

link to site:


<https://cse134-hw2-kiminus.netlify.app/>

Purpose of site

the purpose of the site is to host my class notes about CSE 142, and explain the topics about assembly language, computer architecture history, performance optimizations, C++ templates and interpreting code from assembly. the site intended to aid users to understand complex concepts and provide guidance on how to solve typical problems

Audience Considerations

Persona Diagram:



Name: Ale, James

Relevant Quote: "understanding compiler design is key to understand deeper in programming skills"

Demographics

Gender: Male

Age: 20

Location: San Diego, CA

Relationship Status: Single

Title: CS Student

Education: 3rd year undergraduate

Goals

- Understanding assembly language
- use and apply optimization techniques in projects
- help to learn the class and pass exams

Background Description

Ale is a CS student passionate about low level computer design and compiler design. He wants to get good grade on exam and in the mean time deepen the knowledge to apply the coursework knowledge in the future career

Motivations

- Help with homework and exams
- Improve skill on low level programming
- Develop coding skills for assembly language and compiler design

Frustrations

- Difficulty in understanding assembly language
- Don't know how to debug the compiler and how does basic optimization work

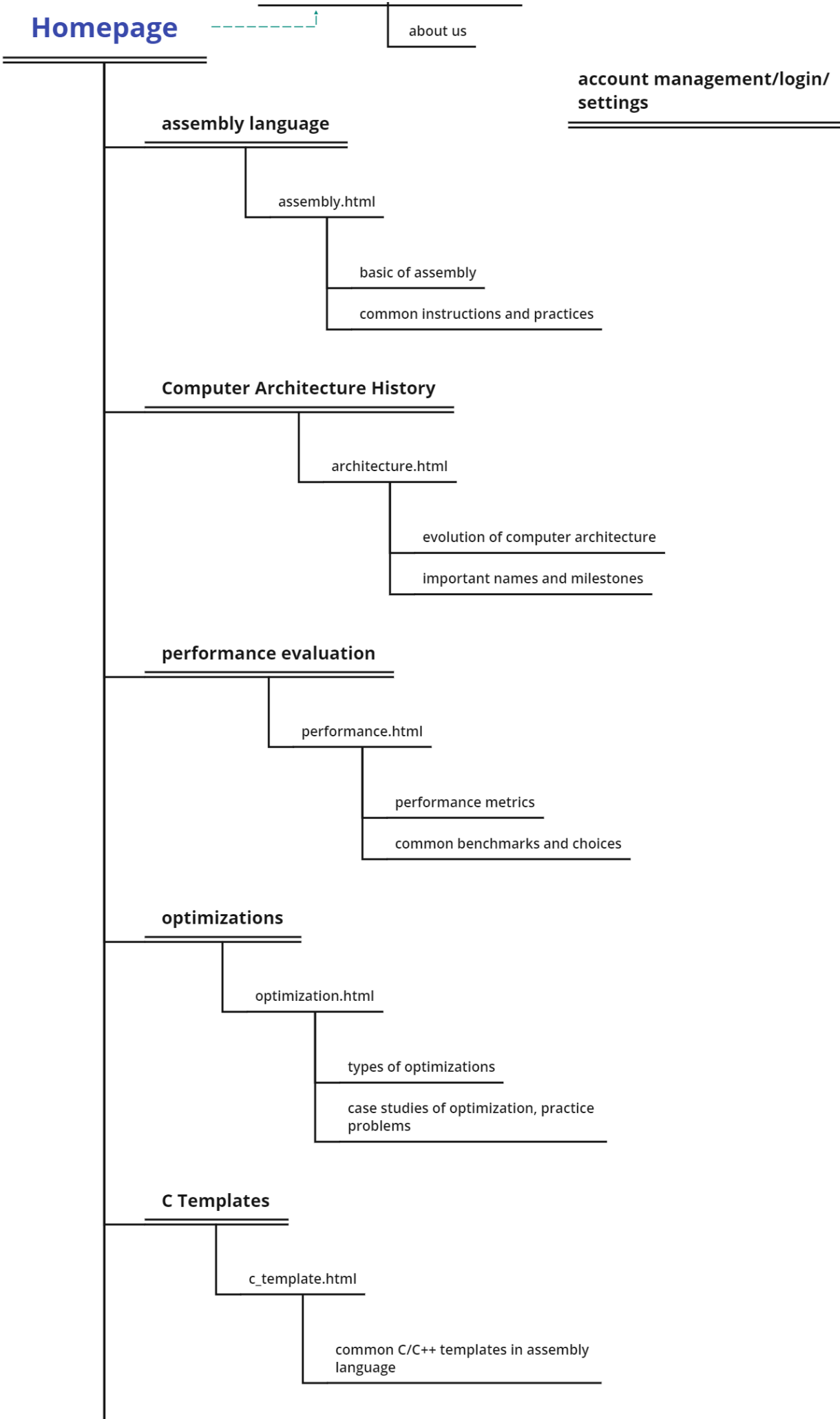
Previous Experience:

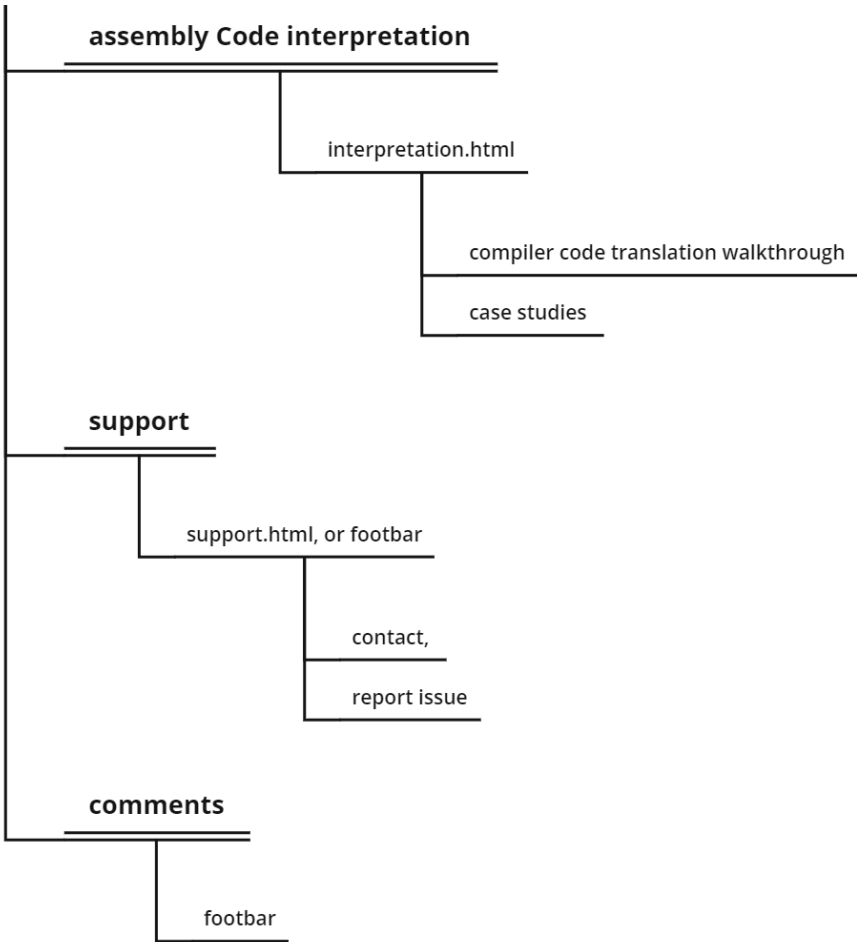
- Completed introduction and intermediate high level programming courses
- Understanding on the mathematic foundation of bitwise operation
- Introductory to assembly language and C programming

Expectations:

- Clear and accurate explanation of assembly language
- Practical problems to help on coursework
-

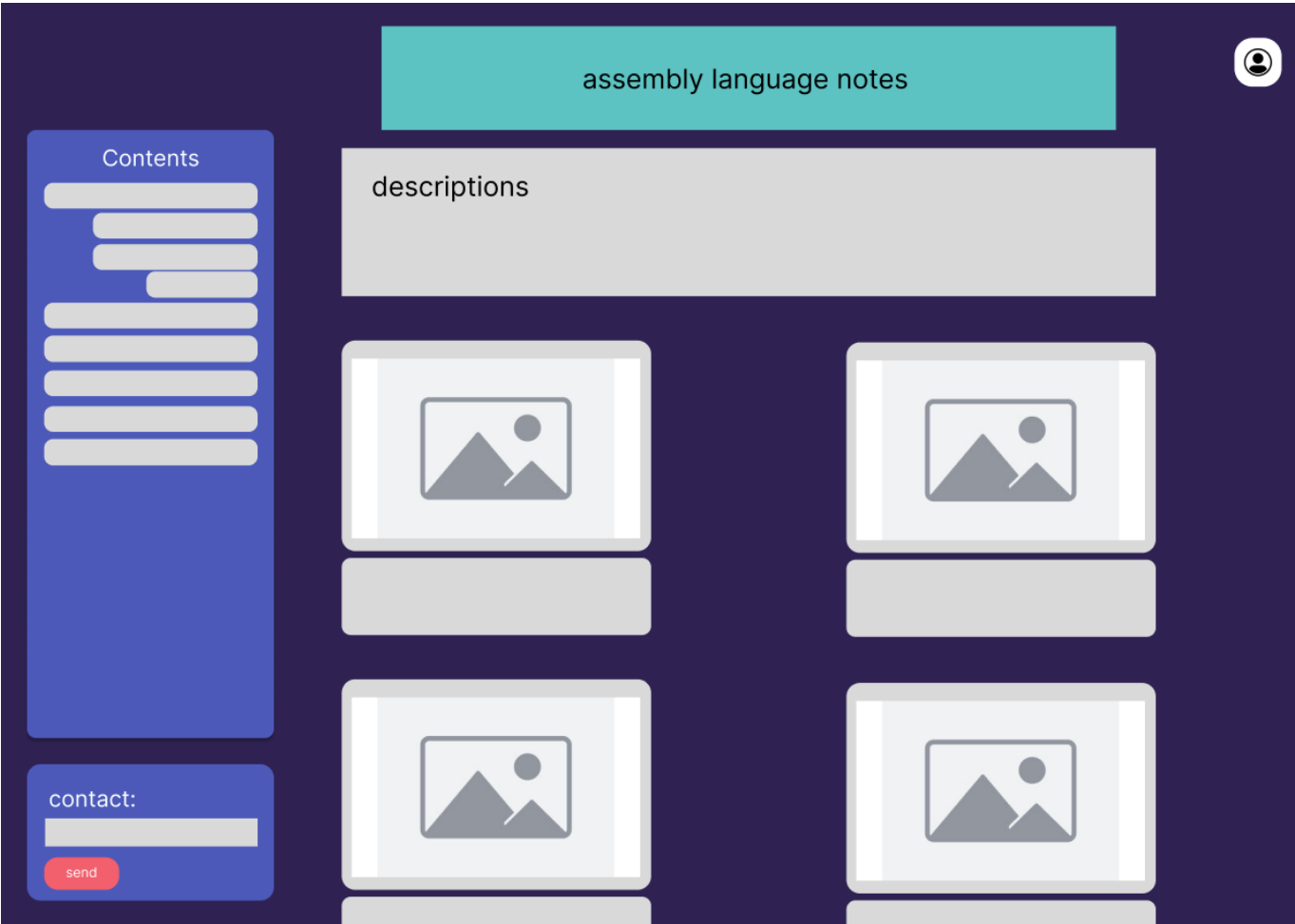
site diagram



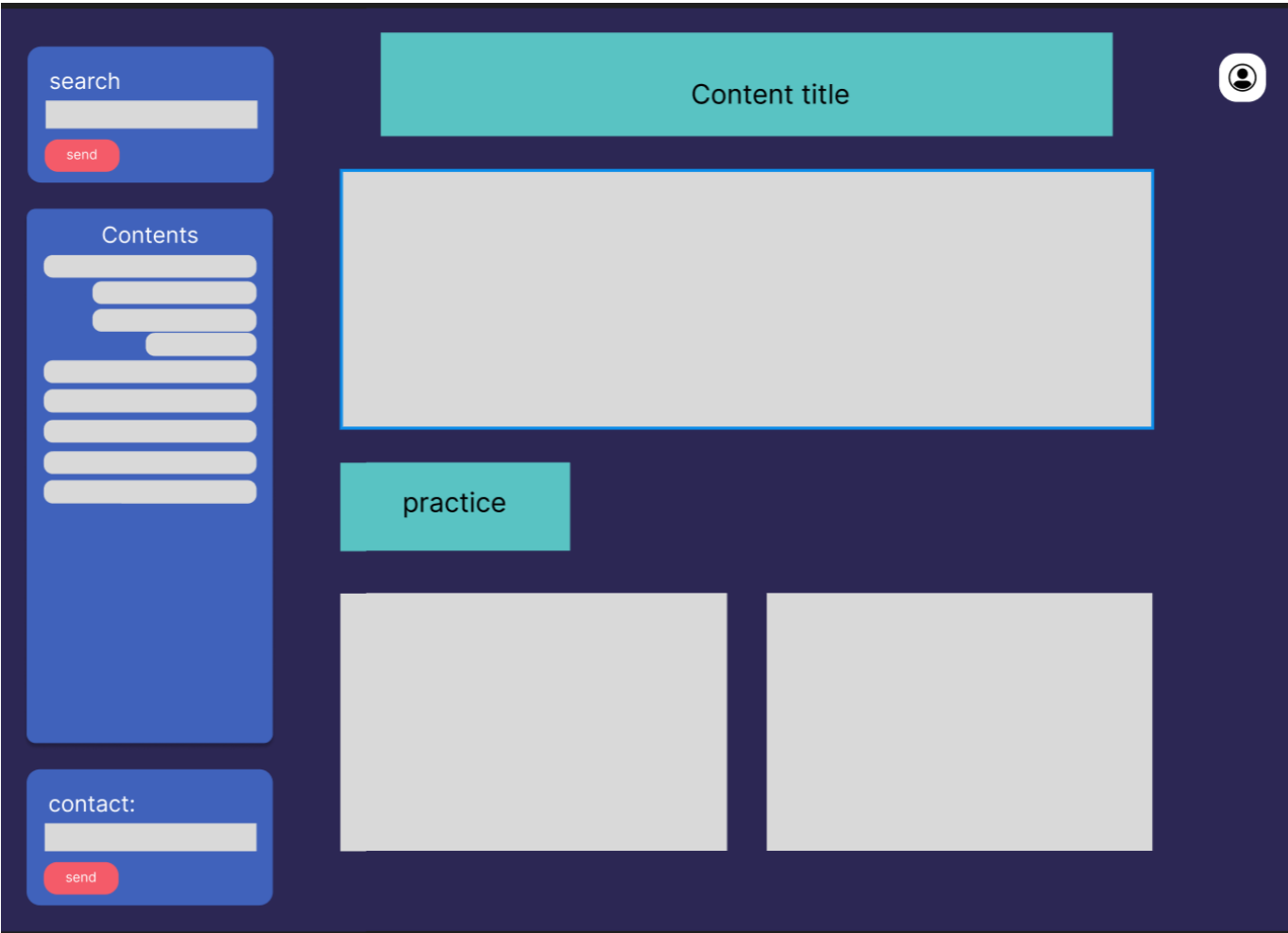


wireframes

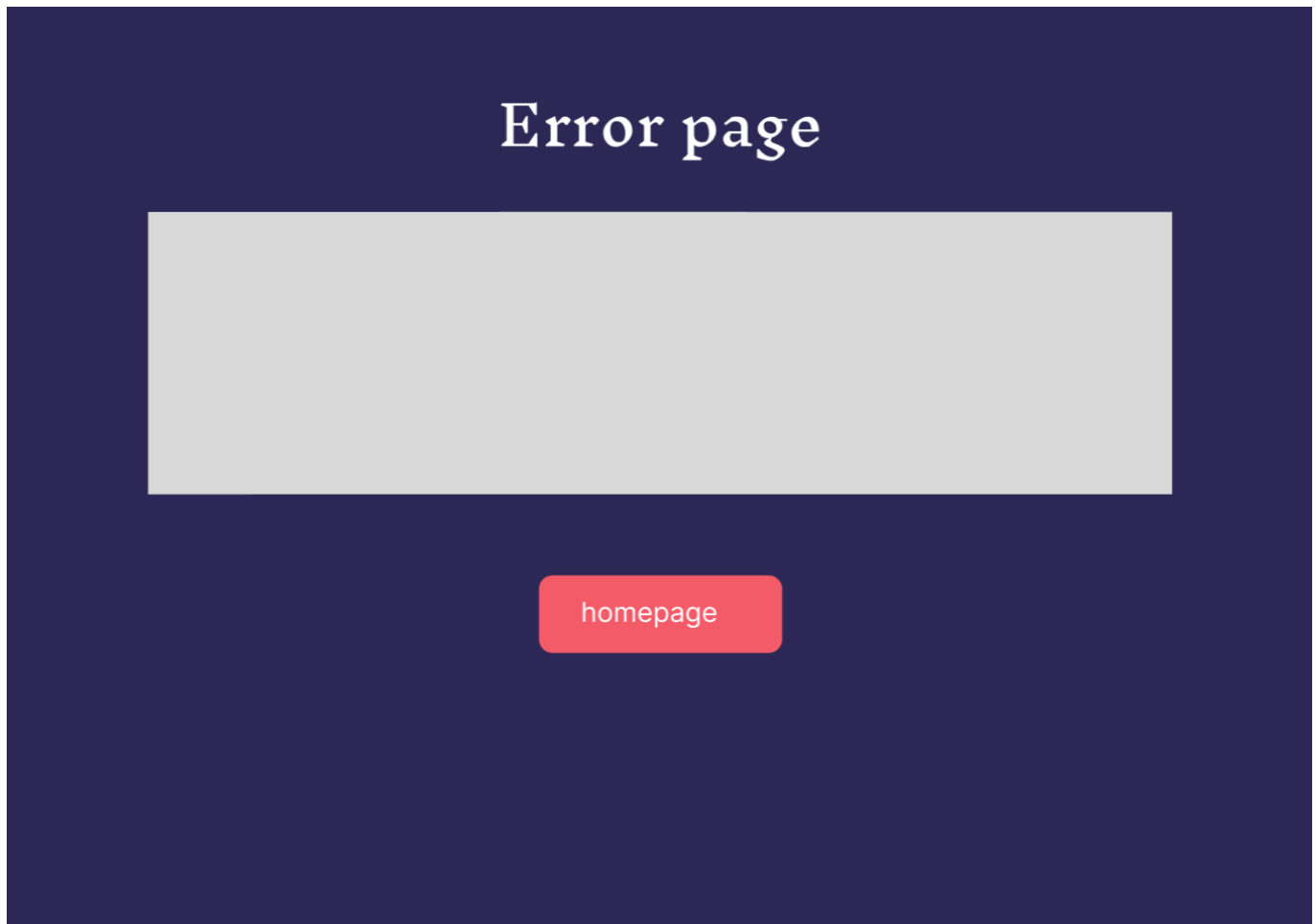
Homepage Wireframe



Common page wireframe



404 page



content requirements

1. will be hosted on netlify
2. site will be built following the practice of progressive enhancement, therefore basic functionalities should be accessible and usable for users without JS/advanced CSS. the site will be using HTML for structure. in the future, we will use typescript instead of JS to ensure site reliability and code maintainability
3. github is used for version control and future collaborations
4. target version should to latest version of chrome, or the earliest version in each browser that supports all the HTML elements implemented in part 2
5. the site is intended for laptop view, but it should feature in responsive design to support mobile device and tablets, although with limited aesthetic
6. any page should be less than 10mb, since there are many images to load.
7. the site should be accesible using only keyboard navigation

Content requirements

1. the site should contain text content, including class notes, lecture summaries, and others
2. there will be chart and images to support learning
3. some sample questions to provide practice examples
4. external links for reference and further explanation

Presentation Quality

1. content side should always have a header, concept introduction, details, and practice question.
2. every site should use different level headers to provide clear format and breakdown of topics.
3. use coolers.co to create a harmonic color scheme and more consistent color pattern
4. use w3 validator to secure code quality and correctness
5. a roadmap and wireframe to ensure the development is planned and developed according to plan
6. the content is designed for the intended audience to assist in learning
7. the platform of presentation is primarily on laptop screen.

EC

math equation

math equation

$$E = mc^2$$

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mi>E</mi>
  <mo>=</mo>
  <mi>m</mi>
  <msup>
    <mi>c</mi>
    <mn>2</mn>
  </msup>
</math>
```

Site Validation

here is the validation report from w3 validator

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://cse134-hw2-kiminus.netlify.app/>

Checker Input

Show

☒ source

☒ outline

☐ image report

Options...

Check by address

https://cse134-hw2-kiminus.netlify.app/

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

1. **Error** Element `search` not allowed as child of element `main` in this context. (Suppressing further errors from this subtree.)

From line 44, column 9; to line 44, column 16

><

`<search>`

<>

Contexts in which element `search` may be used:

Where `flow content` is expected.

Content model for element `main`:

`Flow content`.

Document checking completed.

Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 66 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 25.1.7

tries to validate:

- 1. move search inside the body tag, still throw same error
- 2. move search inside the main tag, still throw same error
- 3. the error thrown may be due to outdated support (or the validator is not updated) to recognize and validate the search tag
- 4. according to the [dev site](#), the search tag is newly introduced at 2023, so the validator may not be updated.
- 5. in the dev doc, it is suggested to put search tag inside the header element, as what I did.

meta for search engine:

prevent robot/search engine to find this page:

```
<meta name="keywords" content="compiler design, assembly language, C++,
performance, computer architecture">
<meta property="og:title" content="Compiler Instruction & Design">
<meta name="robots" content="noindex, nofollow">
```

8 / 8