



LLVM.org Website Redesign

Google Summer of Code 2024

Chaitanya Shahare 06.11.2024

Mentors: Tanya Lattner & Vassil Vassilev





Why do we need a new website?



Current Website

- Hard to Navigate
- Cluttered
- Outdated Design

The LLVM Compiler Infrastructure

LLVM Overview

The LLVM Project is a collection of modular and reusable compiler and toolchain technologies. Despite its name, LLVM has little to do with traditional virtual machines. The name "LLVM" itself is not an acronym; it is the full name of the project.

LLVM began as a [research project](#) at the [University of Illinois](#), with the goal of providing a modern, SSA-based compilation strategy capable of supporting both static and dynamic compilation of arbitrary programming languages. Since then, LLVM has grown to be an umbrella project consisting of a number of subprojects, many of which are being used in production by a wide variety of [commercial and open source](#) projects as well as being widely used in [academic research](#). Code in the LLVM project is licensed under the ["Apache 2.0 License with LLVM exceptions"](#)

The primary sub-projects of LLVM are:

1. The **LLVM Core** libraries provide a modern source- and target-independent [optimizer](#), along with [code generation support](#) for many popular CPUs (as well as some less common ones!) These libraries are built around a [well specified](#) code representation known as the LLVM intermediate representation ("LLVM IR"). The LLVM Core libraries are [well documented](#), and it is particularly easy to invent your own language (or port an existing compiler) to use [LLVM as an optimizer and code generator](#).
2. **Clang** is an "LLVM native" C/C++/Objective-C compiler, which aims to deliver amazingly fast compiles, extremely useful [error and warning messages](#) and to provide a platform for building great source level tools. The [Clang Static Analyzer](#) and [clang-tidy](#) are tools that automatically find bugs in your code, and are great examples of the sort of tools that can be built using the Clang frontend as a library to parse C/C++ code.
3. The **LLDB** project builds on libraries provided by LLVM and Clang to provide a great native debugger. It uses the Clang ASTs and expression parser, LLVM JIT, LLVM disassembler, etc so that it provides an experience that "just works". It is also blazing fast and much more memory efficient than GDB at loading symbols.
4. The **libc++** and **libc++ ABI** projects provide a standard conformant and high-performance implementation of the C++ Standard Library, including full support for C++11 and C++14.
5. The **compiler-rt** project provides highly tuned implementations of the low-level code generator support routines like `__fixunsdftdi` and other calls generated when a target doesn't have a short sequence of native instructions to implement a core IR operation. It also provides implementations of run-time libraries for dynamic testing tools such as [AddressSanitizer](#), [ThreadSanitizer](#), [MemorySanitizer](#), and [DataFlowSanitizer](#).
6. The **MLIR** subproject is a novel approach to building reusable and extensible compiler infrastructure. MLIR

Latest LLVM Release!

18 June 2024: LLVM 18.1.8 is now [available for download!](#) LLVM is publicly available under an open source [License](#). Also, you might want to check out [the new features](#) in Git that will appear in the next LLVM release. If you want them early, [download LLVM](#) through anonymous Git.

Upcoming Events

April 9-11, 2024 - EuroLLVM Dev Mig

ACM Software System Award!

LLVM has been awarded the **2012 ACM Software System Award!** This award is given by ACM to *one* software system worldwide every year. LLVM is [in highly distinguished company!](#) Click on any of the individual recipients' names on that page for the detailed citation describing the award.

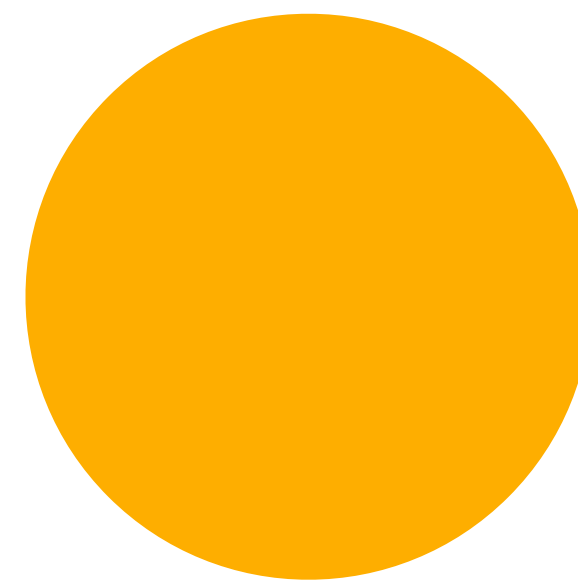
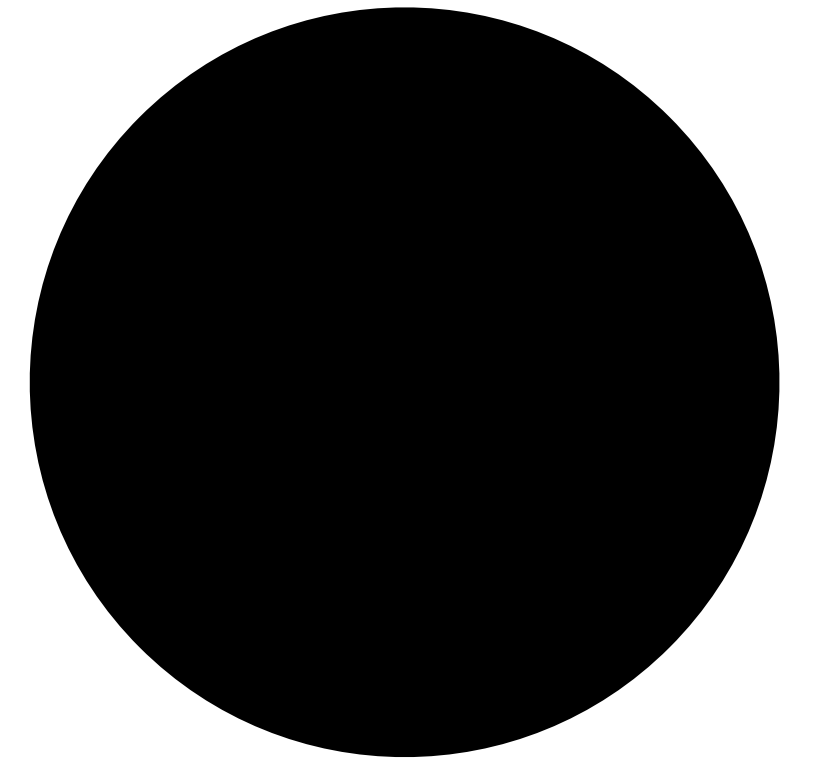
Upcoming Releases

LLVM Release Schedule:

- 19.1.x
 - Jul 23rd: release/19.x branch created
 - Jul 26th: 19.1.0-rc1
 - Aug 6th: 19.1.0-rc2
 - Aug 20th: 19.1.0-rc3
 - Sep 3rd: 19.1.0
 - Sep 17th: 19.1.1
 - Oct 1st: 19.1.2
 - Oct 15th: 19.1.3
 - Oct 29th: 19.1.4

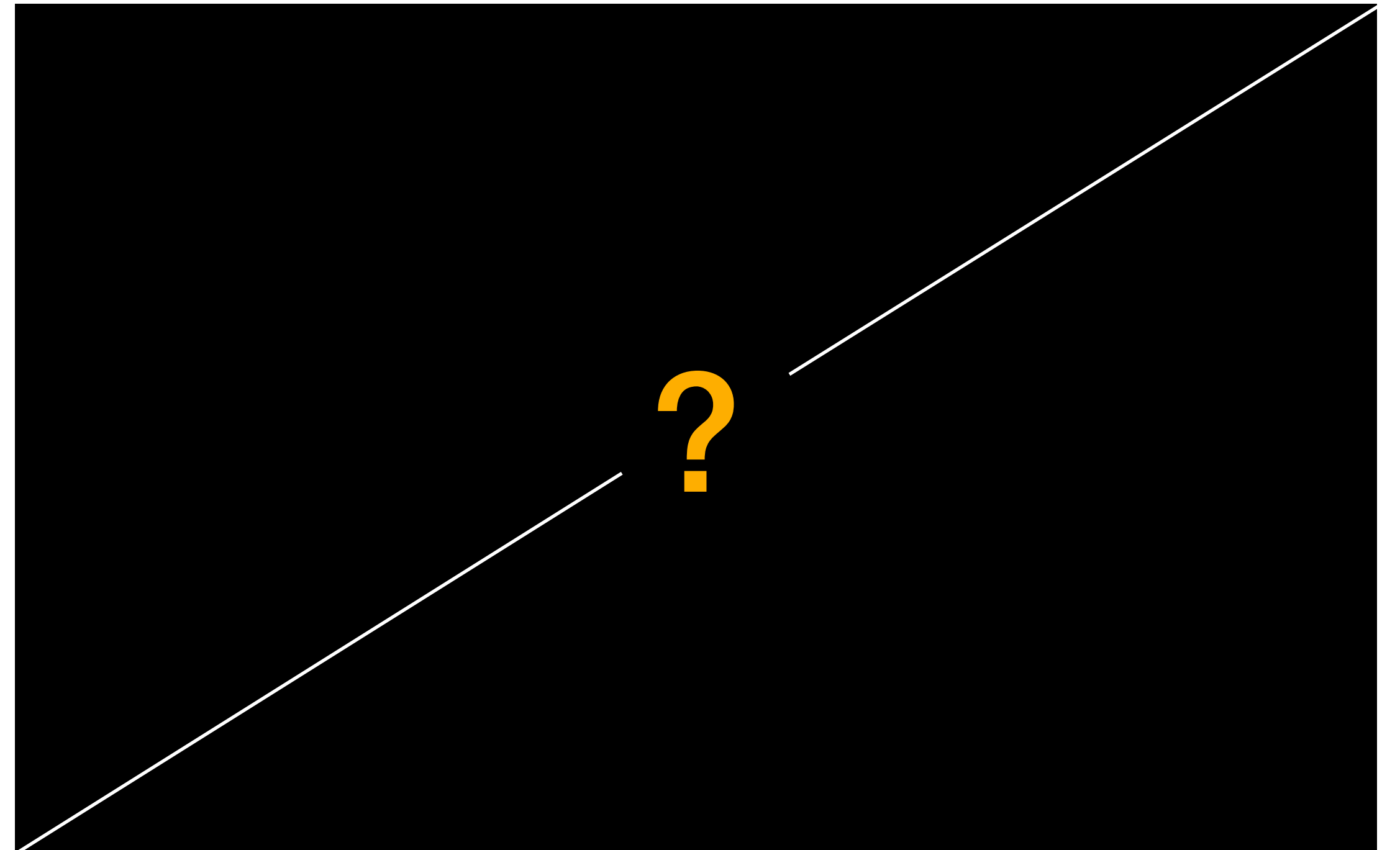
Screenshot of llvm.org landing page

Project Goals



Project Goals

- Modern LLVM.org website
- Navigation, mobile support & accessibility
- Content Discoverability & Usability
- Maintainability





Project **Scope**



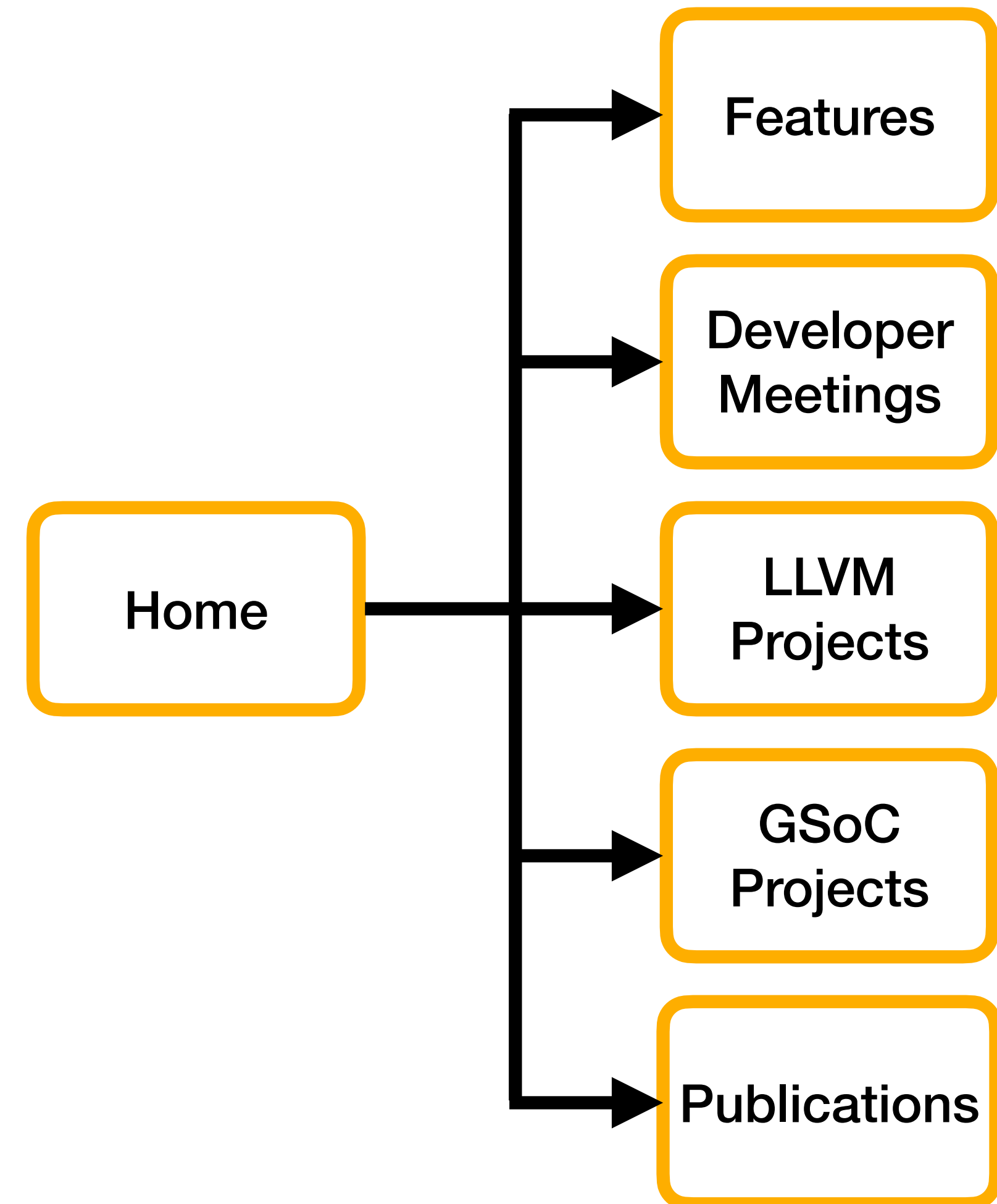
Project Scope

Objective: Redesign the LLVM website to improve user experience and engagement.

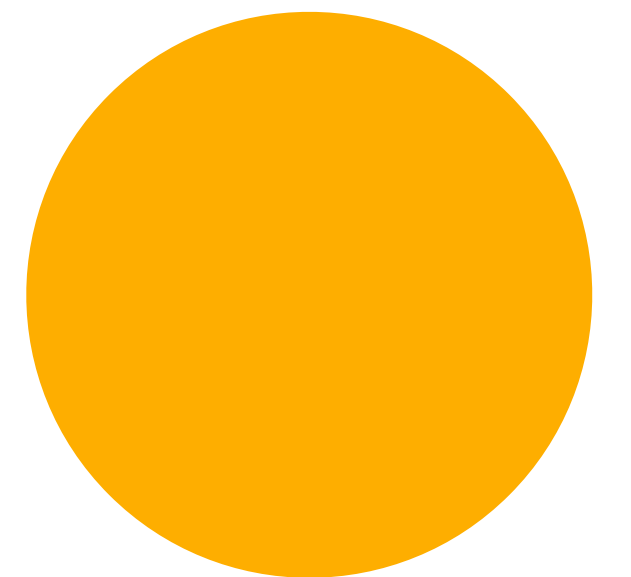
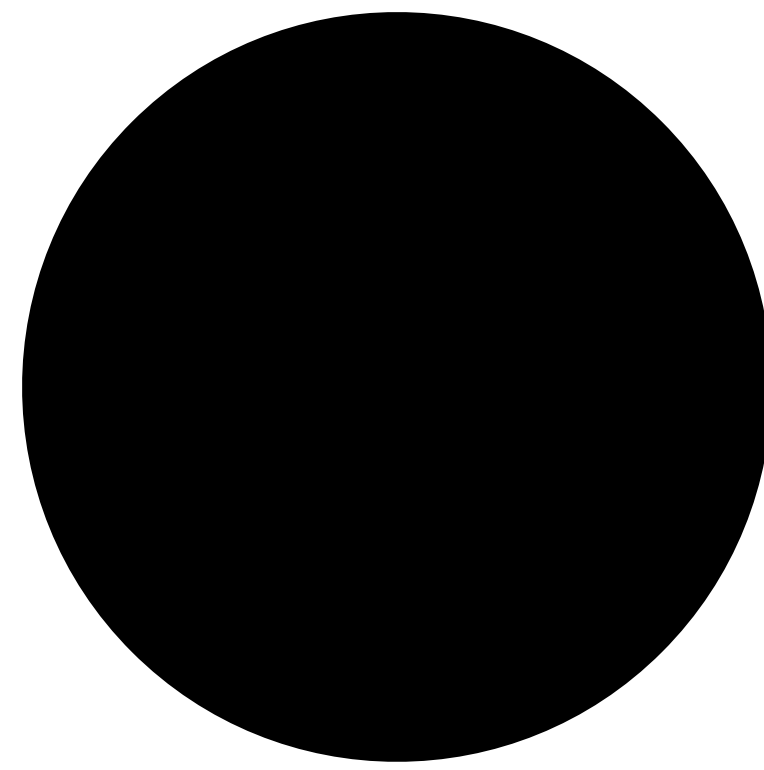
Key Areas of Focus:

- Modernize design and layout
- Accessibility and usability
- Maintainability

Site Map

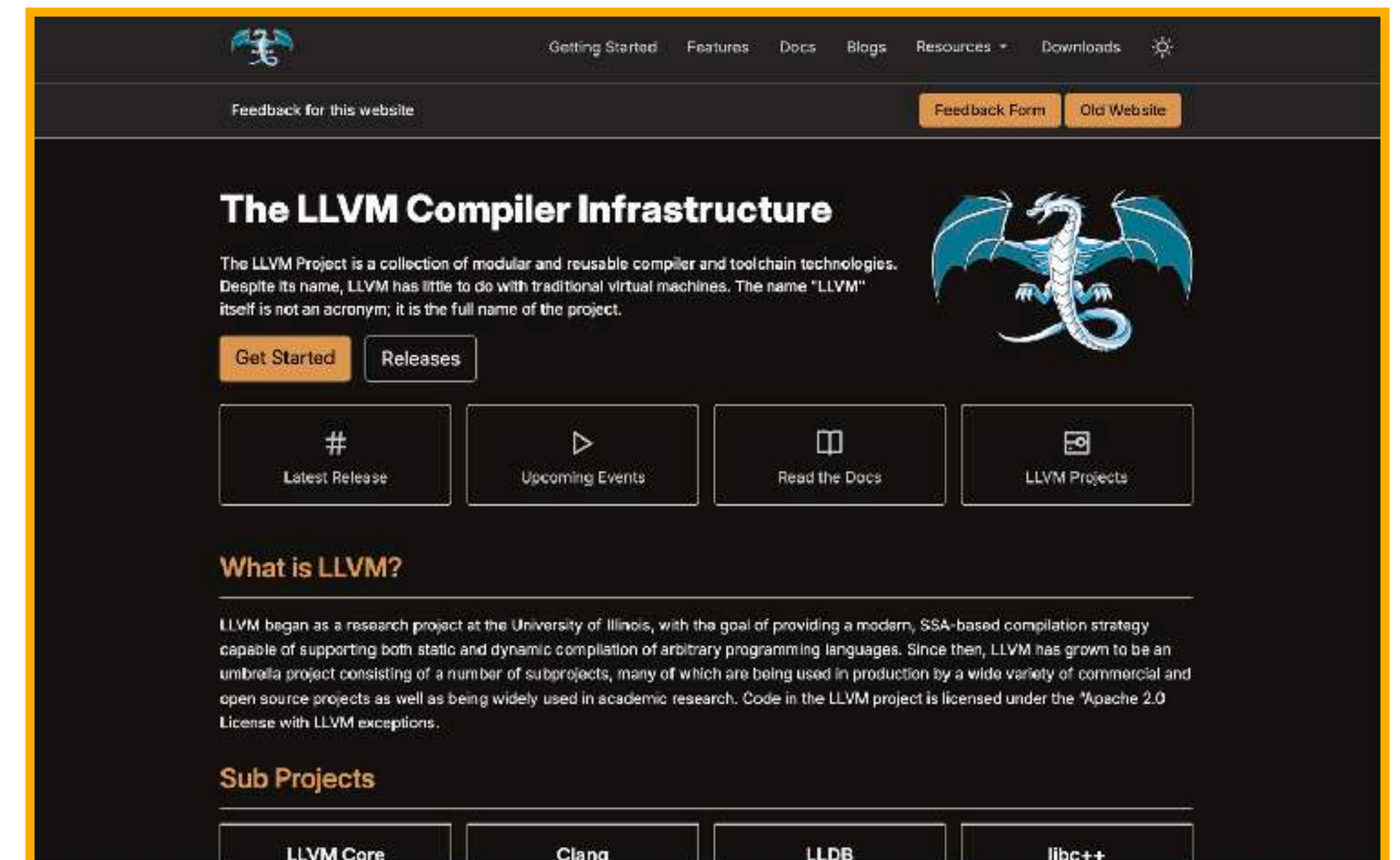
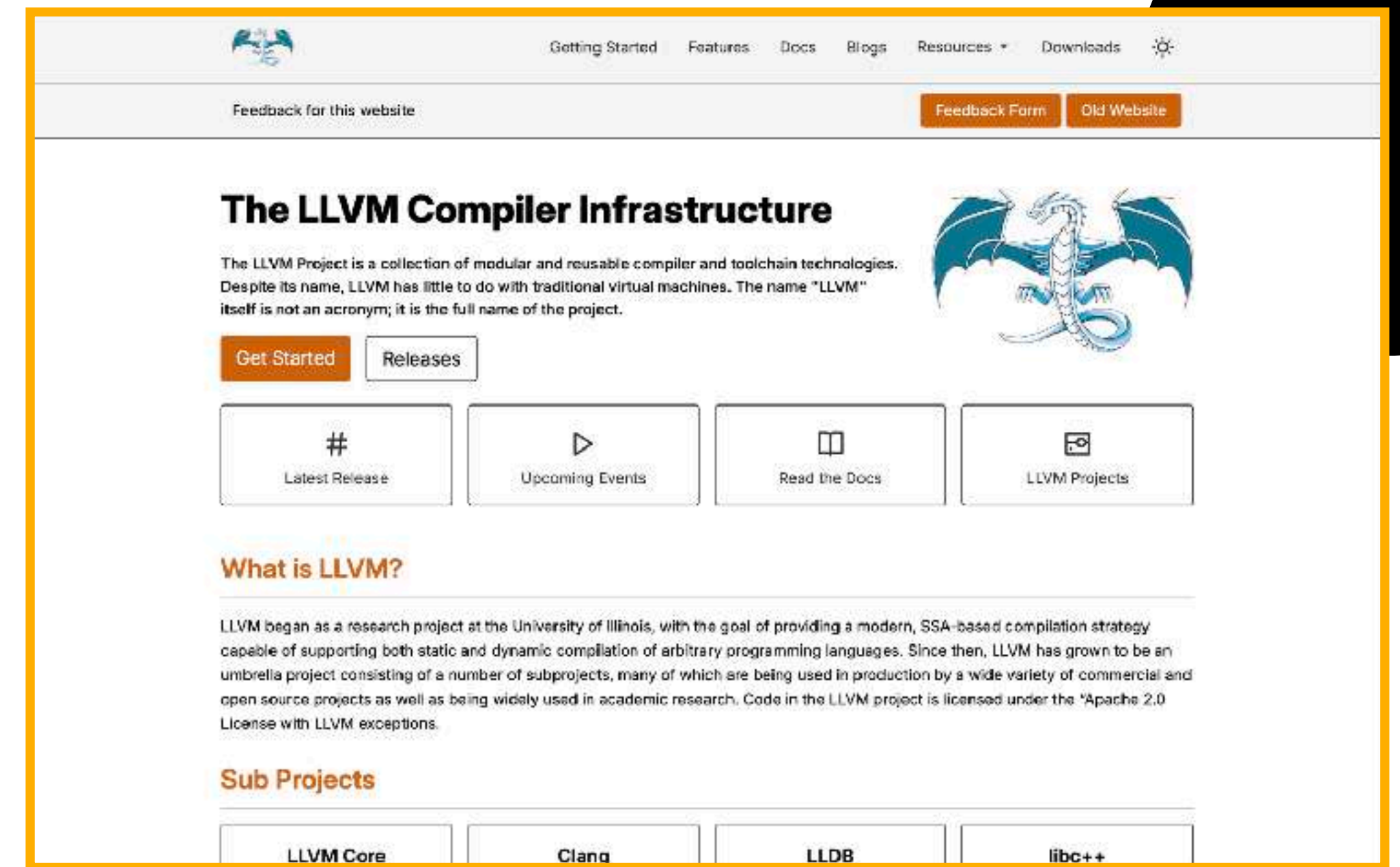


Achievements



Achievements

- Content Audit
- Survey (Current Website)
- RFC
- Create Design Mockups
- Survey for feedback on designs
- Reusable HUGO theme llvm/www-template
- Developed new website llvm/www-new
- Detailed Documentation



Screenshots of Website

[Getting Started](#)[Features](#)[Docs](#)[Blogs](#)[Resources](#)[Downloads](#)[Feedback for this website](#)[Feedback Form](#)[Old Website](#)

The LLVM Compiler Infrastructure

The LLVM Project is a collection of modular and reusable compiler and toolchain technologies. Despite its name, LLVM has little to do with traditional virtual machines. The name "LLVM" itself is not an acronym; it is the full name of the project.

[Get Started](#)[Releases](#)[Latest Release](#)[Upcoming Events](#)[Read the Docs](#)[LLVM Projects](#)

What is LLVM?

LLVM began as a research project at the University of Illinois, with the goal of providing a modern, SSA-based compilation strategy capable of supporting both static and dynamic compilation of arbitrary programming languages. Since then, LLVM has grown to be an umbrella project consisting of a number of subprojects, many of which are being used in production by a wide variety of commercial and open source projects as well as being widely used in academic research. Code in the LLVM project is licensed under the "Apache 2.0 License with LLVM exceptions."

Sub Projects

[LLVM Core](#)[Clang](#)[LLDB](#)[libc++](#)

[Getting Started](#)[Features](#)[Docs](#)[Blogs](#)[Resources](#)[Downloads](#)[Feedback for this website](#)[Feedback Form](#)[Old Website](#)

The LLVM Compiler Infrastructure

The LLVM Project is a collection of modular and reusable compiler and toolchain technologies. Despite its name, LLVM has little to do with traditional virtual machines. The name "LLVM" itself is not an acronym; it is the full name of the project.

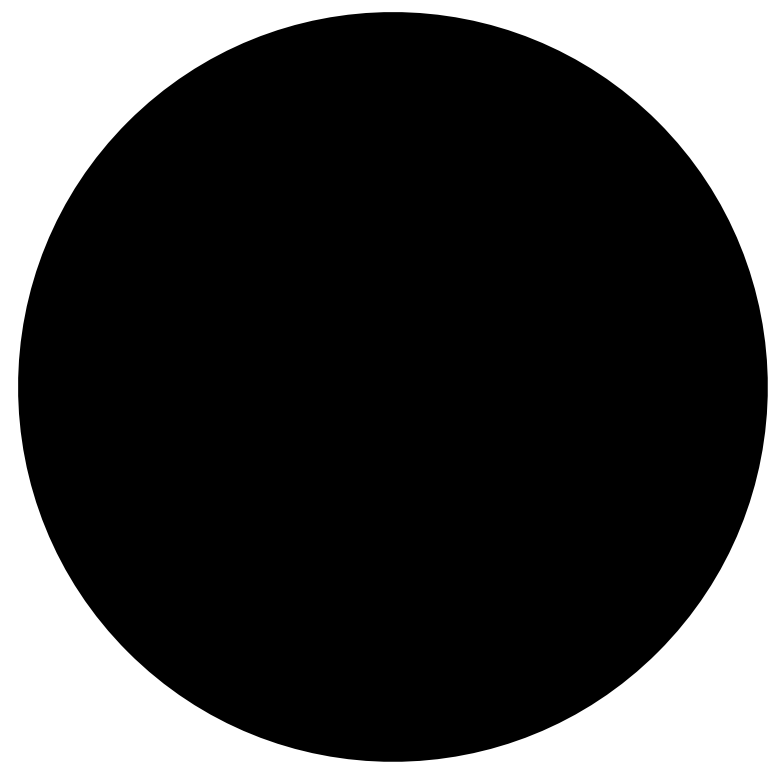
[Get Started](#)[Releases](#)[Latest Release](#)[Upcoming Events](#)[Read the Docs](#)[LLVM Projects](#)

What is LLVM?

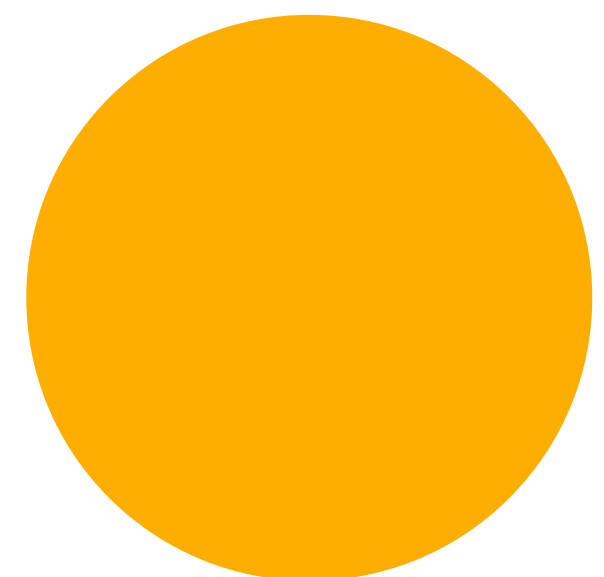
LLVM began as a research project at the University of Illinois, with the goal of providing a modern, SSA-based compilation strategy capable of supporting both static and dynamic compilation of arbitrary programming languages. Since then, LLVM has grown to be an umbrella project consisting of a number of subprojects, many of which are being used in production by a wide variety of commercial and open source projects as well as being widely used in academic research. Code in the LLVM project is licensed under the "Apache 2.0 License with LLVM exceptions."

Sub Projects

[LLVM Core](#)[Clang](#)[LLDB](#)[libc++](#)



Features Of New Website



Features



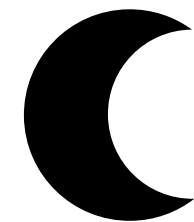
- Modern Design



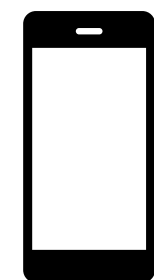
- Reusable Theme



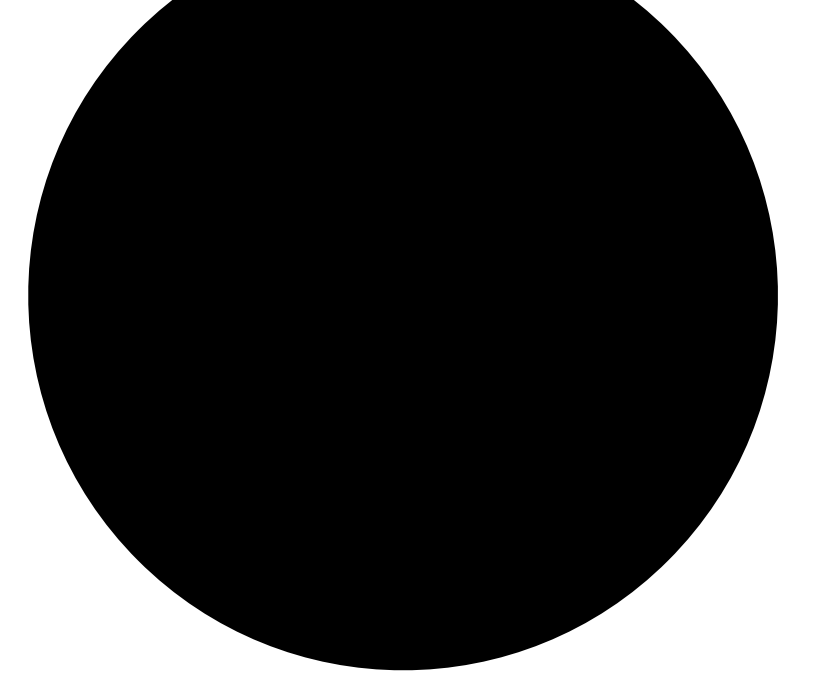
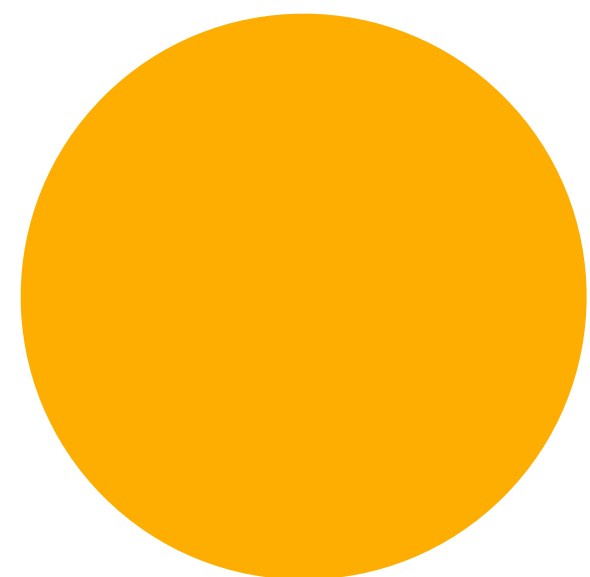
- Content updates with YAML files

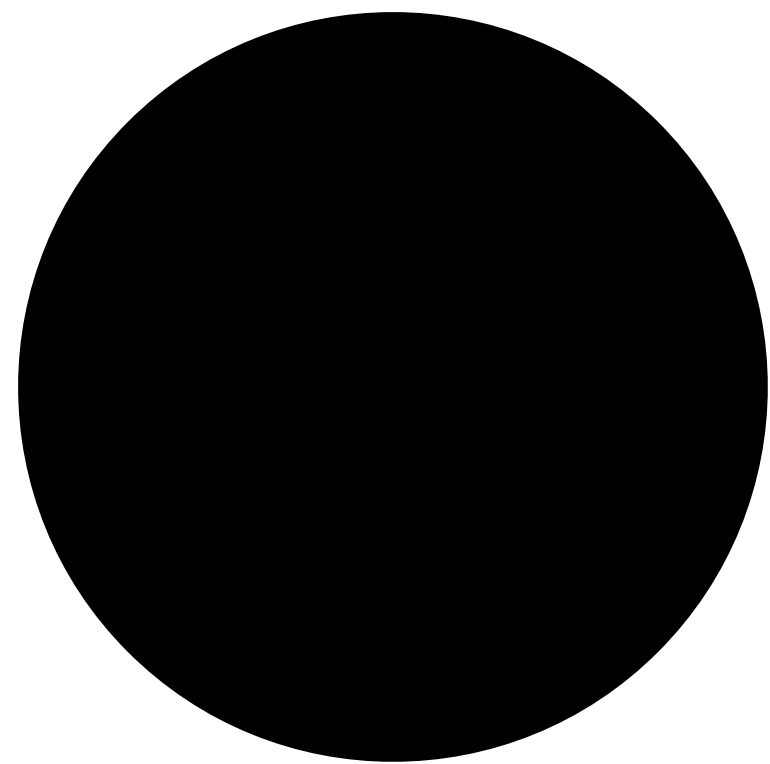


- Dark & Light color schemes

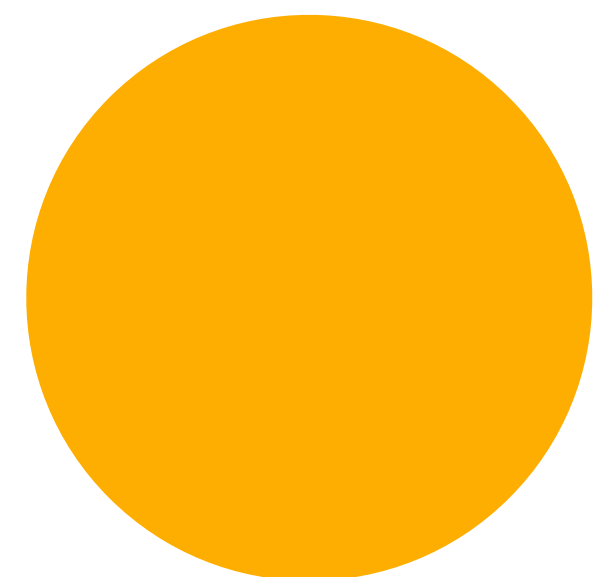


- Responsive layout





Development & Implementation



Development



- LLVM HUGO Theme



- YAML Data Driven Content

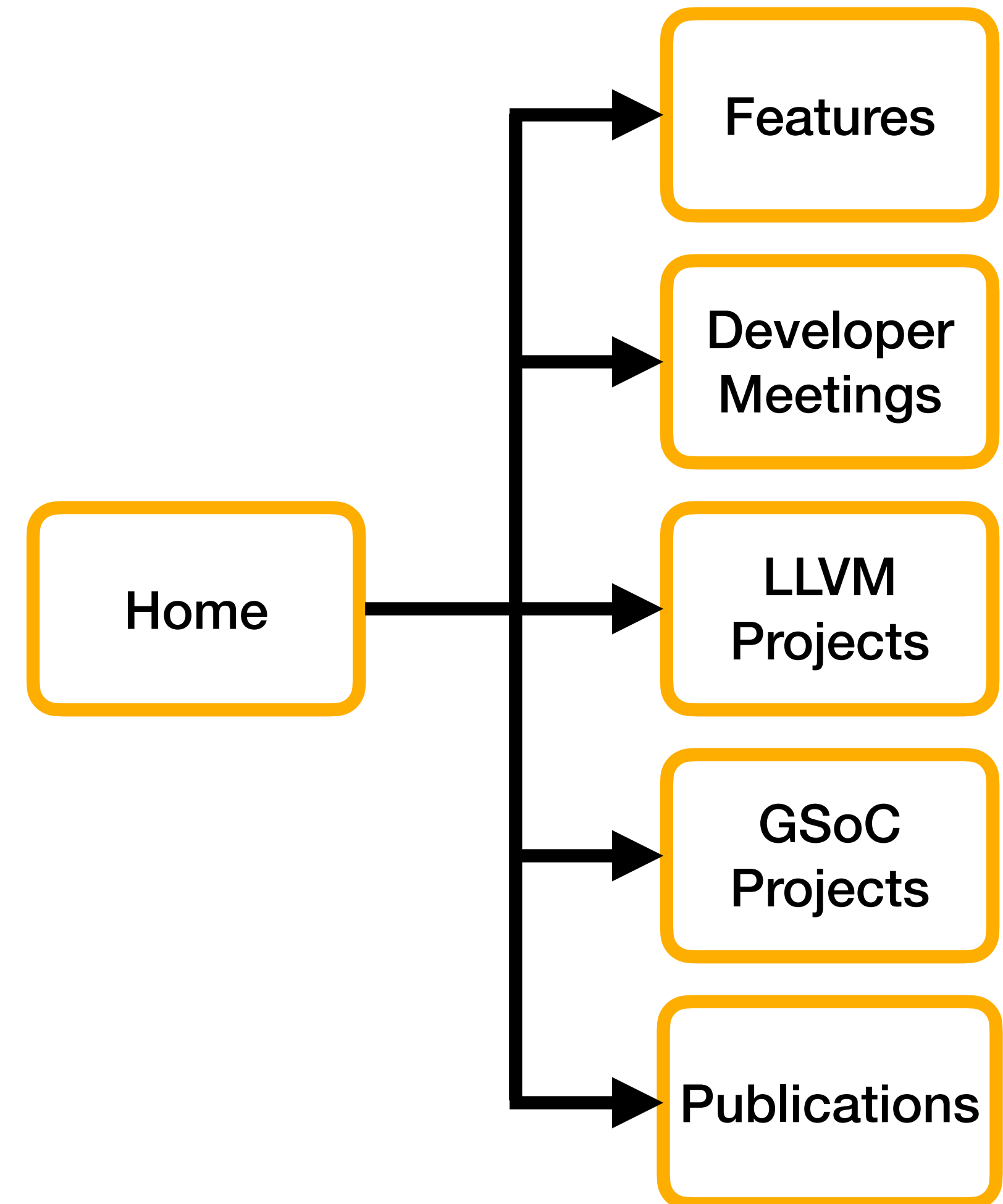


- Custom Shortcodes

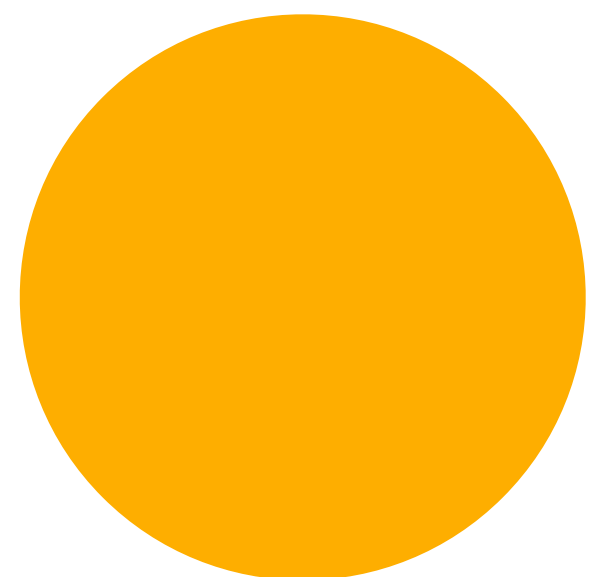
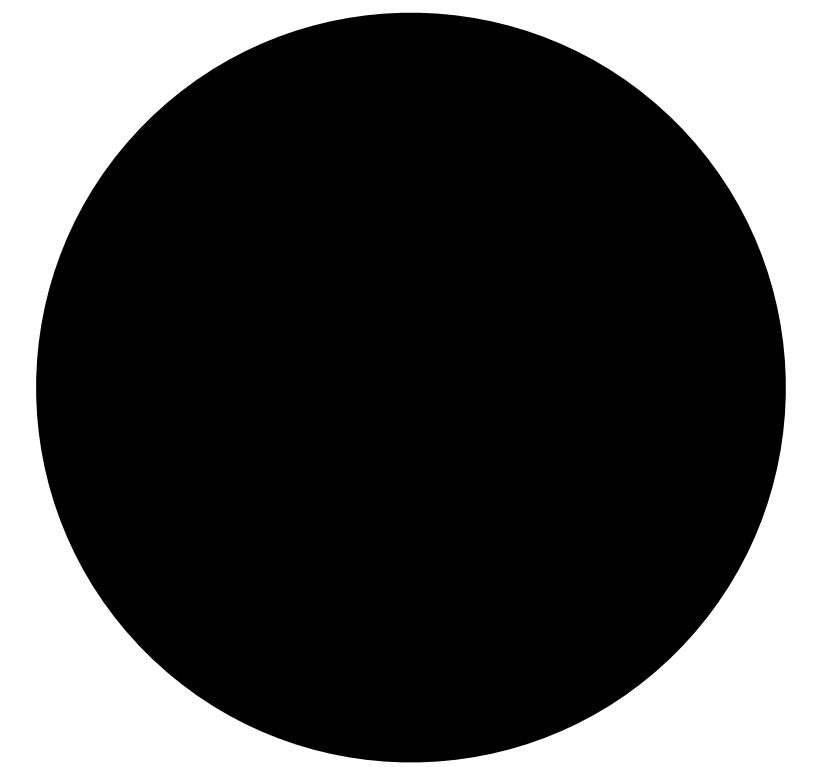


- New Website

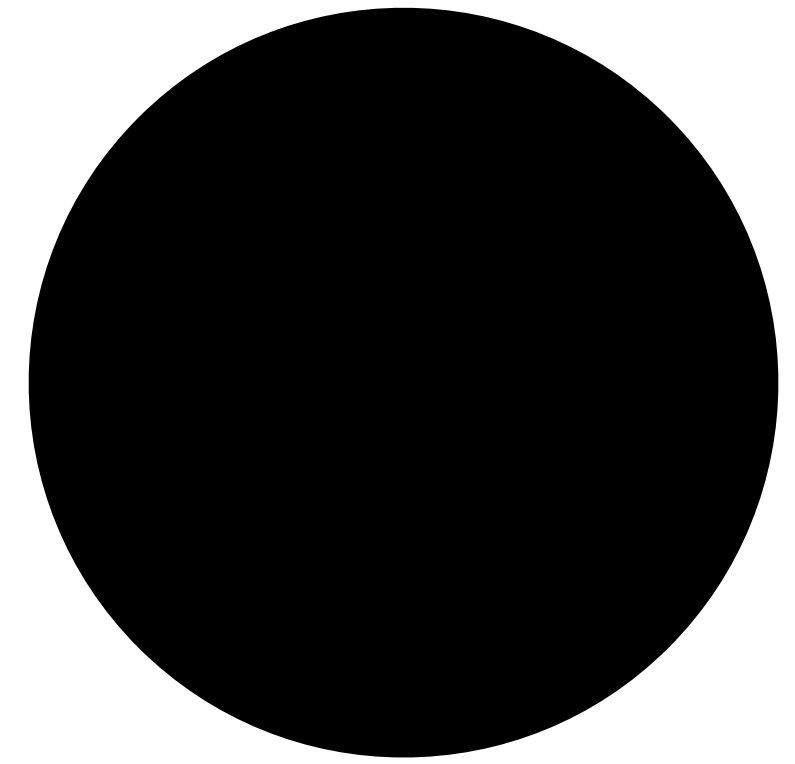
Site Map



Results & Impact



Results & Impact



- User Experience



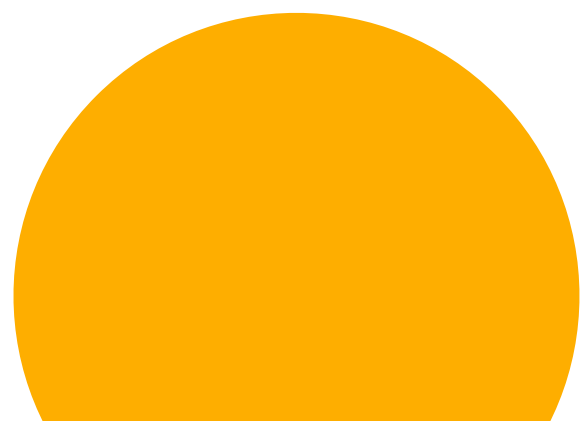
- Simplified Content Management



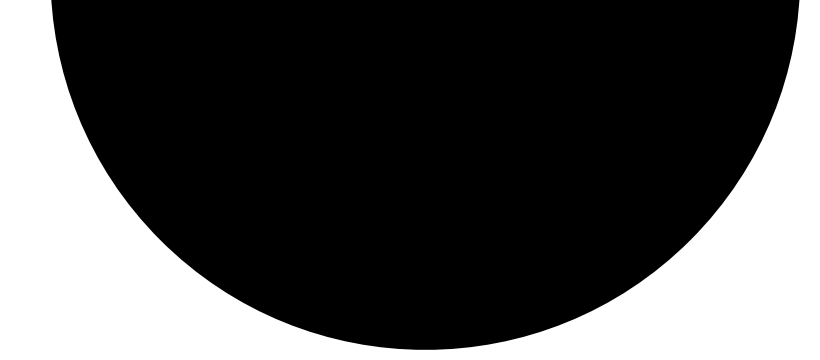
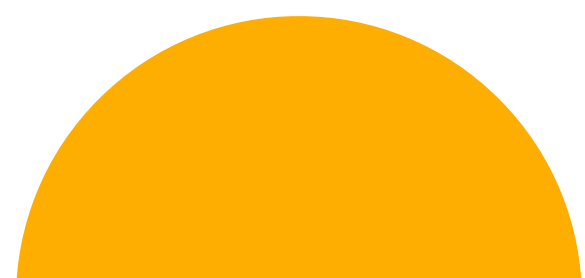
- Maintainability



- Reusable Design



Future Work



Future Work



- Dev Meeting pages



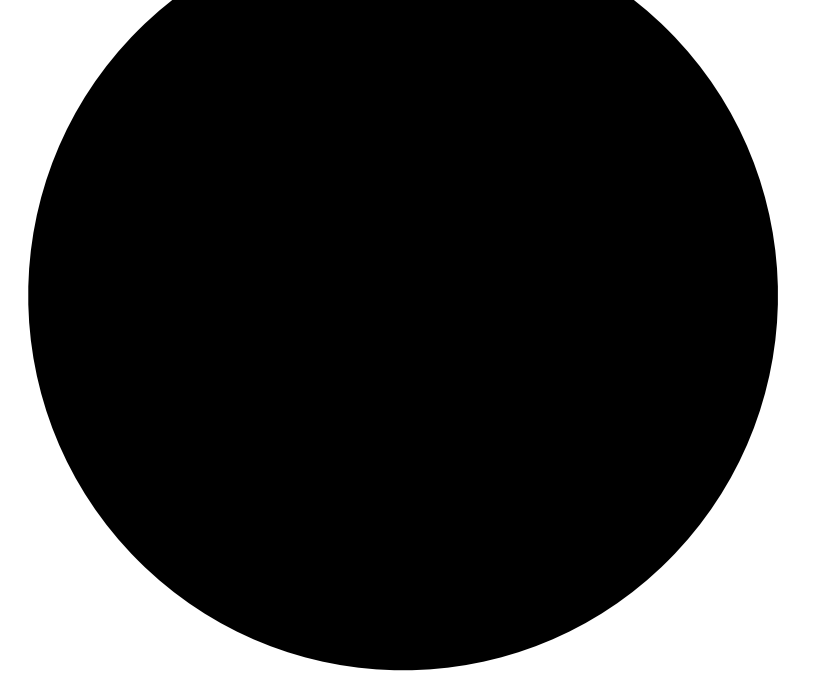
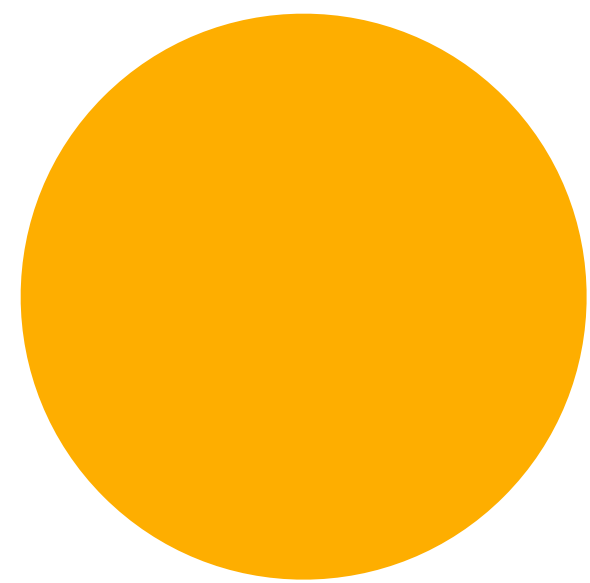
- CI for testing PRs



- Deployment on llvm.org



- Clang Website



Conclusion

Conclusion



- Modernize User Experience



- Increase Engagement



- Simplified Content Management



Acknowledgement



Thank You!

