Task:

Apply a testing technique on a web application using selenium.

Here the task was implemented on a web page consisting of a simple registration form and for simplicity the page was hosted on localhost.

Testing technique:
"Boundary Value Testing" technique was chosen for this purpose.

Testing requirements:
Testing was done using selenium testing tool, and FireFox's gecko driver was used for testing purpose.

Testing fields:

The testing was done on several input fields from the form. Those input fields consisted different boundaries depending on their type. If the submit process of the form becomes successful and an login button appears. We have decided the success of the testing depending on the appearance of the login button. It appears only when the form is filled with valid values within boundary.

The input fields along with their boundaries are described in the table below:

Testing Field	Boundary
Username	User name must be between 4 to 10 characters and and it can only contain alphabets,no number. It must be filled.
Email	The email address must be of the valid format abc@company.com. It must be filled.
Age	Age must be between 15 to 60 for the purpose of registering in this site. It must be filled.
Mobile	It must be a 11 digit mobile number. This field must be filled.
Checkbox	The chekcbox can be either checked or unchecked. For submission purpose, it must be checked.

Boundary ValueTesting:

Now considering on the boundaries of different input field types several test cases were developed and they were tested using selenium webdriver. The test cases and their results are doucmented in the table below:

Test Case ID	Username	Email	Age	Mobile	CheckBox	Testing Output	Expected Output
1	abc	X	X	X	X	Failed	Failed
2	abcd	X	X	X	X	Passed	Passed
3	abcdefghij	X	X	X	X	Passed	Passed
4	abcd1234	X	X	X	X	Passed	Passed
5	abcdefghijk	X	X	X	X	Failed	Failed
6	“EMPTY”	X	X	X	X	Failed	Failed
7	12345	X	X	X	X	Failed	Failed
8	X	abc@xyz.com	X	X	X	Passed	Passed
9	X	abc@def.com	X	X	X	Passed	Passed
10	X	abc@.com	X	X	X	Failed	Failed
11	X	@.com	X	X	X	Failed	Failed
12	X	abcdefgh	X	X	X	Passed	Failed
13	X	1234567	X	X	X	Failed	Failed
14	X	abc@xyz	X	X	X	Failed	Failed
15	X	“EMPTY”	X	X	X	Passed	Failed
16	X	X	14	X	X	Failed	Failed
17	X	X	15	X	X	Failed	Passed
18	X	X	40	X	X	Passed	Passed
19	X	X	60	X	X	Passed	Passed
20	X	X	61	X	X	Failed	Failed
21	X	X	“EMPTY”	X	X	Failed	Failed
22	X	X	X	0162129752	X	Failed	Failed
23	X	X	X	01621297528	X	Failed	Passed
24	X	X	X	016212975289	X	Failed	Failed
25				“EMPTY”		Failed	Failed
26	X	X	X	X	checked	Passed	Passed
27	X	X	X	X	Not checked	Failed	Failed

* ‘X’ here indicates any random valid value inside the respective boundary

We can see from the table above that some test cases failed which should have passed and similarly some test cases passed which should have failed.