

# Introduction

vi/vim

vimium

nvim

window managers

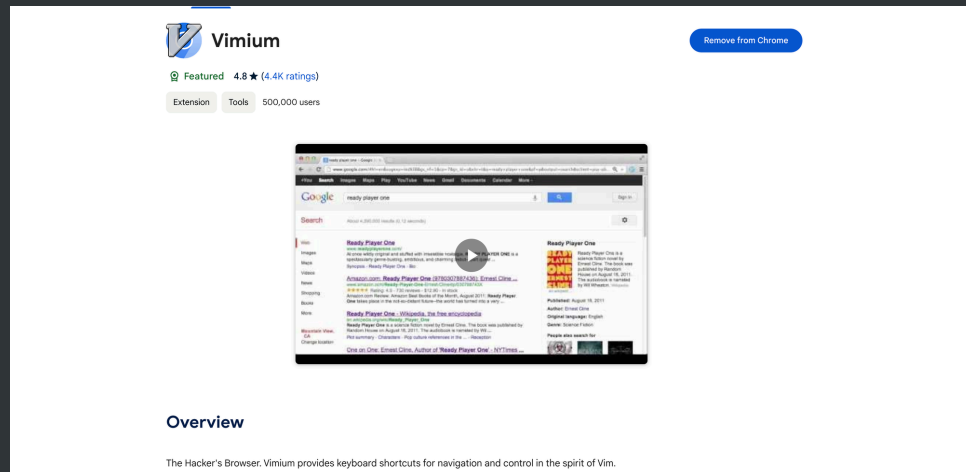
dotfiles

tmux



vi is a terminal based editor

# vimium



Vimium lets you navigate the browser with just your keyboard

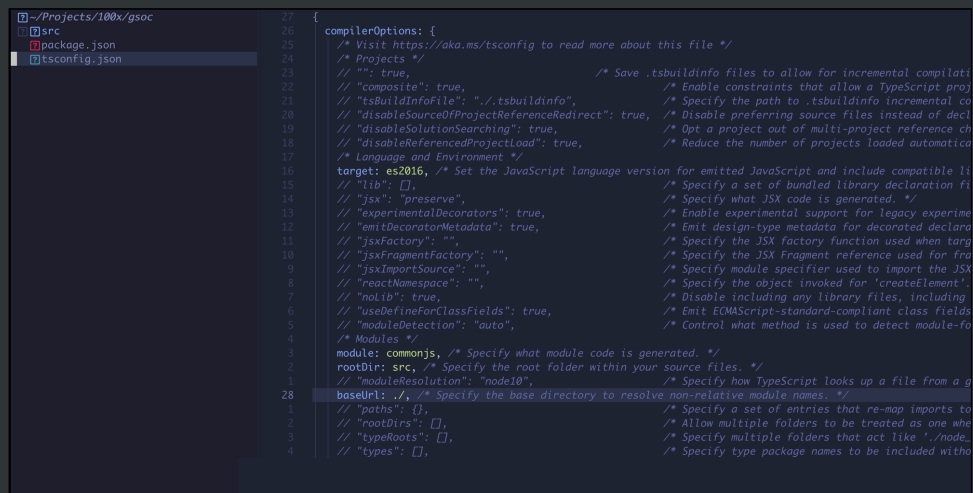
Chrome link -

<https://chromewebstore.google.com/detail/vimium/dbepggeogbaibhgnhhndojpepiihcmeb>

## Helpful keymaps

1. ? - help
2. d, u
3. j, k
4. r to reload
5. f, F to see links (F opens it in a new tab)
6. Tab management
  1. J → Left tab
  2. K → Right tab
  3. g0, g\$ → First tab, Last tab

# Neovim



```
27 {
28   compilerOptions: {
29     /* Visit https://aka.ms/tsconfig to read more about this file */
30     /* Projects */
31     // "": true, // Save .tsbuildinfo files to allow for incremental compilation
32     // "composite": true, // Enable constraints that allow a TypeScript project
33     // "tsBuildInfoFile": "./.tsbuildinfo", // Specify the path to .tsbuildinfo incremental co
34     // "disableSourceOfProjectReferenceRedirect": true, // Disable preferring source files instead of decl
35     // "disableSolutionSearching": true, // Opt a project out of multi-project reference ch
36     // "disableReferencedProjectLoad": true, // Reduce the number of projects loaded automatica
37     /* Language and Environment */
38     target: es2016, // Set the JavaScript language version for emitted JavaScript and include compatible li
39     // "lib": [], // Specify a set of bundled library declaration fi
40     // "jsx": "preserve", // Specify what JSX code is generated.
41     // "experimentalDecorators": true, // Enable experimental support for legacy experime
42     // "emitDecoratorMetadata": true, // Emit design-type metadata for decorated declara
43     // "jsxFactory": "", // Specify the JSX factory function used when targ
44     // "jsxFragmentFactory": "", // Specify the JSX Fragment reference used for fra
45     // "jsxImportSource": "", // Specify module specifier used to import the JSX
46     // "reactNamespace": "", // Specify the object invoked for 'createElement'.
47     // "noLib": true, // Disable including any library files, including
48     // "useDefineForClassFields": true, // Emit ECMAScript-standard-compliant class fields
49     // "moduleDetection": "auto", // Control what method is used to detect module-fo
50     /* Modules */
51     module: commonjs, // Specify what module code is generated.
52     rootDir: src, // Specify the root folder within your source files.
53     // "moduleResolution": "node10", // Specify how TypeScript looks up a file from a g
54     baseUrl: ./, // Specify the base directory to resolve non-relative module names.
55     // "paths": [], // Specify a set of entries that re-map imports to
56     // "rootDirs": [], // Allow multiple folders to be treated as one whe
57     // "typeRoots": [], // Specify multiple folders that act like './node_
58     // "types": [], // Specify type package names to be included witho
```

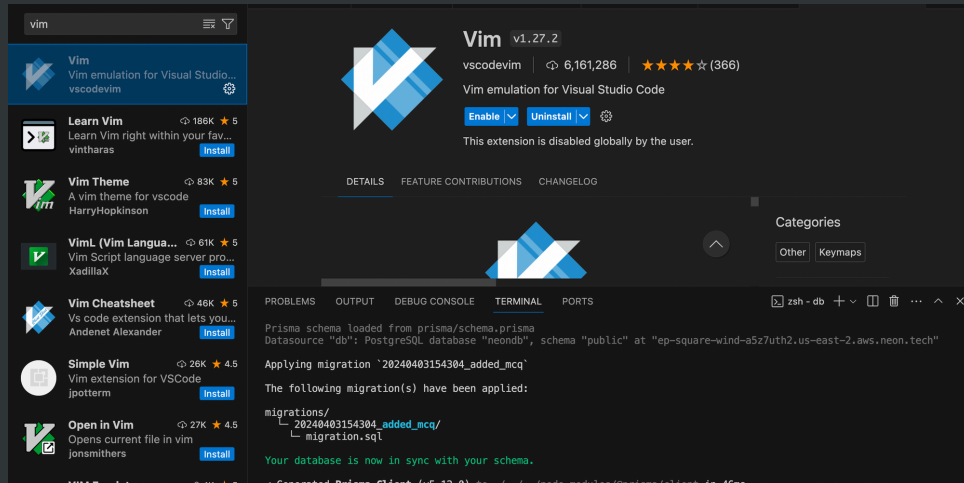
nvim is a fork of vim. It provides you a rich **plugin** system that lets you create plugins in **vimscript** or **lua**

Ref - <https://neovim.io/>

Easiest way to get started is to use some default configurations

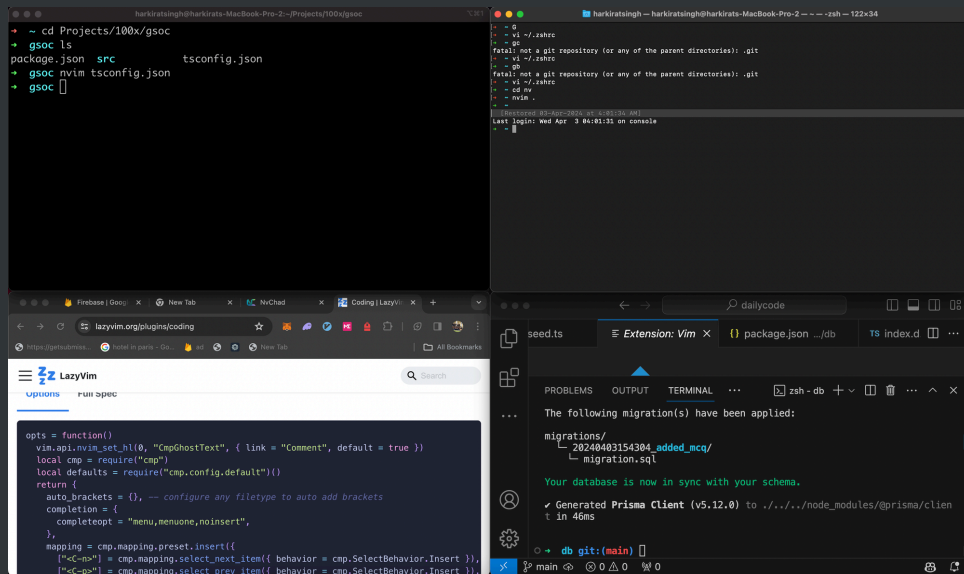
1. nvchad - <https://nvchad.com/>
2. lazyvim - <https://www.lazyvim.org/>

# Vim in VSCode



The vim plugin in VSCode lets you use vim keymaps inside VSCode to start a slow migration towards vi

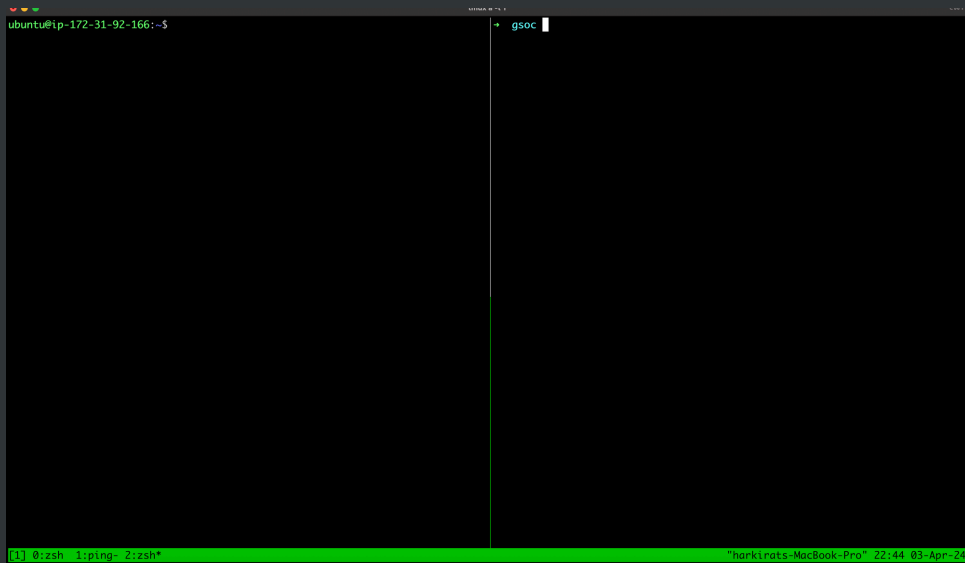
# Window managers



Window managers let you move around and position windows on your machine

1. i3 (linux)
2. Rectangle (mac)
  1. ctrl + option +
    1. left arrow key
    2. right arrow key
  3. enter
  4. U
  5. I
  6. J
  7. K

# TMux



Ref - <https://github.com/tmux/tmux/wiki>

TMux stands for terminal multiplexer

It let's you create multiple

1. Sessions
2. Windows
3. Panes



I have an alias for `t` to point to `tmux`

```
alias t=tmux
```

Common commands

1. `tmux`
2. `ctrl + b + d` - Detach from the session
3. `tmux ls`
4. `tmux a -t 0`
5. `tmux new -s aws`
6. `tmux kill-session -t aws`
7. Pane management
  1. `ctrl + b + %` - split vertically
  2. `ctrl + b + "` - split horizontally
  3. `ctrl + b + arrow key` - move around
  4. `ctrl + b + q` - see index of pane
8. Window management
  1. `ctrl + b + c` - create new window in the same session
  2. `ctrl + b + w` - List of sessions + windows
  3. `ctrl + b + ,` - Rename the window
  4. `ctrl + b + n` - Move around sequentially
  5. `ctrl + b + number` - Move to window number `number`
  6. `ctrl + b + x` or `ctrl + b + &` - kill pane

# Shell config files

Shell config files let you run some code when the shell starts. You can use this to add aliases and functions

.bashrc or .zshrc

1. aliases
2. complex functions

```
export ZSH="$HOME/.oh-my-zsh"
ZSH_THEME="robbyrussell"
source $ZSH/oh-my-zsh.sh

export NVM_DIR="/Users/harkiratsingh/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm

export PATH="$PATH:/Users/harkiratsingh/nvim-macos/bin"
alias gc=git commit -m
alias gh git push origin HEAD
alias gs='git status'
alias gaa='git add --all'
alias gd='git diff'

alias .4='cd ../../../../'
alias .5='cd ../../../../..'

alias t=tmux
function hg() {
    history | grep "$1";
}
```

Copy



