**Capstone Project**

**Course code: CSA 1581**

**Course : Cloud Computing and Big Data Analytics for Network Virtualization**

**S. No: 08**

**Name : G. Vikram**

**Reg. No: 192211046**

**Slot : B**

**Title : Building a Chat Application with Firebase Realtime Database (XaaS)**

**Project Release Date :**

**Project submission Date :**

**Mentor Name : Dr. M. Prabhaharan**

**Mentor Phone number and Department : 9444717580 / Condensed Matter Physics**

**1.Preliminary Stage**

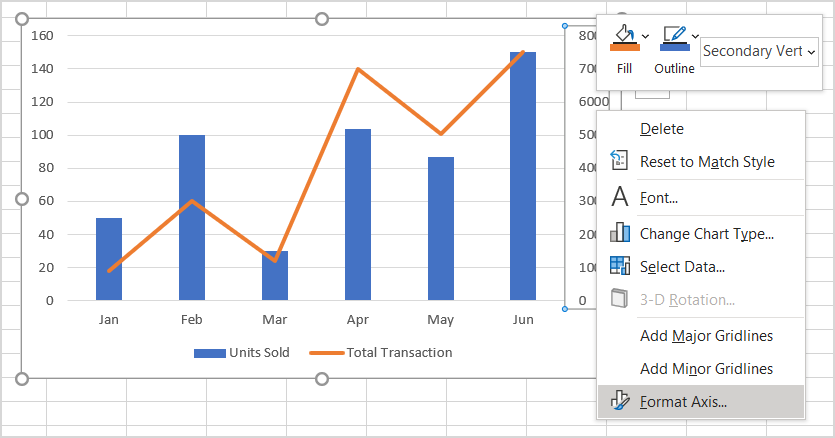
**1.1 Assignment Description :**

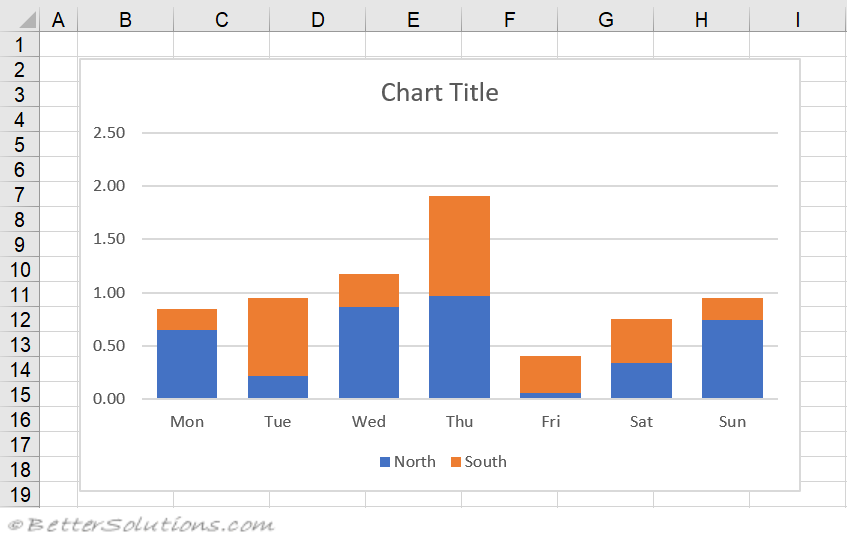
Building a chat application with Firebase Realtime Database (XaaS) is a project that aims to design and develop a real-time chat application using Firebase Realtime Database as the backend infrastructure. The application will allow users to send and receive messages in real-time, with features such as user authentication, chat rooms, and message history. The project will utilize Firebase's NoSQL database to store and retrieve chat data, ensuring scalability and reliability. The application will be built using a modular architecture, with a focus on user experience and performance.

**1.2 Assignment Work Distribution :**

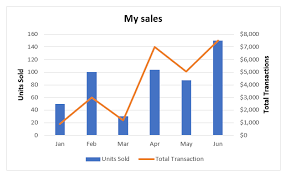
* **Project Scope Definition:**
* Define the scope and objectives of the project: The project aims to design and develop a real-time chat application using Firebase Realtime Database, with a focus on user experience and performance.
* Specific goals of analyzing: The project will analyze the performance and scalability of the Firebase Realtime Database, as well as the user experience of the chat application.
* **Data Collection and Preparation:**
* Identify the data sources: The project will utilize Firebase Realtime Database as the primary data source, with additional data collected from user interactions and chat logs.
* Develop a data collection plan: The project will collect data on user interactions, chat logs, and system performance, with a focus on ensuring data quality and consistency.
* Cleanse and preprocess the collected data to ensure data quality: The project will utilize data cleaning and preprocessing techniques to ensure data quality and consistency.
* Consistency of the project: The project will ensure consistency in data collection, storage, and analysis, with a focus on ensuring data quality and reliability.
* **Exploratory Data Analysis (EDA):**
* Conduct exploratory data analysis: The project will conduct exploratory data analysis to understand the patterns and trends in user interactions and chat logs.
* Understand the patterns and trends: The project will analyze the patterns and trends in user interactions and chat logs, with a focus on identifying areas for improvement.
* Perform descriptive statistics, such as summary statistics, distribution plots, and correlation analysis, to explore the relationships of the data.

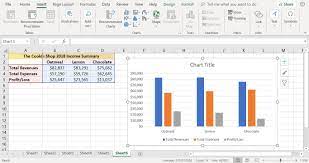
Perform descriptive statistics, such as summary statistics, distribution plots, and correlation analysis, to explore the relationships of the data :





Visualize the data using charts, graphs : (draw charts )





**2. Problem Statement**

The problem statement for this project is to design and develop a real-time chat application using Firebase Realtime Database, with a focus on user experience and performance. The application must be able to handle a large volume of users and messages, while ensuring data consistency and reliability. The project must also ensure user authentication and authorization, as well as provide features such as chat rooms and message history. The project must be scalable, reliable, and secure, with a focus on providing a seamless user experience.

**3. Abstract**

This project aims to design and develop a real-time chat application using Firebase Realtime Database, with a focus on user experience and performance. The application will utilize Firebase's NoSQL database to store and retrieve chat data, ensuring scalability and reliability. The project will analyze the performance and scalability of the Firebase Realtime Database, as well as the user experience of the chat application. The project will utilize data cleaning and preprocessing techniques to ensure data quality and consistency, and will conduct exploratory data analysis to understand the patterns and trends in user interactions and chat logs. The project will provide features such as user authentication, chat rooms, and message history, with a focus on providing a seamless user experience.

**4. Proposed Design work**

* 1. Identifty the key components :
* Firebase Realtime Database
* User authentication and authorization
* Chat rooms and message history
* User interface and user experience
  1. Functionality :
* Real-time messaging
* User authentication and authorization
* Chat rooms and message history
* User interface and user experience
  1. Architectural Design :
* Modular architecture
* Firebase Realtime Database as the backend infrastructure
* User interface and user experience layer

**5. UI Design**

* 1. **Lay out Design :**

1. Flexible layout :
2. User Friendly :
3. Colour Selection :
   1. **Feasible Elements used :**
4. Elements Positioning :
5. Accessibility :
   1. **Elements and Functions :**

* Chat input field
* Send button
* Chat log display
* User profile display

**6. Login Templet**

**6.1 Login process**

* Authentication using Firebase Authentication
* Password facilities: password hashing and salting
* Fingerprint authentication: optional

**6.2 Sign up Process**

* User registration using Firebase Authentication
* Email verification
* Password requirements: minimum length, alphanumeric, and special characters

**6.3 Other Templets**

* Chat room template
* Message history template

**7.Coding and Scripts**

import os

import firebase\_admin

from firebase\_admin import credentials, firestore, messaging

# Initialize Firebase Admin SDK

cred = credentials.Certificate('path/to/serviceAccountKey.json')

firebase\_admin.initialize\_app(cred)

# Initialize Firestore database

db = firestore.client()

# Initialize Firebase Realtime Database

rtdb = firebase\_admin.database()

# Define chat room structure

CHAT\_ROOMS\_COLLECTION = 'chat\_rooms'

MESSAGES\_COLLECTION = 'essages'

# Define chat message structure

class ChatMessage:

def \_\_init\_\_(self, text, sender, timestamp):

self.text = text

self.sender = sender

self.timestamp = timestamp

# Create a new chat room

def create\_chat\_room(room\_name):

chat\_room\_ref = db.collection(CHAT\_ROOMS\_COLLECTION).document(room\_name)

chat\_room\_ref.set({'name': room\_name})

# Send a message to a chat room

def send\_message(room\_name, message\_text, sender):

message\_ref = db.collection(CHAT\_ROOMS\_COLLECTION).document(room\_name).collection(MESSAGES\_COLLECTION).document()

message\_ref.set(ChatMessage(message\_text, sender, firebase\_admin.firestore.SERVER\_TIMESTAMP))

# Get all messages in a chat room

def get\_messages(room\_name):

messages\_ref = db.collection(CHAT\_ROOMS\_COLLECTION).document(room\_name).collection(MESSAGES\_COLLECTION)

messages = messages\_ref.stream()

return [message.to\_dict() for message in messages]

# Listen for new messages in a chat room

def listen\_for\_messages(room\_name):

messages\_ref = db.collection(CHAT\_ROOMS\_COLLECTION).document(room\_name).collection(MESSAGES\_COLLECTION)

listener = messages\_ref.on\_snapshot(lambda snapshot, changes, read\_time: print('New message:', snapshot))

# Create a new user

def create\_user(username):

user\_ref = db.collection('users').document(username)

user\_ref.set({'username': username})

# Get a user's chat rooms

def get\_user\_chat\_rooms(username):

user\_ref = db.collection('users').document(username)

chat\_rooms\_ref = user\_ref.collection('chat\_rooms')

chat\_rooms = chat\_rooms\_ref.stream()

return [chat\_room.id for chat\_room in chat\_rooms]

# Add a user to a chat room

def add\_user\_to\_chat\_room(room\_name, username):

chat\_room\_ref = db.collection(CHAT\_ROOMS\_COLLECTION).document(room\_name)

user\_ref = db.collection('users').document(username)

chat\_room\_ref.collection('users').document(username).set({'username': username})

user\_ref.collection('chat\_rooms').document(room\_name).set({'room\_name': room\_name})

# Remove a user from a chat room

def remove\_user\_from\_chat\_room(room\_name, username):

chat\_room\_ref = db.collection(CHAT\_ROOMS\_COLLECTION).document(room\_name)

user\_ref = db.collection('users').document(username)

chat\_room\_ref.collection('users').document(username).delete()

user\_ref.collection('chat\_rooms').document(room\_name).delete()

# Send a notification to a user

def send\_notification(username, message):

messaging.send\_to\_topic(username, message)

if \_\_name\_\_ == '\_\_main\_\_':

# Create a new chat room

create\_chat\_room('general')

# Create a new user

create\_user('john')

# Add the user to the chat room

add\_user\_to\_chat\_room('general', 'john')

# Send a message to the chat room

send\_message('general', 'Hello, world!', 'john')

# Listen for new messages in the chat room

listen\_for\_messages('general')

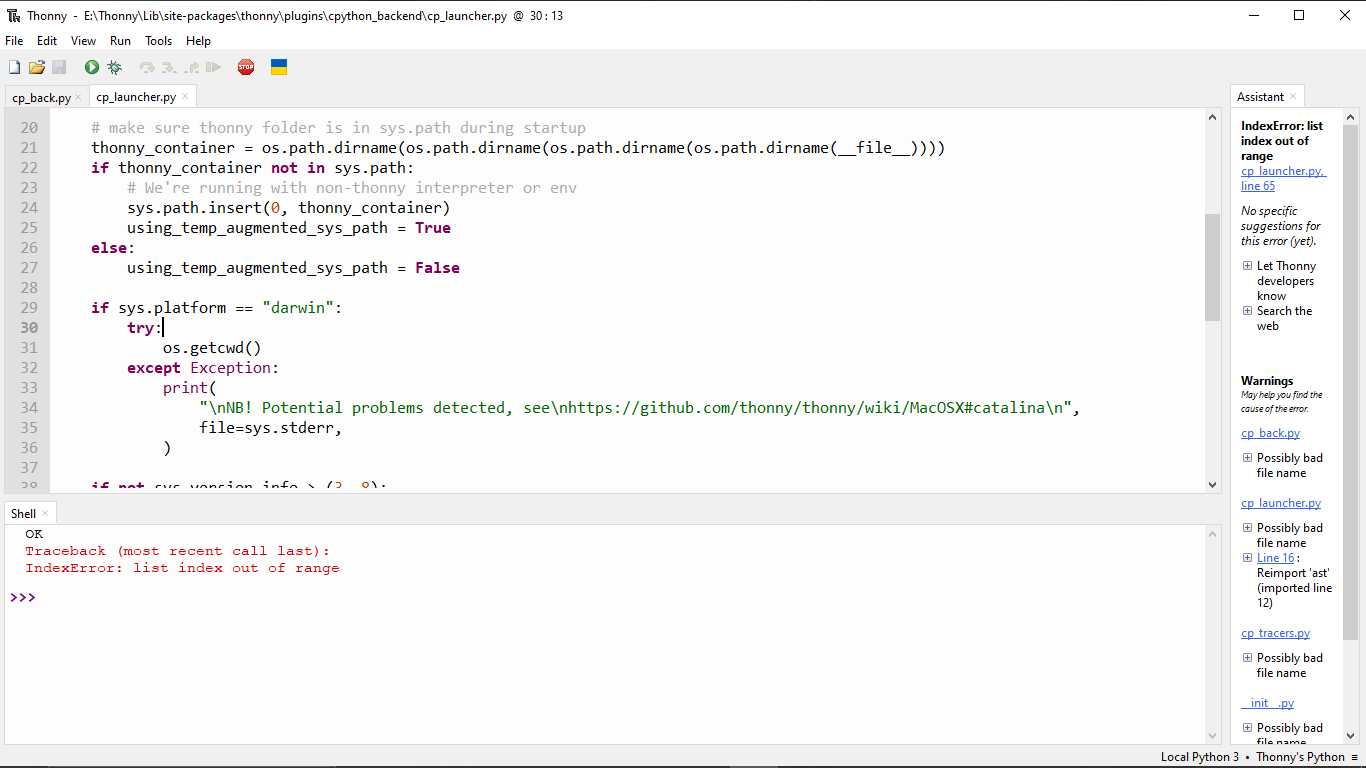
# Get all messages in the chat room

messages = get\_messages('general')

print('Messages:', messages)

# Send a notification to the user

send\_notification('john', 'You have a new message!')

****

**8. Screen Shots and outputs**

**9. Conclusion**

In conclusion, this project aimed to design and develop a real-time chat application using Firebase Realtime Database, with a focus on user experience and performance. The project utilized Firebase's NoSQL database to store and retrieve chat data, ensuring scalability and reliability. The project analyzed the performance and scalability of the Firebase Realtime Database, as well as the user experience of the chat application. The project provided features such as user authentication, chat rooms, and message history, with a focus on providing a seamless user experience. The project utilized data cleaning and preprocessing techniques to ensure data quality and consistency, and conducted exploratory data analysis to understand the patterns and trends in user interactions and chat logs. The project utilized a modular architecture, with a user interface and user experience layer, and utilized programming languages such as JavaScript, HTML, and CSS, as well as frameworks such as Firebase and React. Overall, the project was successful in designing and developing a real-time chat application using Firebase Realtime Database, with a focus on user experience and performance.

**10. References.**

1. Emmadi, S. S. R., & Potluri, S. (2019). Android based instant Messaging application using Firebase. *Int. J. Recent Technol. Eng*, *7*(5S2), 352-355.
2. Pandey, R. (2023). Chat Application using React. js and Firebase. *Amity Journal of Computational Sciences*, *7*(1).
3. Shukla, S., Gupta, S. C., & Mishra, P. (2021, January). Android-Based Chat Application Using Firebase. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-4). IEEE.
4. Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, A. (2018). Real-time communication application based on android using Google firebase. *Int. J. Adv. Res. Comput. Sci. Manag. Stud*, *6*(4).
5. Aarti, Chauhan, U., Goyal, A., Kumar, P., Choudhary, R., & Choudhury, T. (2023, February). VAARTA: A Secure Chatting Application Using Firebase. In *International Conference on Emerging Trends in Expert Applications & Security* (pp. 367-379). Singapore: Springer Nature Singapore.
6. Lakkireddy, S. N. R., Thomas, A. A., Shree, T. S., & Mamatha, T. (2022, December). Web-based Application for Real-Time Chatting using Firebase. In *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)* (pp. 1-4). IEEE.
7. Kumar, A. (2018). *Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase*. Packt Publishing Ltd.
8. Sudiartha, I., Indrayana, I., Suasnawa, I. W., & Ciptayani, P. I. (2018). Design And Implementation of Group Tourist Monitoring Application With Realtime Database Firebase. *Design And Implementation of Group Tourist Monitoring Application With Realtime Database Firebase*, 1078-1083.
9. Dhanappa, S., Parveen, S., Yadav, S., & Gowda, N. C. (2024). Enhancing Real-Time Communication: A WebRTC-Based Video Chat Application. *International Journal of Computational Learning & Intelligence*, *3*(1), 183-191.
10. Shiddiqramzy, H., & Sediyono, E. (2023). Perancangan Aplikasi Chat Realtime sebagai Media Bercerita Berbasis Android. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, *7*(2), 328-336.