**TASK 3: BMI CALCULATOR USING CLOUD RUN, FIREBASE AUTHENTICATION AND FIRESTORE DATABASE PROJECT IN GCP (Google Cloud Platform)**

**Team Members:**

Saadana S (8081980)

Chooriya Prabha(8081955)

Abithavalli(8081953)

Ishwarya Selvam(8079896)

**Task 1: Sign in to the Google Cloud Platform (GCP) Console**

# Task 2: Create a firebase Authentication

Sign into Firebase using your Google account.

If you don't already have a JavaScript project and just want to try out a Firebase product, you can download one of our quickstart samples

## Step 1: Create a Firebase project

Before you can add Firebase to your JavaScript app, you need to create a Firebase project to connect to your app.

**Note:** If you already have a Firebase-registered mobile app and it shares Firebase resources with your JavaScript app, skip this step and configure your JavaScript app to use the same project as your mobile app.

**Create a Firebase project**

## Step 2: Register your app with Firebase

After you have a Firebase project, you can add your web app to it.

1. In the center of the Firebase console's project overview page, click the **Web** icon (plat\_web) to launch the setup workflow.

If you've already added an app to your Firebase project, click **Add app** to display the platform options.

1. Enter your app's nickname.  
   This nickname is an internal, convenience identifier and is only visible to you in the Firebase console.
2. (Optional) Set up Firebase Hosting for your web app.
   * You can set up Firebase Hosting now or later. You can also link your Firebase Web App to a Hosting site at any time in your Project settings.
   * If you choose to set up Hosting up now, select a site from the dropdown list to link to your Firebase Web App.
     + This list displays your project's default Hosting site and any other sites that you've set up in your project.
     + Any site that you've already linked to a Firebase Web App is unavailable for additional linking. Each Hosting site can only be linked to a single Firebase Web App.
3. Click **Register app**.
4. Here's a config object with *example* values:
5. // For Firebase JavaScript SDK v7.20.0 and later, `measurementId` is an optional field  
   *var firebaseConfig = {  
     apiKey: "AIzaSyDOCAbC123dEf456GhI789jKl01-MnO",  
     authDomain: "myapp-project-123.firebaseapp.com",  
     databaseURL: "https://myapp-project-123.firebaseio.com",  
     projectId: "myapp-project-123",  
     storageBucket: "myapp-project-123.appspot.com",  
     messagingSenderId: "65211879809",  
     appId: "1:65211879909:web:3ae38ef1cdcb2e01fe5f0c",  
     measurementId: "G-8GSGZQ44ST"  
   };*

# Task 3 Add Firebase to your login page

# Index.html

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <!-- Compiled and minified CSS -->

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">

  <title>BMI CAL</title>

</head>

<body >

  <!-- NAVBAR -->

  <nav class="z-depth-0 pink lighten-4">

    <div class="nav-wrapper container">

      <a href="#" class="brand-logo">

        <img src="https://storage.cloud.google.com/loginfirestore/logo.svg?authuser=1" style="width: 180px; margin-top: 10px;">

      </a>

      <ul id="nav-mobile" class="right hide-on-med-and-down">

          <li class="logged-in">

            <a href="#" class="grey-text" id="logout">Logout</a>

          </li>

          <li class="logged-out">

            <a href="#" class="grey-text modal-trigger" data-target="modal-login">Login</a>

          </li>

          <li class="logged-out">

            <a href="#" class="grey-text modal-trigger" data-target="modal-signup">Sign up</a>

          </li>

          <li class="logged-out">

            <a href="#" class="grey-text modal-trigger" data-target="modal-signup">BMI calculate</a>

          </li>

        </span>

      </ul>

    </div>

  </nav>

  <!-- SIGN UP MODAL -->

  <div id="modal-signup" class="modal">

    <div class="modal-content">

      <h4>Sign up</h4><br />

      <form id="signup-form">

        <div class="input-field">

          <input type="email" id="signup-email" required />

          <label for="signup-email">Email address</label>

        </div>

        <div class="input-field">

          <input type="password" id="signup-password" required />

          <label for="signup-password">Choose password</label>

        </div>

        <button class="btn yellow darken-2 z-depth-0">Sign up</button>

      </form>

    </div>

  </div>

  <!-- LOGIN MODAL -->

  <div id="modal-login" class="modal">

    <div class="modal-content">

      <h4>Login</h4><br />

      <form id="login-form">

        <div class="input-field">

          <input type="email" id="login-email" required />

          <label for="login-email">Email address</label>

        </div>

        <div class="input-field">

          <input type="password" id="login-password" required />

          <label for="login-password">Your password</label>

        </div>

        <button class="btn yellow darken-2 z-depth-0">Login</button>

      </form>

    </div>

  </div>

  <!-- The core Firebase JS SDK is always required and must be listed first -->

<!-- The core Firebase JS SDK is always required and must be listed first -->

<script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-app.js"></script>

<script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-auth.js"></script>

<script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-firestore.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use

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

<script>

  // Your web app's Firebase configuration

  var firebaseConfig = {

    apiKey: "AIzaSyARdWikuedG7jFfW62Rr-uP1bZDba2mZQw",

    authDomain: "proud-life-280712.firebaseapp.com",

    projectId: "proud-life-280712",

    databaseURL: "https://proud-life-280712.firebaseio.com",

    storageBucket: "proud-life-280712.appspot.com",

    messagingSenderId: "127826886758",

    appId: "1:127826886758:web:4ead825f01801d1b1c6da7"

  };

  // Initialize Firebase

  firebase.initializeApp(firebaseConfig);

// make auth and firestore references

const auth = firebase.auth();

    const db = firebase.firestore();

    // update firestore settings

    db.settings({ timestampsInSnapshots: true });

</script>

  <!-- Compiled and minified JavaScript -->

  <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>

  <script src="auth.js"></script>

  <script src="index.js"></script>

</body>

</html>

# Auth.js

// listen for auth status changes

auth.onAuthStateChanged(user => {

    if (user) {

      console.log('user logged in: ', user);

    } else {

      console.log('user logged out');

    }

  })

  // signup

  const signupForm = document.querySelector('#signup-form');

  signupForm.addEventListener('submit', (e) => {

    e.preventDefault();

    // get user info

    const email = signupForm['signup-email'].value;

    const password = signupForm['signup-password'].value;

    // sign up the user

    auth.createUserWithEmailAndPassword(email, password).then(cred => {

      // close the signup modal & reset form

      const modal = document.querySelector('#modal-signup');

      M.Modal.getInstance(modal).close();

      signupForm.reset();

    });

  });

  // logout

  const logout = document.querySelector('#logout');

  logout.addEventListener('click', (e) => {

    e.preventDefault();

    auth.signOut();

  });

  // login

  const loginForm = document.querySelector('#login-form');

  loginForm.addEventListener('submit', (e) => {

    e.preventDefault();

    // get user info

    const email = loginForm['login-email'].value;

    const password = loginForm['login-password'].value;

    // log the user in

    auth.signInWithEmailAndPassword(email, password).then((cred) => {

      // close the signup modal & reset form

      const modal = document.querySelector('#modal-login');

      M.Modal.getInstance(modal).close();

      loginForm.reset();

    });

  });

# Index.js

// setup materialize components

document.addEventListener('DOMContentLoaded', function() {

    var modals = document.querySelectorAll('.modal');

    M.Modal.init(modals);

    var items = document.querySelectorAll('.collapsible');

    M.Collapsible.init(items);

  });

# Task 4: Create a python-flask BMI application

# Bmi\_cal.html

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <!-- Compiled and minified CSS -->

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">

    <title>BMI CAL</title>

</head>

<body>

    <!-- NAVBAR -->

    <nav class="z-depth-0 pink lighten-4">

        <div class="nav-wrapper container">

            <a href="#" class="brand-logo">

                <img src="https://storage.cloud.google.com/loginfirestore/logo.svg?authuser=1"

                    style="width: 180px; margin-top: 10px;">

            </a>

            <ul id="nav-mobile" class="right hide-on-med-and-down">

                <li class="logged-in">

                    <a href="#" class="grey-text" id="logout">Logout</a>

                </li>

                <li class="logged-out">

                    <a href="#" class="grey-text modal-trigger" data-target="modal-login">Login</a>

                </li>

                <li class="logged-out">

                    <a href="#" class="grey-text modal-trigger" data-target="modal-signup">Sign up</a>

                </li>

                <li class="logged-out">

                    <a href="index.html" class="grey-text modal-trigger" data-target="bmi-data">BMI calculate</a>

                </li>

                </span>

            </ul>

        </div>

    </nav>

    <!-- SIGN UP MODAL -->

    <div id="modal-signup" class="modal">

        <div class="modal-content">

            <h4>Sign up</h4><br />

            <form id="signup-form">

                <div class="input-field">

                    <input type="email" id="signup-email" required />

                    <label for="signup-email">Email address</label>

                </div>

                <div class="input-field">

                    <input type="password" id="signup-password" required />

                    <label for="signup-password">Choose password</label>

                </div>

                <button class="btn yellow darken-2 z-depth-0">Sign up</button>

            </form>

        </div>

    </div>

    <div id="bmi-data"  class="modal">

        <div class="modal-content">

  <div class=" main">

        <form class="pure-form" method="POST" action="/">

            Weight in kgs:<br>

            <input type="text" name="weight"><br>

            Height in cms:<br>

            <input type="text" name="height"><br>

            <button type="submit" class="pure-button pure-button-primary" value="Submit">Submit</button>

        </form>

    </div>

    <br>

    <div class="main">

        {% if bmi %}

        <p>

            {% print("Your BMI is {}.".format(bmi)) %}

            {% print("You are {}".format(output)) %}

        </p>

        {% endif %}

    </div>

    </div>

    </div>

    <!-- LOGIN MODAL -->

    <div id="modal-login" class="modal">

        <div class="modal-content">

            <h4>Login</h4><br />

            <form id="login-form">

                <div class="input-field">

                    <input type="email" id="login-email" required />

                    <label for="login-email">Email address</label>

                </div>

                <div class="input-field">

                    <input type="password" id="login-password" required />

                    <label for="login-password">Your password</label>

                </div>

                <button class="btn yellow darken-2 z-depth-0">Login</button>

            </form>

        </div>

    </div>

    <!-- The core Firebase JS SDK is always required and must be listed first -->

    <!-- The core Firebase JS SDK is always required and must be listed first -->

    <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-app.js"></script>

    <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-auth.js"></script>

    <script src="https://www.gstatic.com/firebasejs/8.2.6/firebase-firestore.js"></script>

    <!-- TODO: Add SDKs for Firebase products that you want to use

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

    <script>

        // Your web app's Firebase configuration

        var firebaseConfig = {

            apiKey: "AIzaSyARdWikuedG7jFfW62Rr-uP1bZDba2mZQw",

            authDomain: "proud-life-280712.firebaseapp.com",

            projectId: "proud-life-280712",

            databaseURL: "https://proud-life-280712.firebaseio.com",

            storageBucket: "proud-life-280712.appspot.com",

            messagingSenderId: "127826886758",

            appId: "1:127826886758:web:4ead825f01801d1b1c6da7"

        };

        // Initialize Firebase

        firebase.initializeApp(firebaseConfig);

        // make auth and firestore references

        const auth = firebase.auth();

        const db = firebase.firestore();

        // update firestore settings

        db.settings({ timestampsInSnapshots: true });

    </script>

    <!-- Compiled and minified JavaScript -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>

    <script>// listen for auth status changes

        auth.onAuthStateChanged(user => {

            if (user) {

                console.log('user logged in: ', user);

            } else {

                console.log('user logged out');

            }

        })

        // signup

        const signupForm = document.querySelector('#signup-form');

        signupForm.addEventListener('submit', (e) => {

            e.preventDefault();

            // get user info

            const email = signupForm['signup-email'].value;

            const password = signupForm['signup-password'].value;

            // sign up the user

            auth.createUserWithEmailAndPassword(email, password).then(cred => {

                // close the signup modal & reset form

                const modal = document.querySelector('#modal-signup');

                M.Modal.getInstance(modal).close();

                signupForm.reset();

            });

        });

        // logout

        const logout = document.querySelector('#logout');

        logout.addEventListener('click', (e) => {

            e.preventDefault();

            auth.signOut();

        });

        // login

        const loginForm = document.querySelector('#login-form');

        loginForm.addEventListener('submit', (e) => {

            e.preventDefault();

            // get user info

            const email = loginForm['login-email'].value;

            const password = loginForm['login-password'].value;

            // log the user in

            auth.signInWithEmailAndPassword(email, password).then((cred) => {

                // close the signup modal & reset form

                const modal = document.querySelector('#modal-login');

                M.Modal.getInstance(modal).close();

                loginForm.reset();

            });

        });</script>

    <script>// setup materialize components

        document.addEventListener('DOMContentLoaded', function () {

            var modals = document.querySelectorAll('.modal');

            M.Modal.init(modals);

            var items = document.querySelectorAll('.collapsible');

            M.Collapsible.init(items);

        });</script>

</body>

</html>

# App.py

#!python3

import os

import uuid

from flask import Flask, request, jsonify

from firebase\_admin import credentials, firestore, initialize\_app

from flask import Flask, render\_template, request

app = Flask(\_\_name\_\_)

cred = credentials.Certificate('serviceAccountKey.json')

default\_app = initialize\_app(cred)

db = firestore.client()

users\_bmi = db.collection('usersbmi')

@app.route('/', methods=['GET', 'POST'])

def index():

    bmi = ''

    output=''

    if request.method == 'POST' and 'weight' in request.form:

        weight = float(request.form.get('weight'))

        height = float(request.form.get('height'))

        bmi = calc\_bmi(weight, height)

        json = {'bmi': bmi, 'height': height, 'weight': weight}

        id = uuid.uuid1()

        users\_bmi.document(str(id)).set(json)

        if (bmi <= 18.5):

            output = "Under Weight"

        elif (bmi > 18.5 and bmi <= 24.9):

            output = "Normal Weight"

        elif (bmi > 24.9 and bmi <= 29.9):

            output = "Over Weight"

        elif (bmi > 30.0):

            output = "OBESE"

    return render\_template("index.html",

                           bmi=bmi,output=output)

def calc\_bmi(weight, height):

    return round((weight / ((height / 100) \*\* 2)), 2)

port = int(os.environ.get('PORT', 8080))

if \_\_name\_\_ == '\_\_main\_\_':

    app.run()

# 

# Task 5: Creation of Firestore db in the firebase console

# 

# 

# 

# 

# Task 6: Create Docker File,Requirements.Txt File For Building Docker Image Using Cloud Build And Deploy Using Cloud Run

Create a new directory named helloworld-php and change directory into it:

*mkdir helloworld-php  
cd helloworld-php*

# Add the following files in the helloworld-php

# dockerfile:

# *# Use Python37*

# *FROM python:3.7*

# *# Copy requirements.txt to the docker image and install packages*

# *COPY requirements.txt /*

# *RUN pip install -r requirements.txt*

# *# Set the WORKDIR to be the folder*

# *COPY . /app*

# *# Expose port 5000*

# *EXPOSE 5000*

# *ENV PORT 5000*

# *WORKDIR /app*

# *# Use gunicorn as the entrypoint*

# *CMD exec gunicorn --bind :$PORT app:app --workers 1 --threads 1 --timeout 60*

# .dockerignore:

# *Dockerfile*

# *README.md*

# *\*.pyc*

# *\*.pyo*

# *\*.pyd*

# *\_\_pycache\_\_*

# *.pytest\_cache*

# Requirements.txt:

# *click==6.7*

# *flask*

# *gunicorn==19.7.1*

# *itsdangerous==0.24*

# *Jinja2==2.9.6*

# *Werkzeug==0.12.2*

# *firebase\_admin*

Build your container image using Cloud Build, by running the following command from the directory containing the Dockerfile:

*gcloud builds submit --tag gcr.io/*PROJECT-ID*/helloworld*

where PROJECT-ID is your GCP project ID. You can get it by running gcloud config get-value project.

Upon success, you will see a SUCCESS message containing the image name (gcr.io/PROJECT-ID/helloworld). The image is stored in Container Registry and can be re-used if desired.

## Deploying to Cloud Run

To deploy the container image:

1. Deploy using the following command:

*gcloud run deploy --image gcr.io/*PROJECT-ID*/helloworld --platform managed*

Replace PROJECT-ID with your GCP project ID. You can view your project ID by running the command gcloud config get-value project.

1. You will be prompted for the service name: press Enter to accept the default name, helloworld.
2. You will be prompted for region: select the region of your choice, for example us-central1.
3. You will be prompted to **allow unauthenticated invocations**: respond y .

Then wait a few moments until the deployment is complete. On success, the command line displays the service URL.

Visit your deployed container by opening the service URL in a web browser.

# Task 7: View Results

# 

# 

# 

# 

# 

# 

# 