structures.

Page No.
Date

Comparison of MongoDB with our proposed database "Apache Druid"

Features	MongoDB	Apache Druid
(1) Real Time Data Processing	It is an excellent for transactional workloads but may not be the best fit for real time data ingestion & fast aggregation queries	It is designed specifically for real time data ingestion & can provide sub-second query responses, which is crucial for real-time data analytics
(2) OLAP workloads	It is a document-oriented database that excels in operational workloads. However, it may not be as efficient for online analytical processing workloads that require fast aggregation queries over large datasets	Apache Druid, with its column oriented storage can perform complex analytical queries more efficiently, making it better fit for OLAP workloads
(3) Data Retention & Roll up	It stores data as it comes without any roll-up feature. This could lead to	It supports data roll-up at ingestion time, which can significantly reduce

		larger storage requirements if detailed raw data needs to be retained	the size of raw data. It is beneficial when retaining detailed data is not the requirement
(4)	Query Flexibility	It's query language is powerful but may not offer the flexibility needed for complex analytical queries	It is fulfilled by Druid Apache

Conclusion :

MakeMyTrip's decision to transition to Apache Druid is not just a technical upgrade but a strategic step towards ensuring our sustained growth & excellence in the travel industry.

Kaleena Shah - 60004210274
Date of Submission - 20/10/23

Page No.
Date

14/15

19/10/2023

ADBMS Exp 3

Aim : Implementation of Query Monitoring

Theory :

SQL Query Monitoring is the process of monitoring the performance & output of SQL queries. A SQL query can be simple or complex, but it is defined with a definite purpose

Eg : Retrieving the details of all cars sold last year.

The main purpose of SQL query monitoring is to determine if the query is able to fulfill the purpose with the expected time & whether it is able to provide the required output. Monitoring is also used for identifying performance improvement measures

Components

(1) Result Grid :

The result area of the screen shows from executed statements. If the script contains multiple statements, a result subtab will be generated for each statement

(2) Field Types :

It is used to list down the fields of all relation mentioned in the query with field name, schema, table name, type of attribute, character set, display size, precision, scale

(3) Query Stats

Query stat editor results tab uses, performance, schema data to gather key statistics collected for executed query such as timing, temporary tables, indexes, joins, etc

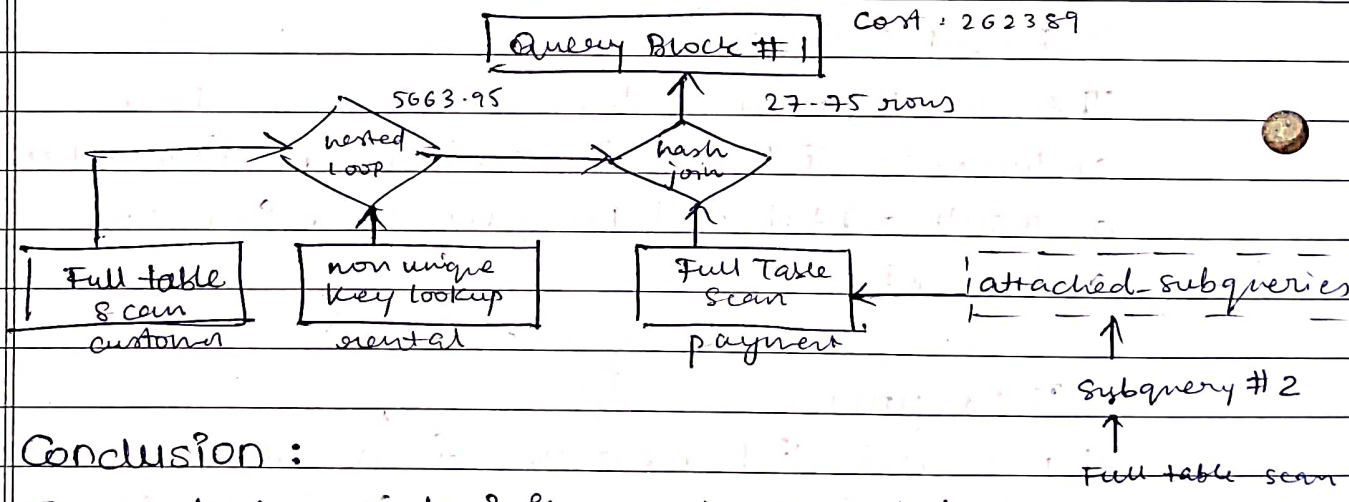
(4) Execution Plan

To view a visual explain execution plan, execute the query & then select execution plan with the results tab

The execution plan defaults to visual explain, but includes a tabular explain

Query :

```
SELECT * FROM CUSTOMER, RENTAL, PAYMENT WHERE
Customer, Customer_id AND amount = (SELECT MAX(Amount)
FROM payment)
```



Conclusion :

To conclude, we briefly understand the components of query monitor & implemented it

Exp 4 : Optimize Using B & B+ Trees

13/15

Name : Kareena Shah
Sapid : 60004210243
Batch : Comps C'32
Subject : ADBMS
Date of Performance : 26/10/2023
Date of Submission : 02/11/2023

Aim : Optimize using B & B+ Trees

Theory :

B - Tree

B-Tree is a self learning, balancing, tree data structure that plays a crucial role in advanced database management systems. It is designed to make a balance between read & write operations making it efficient for indexing & searching in database systems.

Features :

(1) Balanced Structure

B-trees are balanced, meaning all leaf nodes are at the same level. This balance ensures that search & insertion operations remain consistent & efficient

(2) Ordered Data

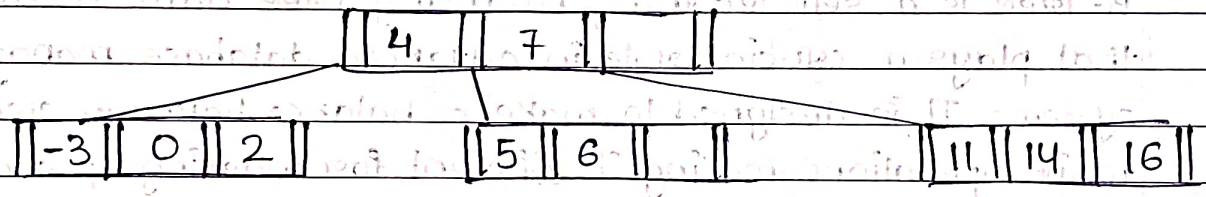
B-Trees organize data in a sorted manner, which is particularly useful for range queries & data retrieval operations.

(3) Split & Merge

When a B-Tree node becomes full during an insertion operation, it splits into two nodes

(4) Fast Search
Provides efficient search operations, often requiring only a few comparison to locate a specific item within large dataset

(5) Node Capacity
The no. of children a node can have is determined by the order of the B-tree. Higher order B-trees reduce the trees height & improve performance



B+ Trees

B+ Trees is an extension of B-Tree data structure with a focus on optimizing range queries & disk storage in database systems. In ADBMS, B+ trees offer several advantages for handling data efficiently

Features :
(1) Balanced & Ordered
Like B-Trees, balanced & ordered data is maintained, ensuring efficient insertion, deletion & search operations.

(2) Reduced Height

This have higher branching factor & Keeps the tree height shallow, leading to quicker access time

(3) Non Leaf Nodes

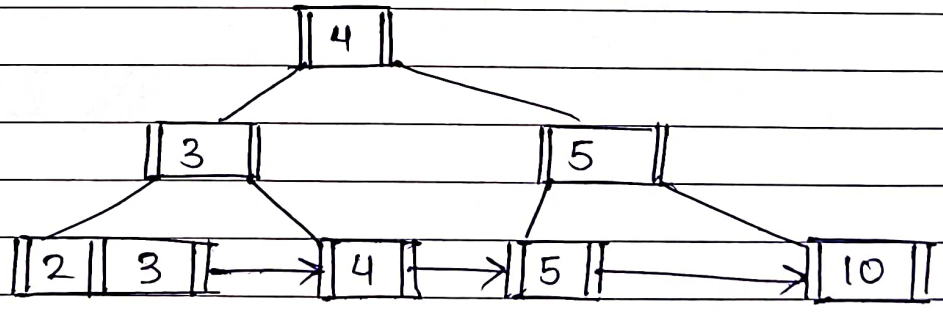
They store key values & add as an index to guide searches in tree

(4) Leaf Nodes

All data is stored in the leaf nodes, which are linked together in a doubly linked list. This structure is advantageous for range queries, as scanning data becomes more straightforward.

(5) Disk Based Storage

B+ Trees are well united for database systems because they optimize disk storage. Sequential access of operations are efficient, reducing disk I/O operations.



Search Time & Insertion Time

For 10000000 records,

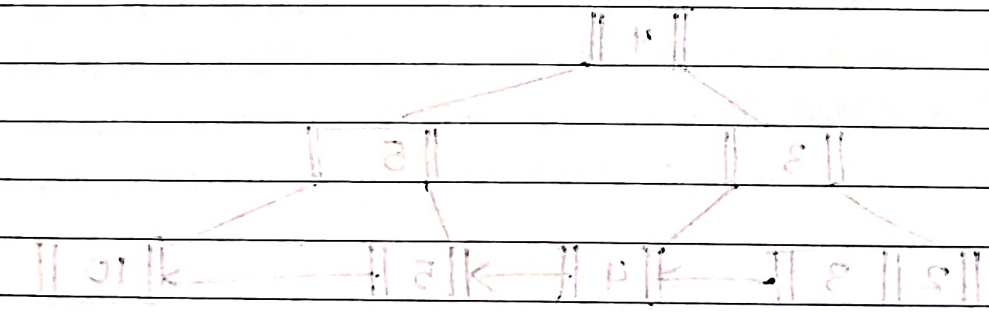
For key = 33333

Search Time for B Tree = 5437.129 millisecond

B+ Tree = 5613.673 millisecond

Conclusion :

Optimization using B & B+ Tree aims to reduce overhead & improve efficiency of data operations resulting in faster & more streamlined data management



Kareena Shah - 60004210245
Date of Submission - 26/10/2023
Date of Performance - 02/11/2023

Page No. (1)
Date

26/10/2023

ADMS - Exp 5

15/15

Aim : To implement Fragmentation (Range, List, Key, Hash)

Theory :

Fragmentation

(1) Fragmentation refers to the division of data & database objects into smaller, more manageable pieces

(2) Fragmentation must be done in such a way that the database can be reconstructed from the fragments

(3) There are two types of fragmentation :

- Horizontal Fragmentation
- Vertical Fragmentation

Horizontal Fragmentation

It involves dividing a table into subsets of rows based on specific condition. It is useful for distributing data in distributed database. The fragments can be then reconstructed using simple union operation.

Vertical Fragmentation

This type divides a table into subsets of columns or attributes. Each subset contains a different set of attributes It is used to improve query performance & manage data access. The fragments can then be reconstructed using JOIN operation

Mixed / Hybrid Fragmentation

Mixed Fragmentation combines both horizontal & vertical

Page No.
Date

fragmentation allowing for more granular control over data distribution in complex distributed database systems. It is used when data needs to be distributed with different structures in various locations.

Partitioning

(1) It is a MySQL database feature that can be used to improve the performance & manageability of your database. It enables you to define groups of rows or index keys within a table according to some algorithm or scheme. You can store each group/fragment/partition in a separate space associated with a specific physical disk. One can use SQL statements to create the fragments & assign them to space.

Types :
⇒ Range Partitioning
⇒ List Partitioning
⇒ Key Partitioning
⇒ Hash Partitioning

Range Partitioning

Here data is partitioned based on a specific range of values for a chosen value/column. It is suitable for data that can be logically divided into non-overlapping ranges

List Partitioning

Here, data is classified into partiti partitions based on

Page No.
Date

specific values for a selected column. It defines a list of discrete values for this column & data rows are placed into partitions according to matching values

Key Partitioning

Here, partition of data is done using a combination of columns. The data is divided based on a function of multiple columns & partitions are defined by the columns i.e. identified by the result of this function.

Hash Partitioning

This involves partitioning data based on a hash function applied to a chosen columns.

Conclusion :

Implementing fragmentation using various techniques like range, list, key, hash partitioning is vital for optimizing database performance & data distribution. These methods enhances data organization & retrieval, benefiting data availability, scalability & performance

15/15

Page No.
Date

Name : Kreena Shah
Sapid : 60004210243
Batch : Comps C'32
Subject : ADBMS
Date of Performance : 02/11/2023
Date of Submission : 02/11/2023

Aim : To Query XML databases.

Theory :

An XML database is a type of database management system specifically designed to store, retrieve & manage XML (extensible Markup Language) data. XML is a widely used data format for structuring & representing data, especially in the context of web services, document storage & data interchange. XML databases are optimized for handling XML data & provide features tailored to the storage & retrieval of XML documents.

Features :

(1) Native XML Storage
XML databases store XML documents in a native format, preserving the structure & heirarchy of data, making it easier to query & manipulate

(2) Indexing
XML databases often use indexing techniques to speed

up data retrieval. These indexes can be based on elements, attribute values

(3) Update Capabilities
XML databases allow you to insert, delete, update XML documents, making them suitable for applications where data is frequently modified

(4) Transaction Support
Many XML databases support transaction to support ensure data consistency & reliability, similar to traditional relational databases

(5) Scalability
XML database can be scaled to handled large volumes of XML data efficiently, making them suitable for applications with growing data requirements

Querying in XML databases

(1) XQuery
XQuery is a powerful query language specifically designed for querying
Eg:

```
for $book in collection("books")/bookstore/book
return $book/title
```

Page No.
Date

XPath

XPath is a simpler query language for selecting elements & attributes from XML documents

Eg :

```
//product [price < 50]
```

Conclusion :

To query XML databases provides valuable insights into the capabilities performance & practical use of XML database system. It equips you with the knowledge & skills necessary to effectively work with XML data in various applications.

Kreena Shah
60004210243
C'32

19/10/2023

ADBMS Exp 2

12/15

Page No.
Date

Submission: 26/10/2023

Aim: Simulate Query Optimization By applying an SQL query on any database

Theory:

SrNo	Query	Normal	Optimized	Indexed
	SELECT COUNT(*) FROM user_details;	0.000166	0.0019	0.000157
	SELECT AVG(user_id) FROM user_details;	0.1495	0.0174	0.0167
	SELECT * FROM user_details ORDER BY last_name DESC	0.5025	0.50426	0.1218
	SELECT * FROM user_details WHERE password IN (SELECT password FROM user_details WHERE password LIKE '2%');	0.9533	0.9581	0.1625
	SELECT u1.username AS user1_username, u2.username AS user2_username FROM User_details u1 JOIN user_details u2 ON u1.first_name = u2.first_name AND u1.user_id = u2.user_id	1.0647	1.065	1.060

Page No.
Date

SQL Query Optimization

SQL query optimization is defined as the iterative process of enhancing the performance of a query in terms of execution time, number of disk accesses & other cost measuring criteria

Optimization of Queries

Optimization in the context of database & query performance involves refining & finetuning various aspects of database operations to make them more efficient. This process aims to reduce query execution time, enhances system responsiveness & optimizes resource utilization.

Optimization can include rewriting queries to improve their logic, creating or adjusting indexes on database tables, optimizing data storage & finetuning database configuration. Optimization results improves system performance, reduced response time, better scalability & ensures that the database can handle heavy workloads efficiently

Indexing in Queries

Indexing is a fundamental database optimization technique that enhances query performance by efficiently organizing & accessing data. Instead of scanning the entire dataset, indexes provide a better & optimized way to accen the desired data records

There are various types of indexes

(1) Primary Indexing

(2) Clustered Indexing
(3) Secondary / Non-clustered Indexing
(4) Multilevel Indexing
Properly implemented indexing can significantly enhance database performance.

Observations :
The mentioned query table provides performance for a set of SQL queries of a normal select query, using aggregate functions, using ORDER BY for ordering on specific attribute, nested queries & using JOINS.
We implemented same set of queries by running queries, using optimization & then by indexing. After optimization of table, not much effect was observed. But after indexing, significant decrease in response time was observed

Conclusion :
The experiment concluded that optimization efforts improved query performance to varying degrees, with some queries showing significant gains while other has modest improvements.
Continuous monitoring & trade off considerations are essential for achieving desired performance

Page No.
Date

Name : Koreena Shah
Sapid : 60004210243
Batch : C'32
Subject : ADBMS
Date of Performance : 27/11/2023
Date of Submission : 27/11/2023

Experiment 6

Aim : Implement 2 phase protocol

Theory :

- As databases handle multiple users making concurrent transactions there is a scope of conflicts & data inconsistencies
- Locking protocols act as an excellent tool for ensuring data integrity by regulating access to shared resources
- The 2 phase locking protocol ensures a systematic & controlled approach to manage lockings adhering to principles of isolation & atomicity
- It plays pivotal role in preventing conflicts between transactions & guaranteering a reliable & coherent state of database
- Each working site in 2ppl has its own log & no global log.

Working :

Phase 1

- The site at which the query is requested becomes controller

In the case
- The transaction is initiated & the controller C_i logs the request in its local log
- The transaction request is sent to the transaction managers across various sites where the changes required
- The transaction managers at these sites, log the incoming request
- If is requested changes at individual sites can be made causing no discripencies, then TM logs <Ready T>, means it's ready for changes
- If changes not applicable, <No T>
- The response generated at each site is transferred back to controlle manager

Phase 2
- The controller is waiting for responses from all sites
- If all sites respond <Ready T>, then the changes proposed are committed
- If even a single site refers the changes are aborted
- If some does not respond within time limit. It is assumed <No T> is response & transaction aborted
- The controller then logs <commit T> if commit. <Abort T> if aborting transaction

Conclusion :
We can conclude execution of 2 phase management system ensures seamless execution & reliability of concurrent transaction

Page No.
Date

ADBMS Experiment 8

Name : Kreena Shah
Sapid : 60004210243
Batch : C'32
Subject : ADBMS
Date of Performance : 02/11/2023
Date of Submission : 30/11/2023

Aim : Data Handling using JSON

Theory :
JSON (JavaScript Object Notation) is a lightweight data interchange format that's easy for humans to read & write.
It is easy for machines to parse & generate
It's widely used for transmitting data between a server & a web application
It is language independent & can be used with modern programming languages
It is self describing
All popular programming languages have support for JSON data

JSON syntax (js)

```
let student = {
  "name" : "Kreena",
  "age" : 19,
  "college" : DJ,
  "subject" : ['ADBMS','AI','DMW','POA','Python','PBC']
```

Page No.
Date

"enrolledFortHonors" : true
}

Conclusion : Thus, we understood & implemented JSON data objectives, usecase.

Page No.
Date

ADBMS Experiment 9

Name : Kreena Shah
Sapid : 60004210243
Batch : C'32
Subject : ADBMS
Date of Performance : 02/11/2023
Date of Submission : 30/11/2023

Aim : Processing of spatial & temporal data

Theory :

Temporal Database

Database that store information which is time dependent in one or the other way, are called as temporal db databases
Temporal DB store information about states of real world across time.
It is a database with built-in support for handling data involving time
It stores information relating to past, present & future time
Examples :
(1) Health Care Systems
History of patients health is required
Here, past records & current condition both have equal significance.

(2) Insurance Systems
We need to keep a track of claims, accident history &

Page No.
Date

many more
Also, the time when policies are in effect

Spatial Database

A spatial database is a db that deals with information associated with geographical locations such as cities, towns, etc
It is optimized to store & query data representing objects
These are the objects which are defined in geometric space
It offers spacial data types in its data model & query language.
It supports spatial datatype in its implementation providing atleast spatial indexing & efficient algorithm for spatial join
They are majorly of 3 types:
(1) Vector data
(2) Rastar data
(3) Image data

Conclusion : Thus, we studied & implemented spatial & temporal data

Page No.
Date

ADBMS Experiment 10

Name : Kreena Shah
Sapid : 60004210243
Batch : C'32
Subject : ADBMS
Date of Performance : 30/11/2023
Date of Submission : 30/11/2023

Aim : Case Study on Database security issues & measures taken to handle those issues.

Theory :
Database security issues are a critical concern in the digital age where vast amounts of data specifically sensitive data is stored & managed.
Here are the primary security challenges
(1) SQL Injection
(2) Weak Authorization & Authentication
(3) Insufficient Data Encryption
(4) Cross-site scripting (XSS) & Cross-site Request Forgery (CSRF)
(5) Insider threats

Case study on recent patent

Title : Systems & methods for determining & executing trusted user access

Filed : July 2022
Published : November 7, 2022

Page No.
Date

Assigned to Walmart Apollo LLC, this patent introduces a system to enhance security in retail settings through use of smart lock technology.

Methodology :
Central to this system is integration of an access member such as a door, secured by a smart lock & a control unit which is a pivotal component, tasked with processing authentication data received from a user interface device. This data is compared with pre-stored authentication data & if match is found, the individual is identified as a trusted user.
The system, upon successful authentication generates & transmits an optimal key to the user interface device which unlocks the smart lock.
This streamlined method replaces the traditional approach.

Advantages
Enhances security
Operational efficiency

Conclusion : This patent offers a superior solution for managing access to secured items in retail setting.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Continuous Assessment for Laboratory / Assignment sessions

Academic Year 2023-24

Name: Kreena Shah SAP ID: 60004210243

Course: Advanced Database Management System Laboratory Course Code: DJ19CEEL5012

Year: T. Y. B. Tech. Sem: V Batch: C'32

Department: Computer Engineering

Performance Indicators (Any no. of Indicators) (Maximum 5 marks per indicator)	1	2	3	4	5	6	7	8	9	10	Σ Avg	Report	Σ Avg
Course Outcome	1	2	2	2	3	3	4	4	4	5			
1. Knowledge (3) (Factual/Conceptual/Procedural/ Metacognitive)	3	3	3	3	3	3	3	3	3	3			
2. Describe (2) (Factual/Conceptual/Procedural/ Metacognitive)	2	1	2	2	2	2	2	2	2	2			
3. Demonstration (3) (Factual/Conceptual/Procedural/ Metacognitive)	3	3	3	3	3	3	3	3	3	3			
4. Strategy (Analyse & / or Evaluate) (3) (Factual/Conceptual/ Procedural/Metacognitive)	3	3	3	3	3	3	3	3	3	3			
5. Interpret/ Develop (2) (Factual/Conceptual/ Procedural/Metacognitive)	2	1	2	1	2	1	2	2	2	2			
6. Attitude towards learning (2) (receiving, attending, responding, valuing, organizing, characterization by value)	1	1	1	1	2	1	2	2	2	1			
7. Non-verbal communication skills/ Behaviour or Behavioural skills (motor skills, hand-eye coordination, gross body movements, finely coordinated body movements speech behaviours)	-	-	-	-	-	-	-	-	-	-	-	-	-
Total	14	12	14	13	15	13	15	15	15	14			
Signature of the faculty member	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Outstanding (5), Excellent (4), Good (3), Fair (2), Needs Improvement (1)

Laboratory marks Σ Avg. =	Assignment marks Σ Avg. =	Total Term-work (25) =
Laboratory Scaled to (15) =	Assignment Scaled to (10) =	Sign of the Student: Kreena

Signature of the Faculty member:
Name of the Faculty member: Prof Kranti Ghag

Signature of Head of the Department
Date: 7/12/2023