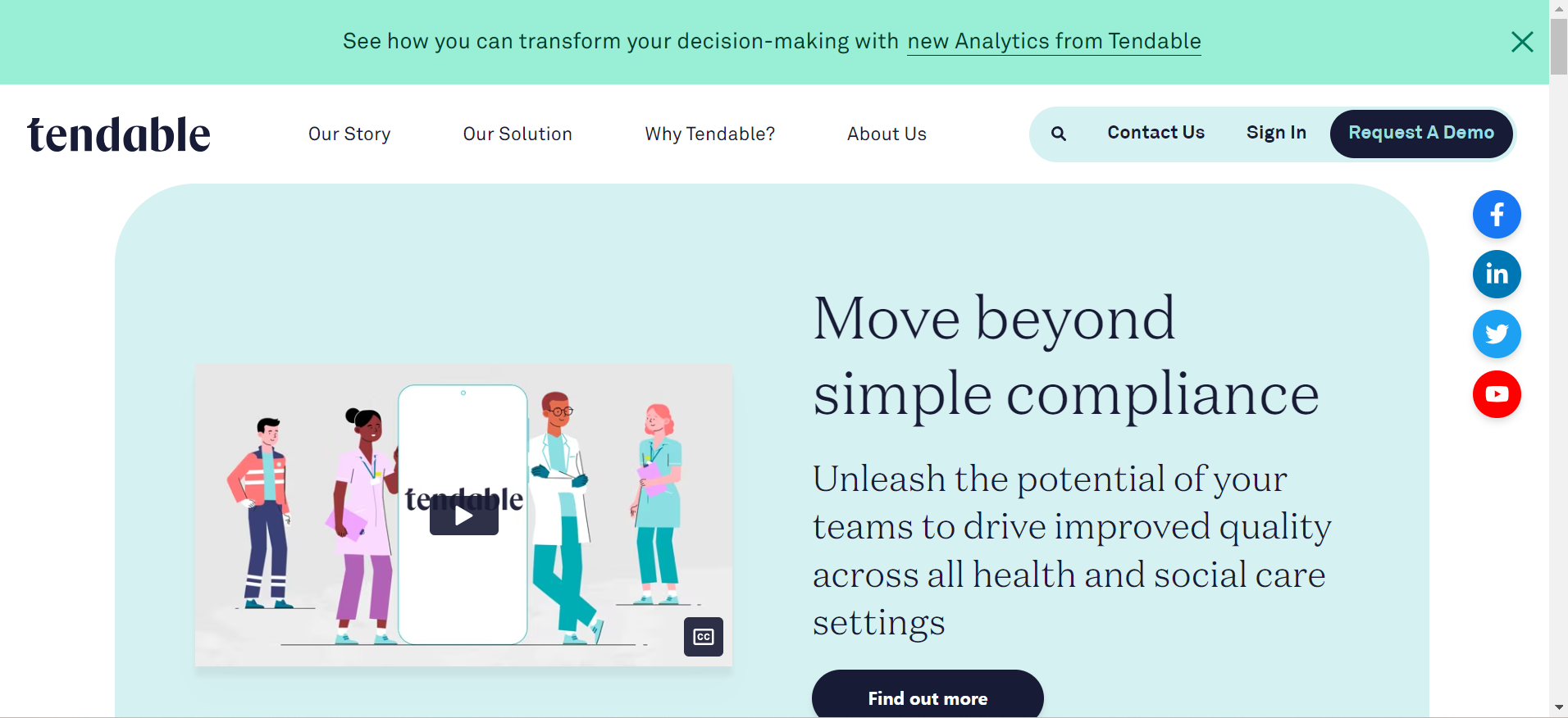
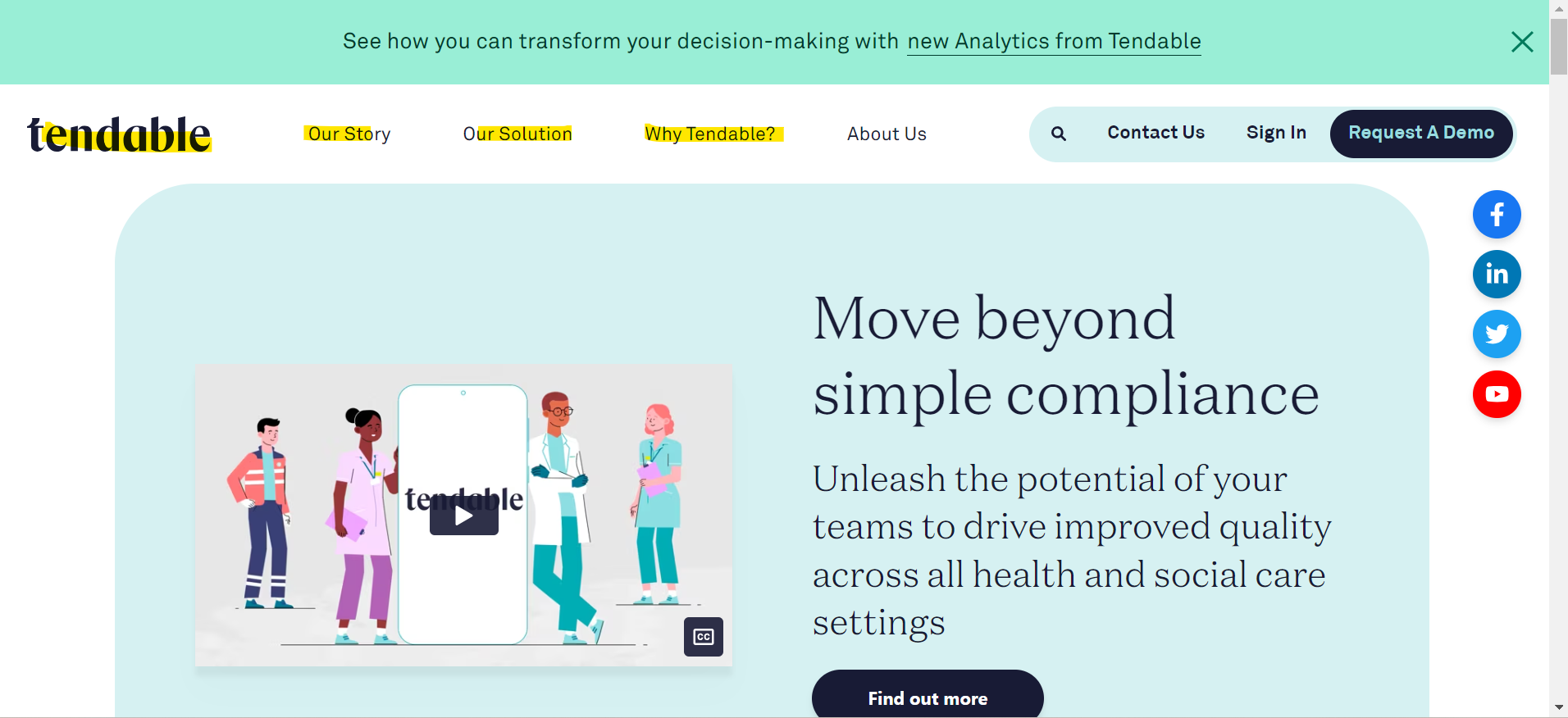
**Execution**

Scenario: End to End Testing for Tendable Application

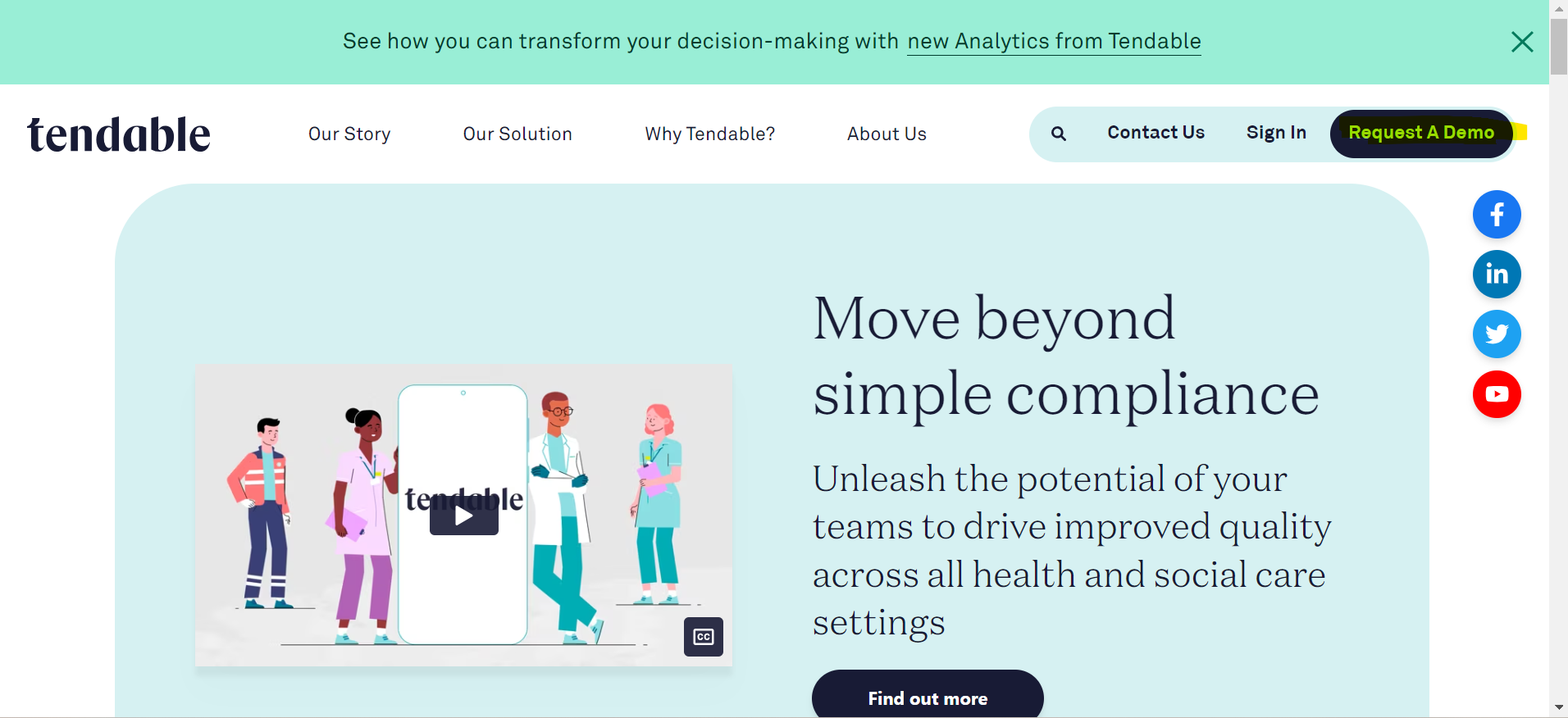
Given user is on SignIn page



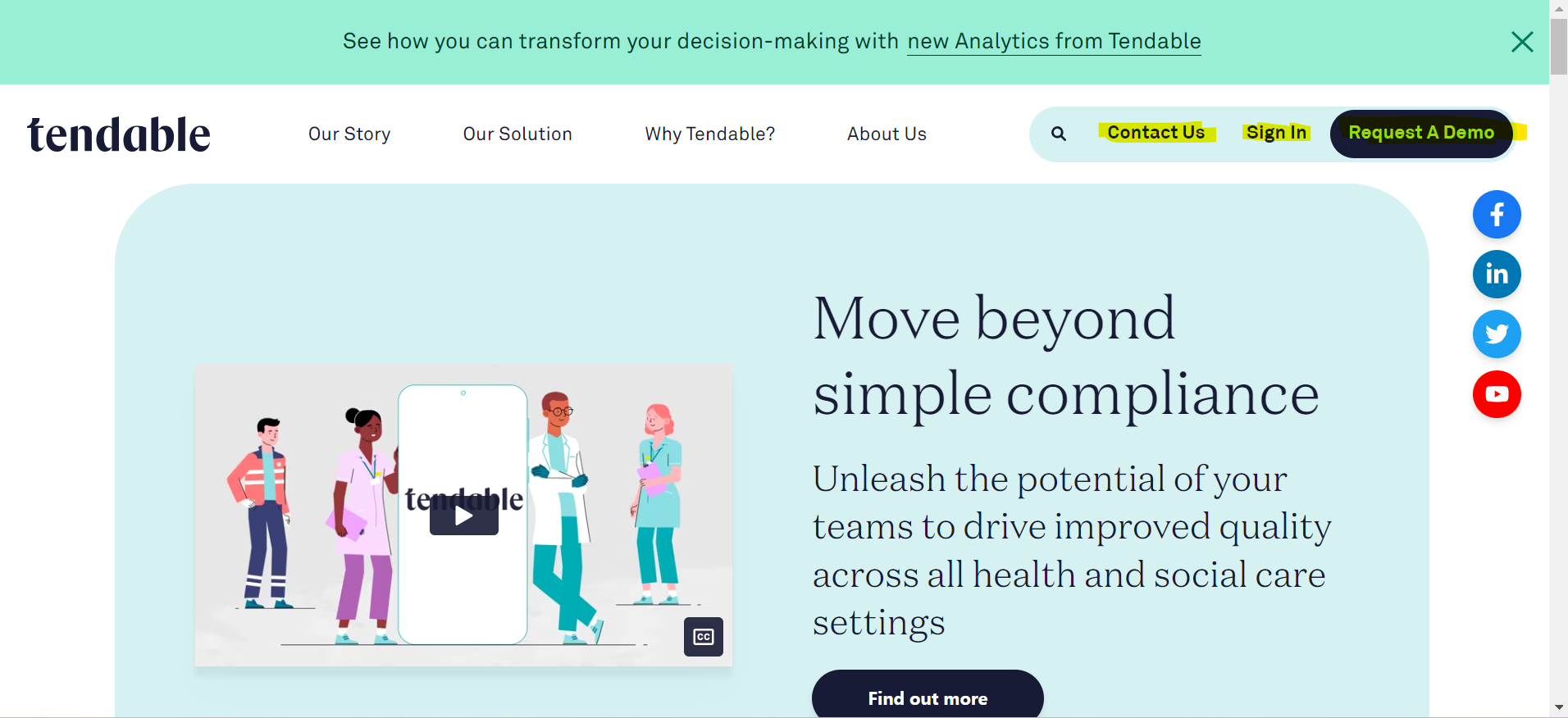
When user verify Home Page Title, Our Story , Our Solution & Why Tendable Is Enabled



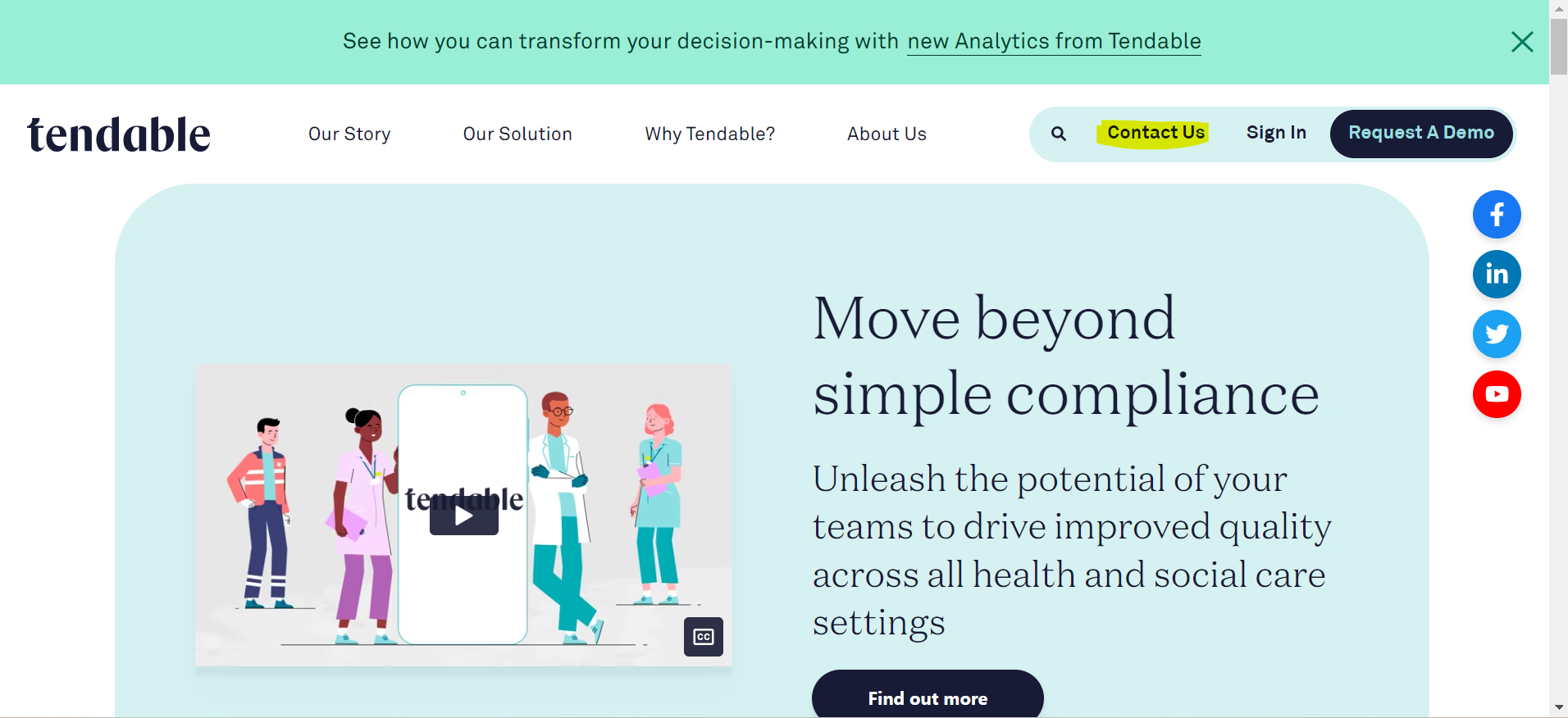
Then user Request a Demo button is displayed on home page



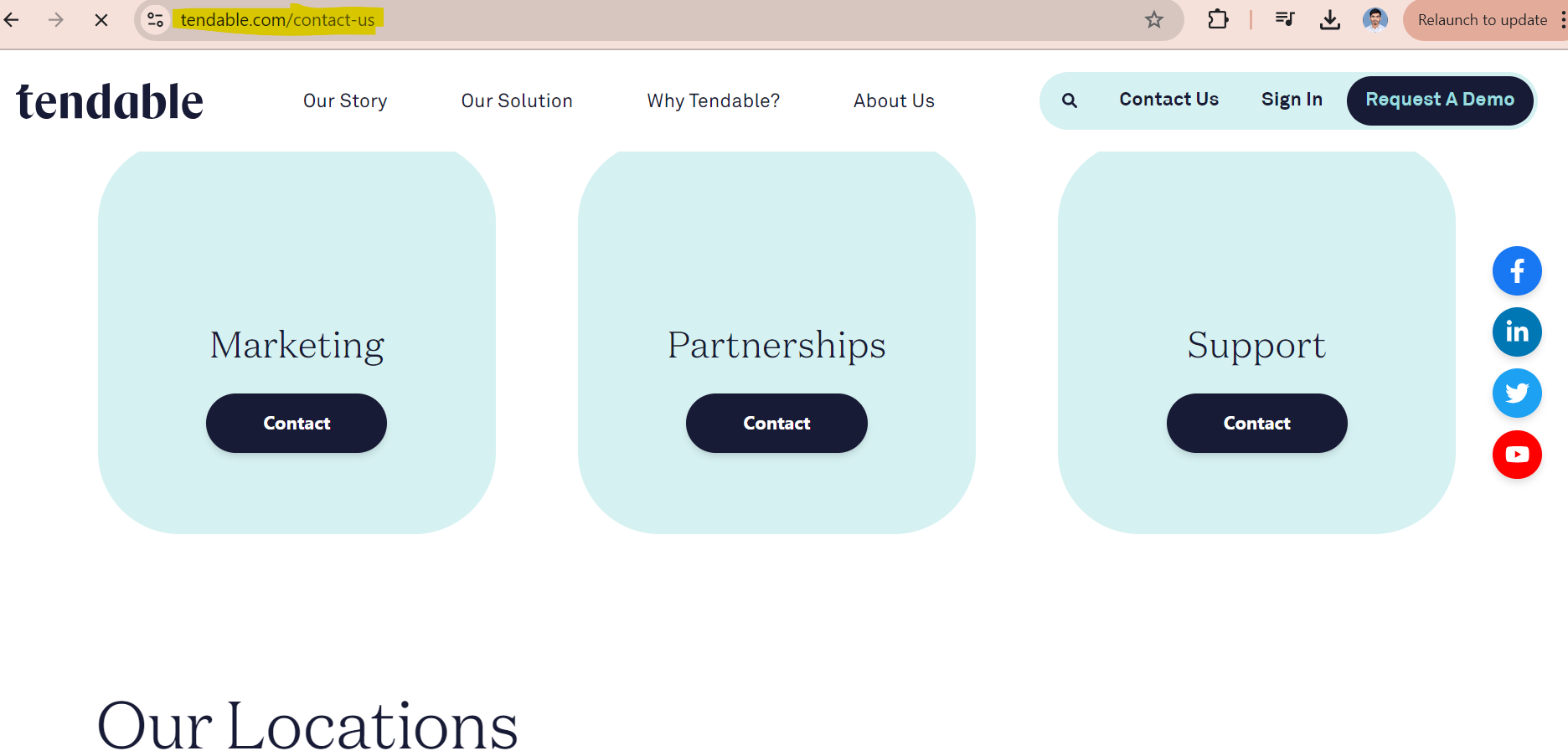
And user verify top level button Enabled



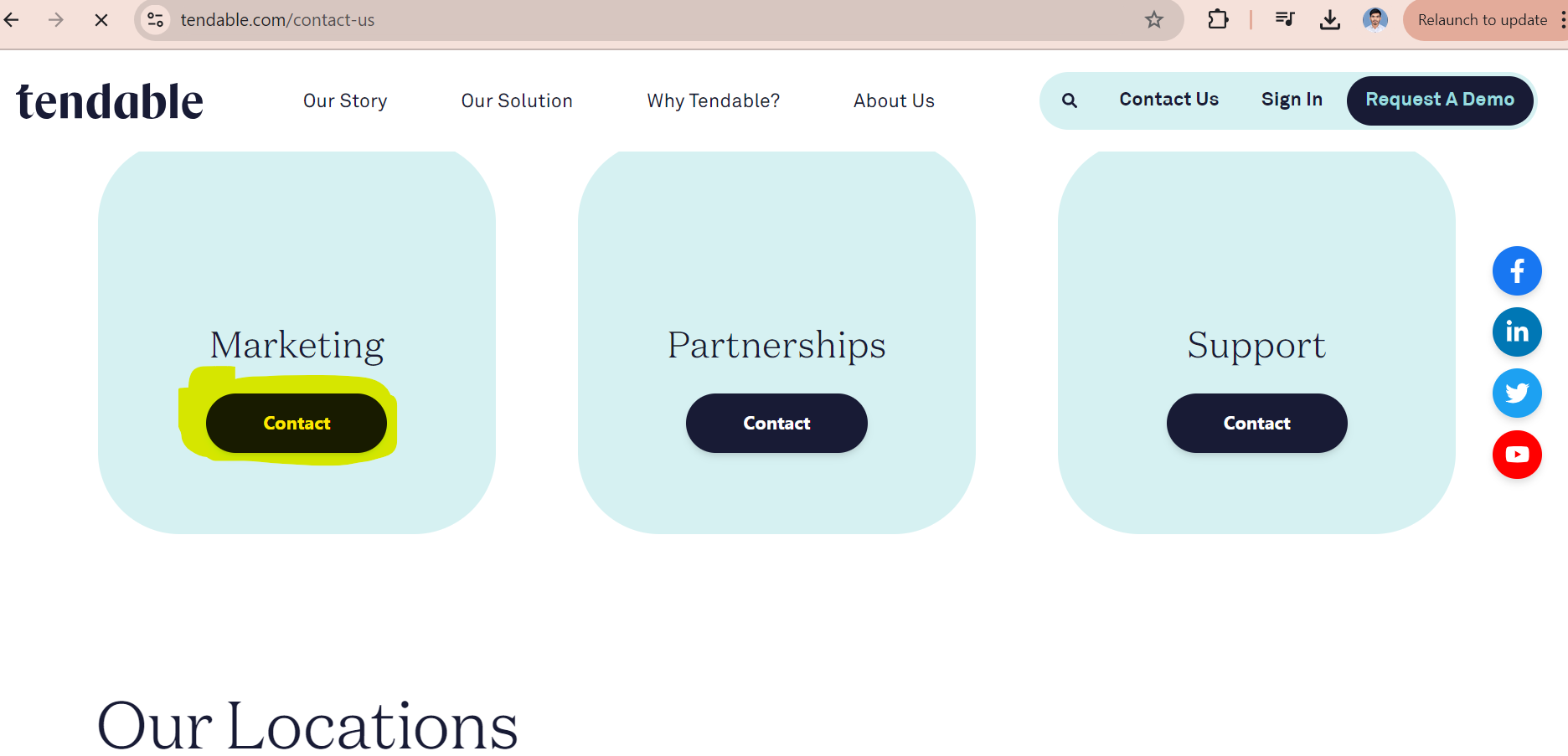
And user click on contactUs button



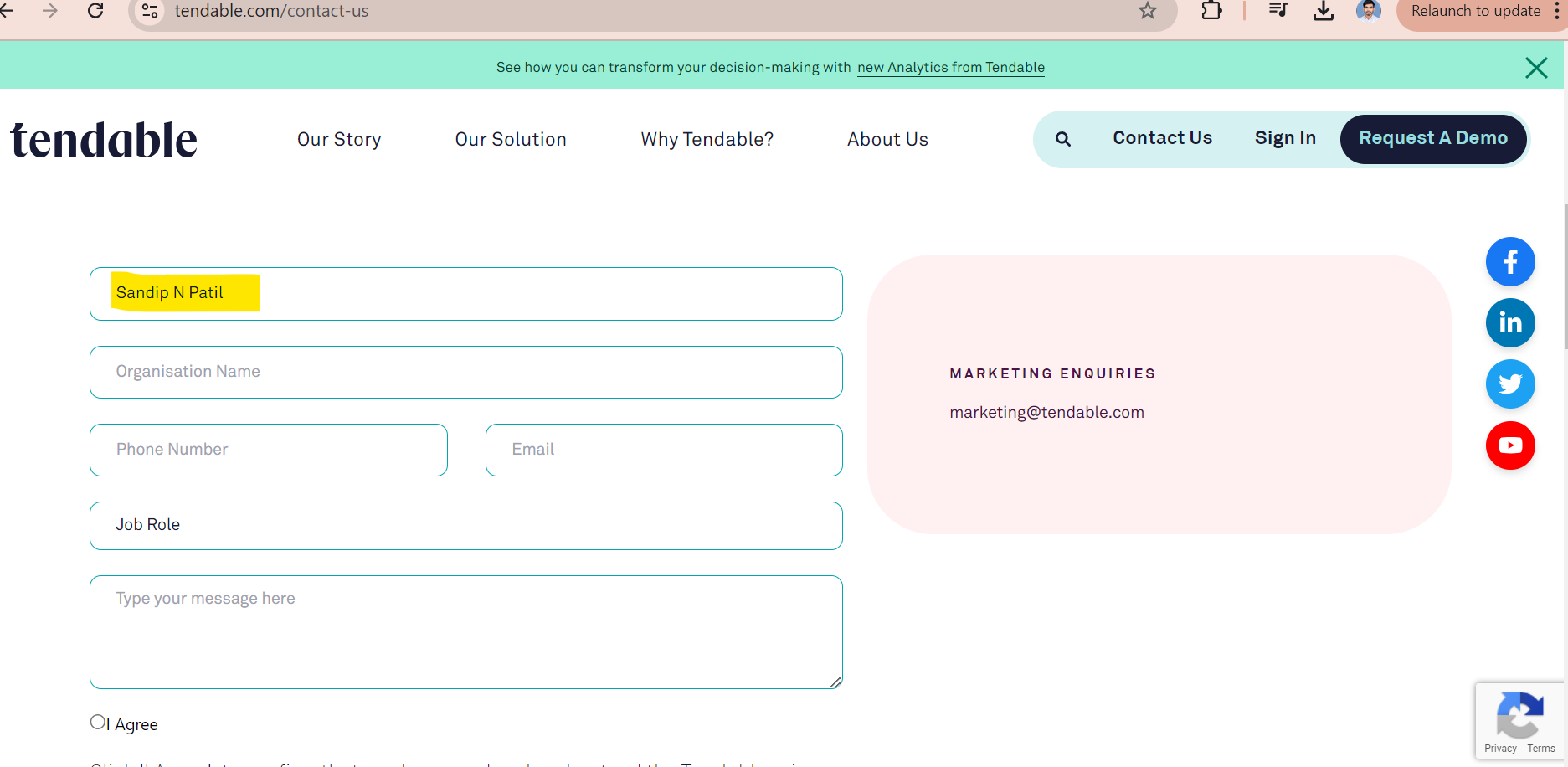
And user verify system redirected to Contact Us page



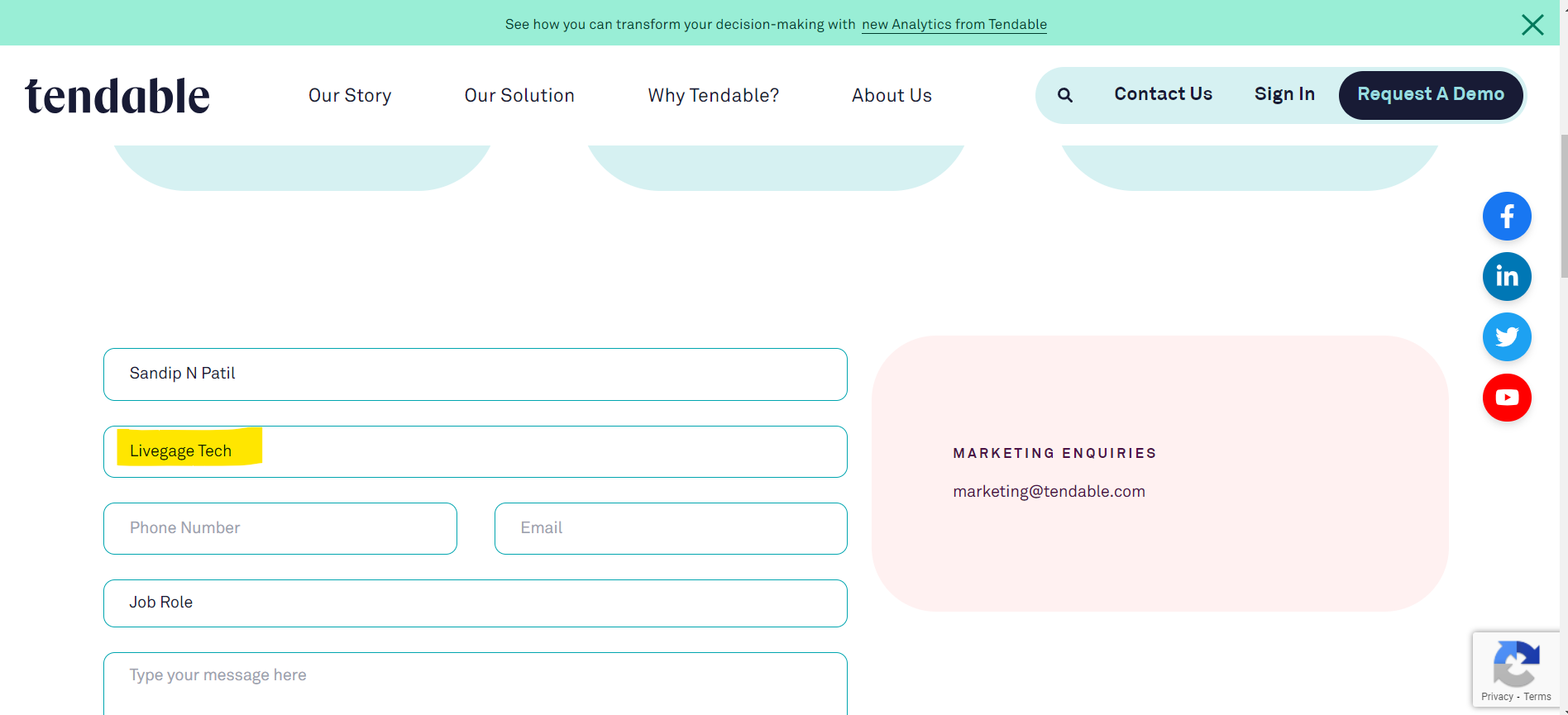
And user click on Contact at Marketing option



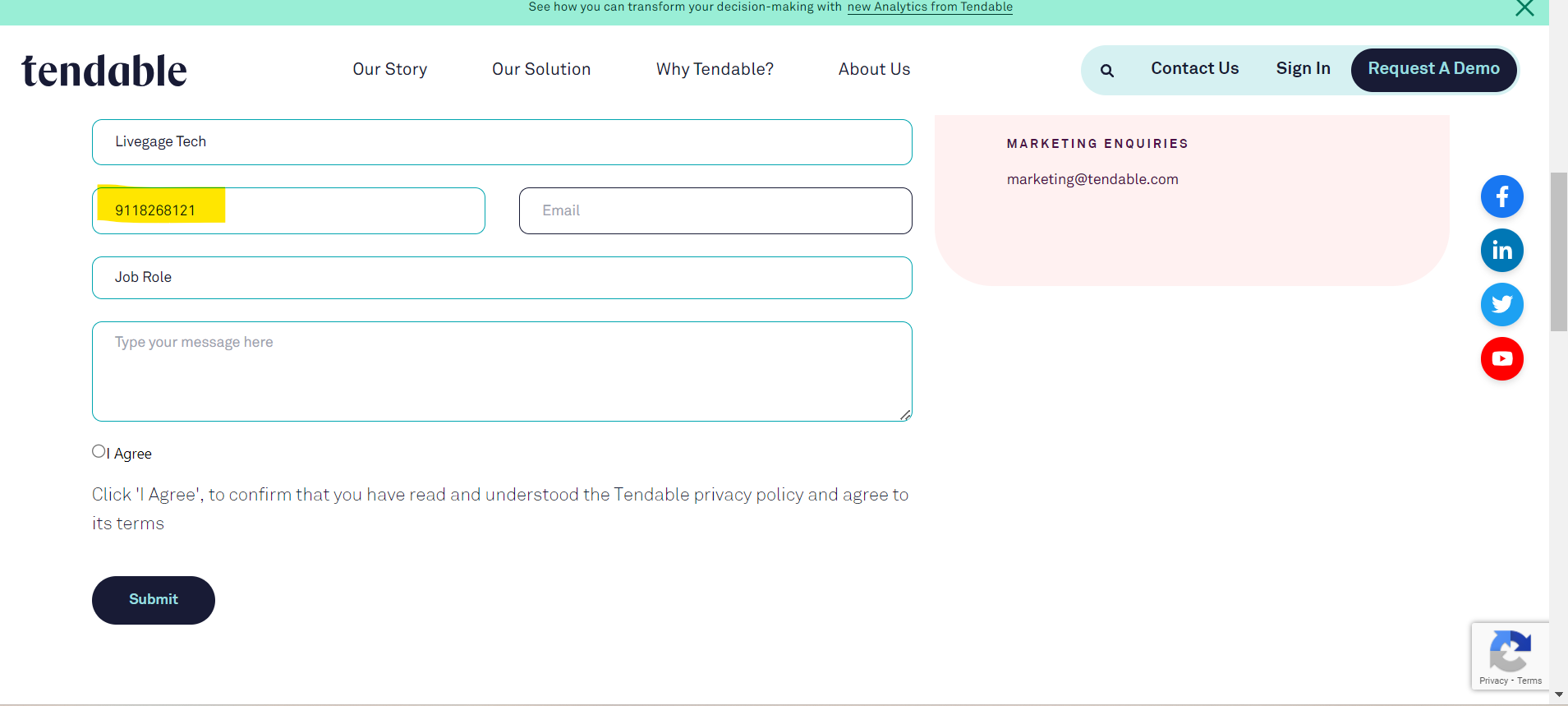
And Enter full name



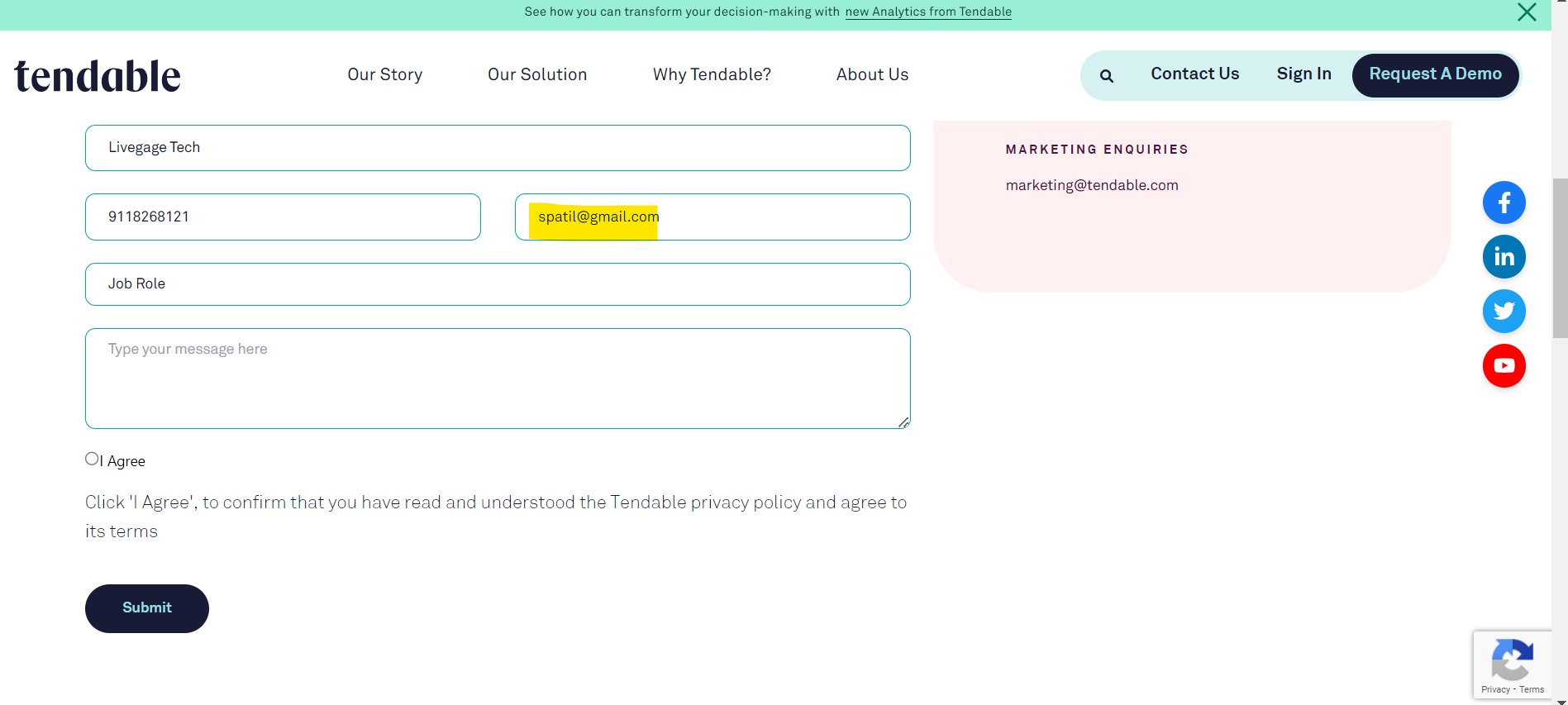
And Enter Oranization Name



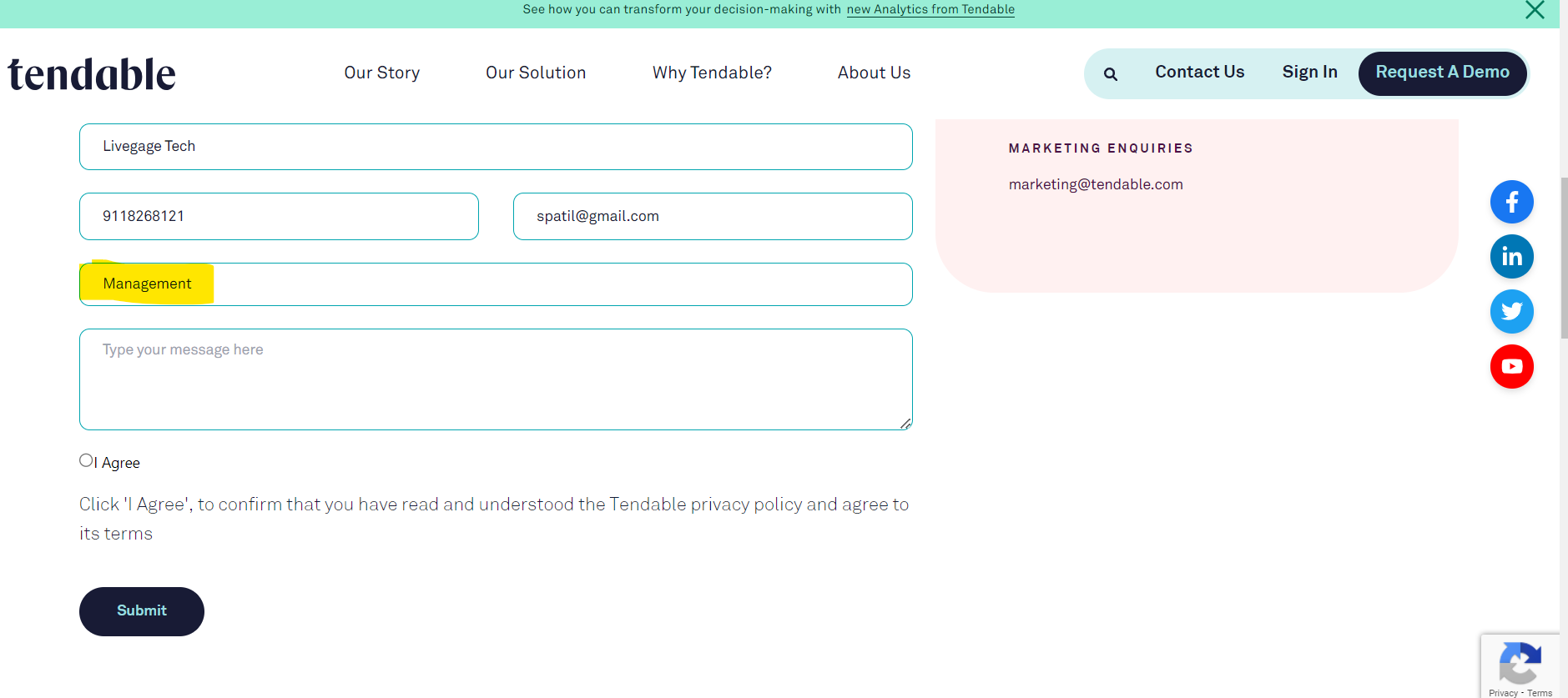
And Enter phone number



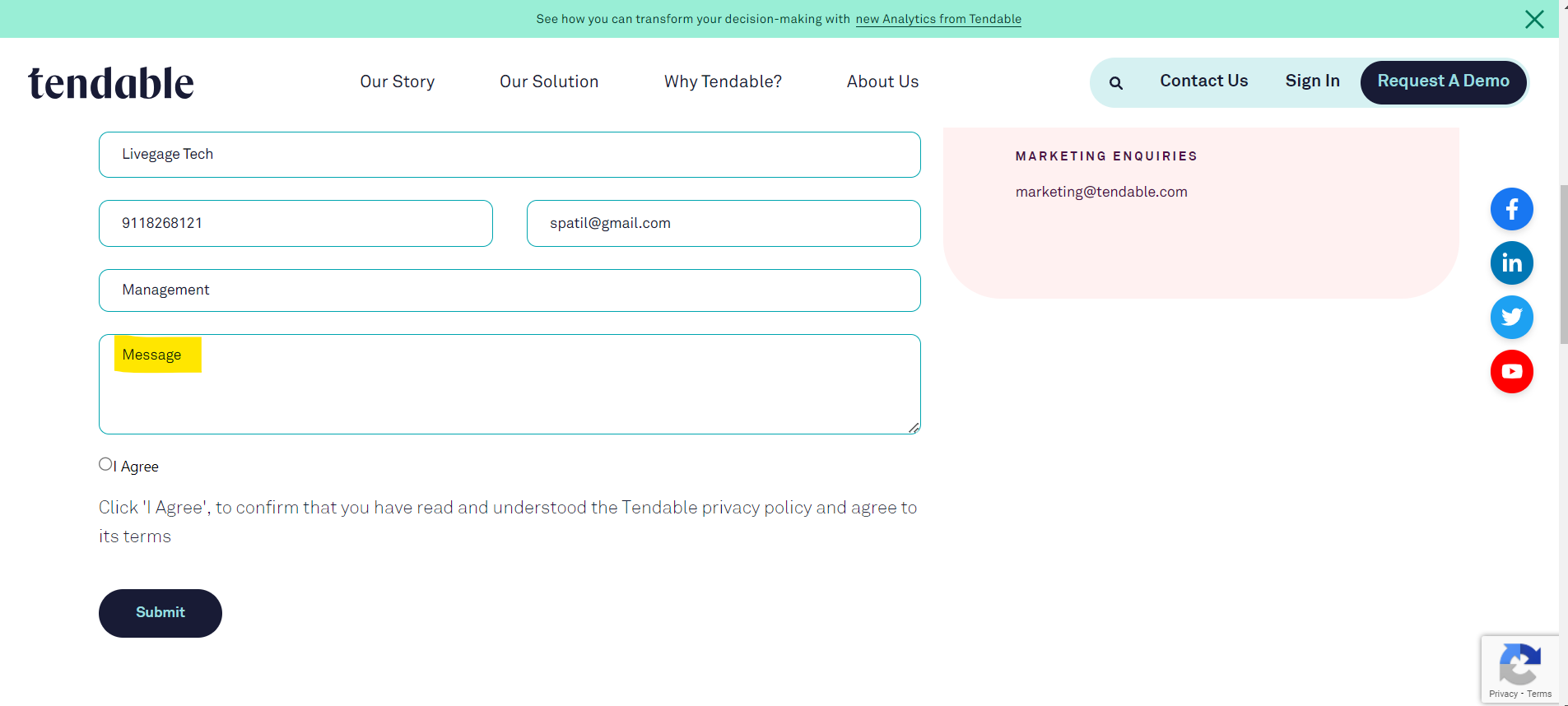
And Enter email



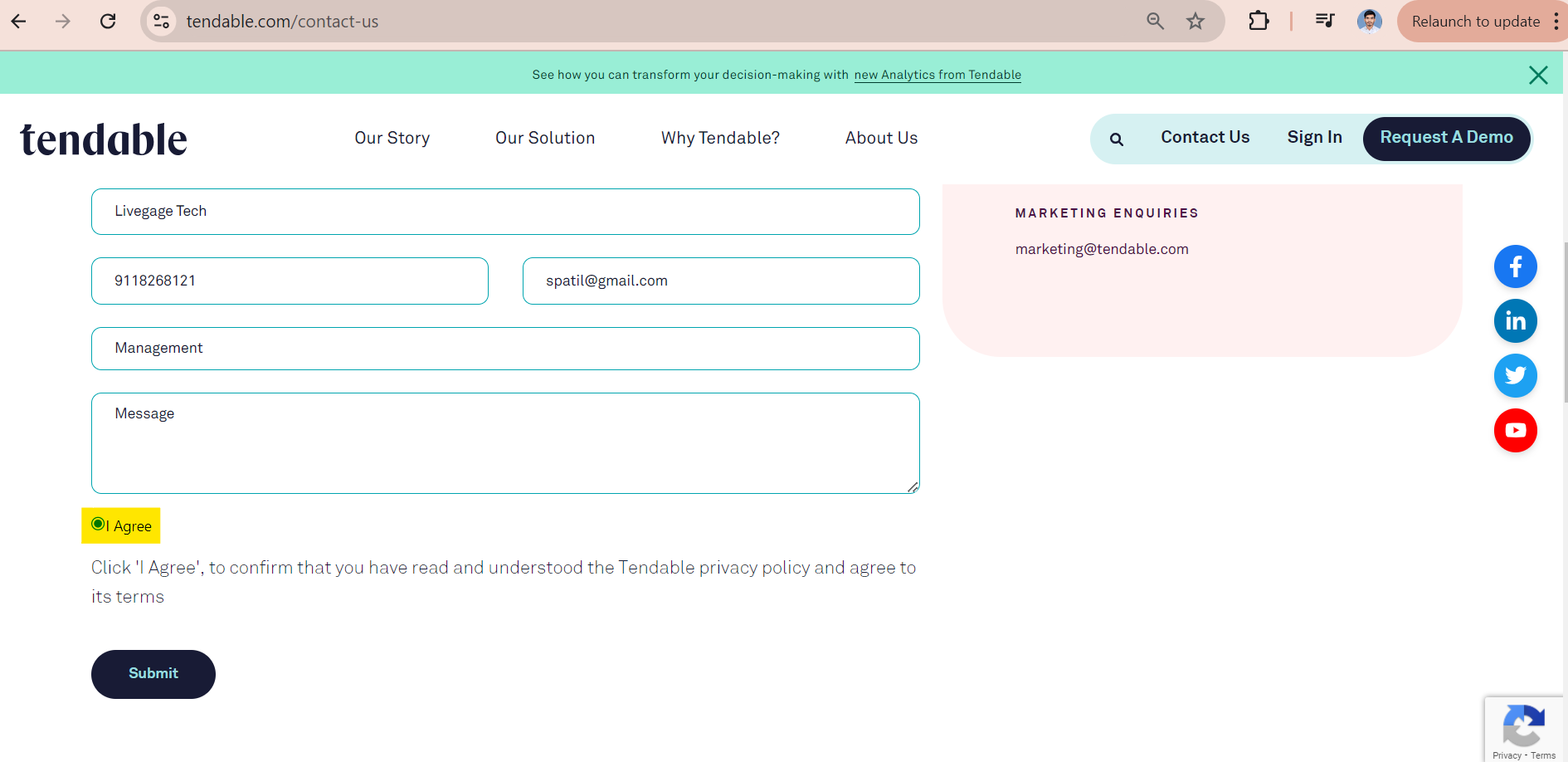
And user select job Role from dropdown



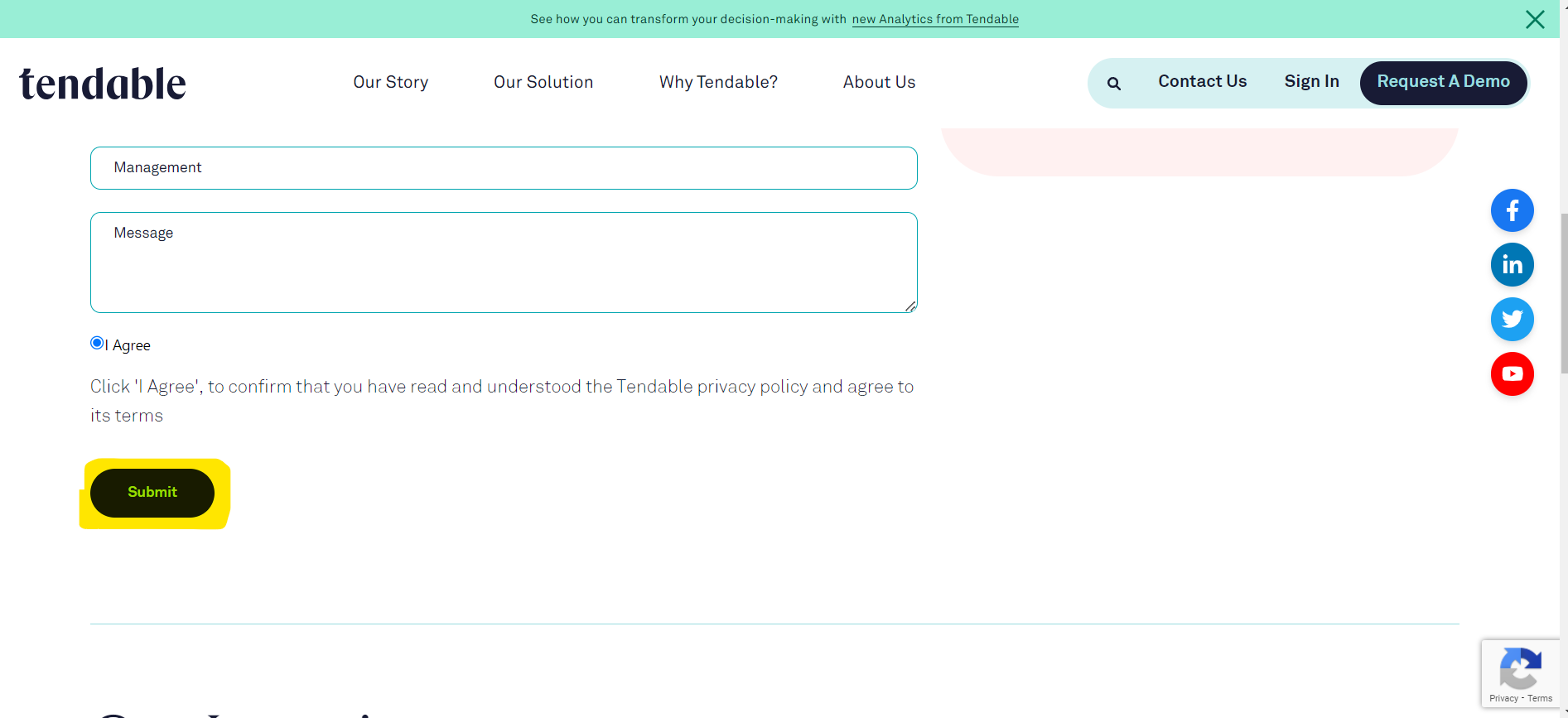
And user type message to textbox



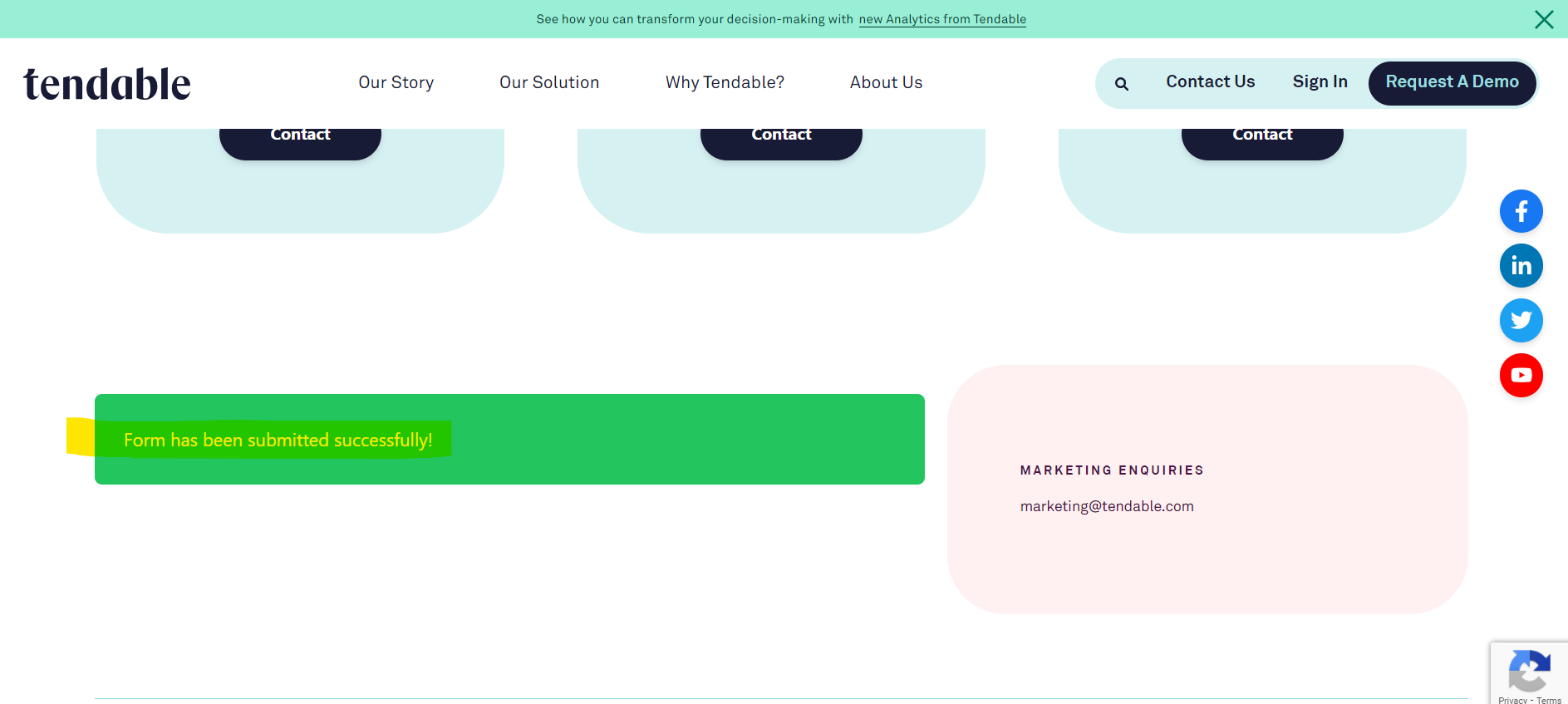
And user click on I Agree radio btn



And user click on Submit btn



And user verify error message displayed



Need to Raise a bug Per Execution because I don’t see Error message while submit Request

**Detail Framework BDD**

1. Developed BDD Framework with the help of TestNG/Junit on Java Platform

2. First we create maven project, at the time of creating maven project we have to mention group id

and artifact id. Group Id is nothing but the package name and artifact id is nothing but the Project

name. Once we create the maven project it will generate two main source packages as like

src/main/java and src/test/java and there is one most important XML file is pom.xml.

3.I have all dependencies like selenium maven , test NG Maven , Junit, Cucumber Test Ng Dependency etc.

4. Different types of source packages and classes added as per requirement .

5. inside the src/main/java we create different types of packages as like 1st like Base Layer package,

2nd Page Layer package, 3rd is config layer package, 4th is test data layer package and 5th is utils

layer package and inside the src/test/java we create 2 different packages as 1st is Step Definition

package and 2nd is test Runner package, and to maintain the all feature files we create Feature file

folder. There 3 main file presents in cucumber tool, 1st is Feature File and 2nd is Step Definition file and 3rd is Test Runner file.

6. For maintain all the feature file here we create feature folders. Inside the feature folder we can

create multiple feature file as per the project requirements wise but extension of each and every file

is dot feature. Inside the feature we use different types of gherkin keywords such as feature,

scenario, scenario outline, given, when, then, and, but,\* background and Examples keyword.

7. Inside the feature file first keyword must be Feature followed by Colon. And then we write short

description of feature, If you want to achieve normal scenario then we are use Scenario keyword

followed by Colon and then we write what is exact reequipments in short line, scenario is nothing

but the what to test or what is your exact requirements. And if you want to achieve data driven

testing in cucumber then we use Scenario Outline keyword followed by Colon and then we short

description of Scenario outline and at end of scenario steps we write examples keyword followed

by colon and there we maintain multiple set of data test data with the help of pipe symbol.

8. Then we are write scenario or scenario outline steps with the help of Gherkins keywords, there are

different types of gherkin keyword available to write the Scenario steps, as like Given keyword is

used to mentions preconditions, then When keyword are used to describe an event, or an action,

then Then keyword are used to describe an expected outcome, or result, then And keyword is used

to combine the 2 different steps, then But keyword are used declare multiple conditions and

asterisk (\*) global keyword which is used in place of any of the normal step keywords. We can use

background keyword followed by colon to mention precondition for each and every scenario in

feature file.

9. Then if create snippets with the help of Feature file or test runner class,

Here we create one package inside the src/test/java as like TestRunnerLayer Package inside this

package here we create test runner class. Inside test runner class here we use some class level

annotation as like @RunWith() annotation – here @RunWith() annotation tell to Junit to run the

test cases with the help of cucumber class. Then next we use @CucumberOptions() annotation: this

annotation tell multiple things as like where is feature file location, we declare feature file location

with the help of features keyword then we have write path of feature file location, where is Step

definition file location here we declare Step Definition file location with the help of glue keyword

or extraGlue keyword then we mention step definition Package location, then if you want to

checking mapping between feature file step and implemented steps in step Definition class then we

use dryRun=true attribute and we can generate different reports by using cucumber as like JSON,

XML , text and html report with the help of plugins keywords.

10. Again inside the source main java package here we declare multiple packages as like

inside the Base Layer package, we create Base Class and this Base class is super parent class of the

classes in framework, here we use inheritance concept. inside the Base Class we create static

method to mention basic property as like how to connect to the browser basically we connect to the

browser by using System.setProperty()method then here we mention up casting concept, as like

reference of WebDriver interface and Object of child class that is ChromeDriver class, basically by

using above up casting concept we can call all the methods from the WebDriver interface then we

mention implicit wait concept because implicit wait it applicable for all the element in webpage or web object. implicit wait is also known as global wait if we declare implicit wait as 30 sec and if

element or object find within the 2 sec then it will ignore remaining all sec and it also known as

dynamic wait, then we mention page load time out it is applicable for all the browser related

properties and then we write code for maximize the browser and also we mention delete All

Cookies code to clear all the cookies and then we get() method to open a specified URL and basically

we mention actual URL inside the config layer package inside the config layer package we create dot

properties file inside the properties file, we mention different environments URL, Username,

password , browser name and database details etc. and we read this properties file inside the Base

Class. if u want to work with properties file then first we have to create object of Properties class,

then we have to read the all the data from the properties file then we create object of

FileInputStream class by passing file path. then we have to load this file inside the class then we use

load() method from the Properties class and u want to get single property value from the

Properties file then we use getProperty()method by passing property key name.

10. In Page layer package we create container classes for each and every page in application. inside the

page layer package first we extends the base class, already i have already told that base class is

super parent class of all the classes in framework, inside the page layer package we create container

classes and inside the each and every classes there we create object repository with the help of

POM with Page Factory, POM means it is page object model it is an design pattern to design the

Object Repository. basically, we can create Object Repository with the help of 2 different ways as

like page object model with page factory and page object model without page factory. in my current

framework we create object repository with the help of POM with page factory at the time of

creating Object repository we use @FindBy() annotation. These @findBy annotation is used to find

the element or object in web page and once object is find then it will return the Web Element and

we store these all object as global as private variable here we are using encapsulation concept from

java OOPs and once we create object repository then we have to initialize the object then we have to

use Page Factory class and from Page Factory class we are use inti Elements method by using

WebDriver instance name and this keyword, this keyword is used to access the all global variable of

current class and inside the same class we declare the associated functionality methods of the all

object repository and as per the requirements wise we are calling utility layer package method in

Page layer package.

11. Utility Package added & inside this package inhave been added for utilsclass as per requirement for Handle Dropdown, Alerts Popups ,windows & Frames ,Action Class , Excel Reader , etc.

12. Inside the source test layer package there we have created step definition package and inside the

this package we stored all the generated snippets and inside this class we call associated

functionality methods from Page Layer package.