

Final Year Project
Report

WARF Engine

Submitted by

Aneesh Kumar - Roll No. 1900300130010
Anmol Ahuja - Roll No. 1900300130013
Ayush Bhandari - Roll No. 1900300100056
Tarun Bansal - Roll No. 1900300100256

In the partial fulfillment for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering



INDERPRASTHA ENGINEERING COLLEGE
GHAZIABAD, UTTAR PRADESH

Final Year Project

Certificate

Certified that **Aneesh Kumar, Anmol Ahuja, Ayush Bhandari, Tarun Bansal** has carried out the project work presented in this report entitled “**Warf Engine**” for the award of **Bachelor of Technology** from Inderprastha Engineering College, Ghaziabad, under my supervision. The report embodies results of original work and studies carried out by Student himself/herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

Prof. Shweta Chaku
Assistant Professor
Inderprastha Engineering College

Date: 21 December 2022

Declaration

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Aneesh Kumar - Roll No. 1900300130010
Anmol Ahuja - Roll No. 1900300130013
Ayush Bhandari - Roll No. 1900300100056
Tarun Bansal - Roll No. 1900300100256
Date: December 2022

Acknowledgement

We take this opportunity to thank our teachers and friends who helped us throughout the project.

First and foremost I would like to thank my guide for the project (**Mr. Shweta Chaku, Assistant Professor, Computer Science and Engineering**) for his valuable advice and time during the development of the project.

We would also like to thank **Dr. Vijai Singh (HoD, Computer Science Department)** for his constant support during the development of the project

Aneesh Kumar - Roll No. 1900300130010

Anmol Ahuja - Roll No. 1900300130013

Ayush Bhandari - Roll No. 1900300100056

Tarun Bansal - Roll No. 1900300100256

Abstract

Warf Engine is a drag-and-drop web builder. It can be used to create a website in just a few simple clicks or in a simple drag-and-drop manner.

This application can be used to generate a desktop app with Electron JS and can be deployed to Netlify.

This system will provide users with professional webpage templates they can select the templates based on their requirements and functionality. Here in this system, users can build the whole website by just selecting the content and images required for their webpage.

Users can design each and every web page on their websites according to their preferences by selecting the content and images.

Users can build custom websites easily using this application. User doesn't have to know any coding languages to build web pages.

Contents

1	Introduction	1
1.1	Problem Definition	1
1.2	Background about the Project Idea	2
1.3	Objective	2
1.4	Feasibility Study, need and Significance	2
1.4.1	Executive Summary	2
1.4.2	Market Demand	2
1.4.3	Technical Feasibility	3
1.4.4	Operational Feasibility	3
1.4.5	Legal and Regulatory Compliance	3
1.4.6	Schedule Feasibility	3
1.4.7	Conclusion	3
1.5	Technical Specification	3
1.5.1	Harware	3
1.5.2	Software	4
2	Literature Review	5
3	Proposed System	6
4	Software Requirement Analysis	7
4.1	Functional Requirements	7
4.2	Major Modules and their Functionalities	8

5	System Analysis and Design	10
5.1	DFD of the Project	10
5.2	Sequence Diagram	11
5.3	Activity Diagram	12
5.4	Gantt Chart	13
6	Implementation	14
6.1	Used Algorithms/Approaches for Project	14
6.2	Implementations of Modules	15
7	Results/Outputs and Testing	19
8	References	22
9	Appendices	23
9.1	Steps to run/execute the project	23

Chapter 1

Introduction

1.1 Problem Definition

There is a number of coding languages available in order to build web pages. So, there is a need to know anyone coding language to build a web page. This innovative software application allows users to build web pages without knowing any coding language. It is specifically designed for internal use by companies. This software helps to build/design graphical Web pages. This software is embedded in a website for professional use and will be available for customers to tailor web pages as per their needs. This system helps the user to build a web page with an effective graphical user interface. The user doesn't have to work specifically for GUI. The system will display various web page template user can select any template according to his preference. With the help of this system, users can work on the main functionality required on their web page. This system will save the time of the user and will help the user to concentrate on the main functionality required in their webpage. This system will provide users with professional webpage templates they can select the templates based on their requirements and functionality. Here in this system, users can build a whole website by just selecting the content and images required in their webpage. Users can design each and every web page on their websites according to their preferences by selecting the content and images. Users can even specify the position of the content to be placed. When the user logs in to the system, the system will display various templates to the user. The user has to select the template. The system will display various pages of that template. By clicking on a particular page, he must specify the content and images that need to be placed on the webpage. Users can place the content and can view the template simultaneously. Once the user clicks on the submit button system will generate the website. The

system will send the code and .rar file to the user's email ID. Users can build custom websites easily using this application. User doesn't have to know any coding languages to build web pages.

1.2 Background about the Project Idea

Nowadays everyone needs a webpage to make their online presence visible to the online world. This solution lets anyone create their own webpage and make their page online very easily just by using the drag-and-drop feature. It will let anyone create their website irrespective of their background whether coding or non-coding background.

1.3 Objective

To solve the problem statement, we have come up with this solution, Warf Engine. A web platform dedicated for people having zero knowledge of coding to build a fully-fledged ecommerce website for their use. Here people (layman) can build their website using new Modern day technologies which are faster than the available ones in the market. Some of the technology which we are implementing are Reactjs, Nodejs Etc

1.4 Feasibility Study, need and Significance

1.4.1 Executive Summary

The purpose of this feasibility study is to evaluate the potential for developing and launching a website builder for small businesses and individuals. The proposed website builder will provide an easy-to-use platform for creating and managing websites, with a variety of customizable templates and design elements.

1.4.2 Market Demand

There is a high demand for website builders in the current market, as more and more small businesses and individuals are looking to establish a strong online presence. According to a recent survey, the website builder market is expected to grow at a CAGR of 14.5% during the forecast period 2020-2025. The study also indicates that the small business segment is expected to hold the largest market share by 2023

1.4.3 Technical Feasibility

The proposed website builder will be developed using the latest web development technologies, such as HTML, CSS, JavaScript, and MERN stack. The development team has the necessary technical expertise and experience to design and implement the website builder.

1.4.4 Operational Feasibility

The website builder will be operated by a dedicated team of web developers and designers, who will be responsible for maintaining and updating the platform. The team will also provide customer support and troubleshoot any technical issues that may arise.

1.4.5 Legal and Regulatory Compliance

The website builder will comply with all relevant laws and regulations, including data privacy and security laws. The development team will work with legal counsel to ensure compliance with all applicable laws and regulations.

1.4.6 Schedule Feasibility

The development of the website builder is expected to take 4 months, with an additional 1 months for testing and deployment. The website builder will be launched in the market in 6 months.

1.4.7 Conclusion

Based on the research and analysis conducted in this feasibility study, the proposed website builder for small businesses and individuals appears to be a viable and profitable project. The high demand for website builders in the market, the technical feasibility of the project, the availability of necessary resources, and the projected financial returns all indicate that the website builder is worth pursuing further.

1.5 Technical Specification

1.5.1 Hardware

- Processor: Intel i5 or higher

- Space On HDD: 5GB
- RAM: 4GB

1.5.2 Software

- Operating System: Windows 10
- Coding Language: Javascript
- IDE: VS Code

Chapter 2

Literature Review

For our project work we have reviewed the following research papers:

1. Hau Tran, “Developing a social platform based on MERN stack” The main goal of this research paper was to study the basic components of the highly popular MERN stack and to create scalable components based on MERN stack technology.
2. Sourabh Mahadev Malewade, Archana Ekbote, “Performance Optimization using MERN stack on Web Application, Proceedings of International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181,2021. By the help of this research paper, we implemented performance optimization measures on each created component and basic application.
3. Rohit Tamrakar, Niraj Wani, “Design and Development of CHATBOT: A Review,” This paper reviews the technique, terminology, and different platforms used to design and develop the CHATBOT.
4. Zujaj Misbah Khan, Kamran Rasheed, Muhammad Imran, “ WordPress: A Versatile Tool for Managing Contents,” This paper recommends developing a customized website using WordPress and its advantages and disadvantages. This research paper also help us to identify various requirements of a user that wants to create his/her webpage
5. Mohammed Farik, Nilesh A. Lal, Shailendra Prasad, “Review Of Authentication Methods,” This Paper describes the importance and types of Authentication mechanisms and how to implement those authentication mechanisms to enhance security and reduce vulnerabilities.

Chapter 3

Proposed System

1. The project is based on the latest web technology, i.e. MERN stack (MongoDB, Express js, React js, Node js).
2. Project will be held on 2 different servers, one being the front end and the other being the backend server. Both servers will run simultaneously and work for hand in hand.
3. The back-end server is based on Node js and the front-end server is based on React js.
4. The Database used is MongoDB, which is non-relational, secure, and better to use.
5. All the frameworks, and libraries used are Open -Source and free to be optimized and used. The editor used to code is VS Code and for management of files and updates, GitHub is used extensively.
6. At the end, Heroku is used to host the project online and provide everyone with hassle-free service.
7. All the code is done from scratch and no code is copied from any online resources, for better handling, of course, the Material framework is used in the Front End part.

Chapter 4

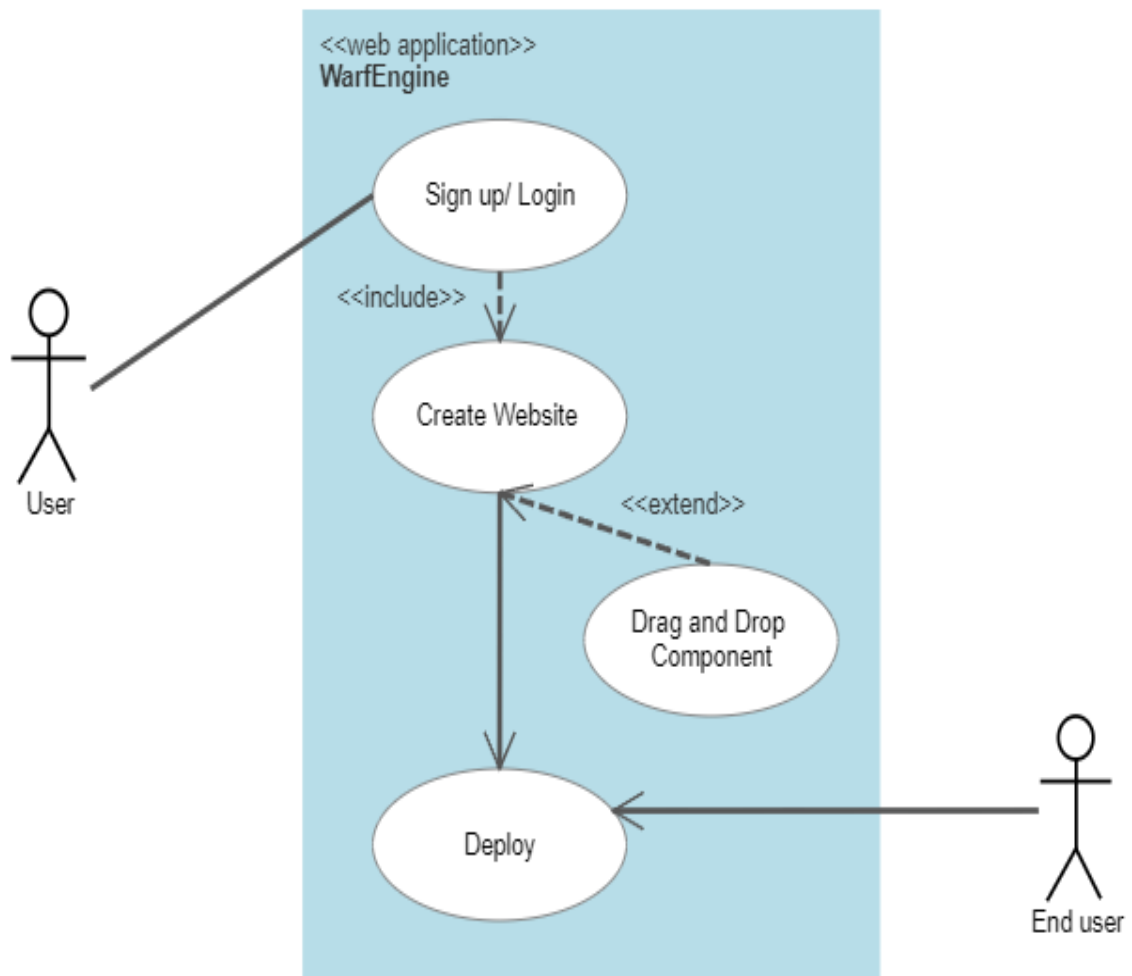
Software Requirement Analysis

4.1 Functional Requirements

The functional requirement of the project includes the:

- Git: The git is installed for pushing the code and it also helps to control the version.
- NodeJS: The project is entirely based on NodeJS. Hence installing a node is one of the key steps to running the project.
- Packages And Libraries
- MongoDB Compass and MongoDB Community Server: These are required to support and run the database servers.

Use Case Diagram



4.2 Major Modules and their Functionalities

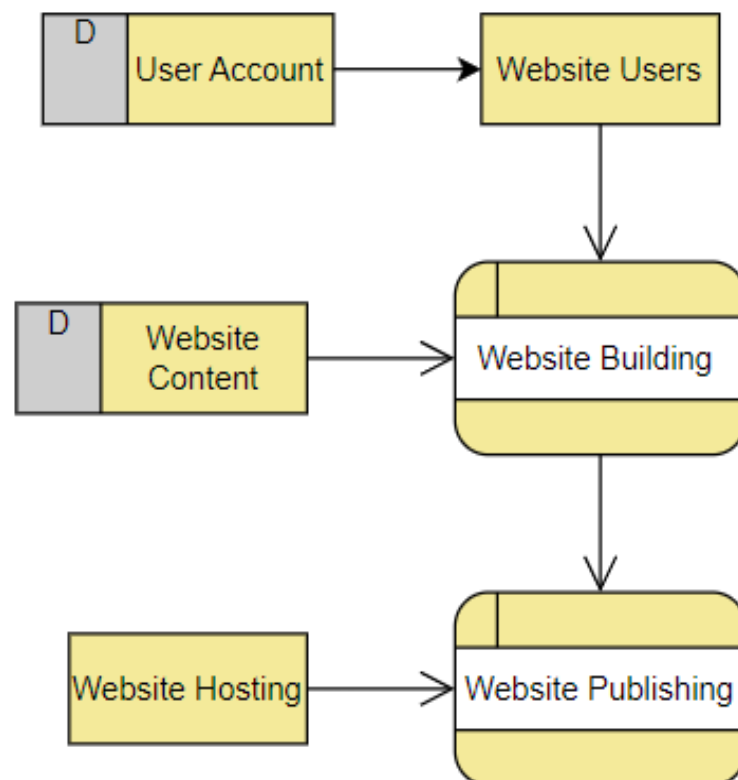
- **Template module:** This module allows users to select from a variety of pre-designed templates, which serve as the foundation for their website. Users can choose from a wide range of templates that are optimized for different types of websites, such as business websites, personal blogs, and e-commerce sites.
- **Customization module:** Once a template is selected, the customization module allows users to modify the template to suit their specific needs. This module typically includes tools for editing the layout, color scheme, and typography of the website, as well as the ability to add and edit content such as text, images, and videos.

- **Content management module:** This module allows users to add, edit, and organize the content of their website. It typically includes a user-friendly interface for creating and editing pages, as well as tools for organizing and managing the site's navigation and structure.
- **SEO module:** Search Engine Optimization (SEO) module helps the website to rank better on search engine. This module allows users to optimize their website for search engines by creating meta tags, generating sitemaps, and other SEO-related tasks
- **Preview and publishing module:** This module allows users to pre-view their website and make any final adjustments before publishing it. It also includes the functionality to publish the website on the internet.
- **E-commerce module:** This module allows users to build an online store, where they can sell products or services. This module includes functionality such as product management, inventory management, and payment processing.
- **Analytics and statistics module:** This module allows users to track and monitor the performance of their website, including the number of visitors, pages viewed, and other metrics.

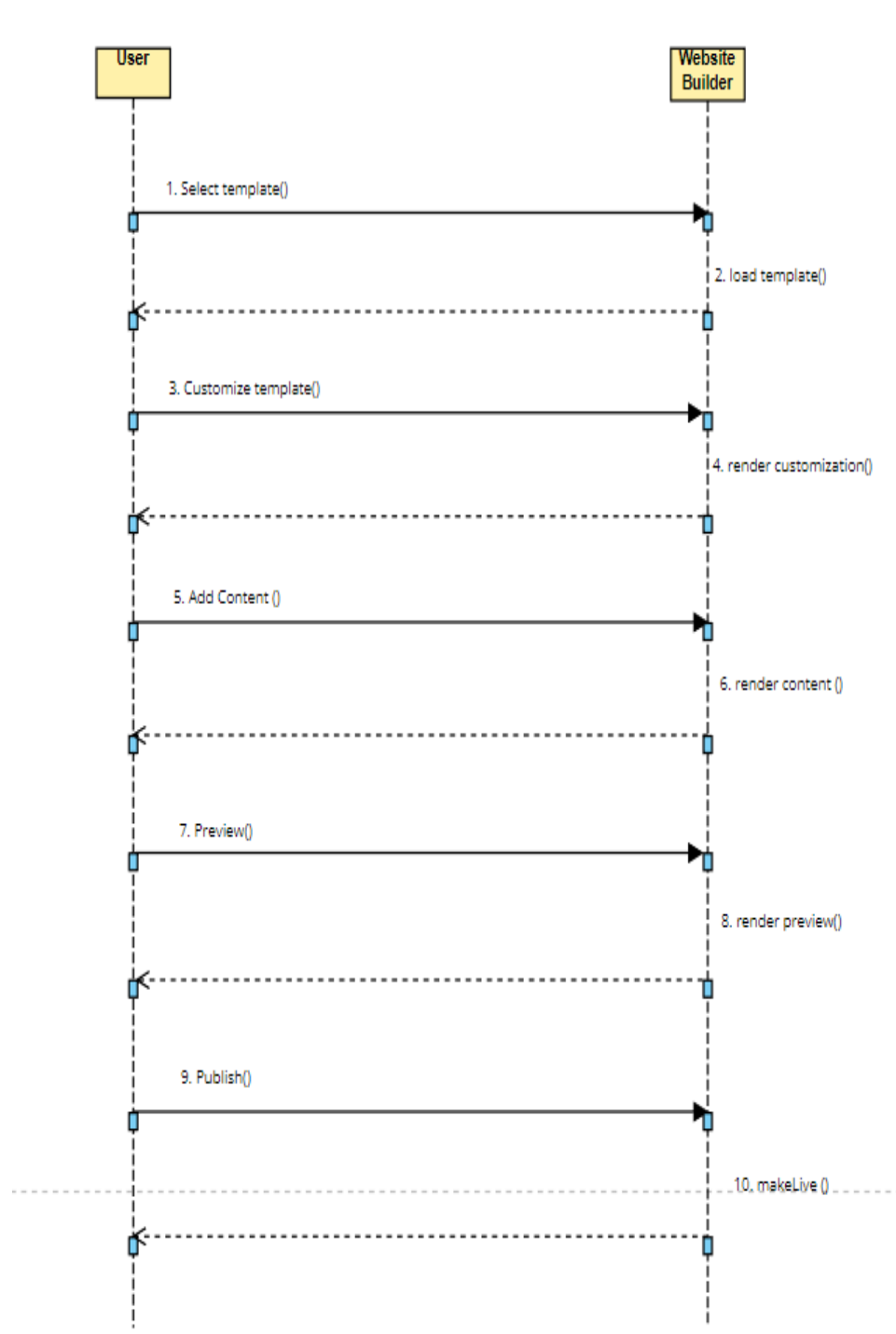
Chapter 5

System Analysis and Design

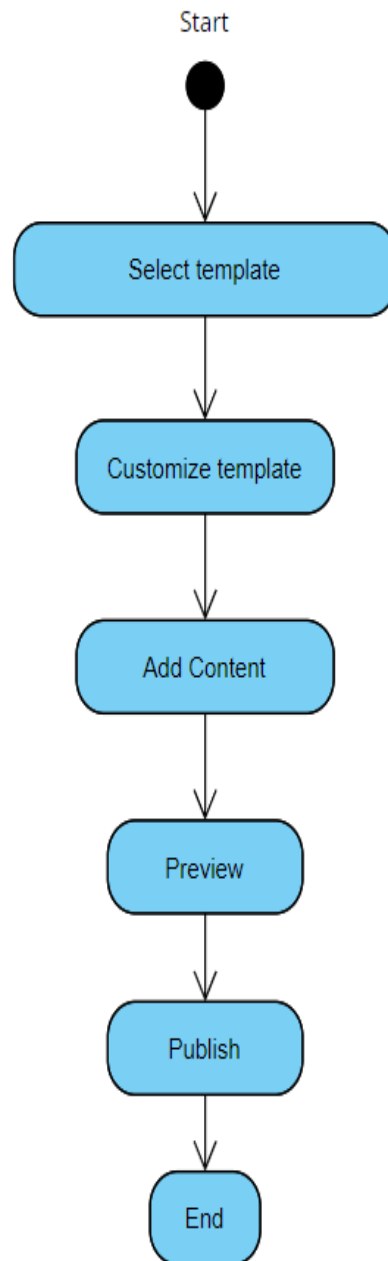
5.1 DFD of the Project



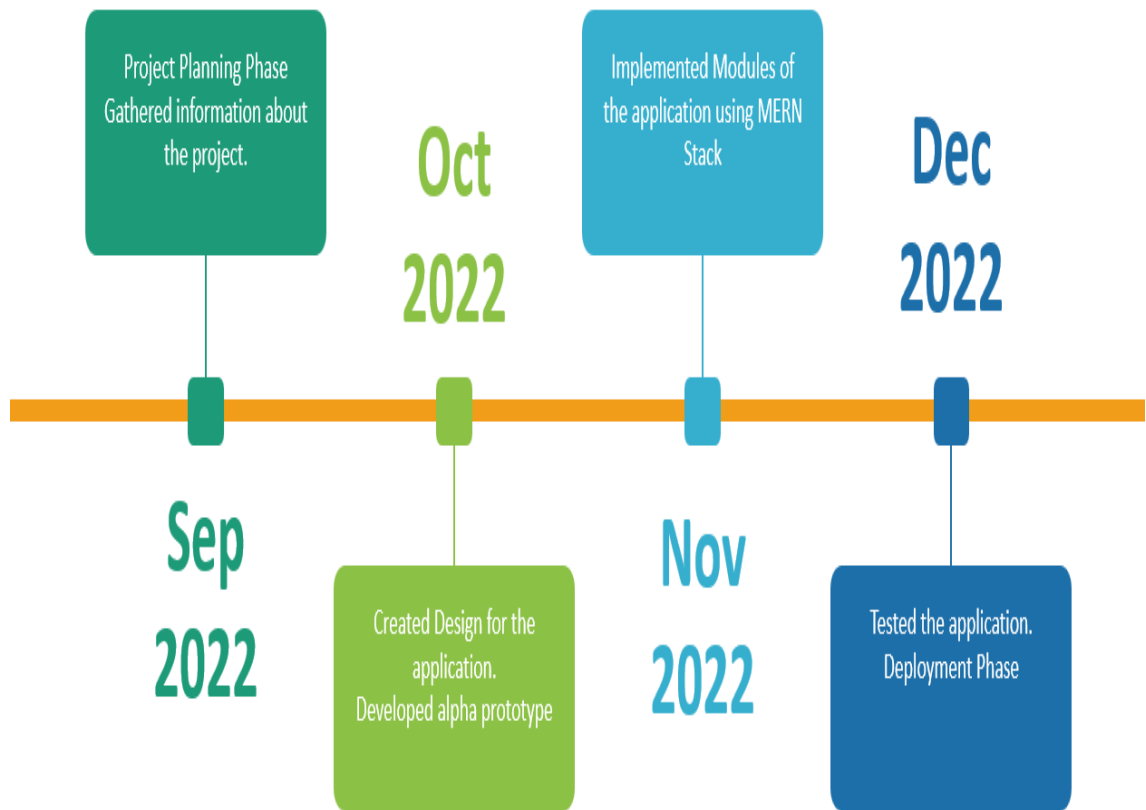
5.2 Sequence Diagram



5.3 Activity Diagram



5.4 Gantt Chart



Chapter 6

Implementation

6.1 Used Algorithms/Approaches for Project

- **React JS:** React is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality. It makes the web application faster and smoother , it provides hot reloading and multiple other features to make the platform better and faster .
- **Node.js:** Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It is an open-source, cross-platform, back-end Javascript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for serverside scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server-side and client-side scripts. Node js is used on the backend side of the code for the backend server with
- **Express.js:** ExpressJs Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web

and mobile applications. Express.js, or simply Express, is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. Express is the back-end component of popular development stacks like the MEAN, MERN, MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library.

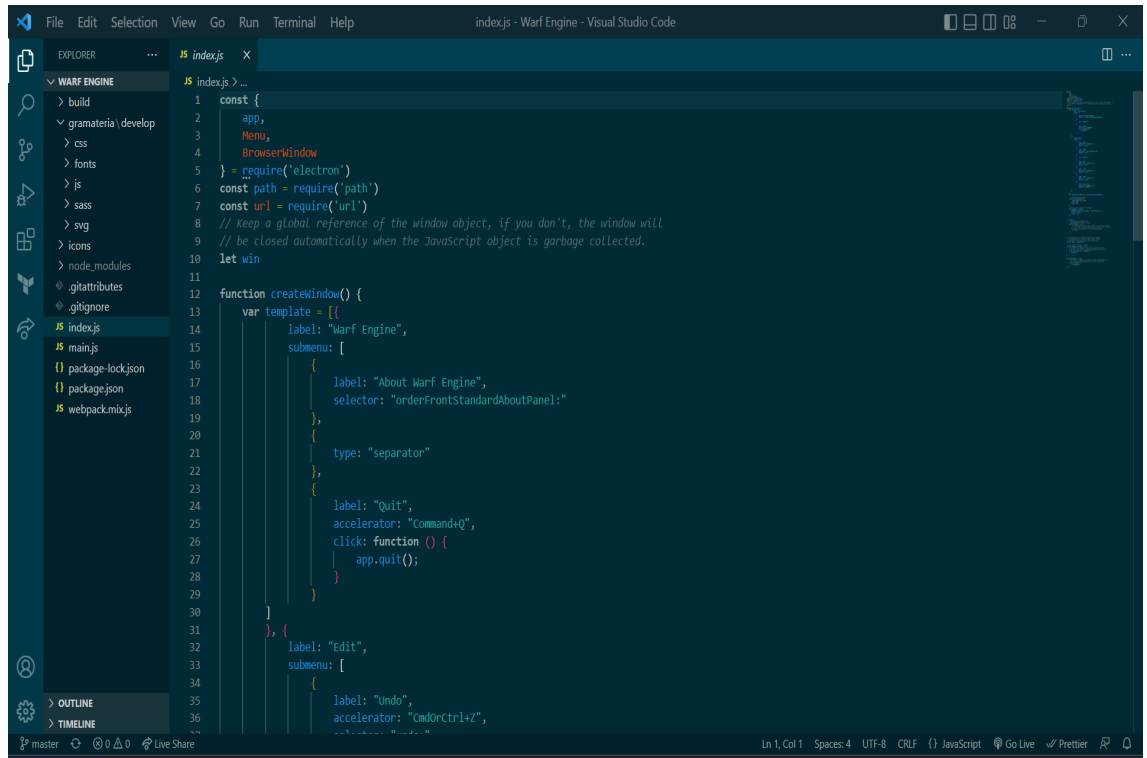
- **MongoDB:** MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server-Side Public License. It is non-relational, which makes it better than SQL and is used in the popular development stack such as MERN, MEAN.

6.2 Implementations of Modules

We have built our project on the latest technologies like Electron.js, React.js, Node.js etc.

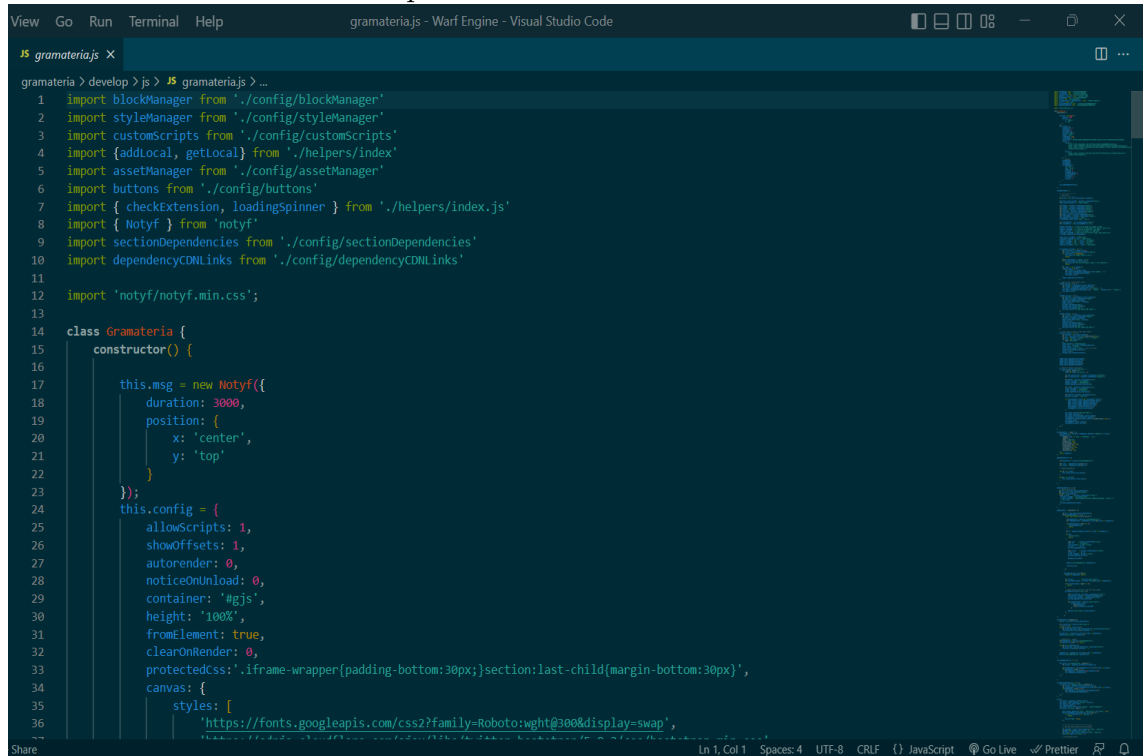
The various modules are :-

- **Electron :** Electron is a framework for building cross-platform desktop applications using web technologies such as JavaScript, HTML, and CSS. It allows developers to create apps that run on Windows, macOS, and Linux with a single codebase. Electron is built on top of Chromium, the open-source version of Google Chrome, and Node.js, which provides access to the file system and other native APIs. This means that developers can use the same technologies and libraries they are already familiar with to build desktop apps.



- **asar** : ASAR (Atom Shell Archive) is a file format and Node.js library used to package and distribute applications built on the Electron framework. It is similar to a ZIP file, but is optimized for use with Electron, as it can directly access individual files within the archive without the need to extract the entire contents. This allows developers to bundle their Electron application and its dependencies into a single, easily distributable file.
- **File-Saver** : File-Saver is a JavaScript library that allows web applications to save files on the client-side. It provides a simple API for creating and saving files in various formats, such as text, JSON, and binary data. File-Saver uses the HTML5 Blob object to create the file, which is then saved to the user's device using the `saveAs()` function. This library is widely supported by modern web browsers and does not require any additional dependencies.
- **jsZip** : JSZip is a JavaScript library that allows you to create, read and write Zip files in the browser. It provides an easy-to-use API for creating and manipulating Zip files, including the ability to add, remove and extract files. The library uses the HTML5 Blob object to create the Zip file and can also read files from the file system using

the FileReader API. It also support the ability to generate a base64 encoded representation of the zip file, which can be sent to the server for download or further manipulation.

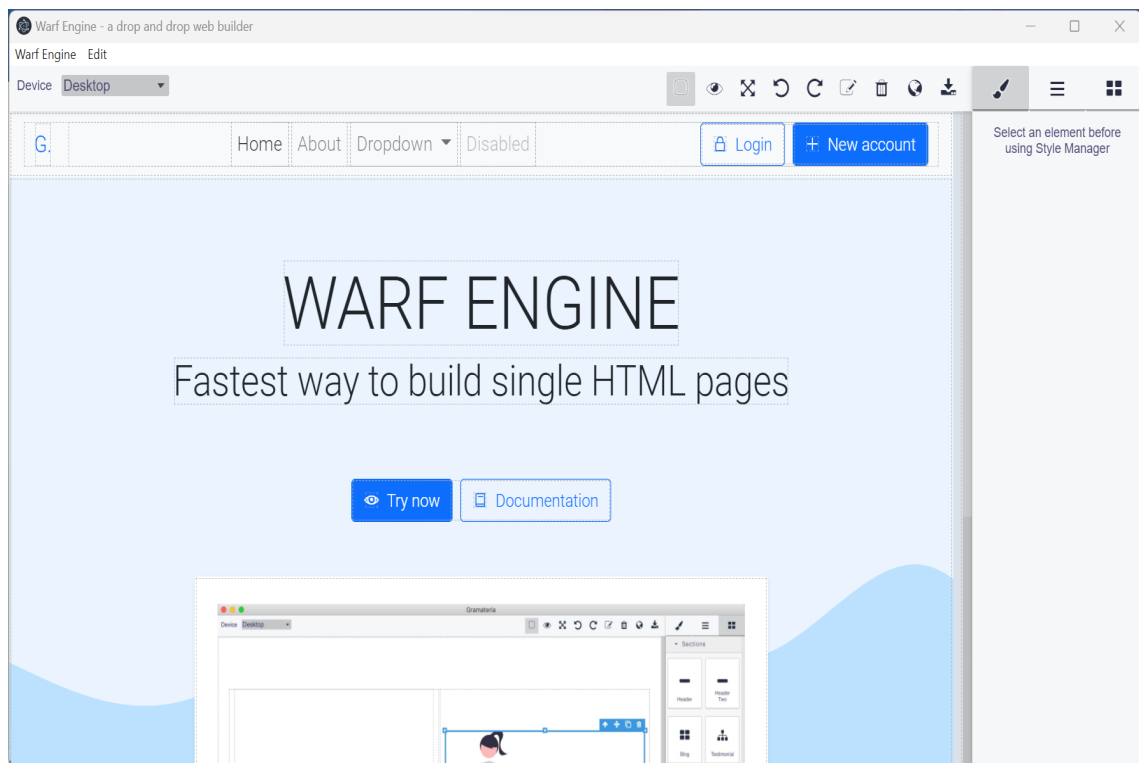


```
View Go Run Terminal Help gramateria.js - Warf Engine - Visual Studio Code
JS gramateria.js x
gramateria > develop > js > JS gramateria.js > ...
1 import blockManager from './config/blockManager'
2 import styleManager from './config/styleManager'
3 import customScripts from './config/customScripts'
4 import {addLocal, getLocal} from './helpers/index'
5 import assetManager from './config/assetManager'
6 import buttons from './config/buttons'
7 import { checkExtension, loadingSpinner } from './helpers/index.js'
8 import { Notyf } from 'notyf'
9 import sectionDependencies from './config/sectionDependencies'
10 import dependencyCDNLinks from './config/dependencyCDNLinks'
11
12 import 'notyf/notyf.min.css';
13
14 class Gramateria {
15   constructor() {
16
17     this.msg = new Notyf({
18       duration: 3000,
19       position: {
20         x: 'center',
21         y: 'top'
22       }
23     });
24     this.config = {
25       allowScripts: 1,
26       showOffsets: 1,
27       autorender: 0,
28       noticeOnUnload: 0,
29       container: '#gjs',
30       height: '100%',
31       fromElement: true,
32       clearOnRender: 0,
33       protectedCss: '.iframe-wrapper(padding-bottom:30px)section:last-child(margin-bottom:30px)',
34       canvas: {
35         styles: [
36           'https://fonts.googleapis.com/css2?family=Roboto:wght@300&display=swap',
37         ]
38       }
39     };
40   }
41 }
```

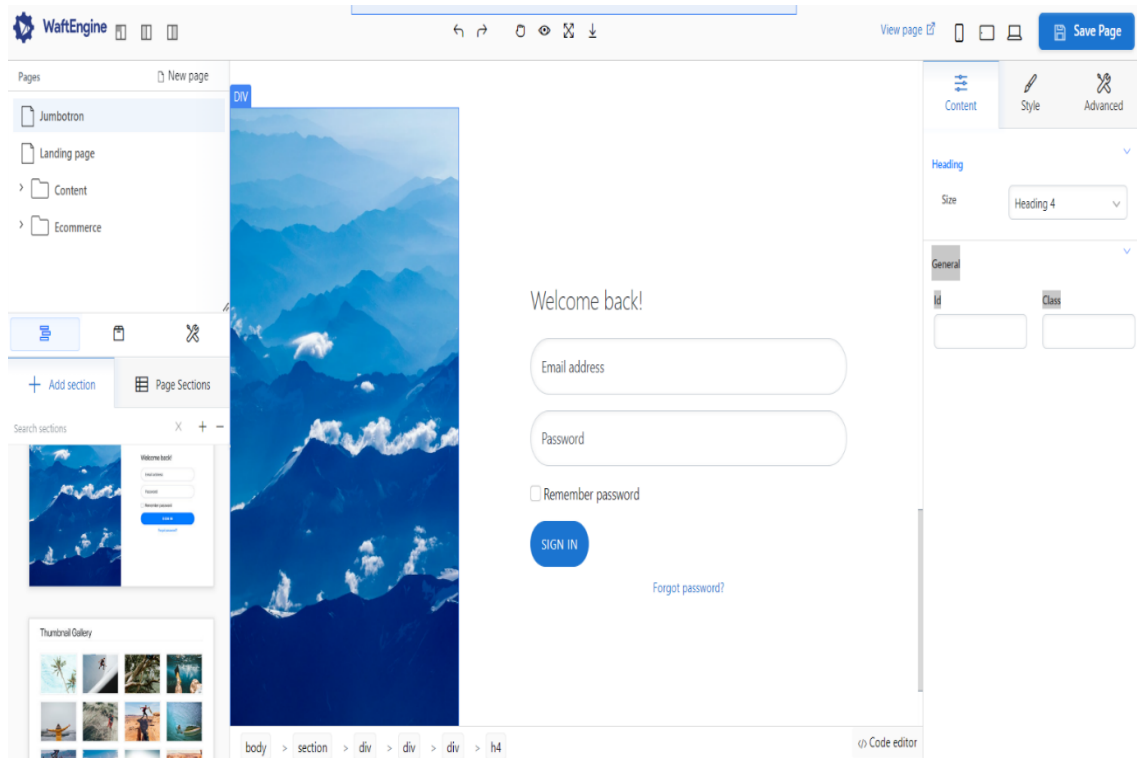
- **notyf** : Notyf is a JavaScript library that allows developers to easily create and manage notifications in web applications. It provides an intuitive and easy-to-use API for creating and customizing different types of notifications, such as success, error, and warning messages. Notyf is highly customizable and allows developers to change the appearance and behavior of notifications to match their application's design and user experience.

Chapter 7

Results/Outputs and Testing



Home Page



Login/SignUp Component

Pages

New page

+ Add section

Page Sections

Search sections

X + -

Thumbnail Gallery

Video Header

body > section > div > div > div > a > img

Code editor

Content

Style

Advanced

Image

Image

<https://source.unsplash.com>

Set Image

Width

Height

Alt

General

Id

Class

img-fluid img-thumt

Media Component

21

Chapter 8

References

- Dominik Ertl TU Wien, Harald Krapfenbauer, “A case study of developing an IDE for embedded software using open source,”
- Rohit Tamrakar, Niraj Wani, “Design and Development of CHATBOT: A Review,” This paper reviews the technique, terminology, and different platforms used to design and develop the CHATBOT.
- Mohammed Farik, Nilesch A. Lal, Shailendra Prasad, “Review Of Authentication Methods,” This Paper describes the importance and types of Authentication mechanism
- Hau Tran, “Developing a social platform based on MERN stack” The main goal of this research paper was to study the basic components of the highly popular MERN stack, 2021
- Sourabh Mahadev Malewade, Archana Ekbote, “Performance Optimization using MERN stack on Web Application, Proceedings of International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181,2021

Chapter 9

Appendices

9.1 Steps to run/execute the project

- Download and install all the required libraries and packages
- Navigate to the Downloaded repository and enter: `npm install` and `npm update`
- Run the development server.
- To deploy locally navigate to the project directory in cmd.
- Now there are 2 servers to be run, the backend one and front end one.
Run the following command to build the project and launch the server:
- `npm start` (in the main directory)
- `cd client/`
- `npm start` (for front end)

The server will now be running and connected to MongoDB. Navigate to **localhost:3000** to view the application. In the above way you can simply run the app on your local device.