International Conference - Innovation-2021-Innovation-2021
International Journal of Scientific Research in Computer Science, Engineering and Information Technology ISSN : 2456-3307 (www.ijsrcseit.com)
Volume 8, Issue 3, May-June-2021

Automated Website Development

Akashy Mahalle, Shivaraya Patil, Tushar Gangurde, Vaibhav Patil
Department of Computer Engineering, Dr. DY Patil School of Engineering, Charholi BK, via lohegaon, Pune, Maharashtra, India

ABSTRACT

A website helps a business to grow by using different marketing strategies. This report describes a novel approach to develop a website by just providing the text (description of the website) or an image as input. Using Text Input, it will suggest template (screenshots) after identifying the theme of the site inferred from the input. Those templates are converted into code for further customizations for their personal use. Current problem was that a web developer would take more than 15 days only to just make the basic structure of a website. This issue is resolved by our work, which will generate the complete code of the webpage/ website in less amount of time. In this paper, it will tokenize each word to find their synonyms and then mapped it with root words for the theme identification and uses deep learning model to convert templates into code.

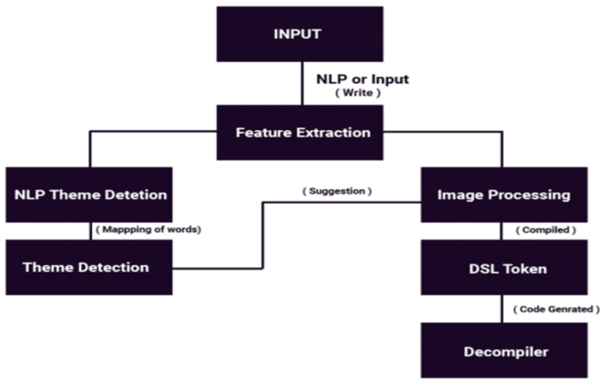
Keywords : Website, Automatic, code

I. INTRODUCTION

The process of implementing client-side software based on a Graphical User Interface (GUI) mock-up created by a designer is the responsibility of developers. Implementing GUI code is, however, time-consuming and prevent developers from dedicating the majority of their time implementing the actual functionality and logic of the software they are building. Moreover, the computer languages used to implement such GUIs are specific to each target runtime system; thus resulting in tedious and repetitive work when the software being built is expected to run on multiple platforms using native technologies. In this paper, we describe a model trained end-to-end with stochastic gradient descent to simultaneously learns to model sequences and spatio-temporal visual features to generate variable-length strings of tokens from a single GUI image as inpu

II. METHODS AND MATERIAL

The proposed system is done in various steps like text segmentation, tokenizing, part of speech tagging, word map and theme, suggesting, CNN, LSTM and decoder as show



- **Input Image and Text**
  User can upload images of UI to convert UI to code.

- **Feature Extraction**
  The methods of input have feature extraction Algorithms which extracts characteristics from both inputs each of them has their own feature extraction algorithm.
  Image input uses CNN as a feature extraction algorithm to get characteristics of input. CNN is widely used in computer vision problems because of its topology, which allows them to extract minor details from the input.

- **Theme Detection**
  After Tokenizing, these phases come into play that uses Name Entity Recognition or Chunking that extracts Noun and Adjectives from tokenized data that are called Chunked data from that data synonym are extracted and mapped with root words.

- **Image Processing and DSL Token generation**
  After the extraction of features from the image using CNN, we used DSL tokens to describe UI Components. To find different graphical components and their relation between each other DSL Token generation is used. DSL reduces the size of search space by reducing the total number of tokens of vocabulary of the DSL.

- **Decoder**
  By using supervised learning method, model is trained by inputting an image I and xt is a contextual sequence of X of T tokens, t ∈ {0 . . . T – 1} as inputs; and xT token is taken as the target label. Input image I is encoded into vector representation p by using CNN-based vision model. LSTM Model is used to encode the input token xt into an intermediate representation qt which allows the model to concentrate more on certain type of tokens and less focus on others.

III. RESULTS AND DISCUSSION

This work will generate code for UI input provided by the user. fig 1 shows input provided by user as an image and code is the output generated by the system. Output contain the actual code of the website that was pass as an input.

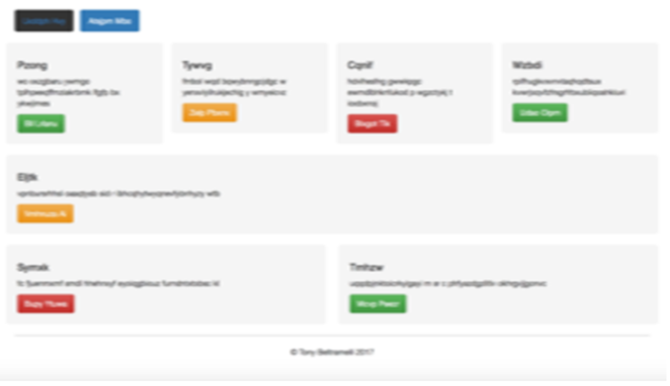
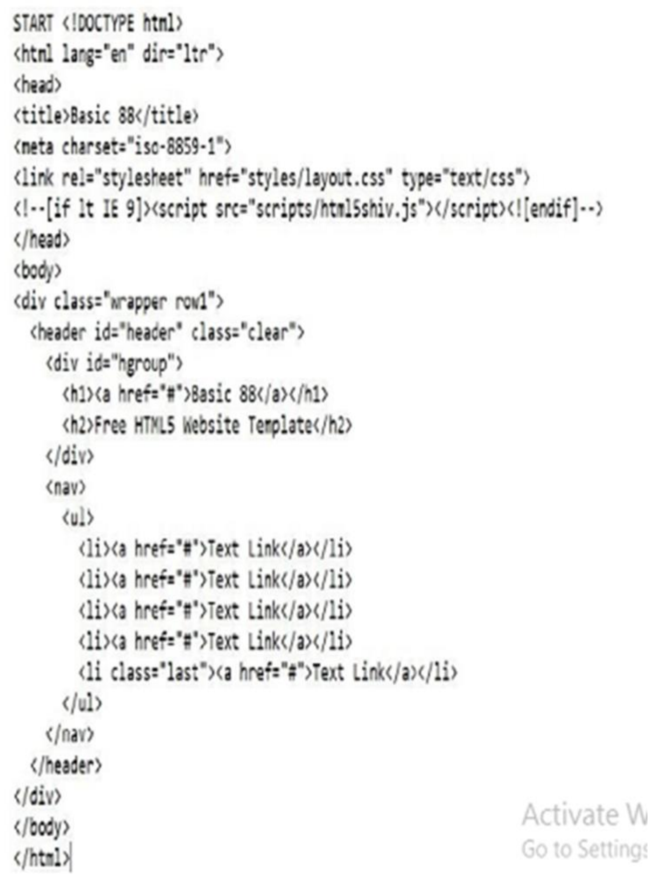


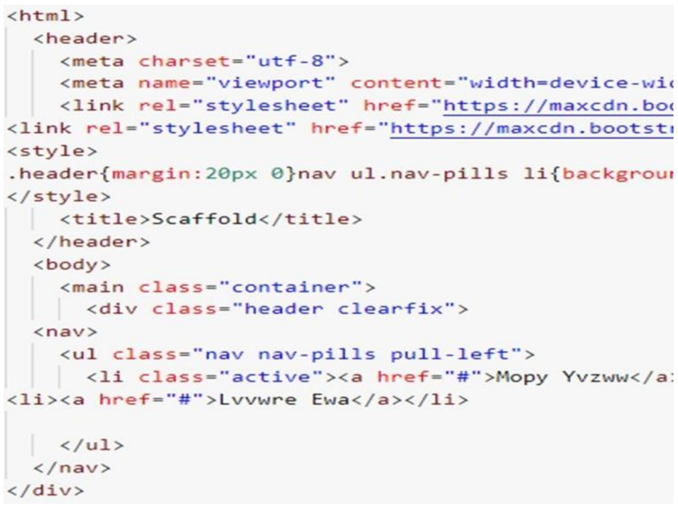
Fig 1



```
START <!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<title>Basic 88</title>
<meta charset="iso-8859-1">
<link rel="stylesheet" href="styles/layout.css" type="text/css">
<!--[if lt IE 9]><script src="scripts/html5shiv.js"></script><![endif]-->
</head>
<body>
<div class="wrapper row1">
  <header id="header" class="clear">
    <div id="hgroup">
      <h1><a href="#">Basic 88</a></h1>
      <h2>Free HTML5 Website Template</h2>
    </div>
    <nav>
      <ul>
        <li><a href="#">Text Link</a></li>
        <li><a href="#">Text Link</a></li>
        <li><a href="#">Text Link</a></li>
        <li><a href="#">Text Link</a></li>
        <li class="last"><a href="#">Text Link</a></li>
      </ul>
    </nav>
  </header>
</div>
</body>
</html>
```

Output

User can copy this code and paste it on any html page as shown in fig 2, and can run the code.



```
<html>
  <header>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-wi
    <link rel="stylesheet" href="https://maxcdn.boo
<link rel="stylesheet" href="https://maxcdn.bootst
<style>
.header{margin:20px 0}nav ul.nav-pills li{backgrou
</style>
    <title>Scaffold</title>
  </header>
  <body>
    <main class="container">
      <div class="header clearfix">
  <nav>
    <ul class="nav nav-pills pull-left">
      <li class="active"><a href="#">Mopy Yvzww</a
<li><a href="#">Lvvwre Ewa</a></li>

    </ul>
  </nav>
</div>
```

IV. CONCLUSION

This proposed system has discussed the theme detection technique, suggesting themes to user, and generating code from the selected template. Current problem was that a web developer will take more than 15 days only to just make the basic structure of a website. This issue is resolved by our work, which will generate the complete code of the webpage/ website in less amount of time. This system can be used by anyone to make a website from just a text input or screenshot of a website to generate code from it.

V. REFERENCES

[1]. J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S.Venugopalan, K. Saenko, and T. Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2625–2634.

[2]. F. A. Gers, J. Schmidhuber, and F. Cummins. 2000. Learning to forget: Continual prediction with lstm. Neural Computation, 12(10), 2451–247.

[3]. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680.

[4]. A. Karpathy and L. Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3128–3137.

[5]. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.