LLVM.org Website Redesign

Google Summer of Code 2024 Proposal

The LLVM Compiler Infrastructure

Contact

Name: Chaitanya Shahare
Email: shaharechaitanya3@gmail.com
Alternate Email: chaitanya@chaitanyashahare.com
Contact no.: +91 7972808820
Time Zone: Kolkata, India (GMT+5:30)
University: National Institute of Technology Srinagar, J&K, India
Github: Chaitanya-Shahare
LinkedIn: Chaitanya Shahare
Discord: chaitanya8098
Website: chaitanyashahare.com
Preferred mode of communication: Email, Google Meet, Discord

Project

Improve the LLVM.org Website Look and Feel

Project size: Large (~ 350 hours)
Project timeframe: 27 May 2024 to 26 August 2024
Project Mentors: Tanya Lattner, Vassil Vassilev

Table of Contents

Time Commitment

I am fully committed to dedicating approximately **42 hours per week** to the GSoC project from **May 1, 2024**, to **August 26, 2024**. However, during my College End Term exams, scheduled tentatively from **June 24, 2024**, to **July 8, 2024**, my availability will be reduced to approximately **21 hours per week**. Any changes to these dates will be promptly communicated to the mentors. Aside from these exam periods, I have **no other commitments** and will be available to contribute to the project as needed. Any other changes and emergencies will also be promptly informed to the mentors.
Additionally, I am open to extending my commitment beyond the stated timeframe if required.

No.	Dates	Days (Total)	Time Commitment
1.	1 May to 24 June	Mon-Sun (7)	6 hr/day
2.	24 June to 8 July	Mon-Sun (7)	3 hr/day
3.	9 July to 26 August	Mon-Sun (7)	6 hr/day

Estimated Total Working Days: 100 - 120
Estimated Hours: 350 hours (This may change as per requirements).

Experience

Web Developer Intern at Zalco Technologies

- Developed **real-time chat application frontend** using **React.js**, integrating with backend APIs for seamless functionality based on **Figma designs**.
- Implemented **Socket.io** for instant messaging feature, enhancing user experience and engagement.
- Collaborated within cross-functional teams to translate Figma designs into responsive, multi-screen interfaces with a total of **30 - 40 screens**.
- **Deployed** the real-time chat application on an **AWS EC2** instance using **Docker**.

Web Developer Intern at Trivia Softwares

- Developed **5+ web applications** and integrated third-party APIs (Weather, News, Currency, Quotes) into applications for real-time data.
- Successfully attracted and managed a **user base of 300+** for the application.

Graphic Design Lead at TEDxNITSrinagar

- Designed posters for social media using Figma, which were posted weekly, a total of **20-30 posters**.
- Assisted in **Web Design** of TEDxNITSrinagar website.

- Designed sponsorship proposal brochures used to raise over **Rs. 5 lakh** (Indian Rupees).

Related Work

Chaitanya's Blog

- blog.chaitanyashahare.com
- Built using **HUGO (Static Site Generator)**, **HTML, CSS, JavaScript**, with a total of **185** individual responsive pages.
- Got **800+** monthly users, **1,600+** Google search clicks, **50K+** impressions in the last 3 months.
- Built **CI/CD pipeline** using GitHub Actions for building using HUGO and deployment with Github Pages.

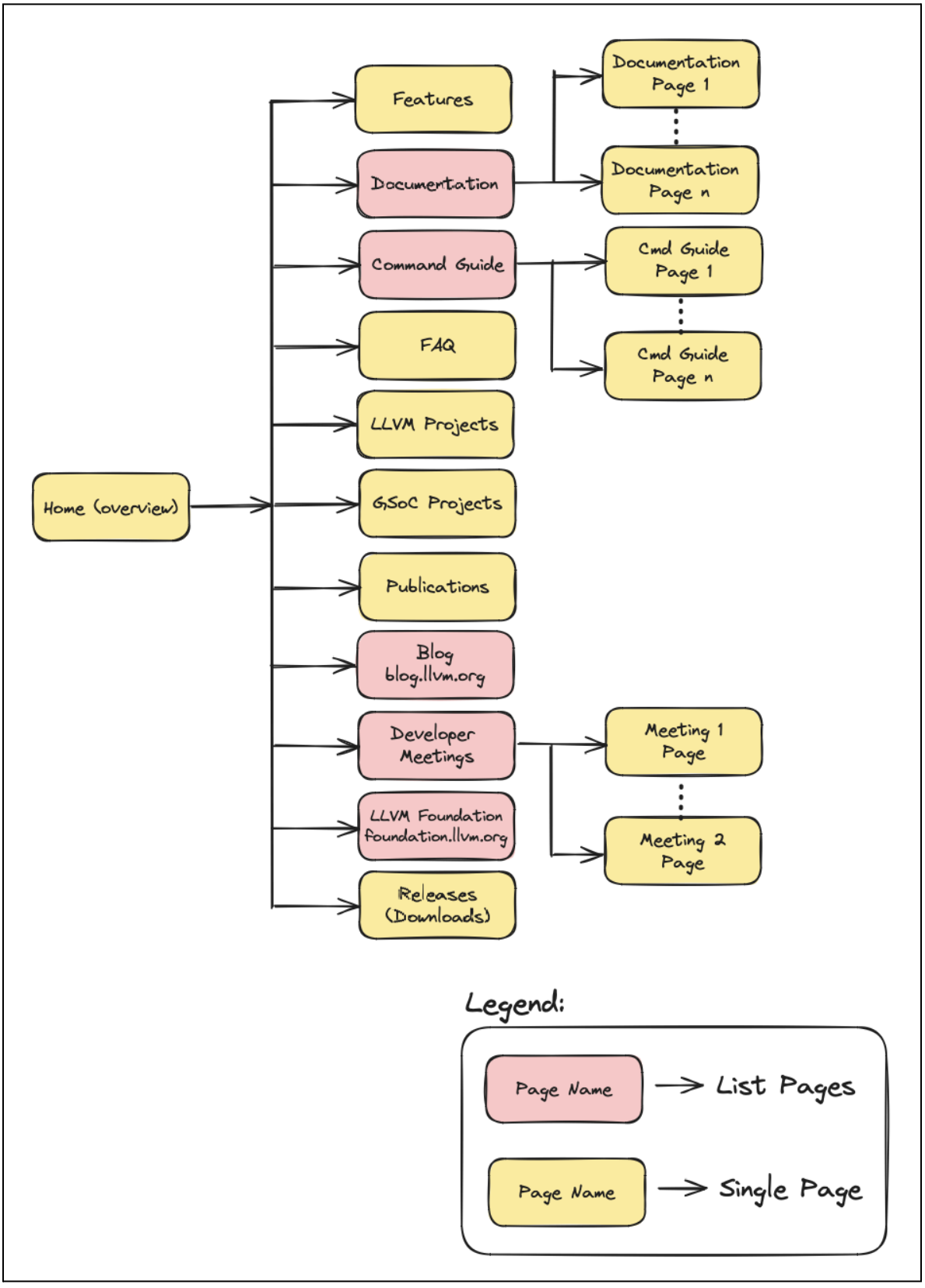
Why am I interested in this project?

I am deeply interested in this project for several reasons. Firstly, the technical requirements of the project closely align with my skill set and passions. I have a strong background in **web design, scripting, and automation**, which I believe will be invaluable in redesigning the LLVM website & migration from the old to new website. Additionally, I have a keen interest in **static site generators** and have actively utilized them in my personal projects, particularly through blogging. This project presents an exciting opportunity to merge my technical expertise with my enthusiasm for static site generators in a real-world context. Moreover, the LLVM website serves as a vital hub for the LLVM community, making this project particularly meaningful to me. I am eager to contribute to an essential part of the LLVM ecosystem and am committed to learning and growing throughout the process. Overall, I am **enthusiastic** about the prospect of working on such a significant project and am eager to apply my knowledge and skills to make a **meaningful impact**.

Benefits to the Community

The redesigned LLVM website will offer significant benefits to the community by enhancing user experience through **improved navigation, responsiveness, and accessibility**. By making it easy to access essential resources, the updated website will empower users to engage more effectively with the LLVM projects. Furthermore, the modern and coherent design will elevate the **project's visibility** and credibility, attracting new users and contributors while fostering a sense of community ownership. The **content creators** of the website will also be benefitted as they won't have to worry about HTML code, and can just **focus on the content**. Overall, the redesigned website will serve as a catalyst for collaboration, knowledge sharing, and the continued growth of the LLVM ecosystem.

Site Map


Home (overview)
Features
Documentation
Command Guide
FAQ
LLVM Projects
GSoC Projects
Publications
Blog
blog.llvm.org
Developer Meetings
LLVM Foundation
foundation.llvm.org
Releases
(Downloads)
Documentation Page 1
Documentation Page n
Cmd Guide Page 1
Cmd Guide Page n
Meeting 1 Page
Meeting 2 Page
Legend:
Page Name
List Pages
Page Name
Single Page

Target Audience of llvm.org

Developers and Contributors:

Experience: Developers and contributors are likely the most frequent visitors to the LLVM website. Their experience should prioritize ease of access to documentation, guides, tutorials, and code repositories. They should be able to quickly find the latest releases, release notes, and development guidelines. Additionally, providing clear pathways to contribute, such as links to bug trackers, mailing lists, and forums, is essential for fostering community engagement.

Students and Educators

Experience: Students and educators may visit the LLVM website for learning purposes, whether as part of coursework or self-study. Their experience should prioritize accessibility to educational materials, including tutorials, exercises, and learning paths tailored to different skill levels. Integration with educational platforms and initiatives, such as LLVM-based projects in programming courses or student internships (GSoC), can enrich their interaction with the website.

Researchers and Academics:

Experience: Researchers and academics often turn to LLVM for its robust compiler infrastructure. Their experience with the website should facilitate easy access to research papers, publications, and case studies highlighting LLVM's applications in various domains. Providing educational resources, such as lectures, workshops, and academic collaborations, can further enhance their interaction with the website.

General Audience and Enthusiasts:

Experience: The general audience and enthusiasts may have a more casual interest in LLVM, such as exploring its capabilities, attending community events, or staying updated on project news. Their experience should cater to easy navigation, engaging content, and opportunities for involvement, such as subscribing to newsletters, attending webinars, or participating in community forums. Providing multimedia content, such as videos, podcasts, and interactive demos, can make their interaction with the website more enjoyable and informative.

Current Website

The current LLVM website is built using **pure HTML** and employs **server-side includes**. Server-side includes enhance the website's maintainability by enabling the reuse of common elements across multiple pages, such as headers, footers, and navigation menus. This setup reflects a commitment to **simplicity and ease of maintenance**, essential considerations for a project of this scale and importance within the LLVM community. As I embark on this project, my familiarity with server-side includes positions me well to navigate and optimize the existing website structure, ensuring a seamless transition to the redesigned LLVM website.

Project Goals

During this GSoC term, I would deliver the following, these can be broadly classified into 2 phases:

Phase 1: Development of New Website

1. Content Audit
2. Selecting the Appropriate Technology
3. Design Mockups
4. Implementing the Chosen Design

Phase 2: Migrating from Old to New Website

5. Utilizing YAML for Data Representation
6. Migrating Content
7. Deployment and Hosting
8. Collaborating with Content Creators (Documentation)

Additional Features & Improvements

This is a list of features & enhancements that can be done if there's some time left, or even after the GSoC period,

1. **Dark Mode Integration:** Implementing a Dark Mode option to enhance user accessibility and provide an alternative viewing experience.
2. **Google Analytics Integration:** Incorporating Google Analytics to gain valuable insights into website traffic, user behavior, and performance metrics.
3. **Schema.org Structured Data:** Enhancing SEO and enabling richer search engine results by implementing Schema.org structured data markup.
4. **Site Performance Enhancements:** Optimizing website performance in alignment with Core Web Vitals metrics to ensure swift and seamless user interactions.

Phase 1: Execution Plan

Content Audit

Given the **absence of analytics** for the existing llvm.org website, our execution plan for the content audit will rely on insights gathered through the **Google form survey**. This survey will serve as the primary method for identifying **usage patterns**, **content preferences**, and **areas of improvement** on the current website. By leveraging responses from the survey, we will conduct a comprehensive assessment of the existing content, **prioritizing pages or sections** based on community feedback. This approach ensures that our content audit is **data-driven** and aligned with the needs and preferences of llvm.org users.

Sample Google Form Survey:
https://docs.google.com/forms/d/1QUYJEcERHX35CIQaLqBkKEesNH80olb2yOrZElffMTI/

Selecting the Appropriate Technology

For selecting the right technology for the llvm.org website redesign, my plan is to refer to the table created (Appendix A). This table thoroughly assesses three popular static site generators: **Gatsby.js, HUGO,** and **Jekyll**.
After careful evaluation, I've decided to go with HUGO. It stood out because of its **simplicity, quick build times, flexibility**, and **well-documented nature**. Based on these factors, I'm confident that HUGO will best meet the needs of the project. Our aim is to create a modern and user-friendly website, and HUGO aligns perfectly with this goal. Also, most of the project websites in LLVM are created using HUGO (mlir-www, llvm-blog-www, llvm-foundation-www), indicating **familiarity of the community** with the technology. Moving forward, my plan is to tailor and fine-tune HUGO website to suit the specific requirements of the llvm.org community, ensuring a smooth and successful redesign process.

Design Mockups

I propose to create **multiple design mockups** to offer the llvm.org community a range of options to choose from while **drawing inspiration from other projects & websites** that showcase effective organization & style. This entails developing **at least three distinct themes**, each with variations in style and layout. The proposed themes include:

1. **Classic**: This mockup will maintain the current design aesthetics (simplicity), providing a familiar look and feel for existing users.
2. **Modern**: Incorporating contemporary design elements and trends, this mockup will offer a fresh and updated appearance to attract new users.
3. **Avant-garde**: This mockup will explore innovative and experimental design concepts, catering to the preferences of forward-thinking web developers in 2024.

Each design will consist of **three pages (Home page, List page, Single page)**. I'll actively seek **feedback from the community** to ensure alignment with their preferences, incorporating diverse perspectives and addressing conflicting requests iteratively. Community members can review and provide feedback on each design mockup via **Google Forms** (sample form), enabling refinement until we achieve consensus.
To address the importance of visuals, I've allocated time to **develop the right set of clipart** to enhance the website's modern look.
You can view the design mockups created for the evaluation task of the project in Figma using the following links:

- Figma Design here
- PDF of Mockups here

References for Design Inspirations

- https://www.gatsbyjs.com/showcase/
- https://gohugo.io/showcase/
- https://jekyllrb.com/showcase/
- https://dribbble.com/

Implementing the Chosen Design

Developing a "HUGO theme" improves reusability of this project by separating structure and design from specific content in the llvm.org website. This theme, acting as a submodule, can also be utilized by other LLVM sub-projects such as clang.llvm.org, facilitating automatic propagation of updates across associated projects and eliminating the necessity for additional configuration.
I'll be creating a new Hugo theme **from scratch**, using **Tailwind CSS** for styling.

1. Create a **new HUGO theme**
   ```bash
   $ hugo new theme llvm-www-hugo-theme
   ```
2. Develop **Navigation bar**
3. Develop **Footer**
4. Develop **Single page**
5. Develop **List page**
6. Develop **Breadcrumbs** for Single page
7. Develop a custom **404 page**
8. Develop a **search functionality** using Pagefind (recommended by HUGO).
9. Develop any **other partials** as needed.

In every step **Responsiveness** of the website will be tested .

Phase 2: Execution Plan

Utilizing YAML for Data Representation

YAML data files offer a versatile solution for storing a variety of information critical to llvm.org, including **Team Members, Important Links, Project Details, Events, Resources, and FAQs**. By integrating these YAML files with HUGO features like **shortcodes and partials**, we can dynamically render this data onto website pages. This approach involves several key steps:

1. **Identifying suitable data:** Identifying the types of data suitable for storage in YAML files.
2. **File Structure:** Designing an intuitive file structure to organize these YAML data files effectively.
3. **Visualizations:** Developing shortcodes and partials to efficiently utilize the data from these files in visualizations.
4. **Documentation:** Documenting the process for adding new YAML data files and creating shortcodes or partials to leverage this data.

By **centralizing common data** such as project details and links in YAML files, we streamline the page creation process, **ensuring consistency and accuracy** across the website.

References:

- https://gohugo.io/methods/site/data/
- https://bwaycer.github.io/hugo_tutorial.hugo/extras/datafiles/

Migrating Content

1. **Write Conversion Script:**
   - Develop a bash script using html-to-markdown-cli to convert HTML pages to Markdown with YAML Front Matter for HUGO.
2. **Manual Correction:**
   - Review and correct converted Markdown files to fit project requirements.
3. **Testing and Validation:**
   - Test the entire site migration to ensure no loss of information.

Deployment and Hosting

1. **Implement Continuous Integration (CI):**
   - Set up GitHub Actions workflow for CI to automate the build process.
   - Configure CI to trigger builds on each commit and push to the main branch.
2. **Build Process:**
   - Define the build process in the GitHub Actions workflow YAML file.
   - Use the selected static site generator (HUGO) to generate the website files.
   - Setup a workflow for Pull Requests to generate a **PR Build Artifact** of the built site, allowing reviewers to download and review.
   - Deadlink checkers & linters can be integrated to enhance the review process.
3. **Deployment Workflow:**
   - Configure the deployment workflow in GitHub Actions to deploy the built website to GitHub Pages.
   - Utilize GitHub Actions to automatically deploy changes to the main branch to the GitHub Pages hosting environment.
   - Set up a **Preview Environment** on the Forked Repository for reviewing Pull Requests.
   - Alternatively for reviewing Pull Requests, PR build artifacts of the build site can be generated using Github Actions, which the reviewers can download and checkout changes.
4. **Environment Configuration:**
   - Ensure proper configuration of GitHub Pages settings, including custom domain.

Documentation

1. **Getting Started Guide:**
   - Create a comprehensive "Getting Started" guide for new contributors and users.
   - Cover installation instructions, basic usage, and common workflows.
2. **Configuration Documentation:**
   - Document the configuration options for the chosen static site generator (HUGO).
   - Include explanations and examples for customizing the website's settings, themes, and plugins.
3. **Content Management Guide:**
   - Provide a guide for content creators on how to create, edit, and manage website content.

- Include instructions on using Markdown syntax, adding images, and structuring content effectively.

Detailed Timeline

GSoC typically spans a duration of approximately **12 weeks**, with an additional Community Bonding Period of around 25 days (**~ 4 weeks**).

Community Bonding Period (**1 May 2024** to **26 May 2024**) (**~ 4 Weeks**)

No	Task	Week	Start Date	End Date
1.	**Content Audit**	**Week 1-4**	**1 May**	**5 May**
1.1	Create Google Form survey for llvm.org insights (sample)		1 May	2 May
1.2	Promote survey within llvm.org community.		3 May	7 May
1.3	Analyze survey responses for actionable insights.	Week 2-4	4 May	26 May
2.	**Design Mockups**	**Week 2-3**	**6 May**	**19 May**
2.1	Create design mockups using Figma for llvm.org redesign. (3 variants for each page)	Week 2	6 May	12 May
2.1.1	Homepage design		6 May	7 May
2.1.2	Single page design		8 May	8 May
2.1.3	List Page design		9 May	9 May
2.2	Develop clipart for website		10 May	12 May
2.3	Share design mockups with community for feedback via Google form (sample)	Week 3	13 May	13 May
2.4	Iterate on designs based on community input	Week 3	13 May	19 May
3.	**Implementing the chosen design**	**Week 4**	**20 May**	**26 May**
3.1	Create new HUGO theme and GitHub repository for the theme		20 May	20 May
3.2	Develop Navigation bar		21 May	26 May

Coding Period 1 (**27 May 2024** to **12 July 2024**) (**~ 6 Weeks**)

No	Task	Week	Start Date	End Date
3.	**Implementing the chosen design.**	**Week 1-6**	**27 May**	**12 July**
3.3	Develop Footer	Week 1	27 May	29 May
3.4	Develop Single Page	Week 2	30 May	6 June
3.5	Develop List Page	Week 3	7 June	14 June
3.6	Develop Breadcrumbs		15 June	17 June
3.7	Develop a custom 404 page	Week 4	18 June	19 June
3.8	Develop a search functionality using Pagefind	Week 5	20 June	30 June
3.9	Develop any other partials as needed	Week 6	1 July	12 July

Coding Period 2 (**12 July 2024** to **26 August 2024**) (**~ 6 Weeks**)

No	Task	Week	Start Date	End Date
4.	**Utilizing YAML for Data Representation**	**Week 1**	**12 July**	**19 July**
4.1	Identifying suitable data		12 July	12 July
4.2	File Structure of data files		13 July	13 July
4.3	Development of Visualizations		14 July	16 July
4.4	Documentation for data files		17 July	19 July
5.	**Migrating Content**	**Week 2-3**	**20 July**	**31 July**
5.1	Write conversion script (html to markdown/yaml)	Week 2	20 July	22 July
5.2	Manual Correction		23 July	25 July
5.3	Testing and Validation	Week 3	26 July	31 July
6.	**Deployment and Hosting**	**Week 3-4**	**1 Aug**	**11 Aug**
6.1	Implement Continuous Integration (CI)		1 Aug	3 Aug
6.2	Build Process	Week 3	4 Aug	6 Aug

6.3	Deployment Workflow		7 Aug	10 Aug
6.4	Environment Configuration	Week 4	11 Aug	11 Aug
7.	**Documentation**	**Week 5**	**12 Aug**	**18 Aug**
7.1	Getting Started Guide		12 Aug	14 Aug
7.2	Configuration Documentation		15 Aug	16 Aug
7.3	Content Management Guide		17 Aug	18 Aug

Gantt Chart

Gantt chart LLVM.org Website Redesign

For a visual representation & clarity, please refer to the **Gantt Chart** in Appendix B.
NOTE: The timeline is flexible and can be adjusted & extended as needed to accommodate additional features.

Expectations from the mentor

As a prospective mentee, I anticipate that my mentor will provide guidance, support, and **constructive feedback** throughout the project duration. I expect regular communication and availability for **scheduled meetings to discuss progress, challenges, and project direction**. Additionally, I look forward to receiving mentorship on technical aspects such as web development best practices, utilization of static site generators, and effective collaboration within the LLVM community. Furthermore, I hope for mentor assistance in **navigating community processes, fostering consensus, and addressing any project-related hurdles** that may arise. Overall, I am eager to collaborate closely with my mentor to ensure the successful execution of the llvm.org website redesign project.

Why I Stand Out: Compelling Reasons to Choose Me

What sets me apart and makes me a strong candidate for the llvm.org website redesign project lies in my unique blend of skills, experiences, and qualities. With a **solid foundation in web development**, I bring a wealth of technical expertise in HTML, CSS, and Markdown to the table. Moreover, **my hands-on experience with static site generators, including HUGO**, positions me well to tackle the project's technological requirements. Beyond technical prowess, my **passion for user-centric design** and accessibility is evident in my proactive efforts to create engaging content and foster community engagement. I am a **proactive communicator**, adept at navigating collaborative processes. Furthermore, my commitment to **continuous learning and growth** ensures that I am always ready to adapt to new challenges and contribute effectively to the project. Overall, my unique combination of technical skills, design sensibilities, and collaborative mindset makes me a standout candidate poised to make meaningful contributions to the llvm.org website redesign project.

After GSoC

Looking ahead beyond the GSoC period, I am eager to continue contributing to the enhancement of the llvm.org website by implementing additional features and improvements as discussed earlier. I am committed to incorporating **Google Analytics** to gain valuable insights into user behavior and website performance, enabling continuous refinement and optimization. Additionally, I am enthusiastic about integrating **Schema.org structured data** to bolster the website's SEO and visibility in search engine results, ensuring its relevance to users and stakeholders. Moreover, I am dedicated to **optimizing site performance** in alignment with Core Web Vitals metrics, prioritizing seamless user experiences and engagement. With a proactive mindset and a passion for ongoing improvement, I am fully prepared to undertake these tasks to further elevate the llvm.org website's effectiveness and **user satisfaction** beyond the scope of the GSoC program.

Important Links

- Gantt chart LLVM.org Website Redesign
- Static Site Generators comparison
- Content Audit Survey Google Form (sample)
- Design Feedback Google Form (sample)
- Design mockups for evaluation task
    - Figma

References

- https://gohugo.io/
- https://www.gatsbyjs.com/
- https://jekyllrb.com/
- https://gohugo.io/tools/search/
- https://gohugo.io/methods/site/data/
- https://bwaycer.github.io/hugo_tutorial.hugo/extras/datafiles/
- https://github.com/cloudcannon/pagefind
- https://www.gatsbyjs.com/showcase/
- https://gohugo.io/showcase/
- https://jekyllrb.com/showcase/
- https://dribbble.com/
- https://github.com/llvm/
- https://www.npmjs.com/package/@wcj/html-to-markdown-cli

Appendix

A. Static Site Generators Comparison for LLVM.org Website Redesign Project

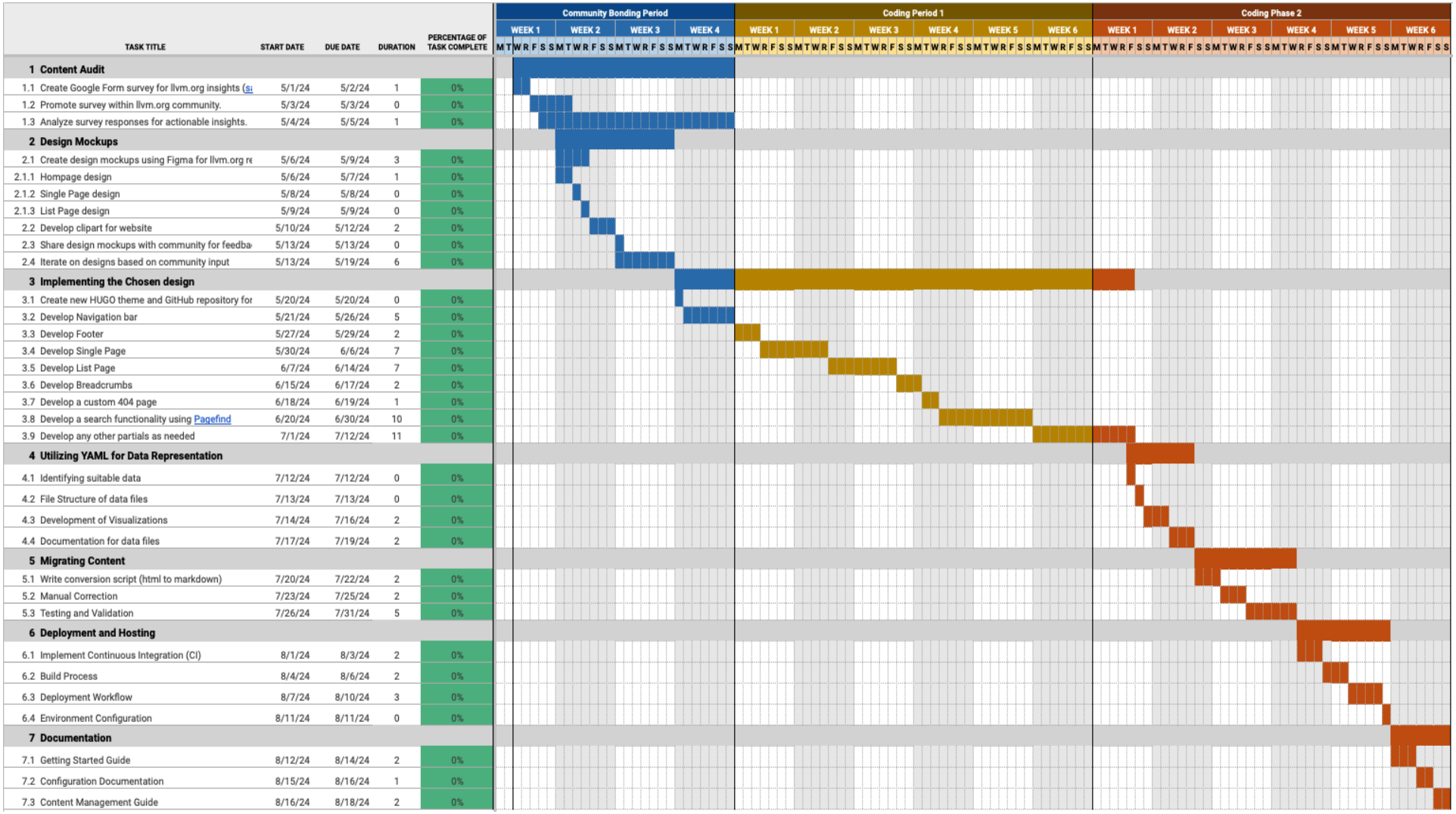
Features / Aspects	Gatsby.js	HUGO	Jekyll
Ease of Use	Gatsby's **React-based** architecture makes it easy for developers familiar with **JavaScript** and React to get started.	Hugo's simplicity and **fast build** times make it user-friendly, especially for those proficient in Markdown and **Go**.	Jekyll's straightforward setup and **minimal configuration** make it easy to use for those comfortable with Markdown.
Templating Language	**React Templating**	**Go Templating**	**Liquid templating**
Performance	Gatsby.js offers **excellent performance** out of the box, with optimized builds and lazy loading for efficient site loading.	HUGO is renowned for its **blazing-fast build times and low resource consumption**, resulting in fast website rendering.	Jekyll's **build times are generally faster** compared to other static site generators, making it suitable for projects with large content bases.
Flexibility	Gatsby's **plugin ecosystem** provides a wide range of customization options, allowing for integration with various data sources and services.	HUGO offers a high degree of flexibility through its powerful templating **system and customizable layouts**, making it suitable for diversity.	Jekyll's simplicity may **limit some advanced customization options**, but its plugin system and community support provide ample opportunities
Community Support	Gatsby.js boasts a **large and active community** that continually contributes plugins, themes, and resources, providing extensive support	HUGO has a **strong and dedicated community of users and contributors**, offering comprehensive documentation	Jekyll has been around for a longer time, resulting in a **mature ecosystem** with abundant resources, tutorials, and community forums for assistance.

	and resources for developers.	and support forums for users.	
Documentation	Comprehensive documentation with clear organization and helpful examples, **though some advanced topics may lack detail**.	Extensive and well-maintained documentation, offering **thorough coverage** and clear explanations, making it highly regarded for **completeness** and clarity.	Solid documentation covering essential topics, but may **lack depth in advanced features and troubleshooting**. Additional resources may be needed for full exploration of capabilities.
Documentation Rating	**4 / 5**	**4.5 / 5**	**3.5 / 5**

References

- https://www.gatsbyjs.com/
- https://gohugo.io/
- https://jekyllrb.co

B. Gantt Chart



	TASK TITLE	START DATE	DUE DATE	DURATION	PERCENTAGE OF TASK COMPLETE
1	**Content Audit**				
1.1	Create Google Form survey for llvm.org insights (si	5/1/24	5/2/24	1	0%
1.2	Promote survey within llvm.org community.	5/3/24	5/3/24	0	0%
1.3	Analyze survey responses for actionable insights.	5/4/24	5/5/24	1	0%
2	**Design Mockups**				
2.1	Create design mockups using Figma for llvm.org re	5/6/24	5/9/24	3	0%
2.1.1	Hompage design	5/6/24	5/7/24	1	0%
2.1.2	Single Page design	5/8/24	5/8/24	0	0%
2.1.3	List Page design	5/9/24	5/9/24	0	0%
2.2	Develop clipart for website	5/10/24	5/12/24	2	0%
2.3	Share design mockups with community for feedba	5/13/24	5/13/24	0	0%
2.4	Iterate on designs based on community input	5/13/24	5/19/24	6	0%
3	**Implementing the Chosen design**				
3.1	Create new HUGO theme and GitHub repository for	5/20/24	5/20/24	0	0%
3.2	Develop Navigation bar	5/21/24	5/26/24	5	0%
3.3	Develop Footer	5/27/24	5/29/24	2	0%
3.4	Develop Single Page	5/30/24	6/6/24	7	0%
3.5	Develop List Page	6/7/24	6/14/24	7	0%
3.6	Develop Breadcrumbs	6/15/24	6/17/24	2	0%
3.7	Develop a custom 404 page	6/18/24	6/19/24	1	0%
3.8	Develop a search functionality using Pagefind	6/20/24	6/30/24	10	0%
3.9	Develop any other partials as needed	7/1/24	7/12/24	11	0%
4	**Utilizing YAML for Data Representation**				
4.1	Identifying suitable data	7/12/24	7/12/24	0	0%
4.2	File Structure of data files	7/13/24	7/13/24	0	0%
4.3	Development of Visualizations	7/14/24	7/16/24	2	0%
4.4	Documentation for data files	7/17/24	7/19/24	2	0%
5	**Migrating Content**				
5.1	Write conversion script (html to markdown)	7/20/24	7/22/24	2	0%
5.2	Manual Correction	7/23/24	7/25/24	2	0%
5.3	Testing and Validation	7/26/24	7/31/24	5	0%
6	**Deployment and Hosting**				
6.1	Implement Continuous Integration (CI)	8/1/24	8/3/24	2	0%
6.2	Build Process	8/4/24	8/6/24	2	0%
6.3	Deployment Workflow	8/7/24	8/10/24	3	0%
6.4	Environment Configuration	8/11/24	8/11/24	0	0%
7	**Documentation**				
7.1	Getting Started Guide	8/12/24	8/14/24	2	0%
7.2	Configuration Documentation	8/15/24	8/16/24	1	0%
7.3	Content Management Guide	8/16/24	8/18/24	2	0%