**CHAPTER 4**

**SRS DOCUMENT**

A software requirements specification (SRS) is a document explaining how and what the software/system will do. It defines the features and functionality that the product requires to satisfy all stakeholders’ (business, users) needs.

**4.1 Introduction**

**4.1.1 PURPOSE**

The main purpose of preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system. This Document plays a vital role in the software development life cycle (SDLC), and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during the testing phase. Any changes made to the requirements in the future will have to go through a formal change approval process.

**4.1.2 SCOPE**

Phishing attacks in the future could take multiple forms and could evolve beyond recognition. For right now, your enterprise needs phishing protections such as email security to prevent most phishing attacks from ever reaching your employees in the first place.

**4.1.3 OBJECTIVE**

The main objective of this project is to make the awareness among people that how the legitimate website and phishing website are different which are looking same in the page layout. Generally, the home page of the website will be same and when we click on the links of the page then they redirect to us on phishing webpage. Also, those links are not stopped they repeatedly open another webpage which is a symbol of that webpage might be a phishing webpage.

**4.2 Specific Requirements**

**4.2.1 Hardware Requirements**

1. Processor – Intel Xeon E2630 v4 – 10 core processor, 2.2 GHz with Turboboost up to 3.1 GHz.
2. Motherboard – ASRock EPC612D8A.
3. RAM – 128 GB DDR4 2133 MHz
4. 2 TB Hard Disk (7200 RPM) + 512 GB SSD.
5. GPU – NVidia TitanX Pascal (12 GB VRAM)
6. Intel Heatsink to keep the temperature under control.

**4.2.2 Software Requirements**

1. **Anaconda Application**

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free. 50 Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

1. **Visual studio Application**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

1. **Python Libraries (NumPy, Pandas, TensorFlow, Keras, Seaborn, Matplotlib)**
2. **NumPy**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. A neat diagram of numpy is shown below in fig. 4.2.2.1.

A picture containing clipart

Description automatically generated

**Fig 4.2.2.1 Numpy**

1. **Pandas**

Pandas is a software library written for the Python programming language for data fifty-one manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at QAR Capital while he was a researcher there from 2007 to 2010. A neat diagram of machine learning is shown below in fig. 4.2.2.2.

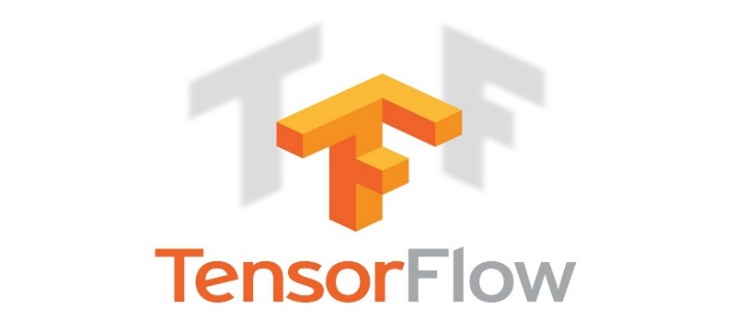
A picture containing text, clipart

Description automatically generated

**Fig 4.2.2.2 Pandas**

1. **TensorFlow**

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on the training and inference of deep neural networks. A neat diagram of tensorflow is shown below in fig. 4.2.2.3.



**Fig 4.2.2.3 Tensorflow**

1. **Keras**

Keras is an open-source software library that provides a python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. A neat diagram of keras is shown below in fig. 4.2.2.4.



**Fig 4.2.2.4 Keras**

1. **Seaborn**

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. A neat diagram of seaborn is shown below in fig. 4.2.2.5.



**Fig 4.2.2.5 Seaborn**

1. **Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of matplotlib. A neat diagram of matplotlibis shown below in fig. 4.2.2.6.



**Fig 4.2.2.6 Matplotlib**

1. **SKlearn**

Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects. Scikit-learn is focused on machine learning tools including mathematical, statistical, and general-purpose algorithms that form the basis for many machine learning technologies. Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering, and dimensionality reduction via a consistence interface in Python. A neat diagram of scikit learn is shown below in fig. 4.2.2.7.



**Fig 4.2.2.7 Scikit-Learn**