Deployment Plan

Server requirements:

(Note: servers on hosting services meet most or all these requirements)

- ➔ At least 20 gigabytes of free storage
- ➔ At least 4 gigabytes of RAM
- ➔ Linux-based Operating System built for server environments (Ubuntu server recommended)
- ➔ Web hosting control panel (cPanel) enabled
- ➔ File transfer protocol (ftp) or secure file transfer protocol (sftp) enabled
- ➔ HTTP Web server software (Apache/Nginx) installed
- ➔ Version control software (git) installed

Site requirements:

- ➔ Domain name (website address)
- ➔ SSL certificate for encrypted connections

Browser compatibility:

(Note: all modern browsers on desktops, smartphones, and tablets should be able to load the site)

- ➔ Internet Explorer 10+
- ➔ Google Chrome
- ➔ Chromium-based browsers (Brave, Opera, New Microsoft Edge 2020+)
- ➔ Safari
- ➔ JavaScript must be enabled
- ➔ Desktop, Tablet, and mobile views are supported

Pre-installation:

(Note: only relevant to installations not using a hosting solution)

- ➔ This pre-installation will assume the usage of Ubuntu Server since it is the easiest to install. First option is to install Ubuntu Server as the primary Operating System of the server computer: https://ubuntu.com/tutorials/install-ubuntu-server#6-choose-your-install

Alternatively, install Ubuntu Server as a virtual machine using virtualization software such as VirtualBox: https://www.ceos3c.com/open-source/install-ubuntu-server-18-04-lts/

- ➔ Install Apache web server: https://ubuntu.com/tutorials/install-and-configure-apache#1-overview
- ➔ Install an ftp server and setup an account:

  https://phoenixnap.com/kb/install-ftp-server-on-ubuntu-vsftpd

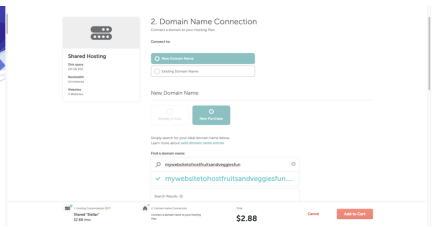
- ➔ Install the git command-line application by typing 'sudo apt install git' in the terminal

Website Installation:

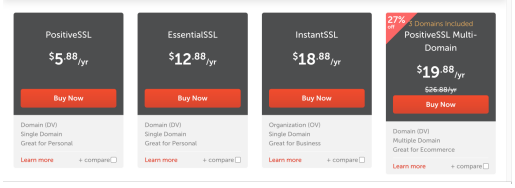
(Note: this installation uses namecheap for demonstration)

- ➔ Purchase a domain name from a domain name registrar such as namecheap. The process involves searching for a name which hasn't been taken and paying a subscription fee to claim ownership over the name.

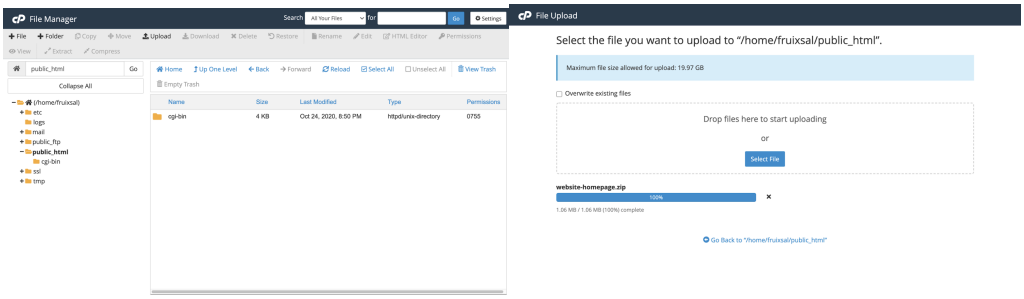
- ➔ Purchase a hosting solution through one of the domain name registrars such as namecheap. It is recommended to purchase an option with backup capabilities and fast response times. Ensure the hosting solution points to the domain which will be used to host the site. Also check to see if cPanel is included (namecheap will include it with the installation).

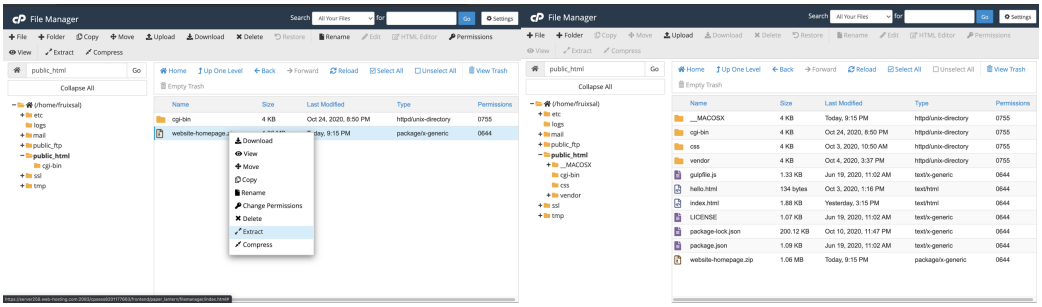


- ➔ Purchase an SSL certificate. With PositiveSSL, the certificate can be automatically installed through the hosting solution.

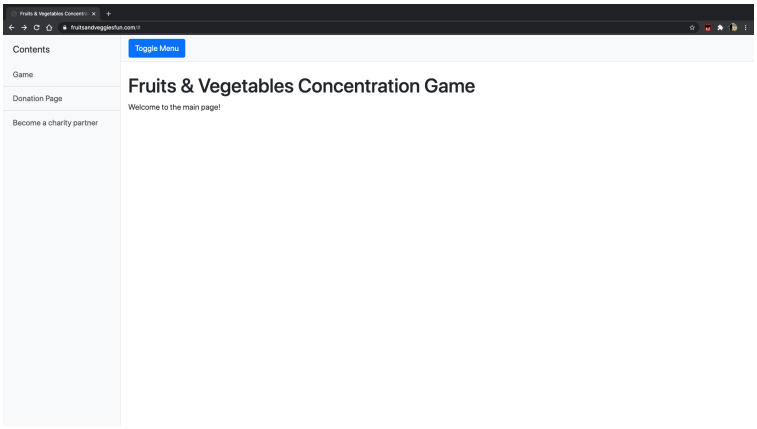


➔ Using the cPanel interface, compress the source web files into a zip file and upload to the public web folder using the file manager.





➔ Make sure website visibility is set to public and type the domain name into a web browser of your choice to see if the content is up.



Removal:

- ➔ Ensure financial information (such as transactions) and other persistent data are cleared from the server
- ➔ Backup all source files to GitHub
- ➔ Purge all files located in the web root folder
- ➔ Delete the domain which was used

Training for customer:

The customer will be provided with 2 forms of training, video and interactive. The training video will consist of steps on setting up the hosting environment, file management with cPanel, and a demonstration of the working product. The interactive portion will require the trainee to install the website contents on a local host to learn the basics of file management. This will provide adequate preparation for updating/maintaining the main website.

Recovery plan & rolling back (version control):

The codebase for the product should be tracked with version control software such as git. The customer should store the web source files in a repository such as GitHub and have redundant copies of source code on different machines. Ensure that there is a stable branch of the codebase, and only pull from the stable branch when updating the site. It is also recommended to maintain a testing (local) environment along with a production (web) environment. If a rollback is necessary due to a faulty update, go to a prior (stable) commit with git, and pull from the codebase. It is recommended that the individual who finds a faulty update notify the repository owner immediately after updating the website.