Goals and Plan for My Website

I have two options for making my website:
Caesar Cipher Encoder/Decoder Website or a University Website.

Goal and Plan

➢ Caesar Cipher Encoder/Decoder Website

Goal: To create a user-friendly web application for encoding, decoding, and brute force attacking Caesar ciphers, providing an educational tool for cryptography enthusiasts. I'll also include a history of Ceasar Cipher and an options table on the left side of the page on what the users want to know.

1. **Project Setup**
    - Visual Studio Code is what I'm going to use to code this project.
    - Set up a new project folder and initialize a new Git repository for version control.

2. **Front-End Development**
    - Create HTML templates for the different sections of the website.
    - Use CSS to style the website, making it visually appealing and intuitive.
    - Implement interactive features using JavaScript to handle encoding, decoding, and brute force functionality.

3. **Back-End Development**
    - Create the necessary logic for encoding, decoding, and brute force algorithms in a server-side language (e.g., JavaScript).

4. **Testing and Debugging**
    - Perform thorough testing to ensure the application works as expected.
    - Debug any issues and optimize the website for performance and responsiveness.

5. **Deployment**
    - Choose a web hosting service for deploying the website.
    - Configure the domain name and DNS settings if a custom domain is desired.

Goal and Plan

➢ University Website

Goal: To develop an informative and user-friendly website for a university, providing essential information about academic programs, faculty, admissions, and campus life.

1. **Project Setup**
    - Select the appropriate tools and frameworks for web development.
    - Create a project structure and set up version control using Git.

2. **Content and Structure**
    - Plan the website's structure, including pages for academic programs, faculty profiles, admissions information, campus facilities, and student life.
    - Gather or create content such as text, images, and videos for the website.

3. **Front-End Development**
    - Design the website layout and navigation using HTML, CSS, and a chosen front-end framework.
    - Create responsive and accessible components for displaying information about the university.

4. **Testing and Feedback**
    - Test the website across different devices and browsers to ensure a consistent and user-friendly experience.