




Implementation of a Web-Based Complex Formula Editor

T.D.B.S. RUBERU
(MS24018678)

A THESIS
SUBMITTED TO
SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

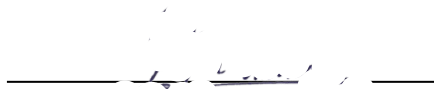
December 2025

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

 4/1/2026

Prof. Samantha Rajapaksha

Approved for MSc. Research Project:



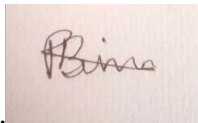
MSc in IT Programme Co-ordinator, SLIIT

Approved for MSc:

Head of Graduate Studies, FoC, SLIIT

DECLARATION

This is to certify that the work is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources). The work has not previously been submitted in any form to the Sri Lanka Institute of Information Technology or to any other institution for assessment for any other purpose.



Sign:

T.D.B.S.Ruberu

Date:2026.01.05.....

ABSTRACT

Implementation of a Web-Based Complex Formula Editor

T.D.B.S. Ruberu

MSc. In Information Technology

Supervisor: Prof.Samantha Rajapaksha

December 2025

Mathematical formula editing in web-based form has also not been easily achieved because it does not have real-time character-level editing, real-time handwriting recognition, and a smooth integration of the database. Current solutions like LaTeX editors or graphical tools that are not dynamically updated have the disadvantage of requiring one to re-write whole expressions to fix small mistakes. Moreover, they can only be accessible to students, researchers, and engineers to some extent due to the use of coding syntax or tedious interfaces. The proposed research is a new Web-Based Complex Formula Editor that would address these limitations. The system permits character level editing in real-time, so that users can edit individual formula portions directly. It is based on artificial intelligence (AI)-driven handwriting recognition to precisely change handwritten input into structured digital representations such as MathML and LaTeX. Besides, the editor has an equation retrieval system which is linked to the OpenMath standard. Through advanced web technologies and AI-based techniques, this performance fills the gap between usability and computing capabilities. The proposed editor has the capability of an easy to use, multipurpose platform to create and manage digital mathematical content and improve productivity and accessibility to a large extent.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who contributed to the successful completion of this research. First and foremost, I am profoundly grateful to my supervisor, Prof. Samantha Rajapaksha, for his invaluable guidance, continuous encouragement, and expert insights throughout this study. Her mentorship played a pivotal role in shaping the direction and quality of this work.

I extend my sincere appreciation to the Sri Lanka Institute of Information Technology (SLIIT) for providing the necessary resources, facilities, and academic support that made this research possible. The institution's commitment to fostering innovation and research excellence greatly facilitated the progress of this project.

Finally, I am deeply thankful to my family for their unwavering support, patience, and encouragement during this challenging yet rewarding journey. Their belief in my efforts kept me motivated at every stage.

This accomplishment would not have been possible without the contributions of these individuals and institutions, and for that, I remain truly grateful.

TABLE OF CONTENTS

DECLARATION	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT	iv
TABLE OF CONTENTS.....	v
List of Figures	viii
List of Tables	ix
Chapter 1 Introduction	9
1.1 Background.....	9
1.2 Problem definition	10
1.3 Research Questions	10
1.4 Significance of Study	11
1.5 Research Objectives.....	12
1.5.1 Main Objective.....	12
1.5.1 Sub-Objectives	13
Chapter 2 Literature Review	16
2.1 Review of Existing Literature	16
2.1.1 Existing Web-Based Formula Editors.....	16
2.1.2 Handwriting Recognition for Mathematical Expressions	17
2.1.3 Real-Time Editing in Mathematical Formula Editors.....	18
2.1.4 Equation Retrieval and Database Integration.....	22
2.1.5 AI and Machine Learning in Mathematical Expression Processing	23
2.2 Gap Identification	24
2.3 Conceptual Framework.....	25
2.3.1 AI-Powered Handwriting Recognition	25
2.3.2 Real-Time Editing Mechanism	26
2.3.3 Equation Retrieval System.....	26
2.3.4 Conclusion	26
Chapter 3 Research Design and Methodology.....	27
3.1 Research Design.....	27
3.2 Methodology	27
3.2.1 System Architecture Overview	29
3.2.2 Development Phases & Implementation Flow.....	32
3.3 Data Collection Methods	43
3.3.1 Initial Requirements and Expert Consultation	44
3.3.2 Developmental and Functional Testing Data	44
3.3.1 Usability and User Acceptance Data.....	45

3.4 Data Analysis Methods	45
3.5 System Requirements.....	47
3.5.1 Functional Requirements	47
3.5.2 Non-functional requirements	47
3.5.3 System Requirements.....	48
3.5.4 User Requirements.....	48
Chapter 4 Results and Discussion.....	49
4.1 Online editor Handwritten Math Formula Recognition (HMF).....	49
4.1.1 Web Interface and MyScript Integration Implementation.	50
4.1.2 Custom Handwriting Recognition Model	52
4.1.3 Interpretation of Model Efficacy.....	52
4.1.4 Comparative Advantage and Hybrid Rationale	56
4.1.5 Discussion of Symbol Recognition Results	56
4.2 Real-Time Character-Level Editing Implementation.....	57
4.2.1 Structured Formula Editing via AST Parsing and Character-Level Operations	59
4.2.2 Flow of AST Parsing Explanation	60
4.3 Comprehensive Performance Benchmarking.....	64
4.3.1 System Latency and Responsiveness Analysis	64
4.3.2 Handwriting Recognition: A Multi-faceted Accuracy Report	65
4.4 User Study: Quantitative and Qualitative Findings	67
4.4.1 Task Performance and Efficiency	67
4.4.2 Thematic Analysis and Qualitative User Feedback.	68
4.5 Discussion: Synthesis and Interpretation	68
4.5.1 Revisiting the Research Questions.....	68
4.5.2 Architectural Efficacy and Trade-offs	69
4.5.3 Limitations and Threats to Validity	69
4.6 Comparative Analysis of the Existing Tools.	70
Chapter 5 Conclusion.....	71
5.1 Achievement of Research Objectives	71
5.2 Significance and Contribution	72
5.3 Recommendations for Future Work.....	73
References.....	74
Appendix.....	77
Appendix 1: Handwritten Mathematical Symbol Recognition Model.....	77
Appendix 2: Real-Time Character-Level Editing Implementation.....	82
Appendix 3: Semi-Structured Interview Questionnaire: Web-Based Complex Formula Editor	92
Appendix 4: Whole System Configuration and Hyperparameters.....	95

Appendix 5: User Study Task Protocol and Materials.....	98
Appendix 6: Handwritten Mathematical Symbol Recognition Model.....	100

List of Figures

Figure 2.1 Conceptual Framework of the Formula Editor	21
Figure 2.2 Node path of the Editor	22
Figure 2.3 Conceptual Framework of the Formula Editor	26
Figure 3.1 Architecture Diagram of the Web-Based Formula Editor	29
Figure 3.2 Data Flow Diagram of the Web Based Formula Editor	30
Figure 3.3 Key components and workflow of the Web Based Formula Editor	31
Figure 3.4 System architecture diagram showing modular components.	32
Figure 3.5 AST parsing algorithm flowchart for the fraction.	33
Figure 3.6 AST Node Construction Process	34
Figure 3.7 Variable Extraction Process.....	35
Figure 3.8 Color-coded AST display.	36
Figure 3.9 Complete Output Generation.....	36
Figure 3.10 Token extraction and navigation logic.	37
Figure 3.11 Hybrid Recognition Architecture	38
Figure 3.12 Unified Workflow Integration	40
Figure 3.13 Iterative Improvement	43
Figure 3.14 Quantitative Data Analysis	46
Figure 3.15 User Feedback Analysis	46
Figure 4.1 Front end of the Handwritten recognition part.	50
Figure 4.2 Call MyScript API.....	51
Figure 4.3 Validation Accuracy for the custom CNN model.....	53
Figure 4.4 Python code for visualizing the training history of the custom Convolutional Neural Network (CNN) model.	53
Figure 4.5 Model Accuracy Progression.....	54
Figure 4.6 Model Loss Convergence.	54
Figure 4.7 Identify symbols in the custom CNN model.	57
Figure 4.8 Advanced Formula Editor Front-End (Real-Time Character-Level Editing).....	58
Figure 4.9 Visual Execution Flow – prototype of Character-Level Editing	60
Figure 4.10 Mean System Latency with Standard Deviation for Key Operations.....	64
Figure 4.11 Comparative Accuracy of Recognition Strategies.....	66

List of Tables

Table 2.1 Technical Issues in Character- level editing of the Existing Formula Editors.....	19
Table 4.1 Abstract Syntax Tree (AST) Representations for Mathematical Operations.....	62
Table 4.2 Detailed Symbol Recognition Accuracy Metrics.....	65
Table 4.3 User Task Performance Metrics (n=10).....	67
Table 4.4 Feature Comparison with Existing Formula Editors.....	70