Static Site Generator

DESIGN DOCUMENT

Team 22
Client: Amanda Dropinski
Advisers: Samik Basu
Team Members:
Andrei Baechle
Joshua Flory
Meet Patel
Bennett Ray
Luke Shay
Ty Trinh
Team Email: sdmay21-22@iastate.edu
Team Website: https://sdmay21-22.sd.ece.iastate.edu/

Revised: 09.18.2020 v1

Executive Summary

Development Standards & Practices Used

Software Practices

- Kanban Development
- Unit Testing
- Branch-Review-Merge Workflow

Summary of Requirements

List all requirements as bullet points in brief.

- Sites generated must be static
- Users will be given base templates to edit
- Sites will have different theme options
- Some content will be integrated with Buildertrend (login, lead generation, contact)
- Admin panel to edit templates
- Basic api to save and load edited templates
- Users will be able to view saved sites by visiting the site domain

Applicable Courses from Iowa State University Curriculum

- S E 319
- COM S 309
- S E 329
- S E 339

New Skills/Knowledge acquired that was not taught in courses

- React
- TinaCMS

- Express
- TypeScript

Table of Contents

List of figures/tables/symbols/definitions

Task Decomposition Tables: For each subtask associated with a task, lists the number of the subtask, name, and which subtasks need to be completed before the task.

Subtasks Lists: Lists the subtasks for each semester. Each subtasks has a start date and expected duration in days indicating how long the subtask will take to complete.

Gantt Charts: The Gantt charts for Fall 2020 and Spring 2021 generated from the subtask lists.

Required Time Table: For each task, lists the task number, duration in days, and the expected number of hours that each task will take to complete.

1 Introduction

1.1 Acknowledgement

Amanda Dropiskini: Project Manager

Amanda will be our project manager for this project. She will be assisting with organizing check ins and understanding the project purpose and goals.

Thomas Judge: Application Developer

Thomas is a full-time application developer from Buildertrend who will be providing technical assistance for us during project development.

1.2 Problem and Project Statement

Problem Statement

Buildertrend wants to provide an easier process for clients to build and maintain their company websites.

Solution Approach

The Static Site Content Management system will allow clients to easily set up and maintain a company website via an administrative panel and will increase usage of this feature, as well as provide Buildertrend's clients a custom website building experience so they can stand out in their respective markets.

1.3 Operational Environment

There is no physical component to this project, therefore the operational environment is not a concern for this project.

1.4 Requirements

Functional Requirements

- Admin panel to edit basic JSON file that drives the site
- Generated sites must be static

- Users will be able to use generic templates for their sites including Home, Blank, Gallery, Showcase, Blog, Login, Contact
- Users will be able to choose from at least 2 themes
- Client sites will be integrated with Buildertrend login form
- Basic API to save and retrieve JSON data for generated sites

Technical Requirements

- Frontend application will a created using React with TypeScript
- TinaCMS will be used to build the Content Management System
- All rendering will be handled on the client side
- Node will be used to build the file server

1.5 Intended Users and Uses

The intended users are Buildertrend clients who run businesses and need a website to serve their customers. They will be able to use our site to create their own websites for their clients.

1.6 Assumptions and Limitations

Assumptions

- We only need to support english because all of their clients are in US
- There will be no unexpected costs since we have no budget
- All planned Buildertrend integrations will have no alternative integrations
- Our NodeJS server does not need to be very in depth because it will be replaced

Limitations

- This semester is shorter than previous semesters, so we have fewer weeks to complete our milestones
- A full time team will be taking over the project at the end of May, so we are expected to only use frameworks and libraries that Buildertrend uses
- During this semester, we will not be performing tests for high volume traffic on client websites or the API endpoints
- The generated sites must be static

1.7 Expected End Product and Deliverables

End Product: Static Site Content Management System

Deliverable 1: File Server

Delivery Date: 10.13.2020

This deliverable is for the basic file server that will store and return the edited files for the sites created by the users. This component will be relatively simple since it will be replaced after the project is given to Buildertrend.

Deliverable 2: Static Site CMS Web Application.

Delivery Date: 04.13.2020

This deliverable is for the frontend React application that users will use to create their websites. Buildertrend's customers will be able to use the admin template to edit provided templates to build their sites. Sample templates will include, home, gallery, login, blank, showcase, blog and contact. Each template will have sample content that users can edit to fit their site. Some features will be integrated with the Buildertrend site, like the login and templates with lead generation. Admins will also have the options to choose different themes for their sites. Once the admin has completed the site, their customers will be able to visit the domain for the site to view it.

2 Project Plan

2.1 Task Decomposition

Task 1: File Server

Subtask Number	Task Name	Predecessor
1	Setup file Server	
2	Create sample text file for saving to file system	
3	Post endpoint for file server	1
4	Get endpoint for file server	1
5	Create Basic React with TypeScript Application	
6	Page to save a file on the file server	3, 5
7	Page to load a file from the file server	4, 5
8	Get endpoint to retrieve a list of files	2
9	Page to retrieve a list of files from file	5, 8

	server	
10	Add option to download files	6
11	Implement versioning system	3,4
12	Allow retrieval based on file versions	11

Task 2: Admin Panel

Subtask Number	Task Name	Predecessor
1	Create Admin Page	
2	Add TinaCMS to React Application	
3	Admin: Edit and save template content	1,2
4	User: view modified templates	1,2

Task 3: Template Creation

Subtask Number	Task Name	Predecessor
1	Home Template	
2	Blank Template	
3	Login Template	
4	Integrate Buildertrend login with Login template	3

Task 4: Finish templates and 2 Themes

Subtask Number	Task Name	Predecessor
1	Gallery Template	
2	Showcase Template	
3	Blog Template	
4	Contact Template	

5	Gallery - Document uploads	1
6	Theme 1	
7	Theme 2	

Task 5: 1 Theme, Content, Google Maps Integration

Subtask Number	Task Name	Predecessor
1	Theme 3	
2	About	
3	Floor plan	
4	Testimonials	
5	Google Maps Integration	

Task 6: Finish content, Reviews Integration, SEO

Subtask Number	Task Name	Predecessor
1	Services	
2	Privacy	
3	Policy Process	
4	BT Reviews Integration	
5	Implement versioning for editor	
6	SEO for templates	

2.2 Risks And Risk Management/Mitigation

Task 1: File Server

1. File permissions

   Risk Factor: 0.1

Explanation: As of right now, we are using our own file server which will have the risk of having other users access files they should not have access to. Since our file system server is temporary, this has little risk to our implementation for the frontend.

2.

Task 2: Admin Panel

1. TinaCMS does not fulfill requirements for our content management system.

   Risk Factor: 0.2

   Explanation: This library was chosen by Buildertrend who already tested the library out and determined it was the best CMS for the site. However, it is possible that as the site features become more advanced, TinaCMS will prevent us from implementing features as originally anticipated.

Task 3: Template Creation

1. Templates mockups may change as project progresses

   Risk Factor: 0.1

   Explanation: As templates are completed, Buildertrend may revise what templates need to look like. This is not a big risk, but this may be in any capacity, so we should be prepared to change.

2. Cross-Browser Compatibility

   Risk Factor: 0.1

   Explanation: Different web browsers interpret web pages in different ways, so some browsers may not support certain stylings or functionality. In order to keep these pages consistent, we will have to be aware of these differences.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Milestone 1: Basic File Server will be completed by 10.13.2020

Evaluation Criteria: Users will be able to run both the file server and frontend application and will be able to save a JSON body to the file system and download a file.

Milestone 2: Admin Panel will be completed by 11.10.2020

Evaluation Criteria: Users will be able to use the admin panel to edit and save a template file. The user will then be able to view the edited template.

Milestone 3: At least 3 templates and one integration will be completed by 12.08.2020

Evaluation Criteria: Users will be able to build sites by editing the templates. One of the templates will include integration

2.4 Project Timeline/Schedule

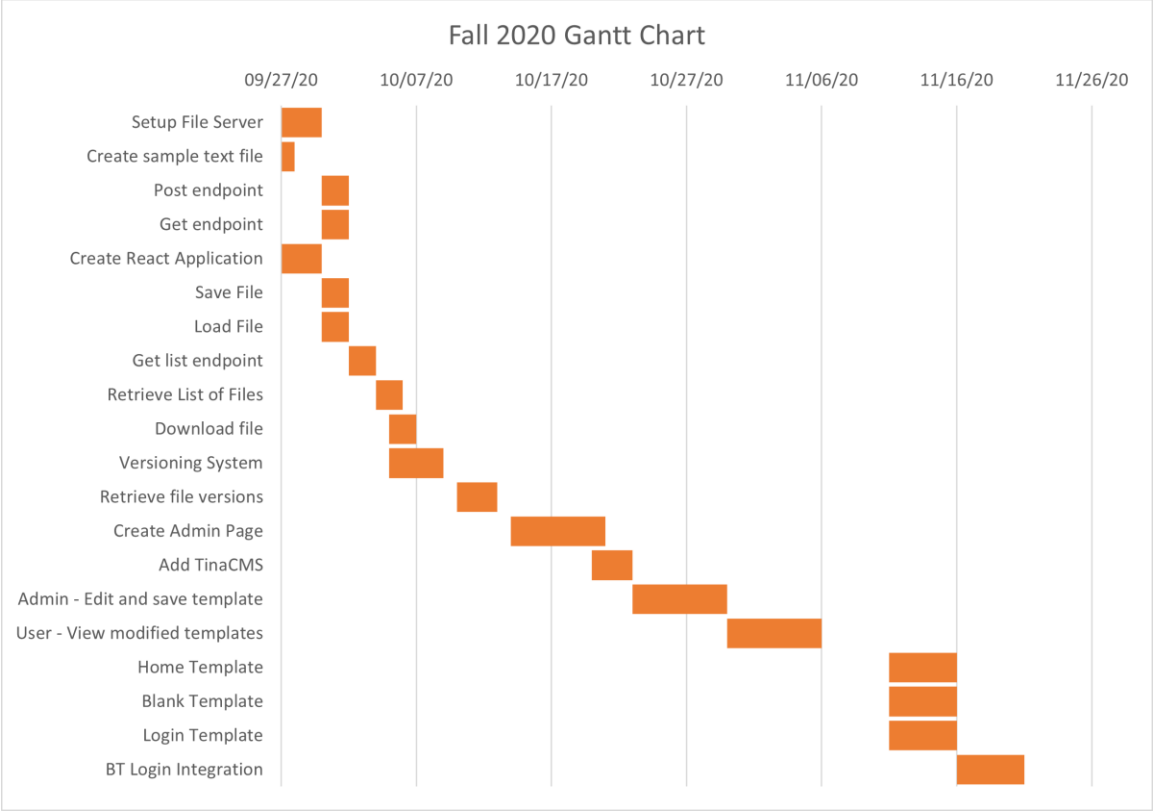
We have already been given the first 3 milestones for the first semester of our project. We also have deadlines for each milestone. Based on these milestones and deadlines, we created tasks and estimated the number of days each task will take. The table below shows each task with their start date and the estimated number of days that each task will take. This table was used to create the Gantt chart.

Subtask Table

Semester 1		
Task	Start Date	Duration
Setup File Server	09/27/20	3
Create sample text file	09/27/20	1
Post endpoint	09/30/20	2
Get endpoint	09/30/20	2
Create React Application	09/27/20	3
Save File	09/30/20	2
Load File	09/30/20	2
Get list endpoint	10/02/20	2
Retrieve List of Files	10/04/20	2
Download file	10/05/20	2
Versioning System	10/05/20	4
Retrieve file versions	10/10/20	3
Create Admin Page	10/14/20	7
Add TinaCMS	10/20/20	3
Admin - Edit and save template	10/23/20	7
User - View modified templates	10/30/20	7
Home Template	11/11/20	5
Blank Template	11/11/20	5
Login Template	11/11/20	5
BT Login Integration	11/16/20	5

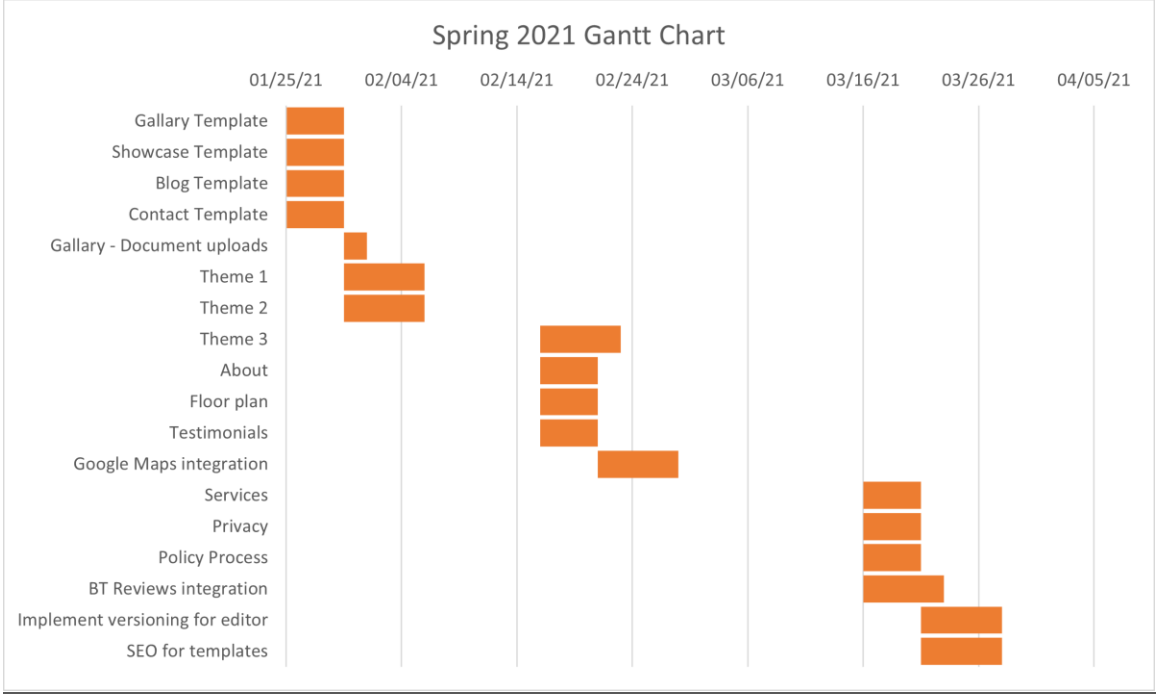
Semester 2		
Task	Start Date	Duration
Gallery Template	01/25/21	5
Showcase Template	01/25/21	5
Blog Template	01/25/21	5
Contact Template	01/25/21	5
Gallery - Document uploads	01/30/21	2
Theme 1	01/30/21	7
Theme 2	01/30/21	7
Theme 3	02/16/21	7
About	02/16/21	5
Floor plan	02/16/21	5
Testimonials	02/16/21	5
Google Maps integration	02/21/21	7
Services	03/16/21	5
Privacy	03/16/21	5
Policy Process	03/16/21	5
BT Reviews integration	03/16/21	7
Implement versioning for editor	03/21/21	7
SEO for templates	03/21/21	7

Semester 1



If we are able to complete all tasks in the expected time, we should be able to complete the first 3 milestones before the end of the semester.

Semester 2



This chart is less complete than the Fall 2020 chart. Since we do not know how much progress we will have made by the end of November, there are gaps between the end of one task and the start of the next to anticipate any new subtasks that might be added for each task.

2.5 Project Tracking Procedures

Github Project

Our team will be using the project feature in Github to track the progress of the project. Each task will be represented as an issue on the project board, and each issue will be assigned to a task member.

Project Board Columns

To-Do: Cards in this column have not been started but are ready to be picked up by a developer

In Progress: Column for cards that have been started but no pull request has been created

Code Review: Column for cards that have pull requests that need to be reviewed

Testing: Cards whose functionality has been merged to master but a developer still needs to verify the functionality still works.

Done: Card works in master and is complete.

2.6 Personnel Effort Requirements

Time Required for Each Task

To determine the number of hours required for each task, multiplied the duration from the table in 2.4 by 2, since we assumed that we would spend, on average, 2 hours a day working on each subtask.

Required Time Table

Task	Duration (Days)	Hours	Explanation
1	28	56	Task 1 involves a lot of setting up components that will be used for the rest of the project. We assumed it would take some time to establish the development cycle during the first task.
2	24	48	Task 2 has fewer subtasks than task 1, but the subtasks are more complex than the ones in task 1. Since the due date for this milestone will most likely need to be moved up, we expect to spend less time on this task.
3	20	40	Task 3 also has fewer subtasks than task 1, and we believe that the subtasks will be relatively easy to implement. Like task 2, we will also have to move up the due date.
4	36	72	Task 4 will require more time due to the number of templates and the 2 themes that will need to be implemented. We expect the themes to take a while to implement across all of the templates.
5	29	58	Task 5 includes the addition of more content to the templates, another theme, and google maps integration. We expect that adding a 3rd theme and implementing the google maps integration will be the most time consuming subtasks for this task.
6	36	72	Task 6 involves implementing the remaining features for this project. This task has some potentially complex subtasks like the reviews integration and SEO, so we anticipate having to spend a significant amount of time on this task.

Total # of Hours: 346

Personal Effort Estimation

Since there are about 23 weeks left in the semester, and we have 6 group members, we expect the personal effort to be about 2.5 hours per week for project implementation.

2.7 Other Resource Requirements

Github will be used for source control for the project.

2.8 Financial Requirements

Financial requirements are not relevant for this project since there are no anticipated costs.

3 Design

3.1 Previous Work And Literature

Squarespace

Squarespace provides software as a service for building and hosting websites. They provide templates for different types of websites including photography, blogs, and online stores. Plans cost $12-$46 a month.

Advantages:

- A lot of templates and customization options
- Allows coding and customization through CSS, HTML, JavaScript
- Drag and drop user interface
- Community support through forums and guides
- Provides Google Analytics
- Fully responsive templates

Shortcomings:

- Squarespace won't help with troubleshooting if CSS, HTML, or JavaScript has been manually changed
- Restricting content through membership on a site can be difficult
- User interface and code changes frequently

Wix

Wix is another website that allows users to build their own websites using templates. The basic plan Wix plan is free, but has limited features. Premium plans cost $14-$39 a month.

Advantages:

- More beginner-friendly than Squarespace (websitetooltester)
- Drag and drop user interface
- Free basic plan
- Provides Google Analytics

Shortcomings:

- Templates are not responsive
- Cannot use CSS, HTML, or JavaScript
- Templates cannot be changed after the website has been built

3.2 Design Thinking

Detail any design thinking driven design "define" aspects that shape your design. Enumerate some of the other design choices that came up in your design thinking "ideate" phase.

Define Aspects:

- Buildertrend customers need to be able to market themselves and their products to their customers
- Buildertrend customers do not have a way to create their own sites without using sites like Squarespace or Wix or hiring a developer to create it for them.

Design Choices (already provided by Buildertrend):

- Creating a file server to save templates and edited site content
- Creating a frontend React with TypeScript application for building the sites
- Using TinaCMS to implement the content management system

3.3 Proposed Design

Include any/all possible methods of approach to solving the problem:

- Discuss what you have done so far – what have you tried/implemented/tested?
- Some discussion of how this design satisfies the **functional and non-functional requirements** of the project.
- If any **standards** are relevant to your project (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here
- This design description should be in **sufficient detail** that another team of engineers can look through it and implement it.

We will be creating a React with TypeScript application that uses TinaCMS to create the web application for building websites. The web application will have an admin panel that will allow a client to choose templates for their sites and then edit the content of the templates. Generic content like about, services, and privacy policy will be provided as well. The customer using the admin panel will be able to save

3.4 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

3.5 Design Analysis

- Did your proposed design from 3.3 work? Why or why not?
- What are your observations, thoughts, and ideas to modify or iterate over the design?

3.6 Development Process

Discuss what development process you are following with a rationale for it – Waterfall, TDD, Agile. Note that this is not necessarily only for software projects. Development processes are applicable for all design projects.

3.7 Design Plan

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or software.

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study or acceptance testing for functional and non-functional requirements).
2. Define/identify the individual items/units and interfaces to be tested.

3. Define, design, and develop the actual test cases.
4. Determine the anticipated test results for each test case
5. Perform the actual tests.
6. Evaluate the actual test results.
7. Make the necessary changes to the product being tested
8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you have determined.

4.1 Unit Testing

File Server:

The logic behind the endpoints for our file server will need unit testing. We will need to make sure that files are being saved and retrieved correctly from the file system. These tests also need to be in place for regression testing as changes are made to the file server logic. This will most likely be done with Mocha.

React Application:

The React Application will need unit tests as well. Buildertrend uses Jest and Enzyme for their React unit tests, so we will most likely use Jest and Enzyme for our project.

4.2 Interface Testing

– Discuss how the composition of two or more units (interfaces) are to be tested. Enumerate all the relevant interfaces in your design.

Interface between File Server and React App:

To test saving and retrieving files without creating or loading the file on the React Application, we will use test files and use Postman to test saving those files to the file system and retrieving them. For testing file server to frontend communication, the files will be saved locally.

4.3 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

4.4 Results

– List and explain any and all results obtained so far during the testing phase

- Include failures and successes
- Explain what you learned and how you are planning to change the design iteratively as you progress with your project

- If you are including figures, please include captions and cite it in the text

5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

6 Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.