

# Introduction

vi/vim

vimium

nvim

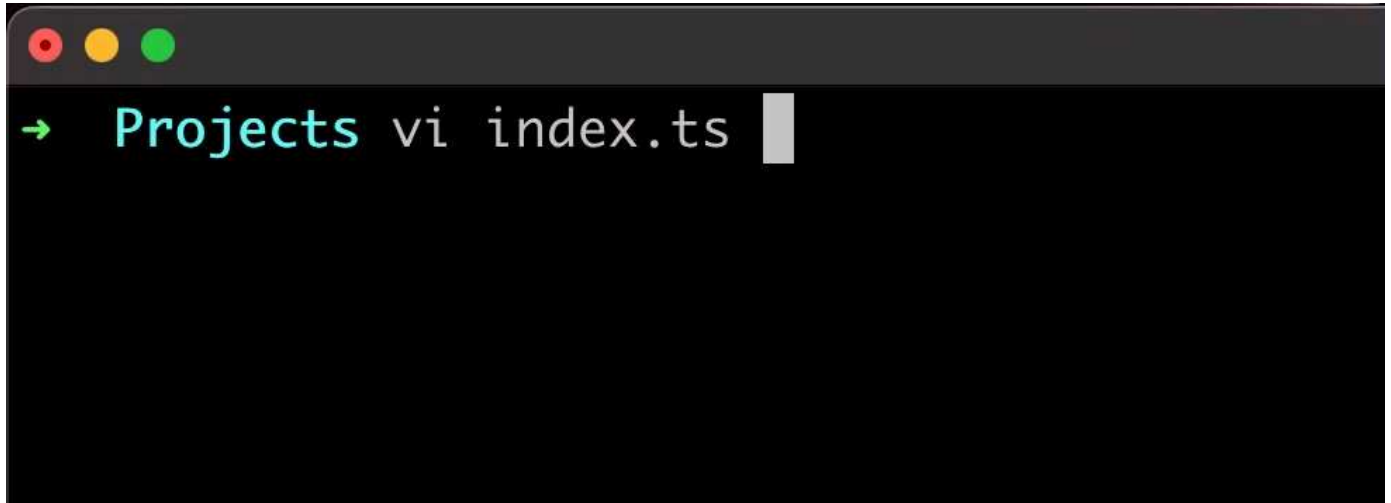
window managers

dotfiles

tmux

# vi

vi is a terminal based editor



## Normal mode

1. h → left, l → right, j → down, k → up, (हंसते लोग लगातार रास्ते जाम देख कर उदास होते हैं)
2. 0, \$
3. e, b
4. gg, G
5. p ⇒ paste

## Visual mode

1. h → left, l → right, j → down, k → up
2. 0, \$
3. e, b
4. gg, G
5. y ⇒ copy
6. d ⇒ cut

# Insert mode

Type away!

## Extra commands

### 1. Exiting out of vi

1. `:q`

2. `:q!`

3. `:wq!`

### 2. Undo redo

1. `u` - undo

2. `ctrl + r` - redo

### 3. Search

1. `/` + what\_to\_search

2. `:line_number` - go to line

# vimium


[Remove from Chrome](#)


## Overview

The Hacker's Browser. Vimium provides keyboard shortcuts for navigation and control in the spirit of Vim.

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 {
26   compilerOptions: {
25     /* Visit https://aka.ms/tsconfig to read more about this file */
24     /* Projects */
23     // "": true, /* Save .tsbuildinfo files to allow for incremental compilation */
22     // "composite": true, /* Enable constraints that allow a TypeScript project to be a part of a composite project */
21     // "tsBuildInfoFile": "./.tsbuildinfo", /* Specify the path to .tsbuildinfo incremental compilation file */
20     // "disableSourceOfProjectReferenceRedirect": true, /* Disable preferring source files instead of declaration files */
19     // "disableSolutionSearching": true, /* Opt a project out of multi-project reference checking */
18     // "disableReferencedProjectLoad": true, /* Reduce the number of projects loaded automatically by TypeScript */
17     /* Language and Environment */
16     target: es2016, /* Set the JavaScript language version for emitted JavaScript and include compatible library declarations */
15     // "lib": [], /* Specify a set of bundled library declaration files that defines the target environment */
14     // "jsx": "preserve", /* Specify what JSX code is generated. */
13     // "experimentalDecorators": true, /* Enable experimental support for legacy experimental decorators */
12     // "emitDecoratorMetadata": true, /* Emit design-type metadata for decorated declarations in source files */
11     // "jsxFactory": "", /* Specify the JSX factory function used when targeting React */
10     // "jsxFragmentFactory": "", /* Specify the JSX Fragment reference used for fragments */
9     // "jsxImportSource": "", /* Specify module specifier used to import the JSX factory modules */
8     // "reactNamespace": "", /* Specify the object invoked for 'createElement'. This defaults to 'React' when targeting browser environments. */
7     // "noLib": true, /* Disable including any library files, including the standard library */
6     // "useDefineForClassFields": true, /* Emit ECMAScript-standard-compliant class fields */
5     // "moduleDetection": "auto", /* Control what method is used to detect module-format JS files */
4     /* Modules */
3     module: commonjs, /* Specify what module code is generated. */
2     rootDir: src, /* Specify the root folder within your source files. */
1     // "moduleResolution": "node10", /* Specify how TypeScript looks up a file from a given module specifier */
28    baseUrl: ., /* Specify the base directory to resolve non-relative module names. */
1     // "paths": {}, /* Specify a set of entries that re-map imports to additional lookup locations */
2     // "rootDirs": [], /* Allow multiple folders to be treated as one when resolving */
3     // "typeRoots": [], /* Specify multiple folders that act like './node_modules/@types' */
4     // "types": [], /* Specify type package names to be included without a type specifier */
  }

```

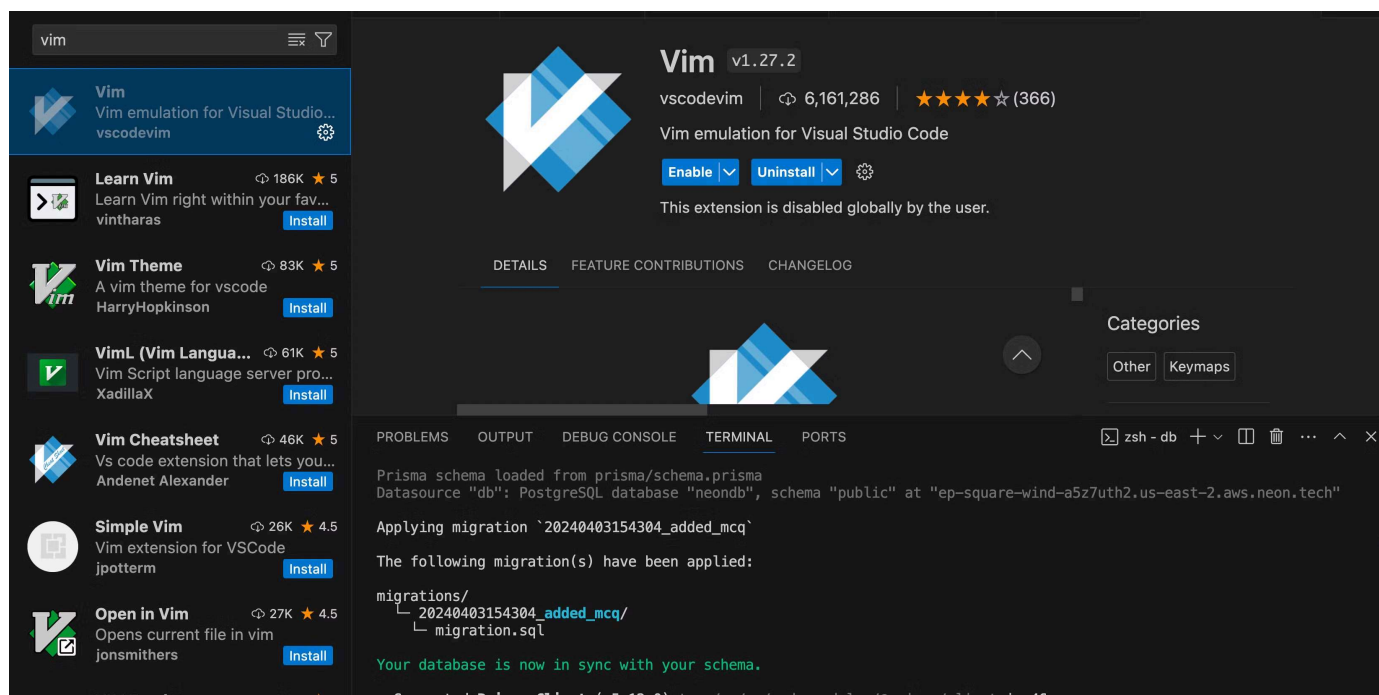
nvim is a fork of vim. It provides you a rich **plugin** system that lets you create plugins in **vimscript** or **lua**

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/>

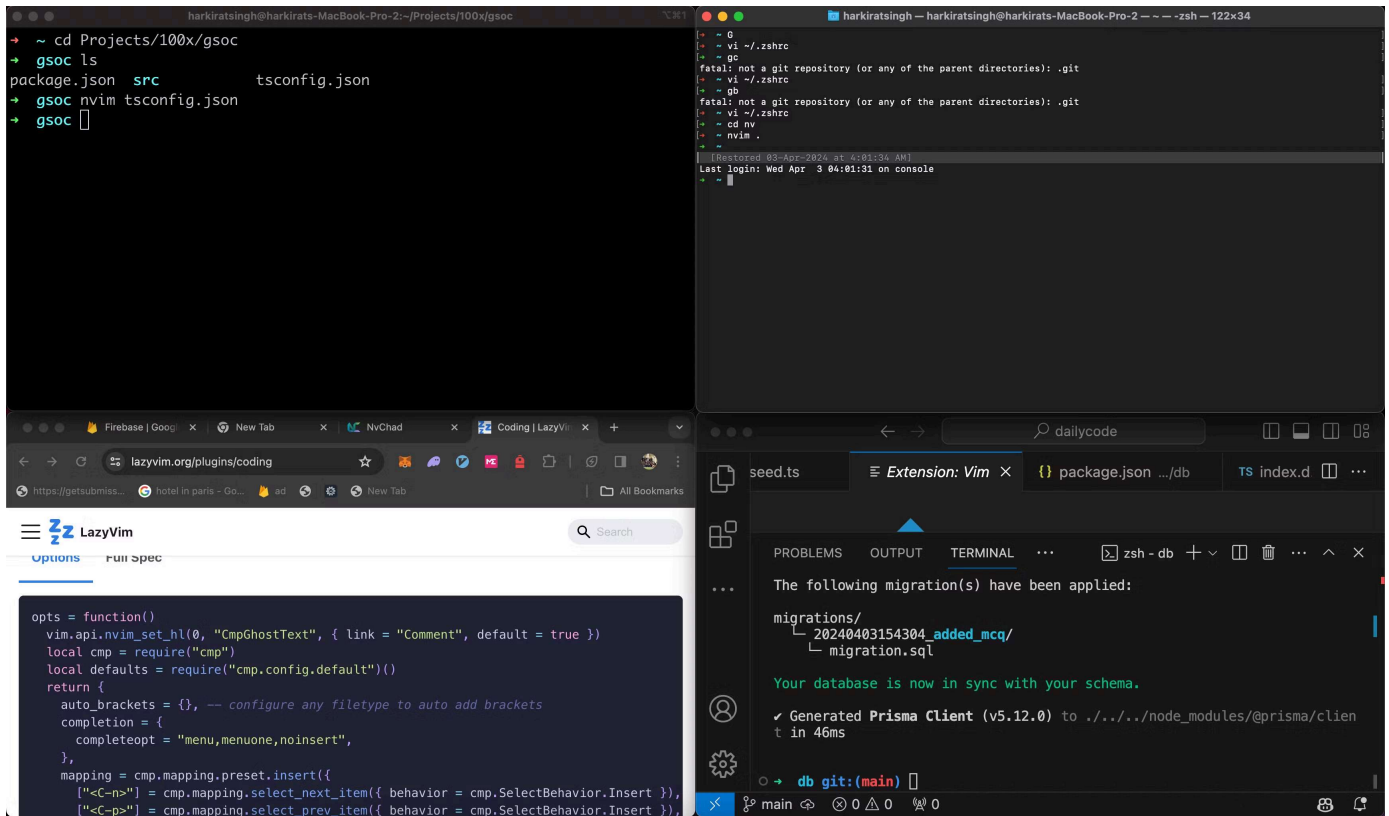
# Vim in VSCode



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 + optoin +

1. left arrow key

2. right arrow key

3. enter

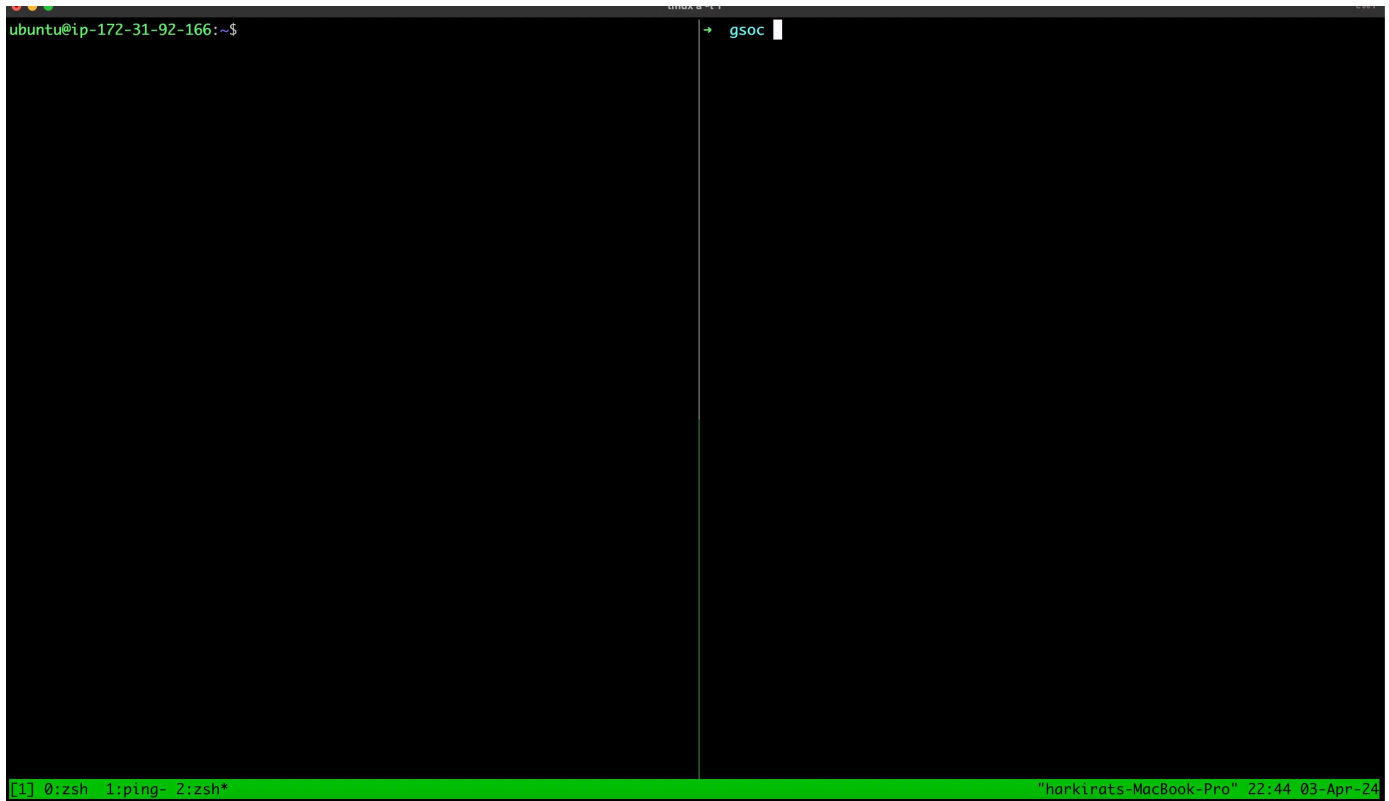
4. U

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

7. `ctrl + b + w` followed by `ctrl + b + x` - kill window

# 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";
}
```

