**Approach:**

* I preferred to use .NET framework in order to approach this task; I am good in C# and installing packages (.nuget) are very easy. My platform is 32-bit Dual-Core one (humble platform), so for all these reasons I apologize for not using suggested tools mentioned in task description ☺
* Extract features from working (valid) links, and neglect non-working ones.
* We make a Bag of Words for possible options for “Rahim” serie: {“Rahim”, “Raheem”, “Rahem”, “رحيم”}.
* If the link(URI) contains a word in the bag, then it is 100% Rahim-related one.
* If not, we go through the page content of link, and look for ‘meta’ tags in the page; ‘meta’ tags contains ‘content’ attribute, if the ‘content’ attribute value is in the above bag of words, then we consider this attribute as a related one.
  + Now we have 2 independent variables that control the classification, ‘meta’ tags and their child attribute ‘content.
  + If number of ‘content’ attributes is near from number of their parent ‘meta’ tags, then the page content (and moreover the link) is “Raheem” related.
* Now, as we have relatively small dataset, then we can use a classifier that is very accurate, easy to model (mathematically). I choosed Linear Regression.
* And as we have 2 independent variables, then we model a 2-variable Linear Regression:
  + <http://faculty.cas.usf.edu/mbrannick/regression/Reg2IV.html>
* There is a problem regarding Arabic words, we need a parser for Arabic sentences, so we are going to try “[Stanford.NLP.CoreNLP.CSharp](https://github.com/sergey-tihon/Stanford.NLP.NET/tree/master/samples/Stanford.NLP.CoreNLP.CSharp)” in order to do so.