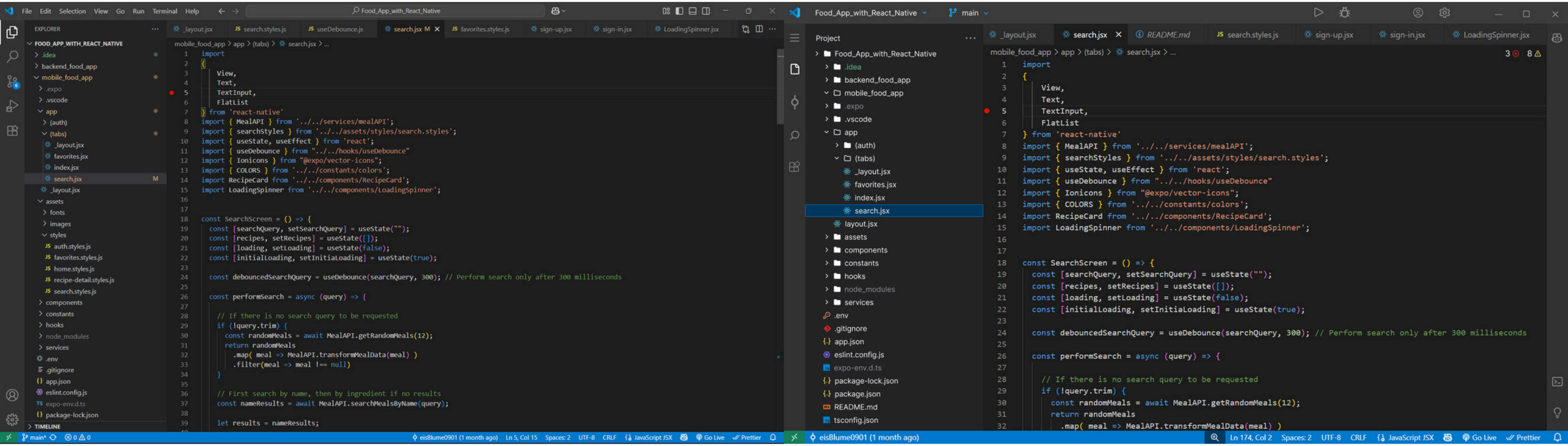
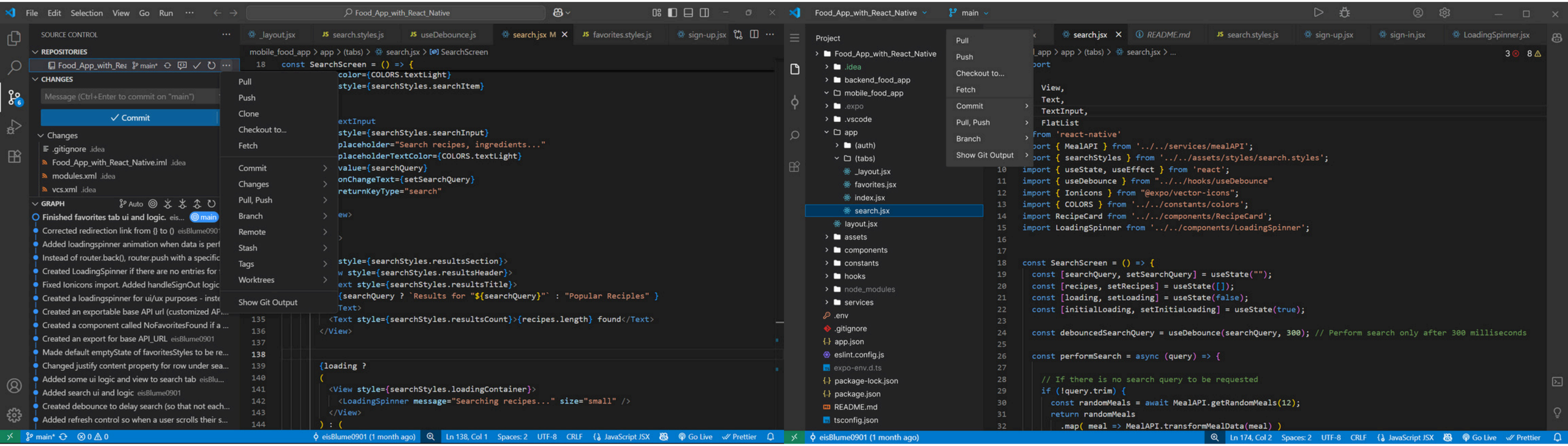
Placement of Icons



Changes done

1. Visibility of terminal by providing an icon at the right side for easy access (since developers used it at all times)
2. Removal of some icons (such as toggle bars) since the provided icon buttons are sufficient
3. Placing the problem and warning icon to the top right corner of the code editor for visibility (since putting it below at the bottom bar could be easily overlooked by developers especially beginners, debugging errors are one of the things that developers deal with)
4. Instead of named buttons (File, Edit, Selection, View, Go, Run, Terminal, Help), a hamburger icon was replaced for cleaner user interface (so that it does not look so busy and overwhelming)
5. Git Branch was originally placed at the bottom and was moved at the top bar (developers always use Git branch and commands for version history, it is significant especially when collaborating with other developers)
6. User Profile and Settings icon was moved from left side bar to top bar (the left side bar can get occupied with extensions installed - unless it is configured to be hidden manually by the developer
7. Run and Debug has the same reason for User Profile and Settings
8. Filled and unfilled folder icons was added to easily indicate if a file under a certain folder is currently edited.
9. Renaming some labels from Explorer to Project - since it matches to real world setting and much more familiar to people in general
10. Changing some icon designs to simplified version such as Git Commit was used instead of Git Branch - since it makes more sense. Maximize Window turned into a simple rectangle icon. Explorer/Project that has two stacked paper changed into one single paper

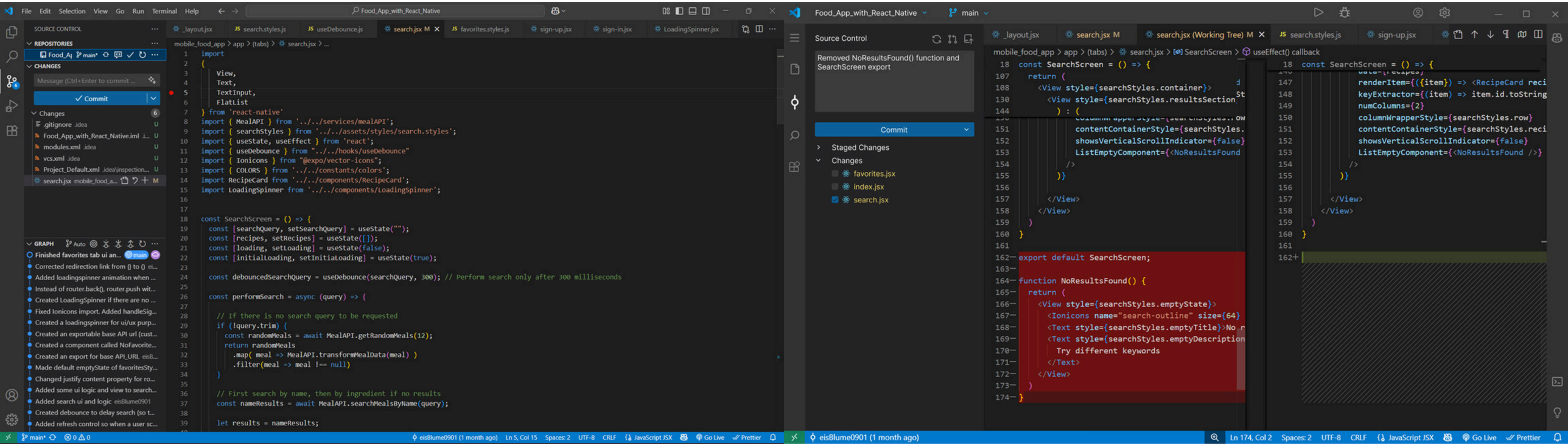
GitHub Commands



Changes done

1. Available Git options to Git Branch icon
2. Removed some Git commands or options. Only the top Git commands or options commonly used was chosen. Senior developers have an option to use the terminal and Git commands if they want for advanced Git configuration.

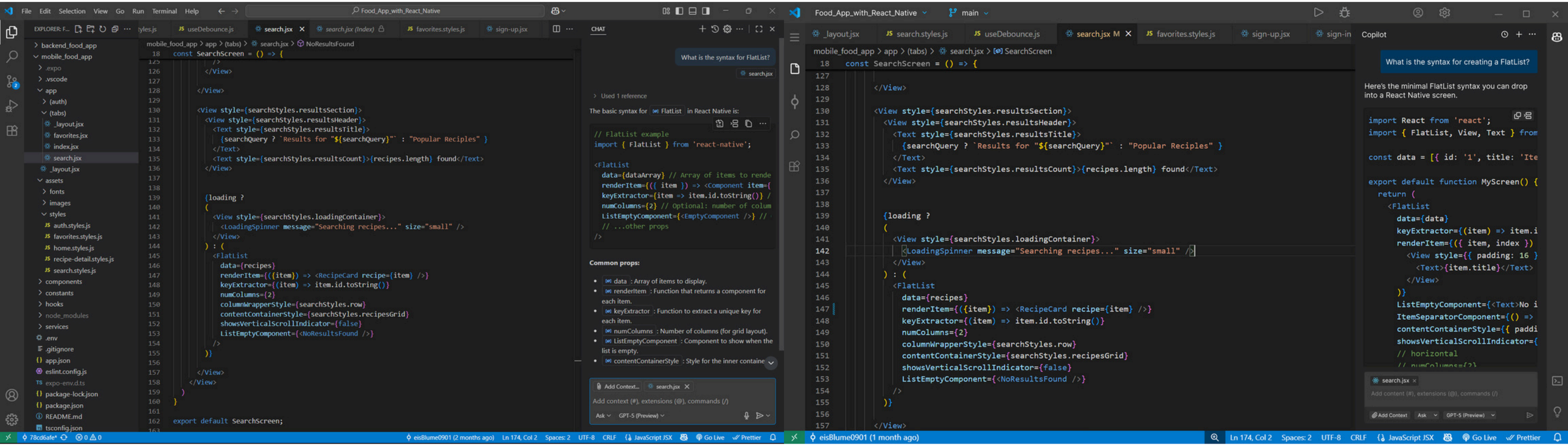
Commiting a File



Changes done

1. Source Control is too busy. Visual Studio Code integrated Graph which is really too overwhelming for beginners. Although it is a good tool for advanced/senior developers (which they have the option to enable), in default mode, it is better to just focus on committing files. Therefore, the Graph is removed
2. Created a hierarchy / grouped the staged changes and changes instead of using labeling M (modified), A (added), and D (deleted) which is really cryptic. Grouping them in semantic organization could improve cognitive mapping since it is a human-readable structure.
3. Checkbox was used instead of a + (add) or x (discard) button in Git source interface. The add or discard button suggest an irreversible actions which can feel abrupt, risky, and even cluttered. On the other hand, a checkbox is visually neutral and familiar, reducing cognitive load and improves visual consistency. For beginners, they can easily glance and instantly see which files they have included or not since check boxes are universally understood instead of being careful what to add or discard and decoding what M, A, or D means.

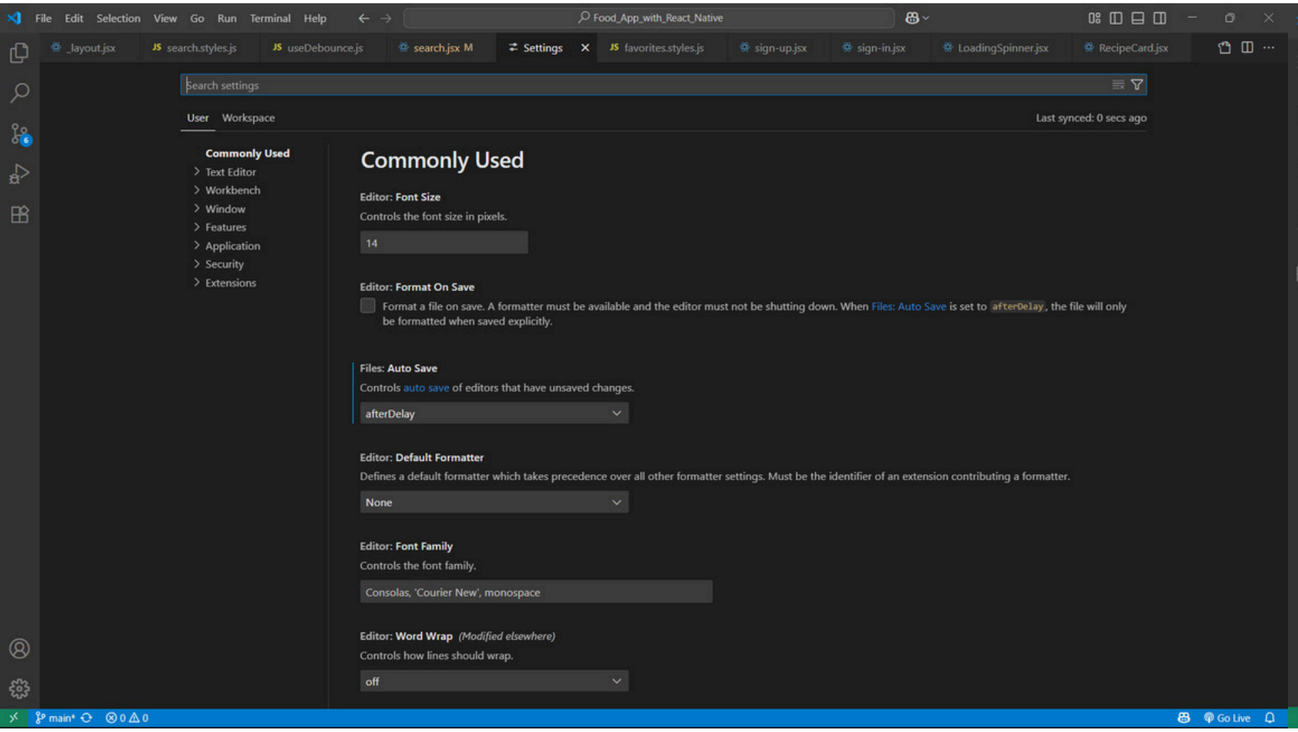
GitHub Copilot

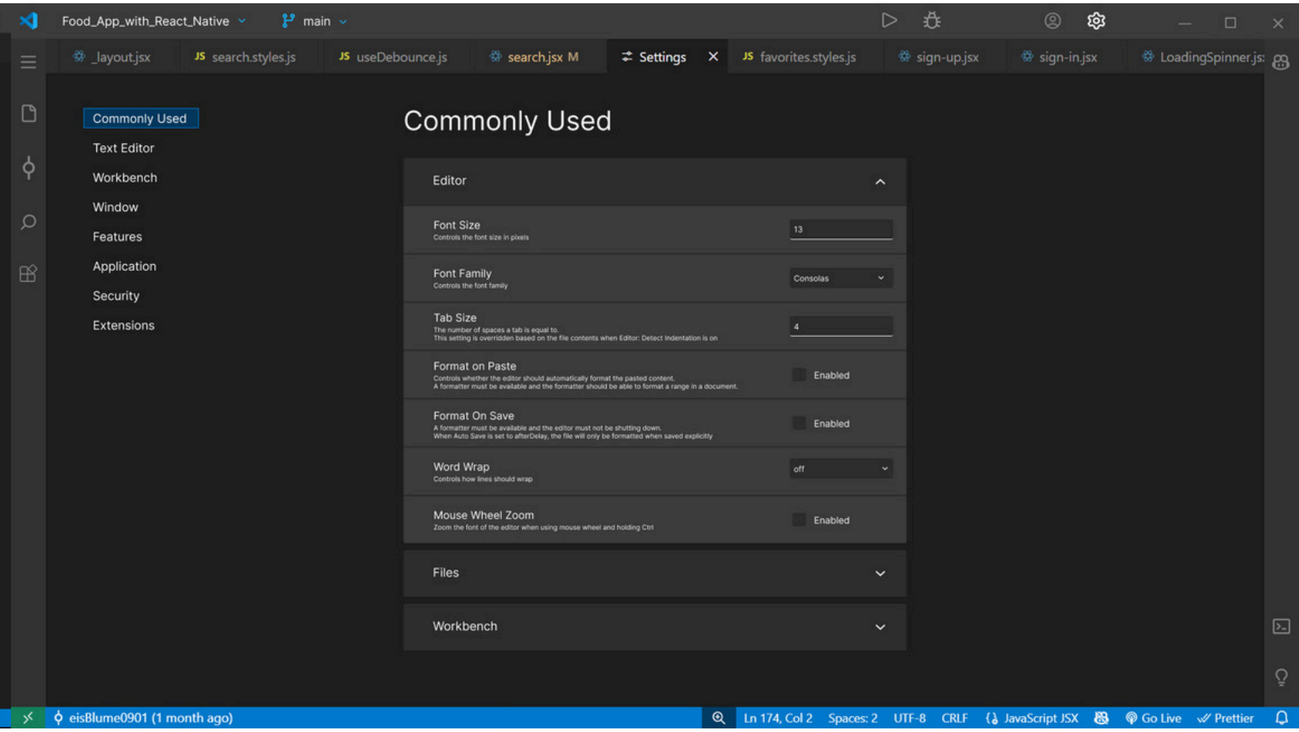


Changes done

1. The message input area in Copilot appears cluttered because it lacks a border, making it feel blended with the surrounding buttons/dropdowns/selections. Adding a border or boundary would help distinguish the message box to buttons/dropdowns/selections.
2. The code-generated output interface uses common actions for developers, such as copy-paste and insert-to-editor icons, as these are the typical operations developers perform. The "Apply to a specific file" option was not included, as it is unnecessary - developers generally prompt Copilot within their current working file.
3. The close, settings, and maximize buttons have been consolidated into a three-dot menu. Maximizing or dragging the chat window can now be integrated, with settings conveniently placed within the menu.

Settings





Changes done

1. Settings is cluttered and disorganized which is overwhelming to beginners. Although Visual Studio Code is designed to be highly customizable and extensible, its flexibility costs simplicity. Therefore, I grouped each category and created a proper hierarchy for Files, Editor, and Workbench of Commonly Used since it used to be spread out and not grouped.
2. I also fixed the number fields and dropdown options to have the same width length for visual consistency and cognitive ease.

Figma Link

403 ERROR

The request could not be satisfied.

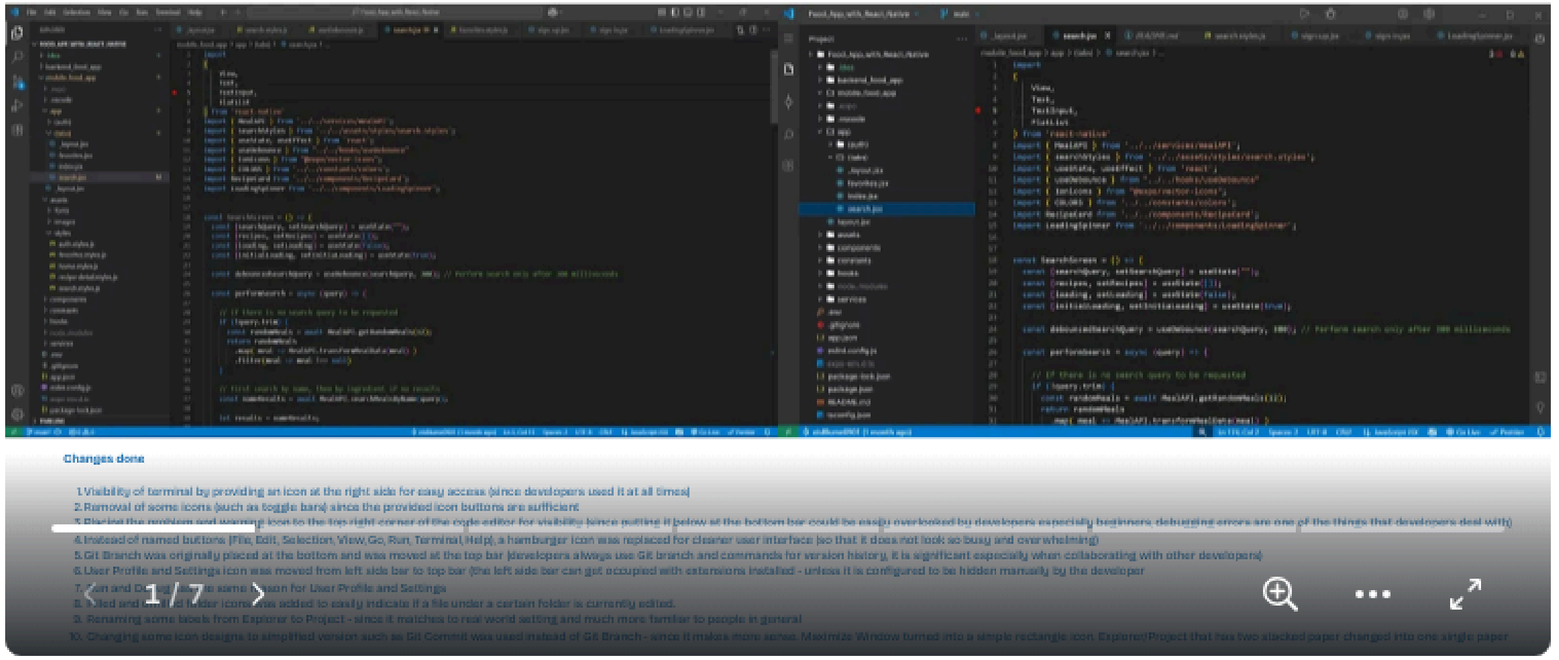
Request blocked. We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner.
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.

```
Generated by cloudfront (CloudFront)
Request ID: 6tcXuV3Zd-2ivbc4sewbz8nrQK2ZUiY4fROF_UQikpOOK8ut6zi-1Q==
```

https://www.figma.com/design/faId6gwJGBOIwbRYiu4eso/Anecito_VisualStudioCode_CodeEditor?node-id=0-3&t=KJznKrBh666qUcq8-1

Canva Link

Placement of Icons



https://www.canva.com/design/DAGyuBFhBVY/hxDbdHOKUvX8jPvxcIN83A/edit?utm_content=DAGyuBFhBVY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton