A Test Case Report on

Person Finder | Google

Under the guidance of

P. Santhi Thilagam (Professor)

Submitted by

Hardik Rana - 16CO138
Harshal Shinde - 16CO223
VII Sem B.Tech (CSE)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

Department of Computer Science & Engineering Technology

National Institute of Technology Karnataka, Surathkal.

17 October 2019

INDEX

1. Unit Tests [Functional Testing]

1.

Test scenario ID	Unit Test	**Test case ID**	TU01
Test case description	To verify if new person's entry is created in the proper format	**Test priority**	High
Prerequisite	The entry should be already existing	**Post-requirement**	NA
Test Procedure		1. Create a new entry using model class 2. Using the Assert function to verify if the entry created is stored in the right format	
Input		Data should be according to the data type of the column	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

2.

Test scenario ID	Unit Test	**Test case ID**	TU02
Test case description	To verify if entry for new photo is created	**Test priority**	High
Prerequisite	For test case to pass the photo entry must already exist	**Post-requirement**	NA
Test Procedure		1. Create a new entry by providing image data 2.Using the url builder creating the url for the new image entry 3. Compare the expected url with the generated url	
Input		Image name	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

3.

Test scenario ID	Unit Test	**Test case ID**	TU03
Test case description	To verify the entry for person search (full text search) is created or not	**Test priority**	High
Prerequisite	For test case to pass the person search entry must already exist [or we can create one]	**Post-requirement**	NA
Test Procedure		1.Create a new search person entry using model person class 2.Using the Assert function to verify if we can search for the entry created based on different functions in the full_text_search file	
Input		Person's given name,family_name,entry_date etc..	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

4.

Test scenario ID	Unit Test	**Test case ID**	TU04
Test case description	To verify the entry for person search which is created in test case 3 can be deleted or not	**Test priority**	High
Prerequisite	For test case to pass the person search entry must already exist [or we can create one]	**Post-requirement**	NA
Test Procedure		1.Create a new search person entry using model person class [if not already exists] 2. Delete the created entry using functions of full_text_search file. 3.Using the Assert function to verify if entry is deleted or not [by searching for that entry in database]	
Input		Person's given name, family_name,entry_date etc..	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

5.

Test scenario ID	Unit Test	**Test case ID**	TU05
Test case description	To verify that various constants which are set in const.py file are matching with the pfif (person finder interchange format) or not	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1. Get various constants which are set in const.py file. 2.Using the Assert function to verify if this constants are matching with the pfif constants or not.	
Input		Constants which are set in const.py file	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

6.

Test scenario ID	Unit Test	**Test case ID**	TU06
Test case description	To verify the format in which date is stored	**Test priority**	High
Prerequisite	The entry should be already existing	**Post-requirement**	NA
Test Procedure		1. Create a new date entry using datetime library of python 2. Using Assert function verify if the format of date stored is correct 3. We can also check if it raises different value error or exception when the date is not in given format.	
Input		A valid date	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

7.

Test scenario ID	Unit Test	**Test case ID**	TU07
Test case description	To verify if normalise function(it converts all string in set to lowercase) is working	**Test priority**	High
Prerequisite	Set of predefined bad words should be created [With the SpamDetector class]	**Post-requirement**	NA
Test Procedure		1. Make an entry to the set using uppercase strings. 2. Use Assert function to check if the uppercase of the input string has been converted to lowercase	
Input		Set of words	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

8.

Test scenario ID	Unit Test	**Test case ID**	TU08
Test case description	To verify if bad words and spam is detected [Input text is None or it contains comma in-between which is not needed]	**Test priority**	High
Prerequisite	Set of predefined bad words should be created [With the Spam-Detector class]	**Post-requirement**	NA
Test Procedure		1. Use Assert function with the predefined bad words and check if their spam probability calculated by the function matches with that of the expected probability	
Input		Valid string	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

9.

Test scenario ID	Unit Test	**Test case ID**	TU09
Test case description	To verify whether the strip function inside importer.py file is working correctly or not	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Have some string to which will be passed to strip function of importer file. 2.Use Assert function for checking whether the string we are getting by applying the strip function of importer file on the string will remove leading and trailing spaces or not.	
Input		Set of Strings	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

10.

Test scenario ID	Unit Test	**Test case ID**	TU10
Test case description	To verify whether the validate_datetime function inside importer.py file is working correctly or not	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1. Using Assert function verify if the format of datetime stored is matching with the datetime we are getting by using datetime library of python 2. We can also check if it raises different value error when the datetime is not matching with the given format.	
Input		Set of Strings consisting of date-time	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

11.

Test scenario ID	Unit Test	**Test case ID**	TU11
Test case description	To verify whether the validate_boolean function inside importer.py file is working correctly or not	**Test priority**	High
Prerequisite	Set of boolean flags (True,False)	**Post-requirement**	NA
Test Procedure		1. Using Assert function pass the boolean flag inside validate function of importer file and check it	
Input		Set of strings	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

12.

Test scenario ID	Unit Test	**Test case ID**	TU12
Test case description	To verify the format in which the note is stored	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1. Create a new note entry 2. Use Assert function to check that the various fields of the note are in the correct format	
Input		Name of the repo,Strings in the note	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

13.

Test scenario ID	Unit Test	**Test case ID**	TU13
Test case description	To verify if the create_person function inside importer.py file is working properly or not	**Test priority**	High
Prerequisite	All fields[attributes of person class] which are there in the website should be known	**Post-requirement**	NA
Test Procedure		1.Take some dummy user fields data and create a model by using on create_person function in importer.py file 2.Use hasattr function of python with Assert to check whether particular attribute belongs to person class or not,and for other attributes check whether that particular attribute is set to what we gave in step1 or not	
Input		Attributes related to person	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

14.

Test scenario ID	Unit Test	**Test case ID**	TU14
Test case description	To verify if the import_records function inside importer.py file is working properly or not	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return type of function.	**Post-requirement**	NA
Test Procedure		1.Create dummy record list of different-different persons,some of which contains bad domain,some contains invalid date etc. 2.Call the import_records function by passing this created record list and other necessary parameters and store the results returned by this function in some other list/variables. 3.Use Assert function to check if total written count is matching with what we are expecting and check whether the entries which are expected to be skipped are actually skipped or not and check total count of entries.	
Input		List of some data related to different-different persons	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			

Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

15.

Test scenario ID	Unit Test	**Test case ID**	TU15
Test case description	To verify if the import_records function inside importer.py file is while importing person records with notes also.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function.	**Post-requirement**	NA
Test Procedure		1.Steps will be the same as what we will do in test-case 14, just one more field note_id will be added. 2.Other steps like calling a function of importer.py file and storing results and comparison will be same as TU14.	
Input		List of some data related to different-different persons	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			

Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

16.

Test scenario ID	Unit Test	**Test case ID**	TU16
Test case description	To verify whether the function rank_and_order inside indexing.py file is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function.	**Post-requirement**	NA
Test Procedure		1.Create some dummy person records by passing given_name,family_name fields. 2.Call the rank_and_order function by passing the above created list of dummy person lists, Textquery (prefix) you want to check and maximum entries you want in your results and store it in some list. 3.Using Assert check whether the result list you got in step2 is matching with what you are expecting or not.	
Input		List of some data related to different-different persons	
Expected Result		Pass/Fail based on the conditions	
Actual Result			

Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

17.

Test scenario ID	Unit Test	**Test case ID**	TU17
Test case description	To verify whether the function rank_and_order inside indexing.py file is working properly or not if given_name or family_name is given as a concatenation of more than one words.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function.	**Post-requirement**	NA
Test Procedure		1. Same as step1 of test-case 20,but this time add entries which contains family_name or given_name as a concatenation of more than one words. 2.Calling the function and comparing results steps will be the same as what we will do in test-case 20.	
Input		List of some data related to	

	different-different persons [In which for entries given_name or family_name should contain more than one words]
Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

18.

Test scenario ID	Unit Test	**Test case ID**	TU18
Test case description	To verify whether the function sort_query_words inside indexing.py file is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function.	**Post-requirement**	NA
Test Procedure		1.Take list of different-different words. 2.Use Assert function and check whether the results we are getting by passing this lists to sort_query_words function is matching with our expectation results or not	

Input	List of different-different words. [Which can be used to check sort by length,sort by popularity functionalities]
Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

19.

Test scenario ID	Unit Test	**Test case ID**	TU19
Test case description	To verify if empty TextQuery is passed to search function inside indexing.py,it will give expected results or not [regression test]	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of search function.	**Post-requirement**	NA
Test Procedure		1.Use Assert function and check if the results we are getting by passing empty TextQuery to search function is an empty list or not	

Input	None
Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

20.

Test scenario ID	Unit Test	**Test case ID**	TU20
Test case description	To verify if the get_repo_and_action function inside main.py file is working properly or not	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of get_repo_and_action function.	**Post-requirement**	NA
Test Procedure		1.Use Assert function and check if the results we are getting by passing some request to get_repo_and_action function is matching with what we are expecting or not	
Input		String consisting URL	

Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

21.

Test scenario ID	Unit Test	**Test case ID**	TU21
Test case description	To verify whether there are bad characters in the lang parameter or not [By using env function of main.py to construct 'env' object]	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Get the request path and construct a webapp.Request object. 2. Apply setup_env function of main.py file on the above created webapp.Request object and get the env object. 3.Using Assert to check whether the env object contains any bad characters	
Input		String consisting of request path	
Expected Result		Pass/Fail based on the conditions	
Actual Result			

Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

22.

Test scenario ID	Unit Test	**Test case ID**	TU22
Test case description	To verify whether the language_menu_options[0] can be used as the default	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Get the request path and construct a webapp.Request object. 2.Set different language_menu_options usin set_for_repo function of config.py file. 3.Use Assert to check whether first language can be used as the default or not	
Input		list of language_menu_options which will be passed to set_for_repo function.	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			

Executed By	
Date of Execution	

23.

Test scenario ID	Unit Test	**Test case ID**	TU23
Test case description	To verify whether Content Security Policy(CSP) is set when the React UI is enabled.	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA

Test Procedure	1.Create response from webapp.response() library and handler. 2.Use Assert function to verify whether CSP is there in the response's header or not.
Input	None
Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

24.

Test scenario ID	Unit Test	**Test case ID**	TU24
Test case description	To verify whether the function convert_description_to_other [which converts description in PFIF 1.4 format to other older version] in pfif.py file is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function to be tested	**Post-requirement**	NA
Test Procedure		1.Use Assert function to check whether the result we are getting by passing some string to convert_description_to_other function is matching with what we are expecting or not.	
Input		Strings which will be passed as a parameter to function	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			

Date of Execution	

25.

Test scenario ID	Unit Test	**Test case ID**	TU25
Test case description	To verify whether the function maybe_convert_other_to_description [Converts 'other' in PFIF 1.3 and earlier to 'description' in PFIF 1.4 if 'other' has only 'description' field. Otherwise it returns 'other' without 'other' has only 'description' field. Otherwise it returns 'other' without] in pfif.py file is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function to be tested	**Post-requirement**	NA
Test Procedure		1.Use Assert function to check whether the result we are getting by passing some string to maybe_convert_other_to_description function is matching with what we are expecting or not.	
Input		None	

Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

26.

Test scenario ID	Unit Test	**Test case ID**	TU26
Test case description	To verify parsing of an XML file for each test case.	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Create person_records, note_records by passing test_case.xml file to parse_file function of pfif.py file 2.Use Assert to verify the person_records and note_records we got are matching with the expected results or not	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			

Executed By	
Date of Execution	

27.

Test scenario ID	Unit Test	**Test case ID**	TU27
Test case description	To verify whether test data is cached in ram or not	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA

Test Procedure	1.Create an instance of cache by using RamCache() class. 2.Put some data (key,value and ttl)in cache by using put function of RamCache() class. 3.Using Assert function check whether by passing key to get function of RamCache() class you are getting its associated value or not.
Input	None
Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	

Date of Execution	

28.

Test scenario ID	Unit Test	**Test case ID**	TU28
Test case description	To verify that test data for which ttl (time to leave) set to 0 will not be cached in ram	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.Create an instance of cache by using RamCache() class. 2.Put some data (key,value and ttl=0)in cache by using put function of RamCache() class. 3.Using Assert function check whether by passing key to get function of RamCache() class you are getting None or not	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			

Date of Execution	

29.

Test scenario ID	Unit Test	**Test case ID**	TU29
Test case description	To verify whether this cached data expires after ttl or not	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA

Test Procedure	1.Create an instance of cache by using RamCache() class. 2.Put some data (key,value and ttl)in cache by using put function of RamCache() class. 3.Using Assert function before and after expiration of ttl to check whether you are getting value or not.
Input	None
Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	
Date of Execution	

30.

Test scenario ID	Unit Test	**Test case ID**	TU30
Test case description	To check whether the clear function to clear the cache is working properly or not	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.Create an instance of cache by using RamCache() class. 2.Put some data (key,value and ttl)in cache by using put function of RamCache() class. 3.Using Assert function before and after cache and compare both results	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

31.

Test scenario ID	Unit Test	**Test case ID**	TU31
Test case description	To verify whether the get function inside Resources class of resources.py function is working properly or not. [To verify whether Resources.get fetches a Resource from the datastore]	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.First create a function that will put a resource in datastore for testing. 2. After that with the use of Assert function we can check whether that resources is there or not. [We can also write delete function and assert to check after that]	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			

Date of Execution	

32.

Test scenario ID	Unit Test	**Test case ID**	TU32
Test case description	To verify whether the set_active_bundle_name function inside resources.py function is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA

Test Procedure	1.First create a function that will put a resource in datastore for testing. 2.Check the results of Assert function used with get_localized and get_rendered before and after the set_active_bundle_name function is used
Input	None
Expected Result	Pass/Fail based on the conditions
Actual Result	
Status	
Remarks	
Created By	Hardik Rana
Date of creation	
Executed By	

Date of Execution	

33.

Test scenario ID	Unit Test	**Test case ID**	TU33
Test case description	To verify whether the get_localized function inside resources.py function is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.Use Assert function to verify by passing the name,lang to get_localized and self.fetched() function we are getting what we expected or not.	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

34.

Test scenario ID	Unit Test	**Test case ID**	TU34
Test case description	To verify whether the get_rendered function inside resources.py function is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.Use Assert function to verify by passing the name,lang and other parameters to get_rendered and self.fetched() function we are getting what we expected or not.	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

35.

Test scenario ID	Unit Test	**Test case ID**	TU35
Test case description	To verify whether the romaize_japanese_word function inside script_variant.py.py function is working properly or not	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.Using assert function to check whether the roman word we are getting by passing the japanese word to the function is same as actual roman word or not	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

36.

Test scenario ID	Unit Test	**Test case ID**	TU36
Test case description	To verify whether the romaize_word_by_unidecode function inside script_variant.py function is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.Using assert function to compare the result we are getting by passing word as an argument to function is same as what we expected or not	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

37.

Test scenario ID	Unit Test	**Test case ID**	TU37
Test case description	To verify whether the romaize_search_query function inside script_variant.py.py function is working properly or not.	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used	**Post-requirement**	NA
Test Procedure		1.Using assert function to compare the result we are getting by passing word as an argument to function is same as what we expected or not.	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

38.

Test scenario ID	Unit Test	**Test case ID**	TU38
Test case description	To verify whether the send_email functions error handling is working properly or not	**Test priority**	High
Prerequisite	None	**Post-requirement**	NA
Test Procedure		1.Take subject, receiver and sender email_address etc.. details. 2.Send mail using send_mail function and check whether there is any error or not.	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

39.

Test scenario ID	Unit Test	**Test case ID**	TU39
Test case description	To verify if the email is empty then false is returned or not.	**Test priority**	High
Prerequisite	None	**Post-requirement**	NA
Test Procedure		1.Use the set function of config.py file and set notification_email as empty message and unreviewed_notes_threshold 2. Create Handler and using Assert,check whether handler should_notify or not	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

40.

Test scenario ID	Unit Test	**Test case ID**	TU40
Test case description	To verify whether results of handler to call should_notify changed based on threshold or not	**Test priority**	High
Prerequisite	None	**Post-requirement**	NA
Test Procedure		1.Steps will be same as what will be used in previous test case. 2. While using Assert different-different threshold values will be used.	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

2. Server Tests [Functional Testing]

41.

Test scenario ID	Server Test	**Test case ID**	TU41
Test case description	Test that verifies that a config will return None, and not throw an exception, when asked for the value of an unknown key	**Test priority**	High
Prerequisite	User should be aware of the parameters passed to the function and return values of function which will be used while writing tests.	**Post-requirement**	NA
Test Procedure		1.First Set configuration settings for a particular repository. 2.Use Assert to verify whether it throws an error or not when the value for unknown key will be asked	
Input		Repo name and other parameters which will be passed to set_for_repo function	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			

Executed By	
Date of Execution	

42.

Test scenario ID	Server Test	**Test case ID**	TU42
Test case description	Test that verifies that in a config file a get will work with default value or not	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.First Set configuration settings for a particular repository. 2.Use Assert to verify whether the results we get by passing default_value and unknown_key to get function is equal to default_value or not	
Input		Repo name and other parameters which will be passed to set_for_repo function	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

43.

Test scenario ID	Server Test	**Test case ID**	TU43
Test case description	Test to verify that configuration instance is not callable	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Call the Configuration class constructor by providing repo name and store it in a variable. 2.For variable which we got in step1,using Assert verify that it is callable or not	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

44.

Test scenario ID	Server Test	**Test case ID**	TU44
Test case description	Test to verify download of xml file	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Get url (location) from where you want to download file. 2.Use download_feed library and download file at that stored at that location 3.Using Assert check some of fields that they are matching with what you are expecting or not.	
Input		String consisting of Url from where you want to download xml file	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

45.

Test scenario ID	Server Test	**Test case ID**	TU45
Test case description	Test to verify download of csv file	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Get url (location) from where you want to download file. 2.Use download_feed library and download file at that stored at that location 3.Using Assert check some of fields that they are matching with what you are expecting or not.	
Input		String consisting of Url from where you want to download csv file	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

46.

Test scenario ID	Server Test	**Test case ID**	TU46
Test case description	Test to verify download of notes	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Get url (location) from where you want to download file. 2.Use download_feed library and download file at that stored at that location 3.Using Assert check some of fields that they are matching with what you are expecting or not	
Input		String consisting of Url from where you want to download notes	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

47.

Test scenario ID	Server Test	**Test case ID**	TU47
Test case description	Test which will Verify that an error message is shown when no CSV file is uploaded.	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Get the id where file will be uploaded. 2.Use assert function to show error message when no csv file is uploaded	
Input		None	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

48.

Test scenario ID	Server Test	**Test case ID**	TU48
Test case description	Test which will Verify that an error message is shown when broken CSV file is uploaded.	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Use the same steps performed in previous test and check whether the csv file is broken or not	
Input		Broken csv file (which contains nul)	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

49.

Test scenario ID	Server Test	**Test case ID**	TU49
Test case description	Test to Verify that a photo is uploaded and properly served on the server.	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Create a new person record with a profile photo 2.Verify the image is uploaded and displayed on the view page. 3.Verify the image is served properly by checking the image metadata using Assert 4.Follow the link on the image and verify the same image is served using Assert	
Input		Png/jpg image	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

50.

Test scenario ID	Server Test	**Test case ID**	TU50
Test case description	Test to Upload an empty image and verifying that there will be no img tag in the view page	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Create a new person record with a zero-byte profile photo. 2.Verify there is no img tag in the view page using Assert	
Input		Empty image	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

51.

Test scenario ID	Server Test	**Test case ID**	TU51
Test case description	Test that Uploads both profile photo and note photo and verifies the images are properly transformed and served on the server[i.e.,jpg is converted to png and a large image is resized to match MAX_IMAGE_DIMENSION.]	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Create a new person record with a profile photo and a note photo. 2.Verify the images are uploaded and displayed on the view page. 3.Verify the profile image is converted to png using Assert. 4.Verify the note image is resized to match MAX_IMAGE_DIMENSION using Assert	
Input		Profile photo and note photo	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

52.

Test scenario ID	Server Test	**Test case ID**	TU52
Test case description	Test which Uploads a broken image and verifies an error message is displayed.	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Create a new person record with a broken profile photo. 2.Verify an error message is displayed using Assert	
Input		Broken image	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

53.

Test scenario ID	Server Test	**Test case ID**	TU53
Test case description	Test which Check the home page with no config (generic welcome page)	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Go to home page url 2.Use Assert to verify the text on home page	
Input		Home page url	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

54.

Test scenario ID	Server Test	**Test case ID**	TU54
Test case description	Test which Checks the start page with no language specified	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Go to Url of start page 2.Use Assert to verify the text on home page	
Input		Url of start page	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

55.

Test scenario ID	Server Test	**Test case ID**	TU55
Test case description	Test which Checks the start page with English language specified	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1.Go to Url of start page with english language specified. 2.Use Assert to verify the text on home page	
Input		Url of start page with english language specified	
Expected Result		Pass/Fail based on the conditions	
Actual Result			
Status			
Remarks			
Created By		Hardik Rana	
Date of creation			
Executed By			
Date of Execution			

3. Usability Tests [Non-Functional Testing]

56.

Test scenario ID	Usability Test	**Test case ID**	TU56
Test case description	Search for a missing person	**Test priority**	High
Prerequisite	1. The website should be opened in a compatible web browser 2. Record of the missing person should exist	**Post-requirement**	NA
Test Procedure		1. On the home page of the website, click on 'I'm looking for someone 2. Enter the name of the person 3. Click on the appropriate record	
Input		1. Name of the person to be searched	
Expected Result		The user is able to find the person	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

57.

Test scenario ID	Usability Test	**Test case ID**	TU57
Test case description	Choose the Japan disaster and create a new record in it	**Test priority**	High
Prerequisite	1. The Japan disaster should exist	**Post-requirement**	NA
Test Procedure		1. Click on the 3 horizontal bars on top left 2. Select Japan 3. Click on 'I have information about someone' 4. Enter name of the person 5. Enter details of the person	
Input		Details for the new record	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

58.

Test scenario ID	Usability Test	**Test case ID**	TU58
Test case description	Provide latest information of a missing person after record already exists	**Test priority**	High
Prerequisite	1. Record of person must already exist	**Post-requirement**	NA
Test Procedure		1. Click on ‘I have information about someone’ 2. Enter name of the person 3. Click on the correct record 4. Click on ‘I have information about someone’ 5. Enter the details	
Input		Name of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

59.

Test scenario ID	Usability Test	**Test case ID**	TU59
Test case description	Insert information of yourself as a missing person	**Test priority**	High
Prerequisite	1. There should be a disaster entry	**Post-requirement**	NA
Test Procedure		1. Click on the 3 horizontal bars on top left 2. Select Japan 3. Click on ‘I have information about someone’ 4. Enter your 5. Enter details	
Input		Details for entering into the record	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

60.

Test scenario ID	Usability Test	**Test case ID**	TU60
Test case description	Upload photo of missing person	**Test priority**	High
Prerequisite	1. Record of the missing person must exist	**Post-requirement**	NA
Test Procedure		1. Click on 'I have information about someone' 2. Enter the name of the person 3. Click on the correct record 4. Click on 'I have information about this person' 5. Click on 'More information about status of the person' 6. Click on URL or Upload and then select the image 7. Enter your details and then click Submit	
Input		Photo in valid format	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

61.

Test scenario ID	Usability Test	**Test case ID**	TU61
Test case description	Delete a specific record	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on ‘I have information about someone’ 2. Enter the name of the person 3. Click on the correct record 4. Click on Delete this record	
Input		Name of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

62.

Test scenario ID	Usability Test	**Test case ID**	TU62
Test case description	Extend expiry date of specified record	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on 'I have information about someone' 2. Enter the name of the person 3. Click on the correct record 4. Click on 'Extend expiration date by 60 days'	
Input		Name of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

63.

Test scenario ID	Usability Test	**Test case ID**	TU63
Test case description	Subscribe to Updates about missing person	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on 'I have information about someone' 2. Enter the name of the person 3. Click on the correct record 4. Click on 'Subscribe to updates about this person'	
Input		Name of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

64.

Test scenario ID	Usability Test	**Test case ID**	TU64
Test case description	Provide latest known location of the missing person using map	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on 'I have information about someone' 2. Enter the name of the person 3. Click on the correct record 4. Click on 'I have information about this person' 5. Click on 'More information about status of the person' 6. In the drop down section click on Show Map and then select the location on the map	
Input		Name of the person, Location of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

65.

Test scenario ID	Usability Test	**Test case ID**	TU65
Test case description	Provide latest known location of the missing person using address	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on 'I have information about someone' 2. Enter the name of the person 3. Click on the correct record 4. Click on 'I have information about this person' 5. Click on 'More information about status of the person' 6. In the drop down section enter the address in the text box	
Input		Name of the person,Location of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

66.

Test scenario ID	Usability Test	**Test case ID**	TU66
Test case description	Provide latest known location of the missing person using current location	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on ‘I have information about someone’ 2. Enter the name of the person 3. Click on the correct record 4. Click on ‘I have information about this person’ 5. Click on ‘More information about status of the person’ 6. In the drop down section click on Use Current Location	
Input		Name of the person,Location of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

67.

Test scenario ID	Usability Test	**Test case ID**	TU67
Test case description	Mark a note as spam	**Test priority**	High
Prerequisite	Record and a note in the record must exist	**Post-requirement**	NA
Test Procedure		1. Click on 'I have information about someone' 2. Enter the name of the person 3. Click on the correct record 4. Click on 'Report Spam' 5. Specify the reason and verify the captcha	
Input		Name of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

68.

Test scenario ID	Usability Test	**Test case ID**	TU68
Test case description	Leave a feedback	**Test priority**	High
Prerequisite	NA	**Post-requirement**	NA
Test Procedure		1. Click on the three horizontal bars 2. Click on feedback 3. Fill the Google form	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

69.

Test scenario ID	Usability Test	**Test case ID**	TU69
Test case description	Find phone number of the person specified	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on ‘I am looking for someone’ 2. Enter name of person 3. Select the person 4. Next to the phone number label click ‘Click to reveal’ 5. Verify the captcha	
Input		Name of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

70.

Test scenario ID	Usability Test	**Test case ID**	TU70
Test case description	Find email id of the person specified	**Test priority**	High
Prerequisite	Record must exist	**Post-requirement**	NA
Test Procedure		1. Click on ‘I am looking for someone’ 2. Enter name of person 3. Select the person 4. Next to the email id label click ‘Click to reveal’ 5. Verify the captcha	
Input		Name of the person	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

4. Load Tests [Non-Functional Testing]

71.

Test scenario ID	Load Test	**Test case ID**	TU71
Test case description	What happens if a sudden spike of users that the website cannot handle happens?	**Test priority**	High
Prerequisite	Website should be running	**Post-requirement**	NA
Test Procedure		1) Find the Max User Load capacity of your software application 2)Create a Testing Environment and configure it to record performance parameters. 3) Set up test scenarios based on expected maximum load to your Software Application using a Performance Tool 4) Rapidly increase in load, to the system for a set period. 5) Gradually reduce the load back to its original level. 6) Analyze the performance graphs. Metric to be considered are Failures, Time Taken, Virtual Users, etc.	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	

Date of creation	
Executed By	
Date of Execution	

72.

Test scenario ID	Load Test	**Test case ID**	TU72
Test case description	What if a high number of users are working with the API/website for an extended period?	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Create a dedicated test environment 2. Predict the following and set up test scenarios: -Number of users -Duration of connection 3. Run the test and measure the results	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			

Executed By	
Date of Execution	

73.

Test scenario ID	Load Test	**Test case ID**	TU73
Test case description	Check if the page load time is within the acceptable range	**Test priority**	High
Prerequisite	1. There must be an internet connection 2. Web servers should be running	**Post-requirement**	NA
Test Procedure		1. Open a compatible browser 2. Enter the URL in the address bar and press 'Enter' 3. Measure the amount of time taken by the website to load using a stopwatch	
Input		URL of the website	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

74.

Test scenario ID	Load Test	**Test case ID**	TU74
Test case description	Check the response time for any action under a light load conditions.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Create a dedicated test environment 2. Predict the following and set up test scenarios: -Number of users -Connection speeds 3. Decide on a specific action(eg: login) to be performed 4. Measure the time taken for the login to happen	
Input		Depends on the action to be performed	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

75.

Test scenario ID	Load Test	**Test case ID**	TU75
Test case description	Check the response time for any action under a moderate load conditions.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Create a dedicated test environment 2. Predict the following and set up test scenarios: -Number of users -Connection speeds 3. Decide on a specific action(eg: login) to be performed 4. Measure the time taken for the login to happen	
Input		Depends on the action to be performed	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

76.

Test scenario ID	Load Test	**Test case ID**	TU76
Test case description	Check the response time for any action under a heavy load conditions.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Create a dedicated test environment 2. Predict the following and set up test scenarios: -Number of users -Connection speeds 3. Decide on a specific action(eg: login) to be performed 4. Measure the time taken for the login to happen	
Input		Depends on the action to be performed	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

77.

Test scenario ID	Load Test	**Test case ID**	TU77
Test case description	Check CPU and memory usage under peak load conditions.	**Test priority**	High
Prerequisite	Tools should be set up to monitor the CPU and memory usage	**Post-requirement**	NA
Test Procedure		1. Create a dedicated test environment 2. Predict the following and set up test scenarios: -Number of users -Connection speeds 3. Decide on a specific action(eg: login) to be performed 4. Measure the CPU and memory usage for the login to happen	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

78.

Test scenario ID	Load Test	**Test case ID**	TU78
Test case description	Verify if a proper error message is shown when the system reaches the breakpoint i.e. crosses the maximum no. of permitted users or requests. The RAM, memory and network are all good.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Web servers must be running	**Post-requirement**	NA
Test Procedure		1. Gather the system data, analyze the system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts and store the stress results. 4. Analyze the Stress Test results.	
Input		None	
Expected Result		Fail	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	

Date of creation	
Executed By	
Date of Execution	

79.

Test scenario ID	Load Test	**Test case ID**	TU79
Test case description	Verify if a proper error message is shown when the system reaches the breakpoint i.e. crosses the maximum no. of permitted users or requests. The RAM, memory and network are all slow.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Web servers must be running	**Post-requirement**	NA
Test Procedure		1. Gather the system data, analyze the system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts and store the stress results. 4. Analyze the Stress Test results.	
Input		None	
Expected Result		Fail	

Actual Result	
Status	
Remarks	
Created By	Harshal Shinde
Date of creation	
Executed By	
Date of Execution	

80.

Test scenario ID	Load Test	**Test case ID**	TU80
Test case description	Verify if the system works as expected when maximum no. of users or requests are being processed.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Automation scripting tool must be set up	**Post-requirement**	NA
Test Procedure		1. Gather the system data, analyze the system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts and store the stress results. 4. Analyze the Stress Test results.	
Input		None	

Expected Result	Pass
Actual Result	
Status	
Remarks	
Created By	Harshal Shinde
Date of creation	
Executed By	
Date of Execution	

81.

Test scenario ID	Load Test	**Test case ID**	TU81
Test case description	Verify that while more than the permitted no. of users or requests are performing the same operation (like searching for a person, etc) and if the system becomes irresponsive, an appropriate error message is shown about the data	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Automation scripting tool must be set up	**Post-requirement**	NA
Test Procedure		1. Gather the system data, analyze the	

	system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts and store the stress results. 4. Analyze the Stress Test results.
Input	None
Expected Result	Pass
Actual Result	
Status	
Remarks	
Created By	Harshal Shinde
Date of creation	
Executed By	
Date of Execution	

82.

Test scenario ID	Load Test	**Test case ID**	TU82
Test case description	Check if more than the permitted no. of users or requests are performing different operation (searching for a person, adding information about someone etc) and if the system becomes irresponsive, an appropriate error message is shown about the data (not saved? – depends on the implementation).	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Automation scripting tool must be set up	**Post-requirement**	NA
Test Procedure		1. Gather the system data, analyze the system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts and store the stress results. 4. Analyze the Stress Test results.	
Input		None	
Expected Result		Fail	

Actual Result	
Status	
Remarks	
Created By	Harshal Shinde
Date of creation	
Executed By	
Date of Execution	

83.

Test scenario ID	Load Test	**Test case ID**	TU83
Test case description	Verify if the response time for breaking point users or requests is in an acceptance value.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Automation scripting tool must be set up	**Post-requirement**	NA
Test Procedure		1. Gather the system data, analyze the system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts and store the stress results. 4. Analyze the Stress Test results.	
Input		None	

Expected Result	Pass
Actual Result	
Status	
Remarks	
Created By	Harshal Shinde
Date of creation	
Executed By	
Date of Execution	

84.

Test scenario ID	Load Test	**Test case ID**	TU84
Test case description	Verify the performance of the app or website when the network is very slow, a proper error message should be shown for 'timeout' condition.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Automation scripting tool must be set up	**Post-requirement**	NA
Test Procedure		1. Gather the system data, analyze the system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts	

	and store the stress results. 4. Verify if timeout message is shown
Input	None
Expected Result	Fail
Actual Result	
Status	
Remarks	
Created By	Harshal Shinde
Date of creation	
Executed By	
Date of Execution	

85.

Test scenario ID	Load Test	**Test case ID**	TU85
Test case description	Verify all the above test cases for a server which has more than one application running on it to check if the other application gets affected etc.	**Test priority**	High
Prerequisite	There should be an internet connection and the page should be loaded on a compatible browser. Automation scripting tool must be set up	**Post-requirement**	NA

	The server must have 2 applications running on it		
Test Procedure		1. Gather the system data, analyze the system, define the stress test goals 2. Create the Stress testing automation scripts, generate the test data for the stress scenarios. 3. Run the Stress testing automation scripts and store the stress results. 4. Results Analysis: See the effects on the other applications execution.	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

5. GUI Tests [Non-Functional Testing]

86.

Test scenario ID	GUI Test	**Test case ID**	TU86
Test case description	Testing the size, position, width, height of the elements.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Verify if the size and position of the elements are correct	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

87.

Test scenario ID	GUI Test	**Test case ID**	TU87
Test case description	Testing of the error messages that are getting displayed.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Perform a transaction which would generate an error	
Input		1. Name of the person to be searched	
Expected Result		The user is able to find the person	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

88.

Test scenario ID	GUI Test	**Test case ID**	TU88
Test case description	Testing the different sections of the screen.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Perform various operations on the elements in different parts of the screen	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

89.

Test scenario ID	GUI Test	**Test case ID**	TU89
Test case description	Testing of the font whether it is readable or not.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Verify if the font is of a minimum specified size	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

90.

Test scenario ID	GUI Test	**Test case ID**	TU90
Test case description	Testing of the screen in different resolutions with the help of zooming in and zooming out like 640 x 480, 600x800, etc.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Zoom in and then verify the resolution 3. Zoom out and then verify the resolution	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

91.

Test scenario ID	GUI Test	**Test case ID**	TU91
Test case description	Testing the alignment of the texts and other elements like icons, buttons, etc. are in proper place or not.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Verify if the elements are in proper position	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

92.

Test scenario ID	GUI Test	**Test case ID**	TU92
Test case description	Testing the colors of the fonts.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Get the id of the text 2. Verify the color using assert	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

93.

Test scenario ID	GUI Test	**Test case ID**	TU93
Test case description	Testing the colors of the error messages, warning messages.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Generate the error/warning message 3. Get the id of the text 4. Verify the color using assert	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

94.

Test scenario ID	GUI Test	**Test case ID**	TU94
Test case description	Testing whether the image has good clarity or not.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Browse to the page with the image 3. Verify the clarity of the image by inspecting its resolution	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

95.

Test scenario ID	GUI Test	**Test case ID**	TU95
Test case description	Testing the alignment of the images.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Browse to the page with the image 3. Verify the image alignment	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

96.

Test scenario ID	GUI Test	**Test case ID**	TU96
Test case description	Testing of the scrollbars according to the size of the page if any.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Use the scrollbar to scroll through the page	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

97.

Test scenario ID	GUI Test	**Test case ID**	TU97
Test case description	Testing of the color of the hyperlink.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Generate the error/warning message 3. Get the id of the text 4. Verify the color using assert	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

98.

Test scenario ID	GUI Test	**Test case ID**	TU98
Test case description	Testing of the headings whether it is properly aligned or not.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Browse to the page with the image 3. Verify the headings' alignment	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

99.

Test scenario ID	GUI Test	**Test case ID**	TU99
Test case description	Testing of the size of the images.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Browse to the page with the image 3. Verify the size of the image	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			

100.

Test scenario ID	GUI Test	**Test case ID**	TU100
Test case description	Verify that after updating any field a proper confirmation message must be displayed.	**Test priority**	High
Prerequisite	The website must be opened on a compatible browser	**Post-requirement**	NA
Test Procedure		1. Open the website in a compatible browser 2. Update the field 3. Verify the confirmation message displayed	
Input		None	
Expected Result		Pass	
Actual Result			
Status			
Remarks			
Created By		Harshal Shinde	
Date of creation			
Executed By			
Date of Execution			