

**1. Name of Practicum:** Static HTML Webpage**2. Procedure ( Say)**

1. Begin with a standard HTML5 structure.
2. Use <h1>, <p>, <b>, and <img> tags to display personal details and an image.
3. Use <a> tags to simulate website and social media links.
4. Use headings and paragraphs to format text.
5. Use the <table> tag with <tr>, <th>, and <td> elements.
6. Use an ordered list (<ol>) to present technical skills.
7. Use an unordered list (<ul>) for general items.
8. Complete and save the document.

**3. Actual Resources Used ( Say )**

<b>Resources</b>	<b>Tool &amp; Version</b>
Operating System	<b>Macos 15.6 Sequoia</b>
Programming Language	<b>HTML ( Markup Language)</b>
Software Development Kit	<b>Not Applicable</b>
Libraries	<b>Not Applicable</b>
Graphics/Animation Tool	<b>Not Applicable</b>
Front-end Tool	<b>1.VS Code ( v 1.103.1) 2. Google Chrome ( v 139.0.7258.67)</b>
Back-end Tool	<b>Not Applicable</b>
Virtual Lab	<b>Not Applicable</b>
Specific Hardware	<b>Not Applicable</b>

Simulation/Emulation	<b>Not Applicable</b>
----------------------	-----------------------

#### **4. Actual Procedure Followed:**

##### Planning & Requirements Gathering

- Identified the purpose of the webpage
- Listed the required features.
- Chose the theme, layout style, and color scheme.

##### Setting Up the Development Environment:

- Installed/used a code editor (VS Code).
- Created a project folder and subfolders for HTML and images.

##### Designing the Structure:

- Wrote the basic HTML skeleton.
- Added headings, paragraphs, images, and hyperlinks as per the design.

##### Testing & Debugging

- Opened the webpage in multiple browsers (Chrome, Edge, etc.) to check compatibility.
- Fixed layout issues and corrected broken links or missing images.

##### Finalizing & Deployment

- Reviewed the webpage for errors and typos.
- Saved and organized all files in the correct folders.
- The Tags used for the forms .

#### **5. Observations:**

- Different tags were used to give different output and perform different tasks in html document.
- These pages source code has been verified by <https://validator.w3.org/> and found errors of CSS not being used.
- Noticed how different tags rendered on the webpage.
- `<td>` displayed table data, while `<th>` emphasized header content with bold styling.
- Changing the type attribute in the ordered list (`<ol>`) affected the numbering style (e.g., numeric, alphabetic, Roman numerals).
- The `<head>` section is used for metadata, title, and linking styles/scripts, while the `<body>` contains visible content.
- Semantic tags like `<header>`, `<nav>`, `<section>`, and `<footer>` improve readability and accessibility.
- Lists (`<ul>`, `<ol>`, `<li>`) help in presenting information in an organized manner.
- The main thing of this practicum is adding tags to the webpage and it shows perfectly.

## 6. Source Code:

### I)Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Flex Layout – Introduction & Properties</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
      margin: 2rem;
      color: #333;
    }
    h1, h2 {
      color: #2c3e50;
    }
    code {
      background: #f4f4f4;
      padding: 0.2rem 0.4rem;
      font-family: Consolas, monospace;
    }
    ul {
      margin: 1rem 0 2rem 1.2rem;
    }
    li {
      margin-bottom: 0.5rem;
    }
    section {
      margin-bottom: 3rem;
    }
  </style>
</head>
<body>

  <h1>Flex Layout</h1>

  <section id="introduction">
    <h2>Introduction</h2>
    <p>
      Flexbox is a one-dimensional CSS layout model designed to distribute space
      along a single axis—either as a row or a column. It simplifies tasks like
      centering, spacing, and ordering items, even when their size is unknown or dynamic.
      To enable Flexbox, set <code>display: flex</code> (or <code>inline-flex</code>)
      on a container and then control alignment and distribution with a handful of properties.
    </p>
  </section>

  <section id="container-properties">
    <h2>Flex Container Properties</h2>
    <ul>
      <li>
        <code>display</code>: <code>flex</code> | <code>inline-flex</code> <a href="flex-display-
        demo.html">Link</a>
      </li>
      <li>
        <code>flex-direction</code>: <code>row</code> | <code>row-reverse</code> |
        <code>column</code> | <code>column-reverse</code> <a href="flex-direction-demo.html">Link</a>
      </li>
    </ul>
  </section>
</body>
```

```

</li>
<li>
  <code>flex-wrap</code>: <code>nowrap</code> | <code>wrap</code> |
  <code>wrap-reverse</code> <a href="flex-wrap-demo.html">Link</a>
</li>
<li>
  <code>flex-flow</code>: <em>shorthand for</em>
  <code>flex-direction</code> & amp; <code>flex-wrap</code>
  (<code>&lt;direction&gt; &lt;wrap&gt;</code>) <a href="flex-flow-demo.html">Link</a>
</li>
<li>
  <code>justify-content</code>: <code>flex-start</code> | <code>center</code> |
  <code>flex-end</code> | <code>space-between</code> |
  <code>space-around</code> | <code>space-evenly</code> <a href="justify-content-
demo.html">Link</a>
</li>
<li>
  <code>align-items</code>: <code>flex-start</code> | <code>center</code> |
  <code>flex-end</code> | <code>stretch</code> | <code>baseline</code> <a href="align-items-
demo.html">Link</a>
</li>
<li>
  <code>align-content</code>: <code>flex-start</code> | <code>center</code> |
  <code>flex-end</code> | <code>space-between</code> | <code>space-around</code> |
  <code>stretch</code> <a href="align-content-demo.html">Link</a>
</li>
<li>
  <code>gap</code>, <code>row-gap</code>, <code>column-gap</code>:
  <em>length values (e.g. <code>1rem</code>, <code>10px</code>)</em> <a href="gap-
demo.html">Link</a>
</li>
</ul>
</section>

<section id="item-properties">
<h2>Flex Item Properties</h2>
<ul>
<li>
  <code>order</code>: <em>integer</em> (default <code>0</code>)
  — controls the visual order of items <a href="order-demo.html">Link</a>
</li>
<li>
  <code>flex-grow</code>: <em>non-negative number</em> (default <code>0</code>)
  — defines how much an item will grow relative to its siblings <a href="flex-grow-demo.html">Link</
a>
</li>
<li>
  <code>flex-shrink</code>: <em>non-negative number</em> (default <code>1</code>)
  — defines how much an item will shrink relative to its siblings <a href="flex-shrink-
demo.html">Link</a>
</li>
<li>
  <code>flex-basis</code>: <code>auto</code> | <em>length</em> | <code>content</code>
  (default <code>auto</code>) — sets the initial main size <a href="flex-basis-demo.html">Link</a>
</li>
<li>
  <code>flex</code>: <em>shorthand</em> for
  <code>flex-grow</code>, <code>flex-shrink</code>, <code>flex-basis</code>
  (e.g. <code>flex: 1 0 100px</code>) <a href="flex-shorthand-demo.html">Link</a>
</li>

```

```

</li>
  <code>align-self</code>: <code>auto</code> | <code>flex-start</code> |
  <code>center</code> | <code>flex-end</code> | <code>baseline</code> |
  <code>stretch</code> — overrides the container's <code>align-items</code>
  for a single item <a href="align-self-demo.html">Link</a>
</li>
</ul>
</section>

</body>
</html>

```

## **II Karunya Map Navigator with Flex Elements:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Karunya University Campus Guide</title>
  <link rel="icon" href="karunya.png">
  <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap"
rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      font-family: 'Inter', -apple-system, BlinkMacSystemFont, sans-serif;
      background: #CCDDD3;
      color: #2d3748;
      line-height: 1.6;
      min-height: 100vh;
      padding: 20px;
    }

    .container {
      max-width: 900px;
      margin: 0 auto;
      background: rgba(255, 255, 255, 0.95);
      backdrop-filter: blur(10px);
      border-radius: 20px;
      box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);
      overflow: hidden;
      animation: fadeInUp 0.8s ease-out;
    }

    @keyframes fadeInUp {
      from {
        opacity: 0;
        transform: translateY(30px);
      }
      to {
        opacity: 1;
        transform: translateY(0);
      }
    }
  </style>

```

```
}  
}  
  
.header {  
  background:#EEF4D4;  
  color: white;  
  padding: 40px 30px;  
  text-align: center;  
  position: relative;  
  overflow: hidden;  
}  
  
.header::before {  
  content: " ";  
  position: absolute;  
  top: -50%;  
  left: -50%;  
  width: 200%;  
  height: 200%;  
  background: radial-gradient(circle, rgba(255,255,255,0.1) 0%, transparent 70%);  
}  
  
h1 {  
  font-size: 2.5rem;  
  font-weight: 700;  
  margin-bottom: 10px;  
  position: relative;  
  z-index: 1;  
  color:#000000;  
}  
  
h2 {  
  font-size: 1.2rem;  
  font-weight: 400;  
  opacity: 0.9;  
  position: relative;  
  z-index: 1;  
  color:#000000;  
}  
  
.content {  
  padding: 40px 30px;  
}  
  
.sections-container {  
  display: flex;  
  flex-wrap: wrap;  
  gap: 20px;  
  justify-content: space-between;  
}  
  
.audio-guide {  
  background: #000000;  
  border-radius: 15px;  
  padding: 25px;  
  margin-bottom: 30px;  
  text-align: center;  
  color: white;  
  box-shadow: 0 10px 25px rgba(240, 147, 251, 0.3);  
}
```

```
.audio-guide p {
  font-size: 1.1rem;
  font-weight: 500;
  margin-bottom: 15px;
}
```

```
.audio-guide i {
  font-size: 1.5rem;
  margin-right: 10px;
}
```

```
audio {
  width: 100%;
  max-width: 400px;
  border-radius: 25px;
  outline: none;
}
```

```
.section {
  background: white;
  margin: 0;
  padding: 25px;
  border-radius: 15px;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);
  border-left: 4px solid;
  transition: all 0.3s ease;
  position: relative;
  overflow: hidden;
  flex: 1 1 calc(50% - 10px);
  min-width: 300px;
}
```

```
.section-title {
  font-size: 1.3rem;
  font-weight: 600;
  margin-bottom: 12px;
  display: flex;
  align-items: center;
  color: #2d3748;
}
```

```
.comment-section .section-title {
  color: #28a745;
}
```

```
.section-title i {
  margin-right: 10px;
  font-size: 1.2rem;
}
```

```
.section p {
  color: #4a5568;
  margin-bottom: 15px;
  font-size: 1rem;
}
```

```
.loc {
```

```
display: inline-flex;
align-items: center;
background: linear-gradient(135deg, #ffecd2 0%, #fcb69f 100%);
color: #744210;
padding: 8px 15px;
border-radius: 20px;
font-size: 0.9rem;
font-weight: 500;
margin-top: 10px;
}
```

```
.loc i {
  margin-right: 8px;
  font-size: 0.9rem;
}
```

```
ul {
  margin: 15px 0;
  padding-left: 0;
}
```

```
li {
  list-style: none;
  padding: 8px 0;
  padding-left: 25px;
  position: relative;
  color: #4a5568;
}
```

```
li::before {
  content: '\2022';
  color: #000000;
  font-weight: bold;
  position: absolute;
  left: 0;
  font-size: 1.2rem;
}
```

```
@media (max-width: 768px) {
  body {
    padding: 10px;
  }
}
```

```
.content {
  padding: 20px 15px;
}
```

```
.header {
  padding: 30px 20px;
}
```

```
h1 {
  font-size: 2rem;
}
```

```
.section {
  padding: 20px 15px;
  flex: 1 1 100%;
}
```

```
.sections-container {
  flex-direction: column;
}

.scroll-indicator {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 4px;
  background: linear-gradient(90deg, #4f46e5, #7c3aed);
  transform-origin: left;
  transform: scaleX(0);
  z-index: 1000;
}

.dm-serif-text-regular {
  font-family: "DM Serif Text", serif;
  font-weight: 400;
  font-style: normal;
}

.comment-section {
  background: #f8f9fa;
  margin: 30px 0;
  padding: 25px;
  border-radius: 15px;
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);
  border-left: 4px solid #28a745;
}

.comment-form {
  margin-top: 15px;
}

.comment-box {
  width: 100%;
  padding: 15px;
  border: 2px solid #e9ecef;
  border-radius: 10px;
  font-family: inherit;
  font-size: 1rem;
  line-height: 1.5;
  resize: vertical;
  transition: border-color 0.3s ease;
}

.comment-box:focus {
  outline: none;
  border-color: #28a745;
  box-shadow: 0 0 0 3px rgba(40, 167, 69, 0.1);
}

.button-group {
  display: flex;
  gap: 15px;
  margin-top: 15px;
}

.submit-btn, .reset-btn {
```

```

padding: 12px 25px;
border: none;
border-radius: 8px;
font-size: 1rem;
font-weight: 600;
cursor: pointer;
transition: all 0.3s ease;
}

.submit-btn {
background: linear-gradient(135deg, #28a745 0%, #20c997 100%);
color: white;
}

.submit-btn:hover {
transform: translateY(-2px);
box-shadow: 0 5px 15px rgba(40, 167, 69, 0.3);
}

.reset-btn {
background: linear-gradient(135deg, #6c757d 0%, #495057 100%);
color: white;
}

.reset-btn:hover {
transform: translateY(-2px);
box-shadow: 0 5px 15px rgba(108, 117, 125, 0.3);
}

```

```
</style>
```

```
</head>
```

```
<body class="dm-serif-text-regular">
```

```
<div class="scroll-indicator"></div>
```

```
<div class="container">
```

```
<div class="header">
```

```
<h1><strong><ins>Karunya University Campus Guide.</ins></strong></h1>
```

```
<h2>Discover Our Academic Excellence & Innovation Hub</h2>
```

```
</div>
```

```
<div class="content">
```

```
<div class="audio-guide">
```

```
<p><i></i><strong></strong><ins>University Anthem</ins><br>Experience the spirit of Karunya
University through our anthem, a melodious tribute to our values and vision.</strong></p>
```

```
<audio controls>
```

```
<source src="KarunyaAnthem.mp3" type="audio/mpeg">
```

```
<source src="KarunyaAnthem.mp3" type="audio/ogg">
```

```
Your browser does not support the audio element.
```

```
</audio>
```

```
</div>
```

```
<div class="sections-container">
```

```
<div class="section">
```

```
<div class="section-title">
```

```
<i></i>Central Building
```

```
</div>
```

```
<p>The heart of our campus where groundbreaking research and academic excellence converge.
This architectural masterpiece welcomes you into an environment of innovation and learning, setting the tone
for your entire university experience.</p>
```

```
</div>
```

```
<div class="section">
```

*Computer Technology Center (CTC)*

The digital backbone of Karunya University, ensuring seamless connectivity and technological infrastructure across our entire campus ecosystem.

- CTC-1:** Strategic location at Central Library's 3rd floor
- CTC-2:** Dedicated facility beyond the Central Library
- CTC-3:** Conveniently positioned near the Main Building

*Computer Science Engineering Block*

A three-story hub of technological innovation where future software engineers and computer scientists develop cutting-edge solutions. Nestled against the scenic Sirivani hills, it provides an inspiring environment for coding and creativity.

*• South campus corner - scenic yet accessible*

*Electronics & Communication Engineering Block*

An expansive facility housing advanced laboratories and research centers, also serving as the home for CTC-3. Here, students explore the frontiers of electronics and communication technologies.

*• Adjacent to the Main Building*

*Mechanical Engineering Department*

Where mechanical innovation comes to life through design, development, and maintenance of sophisticated systems. Students prepare for dynamic careers in automotive, aerospace, energy, and manufacturing industries.

*• East campus, neighboring Emmanuel Auditorium*

*Artificial Intelligence & Machine Learning Department*

The future of technology unfolds here through intelligent systems development. Students dive deep into data science, neural networks, natural language processing, and computer vision to shape tomorrow's AI landscape.

*• Beyond the ECE Block*

*Aerospace Engineering Department*

Where dreams of flight become reality through the design and maintenance of aircraft and spacecraft. Students master aerodynamics, propulsion systems, advanced materials, and flight mechanics.

*• Facing Mechanical Department - perfectly positioned pair*

```

</div>

<div class="section">
  <div class="section-title">
    <i></i>Agricultural Engineering Department
  </div>
  <p>Revolutionizing farming through engineering excellence, focusing on sustainable irrigation systems, advanced machinery, soil management, and innovative post-harvest technologies for global food security.</p>
  <span><i></i>• En route to CSE block, left side</span>
</div>

<div class="section">
  <div class="section-title">
    <i></i>Food Processing Department
  </div>
  <p>Transforming raw ingredients into high-quality food products through advanced processing techniques, with expertise in preservation, innovative packaging, and comprehensive quality assurance systems.</p>
  <span><i></i>• Strategically between AI&ML and ECE blocks</span>
</div>
</div>

<div class="comment-section">
  <div class="section-title">
    Share Your Thoughts
  </div>
  <p>We'd love to hear your feedback about Karunya University!</p>
  <textarea class="comment-box" placeholder="Write your comments here..." rows="4"></
textarea>
  <div class="button-group">
    <button type="submit" class="submit-btn"><a href="signcss.html">submit</a></button>
    <button type="reset" class="reset-btn">Reset</button>
  </div>
</form>
</div>
</div>
</div>
</body>
</html>

```

### **III) Mom's Kitchen with Flex Positions:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>LILLY's Kitchen | Simple Recipes</title>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
      background: #f2f2f2;
      color: #333;
    }

    /* Header */
    header {
      background: #fff;
      padding: 15px;
    }

```

```
text-align: center;
border-bottom: 2px solid #ccc;
position: fixed;
top: 0;
width: 100%;
}
header h1 {
margin: 0;
font-size: 24px;
}
```

```
/* Contact icons */
.contact-icons {
position: absolute;
top: 15px;
right: 20px;
}
.contact-icons img {
width: 30px;
height: 30px;
margin-left: 10px;
}
```

```
/* Page container */
.container {
max-width: 1200px;
margin: 100px auto 20px;
padding: 10px;
background: #fff;
}
```

```
/* Recipe container */
.recipes-container {
display: flex;
gap: 20px;
flex-wrap: wrap;
justify-content: space-between;
}
```

```
/* Recipe index */
.recipe-index {
margin: 20px 0;
text-align: center;
}
.recipe-index a {
display: inline-block;
background: #ccc;
color: #000;
padding: 6px 12px;
margin: 5px;
text-decoration: none;
border-radius: 4px;
}
.recipe-index a:hover {
background: #999;
color: #fff;
}
```

```
/* Recipe cards */
.recipe {
```

```

border: 1px solid #ccc;
padding: 15px;
background: #fafafa;
flex: 1 1 calc(33.333% - 15px);
min-width: 280px;
display: flex;
flex-direction: column;
}
.recipe h2 {
margin: 0 0 10px 0;
color: #c00;
}
.recipe img {
width: 100%;
height: 200px;
object-fit: cover;
margin-bottom: 10px;
}
.recipe h3 {
margin: 10px 0 5px 0;
}
.recipe ul, .recipe ol {
margin: 0;
padding-left: 20px;
}

/* Form */
.form-section {
border: 1px solid #ccc;
padding: 15px;
margin: 20px 0;
background: #fafafa;
}
textarea {
width: 100%;
padding: 8px;
margin-bottom: 10px;
}
button {
padding: 8px 16px;
margin-right: 10px;
border: none;
background: #333;
color: #fff;
cursor: pointer;
}
button:hover {
background: #666;
}
</style>
</head>
<body>

<!-- Header -->
<header>
<h1>LILLY's Kitchen</h1>
<div class="contact-icons">
<a href="mailto:headhunter1608@email.com"></a>
<a href="tel:9949728819"></a>
</div>

```

```
</header>

<!-- Content -->
<div class="container">

  <!-- Recipe Index -->
  <div class="recipe-index">
    <a href="#chickenfry">Chicken Fry</a>
    <a href="#gulabjamun">Gulab Jamun</a>
    <a href="#mangolassi">Mango Lassi</a>
  </div>

  <div class="recipes-container">
    <!-- Chicken Fry -->
    <div class="recipe" id="chickenfry">
      <h2>Recipe 1: Chicken Fry</h2>
      
      <h3>Ingredients:</h3>
      <ul>
        <li>500g chicken</li>
        <li>2 tbsp chili powder</li>
        <li>1 tbsp turmeric</li>
        <li>1 tbsp salt</li>
        <li>Oil for frying</li>
      </ul>
      <h3>Steps:</h3>
      <ol>
        <li>Marinate chicken with spices.</li>
        <li>Heat oil in pan.</li>
        <li>Deep fry until golden brown.</li>
        <li>Serve hot.</li>
      </ol>
    </div>

    <!-- Gulab Jamun -->
    <div class="recipe" id="gulabjamun">
      <h2>Recipe 2: Gulab Jamun</h2>
      
      <h3>Ingredients:</h3>
      <ul>
        <li>1 cup milk powder</li>
        <li>2 tbsp flour</li>
        <li>1 tbsp ghee</li>
        <li>Sugar syrup</li>
        <li>Oil for frying</li>
      </ul>
      <h3>Steps:</h3>
      <ol>
        <li>Mix ingredients into dough.</li>
        <li>Shape into small balls.</li>
        <li>Fry until golden brown.</li>
        <li>Soak in sugar syrup.</li>
      </ol>
    </div>

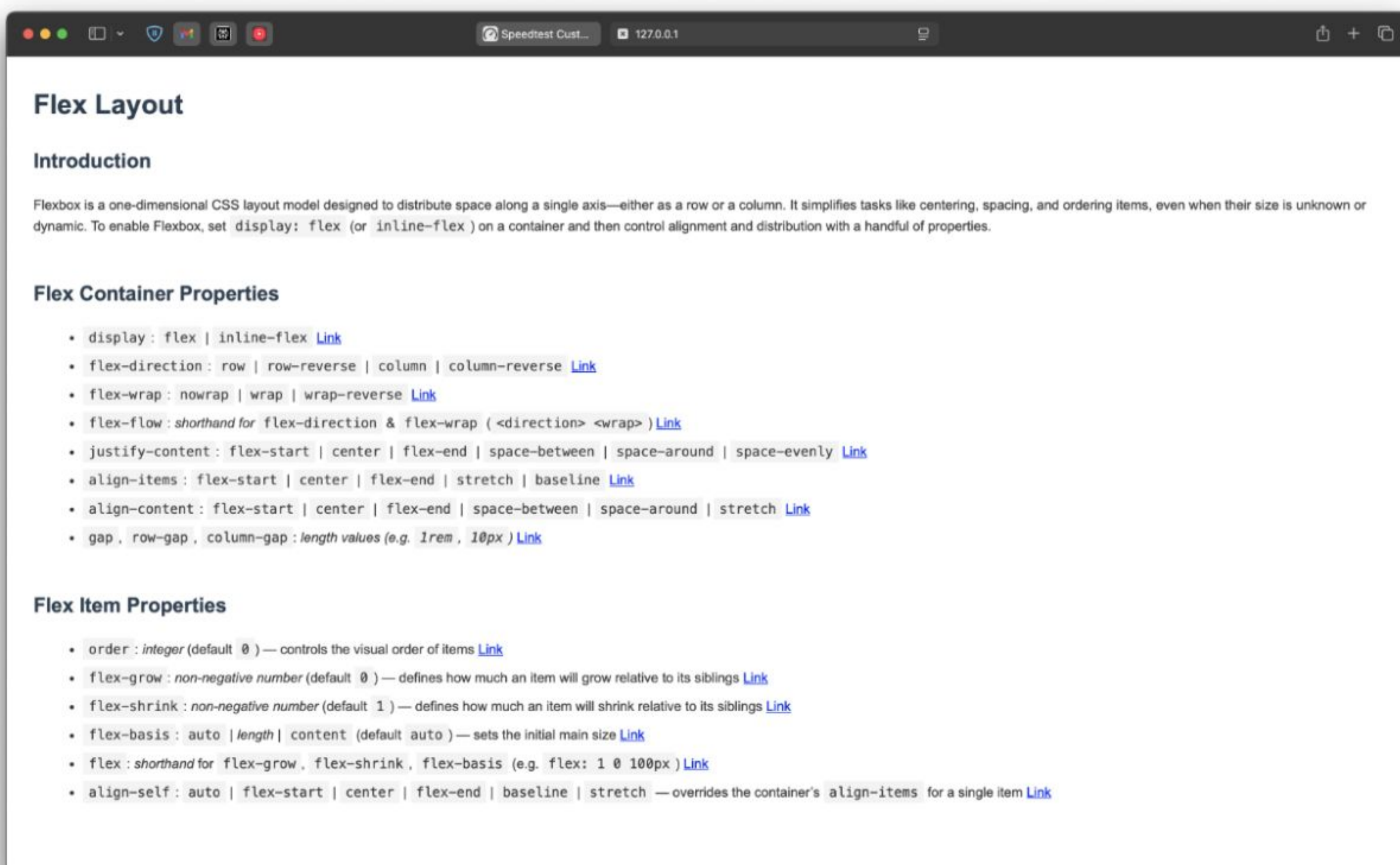
    <!-- Mango Lassi -->
    <div class="recipe" id="mangolassi">
      <h2>Recipe 3: Mango Lassi</h2>
      
      <h3>Ingredients:</h3>
```

```

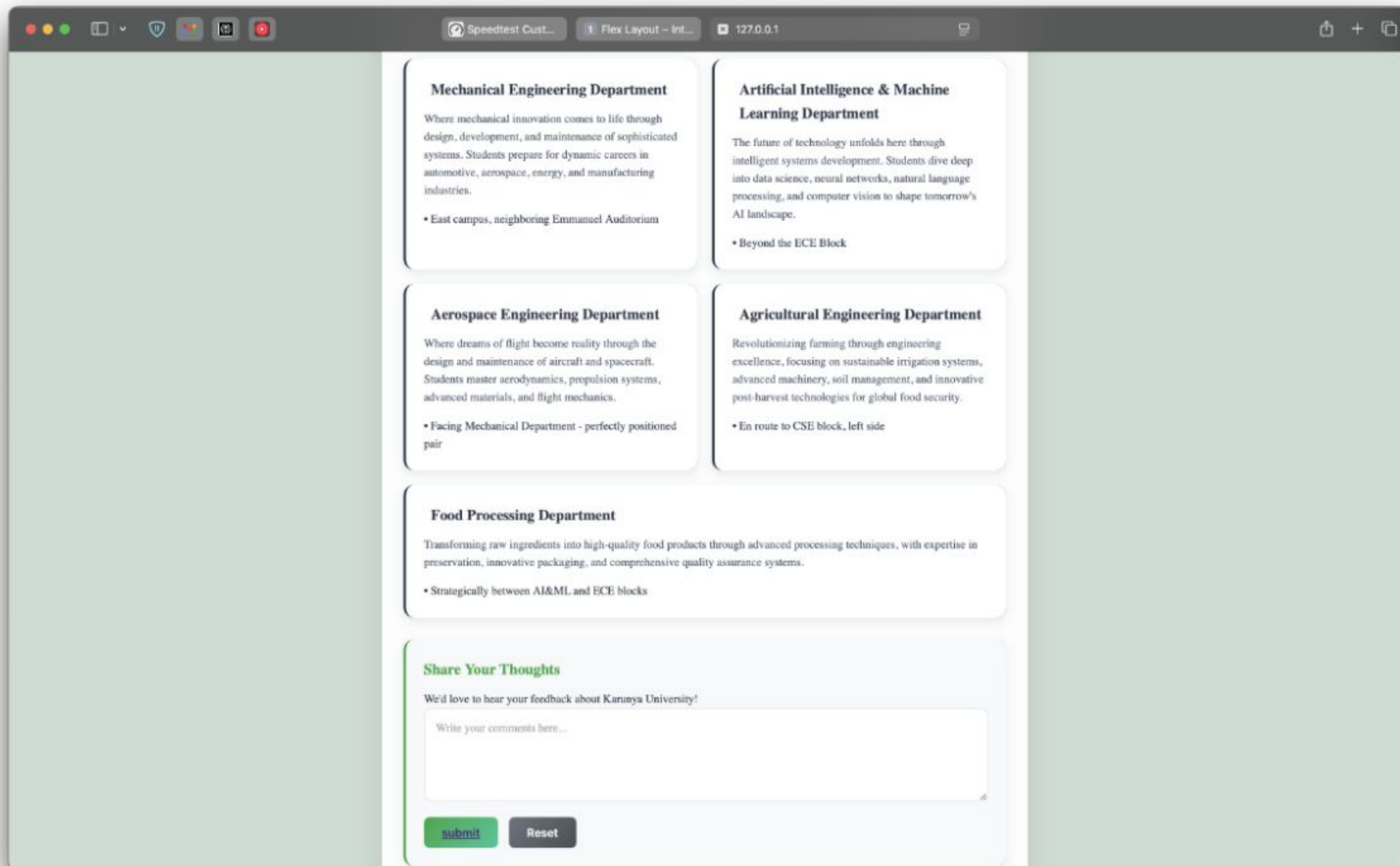
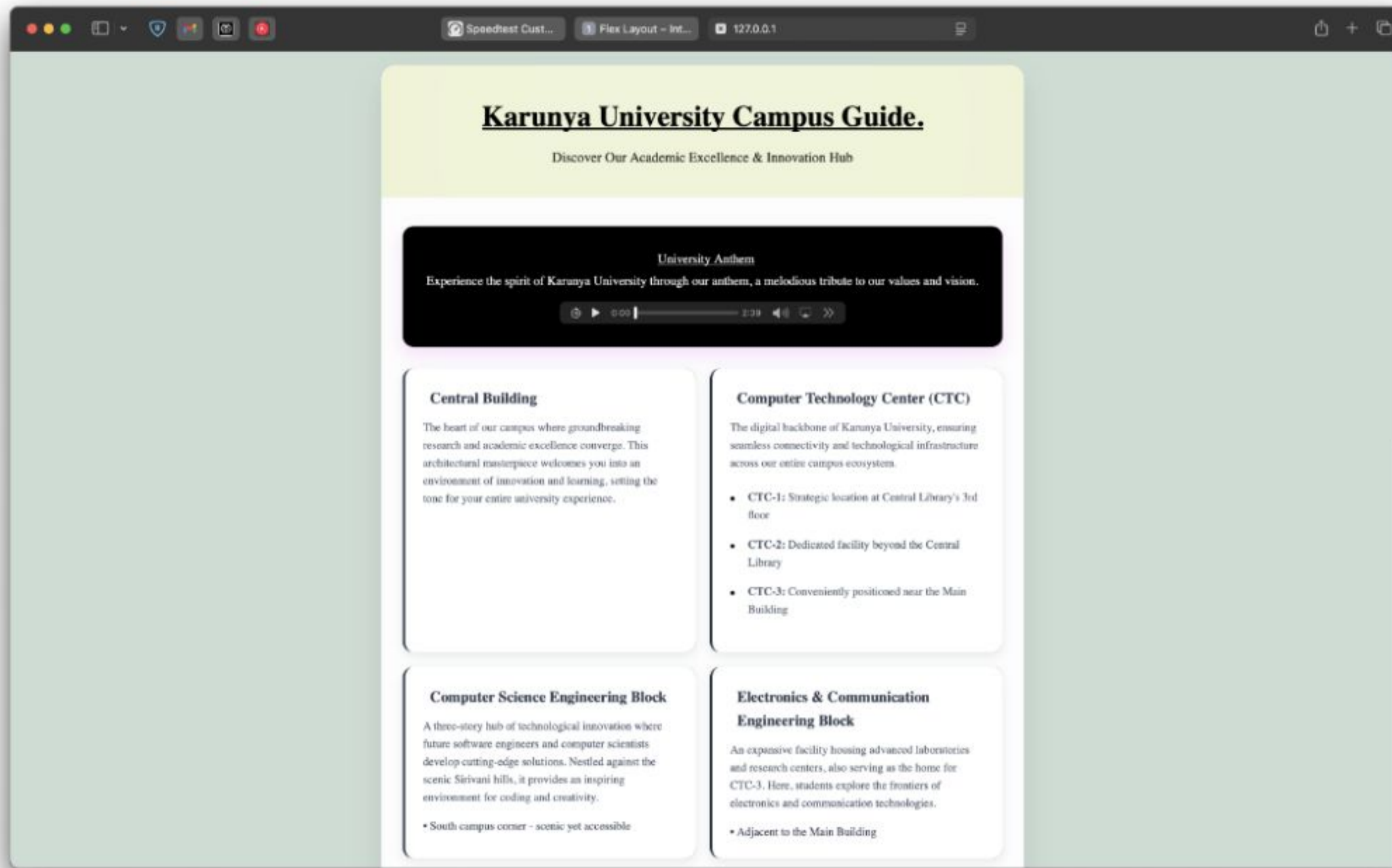
<ul>
  <li>2 ripe mangoes</li>
  <li>1 cup yogurt</li>
  <li>2 tbsp sugar</li>
  <li>Ice cubes</li>
</ul>
<h3>Steps:</h3>
<ol>
  <li>Peel and chop mangoes.</li>
  <li>Blend mangoes with yogurt, sugar, and ice.</li>
  <li>Serve chilled in tall glasses.</li>
</ol>
</div>
</div>
</body>
</html>

```

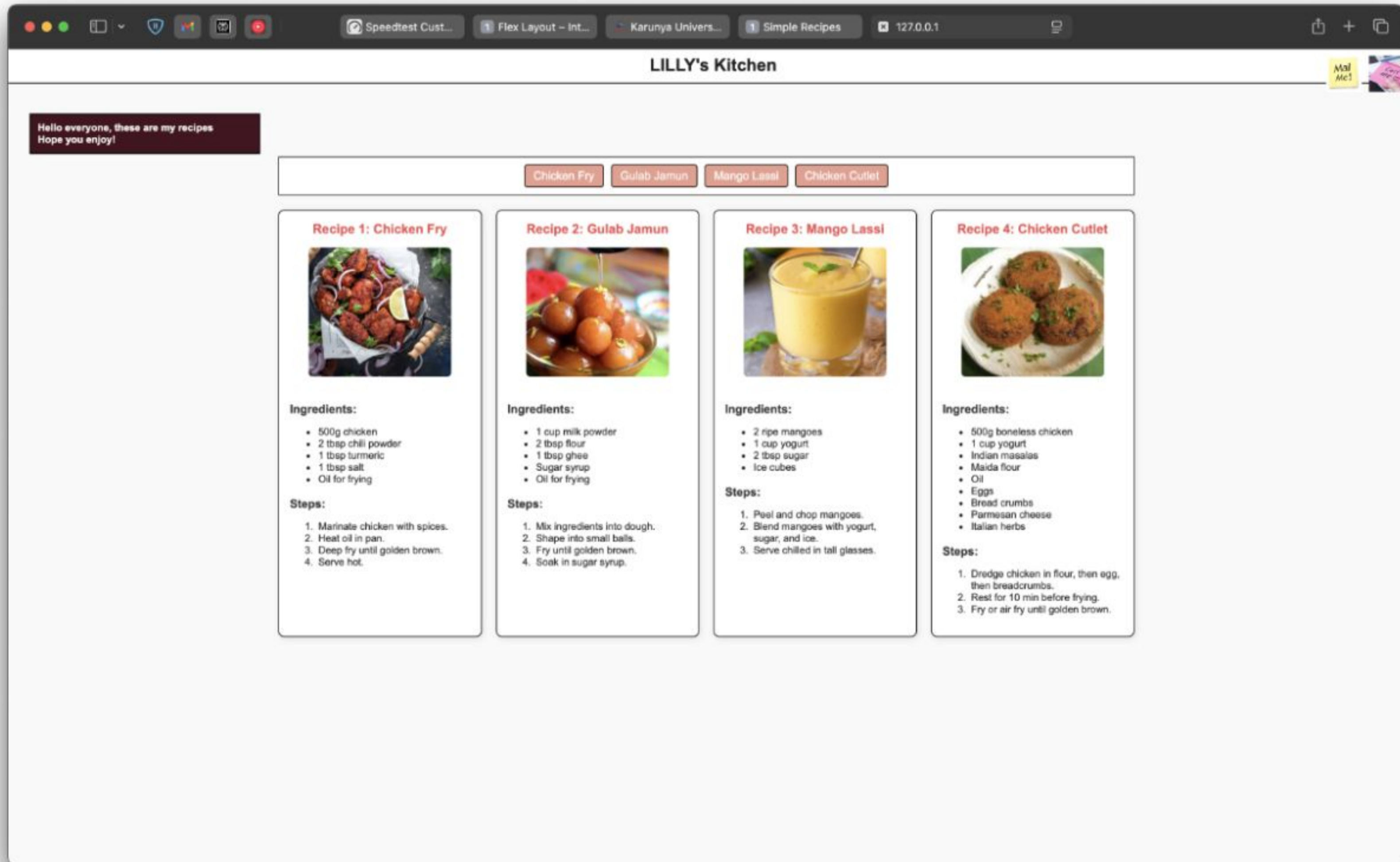
## **7. Results:** **DCSS Flex.html**



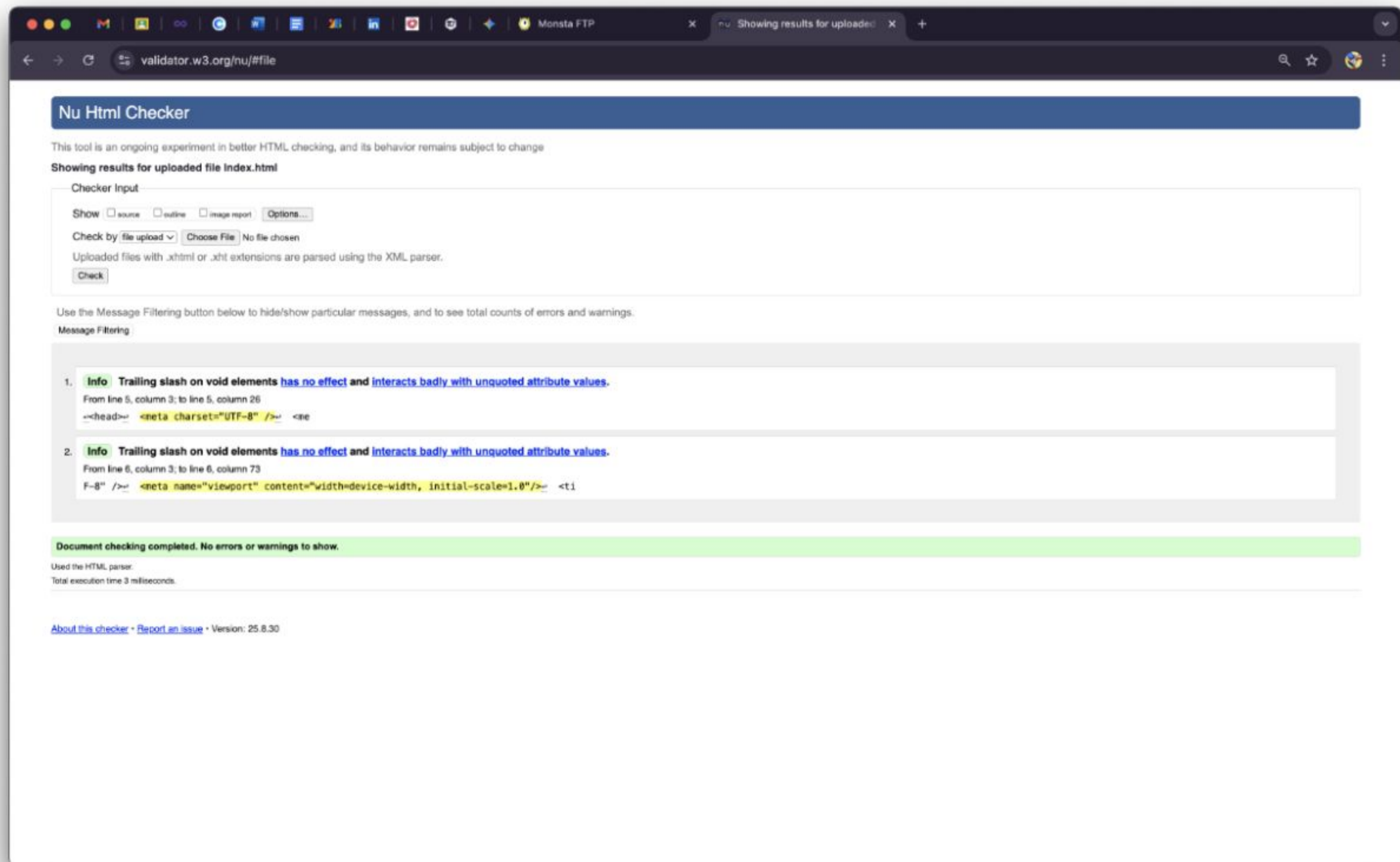
# IIKarunya Map Navigator with CSS:



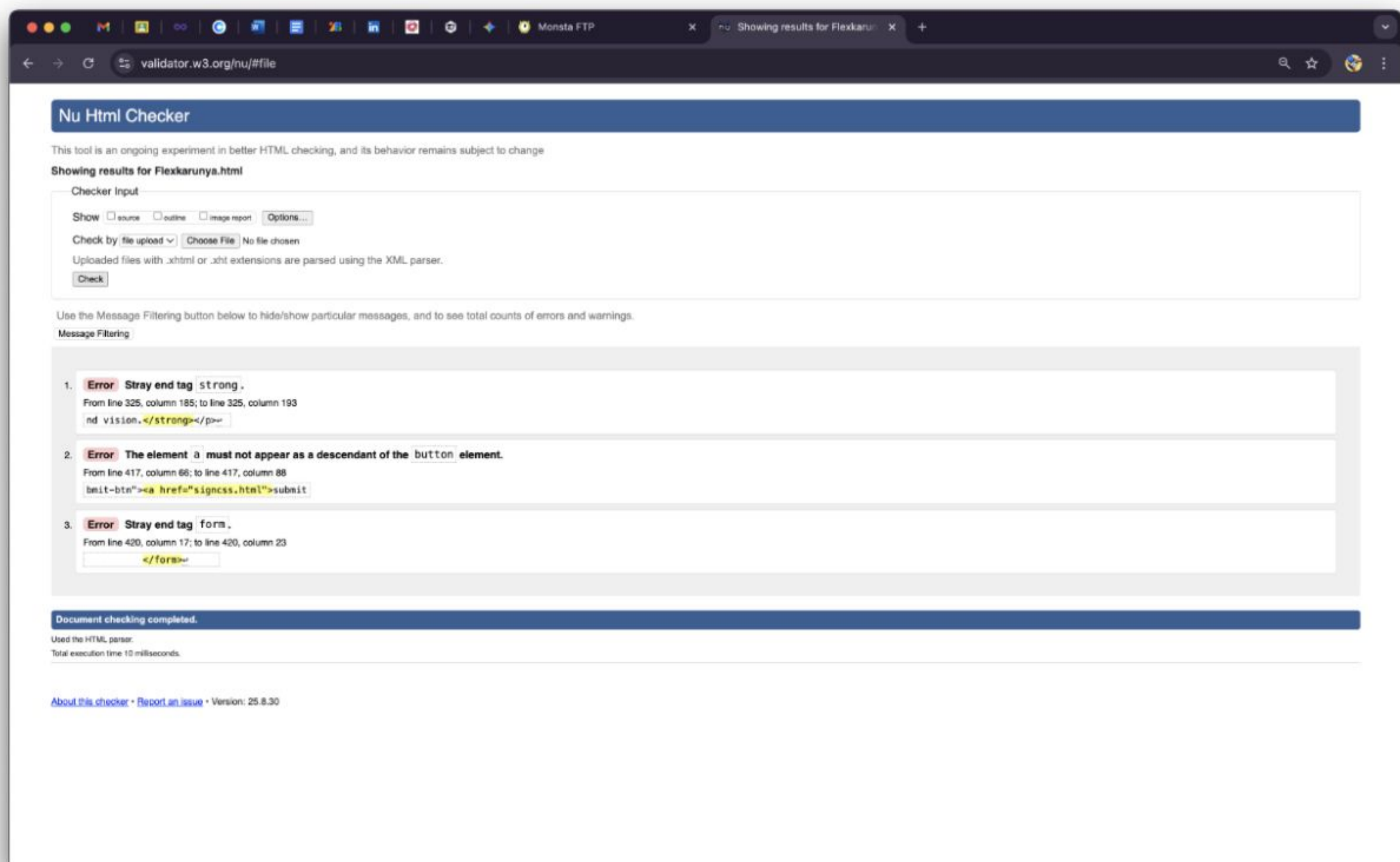
### III) Mom's Kitchen Flexs:



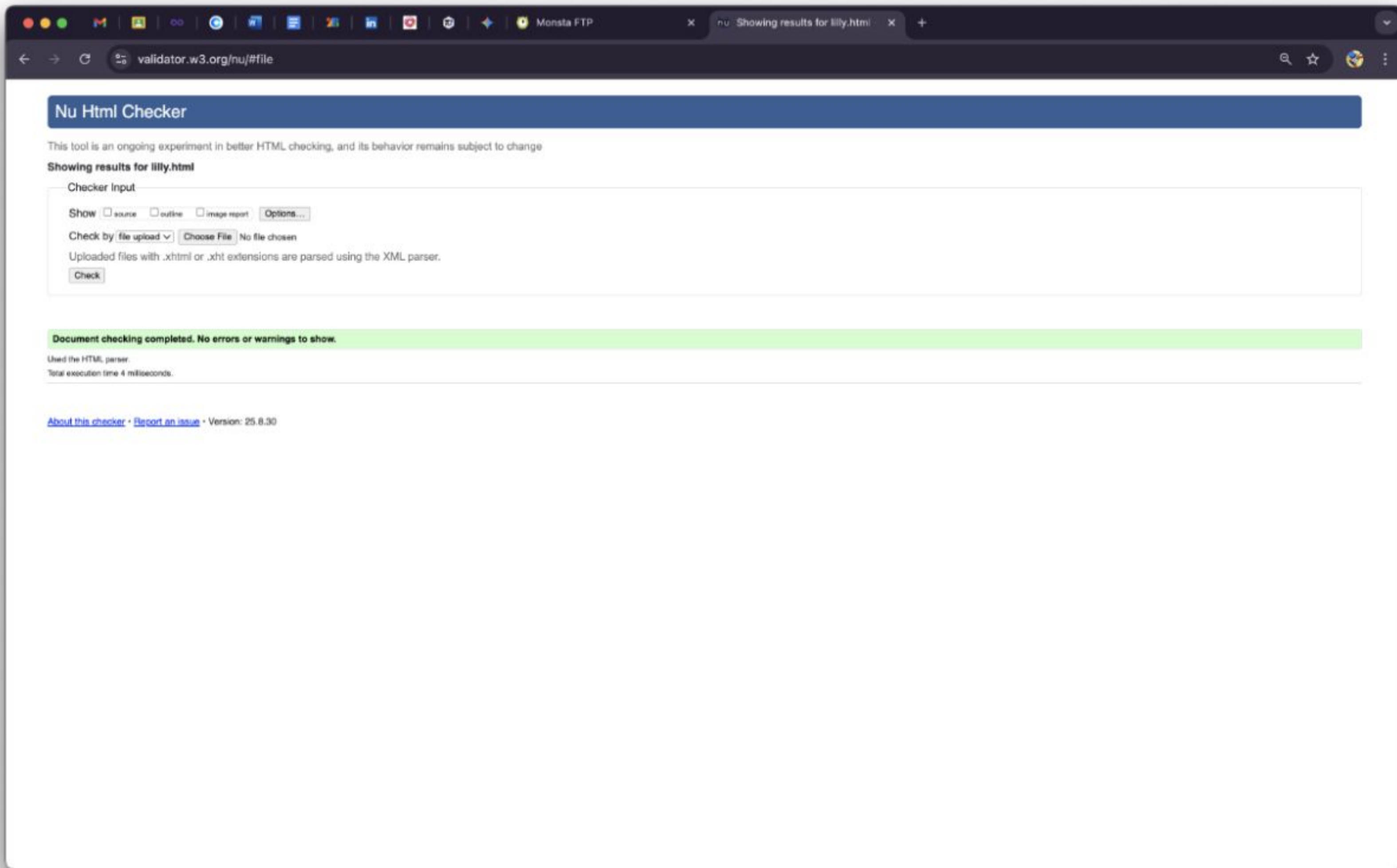
## 7.2 Validation Output: I) CSS Flex.html:



## II)Css Karunya.html:



### III) Mom's Kitchen Flex.html:



## Test Case Report: Practicum 1

### 1.CSS Position:

<b>Test Title/Name</b>	Testing the Demo page of CSS
<b>Module Name</b>	Demo
<b>Test Priority</b>	Importance level: Medium
<b>Test Designed By</b>	Jeedi Joshua Chris Vimal
<b>Test Reviewed By</b>	Dr.I.Titus
<b>Test Designed Date</b>	15-08-2025
<b>Test Executed By</b>	Jeedi Joshua Chris Vimal
<b>Test Execution Date</b>	4-09-2025
<b>Test Summary</b>	Page displays CSS main topics with neat buttons (Selectors, Layouts, Box Model, Text & Font Styling) in a clean, responsive design. Navigation works with hover effects, and your name appears at the bottom-right.
<b>Pre-Requisites</b>	Chrome- Version: 1.103.1  VScode-Version 139.0.7258.128
<b>Dependencies</b>	Not applicable

Test Case ID	Module / Feature	Test Scenario / Description	Input Data	Expected Output	Actual Output	Status (Pass/Fail)	Remarks / Notes
TC001	Loading of Webpage.	Check if the page load in the browser	Open demo.html in browser	Webpage loads with title CSS Demo page	Yes	Pass	
TC002	Links given to each respective pages.	Checking if the links given to the respective pages are working.	Clicking the links in the demo page	Web takes you to the respective linked pages mentioned in the demo	Yes	Pass	
TC003	Detailed Buttons assigned for the webpages	Check the buttons are working perfectly or not and opening the assigned webpages.	Click the button for opening the assigned pages.	All the buttons are working and directing to the pages.	Yes	Pass	
TC004	Borders, margins, and body	Checking the borders and margins.	Open any one of the webpages	Margins Borders are perfectly aligned	Yes	Pass	

TC005	CSS	Check whether CSS is used properly or not?	Check CSS things in the webpages	CSS is used in a well good manner.	Yes	Pass	
TC006	Videos from Youtube	Check the Youtube video list	Check <iframe> section	Links from the YouTube are directly pasted.	Yes	Pass	
TC007		Check the basic forms tags like <label>, <comments>	Verify <label> and <comments> usage	It is shown properly	Yes	Pass	
TC008	CSS	Check the. CSS is used properly or not.	Checking the usage of CSS.	Shows perfectly the way I used CSS	Yes	Pass	
TC009	Validation	Check whether the html code have some error	Validate in W3C	No critical errors	Yes	Pass	

## 2.Lilly's Kitchen

<b>Test Title/Name</b>	Testing Lilly's Kitchen
<b>Module Name</b>	Lilly's Kitchen
<b>Test Priority</b>	Importance level: Medium
<b>Test Designed By</b>	Jeedi Joshua Chris Vimal
<b>Test Reviewed By</b>	Dr.I.Titus
<b>Test Designed Date</b>	24-08-2025
<b>Test Executed By</b>	Jeedi Joshua Chris Vimal
<b>Test Execution Date</b>	04-09-2025
<b>Test Summary</b>	Recipe webpage displaying <b>Chicken Fry, Gulab Jamun, and Mango Lassi</b> with ingredients, procedures, tables, images, and lists. It demonstrates the use of different HTML tags and includes contact links and some video tags.
<b>Pre-Requisites</b>	Chrome- Version: 1.103.1  VScode-Version 139.0.7258.128
<b>Dependencies</b>	Not applicable

Test Case ID	Module / Feature	Test Scenario / Description	Input Data	Expected Output	Actual Output	Status (Pass/Fail)	Remarks / Notes
TC001	Loading of Webpage	Check if the page load in the browser	Open lilly.html in browser	Webpage loads with title Lilly's Kitchen	Yes	Pass	
TC002	Recipe Links	Checking if the recipe links are working or not.	Page scrolls/ jumps to Chicken Fry section	Page scrolls/ jumps to Chicken Fry section	Yes	Pass	
TC003	Unordered List	Check the ordered list and check if the things are organized	Check "List of Products"	All items display with bullet points	Yes	Pass	
TC004	Image Display	Check whether the image is provided	View recipe images (e.g., Gulab Jamun, Mango Lassi)	Images load correctly or alt text shown	Yes	Pass	

TC005	Video Display	Check whether the Video is displayed.	Check the Videos is working	Video displayed in correct size and manner	Yes	Pass	
TC006	Videos from Youtube	Check the Youtube video list	Check <code>&lt;iframe&gt;</code> section	Links from the YouTube are directly pasted.	Yes	Pass	
TC007	Text Emphases	Check the basic word highlights	Verify <code>&lt;mark&gt;</code> and <code>&lt;u&gt;</code> usage	Highlighted text visible	Yes	Pass	
TC008	Contact Links	Check the contact links and mail links are working	Click "Mail me" "Call me" icon	Opens email app And phone log.	Yes	Pass	
TC009	Validation	Check whether the html code have some error	Validate in W3C	No critical errors	Yes	Pass	

## **8. Results:**

- The required webpages
  - i. Demo page for CSS.
  - ii. Mom'swithforms.
  
- The ordered list was chosen to present technical skills in a structured, ranked format.
- Successfully created a functional webpage using appropriate HTML tags.
- Correctly applied tags for headings, paragraphs, images, links, tables, and lists.
- Maintained proper HTML structure with `<!DOCTYPE>`, `<html>`, `<head>`, and `<body>` tags.
- Used CSS to give life to the webpages.
- Validated the webpage using W3C HTML Validator with no major errors.
- Successfully added Video and Audio tags to the code.

## **9. INTERPRETATION OF RESULTS:**

### **•CSS Demo:**

This website contains the guidance of basic CSS elements.

### **•MoM's Kitchen:**

This webpages consists of 3 recipes of my favorite which my Mumma makes for me with Video for Guidance with CSS added to make it look awesome.

## **10. CONCLUSION:**

- Successfully created and tested a functional webpage using HTML.
- Understood the role and usage of different HTML tags.
- Ensured the webpage was W3C standard-compliant after validation.
- Learned the importance of clean and properly structured code.
- Gained hands-on experience in basic web development practice
- Used CSS to give life to the webpages.