VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018, Karnataka

Project Report
on

"Serverless auto scalable E-commerce App"

Submitted by

USN	Name
1BI19CS400	**AKSHAY KUMAR U**
1BI18CS114	**PRATHIKSHA P H**

Under the Guidance of
Dr. SUNEETHA K R
Associate Professor
Department of CS&E, BIT
Bengaluru-560004

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BANGALORE INSTITUTE OF TECHNOLOGY
K.R. Road, V.V.Pura, Bengaluru-560 004

2021-2022 (Odd)

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belagavi-590018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

Bengaluru-560 004

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

Certified that the Mini Project work entitled **" Serverless auto scalable E-commerce App "** carried out by

USN	NAME
1BI19CS400	**AKSHAY KUMAR U**
1BI18CS114	**PRATHIKSHA P H**

of VII semester, Computer Science and Engineering branch as partial fulfillment of the course **Big Data Analytics (18CS82)** prescribed by **Visvesvaraya Technological University, Belgaum** during the academic year 2021-22. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report.

The Mini Project report has been approved as it satisfies the academic requirements in respect of project work in Big Data Analytics.

Dr. Suneetha K R
Associate Professor,
Department of CS&E, BIT,
Bengaluru-560004

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 Introduction

Big Data is a continually progressing term. It is a great deal of sort out amorphous data that can be excavated for information. These educational accumulations are immense and complex that standard data getting ready isn't fit to process them. Enormous Data is being used in various sectors. We will see the effect of Big Data Analytics in changing the E-Commerce business, with the objective that the company surveyed as these E-exchange can benefit the most customers in the relationship from using Big Data because there will be information of the data accumulated on regular bases. Various gigantic retailers regard this present data's information and cause them for predicting the customer interests and give their customers relative and charming looks when they shop on their site, with the objective that they pull in the customer by providing the required and relevant journeys of things or things. These tendencies are inside and out-delivered from the Big Data examination. Huge Data contains two sorts of data one are composed, and the other one is unstructured.

Big Data, similarly as dispersed registering, have been associated in electronic business for a period, which has helped web-based business stages to recommend things even more correctly and rapidly, improve customer web shopping information, streamline collaboration structure and distortion security measure, and so forth. Starting late, a square chain begins to be associated in the web-based business, brings lower trade costs and progressively active portion. Likewise, non-modifying features diminish business distortion and assurance buyer astounding organization. Later on, with the more significant and progressively broad application, enormous information will pass on new a motivating force to cross-edge internet business.

1.1 Motivation

Server less architecture describes a way for companies to build and run applications but not have to manage infrastructure. It provides a way to remove architecture responsibilities from your workload, including provisioning, scaling, and maintenance. Scaling can be automatic, and you only pay for what we use.

Single monolith that handles authentication, payments and sending emails to clients, a micro services architecture has one app for authentication, another for payments and another for filling your inbox with marketing emails. Micro services allow us to scale up just the parts of our application that need it. So a single, tiny server may handle authentication, while ten servers send emails. We get to allocate resource where they're needed most, and the application architecture scales much better. But we still have to manage infrastructure. We still have to spin up ten email servers, and we still need to make sure that those email servers stay in sync as customer data changes.

Server less is much more difficult from a service provider's point of view. The Google engineers who make your code run as if by magic are using extremely sophisticated tooling and programming methods to automatically scale your apps up and down. The upside of server less is that it saves the user significant time and resources. They deploy code to the service provider, and the service provider runs it and scales it up and down.

CHAPTER 2

PROBLEM STATEMENT

2.1 Problem Statement

"To build a Serverless auto scalable E-commerce Application."

2.2 Objectives

- The main purpose of the project to build a serverless E-commerce application.
- The main objective of project is to provide the scalability in various form of Big Data.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 Hardware requirements

A Machine with:

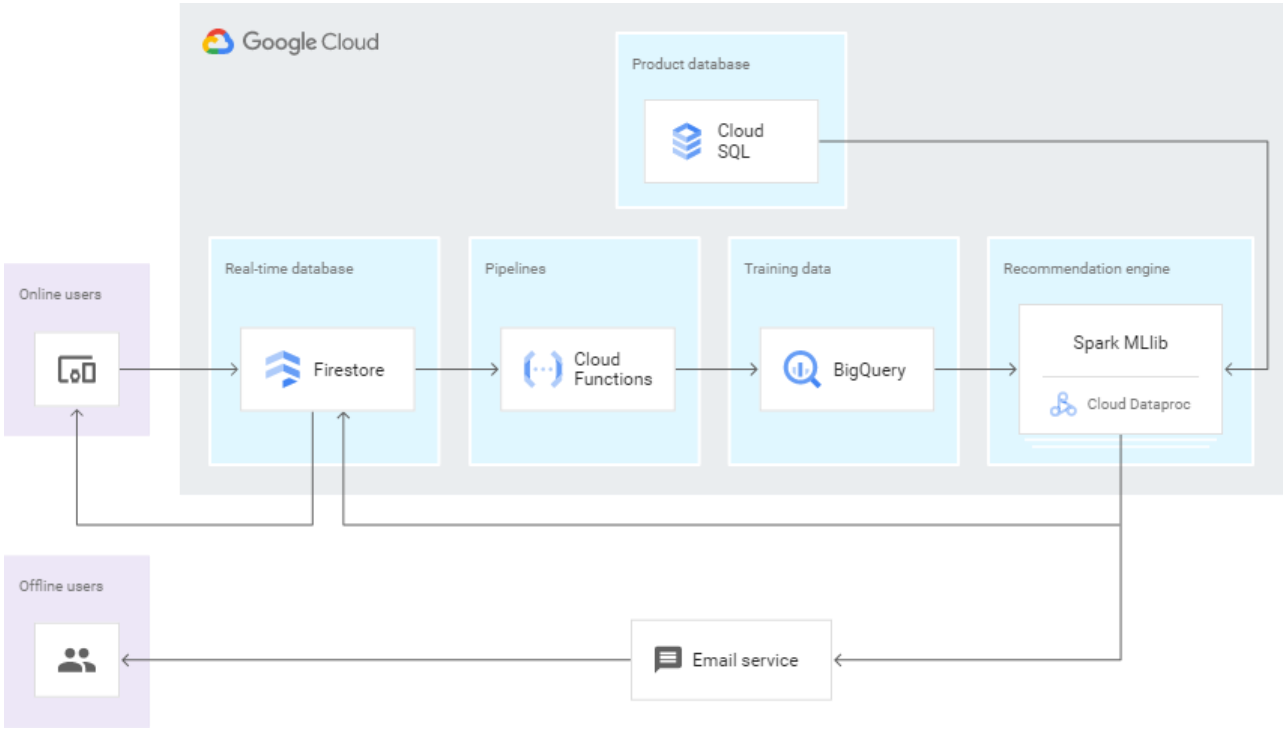
- Hard Disk – 1 TB or Above
- RAM required – 4 GB or Above
- Processor – Core I3 or Above

3.2 Software Requirements

- Cloud Firestore
- Firebase
- Minimum RAM 4 GB and above

CHAPTER 4

ARCHITECTURE



Characteristics of the Firestore:

Firestore is designed for mobile and web applications and for storing hierarchical, transactional data that has a flexible, non-relational schema. Firestore can host any amount of data storage. It handles data amounts from kilobytes to petabytes in the same way, without affecting performance. This specifies with all the characteristics of the big data like Velocity, veracity, velocity and voluminous of Data.

BigQuery: Firestore offers strongly consistent queries across the entire database. Along with primary indexes, Firestore supports secondary and composite indexes to quickly look up locations of items that you request in a query.

Firestore is a non-relational database, so it doesn't support relational schemas or queries that use SQL semantics. In particular, Firestore doesn't support join operations, inequality filtering on multiple properties, or filtering on data that is based on results of a sub query. If the app requires SQL support for non-horizontal scales, use Cloud SQL. If the app requires SQL support for larger horizontal and

global scales, use Cloud Spanner. Firestore is optimized for online transaction processing (OLTP). If your app requires a storage option for full table scans and interactive querying in an online analytical processing (OLAP) system, use Big Query. If your app needs both OLTP and OLAP systems, use Firestore as your OLTP system and incrementally sync or export Firestore data to Big Query for analysis.

Auto scaling:

- Firestore scales up automatically, with no downtime. This scaling mechanism lets Firestore serve thousands of requests per second and millions of concurrent connections. You pay only for your actual usage based on storage size and the number of operations.
- Firestore can handle operations on a massive scale. However, to support complex features like replication and transactions, Firestore makes some tradeoffs that might slow performance for apps that are expected to support extreme loads.
- If your app is extremely write-heavy, consider using Cloud Bigtable for greater data ingestion capabilities at the expense of transactions and secondary indexes.

Analysis real-time capture of data changes

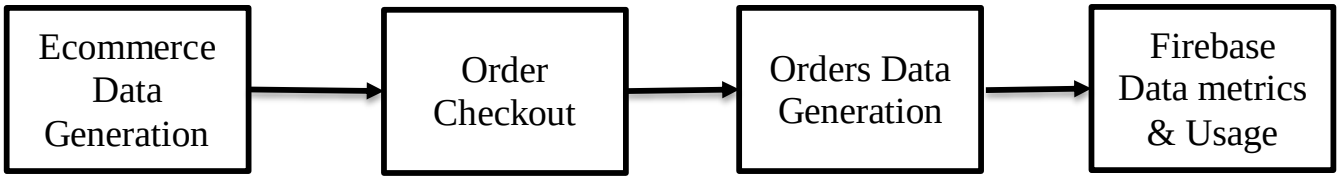
An app receives real-time user input that changes the global state. A dashboard in Data Studio tracks real-time events to better understand user behavior and interactions. When a user action updates any state value, the following events occur:

1. Firestore triggers a Cloud Function that writes the change to BigQuery, including the old and new state values.
2. The Data Studio dashboard runs real-time aggregation queries on the event data in BigQuery.
3. The queries generate metrics like ratio of event changes aggregated to different buckets, unique type of events per time bucket, and event ingestion latency.

CHAPTER 5

MODULE DESCRIPTION

5.1 Module Description



- **Ecommerce Data Generation:** We Load the Ecommerce Product Data to NoSQL Database with thousands of records.
- **Order Checkout:** Users will be accessing the Ecommerce products and generates Order Data which varies day to day by the volume of users accessing the application and playing order.
- **Orders Data Generation and Logging:** The Order data generated by user while check out process will be stored in NoSQL database Firestore.
- **Firebase Data metrics & Usage:** This module is responsible to analyze the bandwidth, Hardware usage, and cost analysis of the Scalable Firebase Database as Backend-as-a-Service.

CHAPTER 6

IMPLEMENTATION DETAILS

6.1 Source Code

1) Initialize Firebase :

```
FirebaseApp defaultApp = await Firebase.initializeApp();
```

2) Read Data from Firebase Firestore (NoSQL Database) :

```
import { doc, getDoc } from "firebase/firestore";

const docRef = doc(db, "Products", "Samsung-JS");
const docSnap = await getDoc(docRef);

if (docSnap.exists()) {
  console.log("Document data:", docSnap.data());
} else {
  // doc.data() will be undefined in this case
  console.log("No such document!");
}
```

3) Write Data to Firebase Firestore (NoSQL Database) :

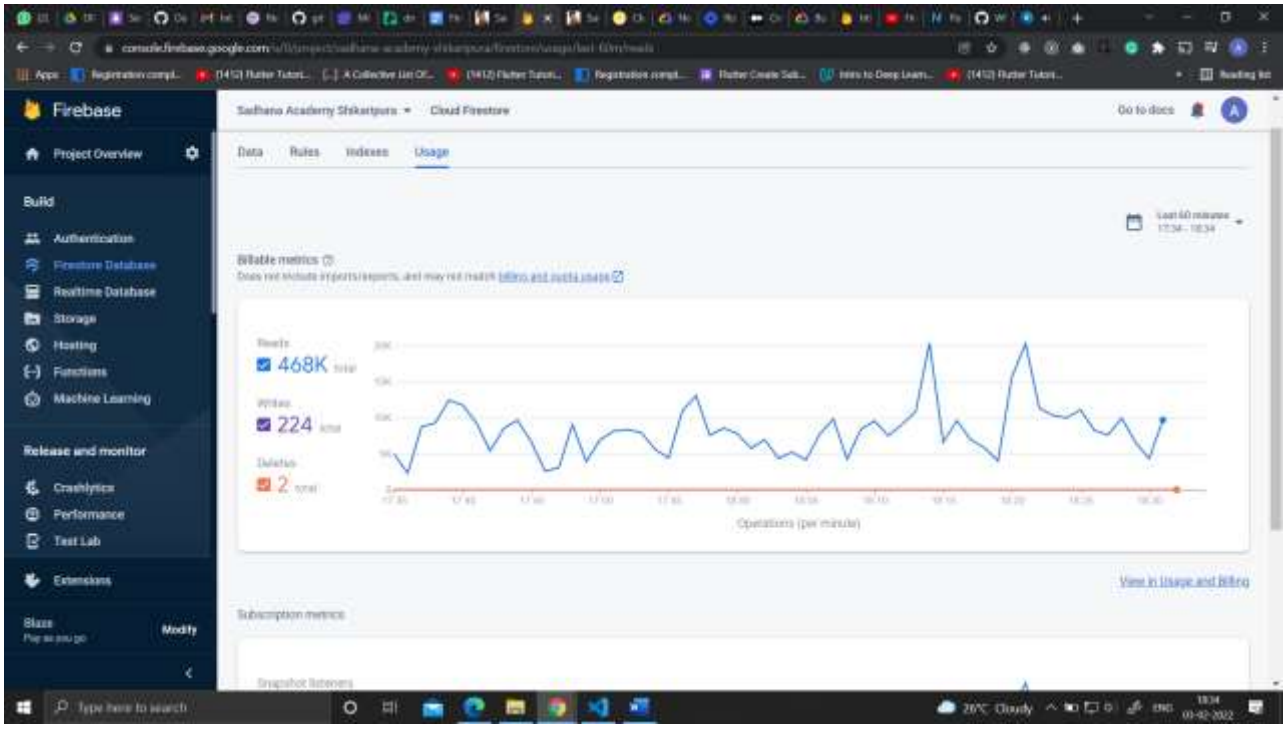
```
import { collection, doc, setDoc } from "firebase/firestore";


const productsRef = collection(db, "Products");


await setDoc(doc(productsRef, "Redmi"), {
    name: "Redmi", price: "10000", manufactured: "USA",
    categories: ["Electronics", "Mobiles"] });


await setDoc(doc(productsRef, "Apple SE"), {
    name: "Apple SE", price: "100000", manufactured: "USA",
    categories: ["Electronics", "Mobiles"] });
```

CHAPTER 7

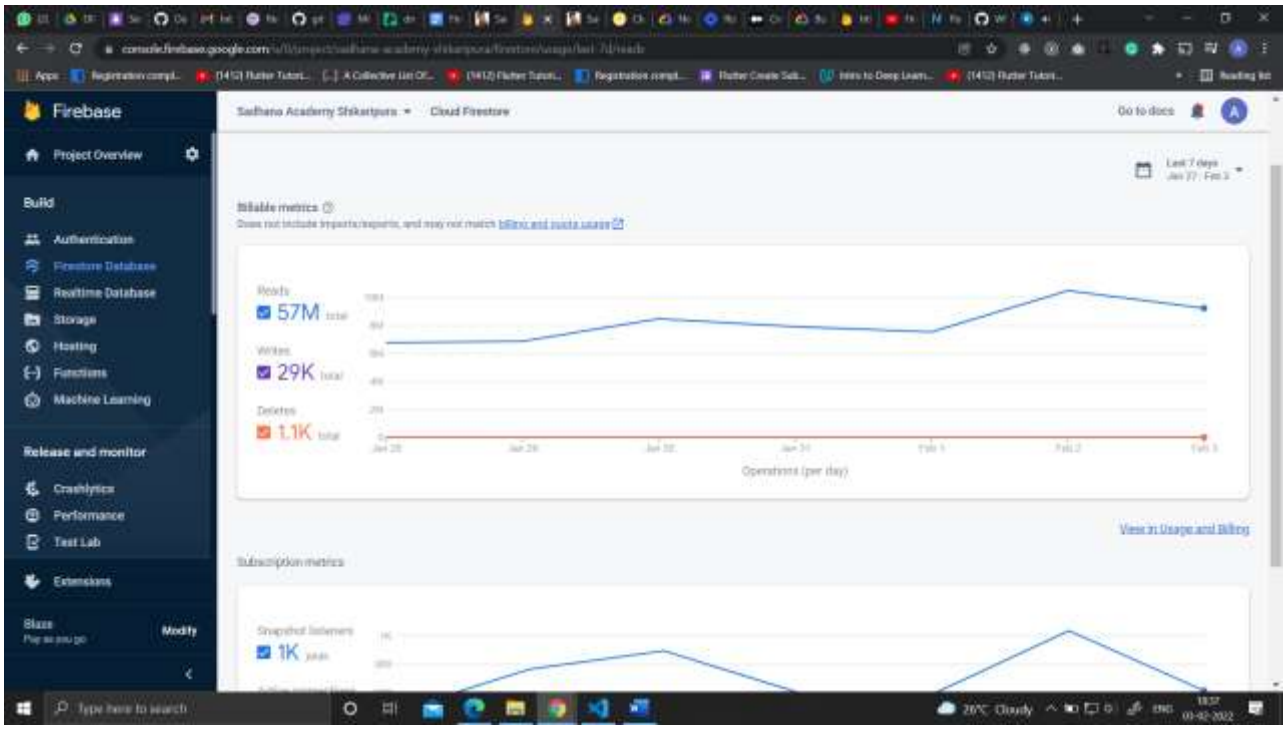
RESULT

7.1. Outputs

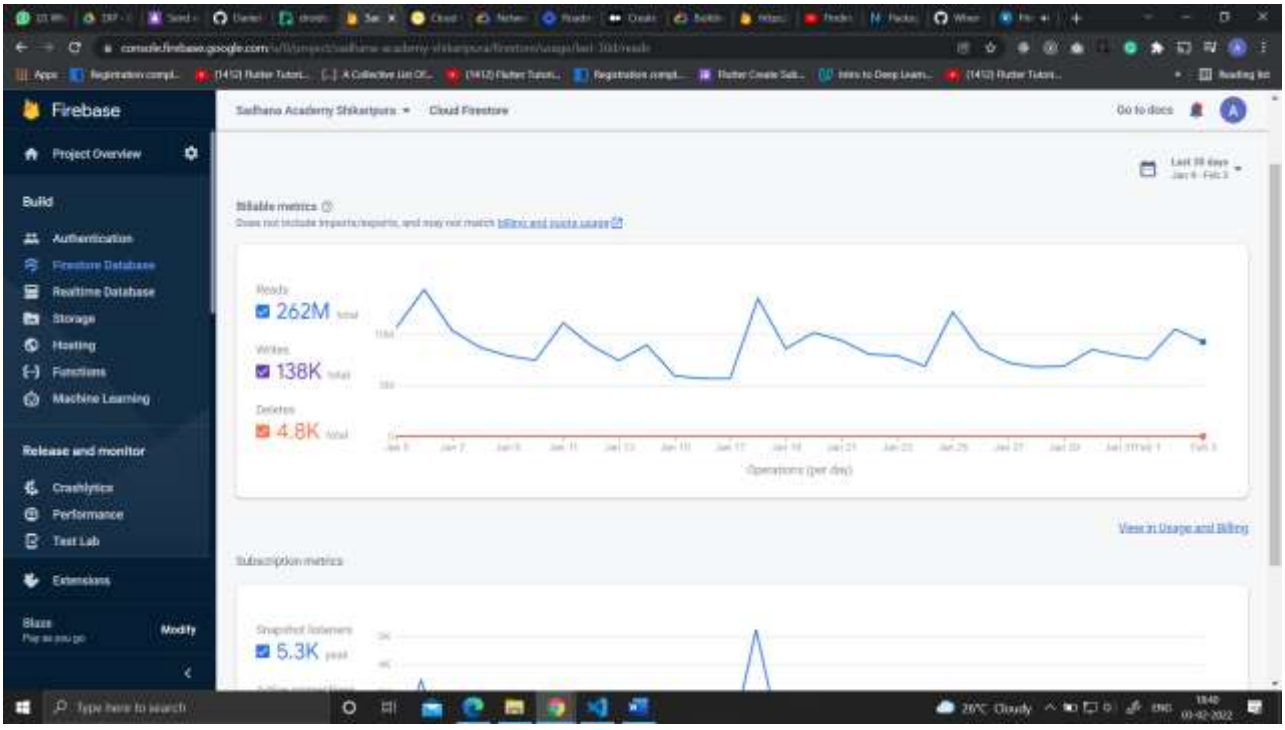
1) Hourly database access metrics :



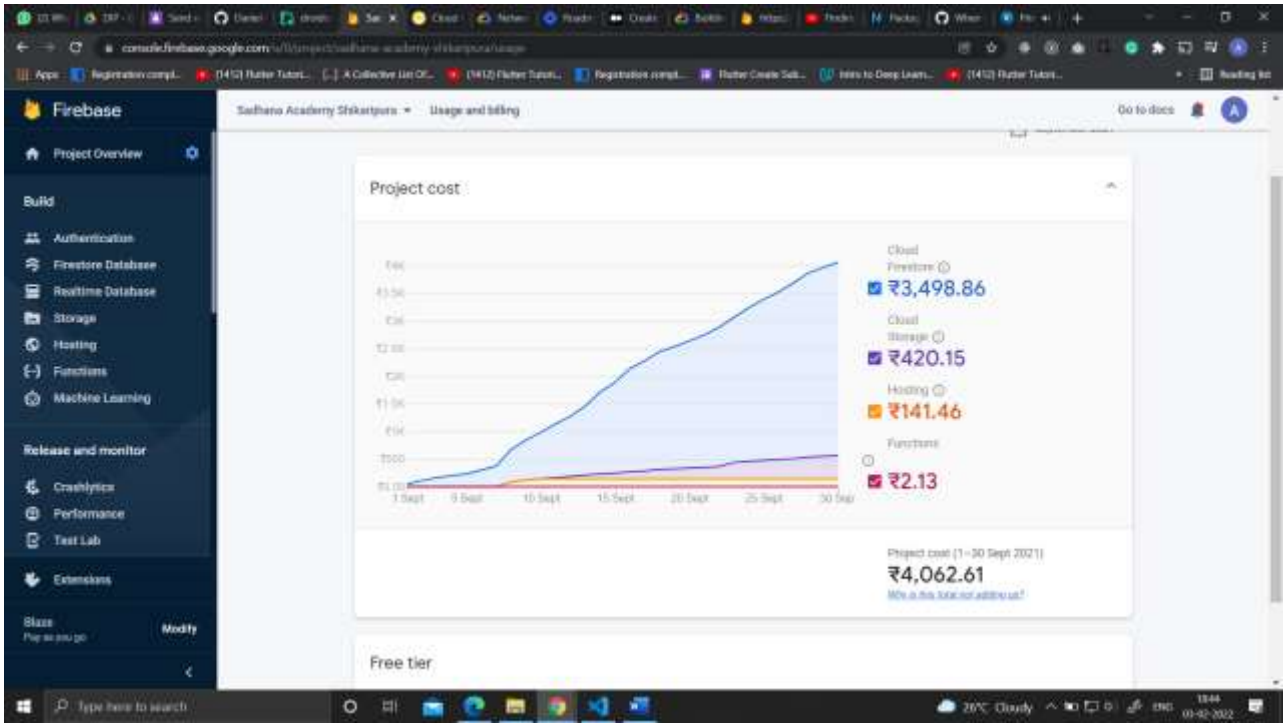
2) Weekly database access metrics :



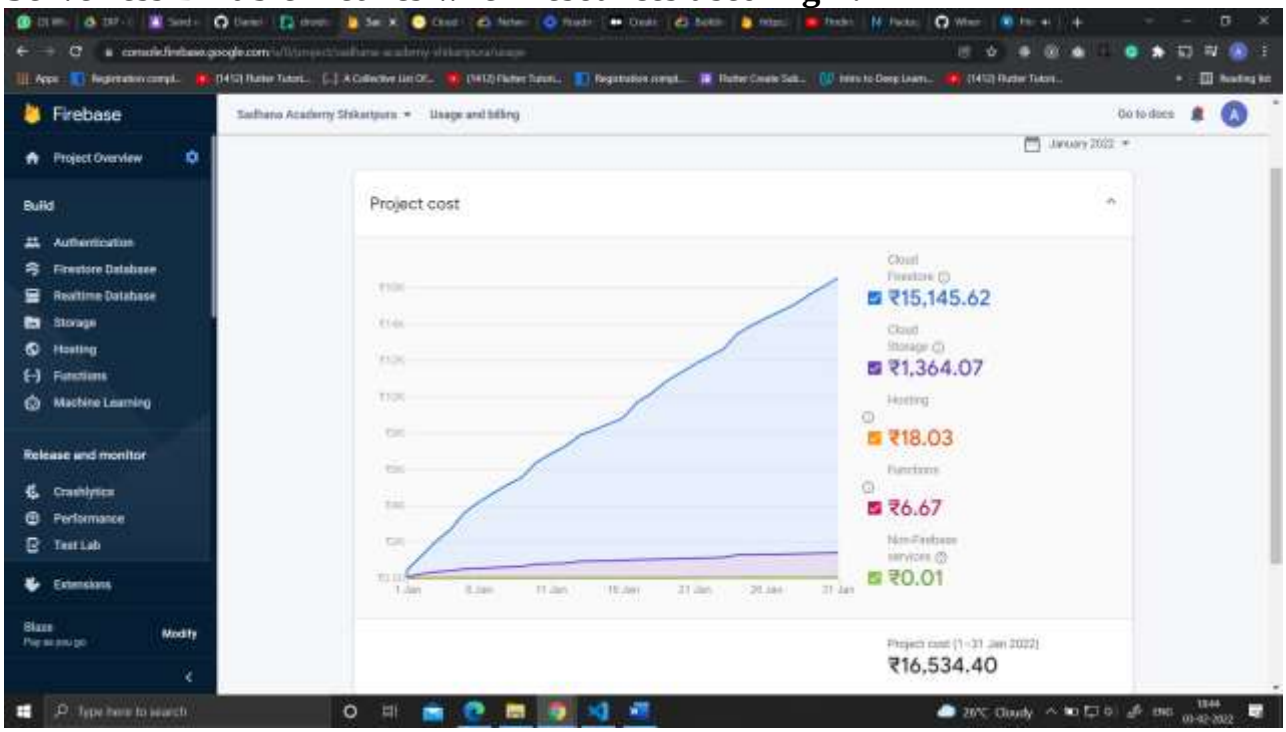
3) Monthly Database Access Metrics :



4) Serverless Billable metrics when resources less :



5) Serverless Billable metrics when resources used high :



CHAPTER 8

APPLICATION

- Companies can save backend costs on Auto Scaling and Handling Big Data using Serverless Solution.
- Serverless backend reduces the backend team task by using Backend-as-a-Service products such as Firebase etc.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

9.1 Conclusion

Big Data is a data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it. Today, Data is generated from various different sources and can arrive in the system at various rates. To process these large amounts of data is a big issue today. Firebase is the core platform for structuring Big Data, and solves the problem of making it useful for analytics purposes. It serves all sorts of operations related to bigdata in faster and convenient way in distributed computing environments.

9.1 Future Enhancement

Future research can be focused on authentication service provided by Firebase, how to implement authentication in your applications, authentication analysis for cross platform development.

BIBLIOGRAPHY

[1] http://www.ijarse.com/images/fullpdf/1515830014_CS07.pdf

[2]https://firebase.google.com/docs/reference/cpp/class/firebase/future

[3]https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0

[4]https://cloud.google.com/blog/products/serverless/6-strategies-for-scaling-your-serverless-applications

[5] https://verygood.ventures/blog/create-an-autoscaling-api-with-firebase-in-just-10-minutes

[6] https://devdactic.com/ionic-e-commerce-firebase/