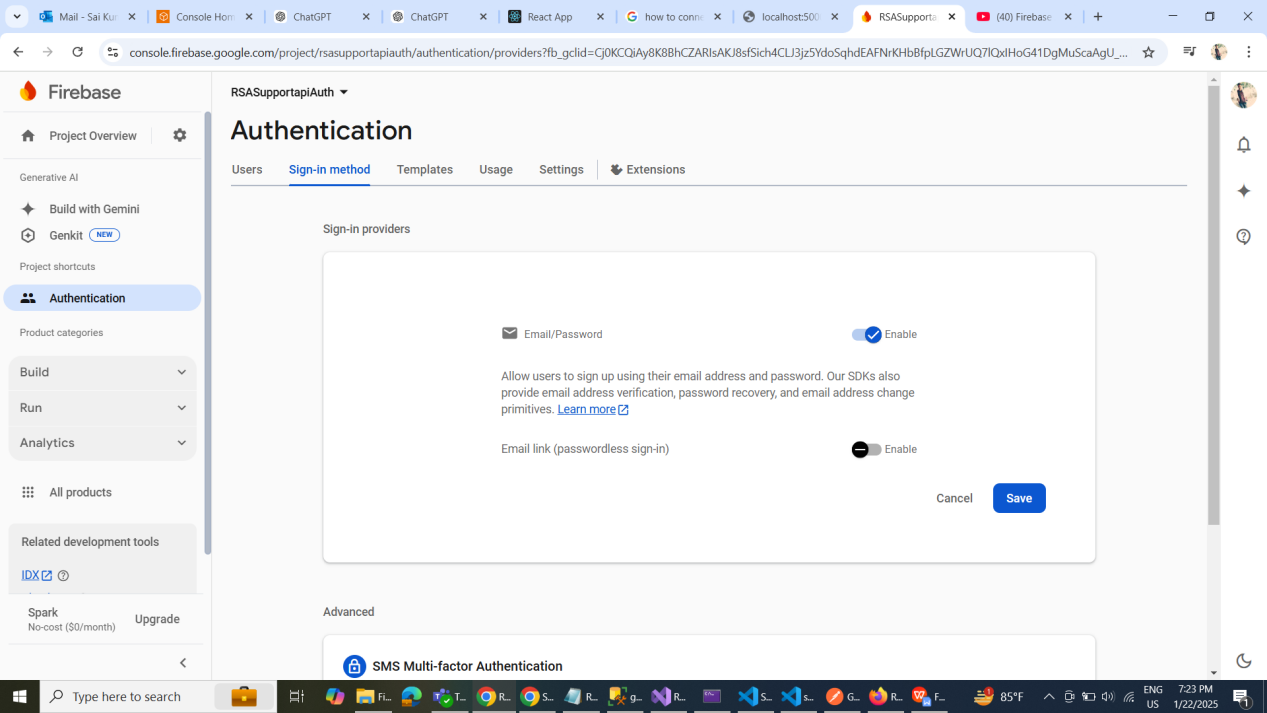
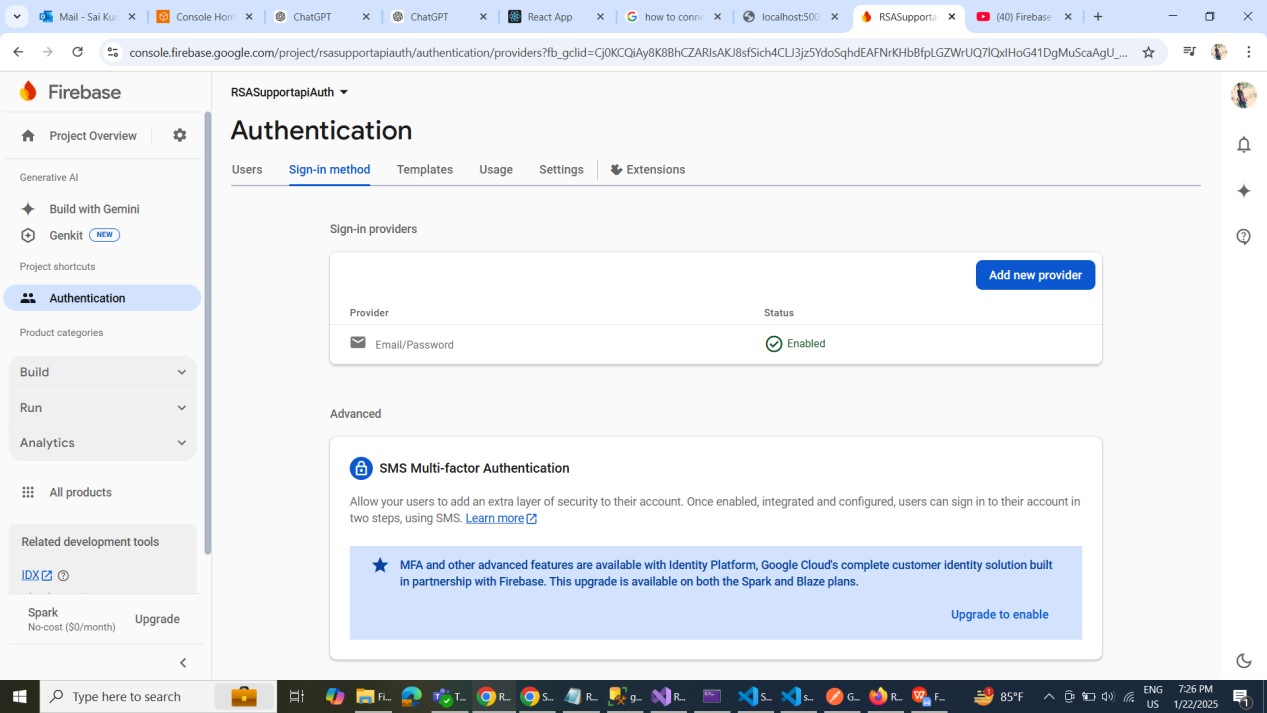
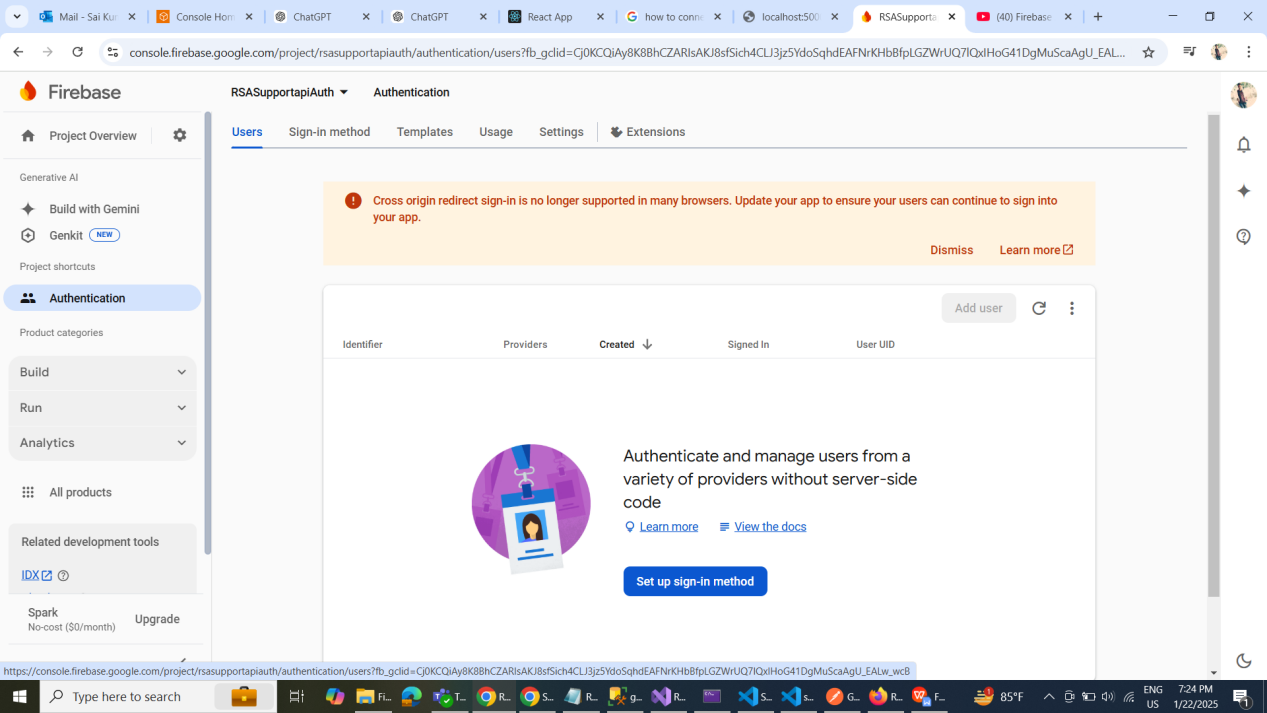
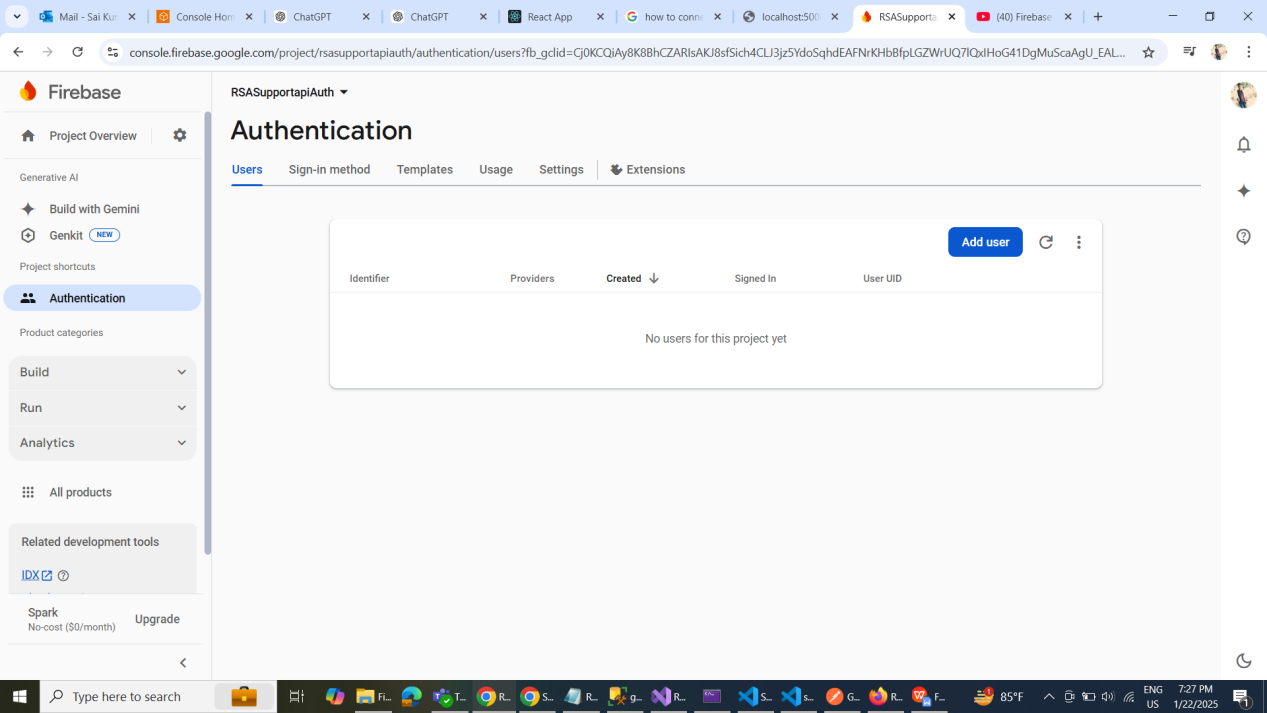
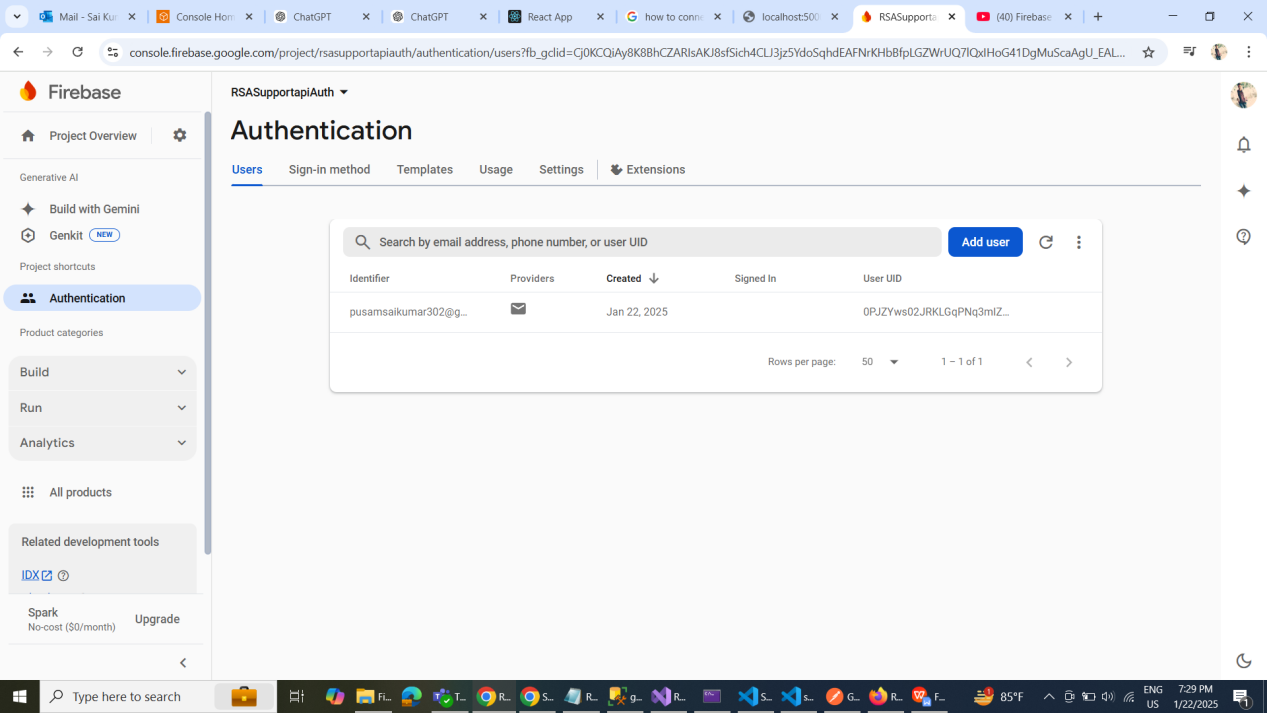
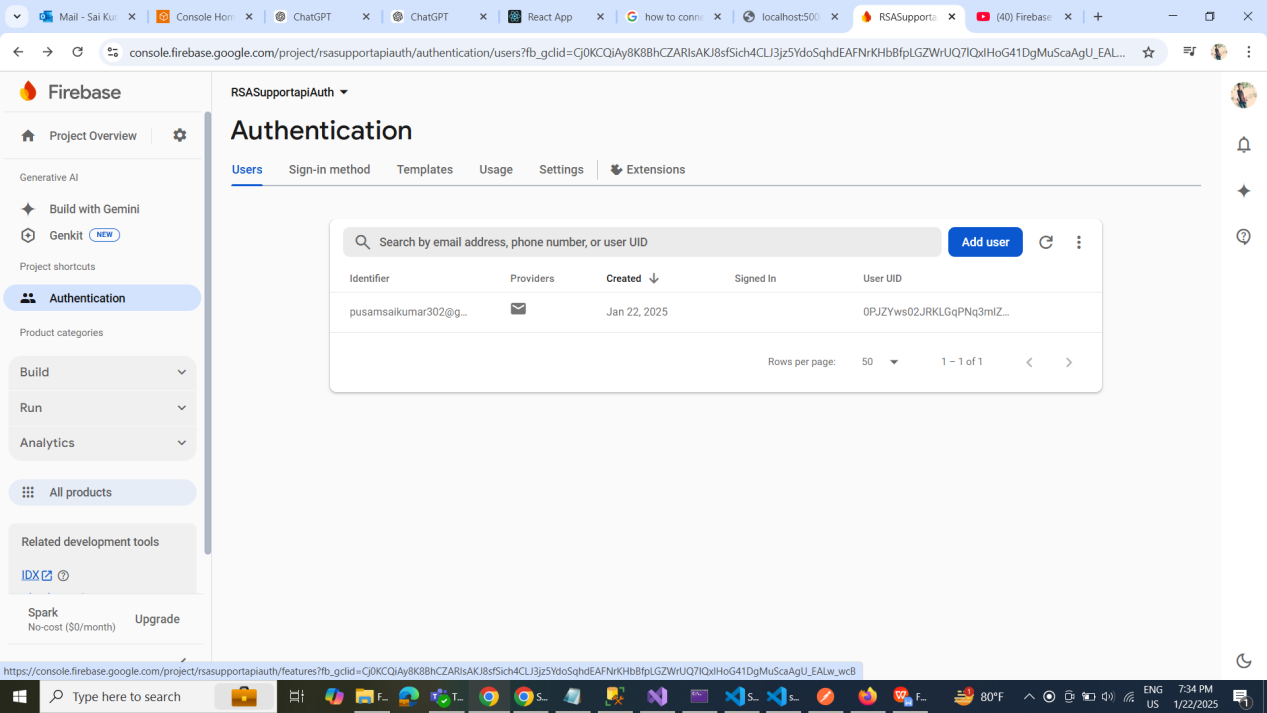
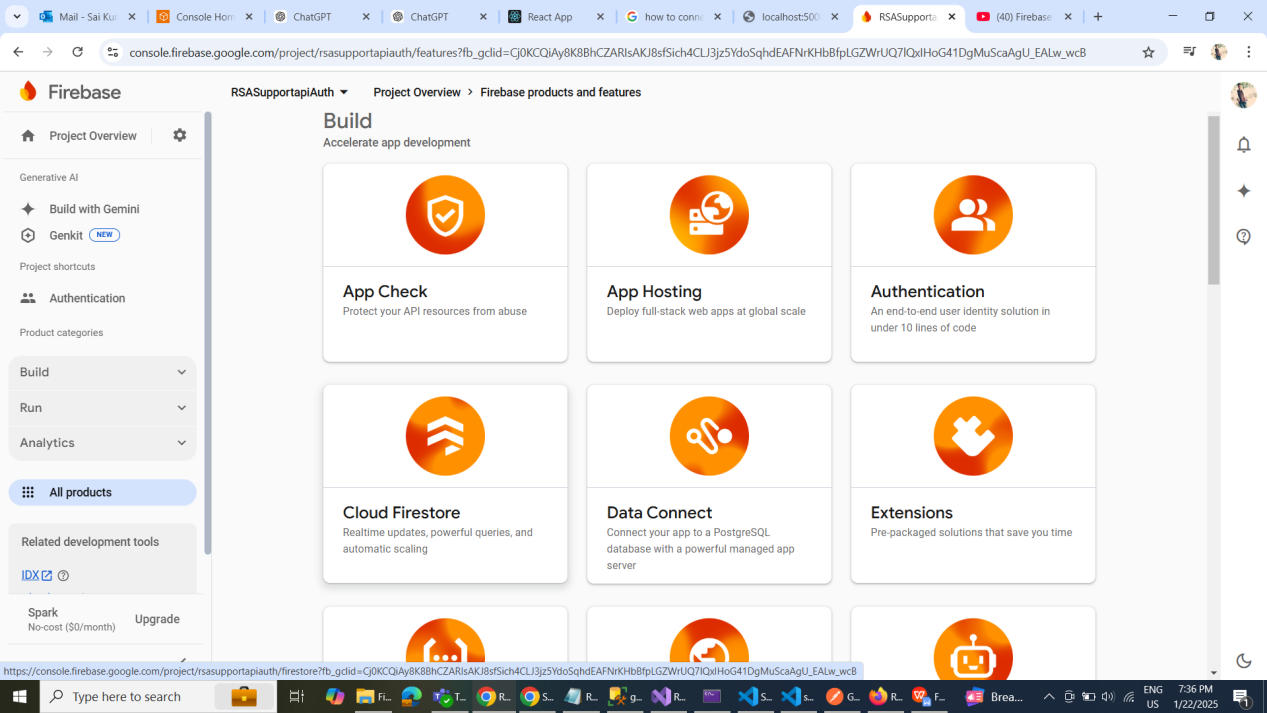
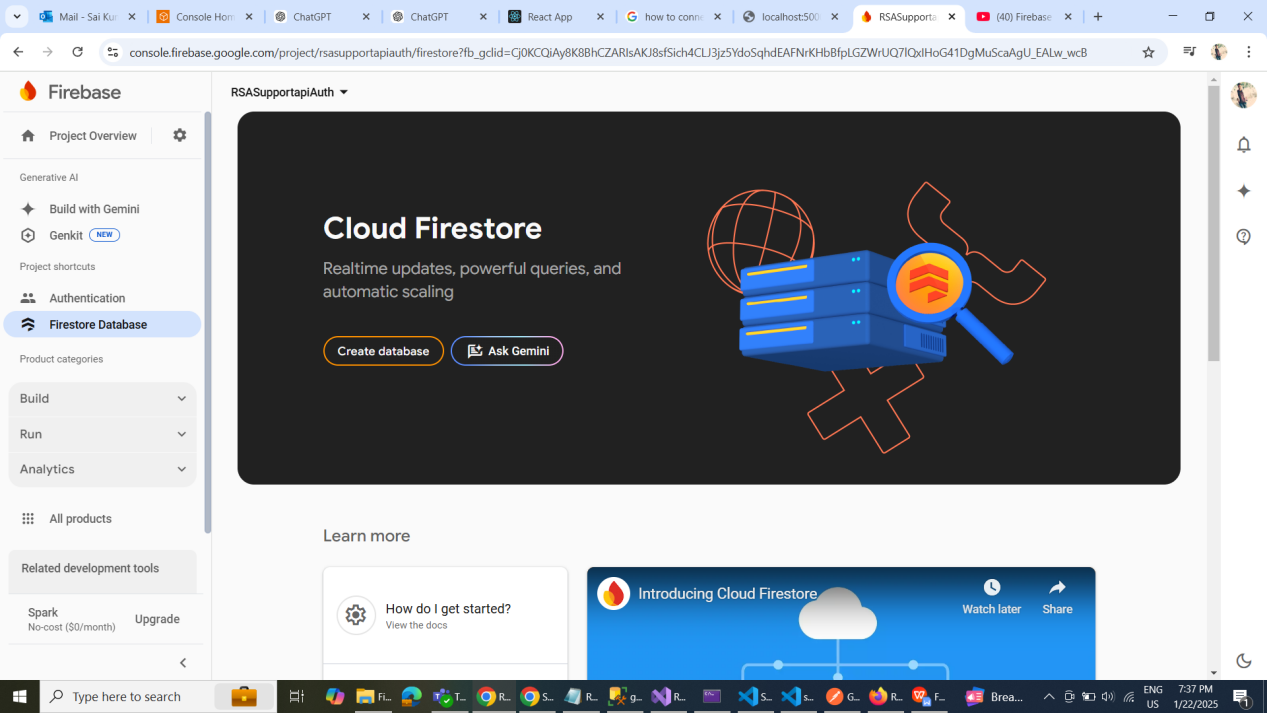
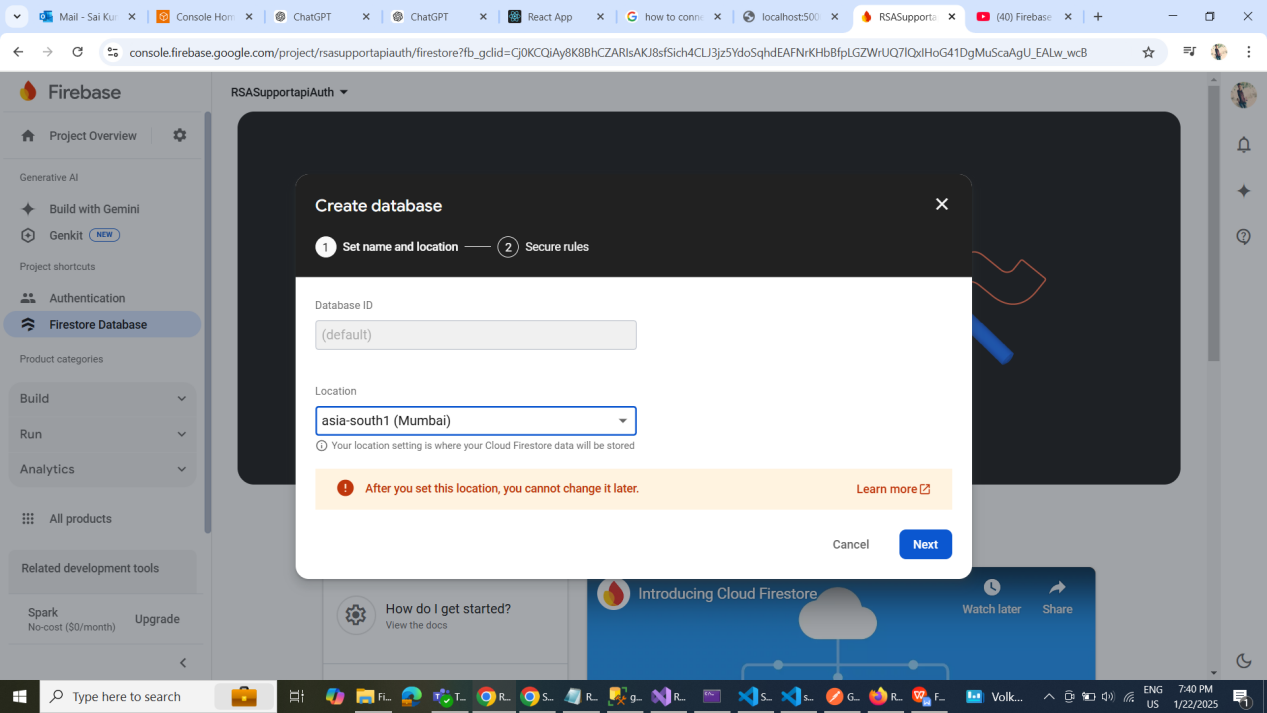
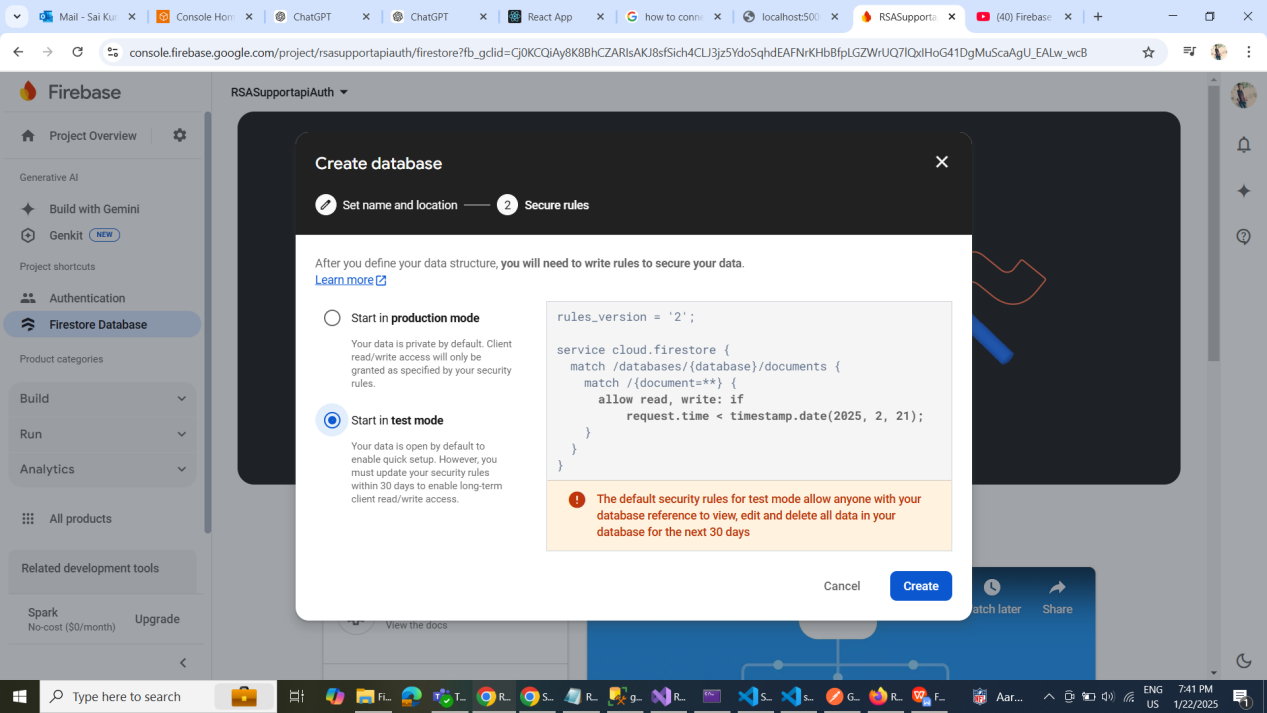
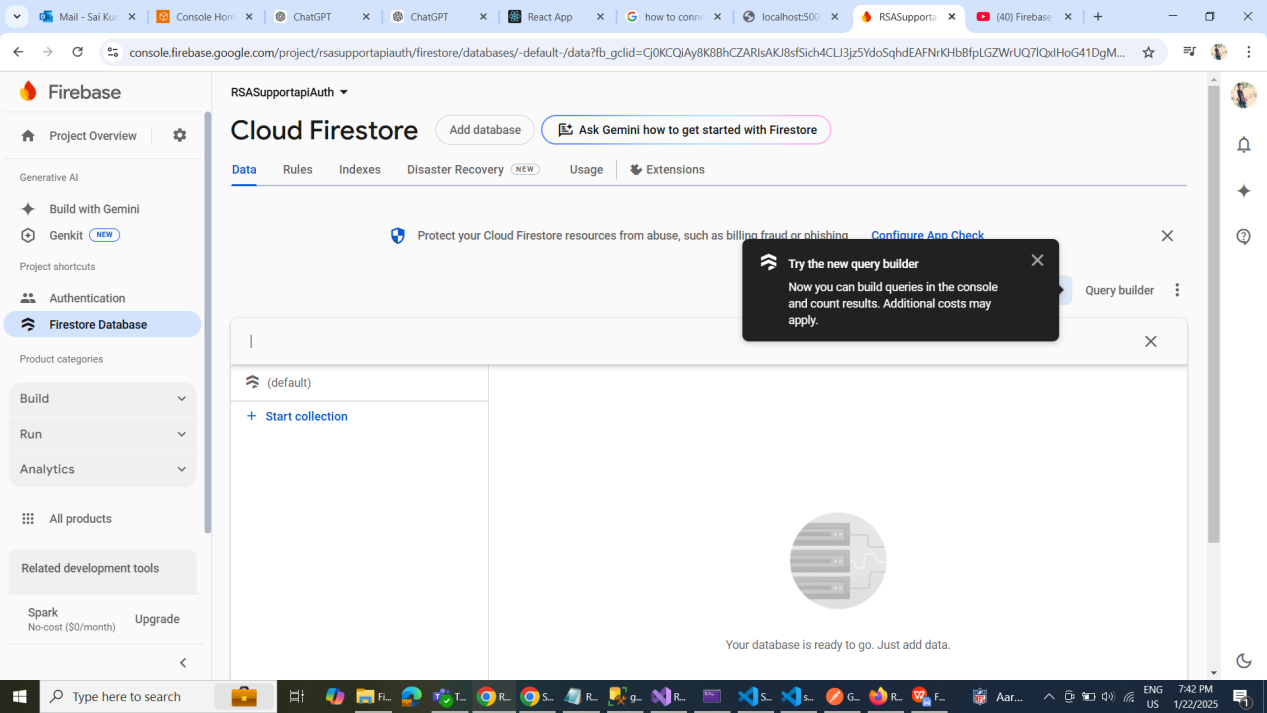
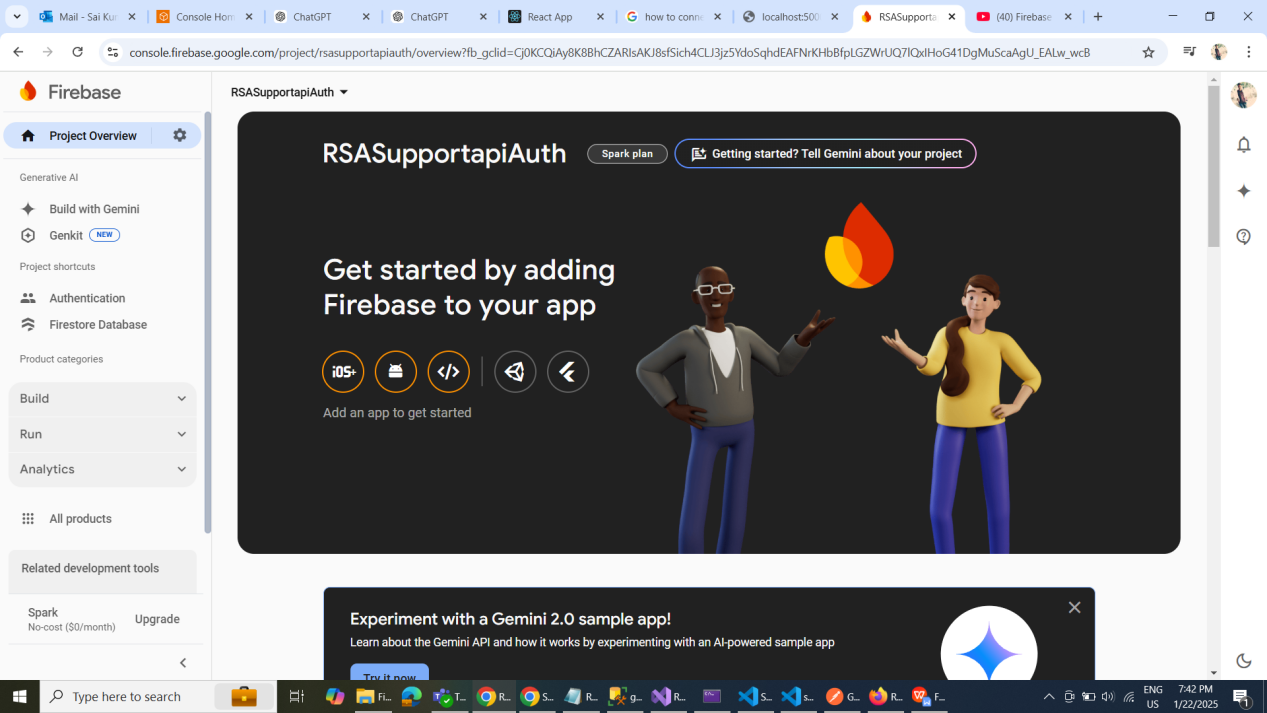
FIRE BASE AUTHENTICATION NODE JS AND REACT JS

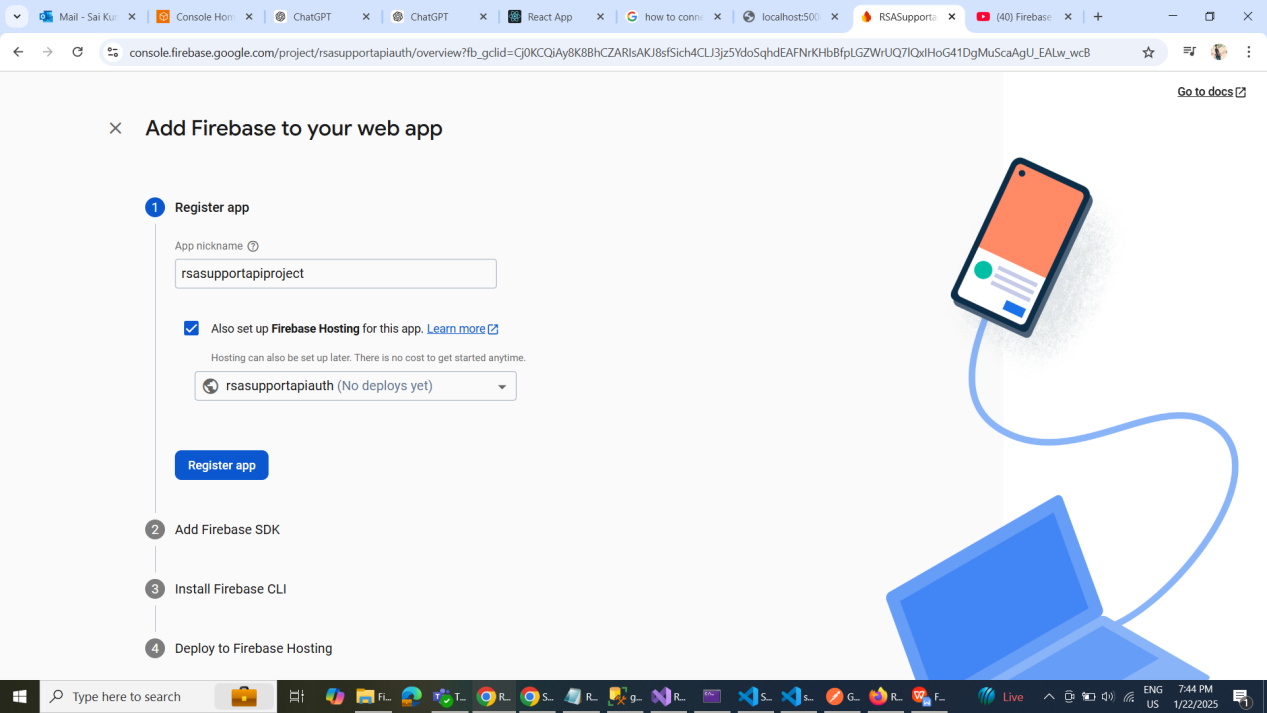
Create project:   
1. PLEASE CREATE PROJECT NAEM  
2. ENABLE Signin Method  
  

  
3.Sign up with users   
  


  
  
email: [pusamsaikumar302@gmail.com](mailto:pusamsaikumar302@gmail.com) password:S@1kumar  
USER UID: 0PJZYws02JRKLGqPNq3mlZPtDFq2  
  
create cloudFire stores: go to allProducts ------ > CloudFire -----> create database  
  
  
CLoubFirestore:

  
  
create database:

  
  
  
  
  
  
  
4. Go to project review:  
accessing web projects: Nodejs and React Js  


  
  
after Register.  
  
npm install firebase --- for node js and react js  
  
  
// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import { getAnalytics } from "firebase/analytics";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

apiKey: "AIzaSyCG9NFCfaRoL1DFZCTBdOkICqvk\_KJEnf4",

authDomain: "rsasupportapiauth.firebaseapp.com",

projectId: "rsasupportapiauth",

storageBucket: "rsasupportapiauth.firebasestorage.app",

messagingSenderId: "1073437246550",

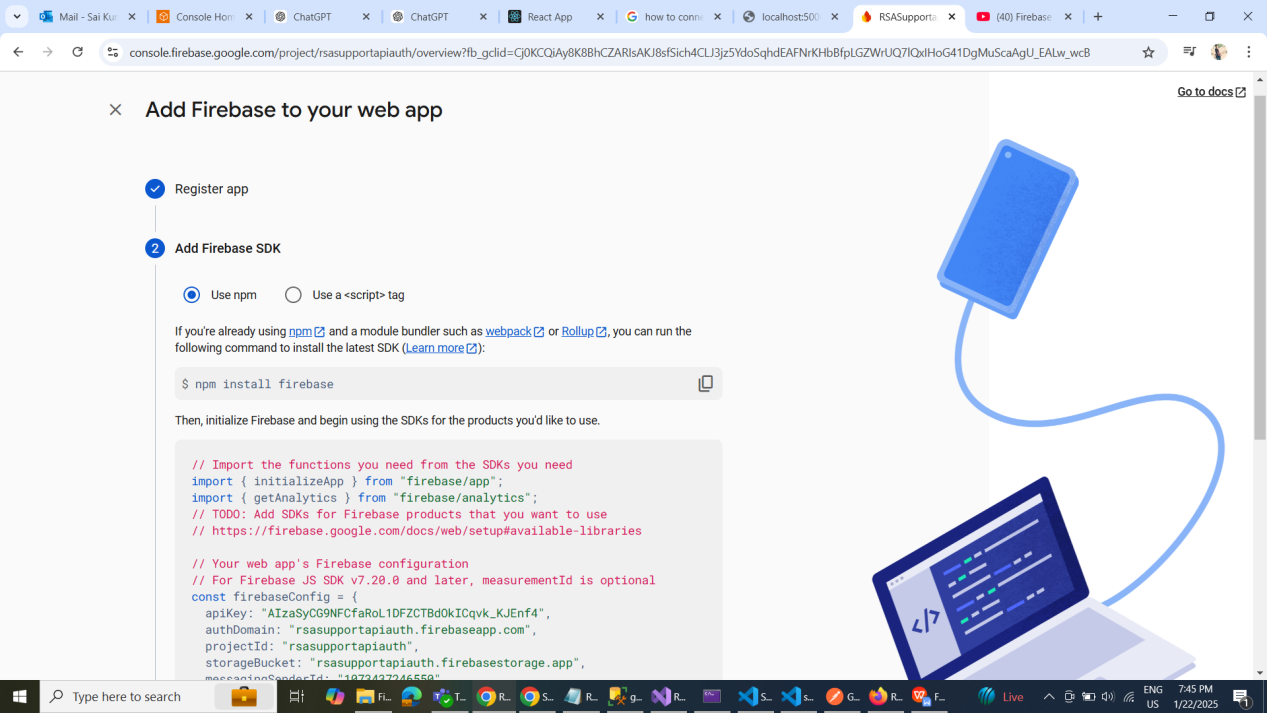
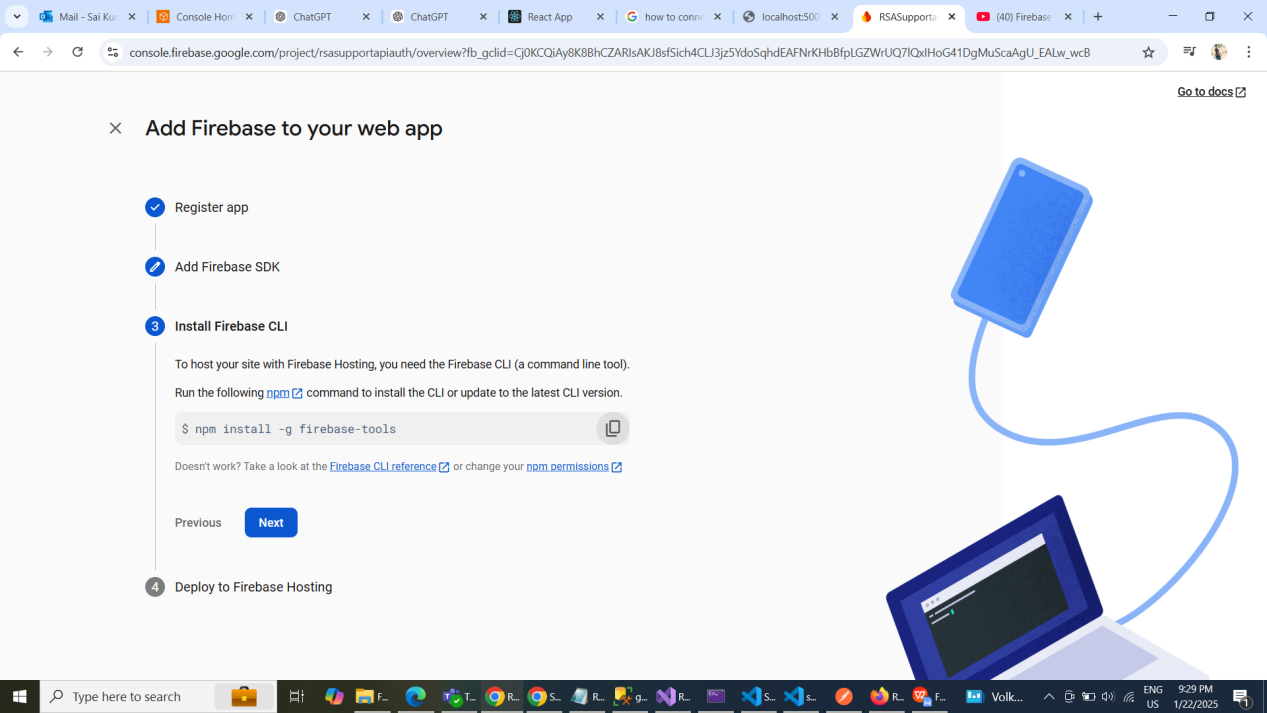
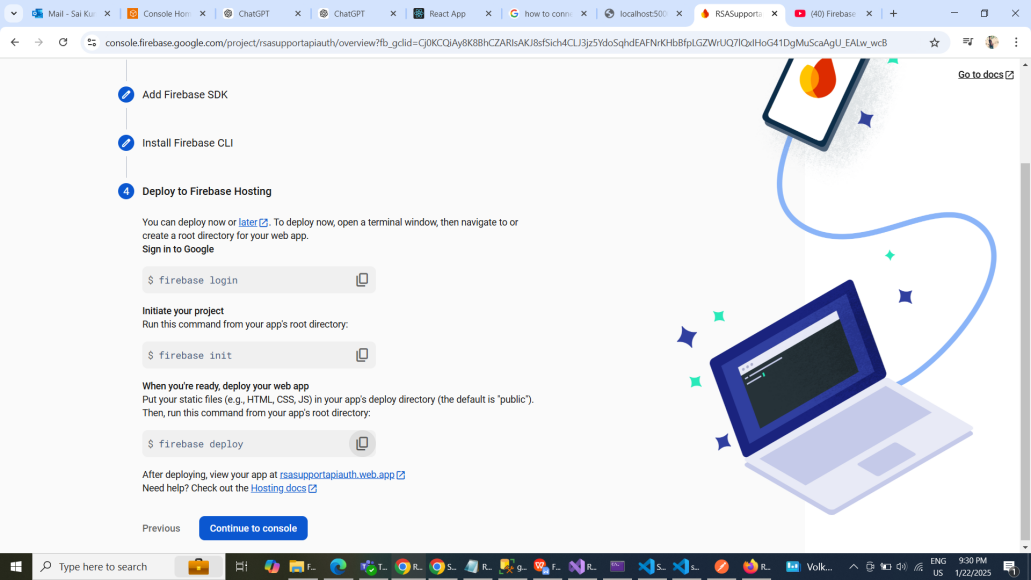
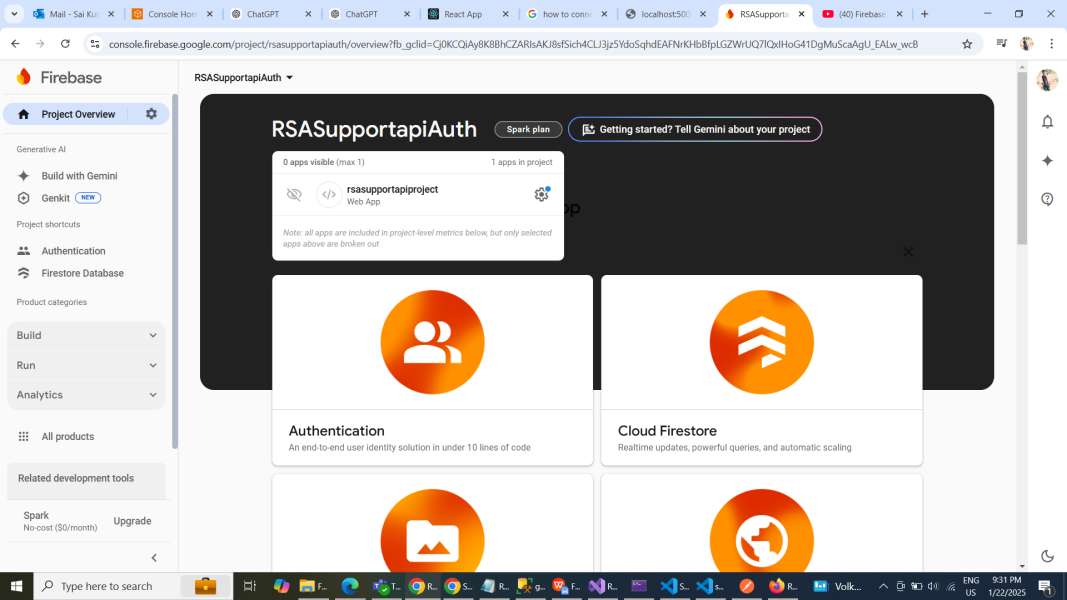
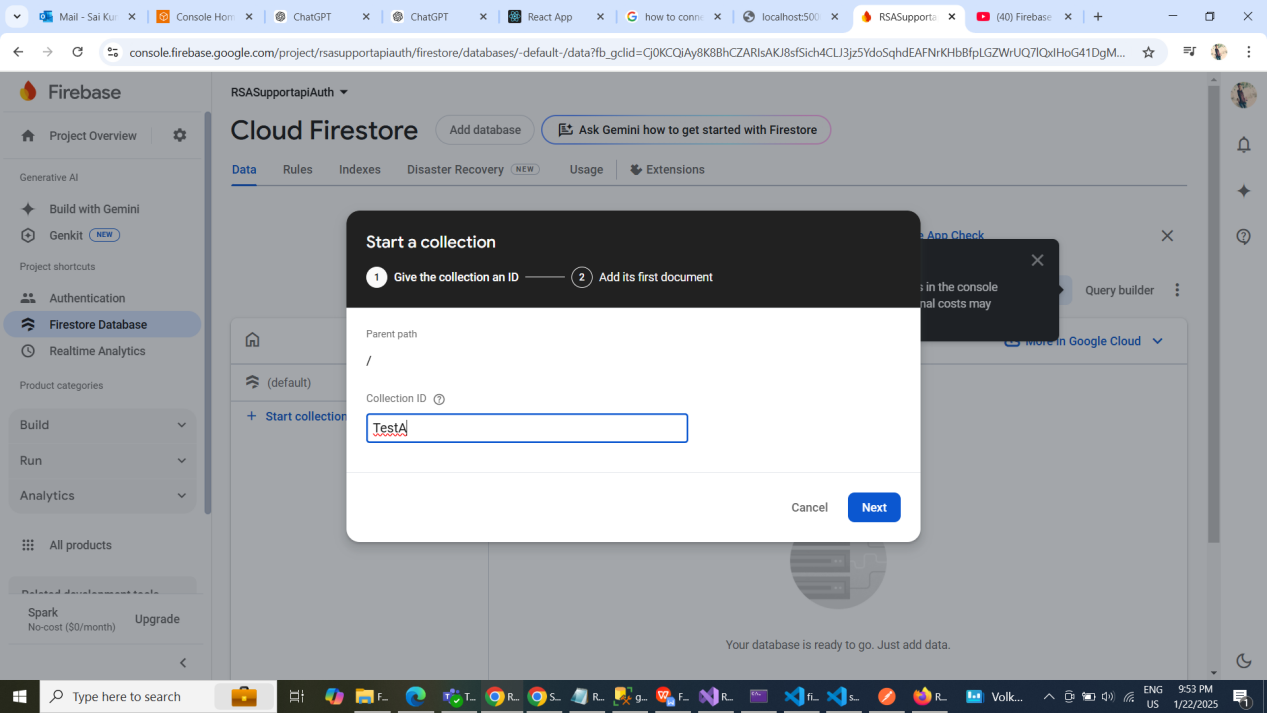
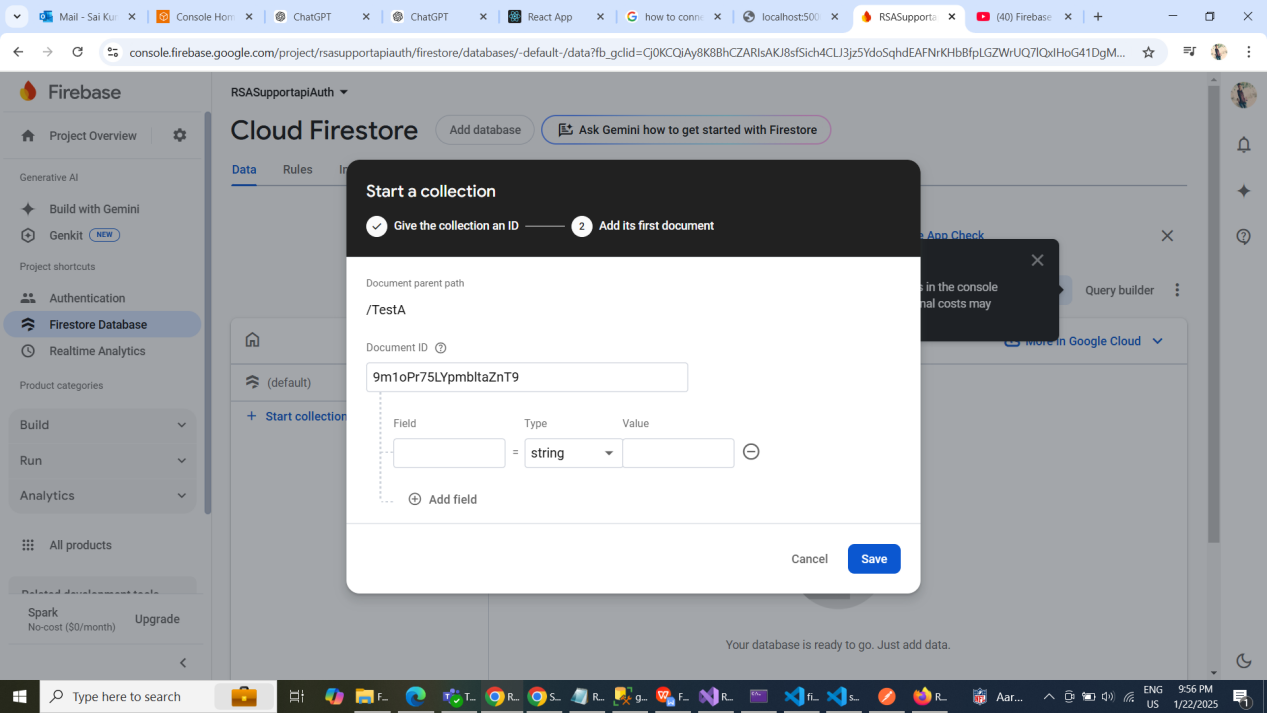
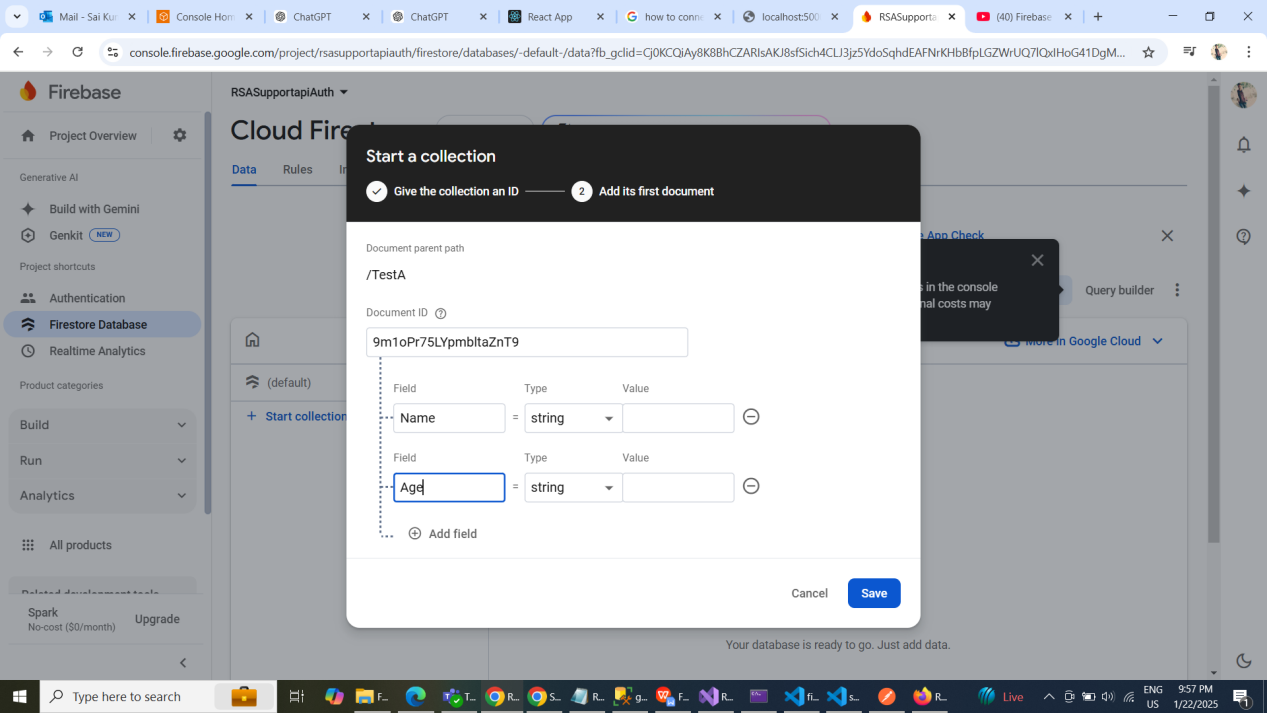
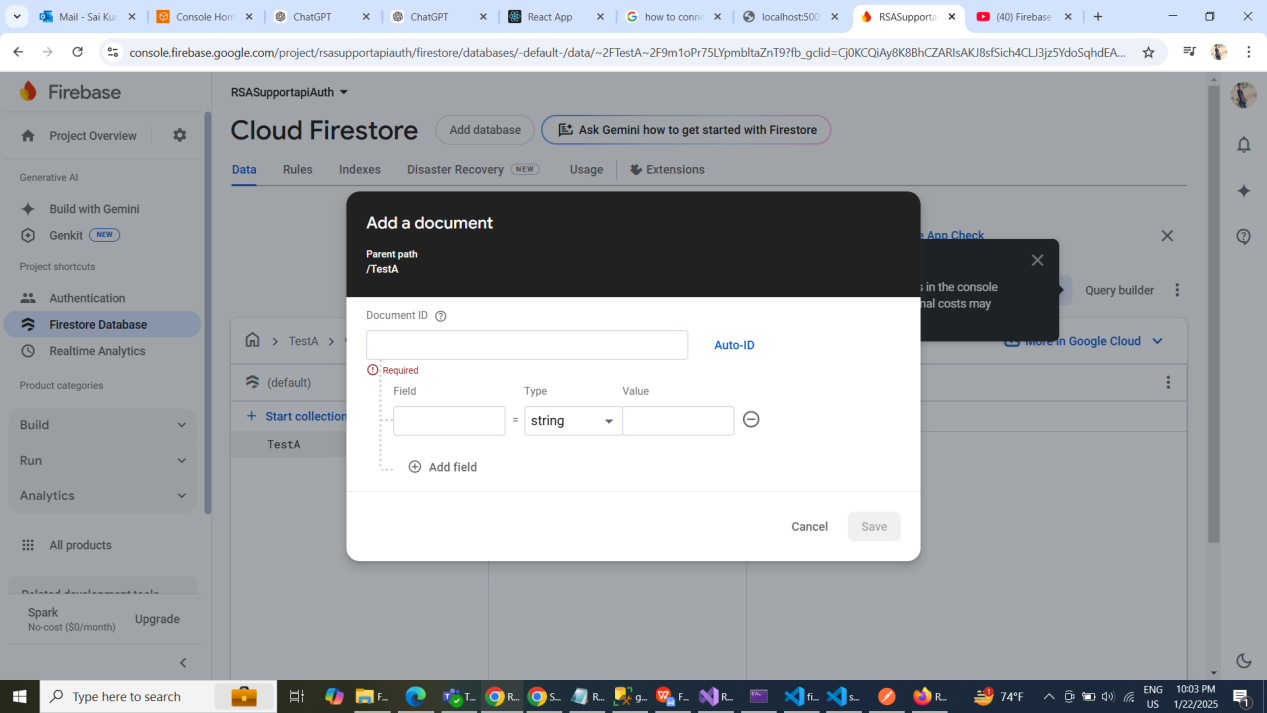
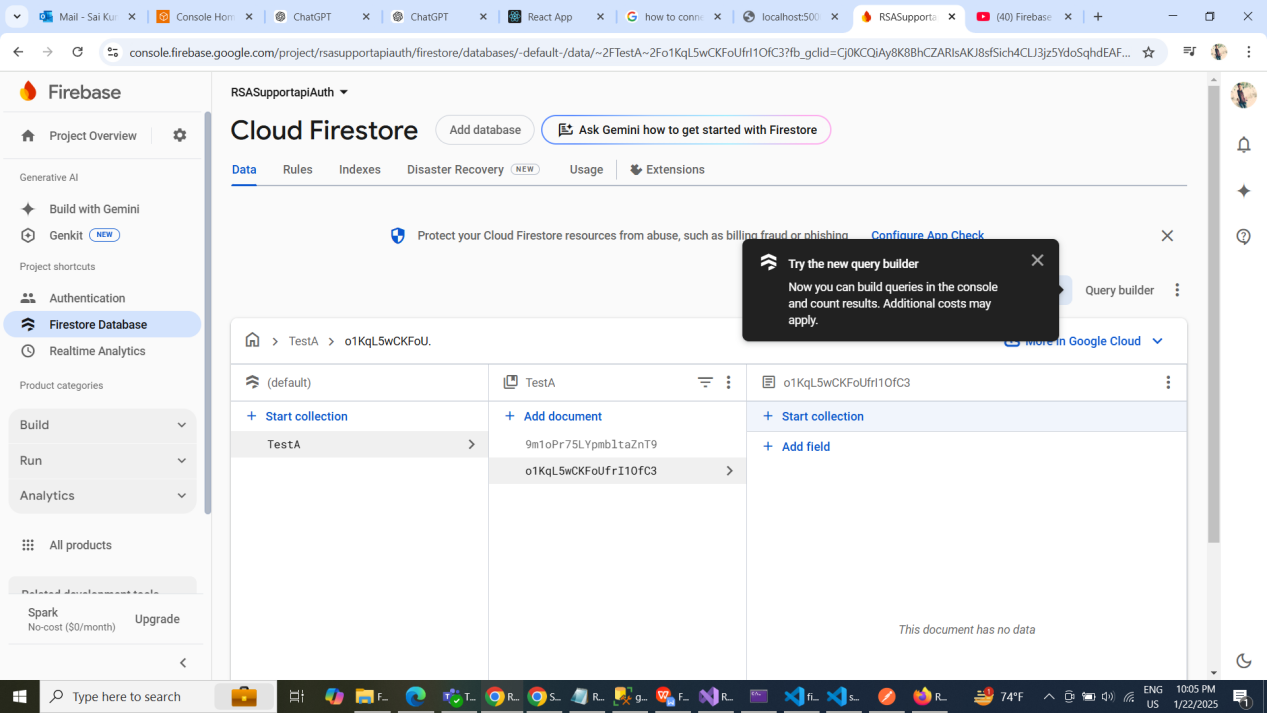
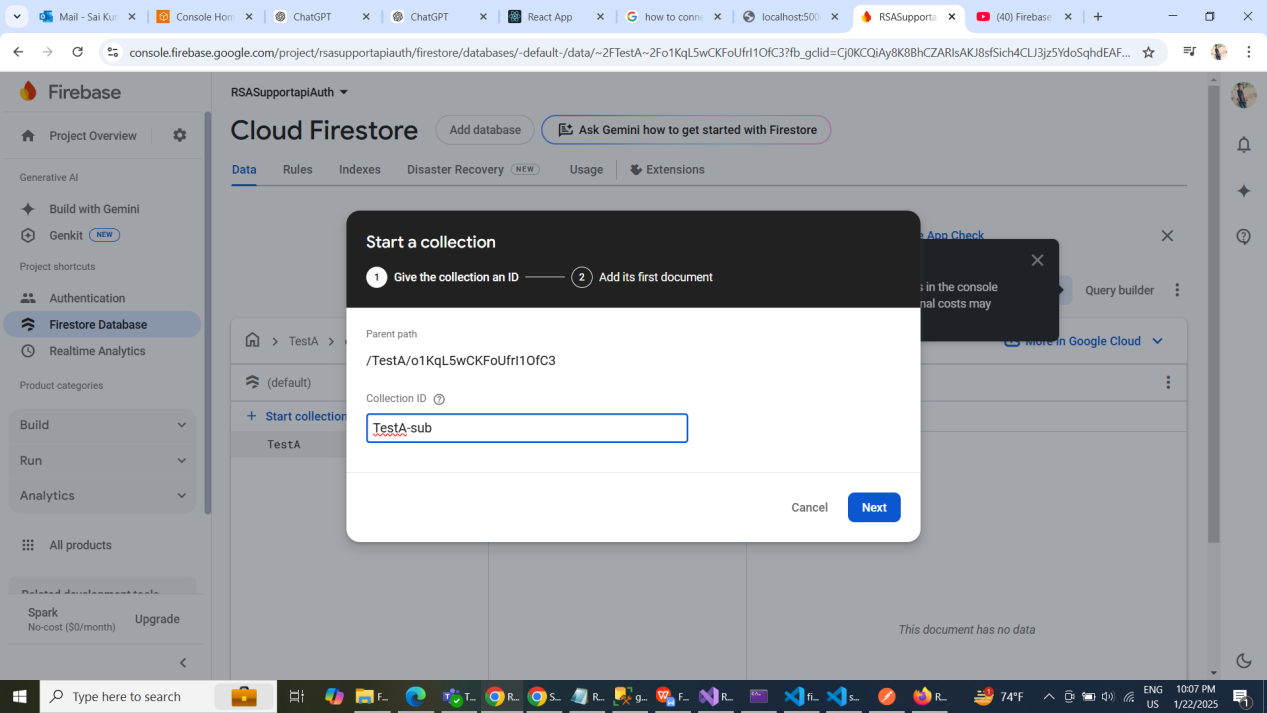
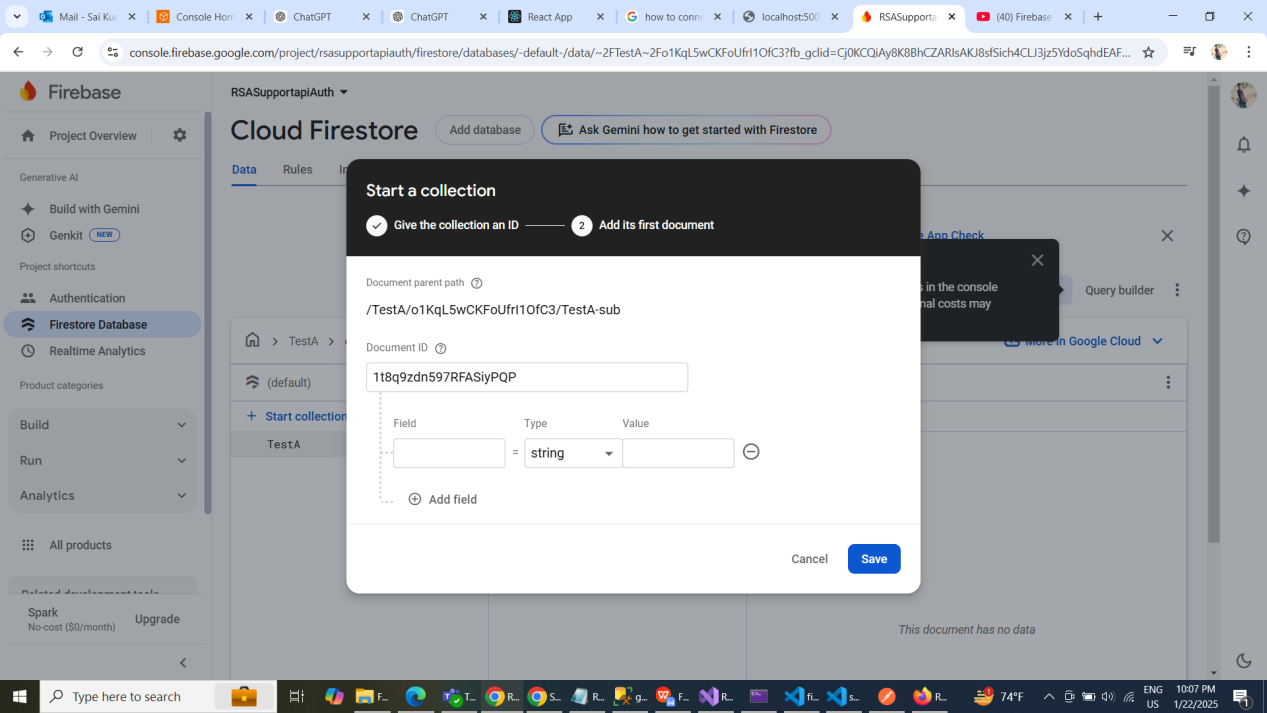
appId: "1:1073437246550:web:2b9796f834905ca43567f7",

measurementId: "G-3BDZEQ8CLE"

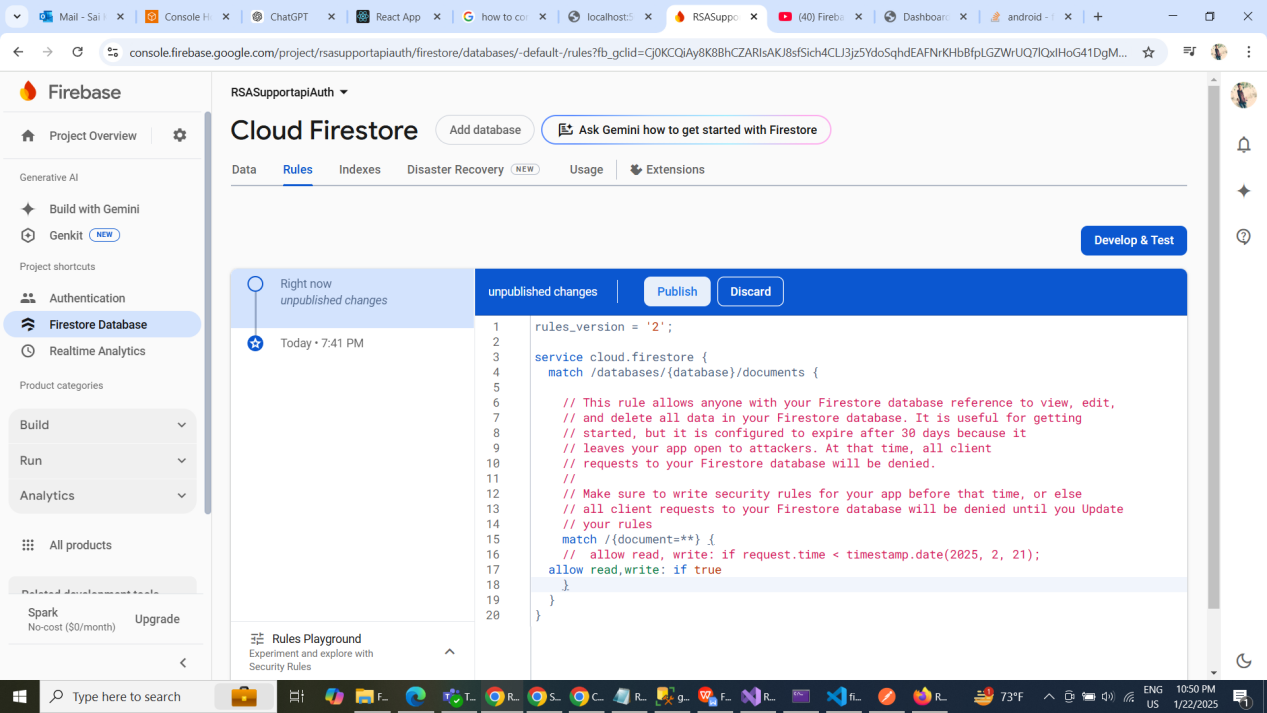
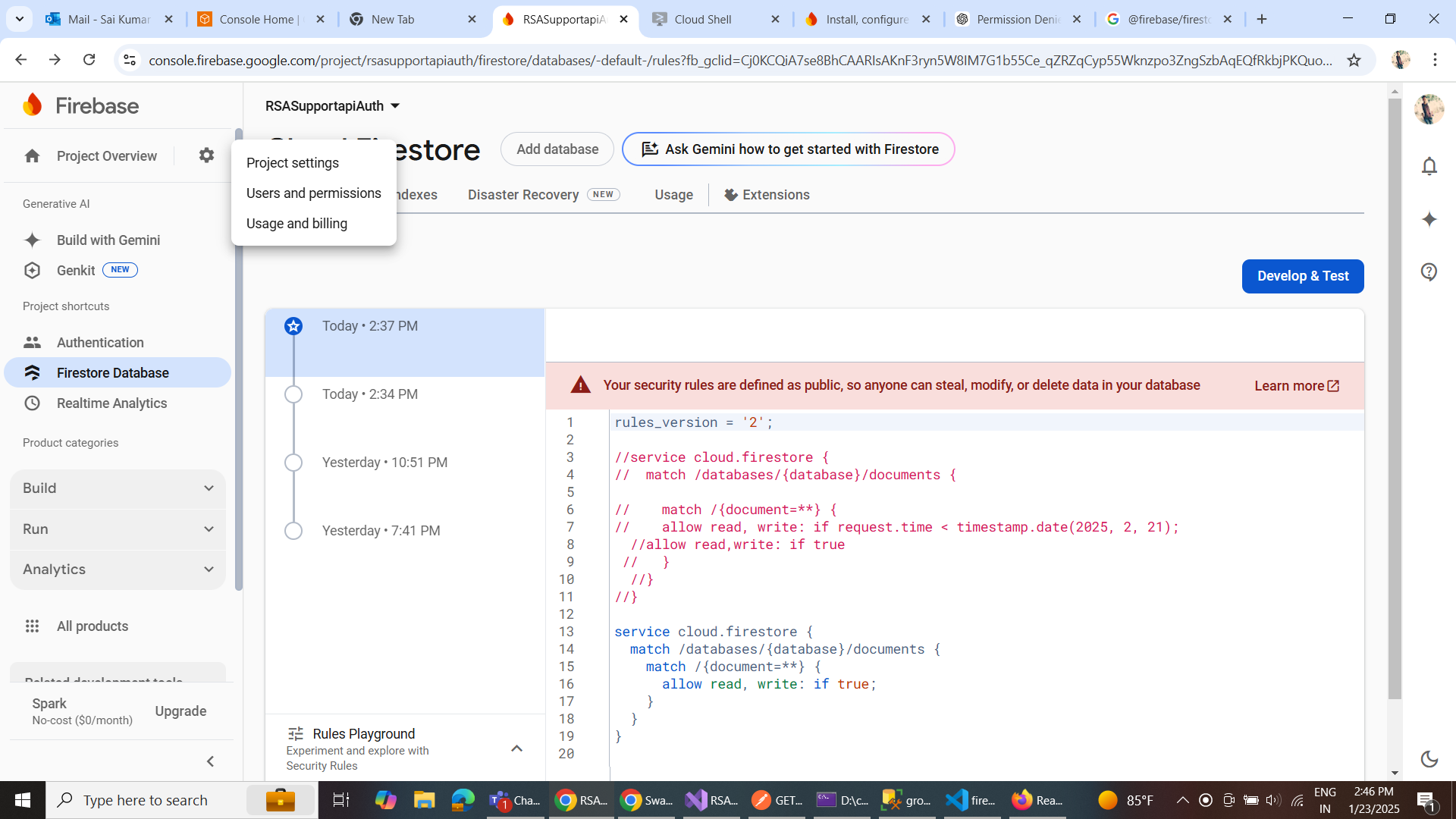
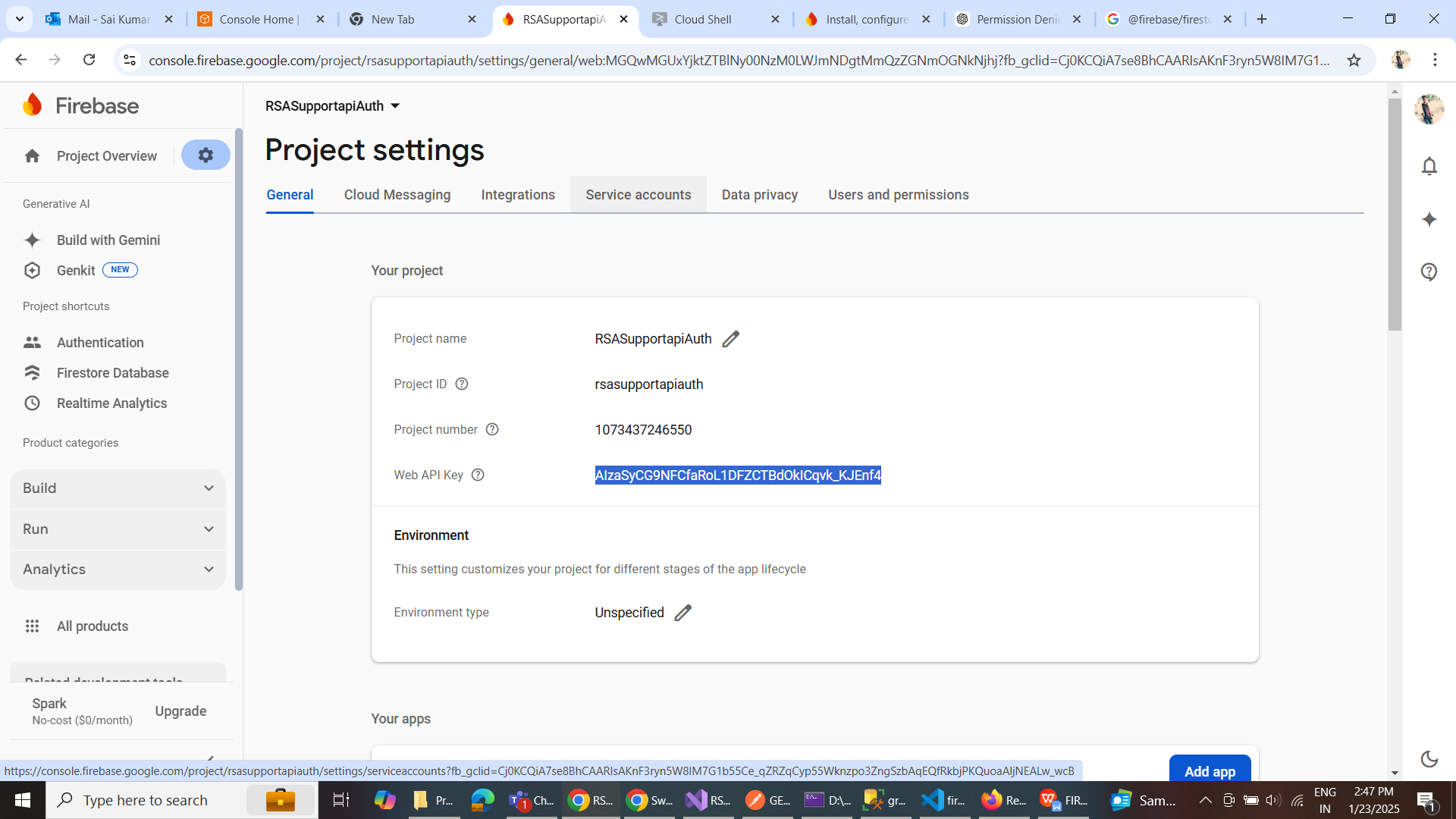
};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);  
  
  
  
  
install firebase CLI:  
  
npm install -g firebase-tools  
  
  
  
Deploy to Firebase Hosting:  
firebase login  
firebase init  
firebase deploy  
  
  
  
Create collection : database  
  
  
  
  
  
  
create sub data collection inside TestA: Click Add Document  
  
Click on Auto-ID:  
  
  
Save  
  
Create Nested collection : start collection --- -under added document  
  
  
  
  
  


Change Cloud rules:

False to true ------- go to Publish  
  
  
Go service account to generate new service secret json file :  
go to project settings:  
  
  
service accounts:  
  
  
Generate new secret key:  
  
  
  
{

"type": "service\_account",

"project\_id": "rsasupportapiauth",

"private\_key\_id": "da1701702da7f7df5327c2a26209faf03816eec7",

"private\_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDO+fSUwwXNogMp\n0UHuwXKjekVvdGk7xEobGAlzMwYLsJ8nGJCDhnJ0lKL2j628cBdKusuI3zV5BL2A\nPJOyf6UtHBXJPqniagQrfbFp1tadSSRQ20VJIXorU/LbyZbgdal2b0gyHGLMDdCF\nOX34+QwFsPqUJWpygtAvJ7MdIabNtnnTYeRM3BuCnGk94O7e0azB+PqOeelGh8WZ\nP5UfJwfn0ltGcRw2mzaOyKR7mkDYovd3yVEDIaeBWrpFSb1Mb8Iy6ZkkbZwNMCt7\nppYVK0WTh/sA/nM7/q7pTc6KyHoA6b/25jTlbAdxLKDqk3E/N7yt53PnpqRZdND3\n6h4DymArAgMBAAECggEAOBS/Ii3DnUeoPsTt7gDpjIUdWkk7GFAj2jqFmPq6ySEW\nxpOf2Jtt2MvLai5dH2Fp1jxN2OzOAtf3ds8zH9r5SQNCnfBGPMyIC7a3gxuMtDp0\nxhnbqKiS/zl0ZPw2zuPiwm39rGJHnM2jce21hegJS2N8SfbnZFawjQ5fOGX9WeB5\n1U9i+ogIm2tIy+SJ/mOyIHIkOETfrxlBn5ZfakvLHTOYSNz+n2bwoMGDKm4so8N2\n8MRaOxcKZ5TrpJYlNueSk0SlbFQPNjtNW+l5rnGaNh6IiLTTbzsATYfsCiAH+xoq\ny8ehAPdWaylZM/7bGxtH58AFLjYUeoPPmC7MAIz4XQKBgQD8adwGEwj1lvaH9Amr\ntBDDJeg3Bauwl3mQn9p8Vux2630NqZenTFN5bcpwH2YEcJDozAHAiSbIJ6MYrBAv\nu+50mdCwBZgeZas4C29ND0lirI/B9KcZNArxDzFyGgxF8d6mFiD9ui+QT2+CYwtS\nBB5cbxViRYeL1bJaXiWpPdNapwKBgQDR6tIiVJEvBnFHTclAUEbescBLyRZN57wP\nD6saX2sQhyzLDP+GBZHU5+5A/WOvyhf46ct8VJC9Gjj8nAR3oQ7or2sAlz2Gykqm\nR0hpH1LDzYGz+yFH4UQ6ZwOMpB1GYylDAkZ4IMWrFrL41wI8KkqLBpce/J3vpIf+\n7gpQyOuy3QKBgFZxOnSqGCtjI9KzdIxlAkiSttABFSdI+MzascxbSGgBhbYkGlel\nAAFZ5DY5yopCsp3HH+J8Fdf+GxDb/D8csTElbB6cC9h9IQH0iI1vasZsKl9WFdO/\nyu97MIfO+90IVCO0+F9sX9WgHqQcn0fbOTqWoFAjV6Q/7vCbxXIMRlGJAoGAFKmr\no4t1ew6ESgODItyEDVmr1wnlMyfzX/wn2IG59TeJFjeiU1vq/e0Si70leeFHuM2y\npPth/+s73smwuuHUqNzTiIxalxEEhheQLiDSMcnh0YiPvxaSc7k+JTLmWs5Z+mmX\n7C01p+ClH09yvd9f1Q2CBJ/TIAqvBE4EH+2XHdUCgYEAo46XeDNnK0seeYCK4uRO\nDkjAaC/BF3ixBmeLBdde6H0GtfnF9JMK/OaxT7PMcDkiOCOjH3nQ1nLTbQ787UWN\nbphzSJj+gIfHt5P5YMPzeZ0lqtbb9ZVlOVaTq16W17abY0+uvdaCOVL6nLD4eUO2\nmkD0mbbDsJSzVfJ+V2ex2v4=\n-----END PRIVATE KEY-----\n",

"client\_email": "firebase-adminsdk-fbsvc@rsasupportapiauth.iam.gserviceaccount.com",

"client\_id": "110348175577910417642",

"auth\_uri": "https://accounts.google.com/o/oauth2/auth",

"token\_uri": "https://oauth2.googleapis.com/token",

"auth\_provider\_x509\_cert\_url": "https://www.googleapis.com/oauth2/v1/certs",

"client\_x509\_cert\_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-fbsvc%40rsasupportapiauth.iam.gserviceaccount.com",

"universe\_domain": "googleapis.com"

}

Update doc in side database collections  
npm install firebase@latest  
  
NODE JS  
  
firebase.js

const {initializeApp} = require("firebase/app");

const {getAnalytics} = require("firebase/analytics");

const {getFirestore,doc,setDoc} = require("firebase/firestore")

const {

    FIREBASE\_APIKEY ,

    FIREBASE\_AUTH\_DOMAIN,

    PROJECT\_ID,

    STORAGE\_BUCKET ,

    MESSAGING\_SENDER\_ID,

    APP\_ID,

    MESSUREMENT\_ID

} = process.env;

const firebaseConfig = {

    apiKey: "AIzaSyCG9NFCfaRoL1DFZCTBdOkICqvk\_KJEnf4",

    authDomain: "rsasupportapiauth.firebaseapp.com",

    projectId: "rsasupportapiauth",

    storageBucket: "rsasupportapiauth.firebasestorage.app",

    messagingSenderId: "1073437246550",

    appId: "1:1073437246550:web:2b9796f834905ca43567f7",

    measurementId: "G-3BDZEQ8CLE"

    // apiKey:FIREBASE\_APIKEY,

    // authDomain: FIREBASE\_AUTH\_DOMAIN,

    // projectId:PROJECT\_ID,

    // storageBucket:STORAGE\_BUCKET,

    // messagingSenderId:MESSAGING\_SENDER\_ID,

    // appId:APP\_ID,

    // measurementId: MESSUREMENT\_ID

  };

  let app;

  let firebaseDB;

  const initializeFirebaseApp = () => {

    try{

        app = initializeApp(firebaseConfig);

        firebaseDB = getFirestore(app);

        console.log("app" , app)

        return app;

    }catch(err){

   errorHandler(err,"firebase");

   throw err;

    }

  }

  const uploadprocessFirebaseData = async () => {

    const data = {

        Name :"saikumaar",

        Age:28,

        DOB:new Date()

    }

    try{

        const document =  doc(firebaseDB,"TestB","some\_unique\_test\_id");

        let updateData = await setDoc(document,data);

        console.log("updated",updateData)

        return updateData;

    }catch(err) {

        console.log("firebase error",err);

         throw err;

    }

  }

  const getFirebaseApp = () => {

    return app;

  }

  module.exports = {initializeFirebaseApp,getFirebaseApp,uploadprocessFirebaseData };

Service.js  
const express = require("express");

const mssql = require("mssql");

const cors = require("cors");

const S3BucketHelpers = require("./helpers/S3BucketHelper");

const  multer = require("multer");

const  {v4:uuidv4}= require("uuid");

const {config,GetDBconnection} = require("./helpers/DbConfig.js");

const dotenv = require("dotenv");

const {initializeFirebaseApp,uploadprocessFirebaseData} = require("./firebase.js");

const admin = require("firebase-admin");

const serviceAccount = require("./serviceAccountKey.json");

dotenv.config();

const app = express();

const upload = multer();

app.use(cors());

app.use(express.json());

// var config = {

//     user:"Veritragrocerydblogin",

//     password:"otTL5kd0buqb7erEuTqX",

//     server:"veritrarsadevdb.clkbmvhusrdv.us-east-1.rds.amazonaws.com",

//     database:"RSA\_GroceryDEV",

//     options:{

//         encrypt:true,

//         trustServerCertificate: false,

//     },

//     pool:{

//         min:0,

//         max:10,

//         idleTimeoutMillis: 30000,

//     }

// }

// var config = {

//     user: "Veritragrocerydblogin",

//     password: "otTL5kd0buqb7erEuTqX",

//     server: "veritrarsadevdb.clkbmvhusrdv.us-east-1.rds.amazonaws.com",

//     database: "RSA\_GroceryDEV",

//     options: {

//         encrypt: true,

//         trustServerCertificate: true,

//     },

// };

admin.initializeApp({

  credential:admin.credential.cert(serviceAccount)

})

const bucketName = "rsasupportapiproject";

const filePath = "RSASupportConfigurations/AppConfigurations.json";

const testS3Helpers = async () => {

  console.log("helloow")

  try {

    const exists = await S3BucketHelpers.checkFileExists(bucketName, filePath);

    if (!exists) {

      console.log("File does not exist.");

      return;

    }

    const content = await S3BucketHelpers.getFileContents(bucketName, filePath);

    console.log("File Content:", content);

  } catch (error) {

    console.error("Error testing S3 helpers:", error);

  }

};

testS3Helpers();

initializeFirebaseApp();

 uploadprocessFirebaseData().then(() => console.log("upload success")).catch((err) => console.log("err",err));

