**Detailed Proposal**

**Plan:**

What will I do:

* Make an equation solver
* User inputs valid text which corresponds to an equation or system of equations
* Invalid input will be called out
* Program then solves said equation
* Types of equations it will solve linear, quadratic, cubic, systems of equations, inequalities, rational and differential equations/integral equations (potentially)
* Make an aesthetic UI
* Show working out

How will I do it:

* Backend will be done in python/flask to handle the problems and produce an output
* Frontend will be done in html, css and js
* Html to display content of my web app
* Css to make the content look nice
* Js to make the app more interactive
* Jinja to pass data from flask to html
* When user inputs problem I will determine what type of equation it is and use the appropriate method, using sympy
* Linear/ system of linear equations – Gaussian elimination
* General purpose calculator
* Quadratics – quadratic equation
* Polynomials – newton Raphson
* Inequalities – context based
* Differential equations (first order)

What will it look like:

* Heading for title
* Text box for problems lower down
* Calculator like GUI for inputting problems
* text box mid-section for displaying answers, initially set to instructions for the user

**Equation Solver Proposal**

**Written By Rishi Rattanpal – 201634390**

**Statement of ethical compliance:**

I state I will follow the university of liverpool’s ethical guidance as detailed here <https://student.csc.liv.ac.uk/internal/modules/comp390/2024-25/ethics.php>

My project will be in the following data/human participant category, A2

**Project Description:**

The aim of my project is to make an equation solver, this will be a web application that will assist users ranging from students to professionals in solving various types of equations/system of equations.

A user will be able to input their problem via text in which my program will then be able to analyse and then solve it outputting the correct answer with working out, given the input is valid.

To enhance the experience and make it simpler my program will also have a simple, yet aesthetic UI styled like a calculator allowing for easier input of numbers and symbols alike in a familiar manner that mathematicians are used to

**Aims and requirements:**

My project will be able to do the following:

* Take in input formatted as equations
* Correctly solve and output the solution
* Have an aesthetic display to do the above two points

My project will have the following essential requirements:

* Allow user to input their equations
* Process problem and decide what type of equation it is
* Apply correct method to solve depending on the type of equation it is
* Ability to solve the following types of equations: linear, system of linear equations, quadratics, polynomials, and inequalities
* Give a clean and aesthetic UI for input/output
* Error handling for incorrect input

My project will have the following desirable requirements:

* Ability to identify and differentiate equations
* Ability to identify and integrate equations
* Matrix maths
* Customisable UI
* Store history solved equations
* Output step by step guide on how to solve
* Draw graph for inputted equation

**Key literature and background reading:**

This section I will explain the theoretical concepts of equation solving as well as the practical applications it can be used for

After reading [1] I have learnt how to solve systems of equations using Gaussian Elimination. The process is as follows, “*extract the coefficients of the variables and the constants, and these become the entries of the matrix. We use a vertical line to separate the coefficient entries from the constants, essentially replacing the equal signs.*” essentially writing your equations as an augmented matrix. This is good due to matrixes being efficient and there being a lot of computational tools available

Then using row operations, we *“convert the matrix to row-echelon form”* where there are *“ones down the main diagonal from the upper left corner to the lower right corner, and zeros in every position below the main diagonal”*

Then using this form, we cab “*do back-substitution to find the solution”*.

After reading [2] I have a method for solving quadratic equations and that is putting them in the form *“ax²+bx+c=0, where a, b, and c are coefficients. Then, we plug these coefficients in the formula: (-b±√(b²-4ac))/(2a)”*

After reading [3] I have a method for solving higher degree which is the Newton-Rapson method. It is a recursive method that uses an initial guess and subtracts it from the result of the original function derivative of the original function. As show “*xn+1​=xn​−f′(xn​)/f(xn​)​.*and continuously feeding that result back until you get a consistent answer, or a max number of iterations have been made.

**Development and implementation summary:**

Development Environment

I will be using visual studio code to write my code as it is lightweight, efficient and I can utilise all the languages I am using inside of it. It also offers a lot of extensions. For my version control I will use github as I will be able to sync it with the university computers as well as my home desktop and laptop allowing me to be able to work on my project in multiple areas as well as revert to an older version if needed. For installing my packages in python, I will be using pip due to its vast range of packages and it having all the ones I need allowing these libraries to do some of the heavy lifting for me.

Implementation languages:

**Python** - I will be using python to handle the backend and core functionality of my project, this is because it is a high-level language where you can do a lot with little readable code, as well as this there are a vast number of libraries and frameworks, I would like to incorporate into my project available to python. These include SymPy which I will be using for symbolic computation, NumPy for numerical methods and matplotlib for plotting my graphs.

**Flask –** I will be using the flask web framework in addition to python to manage web requests and connect my frontend (python) to my backend (html, css, js) using routing and template generation.

**HTML** – I will be using html as part of the frontend to write out all the content of my web application and define the structure of it giving it a foundation to be built upon.

**CSS/Bootstrap** – I will be using CSS and more specifically the bootstrap framework as part of the frontend to style my app making the content look aesthetic and provide a user-friendly UI. Bootstrap has a wide range of prebuilt features already meaning I won’t have to design them myself but rather use them and customize them to my liking

**JavaScript** – I will be using JS as part of my frontend to add interactivity to my app making it more dynamic and engaging for users

**Jinja** – I will be using jinja for template generation to generate content from the backend and display it on the frontend

Project implementation

My project will be split into two main components, the frontend and the backend. I will start on the backend first and once that has some functionality, I will implement a rudimentary front end to go with it. Once my backend has more features, I will flesh out the front and make it look more aesthetic. I will also conduct various testing during production to keep an eye out for further improvements/bugs.

Backend

**User input** – My first step will be allowing the user to input their problem; this will initially be done as a string. Then I will standardise input by removing all spaces, converting input to a form my program can understand e.g. ‘^’ = ‘\*\*’ and then using the *‘sympify()’* function so my program can either return an error and inform the user or will be a valid equation in which my program can check a dictionary with different types of equations and their classifying property and determine what type of equation it is.

**Equation solving –** Once the equation has been classified, I will apply the apply the corresponding method to it this may include gaussian elimination, quadratic formula newton rhapson or some predefined methods from libraries. Each method is different and therefore I will store their specific instance of working out to be displayed in the future, along with its corresponding answer my program works out.Will store iterations in a string to be later displayed as the working out.

**Graph plotting** – Using matplotlibs .plot() method I will be able to plot a graph of my equation, ready to be displayed on the front end.

**Connection to the frontend** – My app different route, ‘/’ will be the homepage and render the template *‘index.html’* where all users start off at and where you can input your equation. I will be using POST to send receive my data from the front end which will then be redirected to my *‘/solve’* route where it will be processed and all components to be displayed will be computed.

Frontend

**Index.html** – homepage where I will allow users to submit their problems as well as displaying my interface styled via bootstrap. I will have a calculator display for users to input their problems in and using AJAX dynamically update the page to show the answer along with the steps and graph

**History.html –** Page that allows a user to see all their past inputted problems with their associated solution. Will have a prop to also delete history

**Calculator** – User friendly calculator where each button will append to the input

**Connecting to the backend** – In my html there will be jinja code which will serve as placeholders for the data that I have send via POST

Testing Phase

I will use a variety of testing methods such as unit testing, integration testing, regression and user acceptance throughout my project.

**Data sources**

My project uses no data sources, the only data this project will use is that which is generated via user input.

**Testing & Evaluation**

To test my final product I will use the testing methods detailed in the testing phase.

**Unit testing** – Used to test each function and ensure it works as expected e.g. IdentifyEquation(), solveQuadratic() etc. To test my equation solving methods work I will handwrite and solve them using mathematics myself, check against other implementations of equation solvers.

**Integration testing** – Used to test my connection to the backend to the frontend, when sending data will look out for HTTP codes to verify data is being sent correctly.

**Regression testing** – After significant updates are made, I will input answers I have already verified as correct as well as test already working features to make sure they are still intact

**User Acceptance Testing** – I will gather users with their own problems and test it against my software, the users will range from students to professionals

**Project Ethics & Human Participants:**

I will have a set of voluntary beta testers where I will hand them an early model of my software and ask them to input their own problems which I can use to identify bugs/errors. The only data I require from them will be the problems they inputted and the incorrect answer meaning it will be completely anonymous, but I can still use their data ethically. The data will be stored securely on a password protected computer and used for debugging purposes only and participants will be well informed of the purpose of their participation and how their data will be used.

**BCS Project Criteria:**

*COMP101* was the module that taught me python which is the foundation this project stands on and using this I will be able to implement flask. As well as that there were lots of other modules that taught me programming and problem-solving skills such as *COMP281, COMP122, COMP228* where I learnt C, java and swift. These modules showed me once you know one language it is a lot easier to learn another which I will be doing due to my limited experience in *JavaScript, sympy, numpy and matplotlib*.

This project demonstrates innovation as it will be a format for users to input, solve and display their problems simultaneously. With dynamic updates via AJAX and an aesthetic design via bootstrap the app will be unique and effective for solving equations.

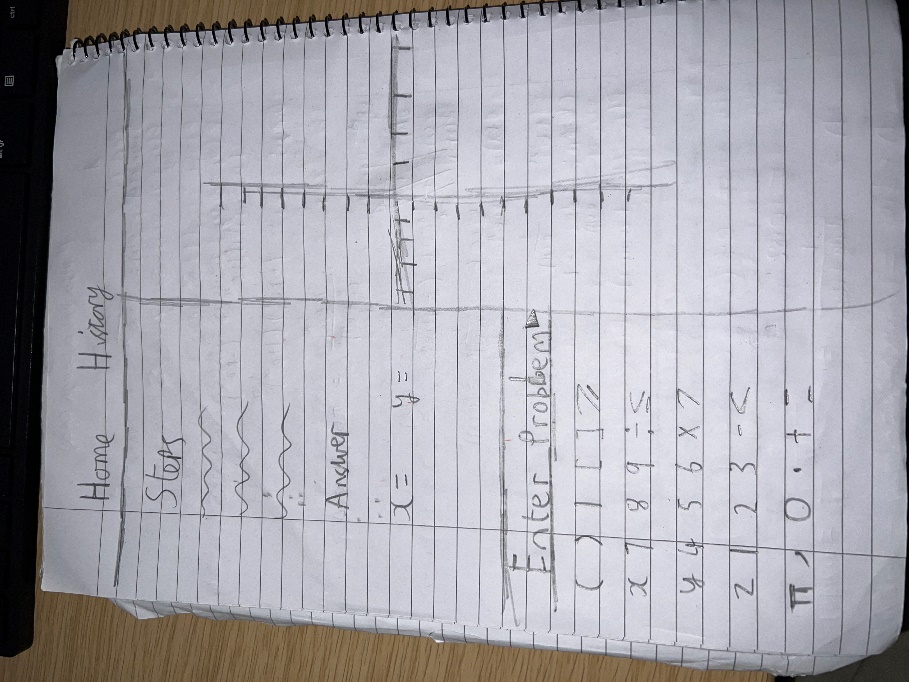
I will be using various languages, frameworks and libraries to achieve certain functionality and integrate them together, as mentioned in the development and implementation summary to synthesize information and ideas. I will use a variety of testing methods/practices as mentioned in the *‘testing & evaluation’* section to ensure the quality of my solution.

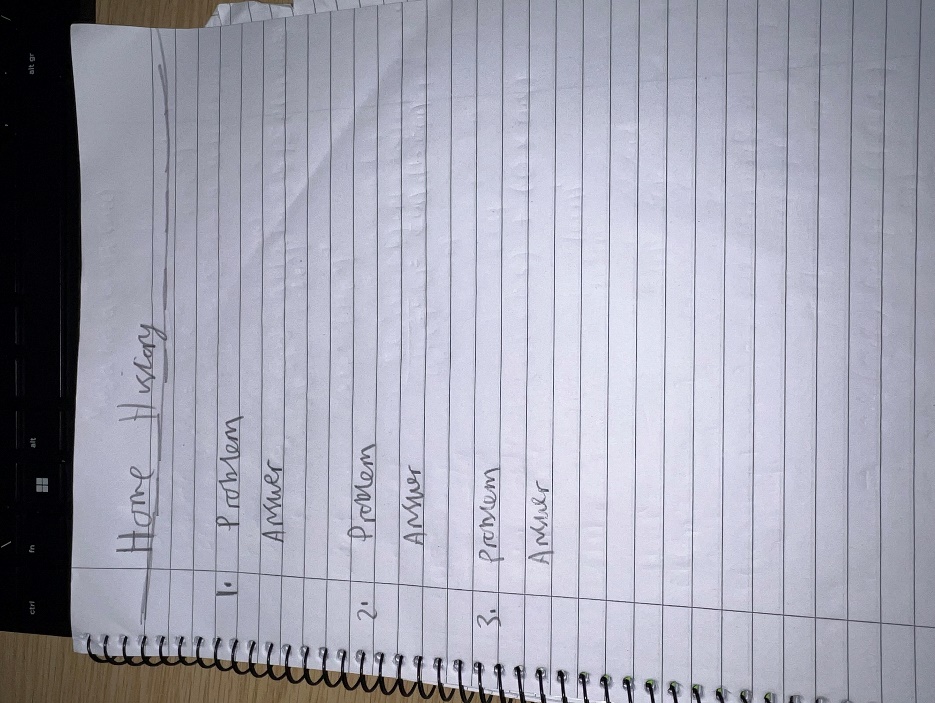
Equations are very important in all fields of maths, science, technology and engineering. They serve as the building blocks to express many concepts whether it is numerically or visually via a graph. An equation solver provides an effective way for professionals to utilise the equation by allowing the computer to both solve and display them. Students on the other hand can view the step-by-step solutions and graphs to learn from.

To create this large-scale software, I have broken it down to several stages as shown in the ‘*Project plan’*, allowing me to continuously work towards my goal via tasks that are either executed in parallel or sequentially.

This project will test core concepts I have already learnt as well as my ability to learn new ideas and implement them in real time. I have always been interested in maths and this will allow me to use my computer science knowledge in a field that I have a passion for, allowing me to produce a final that showcases the concepts and ideas I have learnt on this course.

**UI/UX Mockup**





**Project plan:**

**References:**

<https://math.libretexts.org/Bookshelves/Algebra/Algebra_and_Trigonometry_1e_(OpenStax)/11%3A_Systems_of_Equations_and_Inequalities/11.06%3A_Solving_Systems_with_Gaussian_Elimination>

<https://www.khanacademy.org/math/algebra/x2f8bb11595b61c86:quadratic-functions-equations/x2f8bb11595b61c86:quadratic-formula-a1/v/using-the-quadraticformula#:~:text=The%20quadratic%20formula%20helps%20us,))%2F(2a)%20>

<https://brilliant.org/wiki/newton-raphson-method/>