```php
<?php

namespace App\Jobs;

use App\Model\IterableApi;
use App\Model\IterableRecommendation;
use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Bus\Dispatchable;
use Illuminate\Queue\InteractsWithQueue;
use Illuminate\Queue\SerializesModels;
use Log;

class recommendationToIterableJob implements ShouldQueue
{

    use Dispatchable, InteractsWithQueue, Queueable, SerializesModels;

    public $sessionID;
    public $shopify_api_version;
    public $iterableRecObj;
    public $iterableObj;
    private $customPropertyDbQuestionNum = [2, 4, 5, 7, 11, 17, 22, 31, 32, 33, 46];
    private $responseIdEmptyQuestionNum = [3];
    private $customQuestionArray = ["gender" => 2, "age" => 3, "goal" => 4, "support" => 5,
"seeking" => 7, "diets" => 11,
        "planningPregnancy" => 17, "drinks" => 22, "interests" => 31, "stress" => 32, "sleep" => 33,
"smoke" => 46];

    /**
     * Create a new job instance.
     *
     * @return void
     */
    public function __construct($sessionID)
    {

        $this->sessionID = $sessionID;
        $this->shopify_api_version = env('SHOPIFY_API_VERSION', true);
    }

    /**
     * Execute the job.
     *
     * @return void
     */
    public function handle()
    {

        $this->iterableRecObj = new IterableRecommendation();
        $this->iterableObj = new IterableApi();
        $dataFromRecommendation = $this->iterableRecObj->getRecommendationDetails($this-
>sessionID);
        if (isset($dataFromRecommendation) && !empty($dataFromRecommendation)) {

            foreach ($dataFromRecommendation as $recValue) {

                $recommendationData = '';
                $quizData = '';
                $customRecProperties = $this->iterableRecObj->getDbUserResponseDetails(
                    $recValue->session_id, $this->customPropertyDbQuestionNum,
                    $this->responseIdEmptyQuestionNum, $this->customQuestionArray);

                if (!isset($customRecProperties) && empty($customRecProperties)) {

                    $message = 'The recommendations(session_id: ' . $recValue->session_id . ',id: ' .
$recValue->id . ')
                    user response data is empty';
                    Log::info($message);
                    continue;
                }
                $recommendedDbProducts = json_decode($recValue->recommendation);
                if (isset($recommendedDbProducts) && !empty($recommendedDbProducts)) {

                    foreach ($recommendedDbProducts as $pdtValue) {

                        $recommendationData .= $pdtValue->Name . " | ";
                    }
                }

                $customDbPropertyData = [];
                if (isset($customRecProperties) && !empty($customRecProperties)) {

                    foreach ($customRecProperties as $propertyKey => $propertyValue) {

                        if ($propertyKey == 'gender' || $propertyKey == 'age') {

                            continue;
                        } else {

                            $customDbPropertyData[$propertyKey] = $propertyValue;
                        }
                    }
                }
                $recommendationData = rtrim($recommendationData, " | ");

                $userIterableUpdateRequest = array(
                    "email" => $recValue->email,
                    "dataFields" => array(
                        "recommendation" => $recommendationData,
                        "gender" => isset($customRecProperties['gender']) ? $customRecProperties['gender']
: '',
                        "age" => isset($customRecProperties['age']) ? $customRecProperties['age'] : '',
                        "quizAnswer" => $customDbPropertyData,
                        "quizId" => $this->sessionID,
                    ),
                );

                if ($userIterableUpdateRequest['email']) {

                    $userIterableUpdateResponse = $this->iterableObj-
>updateUserDataFields(json_encode($userIterableUpdateRequest));

                    if (strtolower(trim($userIterableUpdateResponse->code)) == 'success') {

                        //$iterableUpdatedDataInsertStatus = $this->iterableRecObj-
>insertUpdatedIterableDetails($recValue);
                        $insertUpdatedIterableDetails = true;
                        if ($insertUpdatedIterableDetails) {

                            Log::info("Successfully inserted the iterable updated recommendation(
                            session_id:" . $recValue->session_id . ")details to DB");
                        } else {

                            Log::info("Failed to insert the iterable updated recommendation(
                            session_id:" . $recValue->session_id . ") details to DB");
                        }
                        Log::info("User quiz recommendation details - Successfully
                    updated the user quiz recommendation in Iterable for Email (" .
                            strtolower($recValue->email) . ")");
                    } else {
                        Log::info("User quiz recommendation details - Failed to update
                    the user quiz recommendation in Iterable for Email (" .
                            strtolower($recValue->email) . ")");
                        Log::info(print_r($userIterableUpdateResponse, 1));
                    }
                }
            }
        } else {

            $message = 'No pending recommendations in DB(recommendation table) to update in
iterable';
            Log::info($message);
        }
        }
}
```