

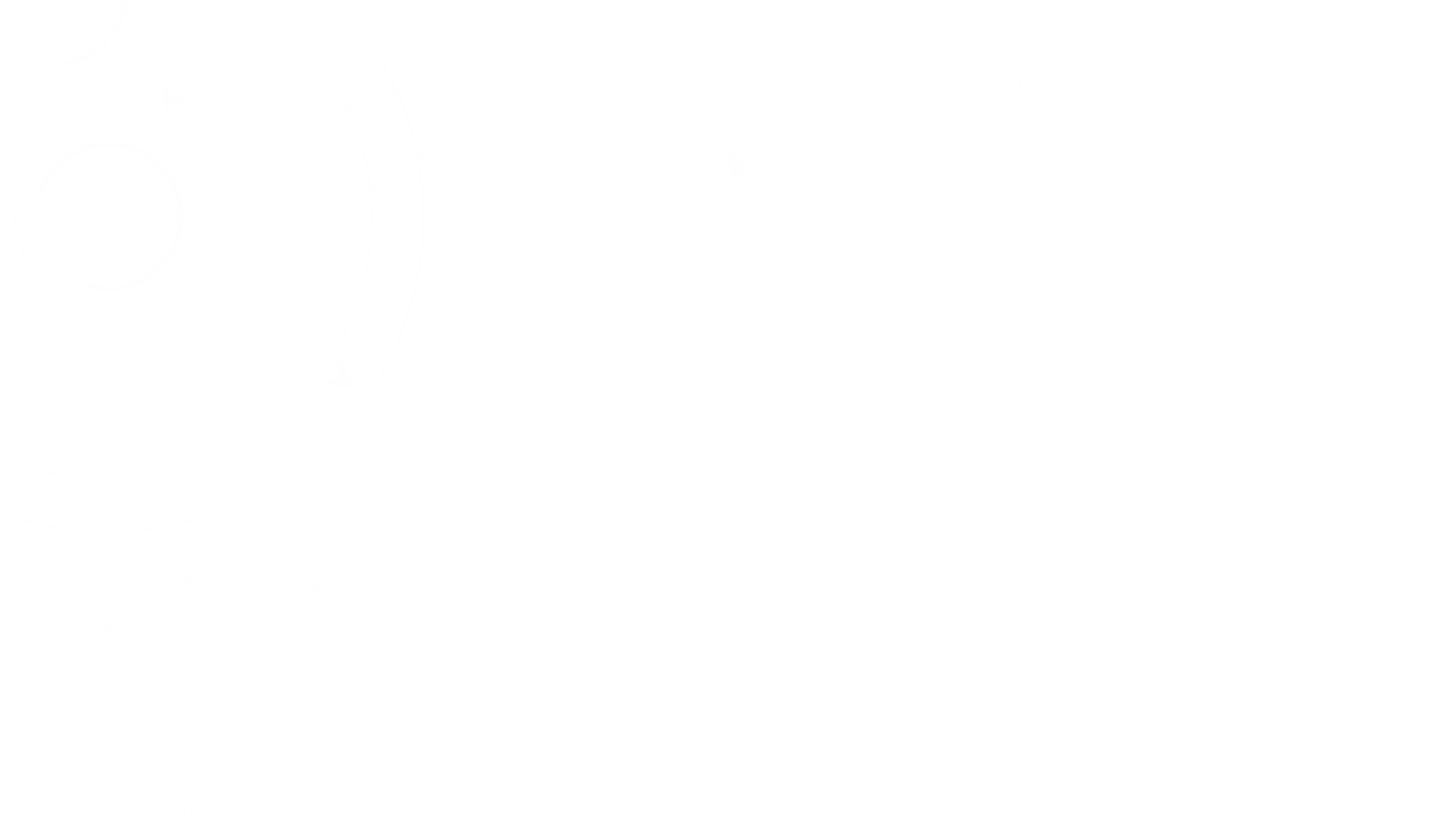
L&T

EDUTECH

FRONT END

UI UX

MINIPROJECT



PROJECT 1:PERSONAL PORTFOLIO WEBSITE

PROJECT 2: RECIPE BLOG TEMPLATE

SUBMITTED BY :

AMAL

KJ (2460323) ,

AMAL.KJ@BTECH.CHRISTUNIVERSITY.IN

AUSTIN K

LUKOSE

(2460342)

,

AUSTIN

.K@BTECH.CHRISTUNIVERSITY.IN

TOVIN

TOM (2460468),

TOVIN.TOM@BTECH.CHRISTUNIVERSITY.IN

COURSE: UI UX

FUNDEMENTALS

INSTRUCTOR NAME :NARENDRA

DATE OF SUBMISSION

:

**Abstract**

This project entails the development of a personal portfolio website aimed at showcasing an individual's professional journey, skills, and projects. The primary goal is to create an engaging, user-friendly platform that effectively communicates the individual's expertise and experiences. Utilizing core technologies such as HTML5 for structure and CSS3 for styling, the website ensures responsiveness across various devices. The final outcome is a polished, interactive online presence that serves as a digital resume. This portfolio not only highlights technical proficiency but also enhances online visibility, making it a valuable tool for professional networking and career advancement.



# Objective 1.Design a User-Friendly Interface Using Modern UI Principles

The primary objectives of this project are:

Create an intuitive and visually appealing layout that enhances user experience. **2.Develop a Fully Responsive Layout Using Only HTML and CSS**

Construct a flexible and adaptive design that ensures optimal viewing across various devices, including desktops, tablets, and smartphones.

# 3.Implement Structured HTML5 Semantic Elements

Utilize HTML5 semantic elements such as <header>, <nav>, <main>, <section>, and <footer> to enhance the document's structure and accessibility.

**4.Apply CSS Styling for Branding, Layout, and Responsive Behavior** Employ CSS to define the visual appearance of the website, including color schemes, typography, and layout. Implement responsive design techniques to ensure the website adapts to various screen sizes and devices.



# 5.Ensure Accessibility and Readability Across Devices

Design the website to be accessible to users with disabilities by adhering to web accessibility standards. Ensure that text is legible on all devices by choosing appropriate font sizes and contrast ratios.



**Scope of the Project**

This project focuses exclusively on front

-

end development, utilizing only HTML5 and CSS3 to create a responsive,

visually appealing personal portfolio website. The scope includes:

•

**Front**

**-**

**End Design Only**

:

The project encompasses the design and layout of the website, without any back

-

end or

server

-

side integration.

•

**No JavaScript or Server**

**-**

**Side Integration**

:

The website operates entirely on the client side, ensuring fast load times

and simplicity.

•

**Responsive Across Devices**

:

The layout is designed to be fully responsive, providing an optimal viewing experience

on desktop, tablet, and mobile devices.

•

**Open**

**-**

**Source Tools and Pure Code**

:

Development is carried out using open

-

source tools and pure code, without

relying on external libraries or frameworks.

This approach ensures a lightweight, accessible, and easily maintainable website that effectively showcases personal

skills and projects.



NOLOGIES USED

TOOLS AND TECH

**Tools/Technology**

**purpose**

HTML5

For structuring the content and ensuring

semantic markup.

**CSS3**

:

For styling the website, including layout,

typography, and animations.

**Font Awesome**

:

For incorporating scalable vector icons.

**Google Fonts**

:

For custom typography to enhance

readability and aesthetics.

GitHub Pages

For hosting the website online.

Visual Studio Code

As the code editor for development.



**HTML Structure Overview**

The website’s structure is built with clean, semantic HTML5, ensuring clarity,

maintainability, and accessibility. At the top, the

<

header

>

tag contains the site's

branding and the navigation menu. Within this, a

<

nav

>

element houses a

<

ul

>

list of

anchor links, enabling smooth scrolling between sections. The main content resides in the

<

main

>

tag, which is divided into distinct, reusable

<

section

>

elements such as

**About**

,

**Projects**

, and

**Contact**

. Each

<

section

>

typically begins with a heading (e.g.,

<

h

2>

)

to define its purpose clearly

—

this aligns with best practices that encourage semantic

labeling for screen readers, search engines, and maintainable page structure

[W3School](https://www.w3schools.com/html/html5_semantic_elements.asp?source=post_page-----35b4309ea06f----------------------&utm_source=chatgpt.com)

[s](https://www.w3schools.com/html/html5_semantic_elements.asp?source=post_page-----35b4309ea06f----------------------&utm_source=chatgpt.com)

[accionlabs.co](https://www.accionlabs.com/blogs/structure-content-using-html-semantic-tags?utm_source=chatgpt.com)

[m](https://www.accionlabs.com/blogs/structure-content-using-html-semantic-tags?utm_source=chatgpt.com)

[Reddi](https://www.reddit.com/r/webdev/comments/ilx73j?utm_source=chatgpt.com)

[t](https://www.reddit.com/r/webdev/comments/ilx73j?utm_source=chatgpt.com)

[.](https://www.reddit.com/r/webdev/comments/ilx73j?utm_source=chatgpt.com)

Finally, the

<

footer

>

provides site

-

wide information

like contact details and legal disclaimers. Using these semantic tags not only enhances

readability but also improves SEO and accessibility

[FreeCodeCam](https://www.freecodecamp.org/news/semantic-html5-elements/?utm_source=chatgpt.com)

[p](https://www.freecodecamp.org/news/semantic-html5-elements/?utm_source=chatgpt.com)

[R](https://www.reddit.com/r/TechSEO/comments/1i2mdph?utm_source=chatgpt.com)

[eddi](https://www.reddit.com/r/TechSEO/comments/1i2mdph?utm_source=chatgpt.com)

[t](https://www.reddit.com/r/TechSEO/comments/1i2mdph?utm_source=chatgpt.com)

.

**CSS Styling Strategy**

The styling is managed via a **single external stylesheet** (style.css), promoting a clear separation between HTML structure and visual presentation—an approach that enhances maintainability and performance [allthingsprogramming.com](https://allthingsprogramming.com/css-best-practices-for-clean-and-maintainable-code/?utm_source=chatgpt.com). The file is

systematically organized with **commented sections** to differentiate areas such as base styles, layout rules, and responsive overrides, improving readability and navigation [jassweb.com](https://jassweb.com/14-css-best-practices-for-beginners/?utm_source=chatgpt.com)[Reddit.](https://www.reddit.com/r/css/comments/11mkdio?utm_source=chatgpt.com)

For layout construction, **Flexbox and CSS Grid** are employed—Flexbox for onedimensional alignments like navigation and section layouts, and Grid for twodimensional constructs such as project galleries [Medium](https://cloudengineer.medium.com/css3-best-practices-how-to-write-better-faster-and-more-efficient-css-d0de5a7b01dc?utm_source=chatgpt.com)[Reddit.](https://www.reddit.com/r/css/comments/ici1ib?utm_source=chatgpt.com) The design follows a **mobile-first approach**, using media queries only to enhance presentation on larger viewports [allthingsprogramming.com](https://allthingsprogramming.com/css-best-practices-for-clean-and-maintainable-code/?utm_source=chatgpt.com)[rtCamp](https://rtcamp.com/handbook/developing-for-block-editor-and-site-editor/css-best-practices/?utm_source=chatgpt.com). Breakpoints are logically grouped to avoid repetition and streamline responsiveness [Daily.dev](https://daily.dev/blog/css-best-practices-for-clean-code?utm_source=chatgpt.com)[webitronix.com](https://webitronix.com/css-layout-techniques/?utm_source=chatgpt.com). This method ensures clean, scalable, and efficient styling suited for current web standards.



**Feature**

**Description**

Fully Responsive Layout

Using HTML and CSS

The design adapts seamlessly across desktop, tablet, and mobile

devices using responsive web design techniques like media

queries and flexible grids, ensuring usability on any screen

[s](https://www.geeksforgeeks.org/websites-apps/responsive-web-design/?utm_source=chatgpt.com)

[GeeksforGeek](https://www.geeksforgeeks.org/websites-apps/responsive-web-design/?utm_source=chatgpt.com)

[M](https://medium.com/%40cssmonster007/getting-better-at-html-and-css-useful-tips-3938b37fa23f?utm_source=chatgpt.com)

[ediu](https://medium.com/%40cssmonster007/getting-better-at-html-and-css-useful-tips-3938b37fa23f?utm_source=chatgpt.com)

[m](https://medium.com/%40cssmonster007/getting-better-at-html-and-css-useful-tips-3938b37fa23f?utm_source=chatgpt.com)

.

Clean and Intuitive User

Interface

Minimalist design and consistent visual elements create a

smooth, clutter

-

free user experience

—

guiding visitors

effortlessly through the content

[a](https://pageoneformula.com/the-impact-of-css-and-html-best-practices-on-ux?utm_source=chatgpt.com)

[Page One Formul](https://pageoneformula.com/the-impact-of-css-and-html-best-practices-on-ux?utm_source=chatgpt.com)

.

Semantic HTML5 Structure

for Accessibility

Employs semantic tags such as <header>, <nav>, <main>,

<

section>, and <footer> to enhance accessibility, readability,

and SEO

[y](https://algocademy.com/blog/the-ultimate-guide-to-building-responsive-websites-that-work-on-all-devices/?utm_source=chatgpt.com)

[AlgoCadem](https://algocademy.com/blog/the-ultimate-guide-to-building-responsive-websites-that-work-on-all-devices/?utm_source=chatgpt.com)

[g](https://basescripts.com/comprehensive-guide-to-web-design?utm_source=chatgpt.com)

[odin](https://basescripts.com/comprehensive-guide-to-web-design?utm_source=chatgpt.com)

[C](https://basescripts.com/comprehensive-guide-to-web-design?utm_source=chatgpt.com)

[s](https://basescripts.com/comprehensive-guide-to-web-design?utm_source=chatgpt.com)

[Help Tips Resources Tutorial](https://basescripts.com/comprehensive-guide-to-web-design?utm_source=chatgpt.com)

.

Organized CSS with

Flexbox and Grid Layouts

Flexbox supports straightforward alignment and navigation

layouts; CSS Grid powers more complex layouts like project

galleries

—

both contributing to responsive adaptability

[AlgoCadem](https://algocademy.com/blog/the-ultimate-guide-to-building-responsive-websites-that-work-on-all-devices/?utm_source=chatgpt.com)

[y](https://algocademy.com/blog/the-ultimate-guide-to-building-responsive-websites-that-work-on-all-devices/?utm_source=chatgpt.com)

[M](https://medium.com/%40nile.bits/html-hacks-10-tips-to-build-better-websites-083f96295921?utm_source=chatgpt.com)

[ediu](https://medium.com/%40nile.bits/html-hacks-10-tips-to-build-better-websites-083f96295921?utm_source=chatgpt.com)

[m](https://medium.com/%40nile.bits/html-hacks-10-tips-to-build-better-websites-083f96295921?utm_source=chatgpt.com)

.

Smooth Navigation via

Anchor Links

A navigation menu using an unordered list (<ul>) and anchor

links enables smooth scrolling between sections for an intuitive

browsing flow.



|  |  |
| --- | --- |
| **challenges** | **solution** |
| Ensuring Consistent  Responsiveness Across All  Screen Sizes | Adapting layouts for desktops, tablets, and mobile devices required meticulous fine-tuning. Achieving this without  JavaScript meant relying heavily on responsive design patterns, flexible units (percentages, vw/vh), and careful breakpoint management. Developers also tested across real devices and emulators to uncover and address inconsistencies.  [Reddit](https://www.reddit.com/r/webdev/comments/1c1hjw9?utm_source=chatgpt.com)[BrowserStack](https://www.browserstack.com/guide/responsive-web-design-challenges?utm_source=chatgpt.com) |
| Managing Layout Complexity  Without JavaScript or  Frameworks | Without interactive scripting or CSS frameworks, layouts— especially multi-column or grid-based ones—needed to be entirely handcrafted. Flexbox and CSS Grid solved many challenges elegantly (akin to the classic "Holy Grail" layout), creating flexible, robust structures. [Wikipedia](https://en.wikipedia.org/wiki/Holy_grail_%28web_design%29?utm_source=chatgpt.com) |
| Maintaining Semantic  Structure While Achieving  Visual Design Goals | Balancing aesthetics with accessible, meaningful HTML5 markup demanded discipline. Semantic elements (headings, sections, etc.) were used together with accessible color contrast and legible typography to ensure usability without compromising design. |
| Debugging Layout Issues Using Chrome DevTools | DevTools proved indispensable for diagnosing spacing,  Flex/Grid alignment, overflow, and z-index issues. Tools like the Box Model visualizer, Flexbox/Grid overlays, and device mode allowed for rapid, effective debugging |



**Outcomes**

•

Demonstrated ability to create a responsive website using only

HTML and CSS

•

Applied semantic HTML5 elements for structured and accessible

content

•

Used CSS Flexbox and Grid to build adaptive layouts

•

Ensured cross

-

device compatibility through media queries

•

Developed a clean, user

-

friendly interface aligned with UI/UX

principles

•

Strengthened understanding of front

-

end development tools and

workflows



**Future Enhancements**

**1.**

**Integrate JavaScript for Interactive Elements**

Adding features like form validation and animations can significantly enhance user engagement and usability.

**2.**

**Add Backend Support for Dynamic Content and Form Submissions**

Embedding a backend

—

such as a lightweight Backend

-

as

-

a

-

Service (BaaS)

—

makes it possible to handle form

submissions securely and update content in real time

[Wikipedi](https://en.wikipedia.org/wiki/Backend_as_a_service?utm_source=chatgpt.com)

[a](https://en.wikipedia.org/wiki/Backend_as_a_service?utm_source=chatgpt.com)

.

**3.**

**Implement Dark Mode Toggle**

Offering a dark mode improves user comfort and personalization, appealing to user preference and modern UI

trends.

**4.**

**Use CSS Variables and Preprocessors like SASS for Scalable Styling**

SCSS (SASS) brings robust features

—

including variables, nesting,

mixins

, and partials

—

that boost maintainability,

promote modularity, and support scalable design systems

[AvailableDe](https://availabledev.com/blog/what-is-sass-in-web-development/?utm_source=chatgpt.com)

[v](https://availabledev.com/blog/what-is-sass-in-web-development/?utm_source=chatgpt.com)

[U](https://www.upwork.com/resources/what-is-scss?utm_source=chatgpt.com)

[pwor](https://www.upwork.com/resources/what-is-scss?utm_source=chatgpt.com)

[k](https://www.upwork.com/resources/what-is-scss?utm_source=chatgpt.com)

[.](https://www.upwork.com/resources/what-is-scss?utm_source=chatgpt.com)

**5.**

**Expand Portfolio with Real**

**-**

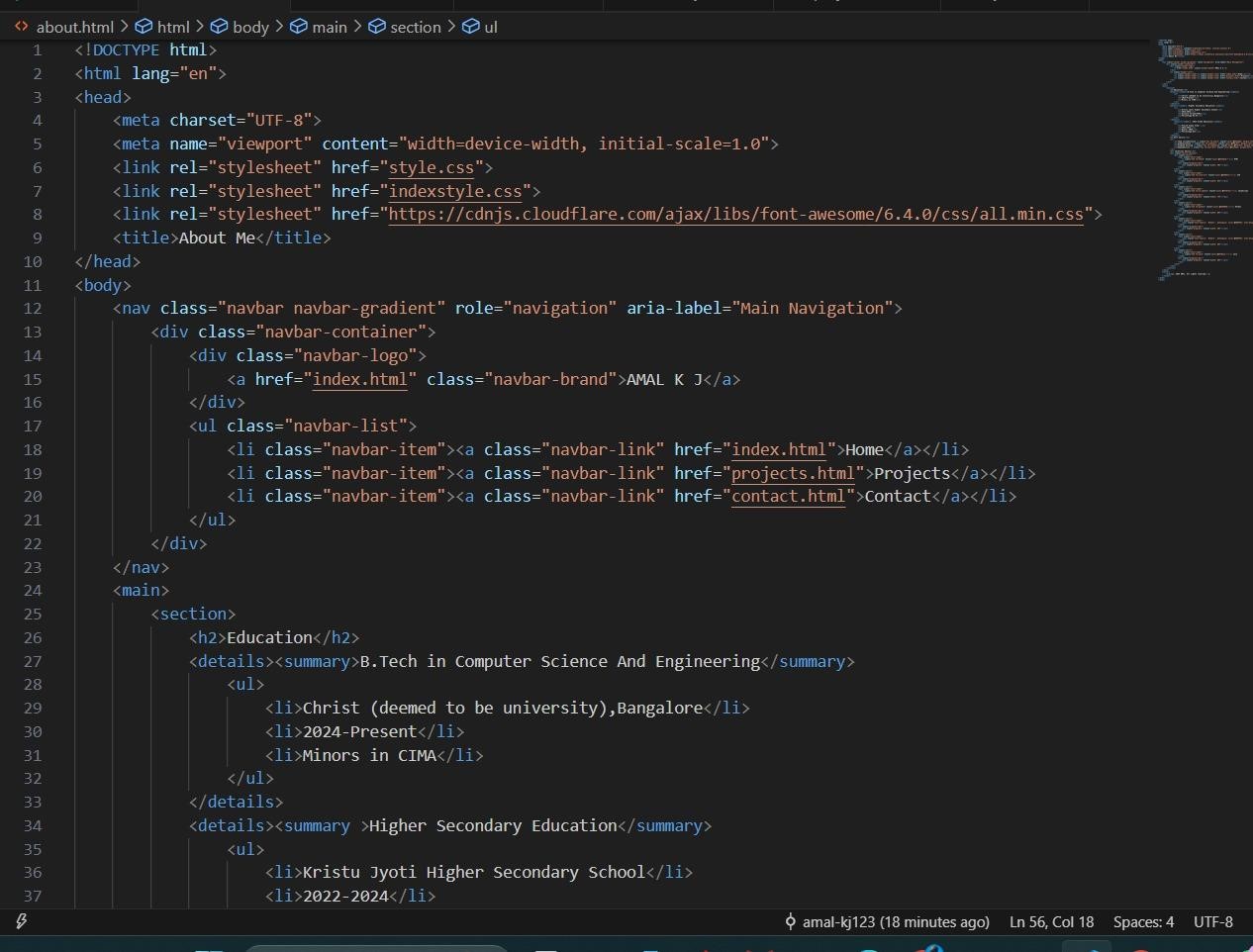
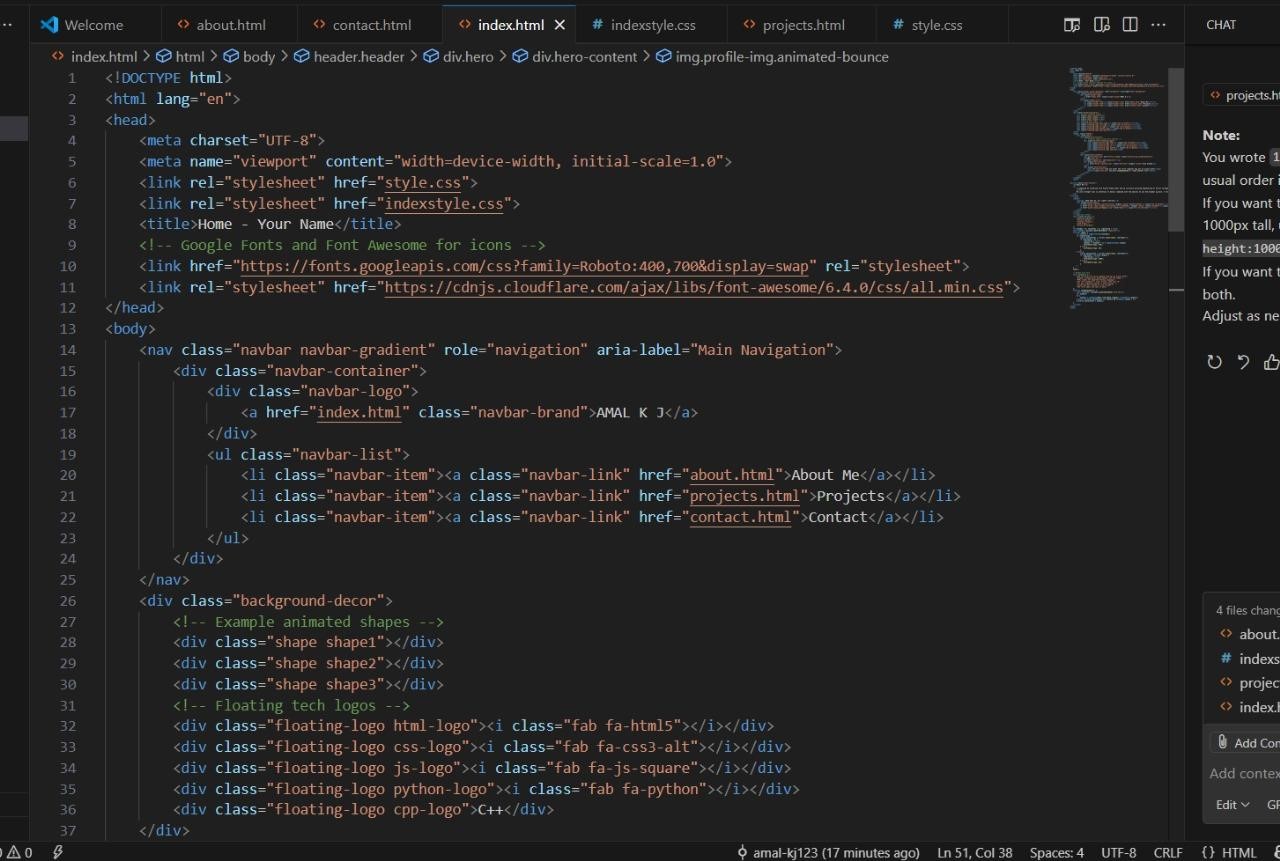
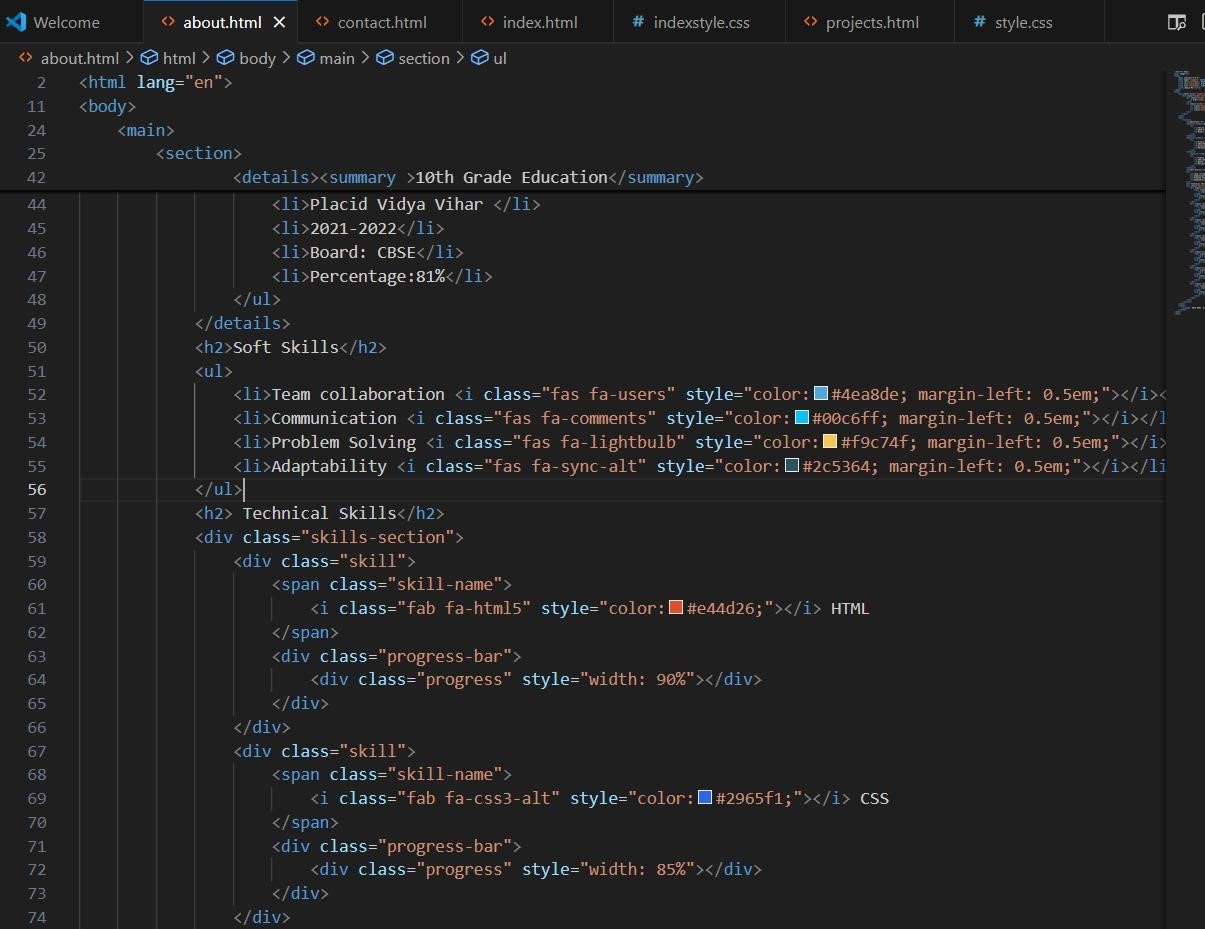
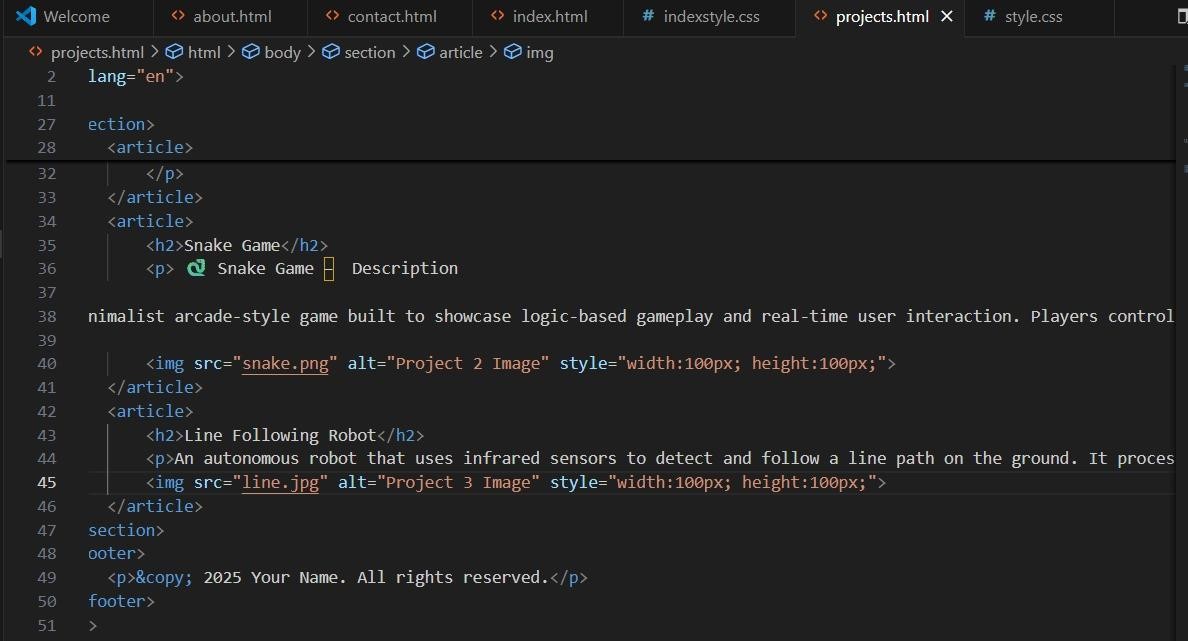
**Time Project Updates**

With a backend in place, you can dynamically display project statuses or new additions, keeping content current and

interactive.



SAMPLE CODE



HTTPS://GITHUB.COM/AMAL

-

KJ123/HTML

-

CSS

-

PROJECT

-

1

.GIT

## Conclusion

This project demonstrates strong front-end fundamentals by crafting a responsive, accessible, and visually appealing personal portfolio using only HTML and CSS. It showcases a solid grasp of modern design principles, from semantic structure and clean layout techniques (Flexbox, Grid) to accessibility best practices—an essential foundation for successful front-end development [Wikipedia](https://en.wikipedia.org/wiki/Front-end_web_development?utm_source=chatgpt.com)[Impressico.](https://www.impressico.com/blog/the-importance-of-front-end-development?utm_source=chatgpt.com) By focusing exclusively on HTML and CSS, the project reinforces core web skills valued in the industry, especially when creating lightweight, maintainable websites [Reddit+1](https://www.reddit.com/r/webdev/comments/ovf1vm?utm_source=chatgpt.com). As the site adapts smoothly across devices and uses structured markup for clarity and SEO, it serves as a standout example of foundational front-end craftsmanship. Overall, this portfolio not only reflects technical proficiency and design sensibility but also lays a strong groundwork for future enhancements—such as introducing interactivity or dynamic content—making it both a functional showcase and a platform for growth.



**References**

•

Importance of front

-

end development and

responsive design

[MoldStu](https://moldstud.com/articles/p-the-importance-of-front-end-development-in-creating-user-friendly-websites?utm_source=chatgpt.com)

[d](https://moldstud.com/articles/p-the-importance-of-front-end-development-in-creating-user-friendly-websites?utm_source=chatgpt.com)

[Wikipedi](https://en.wikipedia.org/wiki/Responsive_web_design?utm_source=chatgpt.com)

[a](https://en.wikipedia.org/wiki/Responsive_web_design?utm_source=chatgpt.com)

•

Benefits of HTML and CSS for portfolio

websites

[ConsumerSearch.co](https://www.consumersearch.com/technology/benefits-using-html-css-create-portfolio-website?utm_source=chatgpt.com)

[m](https://www.consumersearch.com/technology/benefits-using-html-css-create-portfolio-website?utm_source=chatgpt.com)

•

Value of semantic structure and accessibility

[Wikipedi](https://en.wikipedia.org/wiki/Web_accessibility?utm_source=chatgpt.com)

[a](https://en.wikipedia.org/wiki/Web_accessibility?utm_source=chatgpt.com)

[FreeCodeCam](https://www.freecodecamp.org/news/best-practices-for-accessibility-in-web-development/?utm_source=chatgpt.com)

[p](https://www.freecodecamp.org/news/best-practices-for-accessibility-in-web-development/?utm_source=chatgpt.com)

•

Perspectives on using plain HTML/CSS without

frameworks

[Reddit+](https://www.reddit.com/r/webdev/comments/nbfml9?utm_source=chatgpt.com)

[1](https://www.reddit.com/r/webdev/comments/nbfml9?utm_source=chatgpt.com)



Program 2

Recipe Blog Template

# Abstract

This project centers on crafting a **Recipe Blog Template** using pure HTML5 and CSS3. It aims to offer a clean, responsive, and semantically organized platform to showcase culinary content such as ingredient lists, preparation steps, and recipe descriptions. Core technologies include semantic HTML5 for structure and CSS Flexbox/Grid for responsive design. The end result is an accessible, intuitive template—optimized for desktop, tablet, and mobile —that functions as a polished foundation for any food-related blog. It serves as both a visual and functional template, ideal for food enthusiasts and bloggers.



**Objective**

The primary objectives are:

**1.**

**Design a user**

**-**

**friendly interface using modern UI principles**

**2.**

**Develop a fully responsive layout using only HTML and CSS**

**3.**

**Implement structured HTML5 semantic elements**

**4.**

**Apply CSS styling for branding, layout, and responsive behavior**

**5.**

**Ensure accessibility and readability across devices**



**Scope of the Project**

•

**Front**

**-**

**End Design Only**

:

Focused exclusively on the

HTML and CSS layer; no JavaScript or server

-

side logic

involved.

•

**Device Compatibility**

:

Fully optimized for desktop,

tablet, and mobile viewports.

•

**Pure Code**

:

No external frameworks or libraries;

relies purely on open

-

source tools and custom code.



**Tools and Technologies**

**purpose**

**HTML5**

:

For clean and semantic structure

CSS3

implements layout (Flexbox, Grid), layout

responsiveness, typography, and color

schemes

Flexbox & Grid

For creating complex, flexible layouts

responsive across screens

Responsive Design Practices

Using media queries and fluid layouts to

adapt to different devices



**HTML Structure Overview**

The template uses semantic HTML5 tags to ensure clarity and accessibility:

•

<

header

>

and

<

nav

>

for branding and navigation links

•

<

main

>

wraps multiple

<

section

>

s such as Featured Recipes, Ingredient Lists, and Instructions

•

A

<

footer

>

contains author or copyright info

This layout aids screen readers, improves SEO, and supports maintainable code structure.



**CSS Styling Strategy**

•

Single external stylesheet (

style.css

)

with commented sections for structure, layout, and responsiveness

•

**Flexbox**

used for one

-

dimensional layouts like ingredient lists

•

**Grid**

applied for structured recipe galleries

•

**Media Queries**

ensure layouts adapt smoothly across viewports

[The Linux Cod](https://thelinuxcode.com/html-css-tutorial-build-a-recipe-website/?utm_source=chatgpt.com)

[e](https://thelinuxcode.com/html-css-tutorial-build-a-recipe-website/?utm_source=chatgpt.com)

•

Mobile

-

first styling enhances performance and user experience

|  |  |
| --- | --- |
| **Key Features** | **Description** |
| Fully Responsive Design | Optimized for smooth viewing across desktop, tablet, and mobile devices. Layouts adapt gracefully using CSS Flexbox, Grid, and media queries for a seamless reading experience. |
| Semantic HTML5 Structure | Utilizes meaningful tags like <header>, <nav>, <main>, <section>, and <footer> to improve accessibility, SEO, and maintainability, following web standards for structured markup. |
| Organized CSS with Flexbox & Grid | Uses Flexbox for ingredient lists and smaller layout  segments, and CSS Grid for complex structures like  photo galleries or recipe cards. This allows for clean,  flexible styling and responsive layouts  [Wikipedi](https://en.wikipedia.org/wiki/CSS_grid_layout?utm_source=chatgpt.com)  [a](https://en.wikipedia.org/wiki/CSS_grid_layout?utm_source=chatgpt.com)  .  **Smooth Navigation with "Jump to Recipe" Links** |
| Content-Centric Design with Visuals | Features structured layout for ingredients, method, prep times, and servings. Though simple, it supports expansion with images and clear labels to prioritize usability and clarity |



**Challenges Faced**

**Solution**

Responsive consistency

Overcame inconsistencies across devices

using fluid units and careful breakpoint

testing

Layout complexity

Achieved clean layout without JavaScript

by combining Flexbox and Grid

Semantic vs. visual balance

Maintained accessibility while ensuring

appealing visuals

Debugging

Used Chrome

DevTools

to fix alignment,

spacing, and mobile layout issues



**Outcome**

This project delivers a

**clean, visually appealing, and**

**accessible Recipe Blog Template**

, demonstrating

proficiency in front

-

end development using just

HTML and CSS. It provides an effective and

lightweight framework for content presentation

—

ready to be customized or extended by bloggers and

developers alike.



***https://github.com/amal***

***-***

***kj123/HTML***

***-***

***CSS***

***-***

***PROJECT***

***-***

***2***



**Future Enhancements**

•

Introduce

**JavaScript**

for interactive elements

like ingredient toggles or animation

•

Add

**backend integration**

for saving and

retrieving recipes dynamically

•

Apply a

**dark mode toggle**

for better reader

experience

•

Use

**CSS preprocessors**

like SASS and variables

for scalable and maintainable styling

•

Expand templates with

**real**

**-**

**time updates**

for

recipe comments or featured posts



**Conclusion**

This Recipe Blog Template showcases solid front

-

end

skills

—

creating a responsive, visually clean, and semantic

framework using only HTML and CSS. It serves as a

robust foundation that could easily evolve with

JavaScript or backend functionality for richer interactivity

and content management.



**References**

•

Recipe page layout concepts and

mobile

-

first grid patterns

[The Linux Cod](https://thelinuxcode.com/html-css-tutorial-build-a-recipe-website/?utm_source=chatgpt.com)

[e](https://thelinuxcode.com/html-css-tutorial-build-a-recipe-website/?utm_source=chatgpt.com)

•

CSS Grid usage for responsive layout

design