Welcome to your VS Code Extension

What's in the folder

- This folder contains all of the files necessary for your extension.
- `package.json` - this is the manifest file in which you declare your extension and command.
  - The sample plugin registers a command and defines its title and command name. With this information VS Code can show the command in the command palette. It doesn't yet need to load the plugin.
- `src/extension.ts` - this is the main file where you will provide the implementation of your command.
  - The file exports one function, `activate`, which is called the very first time your extension is activated (in this case by executing the command). Inside the `activate` function we call `registerCommand`.
  - We pass the function containing the implementation of the command as the second parameter to `registerCommand`.

Get up and running straight away

- Press `F5` to open a new window with your extension loaded.
- Run your command from the command palette by pressing (`Ctrl+Shift+P` or `Cmd+Shift+P` on Mac) and typing `Hello World`.
- Set breakpoints in your code inside `src/extension.ts` to debug your extension.
- Find output from your extension in the debug console.

Make changes

- You can relaunch the extension from the debug toolbar after changing code in `src/extension.ts`.
- You can also reload (`Ctrl+R` or `Cmd+R` on Mac) the VS Code window with your extension to load your changes.

Explore the API

- You can open the full set of our API when you open the file `node_modules/@types/vscode/index.d.ts`.

Run tests

- Install the [Extension Test Runner](#)
- Run the "watch" task via the **Tasks: Run Task** command. Make sure this is running, or tests might not be discovered.
- Open the Testing view from the activity bar and click the Run Test" button, or use the hotkey `Ctrl/Cmd + ; A`
- See the output of the test result in the Test Results view.
- Make changes to `src/test/extension.test.ts` or create new test files inside the `test` folder.
  - The provided test runner will only consider files matching the name pattern `**.test.ts`.
  - You can create folders inside the `test` folder to structure your tests any way you want.

Go further

- [Follow UX guidelines](#) to create extensions that seamlessly integrate with VS Code's native interface and patterns.
- Reduce the extension size and improve the startup time by [bundling your extension](#).
- [Publish your extension](#) on the VS Code extension marketplace.

- Automate builds by setting up Continuous Integration.
- Integrate to the report issue flow to get issue and feature requests reported by users.