To tackle this problem statement of developing a personal website, here’s a step-by-step plan that follows the instructions and guidelines provided:

# Step 1: Website Structure Design

1. Home Page

- A clean introduction about yourself.

- A brief bio (profession, passion, skills, etc.).

- Include a navigation bar that links to the Achievements and Projects pages.

- Consider adding a profile picture and some simple animations or hover effects to make the page visually appealing.

2. Achievements Page

- Create a simple layout (either grid or list format) to display your achievements.

- Each achievement should have:

- Title

- Description

- Date

- You can use card-style containers to make it aesthetically pleasing.

3. Working On & Projects Page

- Display current and past projects.

- Each project should contain:

- Project name

- Brief description

- Tools and technologies used

- Link to project demo or GitHub repository

- Ensure projects are categorized for easy navigation (e.g., current projects and past projects).

#Step 2: Design Requirements

1. Responsiveness

- Use media queries in CSS to ensure that the website looks good on all screen sizes (desktops, tablets, smartphones).

- Test the site using tools like Google Chrome Developer Tools to check how it responds on different devices.

2. Clean and Modern Design

- Select a modern typography (fonts like 'Roboto', 'Lato', or 'Open Sans').

- Choose a color scheme that’s easy on the eyes—e.g., light background with dark text, or vice versa. Use tools like [Coolors](https://coolors.co/) to help you select complementary colors.

- Pay attention to whitespace and avoid clutter.

- Use CSS Flexbox or Grid to layout elements neatly.

3. Accessibility

- Ensure all images have descriptive alt text.

- Ensure text contrast is high for readability (test with tools like [WebAIM Contrast Checker](https://webaim.org/resources/contrastchecker/)).

# Step 3: Functionality and Interactivity

1. Navigation

- A navigation bar that sticks to the top or side for easy access to each page.

- Use hover effects for buttons and links to enhance user experience.

2. Interactivity (Optional)

- You can use simple animations with CSS for hover effects or transitions when switching between pages or hovering over cards.

- JavaScript can be used for more complex interactions, like dynamically loading content or enabling dark/light mode.

# Step 4: Technical Implementation

1. \*\*Frontend Tools\*\*

- \*\*HTML\*\* for structure.

- \*\*CSS\*\* for styling.

- \*\*JavaScript\*\* (optional) for interactive elements.

- \*\*Bootstrap\*\* or other CSS frameworks can help speed up the responsive design process.

2. Deployment on GitHub Pages

- Once the website is complete, push the code to a GitHub repository.

- Enable GitHub Pages in the repository settings to host the site.

- Ensure all assets (images, CSS, JavaScript files) are correctly linked and functioning post-deployment.

# Step 5: Source Code and Documentation

1. \*\*Source Code\*\*

- Ensure the code is clean and follows best practices:

- Proper indentation.

- Comments explaining each section of the code.

- Use meaningful class and ID names in HTML and CSS.

2. \*\*README.md\*\*

- Create a README.md file in the repository explaining:

- Design choices.

- Instructions on how to modify or run the website locally.

- Include links to live site and project repository.

#Step 6: Testing and Submission

1. Testing

- Ensure all page’s load correctly and that navigation works smoothly.

- Test the responsiveness on different devices.

- Check for any broken links or missing assets.

2. Submission

- Submit the GitHub repository link along with the live site URL hosted on GitHub Pages.

- Ensure that all files are well-organized in the repository and that documentation is clear.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

To help you get started with your personal website development, I’ll provide the necessary HTML, CSS, and JavaScript code that meets the requirements of your objective. The project consists of three main pages: Home, Achievements, and Projects. Below is the code for each page, the navigation menu, and responsive design using CSS.

# 1. Folder Structure:

```

/website-project

│

├── /images (place images used in the project here)

│

├── index.html (Home Page)

├── achievements.html (Achievements Page)

├── projects.html (Projects Page)

│

├── style.css (Styles for all pages)

├── script.js (Optional JavaScript for interactive elements)

├── README.md (Documentation)

```

# 2. Home Page (index.html):

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Personal Website - Home</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<!-- Navigation Menu -->

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="achievements.html">Achievements</a></li>

<li><a href="projects.html">Projects</a></li>

</ul>

</nav>

<!-- Home Page Content -->

<section id="home">

<h1>Welcome to My Personal Website</h1>

<p>Hello! I'm [Your Name], a [Your Profession]. This website showcases my achievements and ongoing projects. Explore and get to know my work!</p>

<img src="images/profile.jpg" alt="Profile Picture" class="profile-pic">

</section>

</body>

</html>

```

# 3. Achievements Page (achievements.html):

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Personal Website - Achievements</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<!-- Navigation Menu -->

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="achievements.html">Achievements</a></li>

<li><a href="projects.html">Projects</a></li>

</ul>

</nav>

<!-- Achievements Page Content -->

<section id="achievements">

<h1>Achievements</h1>

<div class="achievement-item">

<h2>Won XYZ Hackathon</h2>

<p>Date: June 2023</p>

<p>Built an AI-based room allocation system that efficiently manages resources.</p>

</div>

<div class="achievement-item">

<h2>Published a Research Paper on ABC</h2>

<p>Date: April 2022</p>

<p>Co-authored a paper on the future of blockchain technology in supply chain management.</p>

</div>

</section>

</body>

</html>

```

# 4. Projects Page (projects.html):

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Personal Website - Projects</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<!-- Navigation Menu -->

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="achievements.html">Achievements</a></li>

<li><a href="projects.html">Projects</a></li>

</ul>

</nav>

<!-- Projects Page Content -->

<section id="projects">

<h1>Projects</h1>

<div class="project-item">

<h2>Room Allocation Web App</h2>

<p>A tool to efficiently allocate rooms in hostels based on group ID, gender, and capacity.</p>

<p>Technologies Used: JavaScript, Python, HTML, CSS.</p>

<a href="https://github.com/yourrepo/room-allocation" target="\_blank">View Project</a>

</div>

<div class="project-item">

<h2>E-commerce Website</h2>

<p>An online platform for selling products with an integrated payment gateway.</p>

<p>Technologies Used: Node.js, Express, MongoDB.</p>

<a href="https://github.com/yourrepo/ecommerce" target="\_blank">View Project</a>

</div>

</section>

</body>

</html>

```

# 5. CSS Stylesheet (style.css):

```css

/\* General Styles \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f0f0f0;

}

nav ul {

display: flex;

justify-content: center;

background-color: #333;

list-style: none;

padding: 10px 0;

margin: 0;

}

nav ul li {

margin: 0 15px;

}

nav ul li a {

text-decoration: none;

color: white;

font-weight: bold;

}

nav ul li a:hover {

color: #4CAF50;

}

/\* Home Page \*/

#home {

text-align: center;

padding: 50px;

background-color: #fff;

}

.profile-pic {

border-radius: 50%;

width: 150px;

height: 150px;

}

/\* Achievements & Projects Page \*/

#achievements, #projects {

padding: 20px;

max-width: 1000px;

margin: 0 auto;

background-color: #fff;

}

.achievement-item, .project-item {

background-color: #e0e0e0;

padding: 15px;

margin: 10px 0;

border-radius: 5px;

}

.project-item a {

color: #007BFF;

text-decoration: none;

}

.project-item a:hover {

text-decoration: underline;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

nav ul {

flex-direction: column;

}

.profile-pic {

width: 100px;

height: 100px;

}

}

```

#6. Optional JavaScript (script.js) (For Hover Effects/Animations):

```javascript

// Example JavaScript for animations or interactions

document.querySelectorAll('nav ul li a').forEach(link => {

link.addEventListener('mouseover', function () {

this.style.color = '#ff5722';

});

link.addEventListener('mouseout', function () {

this.style.color = '#fff';

});

});

```

#7. README.md (Documentation):

```markdown

# Personal Website

## Project Overview

This is a simple personal website that showcases my achievements and projects. The website is designed with responsiveness in mind and provides a clean, user-friendly interface. It is structured into three main pages: Home, Achievements, and Projects.

## Features

- Responsive design to ensure compatibility across devices (desktop, tablet, and mobile).

- Clean navigation to easily switch between pages.

- Hover effects and interactive elements for better user experience.

## Technologies Used

- HTML5

- CSS3

- JavaScript (Optional)

- GitHub Pages for deployment

## How to Run the Project Locally

1. Clone this repository:

```

git clone https://github.com/yourusername/your-repo.git

```

2. Navigate to the project directory:

```

cd your-repo

```

3. Open `index.html` in your browser to view the website locally.

#8. Deployment on GitHub Pages

1. Push the code to a public repository on GitHub.

2. Go to the repository settings.

3. Scroll down to "GitHub Pages" and select the branch (usually `main` or `master`) from which you want to deploy the site.

4. Save the settings and your website will be live at `https://<username>.github.io/<repository-name>`.

#Conclusion:

This structure and code will give you a solid foundation for your personal website. After deploying it on GitHub Pages, you can submit the public URL along with the repository link as required.