User Interface Code Generation from Hand-drawn Sketch

Team Members

Manoj Paudel	(THA077BCT025)
Prince Poudel	(THA077BCT036)
Ronish Shrestha	(THA077BCT040)
Sonish Poudel	(THA077BCT042)

Supervised By:

Er. Devendra Kathayat

Department of Electronics and Computer Engineering
Institute of Engineering, Thapathali Campus

July, 2024

Presentation Outline

- Problem Statement and Objectives of Project
- Proposed Methodology
- Results and Discussion
- References

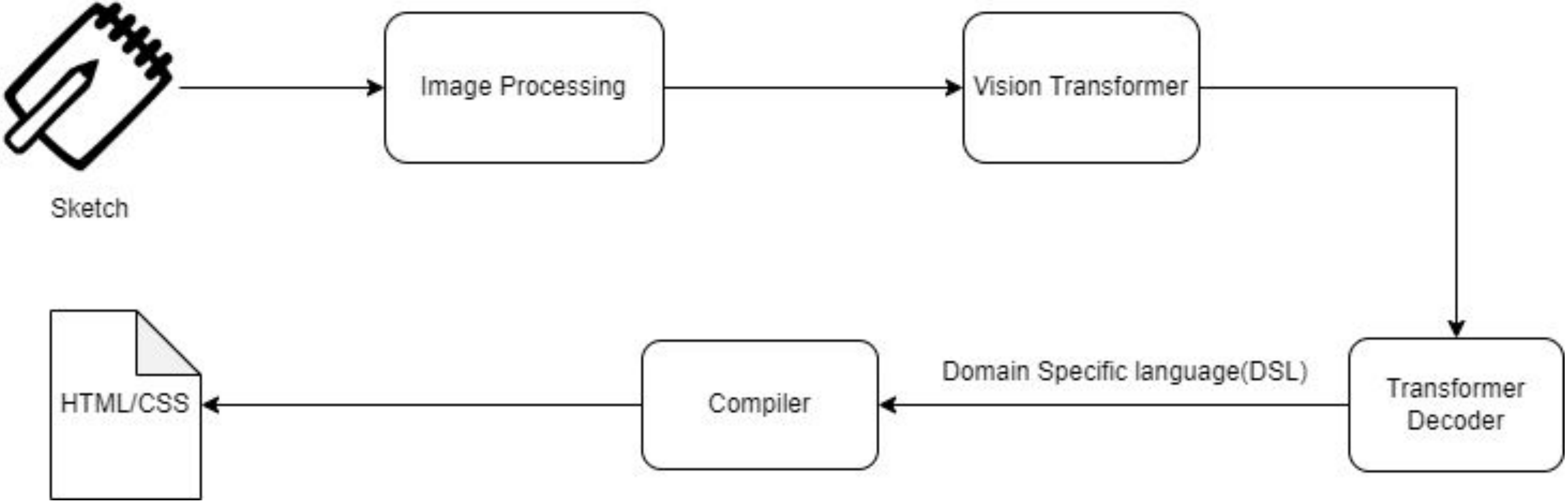
Problem Statement

- Conversion of sketches into function GUI code.

Objectives of Project

- To construct a model able to generate quick GUI prototype from sketch into HTML code.
- To create interactive user interface.

Methodology - [1] (System Block Diagram)



Methodology - [2] (Datasets)

- Dataset is a wireframe sketch and associated DSL code.
- We were not aware of any dataset which contained wireframes sketches and DSL code
- We will create our own dataset.

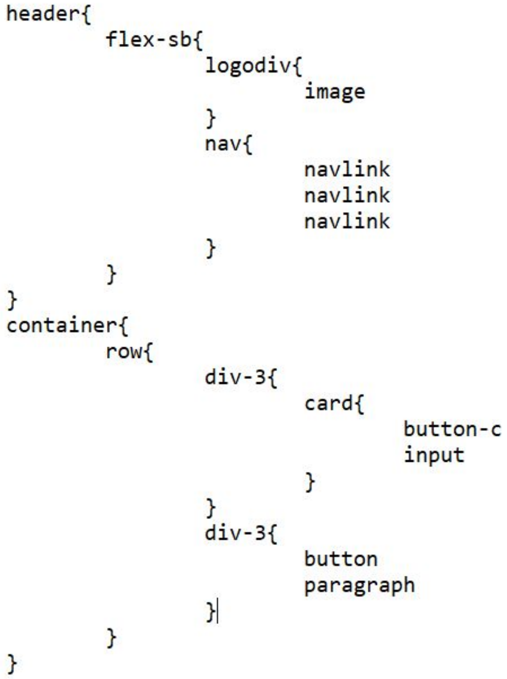
Methodology - [3] (DSL)

- Specialized language designed to address specific aspect or needs of a particular language.
- Designed the simple lightweight DSL to describe the GUI.
- Elements in DSL will be categorized into different hierarchical structures.

Methodology - [4] (DSL elements)

- Body
- Root
- Header
- Nav
- Navlink
- Logodiv
- Container
- Row
- Div-3
- Div-6
- Div-9
- Div-12
- Flex
- Flex-sb
- Flex-c
- Flex-r
- Text
- Text-c
- Text-r
- Paragraph
- Image
- Card
- Input
- Button
- Button-c
- Button-r
- Footer

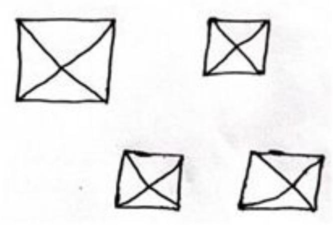
Methodology - [5]
(DSL code)



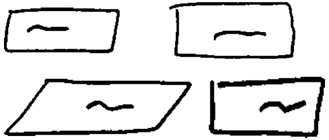
```
header{
        flex-sb{
                logodiv{
                        image
                }
                nav{
                        navlink
                        navlink
                        navlink
                }
        }
}
container{
        row{
                div-3{
                        card{
                                button-c
                                input
                        }
                }
                div-3{
                        button
                        paragraph
                }
        }
}
```

Figure: DSL code

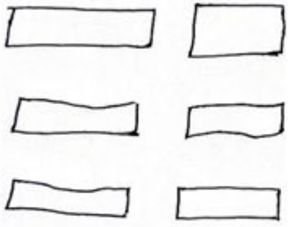
Methodology - [6]
(Sample elements)



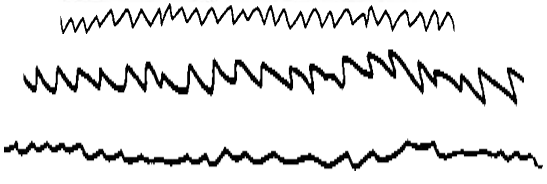
Image



Button



Input

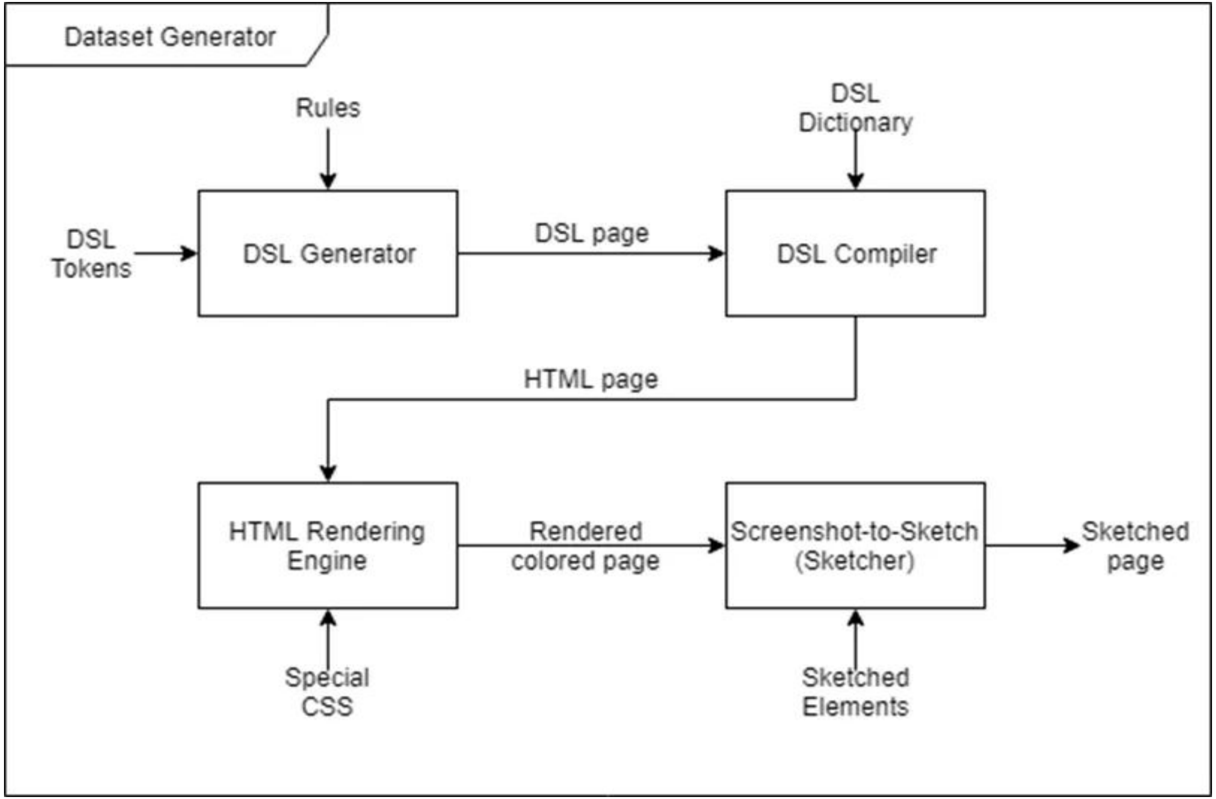


Paragraph

Methodology - [7]
(Dataset Generation Process)

- Dataset is the hand-drawn sketch and it's associated dsl.
- There is no existing dataset for it.
- Dataset generator is created to create as many dataset as required.

Methodology - [8] (Block Diagram)



Methodology - [9]
(Random DSL Generator)

- First step is to generate a DSL randomly.
- It denotes the elements in the user interface.
- DSL is generated by mixing the different possible combination of the element.

Methodology - [10] (Compiling the DSL code)

- Randomly generated DSL code is mapped into the corresponding HTML tag.

Methodology - [11]
(Rendering the Produced HTML)

- Mapped HTML file is rendered into the webpage with the special CSS file.
- CSS file helps to denote the different element with separate colour.

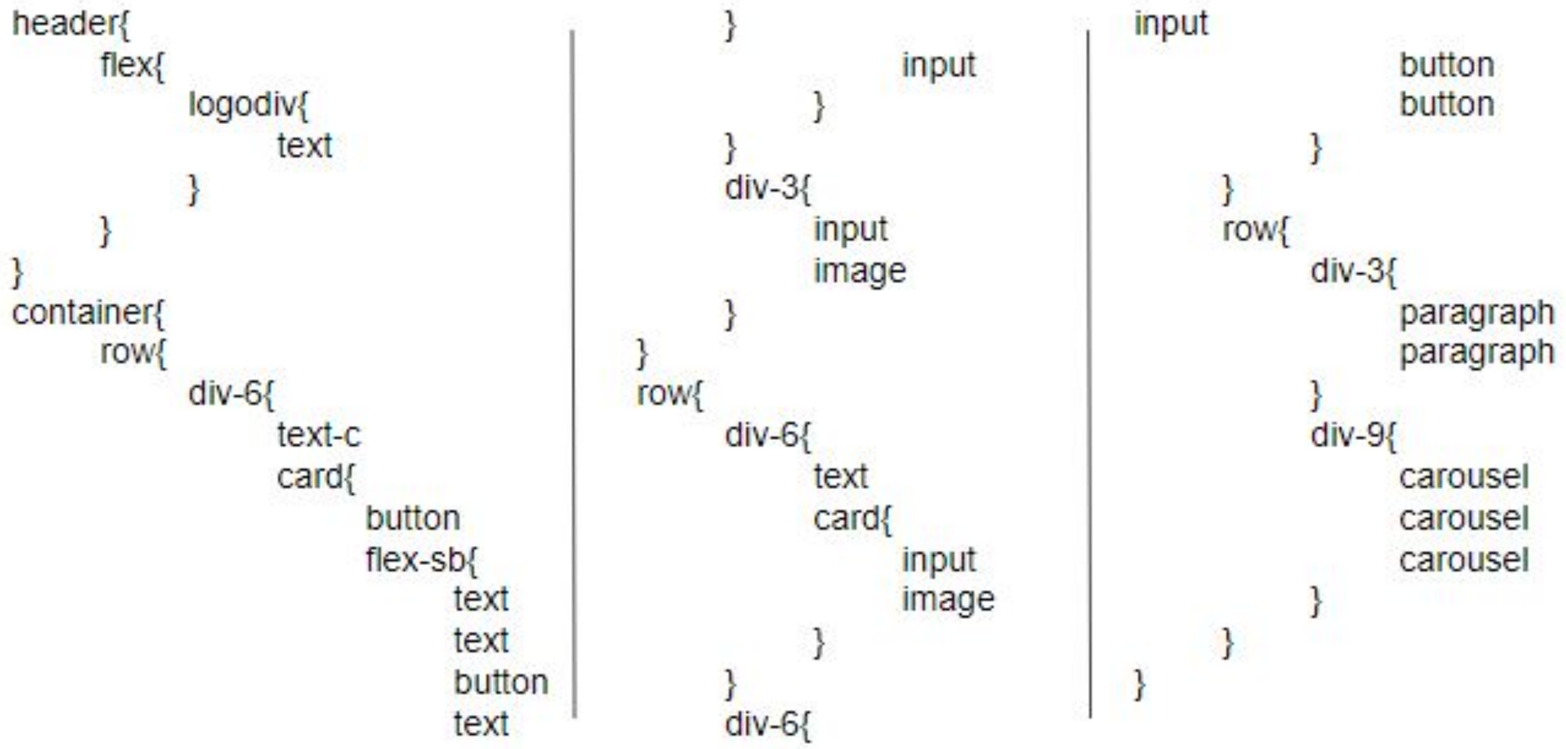
Methodology - [12]
(Finding outline of elements)

- Contour detection is applied to find the boundary of all the elements
- Then, the position is identified.

Methodology - [13]
(Placing hand-drawn sketch of element)

- Hand-drawn sketch of all the element is place at the position identified from above step.
- At last, sketch is obtained with its associated DSL code.

Results and Discussion-[1]



```
header{
    flex{
        logodiv{
            text
        }
    }
}
container{
    row{
        div-6{
            text-c
            card{
                button
                flex-sb{
                    text
                    text
                    button
                    text
```

```
    }
            input
        }
    }
    div-3{
        input
        image
    }
}
row{
    div-6{
        text
        card{
            input
            image
        }
    }
    div-6{
```

```
input
            button
            button
        }
    }
    row{
        div-3{
            paragraph
            paragraph
        }
        div-9{
            carousel
            carousel
            carousel
        }
    }
}
```

Generated DSL code

Results and Discussion-[2]

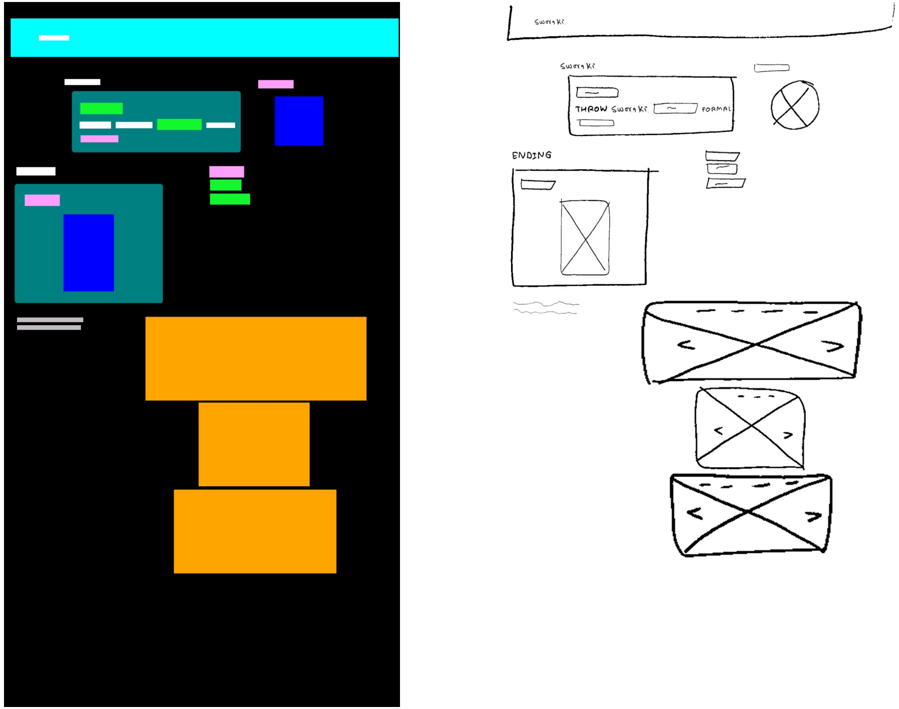


Fig. Generated HTML page and Sketch

References-[1]

1. Tony Beltramelli,"Pix2code: Generating Code from a Graphical User Interface Screenshot," 2017. [Online]. Available: https://www.researchgate.net/publication/325920827_pix2code_Generating_Code_from_a_Graphical_User_Interface_Screenshot. [Accessed: June 2024].
2. Sarah Suleri, Vinoth Pandian,Svetlana Shiskovets,Matthias Jarke,"Eve: A sketch-based Software Prototyping Workbench," 2019. Available: https://www.researchgate.net/publication/332777261_Eve_A_Sketch-based_Software_Prototyping_Workbench. [Accessed: June 2024].
3. Biniam Behailu Adefris, Ayalew Belay Habtie, Yesuneh Getachew Taye, "Automatic Code Generation from Low Fidelity Graphical User Interface Sketches Using Deep Learning," 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9971204. [Accessed: June 2024].
4. Daniel Baulé, Christiane Gresse von Wangenheim, Aldo von Wangenheim, Jean C. R. Hauck and Edson C. Vargas Júnior, "Automatic Code Generation from Sketches of Mobile Applications in End-User Development Using Deep Learning," [Online]. Available: https://www.researchgate.net/publication/349963791_Automatic_code_generation_from_sketches_of_mobile_applications_in_enduser_development_using_Deep_Learning . [Accessed: June 2024].

References-[2]

5. Jia Li, Yongmin Li, Ge Li, Zhi Jin, Yiyang Hao, and Xing Hu, "STC (Sketch To Code) - An Enhanced HTML & CSS Autocode Generator from Handwritten Text and Image Using Deep Learning," IEEE, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10537336. [Accessed: June 2024].
6. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," [Online]. Available: https://arxiv.org/abs/2010.11929. [Accessed: June 2024].
7. "Canny edge detector," Wikipedia, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/Canny_edge_detector. [Accessed: 17-Jun-2024].
8. Q. Xin, Y. Zhang and B. Tan, "Image Captioning with Vision/Text Transformers," [Online]. Available: https://qi-xin.github.io/image%20caption%20generation.pdf. [Accessed: June 2024].