**Q1- Select a domain**

Domain: Cybersecurity

**Q2- Select a problem**

Problem: Developing a system for detecting and blocking phishing websites in real-time

**Q3 - Motivation behind selecting that problem (why?)**

For the problem of developing a system for detecting and blocking phishing websites in real-time is to protect internet users from the serious financial and personal consequences of falling victim to phishing scams.

Phishing is a type of online fraud where attackers impersonate legitimate organizations, in order to trick users into providing personal or sensitive information, such as login credentials or credit card details. These attackers can also use the phishing websites to install malware on the victim's computer, which can lead to identity theft, loss of personal information, and financial losses.

A system for detecting and blocking phishing websites in real-time can help protect internet users by automatically identifying and blocking phishing websites before they can be accessed by unsuspecting users. This can greatly reduce the likelihood that a user will fall victim to a phishing scam, which can have serious personal and financial consequences.

This system can be an important aspect of overall cybersecurity strategy by organizations and individuals, as it can prevent and minimize the potential damage caused by phishing attacks. Additionally, this system can also help companies and individuals to comply with the regulations for data protection and anti-phishing laws.

In summary, the motivation behind this problem is to create a safer online experience for internet users by automatically detecting and blocking phishing websites in real-time and thus protecting them from the financial and personal consequences of falling victim to phishing scams.

**Q4- who will be user once you create that product**

The users of this product would be internet users who could be at risk of encountering a phishing website.

**Q5- Divide your problem into modules**

Modules for this problem could include:

For the problem of detecting and blocking phishing websites in real-time, involves breaking the problem down into several different modules. These modules are designed to work together to achieve the overall goal of identifying and blocking phishing websites.

Website collection: This module is responsible for collecting new websites as they become accessible on the internet. This could involve using web crawlers or scraping tools to access new websites and gather information about them. The information collected in this module would include the website's URL, HTML content, images, and any other relevant data.

Website analysis: This module is responsible for analysing the websites collected in the previous module. The analysis can include both static and dynamic analysis. It can include analysing the website's structure, layout, source code, and any other relevant information such as URLs, images, and text. It can also include analysing the behaviour of the website while interacting with it.

Phishing detection: This module uses the information gathered in the previous two modules to identify potential phishing websites. This can involve training machine learning models using historical data and using those models to classify new websites. This module can also use a set of rules-based approaches to identify phishing websites based on certain characteristics like domain name, spelling, grammar, link structure and other identifiable patterns.

Result reporting: This module is responsible for presenting the results of the phishing detection to the users. This could involve displaying a list of identified phishing websites, along with a phishing score indicating the likelihood that a website is a phishing website. Additionally, this module could also provide detailed information on the decision taken by the system and the metrics on the system performance.

It's worth noting that these modules are not mutually exclusive, they might have overlapped functionality and cooperation. And the implementation of these modules may have a more complex structure than what I've described above, depending on the actual requirements of the problem and the use case.