# Technical Test
=============

## Task 1
=====

Add a custom field Total Contacts of type Number on Account object with default value 0.

Add a custom field Active of type Boolean on Contact object with default value false. On creation of a contact for an account, automatically trigger an approval process to approve the contact. If the contact is approved, set the active flag to true and increment the value of field Total Contacts on Account by 1.

On deleting the active contact, decrement the value of field Total Contacts on the Account by 1.

Write the logic to increment and decrement the total contacts count using bulk pattern.

**References**:
Trigger: https://trailhead.salesforce.com/en/content/learn/modules/apex_triggers/apex_triggers_intro
Approval Process: https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/create-an-approval-process
Bulk: https://trailhead.salesforce.com/content/learn/modules/apex_triggers/apex_triggers_bulk

## Task 2
=====

Create a Visualforce page to accept a String as an input and search Accounts and Contacts with the name starting with the entered string. The search should be case-insensitive.

As an output show a table of the records found with the type of record.

For example: if input string is 'ABC' and there is an Account with name ABC Corp and Contact with name ABC Test Person, show the output in the following format:

| Record Name | Record Type | Click to open |
|---|---|---|
| ABC Corp | Account | View |
| ABC Test Person | Contact | View |

On clicking the View link, open the record in another screen/tab.

**References**:
Visualforce: https://trailhead.salesforce.com/content/learn/modules/visualforce_fundamentals

## Task 3
=====

Extend the functionality of above created page. Add a custom field Counter of type Number on both Account and Contact object with default value 0. Show the value of the counter as shown below. Add a column '*Click to update counter*' that should display a command link to update the counter of the searched record on click.

The call to update the counter should be done using Visualforce remoting and as a callback refresh the value of the counter on the VF page.

| Record Name | Record Type | Counter | Click to update counter |
|---|---|---|---|

| ABC Corp | Account | 0 | Increment Counter |
| ABC Test Person | Contact | 0 | Increment Counter |

**References**:
Remoting: https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_js_remoting_example.htm

**Task 4**
=====
Write a RESTful web service to accept a payload of multiple contacts and update those contacts. Add basic validations like the input contacts should exist in the system. API should return proper success and error responses.

References:
Apex REST: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_rest_intro.htm