

Web Scalar- Scale Your Own Website

Saahil Bhatnagar¹, Dr. Asha N²

¹MCA Student at The National Institute of Engineering, Mysuru

²Asst. Professor, Head of Dept. of MCA, The National Institute of Engineering, Mysuru

Abstract – Website theme Builder is web domain project which enhances the website theme experience for a non-coder or less developing experience individual to develop the perfect front end using the theme and editing the front-end components of the website they want to build. The editor also transfers the component edited into a code which can be edited with drag and drop technique. The website theme builder also uses ML (Machine Learning) algorithm about the taste of the user by asking them various type of questions and combining that and analysis the type of the theme to be given to the user using combination of many theme components and create a user taste customized theme without the user searching for it and also after the choosing of the theme of the web page the user can use his designing skills and design the web page with drag and drop tools of the website.

Key Words: WebScalar, Theme Builder, Theme Editor, ReactJS, MongoDB, Node JS, Express JS, MERN

1. INTRODUCTION

WebScalar is a web-based project used to adapt the new trending technology of MERN to build a user experience where the user can create their website without prior knowledge of coding and create their dream website. The user does not need to search for enormous themes rather should just answer the questionnaire asked in the application and then by the combination technique the application would suggest the best suitable theme for the user to look into and then the user can choose from the given themes and then edit that particular theme according to design idea that the user has.

The questionnaire consists of the taste of the user and also all the adequate style options and the purpose of the website to choose from and also it suggests the user to then get an output of the relevant theme for their desired website.

The web application is user-friendly, accurate and also very creative front end where the user gets the information suitable to the user. This application also has the website idea and strategy page where the user can get the ideas and strategies to build the website and the market trend that is suitable to develop the website.

2. PROPOSED WORK

Webscalar is developed using the latest technology which is adapted by many big companies that is MERN stack

development so comparatively it is more stable and uses supervised learning concept by the questionnaire session and helps the users to get the theme according to their choices and also enables the editor for the user to create the content according to their needs and also it also helps the developers who wants build their website faster and has a basic coding structure of the project and just want to edit the content management can also be supported by this application.

Table -1: Software Requirement

Operating System	Windows 7 or Higher
Language	Express JS, React JS
IDE	Visual Studio Code
Back End	Mongo DB and Node JS
Prototype	Adobe XD

Table -2: Hardware Requirement

Processor	Intel i3+
RAM	4Gb+
Hard Disk	20GB
CPU Clock Speed	2.9Ghz+
Graphic Card	NVidia GT 740M+
Video Memory	2GB+

3. IMPLEMENTATION

The implementation process starts by forming the structure of the backend of this application using node.js and creating the backend in a separate folder by initializing using the command node init and package.json file will be initialized and then by creating the app.js file where we can do the backend operations of the application. The IDE used in this application is Visual Studio Code since it is the one of the best environments to work with since it has integrated terminal to perform the operations and also extensions for react so hence this IDE was the best choice to implement the application. For the backend operations by using node.js we

also install mongoose and express for the backend and database connections by using terminal and the IDE used is Visual Studio Code where we can smoothly run the backend and frontend operations at the same time using the terminal.

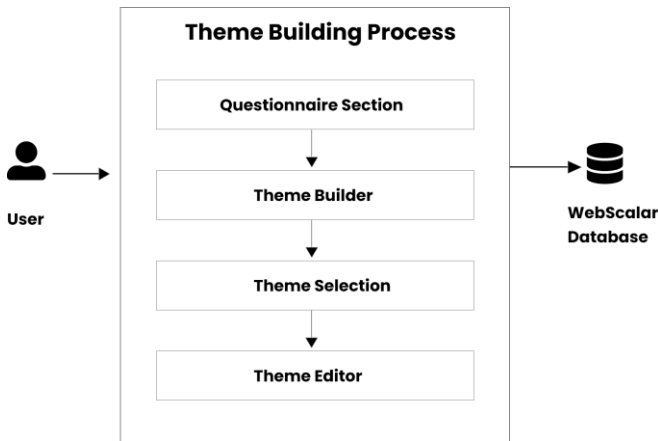


Fig-2 Architecture of Theme building Process

3.1 FRONT END DEVELOPMENT

The front-end development uses react.js which is a framework for combining JSX with HTML and the front-end development is started. We create react app using NodeJS and then create the routes in the app.js file and then start with dividing the front end into components which can be reused in the different pages where similar component is required. This makes the development faster and also simple and the component reusability is increased.

The next important part of the development of the front end is known as routing which are routes from each and every page in the development of this application. The router react-dom should be installed as a package to use the router in react. All the routes are defined in the app.js file of this application.

All the components which are defined in the components folder are also imported in separate pages by the name of the component and then are used in the function and are called using JSX code where they need to be placed.

3.2 BACK-END DEVELOPMENT

The back-end development is not in the same folder as the react application it is created in the separate folder where we develop all the operations related to the back-end of this application. All the requests and responses from the front end are handled using Axios to post it to the backend and then from the back-end it is posted to the data base of the application which is MongoDB in this application. The backend of this application is handled by Express JS which defines all the operations related to each component and page of the application.

The base logic used in this application is the tag matching process where the response from each user is taken into consideration and those response is stored in the string array and are then matched with tags assigned to each and every theme in the MongoDB collection and the best suited

theme which matches those string array tags would be displayed on a rank search basis and the user can then choose the desired theme and hence then can edit the theme with the theme editor.

MongoDB atlas is used in this application which is the cloud database and can be accessed by connecting the cluster to the application. The collections are similar to the tables of SQL and the records are the rows and columns of the SQL and records can be in any order in MongoDB.

The theme editor is integrated using the open-source editor known as the GrapeJS which can be used in the application by installing it using NodeJS and then the selected theme code which is in the database are imported to the theme editor and hence then the user can edit the content of that particular theme and then can export the code in a zip format.

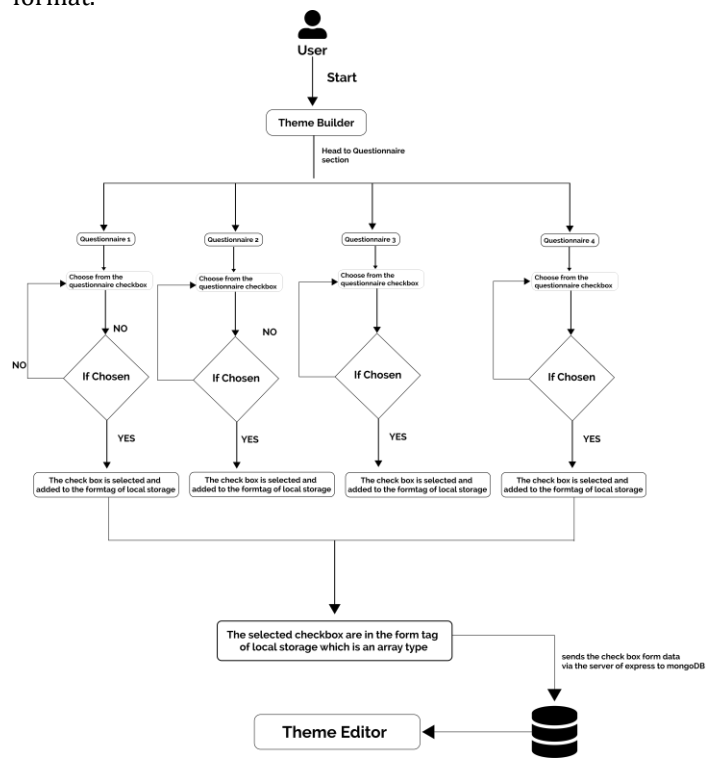


Fig-2 The flow chart of the theme building and editing process

4. RESULT ANALYSIS

The result analysis of this application is the accuracy of the tag matching and the filtration based on the tags that match the most choices of the user to the least matching tags that are then ranked while displaying the theme to the end-user. The end result which is the most matching theme is then chose by the user and then it is linked to the theme editor and then exported as HTML and CSS code in ZIP format which user can download.

5. CONCLUSION

The project “WEBSCALAR- Scaling your Own Website” aims on the website development for the user to build their own website without coding and just answering few of the questionnaire asked by the application and it develops the theme which is according to the taste of the user and accurately show them to the theme editor. This web application eases the user experience giving them to select the theme and then edit the content of the theme that has been selected and the user can easily export to the local storage and then upload the website to the hosting services.

REFERENCES

- [1] <https://www.typescriptlang.org/docs/handbook/jsx.html>
- [2] <https://www.javatpoint.com/react-jsx>
- [3] [https://expressjs.com/en/5x/api.h](https://expressjs.com/en/5x/api.html)
- [4] <https://docs.mongodb.com>
- [5] <https://nodejs.dev/learn>
- [6] <https://reactjs.org/docs/getting-started.html>