```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb+srv://
                                                                                                                                                                                   .mongodb.net/?retryWrites=true&w=majority";
         function processTfs_accepted(taskownersId, tasksInProject, accepted, acceptedCount, taskNumber, taskhelpersId, phase, s4AprvTfs, tfsStatusID){
                     var MongoClient = require('mongodb').MongoClient;
Var MongoClient = require('mongodb').MongoClient = require('mongodb').Mon
                      if(phase == 1){
                      if(s4AprvTfs != 1){
                         if(tfsStatusID == 5){
                              if(accepted == 1){
               acceptedCount += 1;
        if(acceptedCount - 24){
        Mongo(lient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db, db("usersdatabase");
    var query = { userNumber; taskhelpersId };
    var query = { userNumber; taskhelpersId };
    dbo_collection("usersdatab").find(query).toArray(function(err, result) {
                       if (err) throw err;
var mobileNumber = result[0] mobileNumber:
         var acceptTask = "https://task.help/accept_task/?tnum=" + taskNumber;
         var sms_msqLink19 = "TASK HELP. Last chance to accept task. This is final reminder to accept task. Please click link to accept task." + acceptTask;
         var accountSid = 'AC220046E4-27L1622E06-EE64EL21-2cb'; // Your Account SID from www.twilio.com/console var authToken = '244004044(canasaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccountsaccount
         var twilio = require('twilio');
var client = new twilio(accountSid, authToken);
        client.messages.create({
body: sms.msglinkl9,
to: mobileNumber, // Text this number
from: +14155826487 // From a valid Twilio number
db.close();
                      if(acceptedCount == 48){
              MongoClient.connect(unl, function(err, db) {
    vor dbo - db.dbC'resssignque');
    vor dbo - db.dbC'resssignque');
    vor myob) - { "ressign; 1. tessMumber: taskNumber: };
    dif (err) throw err;
    db.close();
    db.close();
                                                                                                                                                                                                                                                                                                                            Return the fields "name" and "address" of all documents in the customers collection:
                                                                                                                                                                                                                                                                                                                            MongoClient.connect(url, function(err, db) {
                                                                                                                                                                                                                                                                                                                            if (err) throw err;
                                                                                                                                                                                                                                                                                                                            var dbo = db.db("mvdb")
                                                                                                                                                                                                                                                                                                                             if (err) throw err; console.log(result); db.close();
```

```
process Tfs accepted.js has just one function process Tfs accepted() This function will check that:
phase is 1 which means that the task in the taskflow-suggest phase where the task-helper is just coming up with a list of tasks and 2 means that the task-helper has actually started working on the task
s4ApryTfs is not 1 means that because s4ApryTfs used to represent "submit taskflow-suggest for an approval" and when a taskflow-suggest is submitted for an approval s4ApryTfs gets set to a 1 so we are
interested only in tasks where taskflow-suggest have not yet been submitted for an approval.
tfsStatusID is 5 means that because taskStatusID is set to a value of 5 when the task-helper is initially assigned to a task and they have not yet accepted the task. IfsStatusID can have a value of 5, 1, or 2.
tfsStatusID is 5 when task-helper is just assigned task and has not yet accepted task
tfsStatusID is 1 when task-helper has accepted the task
tfsStatusID is 2 when task-helper has accepted the task and the 72 hour time limit to complete and submit their taskflow-suggest and ran out.
```

Once the condition are met, we then check to see if accepted is 1 and if it is we will shift the progress forward and update tfsStatusID with a value of 1 in the database Fise If the task has not yet been accepted or accepted isn't 1, we will increment the accepted Count by 1 and also undate it to database Look to the left On line 50 if acceptedCount reaches 24, we will send notification to taskhelper that time is running out Look to the left On line 91 if accepted Count reaches 48, it means that time has ran out, so we queue the task for get reassigned to another task-helper by using a standard mongodb insertOne

Here is a standard Mongodb update that we are using. You can find this information on the following link https://www.w3schools.com/nodeis/nodeis_mongodb_find.asp

```
Update the document with the address "Valley 345" to name="Mickey" and address="Canyon 123":
var MongoClient = require('mongodb').MongoClient; var url = "mongodb://127.0.0.1:27017/";
MongoClient,connect(url, function(err, db) {
if (err) throw err:
var dbo = db.db("mydb");
var myguery = { address: "Valley 345" }:
var newvalues = { $set; {name; "Mickey", address; "Canyon 123" } }; dbo.collection("customers").updateOne(myguery, newvalues, function(err, res) {
if (err) throw err
console.log("1 document updated"); db.close();
Look to the left on and find that this is a very similar method we are using on line 19 to 28, the same thing on line 38 to 47
```

Here is a standard Mongodb query that we are using. You can find this information on the following link https://www.w3schools.com/nodeis/nodeis mongodb find asp

```
Find documents with the address "Park Lane 38":
var MongoClient = require('mongodb').MongoClient; var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db) {
if (err) throw err:
var dbo = db.db("mvdb");
var query = { address: "Park Lane 38" }; dbo.collection("customers").find(query).toArray(function(err, result) {
if (err) throw err; console.log(result); db.close();
Look to the left on line 53 to 82 we are using a very similar find. Then the resulting array since there should only be one will be in the first array of the result set which is result[0] and we want the
mobileNumber field. Because we have inserted the Twillio api here, the mobile number is then readily accessible to be used to send the notification.
```

Here is a standard Twilio api call. You can find this information on the following link, https://www.twilio.com/docs/sms/quickstart/node#install-nodejs-and-the-twilio-module

```
const accountSid = process.env.TWILIO ACCOUNT SID:
const authToken = process.env.TWILIO_AUTH_TOKEN: const client = require('twilio')(accountSid, authToken):
 client.messages .create({
  body: 'This is the ship that made the Kessel Run in fourteen parsecs?',
  from: '+15017122661'.
  to: '+15558675310'
.then(message => console.log(message.sid));
Look to the left from on line 67 to liine 77, we are inserting the Twilio api here to use the mobileNumber returned from the find query directly above it and send out a notification
 The find query actually starts from line 53 and ends on line 82. So the Twilio api is just inserted in between
```

Here is a standard Mongodb guery that we are using. You can find this information on the following link https://www.w3schools.com/nodeis/nodeis mongodb find.asp

```
var MongoClient = require('mongodb').MongoClient; var url = "mongodb://localhost:27017/";
dbo.collection("customers").find({}, { projection: { _id: 0, name: 1, address: 1 } }).toArray(function(err, result) {
```

Look to the left on line 127 to 139, we using a very similar technique then we use a forEach loop to pass the fields to process_Tfs_accepted() function. The first parameter of the find() method is a query object. In this example, it is an empty query object, which selects all documents in the collection. However on line 130, to the left, we use a find query (phase: 1, s4pryTfs: 0, tfsStatusID; 5). Which enables us to get only documents that are in phase 1, where taskflow-suggest have not yet been submitted for an approval, and tfsStatusID is 5 meaning it has not been accepted.

result.forEach(function (result) {
processIfs_accepted(result.tasksomersid, result.tasksInProject, result.accepted, result.accepted(ount, result.taskID, result.taskhelpersId, result.phase, result.s4AprvTfs, result.tfsStatusID);

You are not allowed to specify both 0 and 1 values in the same object (except if one of the fields is the id field). If you specify a field with the value 0, all other fields get the value 1, and vice versa: We are using this technique to specify only the fields that we are going to be needing in this case: taskownerId: 1, accepted: 1, acceptedCount: 1, taskID: 1, taskHelpersId: 1, phase: 1, s4prvTfs: 1, tfsStatusID: 1