Signup   
  
import {

  DynamoDBClient,

  PutItemCommand,

} from "@aws-sdk/client-dynamodb";

const db = new DynamoDBClient();

export async function handler(event) {

  const headers = {

    'Access-Control-Allow-Origin': '\*',

    'Access-Control-Allow-Headers': 'Content-Type',

    'Access-Control-Allow-Methods': 'POST, OPTIONS'

  };

  if (event.httpMethod === "OPTIONS") {

    return {

      statusCode: 200,

      headers,

      body: JSON.stringify({ message: "CORS preflight success" }),

    };

  }

  const body = JSON.parse(event.body || '{}');

  const { email, password, city, frequency, time } = body;

  if (!email || !password || !city || !frequency || !time) {

    return {

      statusCode: 400,

      headers,

      body: JSON.stringify({ message: "All fields are required" }),

    };

  }

  try {

    const command = new PutItemCommand({

      TableName: "Users",

      Item: {

        email: { S: email },

        password: { S: password },

        city: { S: city },

        frequency: { S: frequency },

        time: { S: time },

        subscribed: { BOOL: true }

      },

      ConditionExpression: 'attribute\_not\_exists(email)',

    });

    await db.send(command);

    return {

      statusCode: 200,

      headers,

      body: JSON.stringify({ message: "User registered successfully" }),

    };

  } catch (err) {

    if (err.name === 'ConditionalCheckFailedException') {

      return {

        statusCode: 400,

        headers,

        body: JSON.stringify({ message: "User already exists" }),

      };

    }

    console.error("❌ DynamoDB Error:", err);

    return {

      statusCode: 500,

      headers,

      body: JSON.stringify({ message: "Signup failed", error: err.message }),

    };

  }

}

Weather

import { SNSClient, PublishCommand } from "@aws-sdk/client-sns";

import { DynamoDBClient, ScanCommand } from "@aws-sdk/client-dynamodb";

const sns = new SNSClient({ region: "eu-north-1" });

const db = new DynamoDBClient({ region: "eu-north-1" });

const WEATHER\_API = "https://api.weatherapi.com/v1/current.json";

const WEATHER\_API\_KEY = "7aacc04ea876416f8d7170130250108";

const TOPIC\_ARN = "arn:aws:sns:eu-north-1:388585302770:email";

export const handler = async () => {

  try {

    const scanResult = await db.send(

      new ScanCommand({

        TableName: "Users",

        FilterExpression: "subscribed = :val",

        ExpressionAttributeValues: {

          ":val": { BOOL: true },

        },

      })

    );

    const users = scanResult.Items || [];

    if (users.length === 0) {

      return {

        statusCode: 200,

        body: "No subscribed users found.",

      };

    }

    // Just use the first user’s city as a demo (since SNS can't personalize)

    const city = users[0].city?.S || "Kolkata";

    const weatherRes = await fetch(`${WEATHER\_API}?key=${WEATHER\_API\_KEY}&q=${city}`);

    const weather = await weatherRes.json();

    const message = `

🌤️ Weather Update for ${city}

📍 ${weather.location.name}, ${weather.location.country}

🌡️ Temp: ${weather.current.temp\_c}°C

💧 Humidity: ${weather.current.humidity}%

🌬️ Wind: ${weather.current.wind\_kph} kph

📖 Condition: ${weather.current.condition.text}

~ Weatherly 🌈

`;

    await sns.send(

      new PublishCommand({

        TopicArn: TOPIC\_ARN,

        Message: message,

        Subject: `🌦️ Daily Weather Update for ${city}`,

      })

    );

    console.log(`✅ Published weather update for ${city}`);

    return {

      statusCode: 200,

      body: `Published weather update for ${users.length} users.`,

    };

  } catch (err) {

    console.error("❌ Error:", err);

    return {

      statusCode: 500,

      body: JSON.stringify({ error: err.message }),

    };

  }

};

SES code   
  
import { SESClient, SendEmailCommand } from "@aws-sdk/client-ses";

import { DynamoDBClient, ScanCommand } from "@aws-sdk/client-dynamodb";

const ses = new SESClient({ region: "eu-north-1" });

const db = new DynamoDBClient({ region: "eu-north-1" });

const WEATHER\_API\_KEY = "7aacc04ea876416f8d7170130250108";

const WEATHER\_API = "https://api.weatherapi.com/v1/current.json";

const SENDER\_EMAIL = "sohammondal12345@gmail.com"; // Must be SES verified

export const handler = async () => {

  try {

    const result = await db.send(

      new ScanCommand({

        TableName: "Users",

        FilterExpression: "subscribed = :val",

        ExpressionAttributeValues: {

          ":val": { BOOL: true },

        },

      })

    );

    const users = result.Items || [];

    for (const user of users) {

      const email = user.email?.S;

      const city = user.city?.S || "Kolkata";

      // ✅ Fetch weather

      const weatherRes = await fetch(`${WEATHER\_API}?key=${WEATHER\_API\_KEY}&q=${city}`);

      const weather = await weatherRes.json();

      const message = `

🌤️ Weather Update for ${city}

📍 ${weather.location.name}, ${weather.location.country}

🌡️ Temp: ${weather.current.temp\_c}°C

💧 Humidity: ${weather.current.humidity}%

🌬️ Wind: ${weather.current.wind\_kph} kph

📖 Condition: ${weather.current.condition.text}

~ Weatherly 🌈

      `;

      const emailParams = new SendEmailCommand({

        Destination: {

          ToAddresses: [email],

        },

        Message: {

          Subject: { Data: `🌦️ Daily Weather for ${city}` },

          Body: { Text: { Data: message } },

        },

        Source: SENDER\_EMAIL,

      });

      await ses.send(emailParams);

      console.log(`✅ Email sent to ${email}`);

    }

    return {

      statusCode: 200,

      body: JSON.stringify({ message: `Emails sent to ${users.length} users.` }),

    };

  } catch (err) {

    console.error("❌ Error sending emails:", err);

    return {

      statusCode: 500,

      body: JSON.stringify({ error: err.message }),

    };

  }

};