 Shell Commands (Git Bash):

* cd "//DavServe/home/David/CNC FILES/COASTERS/Vector-Woodworks"
  + Changes the current directory to the vector-woodworks folder on the external drive.
* cd ~/Documents
  + Changes the current directory to the Documents folder in your home directory (e.g., C:\Users\dheis\Documents).
* cd ~/Documents/vector-woodworks
  + Changes to the vector-woodworks folder in Documents (used in alternative C drive setup).
* cd "/DavServe/home/David/CNC FILES/COASTERS/Vector-Woodworks/images"
  + Changes to the images folder within vector-woodworks.
* cd ~/Documents/Vector-Woodworks-New
  + Changes to a new folder (e.g., for moving the repository to the C drive).
* cd ..
  + Moves up one directory level (e.g., from Vector-Woodworks to COASTERS).
* pwd
  + Prints the current working directory path (e.g., //DavServe/home/David/CNC FILES/COASTERS/Vector-Woodworks).
* ls
  + Lists files and folders in the current directory (e.g., to check for index.html or images).
* ls images
  + Lists files in the images folder to verify image filenames.
* mkdir images
  + Creates an images folder in the current directory.
* mkdir vector-woodworks
  + Creates a vector-woodworks folder (used when setting up the repository).
* mkdir Vector-Woodworks-New
  + Creates a new folder for moving the repository to the C drive.
* cp -r "//DavServe/home/David/CNC FILES/COASTERS/Vector-Woodworks"/\* ~/Documents/Vector-Woodworks-New/
  + Copies all files from vector-woodworks on the external drive to a new folder on the C drive.
* cp index.html index.html.bak
  + Creates a backup copy of index.html named index.html.bak.
* cp -r . ../vector-woodworks-backup
  + Copies the entire current directory to a backup folder one level up.
* curl -o images/coaster-holder.jpg https://cdn.shopify.com/s/files/1/0911/9134/5440/files/IMG-0302.jpg?v=1733328177
  + Downloads an image from a Shopify URL and saves it as coaster-holder.jpg in the images folder.
* nano index.html
  + Opens index.html in the nano text editor for editing in Git Bash.
* ls -ld .
  + Lists detailed information (including ownership) for the current directory.

 Git Commands:

* git --version
  + Checks the installed Git version.
* git config --global user.name "Your Name"
  + Sets your name for Git commit authorship globally.
* git config --global user.email "your.email@example.com"
  + Sets your email for Git commit authorship globally.
* git config --global --add safe.directory '%(prefix)///DavServe/home/David/CNC FILES/COASTERS/Vector-Woodworks'
  + Marks the repository path as trusted to bypass “dubious ownership” errors.
* git config --global --add safe.directory "//DavServe/home/David/CNC FILES/COASTERS/Vector-Woodworks"
  + Alternative syntax for marking the repository as safe.
* git init
  + Initializes a new Git repository in the current directory.
* git status
  + Shows the current state of the working directory and staging area (e.g., modified or staged files).
* git add .
  + Stages all changes (new, modified, deleted files) in the current directory for the next commit.
* git add index.html
  + Stages changes to index.html for the next commit.
* git commit -m "Message"
  + Commits staged changes with a descriptive message (e.g., “Updated image sources”).
* git remote add origin https://github.com/your-username/vector-woodworks.git
  + Links the local repository to a remote GitHub repository.
* git remote -v
  + Lists the remote repository URLs to verify the origin link.
* git push -u origin master
  + Pushes committed changes to the main branch of the remote repository and sets upstream tracking.
* git push origin master
  + Pushes committed changes to the main branch (or master if applicable).
* git restore index.html
  + Discards uncommitted changes to index.html, reverting to the last committed version.
* git restore .
  + Discards all uncommitted changes in the working directory.
* git restore --staged index.html
  + Unstages index.html from the commit area, keeping changes in the working directory.
* git restore --staged .
  + Unstages all files from the commit area.
* git reset HEAD index.html
  + Unstages index.html (alternative for older Git versions).
* git reset HEAD .
  + Unstages all files (alternative for older Git versions).
* git reset --soft HEAD~1
  + Undoes the last commit, keeping changes in the working directory/staging area.
* git reset --hard HEAD~1
  + Undoes the last commit and discards changes, reverting to the previous commit.
* git revert <commit-id>
  + Creates a new commit that reverses the changes of the specified commit.
* git push origin main --force
  + Force-pushes changes to overwrite the remote main branch (use with caution).
* git log --oneline
  + Displays a concise commit history with commit IDs and messages.
* git checkout -- index.html
  + Discards uncommitted changes to index.html (alternative for older Git versions).
* git checkout -- .
  + Discards all uncommitted changes in the working directory (alternative for older Git versions).