NUMERICAL ANALYSIS (CPT_S 530)

PROJECT REPORT

Function Analysis

by

Parikshit Panwar

WSU ID: 01172 – 5522

MS Computer Science (Non - Thesis) Fall 2020.

Submitted to

Prof. V.S. Manoranjan

Washington State University, Pullman

Fall Semester 2020

TABLE OF CONTENT

Contents

1. Introduction

As in the earlier times, operations such as finding derivative and integral of a function was done by hand. Hence, performing the derivative and integration of complex and long functions used to consume a lot of time. Also, calculating the root of equation or performing prediction based on the data set provided was a tedious task since, interpreting large amount of data consumed huge of amount time and thereby making work less efficient.

The project is to build a software that is able to perform various analysis of a function. The software will be able to perform operations such as finding the expression for rate of change of the function and calculating its rate of change value at a provided value, calculate the integral expression of the function and hence calculating the integral value of the function by accepting limits. The software also has the potential to calculate the root of a given function using the *Bisection method.* The software can also obtain a linear fit and an exponential fit based on the user's choice; therefore, is also able to predict values for any provided point. The software enables user to select his/her choice of operations from the start menu from the given set of options. The user has the option for performing derivative analysis, integration, find root, and predicting outcomes using linear fit and exponential fit.

2. Statement of the Problem

Performing complex mathematical operations such as finding the derivative or integration of various complex and long functions consumes a lot of time. Also, performing predictions based on the data set provided is also a tedious task since, interpreting large amount of data consumed huge of amount time and thereby making work less efficient. The amount of time that is being wasted in performing long and complex calculations can be eliminated or reduced by introducing automation. A software can be built which has the potential to perform various mathematical operations, as a result improving efficiency.

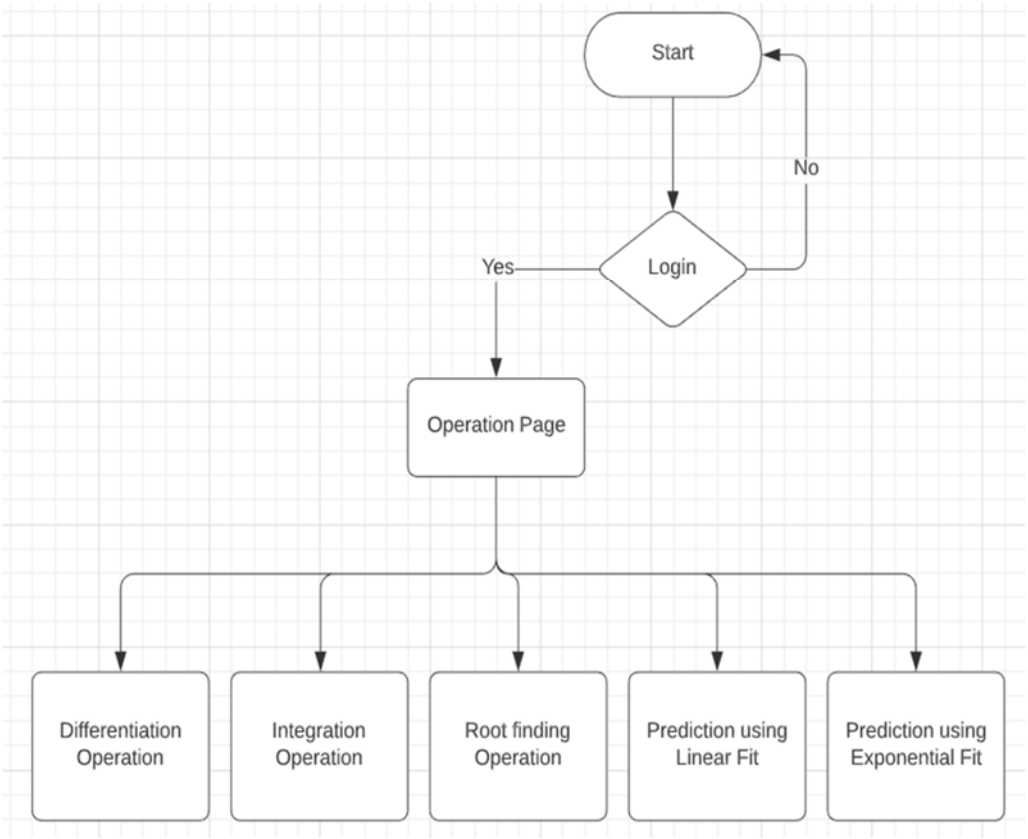
3. Objectives

Objective is to create a software that will perform the following major functions: -

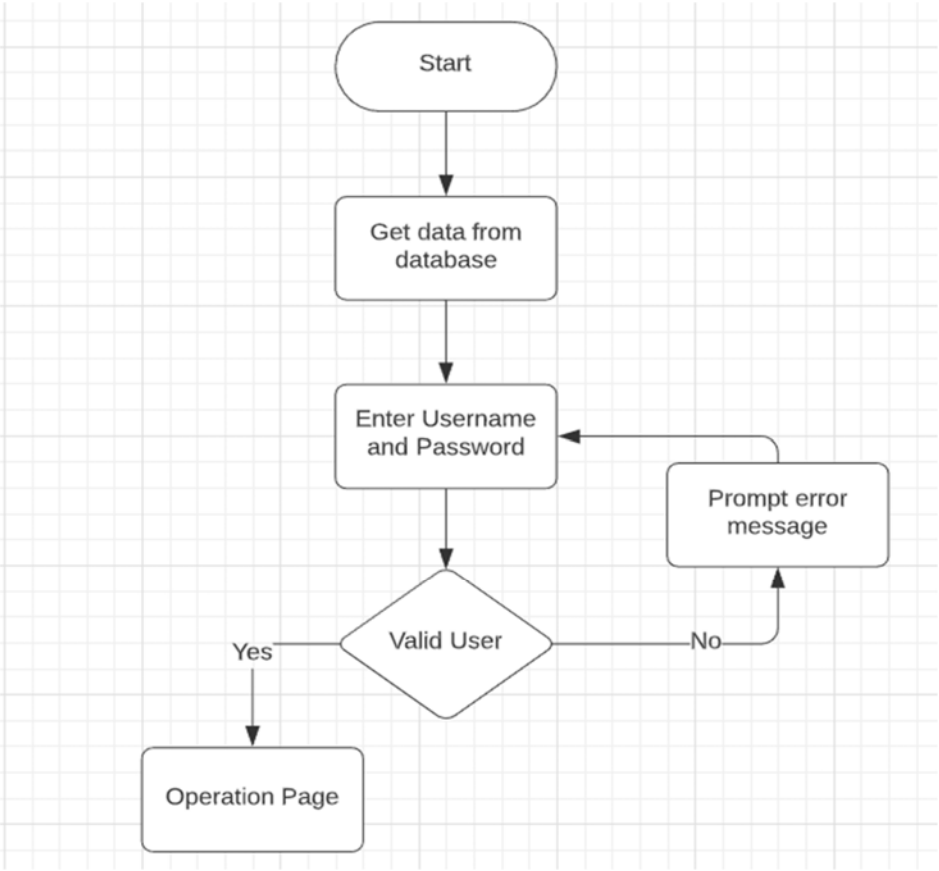
1. Derive a derivative expression of a given function and perform analysis.
2. Derive an integral expression of a given function and perform analysis.
3. Calculate the root of the function provided.
4. Perform Linear fit on the provided dataset.
5. Perform Exponential fit on the provided dataset.

4. Relevant Diagrams:

4.1 Project Flow Diagram:



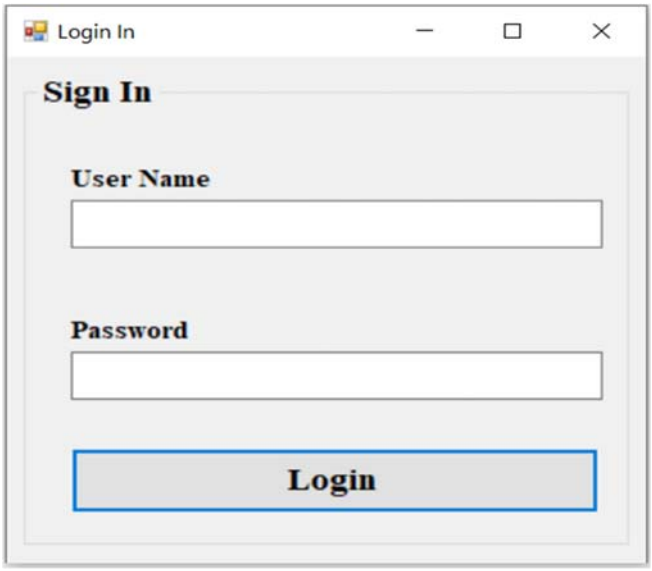
4.2 Login Flow Diagram:



5. Methodology

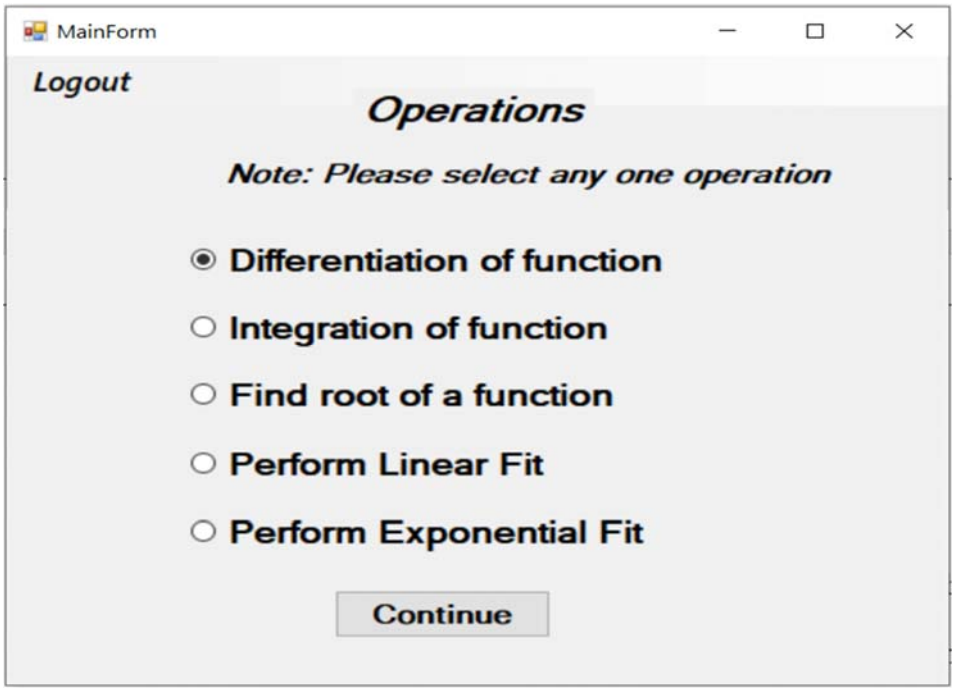
To accomplish the project, following work must be carried out:

1. Login Page: - This function provides a security to the application. Only an authorized user can gain access to using this software. The login module contains requires credentials, **Username** and **Password**. A separate text file is maintained which contains the official username and password of all authorized users.

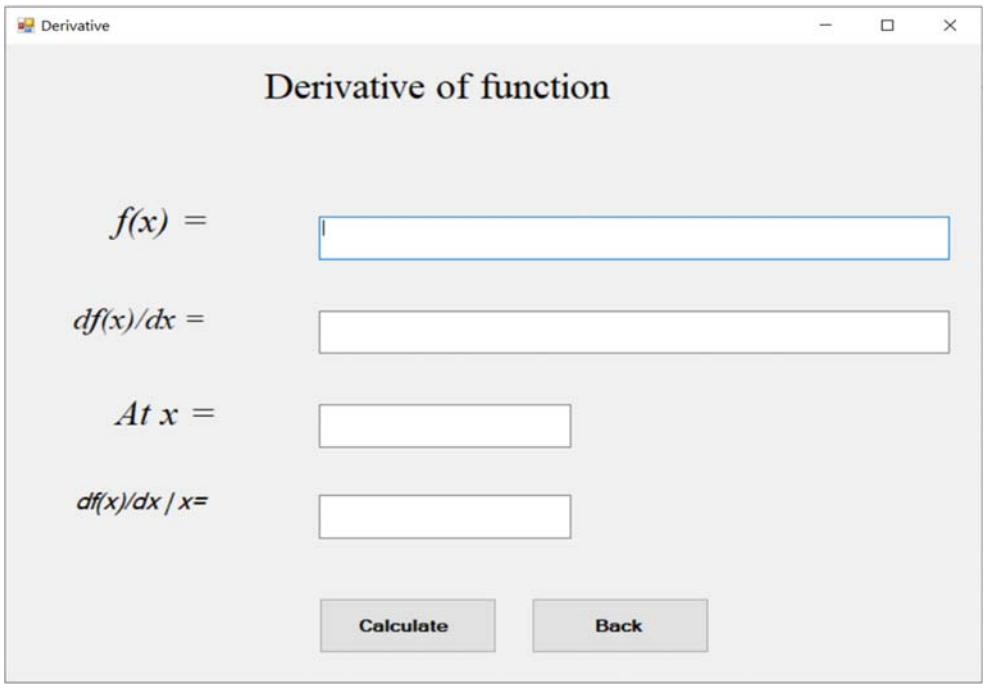


After providing the required credentials, press login, the software will then scan the text file and verify whether the credentials provided match the credentials in the text file. If yes, the menu page will pop up. If no, an error prompt will appear and the user will have to repeat the process.

2. Menu Page: - This functionality allows the user to explore different functionalities of the software. The user can simply select any desired option and click on the **"Continue"** button to proceed with the desired option.

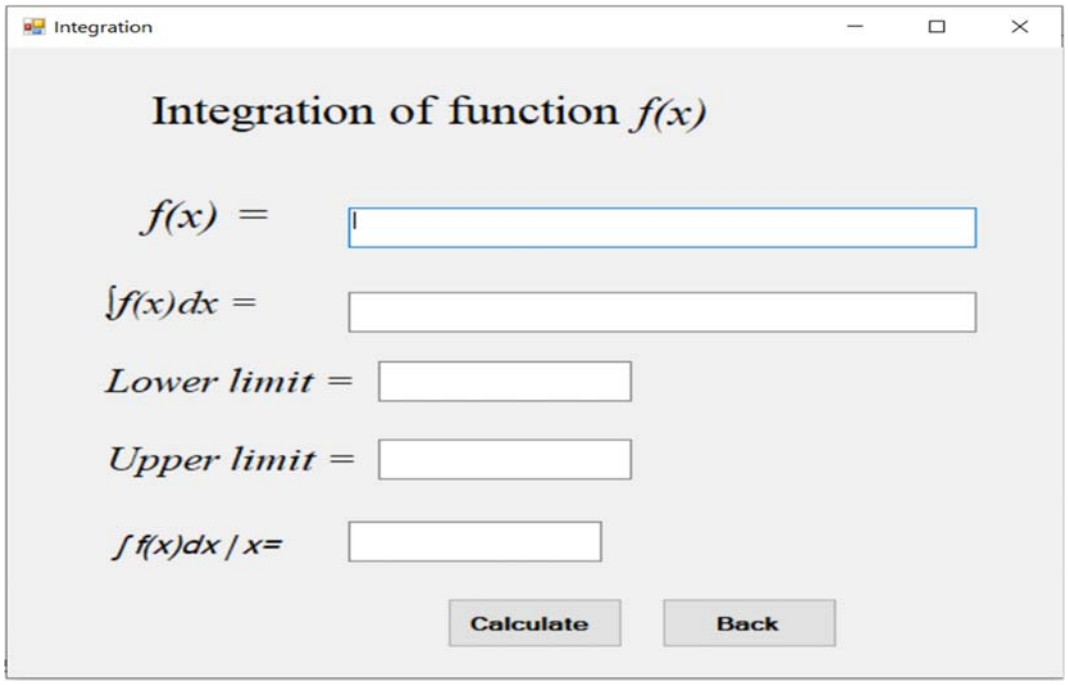


3. Differentiation of function: - This functionality allows the user to derive a derivative expression of a polynomial function and hence calculate its derivative at a desired point.



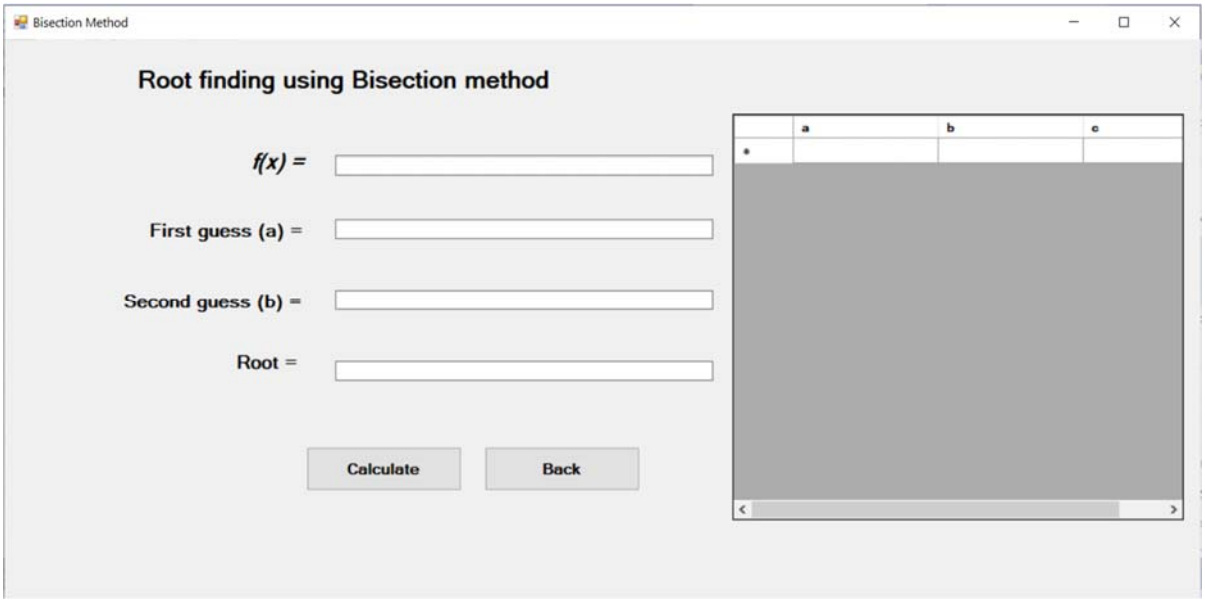
The ***"f(x)"*** textbox takes the polynomial function with variable ***"x".*** The derivative expression of ***"f(x)"*** is obtained in ***"df(x)/dx"*** textbox after clicking on the **"Calculate"** button. The ***"At x"*** textbox takes the value of **"*x*"** at which the user wants to calculate the derivative of the function ***"f(x)",*** its result is obtained in the ***"df(x)/dx | x"*** textbox.

4. Integration of function: - This functionality allows the user to derive an integral expression of a polynomial function and hence calculate its integral within its provided lower and upper limit.



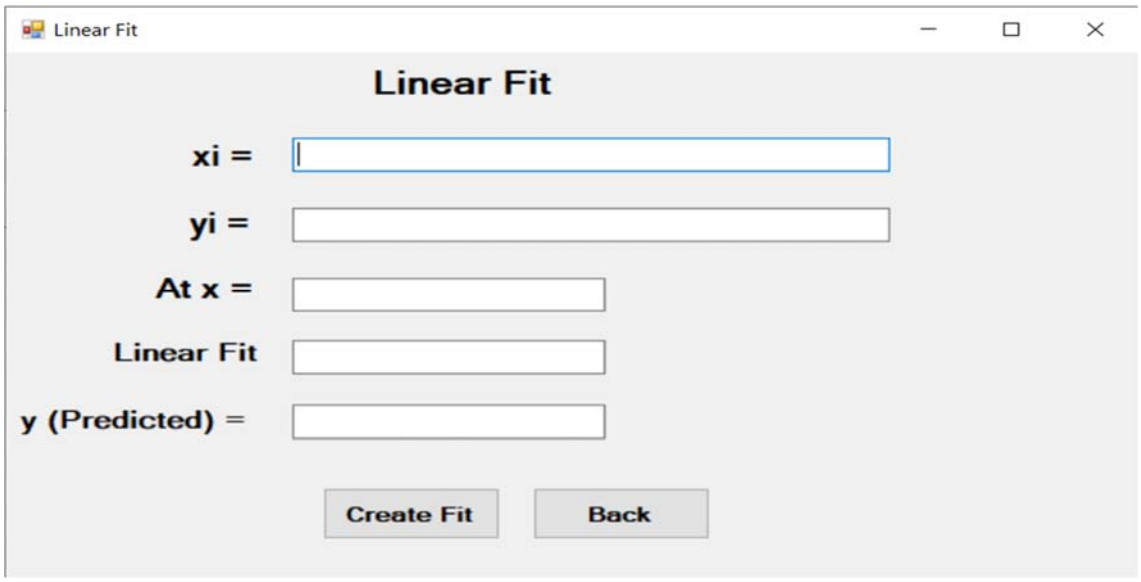
The ***"f(x)"*** textbox takes the polynomial function with variable **"x".** The integral expression of ***"f(x)"*** is obtained in **"$\int f(x)dx$"** textbox. The lower and upper limits need to be specified in the textbox provided. After clicking on the **"Calculate"** button, **"$\int f(x)dx \mid x$"** textbox shall provide the calculated integral value of the function ***"f(x)"*** within the desired limit.

5. Root of function: - This functionality allows the user to find the root of the using *Bisection method*.



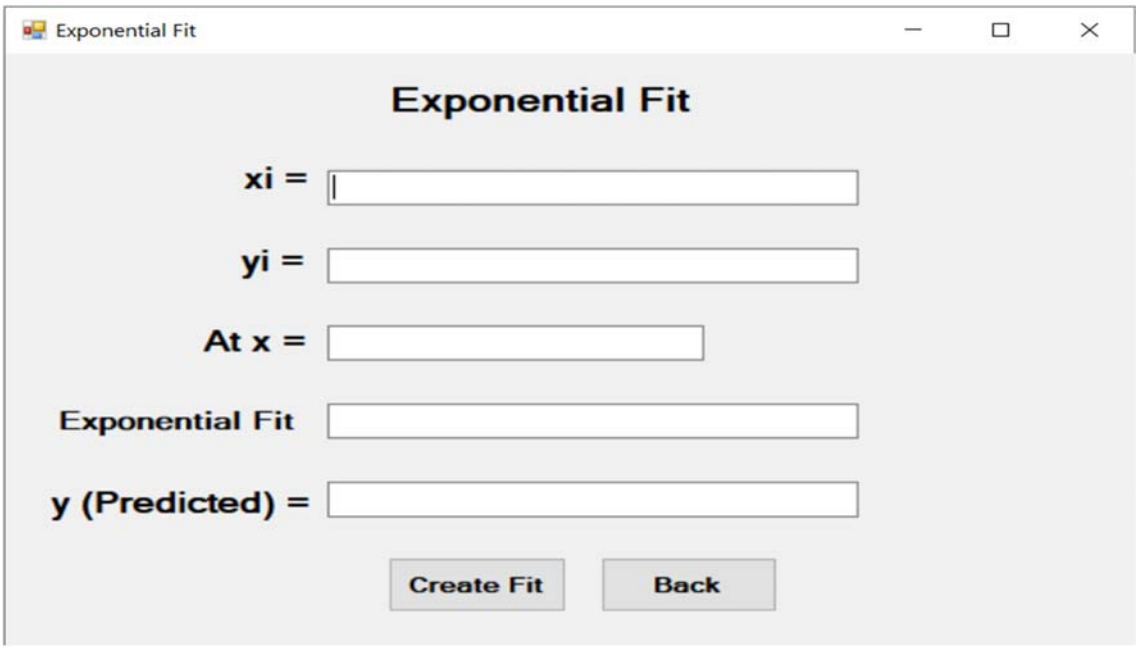
The ***"f(x)"*** textbox takes the polynomial function with variable **"x". "First guess"** and the **"Second guess"** need to be specified, click the **"Calculate"** button to perform the calculations. If the guess provided does not satisfy the condition of the *Bisection method*, an error prompt shall be displayed and the user will have to provide satisfactory guess values to proceed. After, the guess is satisfied. The calculated root value of the function ***"f(x)"*** will be displayed in the **"Root"** textbox. The **"datagrid"** will display the value of each variable that is used to calculate the root at every iteration.

6. Linear Fit: - This functionality allows the user to predict the value using *Linear Fit method* when a dataset is provided.



The ***"xi"*** textbox takes the x – coordinate of the dataset with each value separated by comma**.** The ***"yi"*** textbox takes the y – coordinate of the dataset with each value separated by comma**.** The ***"At x"*** textbox takes the value of ***"x***" at which the user wants to predict the value. The **"Linear fit"** text box displays the new linear fit equation that is formed. After clicking on the **"Create Fit"** button, the calculated predicted value - using *Linear fit method* - will be displayed in the **"y (Predicted)"** textbox.

7. Exponential Fit: - This functionality allows the user to predict the value using *Exponential Fit method* when a dataset is provided.



The ***"xi"*** textbox takes the x – coordinate of the dataset with each value separated by comma**.** The ***"yi"*** textbox takes the y – coordinate of the dataset with each value separated by comma**.** The ***"At x"*** textbox takes the value of ***"x***" at which the user wants to predict the value. The **"Exponential fit"** text box displays the new exponential fit equation that is formed After clicking on the **"Create Fit"** button, the calculated predicted value - using *Exponential fit method* - will be displayed in the **"y (Predicted)"** textbox.

6. Challenges Faced

1. **Unauthorized Login**: As the authorized credentials are stored in a text file, while verifying the credentials, the software scans each line in the text file, to match the credentials specified in the textbox. This caused a problem of a user logging in his/her username but having another authorized password. For example, if a **User1** having the credentials, **"username = testusername1"** and **"password = testpassword1"** (which are authorized credentials and are specified in the text file) logs in with **"username = testusername1"** and **"password = testpassword2"**, (**"testpassword2"** being the password for **User2**) would gain access and would be the case of unauthorized login.

   Solution: Every time a new user is authorized, his/her credentials are stored in format **"username = testusernameX"** and in the line below, **"password = testpasswordX"** (for any new user x). The software is modified such that every time the system runs the credential match, the username and password for a user must exist in pair. If the pair is valid, the use can gain access. If not, then an error will prompt.

2. **Parsing mathematical expressions**: To parse a math expression, the expression is extracted from the textbox and segmented into terms depending on the separation signs (i.e., + or -). Each term then is converted to its derivative expression. After all the terms are converted to their respective derivative or integral terms, they are joined together to form a complete expression. Parsing this, expression is tedious as the expression is again in a string format and values of variable x cannot be calculated unless the string can be converted to an operatable function.

   Solution: The code was then optimized by using a library **"MathParser"**. This library has the ability to parse any string mathematical expression. The functions in this library provide a way to convert string expressions into operatable functions; therefore, making it possible to parse the string and calculate the required value.

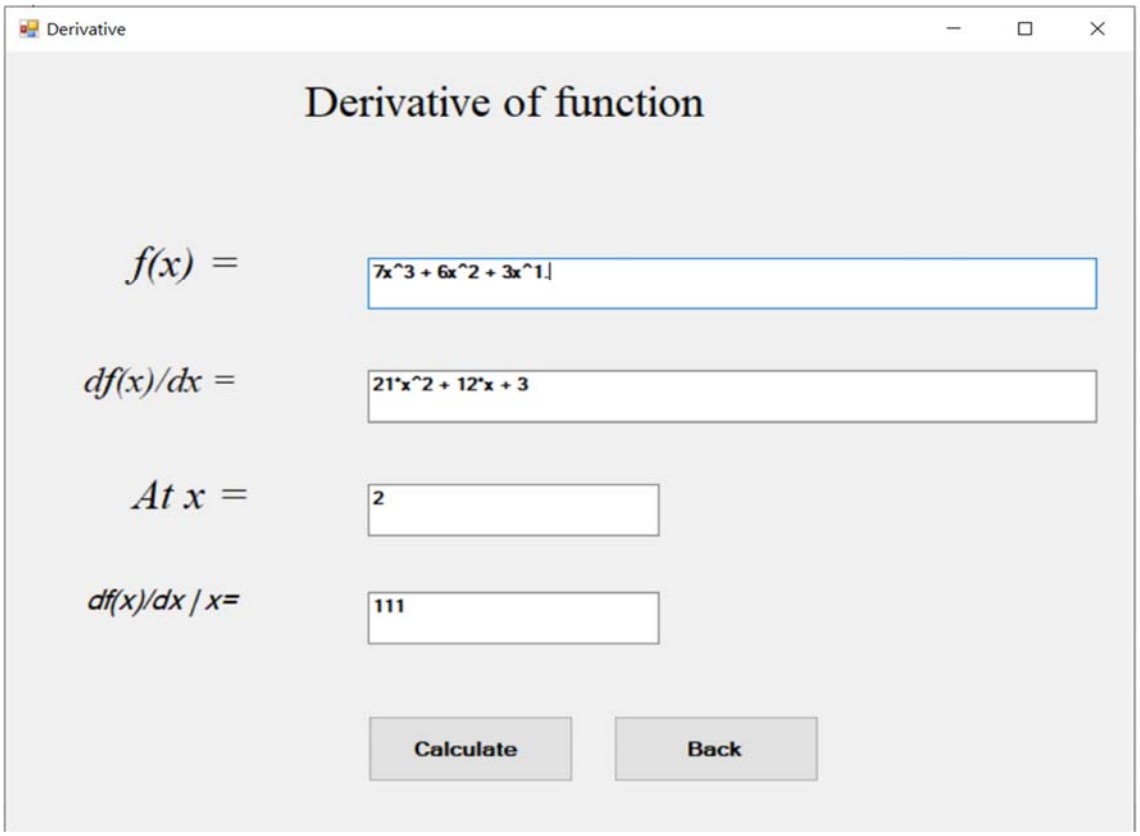
7. Limitations

1. **.NET Framework:** The software application is built on .NET framework; therefore, it requires a .NET framework to run the executable file.

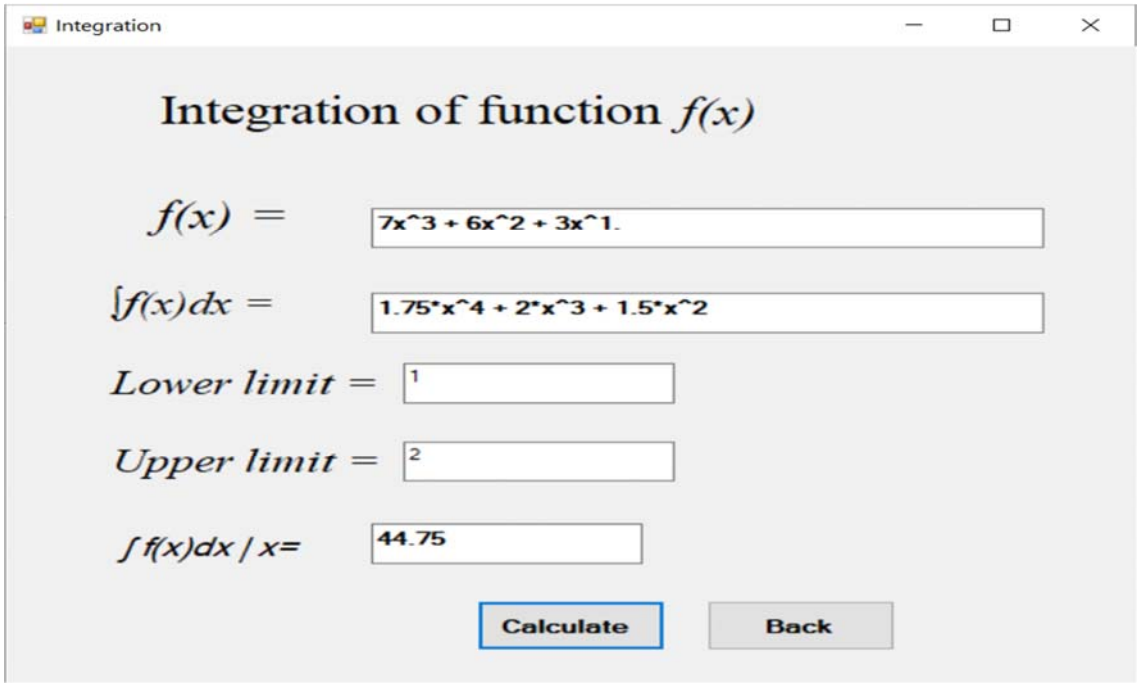
2. **Input function:** The software is capable of finding the derivative or integral of polynomial functions only.

8. Results

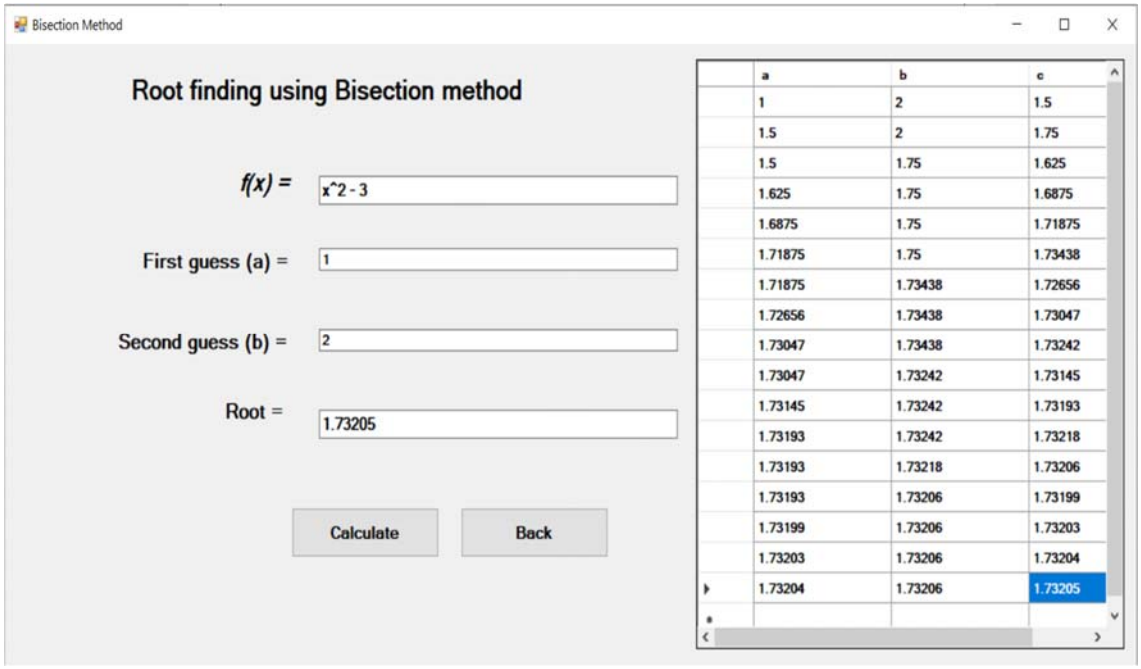
1. Differentiation of function:



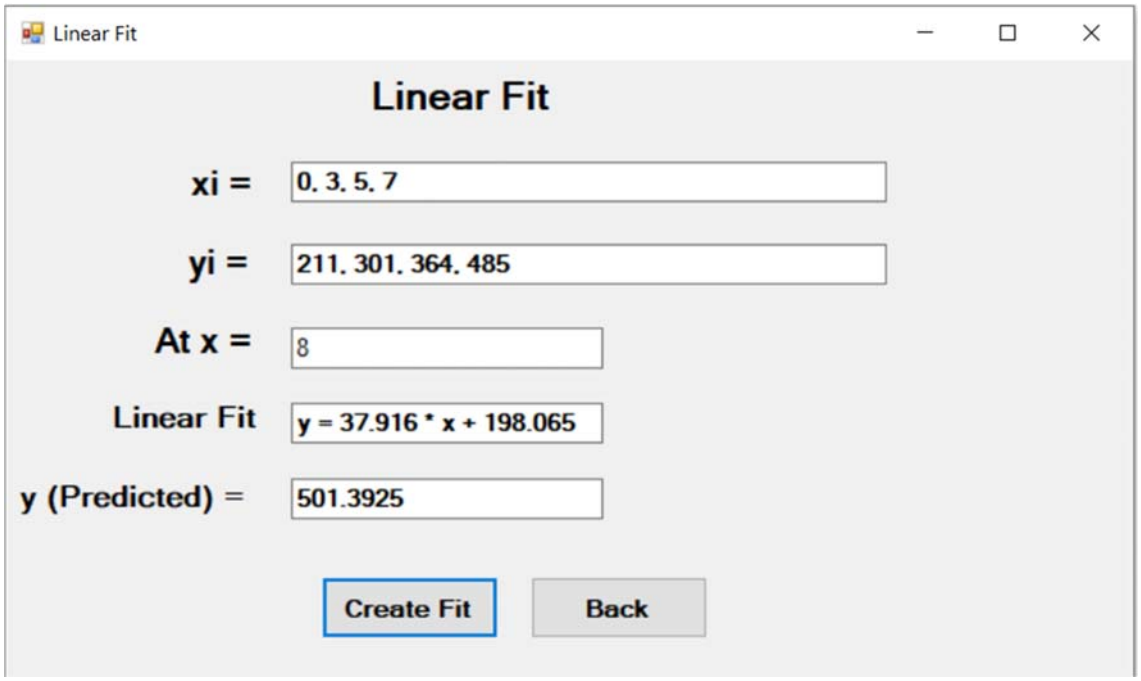
2. Integration of function:



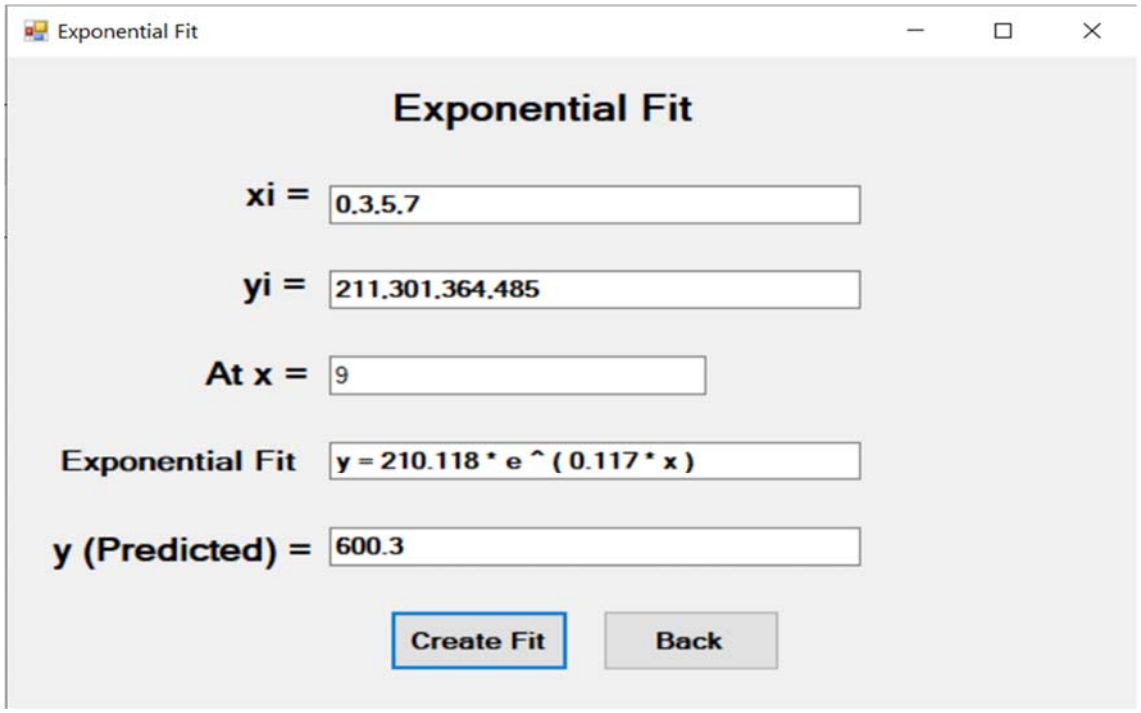
3. Root of function:



4. Linear Fit:



5. Exponential Fit:



9. Conclusion

The software works effectively to calculate the derivative and integral expression and values, finding root of the function using *Bisection method,* perform prediction using linear fit and exponential fit with respect to the provided dataset. The software provides a facility to the user to select between the provided options of mathematical operations. The software is developed in .NET framework and requires knowledge of C#.NET, different namespaces are used for data handling, file Handling and controls of Visual Studio are used for creating the frontend of the software.

10. References

Web Links:

1. BestProg. (n.d.). *C#. Example of creating menu. Control MenuStrip.* https://www.bestprog.net/en/2016/01/28/008-an-example-of-creating-a-menu-in-c-control-menustrip/.

2. C-sharpcorner. (n.d.). *File Handling in C#.* https://www.c-sharpcorner.com/UploadFile/puranindia/file-handling-in-C-Sharp/.

3. C-sharpcorner. (n.d.). *Insert, Update and Delete Records in a C# DataGridView.* https://www.c-sharpcorner.com/UploadFile/1e050f/insert-update-and-delete-record-in-datagridview-C-Sharp/.

4. Tutorials Teacher. (n.d.). *C#Files&Directories.* https://www.tutorialsteacher.com/csharp/csharp-file.

5. CSharp. (n.d.). *Get Files from Directory [C#].* https://www.csharp-examples.net/get-files-from-directory/.

6. Geeks for Geeks. (n.d.). *How to extract filename from a given path in C#.* https://www.geeksforgeeks.org/how-to-extract-filename-from-a-given-path-in-c-sharp/.

7. MXParser(n.d.). *mXparser – tutorial*. http://mathparser.org/mxparser-tutorial/