

## P3 - UI Mockups Screenshot for User Stories

1. **Repo Awareness** *As a developer, I want AI to have awareness of my entire folder (repo) of code, so that I can ask AI for a repo summary or change code without needing to manually append context.*

The screenshot shows the VS Code AI Assistant interface. The Explorer panel on the left shows the file structure: src, components (Header.tsx, Sidebar.tsx, ChatInterface.tsx), and utils (App.tsx, index.tsx). The main editor displays the Header.tsx file with the following code:

```
1 import React from 'react';
2 import { Menu, Search, Settings } from 'lucide-react';
3
4 export function Header() {
5   return (
6     <header className="flex items-center justify-between
7       p-2 bg-background border-b">
8       <div className="flex items-center gap-2">
9         <Menu className="w-5 h-5" />
10        <span className="font-medium">AI Code
11          Assistant</span>
12      </div>
13      <div className="flex items-center gap-2">
14        <Search className="w-5 h-5" />
15        <Settings className="w-5 h-5" />
16      </div>
17    </header>
18  );
19 }
```

The AI Assistant panel on the right shows the assistant's response to the query "Can you give me a summary of this repository?". The assistant provides a summary of the repository, mentioning the main components and the core logic.

Repository-aware coding assistant

Hello! I'm your AI coding assistant. I have full awareness of your repository and can help you with code generation, explanations, and refactoring. What would you like to work on?

Can you give me a summary of this repository?

This is a VS Code extension project for an AI coding assistant. The main components include:

- React Components: [Header component](#), [Sidebar navigation](#), and [ChatInterface](#) for the UI
- Core Logic: Main [App function](#) with [useState hooks](#) for state management

Ask about your code, generate files, or get

Repository Indexed • Editing Header.tsx

The screenshot shows the VS Code AI Assistant interface. The Explorer panel on the left shows the file structure: src, components (Header.tsx, Sidebar.tsx, ChatInterface.tsx), and utils (App.tsx, index.tsx). The main editor displays the App.tsx file with the following code:

```
1 import React from 'react';
2 import { Header } from './components/Header';
3 import { Sidebar } from './components/Sidebar';
4 import { ChatInterface } from
5   './components/ChatInterface';
6
7 function App() {
8   return (
9     <div className="flex flex-col h-screen">
10      <Header />
11      <div className="flex flex-1">
12        <Sidebar />
13        <main className="flex-1 p-4">
14          <h1>Welcome to AI Code Assistant</h1>
15          <p>Select a file to edit or start chatting
16            with AI</p>
17        </main>
18        <ChatInterface />
19      </div>
20    </div>
21  );
22 }
```

The AI Assistant panel on the right shows the assistant's response to the query "Can you give me a summary of this repository?". The assistant provides a summary of the repository, mentioning the main components and the core logic.

Repository-aware coding assistant

ChatInterface.tsx :45

- Core Logic: Main [App function](#) with [useState hooks](#) for state management
- Package Configuration: Standard VS Code extension setup with OpenAI integration in [package.json](#)

The architecture follows a typical VS Code extension pattern with a clean React-based interface defined in the [default export](#).

Ask about your code, generate files, or get

Repository Indexed • Editing App.tsx

## 2. Repo Summaries with Citations As a developer, I want the AI's repo summaries to include direct code files or line citations so that I can better check explanations and quickly verify them in context

The screenshot shows the VS Code AI Assistant interface. On the left, the Explorer panel shows a file tree with 'src' containing 'components' (Header.tsx, Sidebar.tsx, ChatInterface.tsx) and 'utils' (App.tsx, index.tsx). The main editor displays the code for Header.tsx, with line 8 highlighted. The AI Assistant panel on the right provides a summary of the repository, including a list of React components and core logic, with citations to specific files and lines.

**AI Assistant**  
Repository-aware coding assistant

refactoring. What would you like to work on?

Can you give me a summary of this repository?

This is a VS Code extension project for an AI coding assistant. The main components include:

- React Components: [Header component](#), [Sidebar navigation](#), and [ChatInterface](#) for the UI
- Core Logic: Main [App function](#) with [useState hooks](#) for state management
- Package Configuration: Standard VS Code extension setup with OpenAI integration in [package.json](#)

Ask about your code, generate files, or get

Repository indexed • Editing Header.tsx

The screenshot shows the VS Code AI Assistant interface. On the left, the Explorer panel shows a file tree with 'src' containing 'components' (Header.tsx, Sidebar.tsx, ChatInterface.tsx) and 'utils' (App.tsx, index.tsx). The main editor displays the code for Header.tsx, with line 5 highlighted. The AI Assistant panel on the right provides a summary of the repository, including a list of React components and core logic, with citations to specific files and lines.

**AI Assistant**  
Repository-aware coding assistant

Can you high light the app core function for me?

I understand your request. Let me analyze your codebase and provide a helpful response based on the current context.

The main application structure is defined in [App.tsx with imports](#) and [component setup](#).

The project uses a typical React component architecture with separate files for [Header component](#), [Sidebar navigation](#), and [ChatInterface logic](#).

Ask about your code, generate files, or get

Repository indexed • Editing Header.tsx