

Team 4 Report

Index.html

Head Section

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fantastic Four - Team Website</title>
  <link rel="stylesheet" href="css/styles.css">
  <link
href="https://cdnjs.cloudflare.com/ajax/libs/prism/1.24.1/themes/prism-
tomorrow.min.css" rel="stylesheet" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/pdf.js/3.4.120/pdf.min.js"></scri
pt>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/prism/1.24.1/prism.min.js"></scri
pt>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/prism/1.24.1/components/prism-
javascript.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/prism/1.24.1/components/prism-
css.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/prism/1.24.1/components/prism-
markup.min.js"></script>
</head>
```

Here is the head of our html file where we created the basis for our website. We created a title, linked this file to our CSS file and importing different JavaScript libraries.

```
<nav class="navbar">
  <div class="container">
    <a href="#" class="logo">Fantastic 4</a>
    <input type="checkbox" id="nav-toggle" class="nav-toggle">
    <label for="nav-toggle" class="nav-toggle-label">
      <span></span>
      <span></span>
      <span></span>
    </label>
    <ul class="nav-links">
```

```

        <li><a href="#about">About</a></li>
        <li><a href="#presentation">Presentation</a></li>
        <li><a href="#documentation">Documentation</a></li>
    </ul>
</div>
</nav>

```

Here we used the <nav> function in HTML to add a navigation bar to our website. The navigation bar allows for quick access to the sections: About, Presentation, and Documentation on our website. Finally, we used a checkbox-based toggle for mobile navigation.

```

<section id="about" class="team-section">
    <div class="container">
        <h1 class="welcome-title">Welcome to our Website <span
class="wave">👋</span>!</h1>
        <div class="team-content">
            <h2 class="section-title">Meet the Team</h2>
            <div class="team-grid">
                <div class="team-member">
                    <div class="member-container">
                        <div class="image-container">
                            
                        </div>
                        <h3>Pranaav Venkatasubramanian</h3>
                    </div>
                </div>
                <div class="team-member">
                    <div class="member-container">
                        <div class="image-container">
                            
                        </div>
                        <h3>Jerry Steve Jeanty</h3>
                    </div>
                </div>
                <div class="team-member">
                    <div class="member-container">
                        <div class="image-container">
                            
                        </div>
                        <h3>Shafil Jayed</h3>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="team-member">
            <div class="member-container">
                <div class="image-container">
                    
                </div>
                <h3>Bryan Belizaire</h3>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</section>

```

This section shows how we added our pictures, names, and a welcome message to the website.

Presentation Section

```

<section id="presentation" class="section presentation-section">
    <div class="container">
        <h2 class="section-title">Project Presentation</h2>
        <div class="presentation-container">
            <div id="canvas-container"></div>
            <div class="presentation-controls">
                <button id="prevSlide">Previous</button>
                <span id="slideNumber">1/1</span>
                <button id="nextSlide">Next</button>
            </div>
        </div>
    </div>
</section>

```

This section establishes the section for our presentation slides and creates the buttons needed for navigating between the slides.

```

<section id="documentation" class="section">
    <div class="container">
        <h2 class="section-title">Documentation</h2>
        <div class="flex-container">
            <div class="code-section">
                <h3>Source Code</h3>
                <div class="code-viewer">
                    <div class="code-tabs" role="tablist">
                        <button class="code-tab active" role="tab"
aria-selected="true" data-tab="html">index.html</button>

```

```

        <button class="code-tab" role="tab" aria-
selected="false" data-tab="css">styles.css</button>
        <button class="code-tab" role="tab" aria-
selected="false" data-tab="js">main.js</button>
    </div>
    <div class="code-content" id="html-content"
role="tabpanel"></div>
    <div class="code-content" id="css-content"
role="tabpanel"></div>
    <div class="code-content" id="js-content"
role="tabpanel"></div>
</div>
</div>
<div class="report-section">
    <h3>Project Report</h3>
    <div class="pdf-wrapper">
        <div class="pdf-container">
            <iframe src="assets/report.pdf"
title="Project Report"></iframe>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</section>

```

This part displays the documentation, source code and embeds a PDF report.

```

<footer class="footer">
    <div class="container">
        <div class="footer-content">
            <a href="#" class="footer-logo-container">
                <div class="back-to-top-content">
                    
                    <span class="back-to-top-text">Back To Top</span>
                </div>
            </a>
            <h2 class="footer-title">Fantastic 4</h2>
            <p class="footer-copyright">&copy; 2025 Fantastic 4. All
rights reserved.</p>
        </div>
    </div>
</footer>
<script src="js/main.js"></script>
</body>

```

```
</html>
```

This last part of the html file creates a footer for the site which includes a back-to-top link, the team name ("Fantastic 4") and copyright information.

Main.js

```
document.addEventListener('DOMContentLoaded', () => {  
  initializeTeamImages();  
  initializeMobileMenu();  
  initializeNavigation();  
  initializePresentation();  
  initializeCodeViewer();  
  initializeTeamAnimations();  
  initializeFooter();  
});
```

This beginning portion ensures that all functions are executed only after the DOM (Document Object Model) has finished loaded through an event listener. It also creates various functionalities like team images, navigation, mobile menu, presentation, code viewer, animations, and footer.

```
const scrollToTop = () => {  
  window.scrollTo({ top: 0, behavior: 'smooth' });  
};
```

This small section allows the user to simply go back to the top of the page without needing to scroll.

```
function initializeFooter() {  
  const elements = [  
    '.footer-logo-container',  
    '.footer-title',  
    '.back-to-top-content'  
  ].map(selector => document.querySelector(selector));  
  
  elements.forEach(element => {  
    if (element) element.addEventListener('click', scrollToTop);  
  });  
}
```

```

    });
}

function initializeTeamImages() {
  document.querySelectorAll('.image-container img').forEach(img => {
    img.onerror = () => {
      const initials = (img.alt || 'Team Member')
        .split(' ')
        .map(n => n[0])
        .join('');

      const fallbackDiv = document.createElement('div');
      fallbackDiv.className = 'image-fallback';
      fallbackDiv.textContent = initials;
      img.parentNode.replaceChild(fallbackDiv, img);
    };
  });
}

function initializeMobileMenu() {
  const mobileMenuBtn = document.querySelector('.mobile-menu-btn');
  const navLinks = document.querySelector('.nav-links');

  if (!mobileMenuBtn) return;

  mobileMenuBtn.addEventListener('click', () => {
    navLinks.classList.toggle('active');
    mobileMenuBtn.classList.toggle('active');
  });

  document.addEventListener('click', (e) => {
    if (!mobileMenuBtn.contains(e.target) && !navLinks.contains(e.target)) {
      navLinks.classList.remove('active');
      mobileMenuBtn.classList.remove('active');
    }
  });
}

function initializeNavigation() {
  const logo = document.querySelector('.logo');
  if (logo) logo.addEventListener('click', scrollToTop);

  document.querySelectorAll('.nav-links a').forEach(link => {
    link.addEventListener('click', (e) => {
      e.preventDefault();
      const targetSection = document.querySelector(link.getAttribute('href'));

      if (targetSection) {
        const navHeight = document.querySelector('.navbar').offsetHeight;
        const targetPosition = targetSection.offsetTop - navHeight;
        window.scrollTo({ top: targetPosition, behavior: 'smooth' });
      }
    });
  });
}

function initializeTeamAnimations() {
  const observer = new IntersectionObserver(
    entries => {
      entries.forEach(entry => {
        if (entry.isIntersecting) {

```

```

        entry.target.classList.add('visible');
        observer.unobserve(entry.target);
    }
    });
},
{ threshold: 0.2 }
);

document.querySelectorAll('.team-member').forEach(member => {
    member.style.opacity = '1';
    member.style.transform = 'none';
});
}

function initializePresentation() {
    const container = document.querySelector('.presentation-container');
    const prevButton = document.getElementById('prevSlide');
    const nextButton = document.getElementById('nextSlide');
    const slideNumber = document.getElementById('slideNumber');

    pdfjsLib.GlobalWorkerOptions.workerSrc =
'https://cdnjs.cloudflare.com/ajax/libs/pdf.js/3.4.120/pdf.worker.min.js';

    let pdfDoc = null;
    let pageNum = 1;
    let pageRendering = false;
    let pageNumPending = null;

    function calculateScale() {
        return (container.clientWidth - 20) / 1000;
    }

    function renderPage(num) {
        pageRendering = true;
        const loadingDiv = document.createElement('div');
        loadingDiv.className = 'loading-indicator';
        loadingDiv.textContent = 'Loading page...';
        container.appendChild(loadingDiv);

        pdfDoc.getPage(num).then(page => {
            const viewport = page.getViewport({ scale: calculateScale() });
            const canvas = document.getElementById('presentation-canvas');
            const context = canvas.getContext('2d');

            canvas.height = viewport.height;
            canvas.width = viewport.width;

            page.render({
                canvasContext: context,
                viewport: viewport
            }).promise.then(() => {
                pageRendering = false;
                if (pageNumPending !== null) {
                    renderPage(pageNumPending);
                    pageNumPending = null;
                }
                loadingDiv.remove();
                updatePageCount();
            });
        });
    }
}

```

```

function queueRenderPage(num) {
  if (pageRendering) {
    pageNumPending = num;
  } else {
    renderPage(num);
  }
}

function updatePageCount() {
  slideNumber.textContent = `${pageNum} / ${pdfDoc.numPages}`;
  prevButton.disabled = pageNum <= 1;
  nextButton.disabled = pageNum >= pdfDoc.numPages;
}

const canvas = document.createElement('canvas');
canvas.id = 'presentation-canvas';
canvas.className = 'presentation-canvas';
container.appendChild(canvas);

pdfjsLib.getDocument('assets/presentation/presentation.pdf').promise
  .then(pdfDoc => {
    pdfDoc = pdfDoc_;
    renderPage(pageNum);

    prevButton.addEventListener('click', () => {
      if (pageNum > 1) {
        pageNum--;
        queueRenderPage(pageNum);
      }
    });

    nextButton.addEventListener('click', () => {
      if (pageNum < pdfDoc.numPages) {
        pageNum++;
        queueRenderPage(pageNum);
      }
    });

    document.addEventListener('keydown', (e) => {
      if (e.key === 'ArrowLeft' && pageNum > 1) {
        pageNum--;
        queueRenderPage(pageNum);
      }
      if (e.key === 'ArrowRight' && pageNum < pdfDoc.numPages) {
        pageNum++;
        queueRenderPage(pageNum);
      }
    });
  })
  .catch(error => {
    container.innerHTML = `
      <div class="error-message">
        Error loading presentation. Please ensure the PDF file exists and try
again.
      </div>
    `;
  });

window.addEventListener('resize', debounce(() => {
  if (pdfDoc) renderPage(pageNum);

```



```

    }, 250));
}

function initializeCodeViewer() {
    const tabs = document.querySelectorAll('.code-tab');
    document.querySelectorAll('.code-content').forEach(content => {
        content.style.display = 'none';
    });

    function switchTab(tab, fileType) {
        document.querySelectorAll('.code-tab').forEach(t => t.classList.remove('active'));
        document.querySelectorAll('.code-content').forEach(c => {
            c.classList.remove('active');
            c.style.display = 'none';
        });

        tab.classList.add('active');
        const content = document.getElementById(`${fileType}-content`);
        if (content) {
            content.classList.add('active');
            content.style.display = 'block';
            loadCodeFile(fileType, content);
        }
    }

    tabs.forEach(tab => {
        tab.addEventListener('click', () => {
            switchTab(tab, tab.getAttribute('data-tab'));
        });
    });

    if (tabs.length > 0) {
        switchTab(tabs[0], tabs[0].getAttribute('data-tab'));
    }
}

async function loadCodeFile(fileType, contentElement) {
    const fileMap = {
        html: 'indexHTML.txt',
        css: 'stylesCSS.txt',
        js: 'mainJS.txt'
    };

    try {
        contentElement.innerHTML = '<div class="loading">Loading...</div>';

        const response = await fetch(`assets/code/${fileMap[fileType]}`);
        if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);

        const content = await response.text();
        const escapedContent = content
            .replace(/&/g, '&amp;')
            .replace(/</g, '&lt;')
            .replace(/>/g, '&gt;')
            .replace(/"/g, '&quot;')
            .replace(/'/g, '&#039;');

        contentElement.innerHTML = `<pre><code class="language-${fileType}">${escapedContent}</code></pre>`;

        if (window.Prism) {

```

```

        Prism.highlightElement(contentElement.querySelector('code'));
    }
} catch (error) {
    contentElement.innerHTML = `
        <div class="error">
            Error loading file: ${error.message}
        </div>
    `;
}
}
}

```

Each of these sections create and detail the various sections.

Footer

- Adds a click element to the footer, adding functionality to the “Back to Top” section

Team images

- The code sets up a backup plan incase the pictures fail to load as well as animating team members as they appear.

Mobile menu

- Enables toggling the mobile menu when clicking the menu button.
- Closes the menu if the user clicks outside it.

Navigation bar

- Ensures smooth scrolling when clicking navigation links.
- Adjusts scroll position to account for the fixed navbar height.

Presentation viewer

- Loads a PDF presentation
- Allows users to navigate through slides with buttons or arrow keys.

Code viewer

- Allows users to switch tabs in the code viewer section.
- Dynamically loads and highlights code files.