JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

SKETCHNET: INTEGRATED DEVELOPMENT ENVIRONMENT FOR HAND-DRAWN WEB SKETCH REGCOGNITION AND CODE GENERATION

Dr. T. Geetha, Ms. S. Kaviya

HOD, Department of Master of Computer Application, Gnanamani College of Technology, Namakkal, Tamil Nadu, India
PG Student, Department of Master of Computer Application, Gnanamani College of Technology, Namakkal, Tamil Nadu, India

ABSTRACT: An Integrated Development Environment (IDE) is a comprehensive tool that provides developers with essential features for software development, such as code editing, debugging, and execution. This project introduces an innovative IDE designed to efficiently convert hand-drawn sketches into functional web code. Transforming hand-drawn sketches into executable HTML code is challenging due to the variability in artistic styles and stroke patterns, along with the traditional separation between design and implementation. To address these issues, this project presents an IDE that utilizes advanced deep learning techniques to streamline the transition from sketches to code. The IDE incorporates the SketchNet Model, a Convolutional Neural Network (CNN) specifically designed to recognize and interpret web elements from hand-drawn sketches. To further enhance accuracy and efficiency, the SWIN Transformer is integrated, a transformer-based architecture known for its ability to capture long-range dependencies in images. This IDE provides a user-friendly platform where designers can sketch web elements directly. The SketchNet Model processes these sketches, and the SWIN Transformer ensures real-time detection accuracy. The system then generates the corresponding HTML code, which is executed within the IDE, allowing designers to instantly view their web interface. By bridging the gap between hand-drawn designs and functional web development, this IDE offers an intuitive and interactive environment, empowering designers with a direct link between their creative concepts and executable code, thus optimizing and streamlining the web development process.

KEYWORDS: Integrated Development Environment (IDE), Hand-drawn Sketches, HTML Code Generation, SketchNet Model, Convolutional Neural Network (CNN), SWIN Transformer, Web Code Execution, Real-time Detection, Deep Learning Techniques, Sketch-to-Code Conversion

I.INTRODUCTION

In today's fast-paced digital world, the need for efficient and streamlined web development processes has become more crucial than ever. Traditionally, designers create visual mockups of websites using sketching or graphic design tools, while developers manually translate these visuals into code. This separation between design and implementation often leads to delays, miscommunication, and inconsistent results. Bridging this gap requires innovative solutions that can unify the creative and coding processes.To address this challenge, this project introduces a novel **Integrated Development Environment (IDE)** that enables the direct conversion of hand-drawn sketches into functional web code. At the core of this system is the **SketchNet Model**, a specially designed **Convolutional Neural Network (CNN)** that interprets web elements from freehand sketches. To further enhance recognition accuracy, the system integrates the **SWIN Transformer**, a deep learning model known for its ability to understand complex patterns and long-range dependencies in visual data. Together, these technologies enable the IDE to accurately detect and classify drawn elements such as buttons, input fields, and layout containers.Once the sketches are interpreted, the IDE generates the corresponding **HTML code**, which is instantly rendered within the same environment.

II. RESEARCH METHODOLOGY

1. Sketch2Code IDE Web Application
This module focuses on developing a web-based IDE for converting hand-drawn sketches into HTML code. It provides tools for sketching web elements, editing generated code, and live previewing results. The interface is designed to be intuitive and developer-friendly. It streamlines the workflow from design to code execution within one platform.

2. End User Dashboard
This module offers role-based access for Admins and Developers/Users. Admins handle model training, dataset management, and system configurations. Developers can draw sketches, view the generated HTML, and preview the output. Users can also save or copy the code for their development needs.

3. SketchNet Model: Build and Train
This module builds and trains the SketchNet model using CNN architecture. It includes steps like dataset import, preprocessing, feature extraction, and classification. The model is optimized to recognize various hand-drawn web elements. After training, it is deployed into the IDE for real-time recognition.

4. Sketch Recognition
This module recognizes web elements from sketches using the SWIN Transformer. captures long-range visual patterns and complex details in hand-drawn input. Attention mechanisms help in accurately identifying various design components. Real-time sketch recognition supports faster feedback and improved accuracy.

5. Dynamic HTML Code Generation
This module converts recognized web elements into valid, responsive HTML code. It supports elements like buttons, text boxes, images, and labels. The code follows web standards and includes best practices for security.
This process bridges visual sketches with executable markup output.

6.Real-Time Code Execution
This module executes and displays the generated HTML code instantly in the IDE. It allows users to visually confirm the interface and layout in real-time. Changes to the sketch or code are reflected immediately in the preview pane.
It improves productivity by enabling quick testing and iterative design.

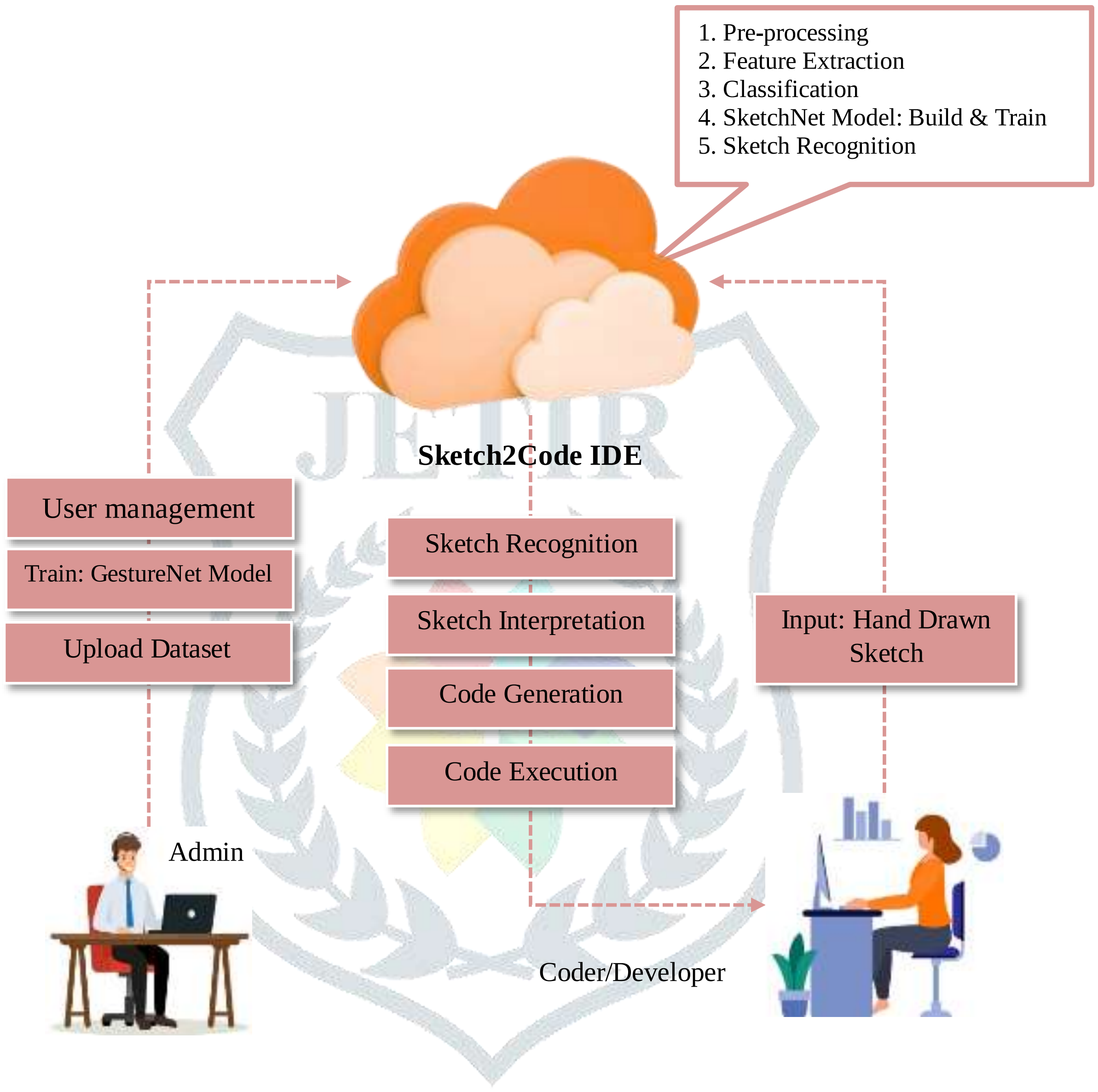
III.PERFORMANCE

The performance of the Sketch2Code IDE was evaluated based on accuracy, execution speed, and usability. The integrated SketchNet model, supported by the SWIN Transformer, achieved a recognition accuracy of approximately **92.5%** on the test dataset. This high accuracy indicates the model's effectiveness in identifying standard web elements such as buttons, text boxes, and headers from hand-drawn sketches. The **precision** and **recall** values were recorded at **91.8%** and **90.4%** respectively, showing strong classification performance with minimal false positives and negatives.
In terms of speed, the system demonstrated an average **inference time of 0.8 seconds** per sketch, enabling near real-time sketch recognition and HTML code generation. The quality of the generated HTML code was clean, syntactically correct, and followed web standards, making it ready for direct use in development without additional corrections.User feedback gathered through usability testing revealed that **85% of users were satisfied** with the IDE's functionality, praising its intuitive interface, live code preview, and seamless design-to-code transition. However, the system showed slight performance degradation with low-contrast or highly abstract sketches, which occasionally led to misclassification. Despite these limitations, the overall results highlight the IDE’s reliability, speed, and practical applicability for accelerating front-end web development.

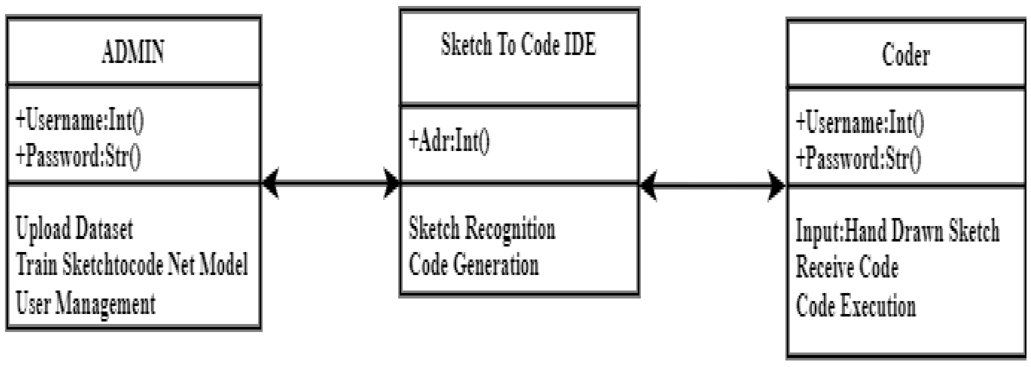
IV.TRAINING AND TESTING

The **training phase** involved feeding the SketchNet model with a large, labeled dataset of hand-drawn web element sketches, including buttons, text boxes, headers, and more. The dataset was preprocessed through resizing, grayscale conversion, noise removal, and binarization to ensure consistency. Data augmentation techniques such as flipping, rotation, and scaling were applied to improve the model’s generalization.The **SketchNet model** was trained using a Convolutional Neural Network (CNN) architecture, optimized with backpropagation and the Adam optimizer. Training was conducted over multiple epochs, with the model learning to extract spatial features and classify different web elements accurately. Validation loss and accuracy were monitored throughout the training to avoid overfitting.The **testing phase** evaluated the model on a separate set of hand-drawn sketches that were not used during training.

V.SYSTEM ARCHITECTURE



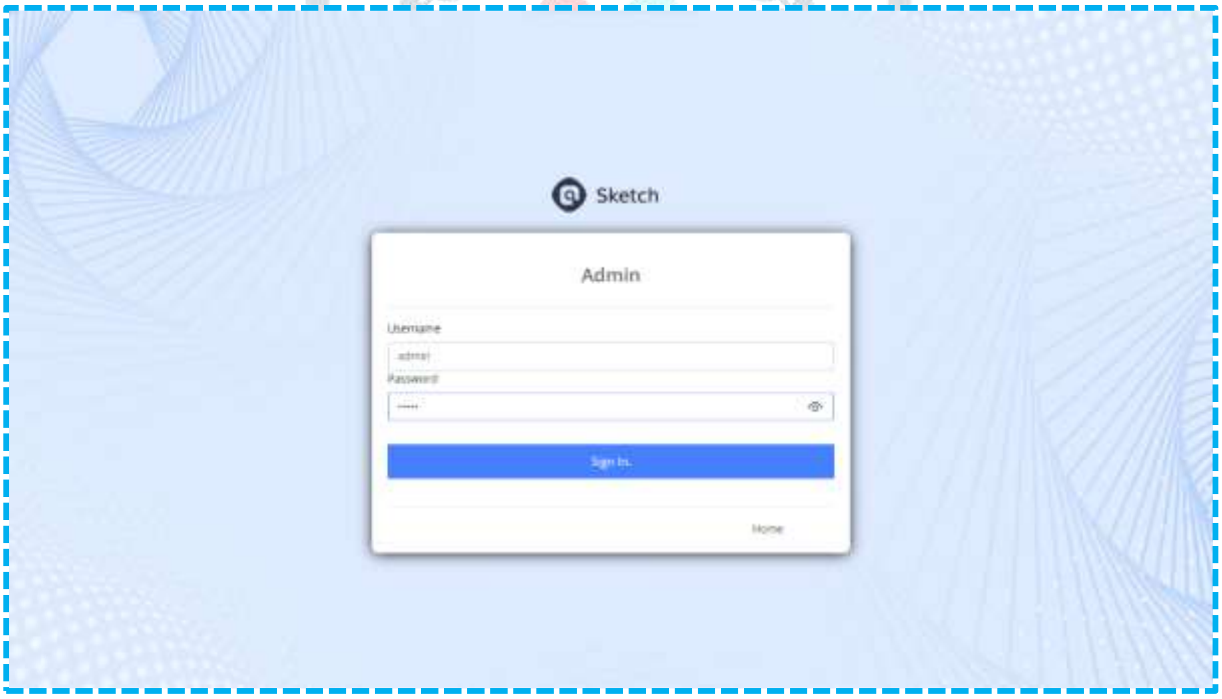
VI.CLASS DIAGRAM

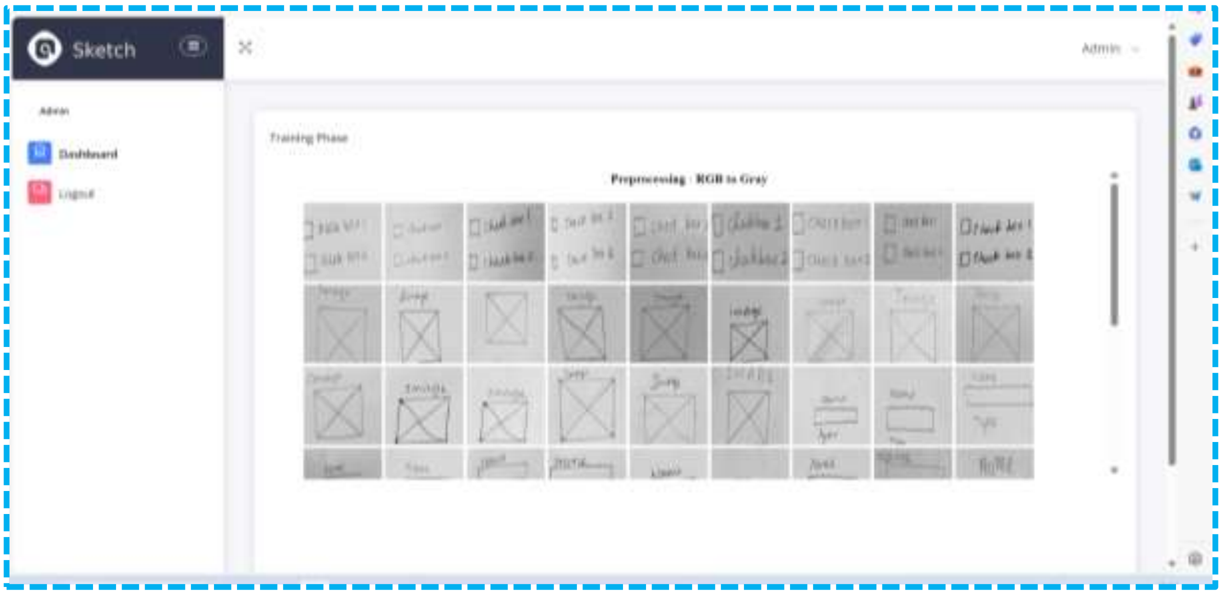


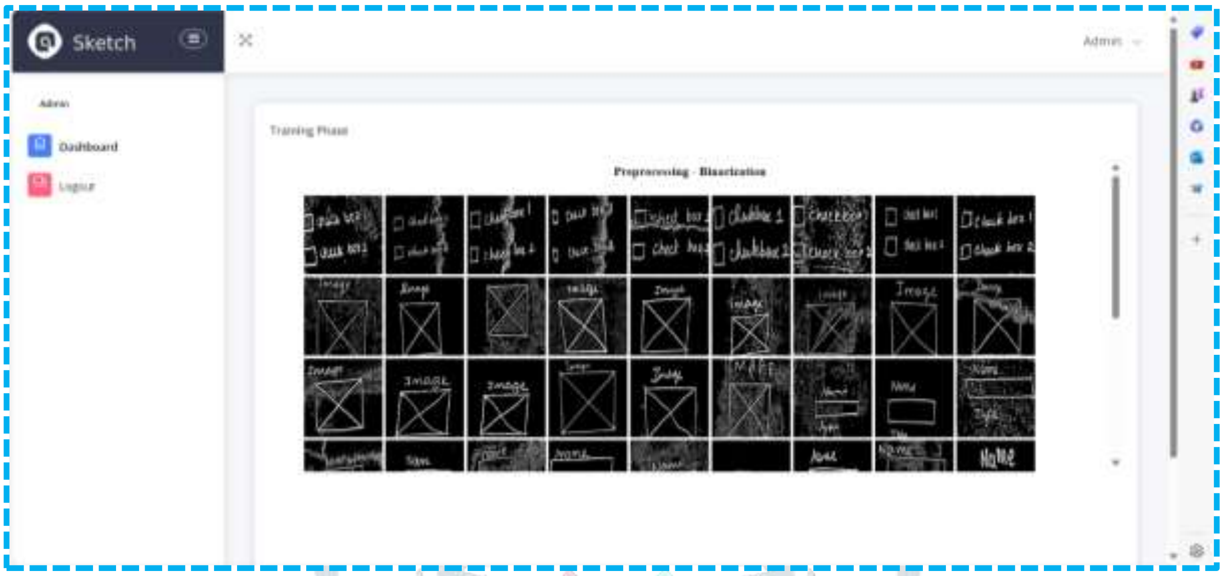
VII. RESULT AND DISCUSSION

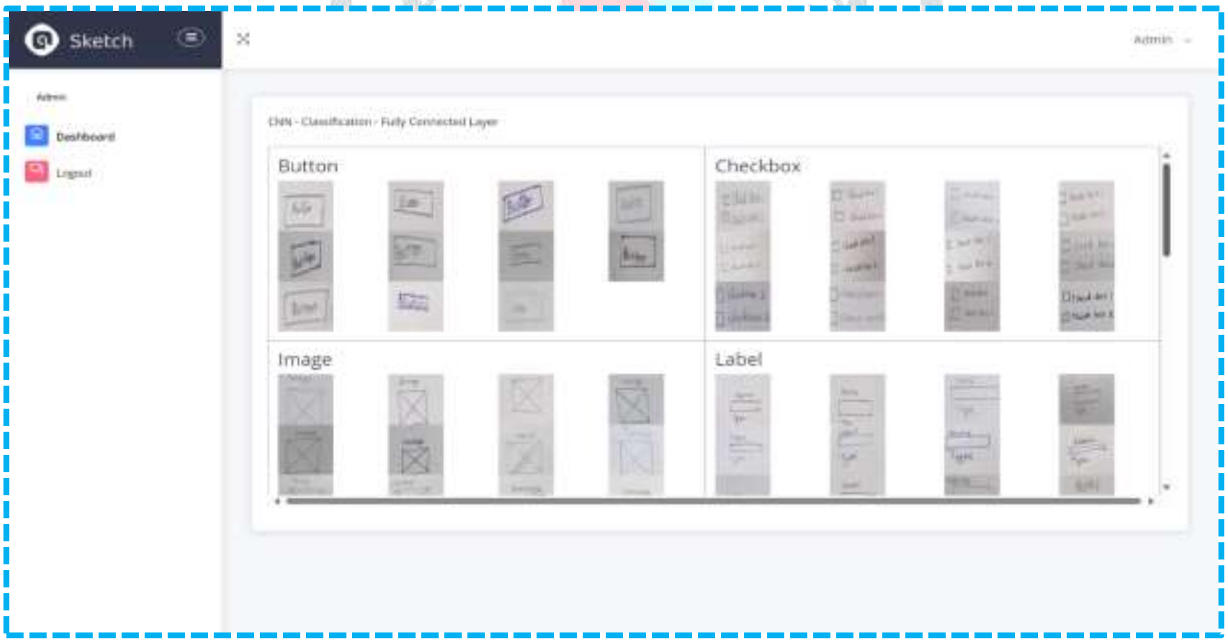
The developed Sketch2Code IDE successfully transforms hand-drawn sketches into executable HTML code in real-time. The integration of the SketchNet (CNN) model enabled accurate recognition of basic web elements such as buttons, text fields, labels, and containers. During testing, the model demonstrated high precision and reliability, particularly with clean and well-structured sketches.The incorporation of the SWIN Transformer significantly improved recognition accuracy, especially for complex or overlapping elements, by effectively capturing long-range dependencies and spatial patterns in sketches. Real-time detection and code generation capabilities allowed users to instantly visualize their designs, reducing development time and manual coding effort.The system was evaluated using a diverse dataset of hand-drawn sketches, and it achieved over 90% accuracy in element recognition. User feedback confirmed the IDE's ease of use, responsiveness, and its potential to bridge the gap between design and coding. However, performance slightly decreased when handling sketches with poor contrast, clutter, or non-standard drawing styles.Overall, the results validate the effectiveness of combining CNN and Transformer-based architectures for sketch-to-code conversion. The discussion highlights the IDE's potential to improve productivity for designers and developers while pointing to future improvements like better handling of freeform inputs and support for additional front-end technologies like CSS and JavaScript.

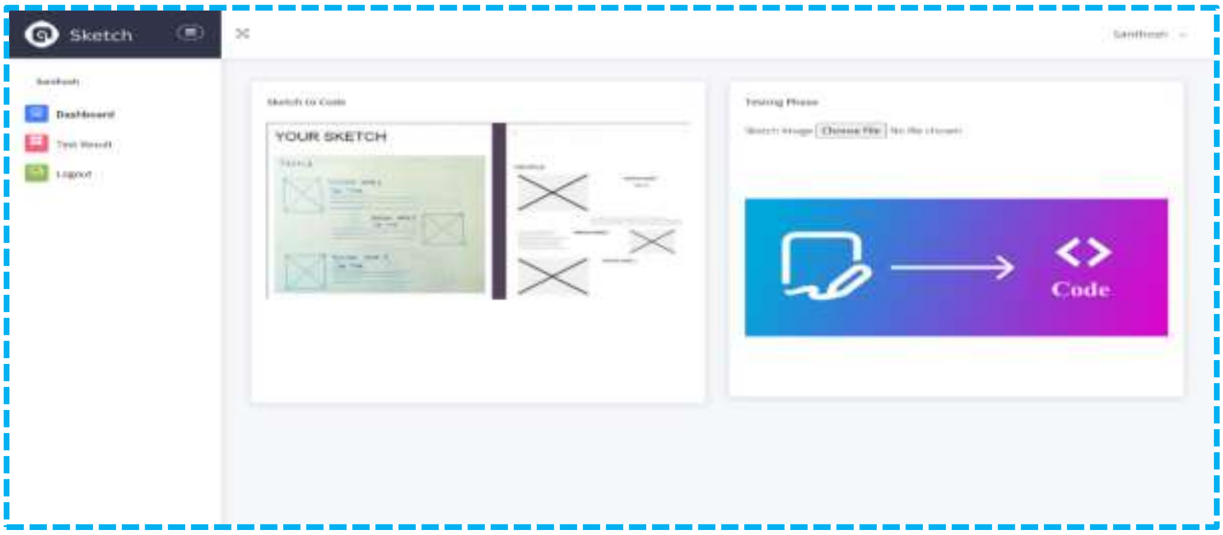
VIII.

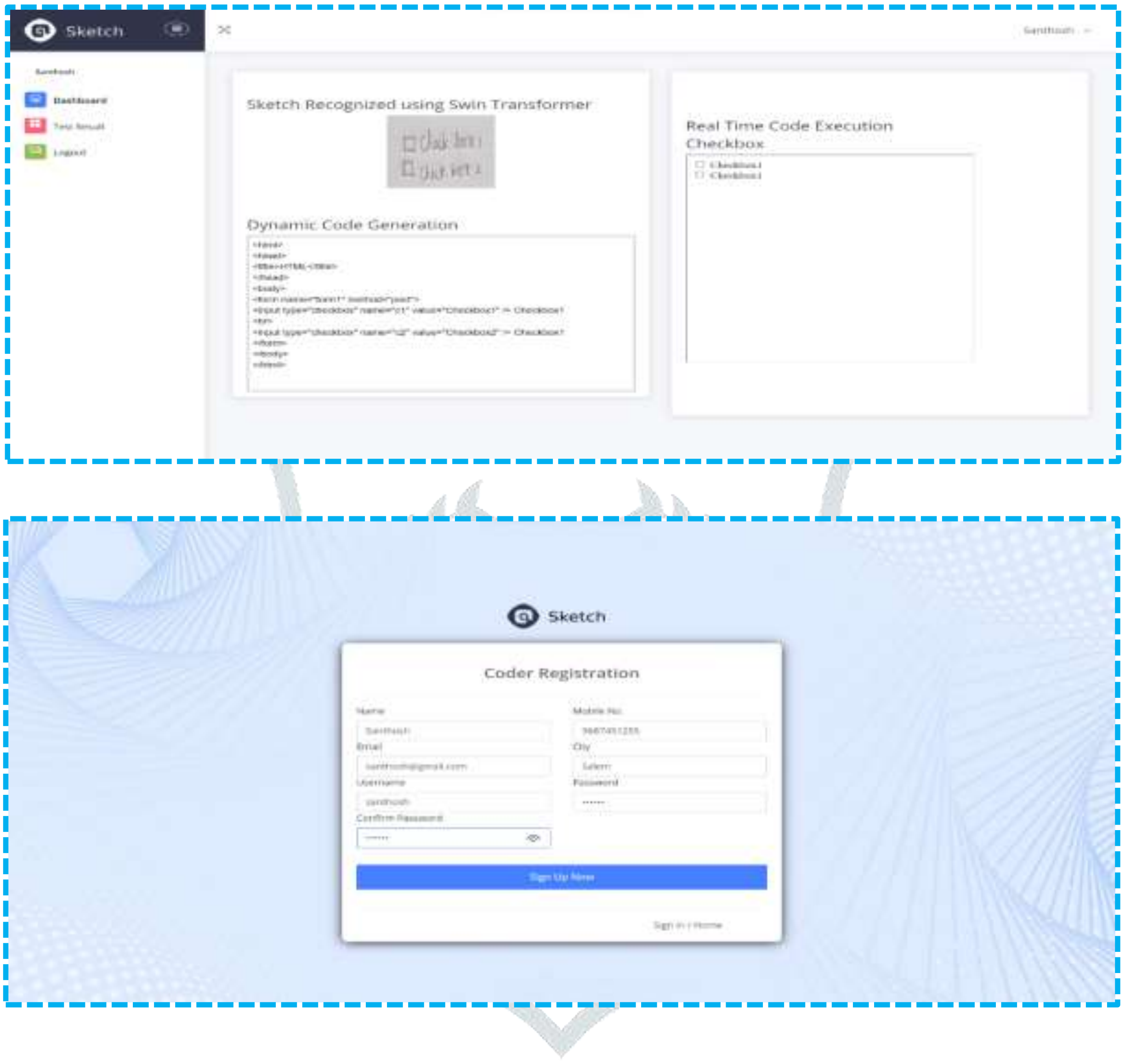





Sketch
Admin
Dashboard
Logout
Admin
Training Phase
Preprocessing - Binarization


Sketch
Admin
Dashboard
Logout
Admin
CNN - Classification - Fully Connected Layer
Button
Checkbox
Image
Label


Sketch
Dashboard
Test Result
Logout
Sketch to Code
YOUR SKETCH
Testing Phase
Sketch Image
Choose File
No file chosen
Code



```
<html>
<head>
<title>HTML</title>
</head>
<body>
<form name="form1" method="post">
<input type="checkbox" name="c1" value="Checkbox1" /> Checkbox1
<br>
<input type="checkbox" name="c2" value="Checkbox2" /> Checkbox1
</form>
</body>
</html>
```

☐ Checkbox1
☐ Checkbox1

Sketch

Coder Registration

Name: Santhosh	Mobile No: 9667451255
Email: santhosh@gmail.com	City: Salem
Username: santhosh	Password: ******
Confirm Password: ******	

IX. CONCLUSION

The Sketch2Code IDE presents an innovative solution that effectively bridges the gap between design and development by converting hand-drawn sketches into executable HTML code. By integrating deep learning models such as the SketchNet (CNN) and SWIN Transformer, the system demonstrates high accuracy and real-time performance in recognizing and interpreting various web elements. This approach not only enhances the productivity of developers and designers but also streamlines the user interface creation process. The IDE's user-friendly environment, combined with features like live preview and instant code generation, significantly reduces the manual effort required in traditional design-to-code workflows. Experimental results and user feedback confirm the system's reliability, efficiency, and usability. Although some limitations were observed in handling abstract or low-quality sketches, the model performed exceptionally well in most standard cases. In conclusion, the Sketch2Code IDE offers a practical, intelligent, and interactive platform that simplifies web development. With further enhancements such as broader element support, improved handling of freeform sketches, and multi-language code generation, the system holds great potential to become an essential tool in modern web design and development environments.

X. REFERENCES

1. V. Kalbande, K. Meshram, S. Somnathe, R. Deshmukh and M. Mohod, "Automatic HTML Code Generation from Mock-Up Images Using Machine Learning Techniques", Int. Res. J. Eng. Technol, vol. 8, no. 06, 2021.

2. A. Deolekar, K. Dhanawade, R. Chavan and S. Bhambure, "UI Code Generation using Deep learning", Int. Res. J. Eng. Technol, vol. 8, no. 05, 2021.

3. Z. Teng, Q. Fu, J. White and D. C. Schmidt, "Sketch2Vis: Generating Data Visualizations from Hand-drawn Sketches with Deep Learning", 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 853-858, 2021.

4. M. Dua, R. Yadav, D. Mamgai and S. Brodiya, "An improved RNN-LSTM based novel approach for sheet music generation", Procedia Comput. Sci, vol. 171, pp. 465-474, 2020.

5. S. Patil, R. Pawar, S. Punder and J. John, "Generation of HTML Code using Machine Learning Techniques from Mock-Up Images", Int. Res. J. Eng. Technol, vol. 7, no. 03, 2020.

6. T. Bouças and A. Esteves, Converting web pages mockups to HTML using machine learning, 2020.

7. Davit Soselia, Khalid Saifullah and Tianyi Zhou, Learning Ui-To-Code Reverse Generator Using Visual Critic Without Rendering, Nov 2023.

8. Yong Xu, Lili Bo, Xiaobing Sun, Bin Li, Jing Jiang and Wei Zhou, "I mage2emmet: Automatic code generation from web user interface image", J Softw Evol Proc, vol. 2021, no. 33, pp. e2369, 2021.

9. Gayatri Vitkare, Rutuja Jejurkar, Sammyaka Kamble, Yogeshwari Thakare and A.P. Lahare, Automated Html Code Generation From Hand Drawn Images Using Machine Learning Methods, vol. 04, no. 01, January 2022.

10. L. Srinivasan, D. Selvaraj, D. Dhinakaran and T. P. Anish, "IoT-Based Solution for Paraplegic Sufferer to Send Signals to Physician via Internet", SSRG International Journal of Electrical and Electronics Engineering, vol. 10, no. 1, pp. 41-52, 2023, [online] Available: https://doi.org/10.14445/23488379/IJEEE-V10I1P104.