

```

<?php
namespace App\Model;

use DB;
use Illuminate\Database\Eloquent\Model;

class IterableRecommendation extends Model
{
    public function __construct()
    {

    }

    public function getRecommendationDetails($sessionID)
    {
        $dataFromRecommendation = DB::select('SELECT * FROM recommendations WHERE
session_id = ' . $sessionID . ');

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

            return $dataFromRecommendation;
        }

        return null;
    }

    public function insertUpdatedIterableDetails($recValue)
    {

        if (isset($recValue) && !empty($recValue)) {
            $current_time = date('Y-m-d h:m:s');
            $recommendationId = $recValue->id;
            $sessionId = $recValue->session_id;
            $status = "updated";

            $insertedResponse = DB::insert('insert into iterable_recommendation_cron
(recommendation_id,session_id,iterable_updated_at,status) values
(' . $recommendationId . ',' . $sessionId . ',' . $current_time . ',' . $status . ')');

            return $insertedResponse;
        }
        return null;
    }

    public function getDbUserResponseDetails($sessionID, $dbQuestionNumbers,
        $responseIdEmptyQuestionNum, $customQuestionArray) {

        if (!empty($sessionID)) {

            $customPropertyArr = [];

```

```

    $userDbResponse = DB::select('SELECT * FROM user_responses INNER JOIN
question_responses ON
    user_responses.response_id = question_responses.id WHERE user_responses.session_id = '
. $sessionID . ''');
    $userDbEmptyResponseIdResponse = DB::select('SELECT * FROM user_responses
WHERE session_id = ' . $sessionID . '' AND response_id IS NULL');
    if (isset($userDbResponse) && !empty($userDbResponse)) {

        $processedQuestionDetails = [];
        foreach ($userDbResponse as $resValue) {

            $questionValue = "";
            if (in_array($resValue->qno, $dbQuestionNumbers)) {

                $questionValue = $resValue->response;
                if (array_key_exists($resValue->qno, $processedQuestionDetails)) {

                    $processedQuestionDetails[$resValue->qno] .= ' . trim($questionValue);
                } else {

                    $processedQuestionDetails[$resValue->qno] = trim($questionValue);
                }
            }
        }
    }
    if (isset($processedQuestionDetails) && !empty($processedQuestionDetails)) {

        foreach ($dbQuestionNumbers as $questValue) {

            if (array_key_exists($questValue, $processedQuestionDetails) == false) {

                $processedQuestionDetails[$questValue] = "";
            }
        }
        if (isset($userDbEmptyResponseIdResponse) && !
empty($userDbEmptyResponseIdResponse)) {

            foreach ($userDbEmptyResponseIdResponse as $emptyResValue) {

                if (in_array($emptyResValue->qno, $responseIdEmptyQuestionNum)) {

                    $processedQuestionDetails[$emptyResValue->qno] = $emptyResValue-
>response_text;
                }
            }
        }
        ksort($processedQuestionDetails);
        $customPropertyDetails = [];
        foreach ($customQuestionArray as $customKey => $customValue) {

            foreach ($processedQuestionDetails as $processKey => $processValue) {

```

```
        if ($customValue == $processKey) {  
            $customPropertyDetails[$customKey] = $processValue;  
        }  
    }  
} else {  
    return null;  
}  
return $customPropertyDetails;  
}  
return null;  
}  
}
```