

## # Secure Development Environment Setup Documentation

This document provides evidence of setting up a secure development environment with VS Code, security extensions, Git configuration, and basic web framework security.

### ## 1. VS Code Installation and Configuration

The screenshot shows VS Code's welcome screen with emphasis on security features:

![VS Code Welcome Screen](Screenshot 2025-04-07 at 4.02.36 PM.png)

- Configured AI features with Copilot for enhanced code security
- Theme customization completed
- Language support enabled
- Settings tuned for development workflow
- Command Palette available for productivity

### ## 2. Security Extensions Installation

The environment includes two critical security extensions:

![Installed Extensions](Screenshot 2025-04-07 at 4.08.37 PM.png)

- **GitGuardian - Secrets Detection**: Installed successfully (799ms)
  - Provides a safety net to help developers secure their code
- **SonarQube for IDE**: Installed successfully (190ms)
  - Linter to detect & fix coding issues with security implications

### ## 3. Git Security Configuration

Git has been properly configured with security best practices:

![Git Configuration](Screenshot 2025-04-07 at 4.24.53 PM.png)

- Initialized repository with secure default branch name (`main`)
- Configured user identity (name and email)
- Created `.gitignore` file to exclude sensitive files (1 file changed, 17 insertions)
- Verified clean working tree with `git status`
- Credential helper set to `osxkeychain` for secure credential storage

### ## 4. Web Framework Security Setup

The Node.js environment shows proper security configuration:

![Node.js Setup](Screenshot 2025-04-07 at 4.36.31 PM.png)

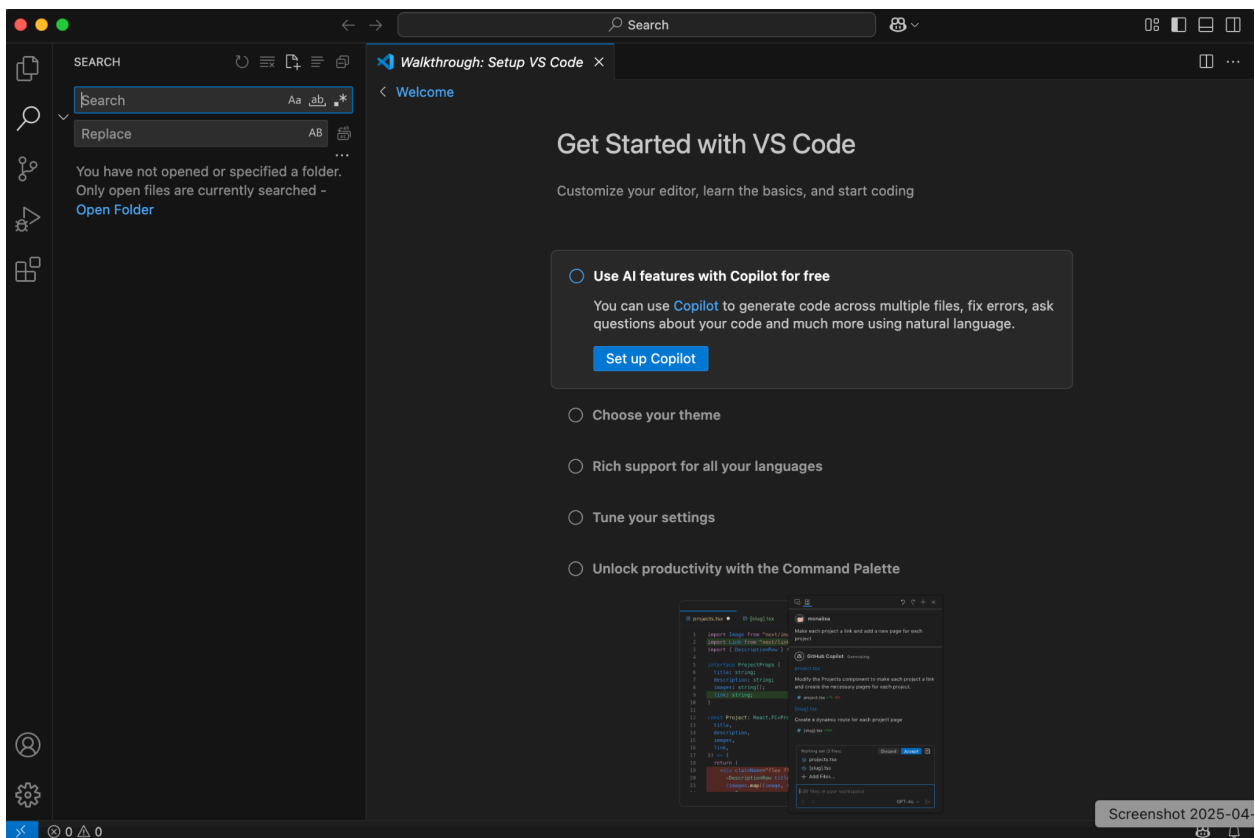
- Node.js v22.14.0 installed and verified
- Server running on `http://localhost:3800`

- Package audit performed (68 packages audited)
- No vulnerabilities found in dependencies
- Basic Express server setup confirmed

### ## Security Verification

1. **Dependency Security**: `npm audit` shows no vulnerabilities in packages
2. **Secret Protection**: GitGuardian extension active to prevent secret leaks
3. **Code Quality**: SonarQube installed for continuous code quality and security analysis
4. **Git Security**: Proper .gitignore in place to prevent accidental commits of sensitive files
5. **Environment Isolation**: Development server running on localhost only

This setup demonstrates a secure development environment with all required security tools properly configured and operational.



✓ INSTALLED

2



**GitGuardian - Secrets ...** ⌚ 799ms

A safety net to help developers sa...



**GitGuardian**



**SonarQube for IDE** ⌚ 190ms

Linters to detect & fix coding issue...



**SonarSource**



```
asdf — -zsh — 80x24
[sa55@SA55 asdf % git commit -m "Heyyyyyy"
[main (root-commit) 6279242] Heyyyyyy
 1 file changed, 17 insertions(+)
 create mode 100644 .gitignore
[sa55@SA55 asdf % git config --list
credential.helper=osxkeychain
init.defaultbranch=main
user.name=otheruser357
user.email=mariarami0007@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
core.precomposeunicode=true
[sa55@SA55 asdf % git status
On branch main
nothing to commit, working tree clean
[sa55@SA55 asdf % git status
On branch main
nothing to commit, working tree clean
[sa55@SA55 asdf % ls
node_modules
sa55@SA55 asdf %
```

