**Introduction –**

FoxGoatOmega (FGO) is a simple software project that has been developed to automate complex equation solving for scientific and mathematical applications that deal with higher degree polynomials, trigonometric, exponential and logarithmic functions. It is a simplified version of mathematical and scientific tools like Wolfram Alpha that has been specifically tailored to solve various kinds of equations that cannot be solved by hand and are computationally intensive.

The application consists of a Graphical User Interface (GUI) that provides a simple, consolidated and user-friendly platform to execute various tasks such as solving and graphing. It also consists of a computerized database to store relevant data about equations for future reference and utilization.

Entering equations in a graphical environment and being able

to plot, store and retrieve them is possible with this application, which is important for scientific analysis. Additionally, the application supports multiple users who can log in freely one at a time and use the various features available till they log out. There is great connectivity between different features that enables the user to execute tasks quicker and more conveniently. Guidelines are available for help and warnings are provided to the user to avoid improper usage or exploitation of the features.

The coding and design of the user interface as well as the equation solving/plotting engine is done in Python IDLE 3.7 using various in-built as well as external libraries like GuiZero, SciPy, Matplotlib, Numpy and Pandas .The application is interfaced with MySQL 8.0 to allow for permanent, back-end storage and on-demand retrieval of information as per user requirement.

**Objective and Scope**

The main objective of this software project is to demonstrate the skillful usage of programming and data handling by creating an application with the capability to solve equations that are not possible to solve on paper and need sophisticated methods and analytical methods for solving. Furthermore, back-end storage of results obtained after solving; done by interfacing Python with MySQL displays the usage of concepts of Database Management and

Connectivity.

The mathematical tool, FoxGoatOmega (FGO) is expected to have the following functionality-

1. To provide a simple, consolidated and user-friendly graphical user interface (GUI) for carrying out required tasks with the features available.
2. To allow for multiple users to utilize the power of this application in a sequential manner without compromising the activity and efficiency of the application.
3. To solve polynomial, exponential, trigonometric and logarithmic equations that are entered by the user and provide a platform to analyze results.
4. To provide easier interpretation of the equations or expressions entered by the user by graphing as per user requirement.
5. To store relevant information regarding the equations entered for future reference or utilization.
6. To provide an efficient front-end environment to interact with the database in order to extract necessary data about the equations based on different parameters.
7. To prevent malpractice or misuse of features by providing warnings messages as well as easily comprehensible instructions to the user.

The advantages that are expected to be visible in this application at this stage include-

1. Less training time for users due to simple interface along with guidelines and warning messages.
2. High efficiency in solving computationally intensive equations due to dedicated equation-solving engine.
3. Easy interpretation of results through graphing capability and related information.
4. Long-term storage and usability of results through permanent back-end storage.

The application is far from being perfect as it is just a model to demonstrate proof of concept. However it can be improved over time with more enhanced features. At this stage, certain limitations that can be observed in this application are-

1. There is a trade-off between speed and size of solution set. If there are many solutions that belong to the solution set, there could be more delay than expected.
2. If the values that belong to the solution set are different from each other by less than 0.001 then the accuracy and speed of the equation solver is impacted negatively.
3. The application can freeze if multiple windows with different tasks are opened simultaneously.
4. Certain equations with trivial solutions (called “contact equations like (x-2)\*\*(0.5) ”) cannot be solved due to fundamental domain and range constraints that cannot be overcome by the equation-solving engine at this stage.

**System Implementation-**

**Hardware used-**

**Software used-**

1) Operating system – macOS Mojave 10.14.6