



NaruhoDocs

User Guide

An AI-powered VS Code extension for intelligent documentation assistance.

Prepared by: JavaMee

GitHub Repository: <https://github.com/shinyeongbeh/NaruhoDocs>

Table Content

1.0 INTRODUCTION.....	3
2.0 INSTALLATION GUIDE.....	3
3.0 GETTING STARTED.....	3
3.1 Open NaruhoDocs Sidebar.....	3
3.2 Add a Document Tab.....	4
3.3 Switch Tab.....	5
3.4 Document Specific Features.....	6
4.0 CONFIGURATIONS.....	7
4.1 LLM Provider.....	8
4.2 Embedding Modes.....	10
4.3 Visualization Library.....	12
4.4 Other Settings.....	12
5.0 FEATURES.....	13
5.1 AI Chatbot.....	13
5.2 Rebuild RAG Vector Database.....	14
5.3 Clear Current Thread History.....	15
5.4 Beginner and Developer Mode.....	16
5.5 Documentation Drift Detection.....	17
5.6 General Tab Specific Features.....	18
5.6.1 Generate Documentation.....	18
5.6.2 Suggest Documentation Template.....	23
5.6.3 Visualize Project.....	28
5.7 Document Specific Features.....	31
5.7.1 Summarize Document.....	31
5.7.2 Translate Document.....	35
5.7.3 Check Grammar.....	39
5.7.4 Validate Markdown.....	43
5.7.5 Ignore All Issues.....	47
6.0 TROUBLESHOOTING.....	48
6.1 General Tab Icons Not Visible.....	48
6.2 Newly Opened Document Tab Not Showing.....	48
6.3 Chatbot Not Responding or Showing an Error Message.....	48
6.4 RAG or Embedding Errors.....	49
7.0 ACKNOWLEDGEMENT.....	50
8.0 USER FEEDBACK.....	50

1.0 INTRODUCTION

NaruhoDocs is an AI-powered VS Code extension that helps users write, read, and manage documentation efficiently.

2.0 INSTALLATION GUIDE

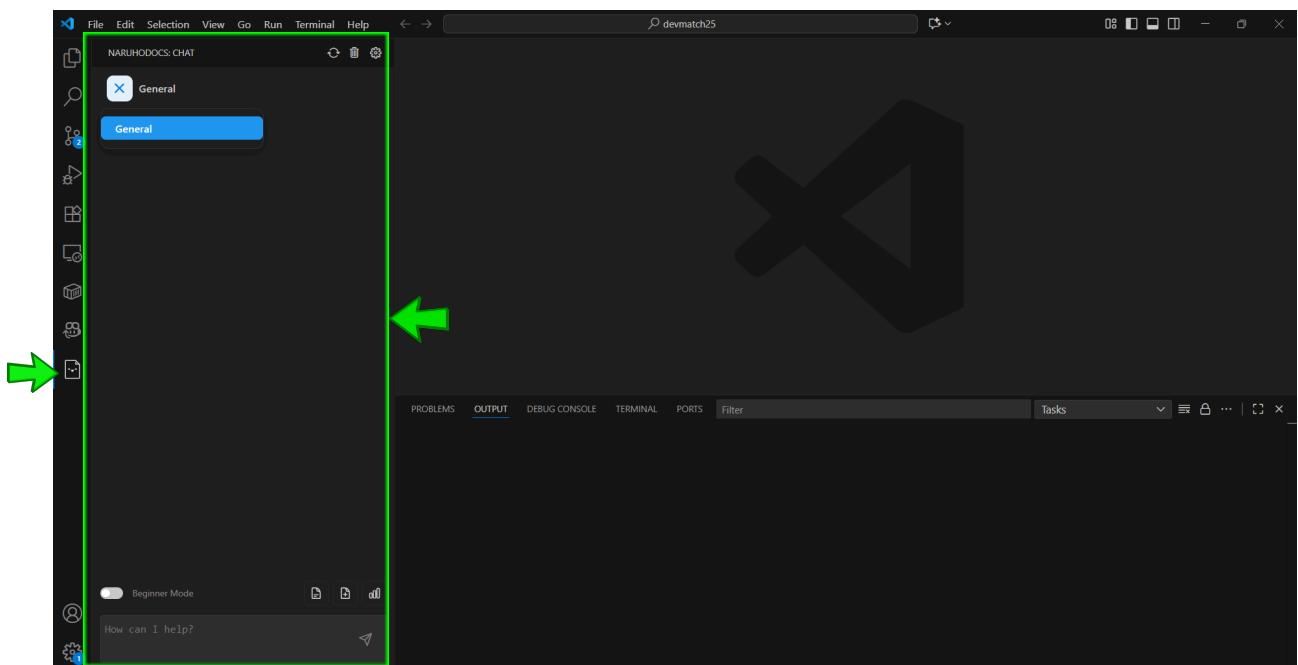
Please refer to the [INSTALLATION_GUIDE.md](#) file for detailed installation steps and setup instructions. It includes information on how to install the extension from VS Code Marketplace.

3.0 GETTING STARTED

3.1 Open NaruhoDocs Sidebar

The sidebar provides easy access to key features like the AI chatbot and documentation tools.

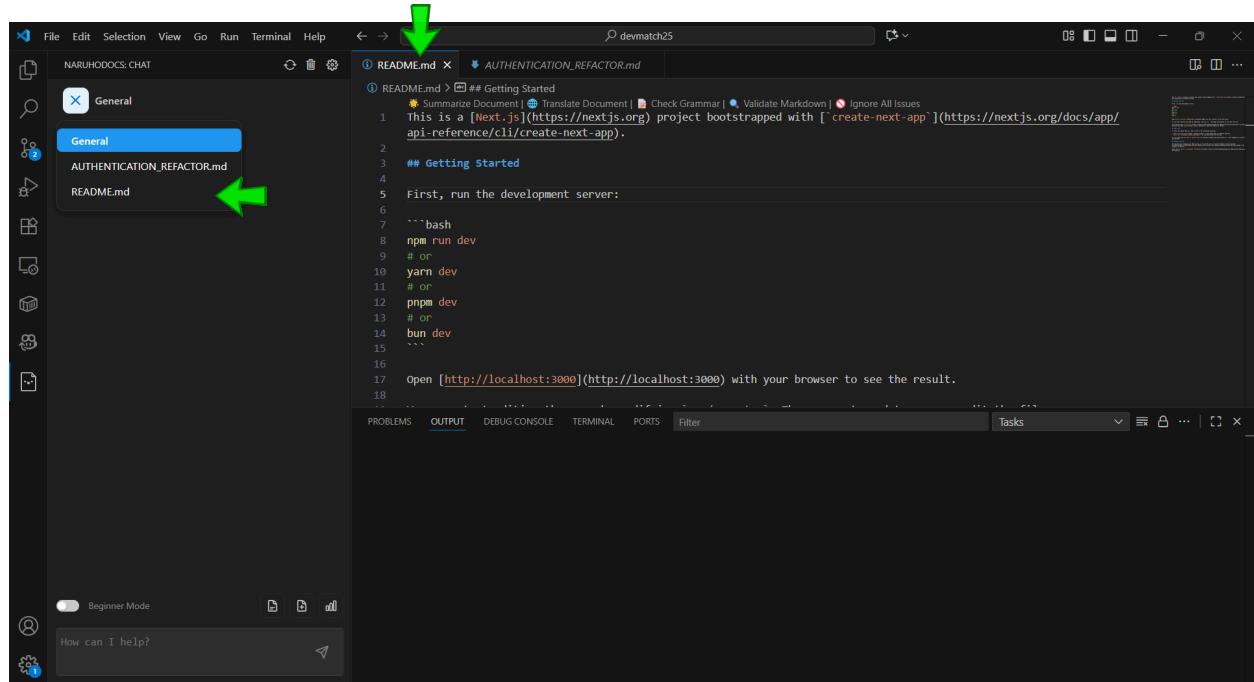
- a. After installation, click the NaruhoDocs icon on the VS Code sidebar.
- b. This will open the NaruhoDocs panel — by default, it only displays the General Tab.



3.2 Add a Document Tab

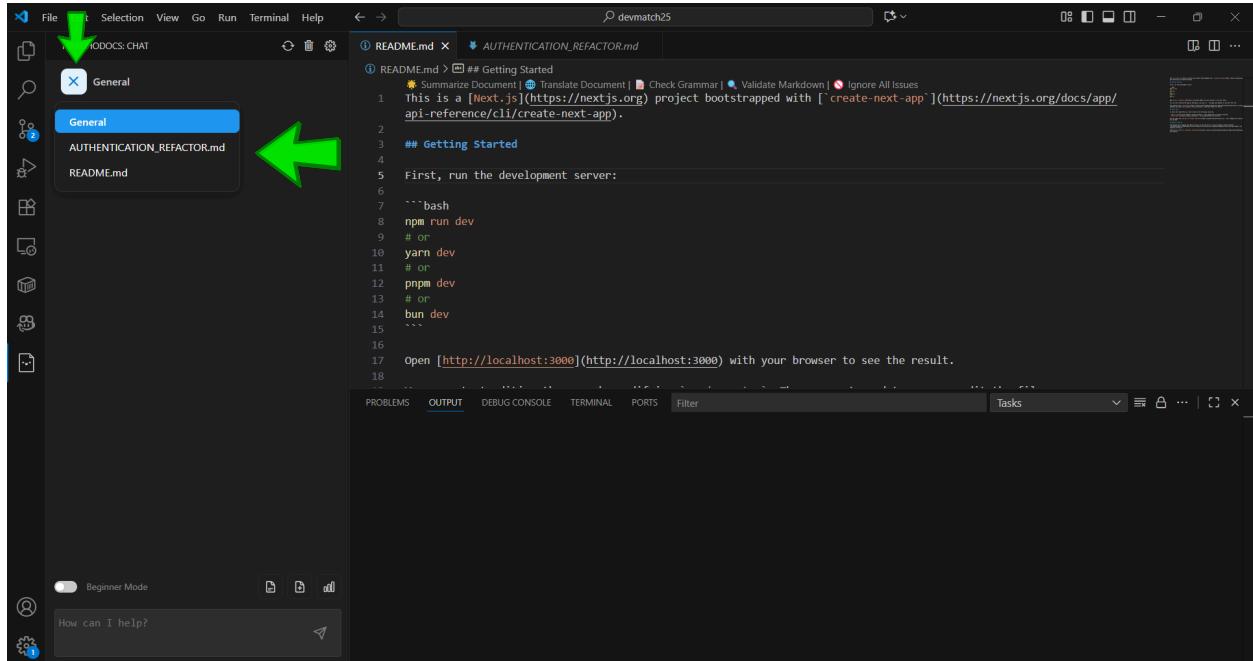
Unlike the General Tab, the Document Tab focuses on one document, allowing the AI chatbot to interact directly with its content.

- a. To create a Document Tab, simply open any Markdown file (`.md`) from your project in the Explorer view.
- b. Once opened, NaruhoDocs will automatically create a new tab for that document, allowing you to use document-specific features.



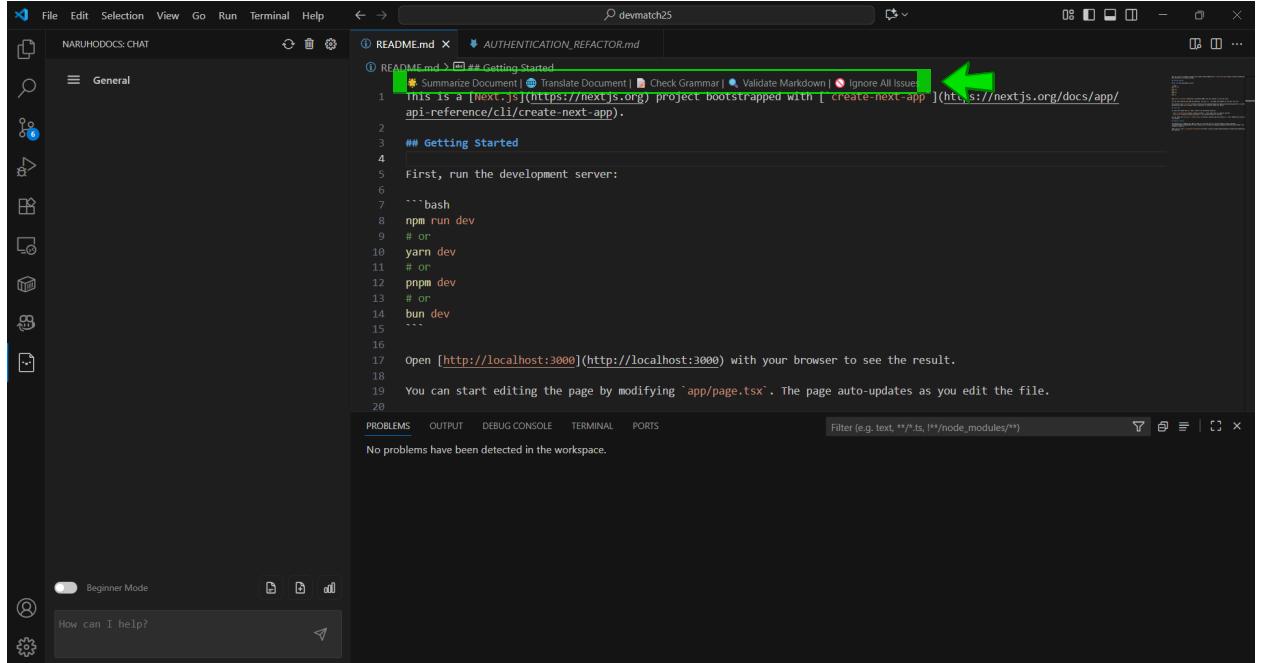
3.3 Switch Tab

- a. To switch between the General Tab and Document Tab, click the hamburger menu and choose the tab you wish from the drop down menu.
- b. To close the hamburger menu, simply click the cross (X) icon.



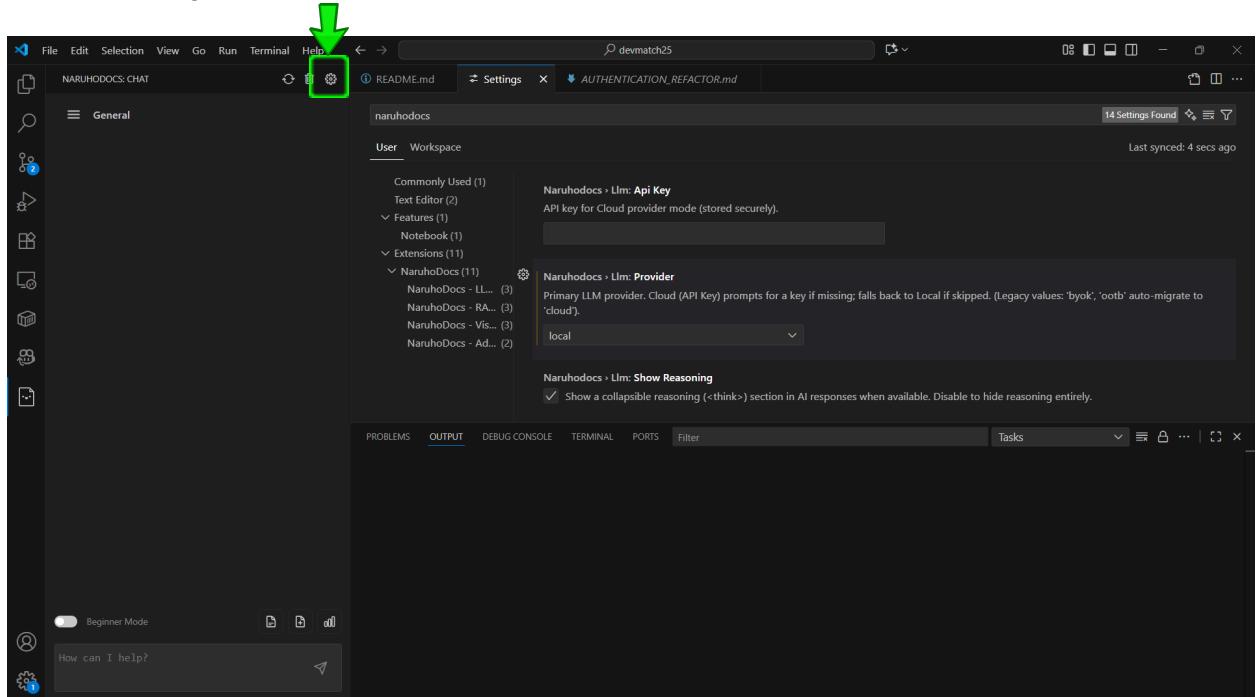
3.4 Document Specific Features

- When you open a Markdown (.md) or text (.txt) file, five action button options will appear at the top of the editor which are Summarize Document, Translate Document, Check Grammar, Validate Markdown, and Ignore All Issues.



4.0 CONFIGURATIONS

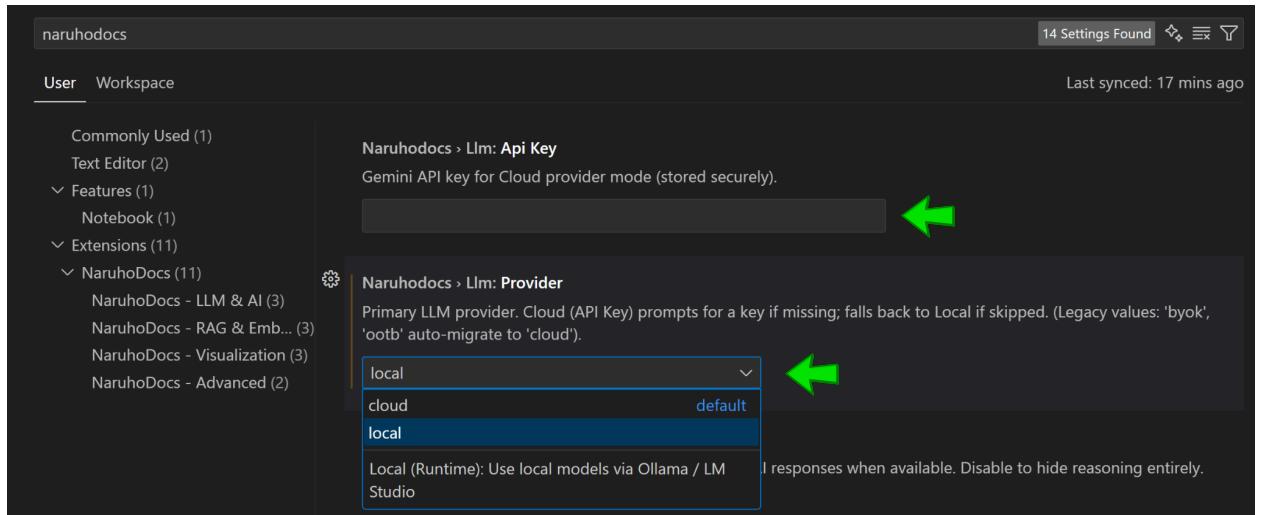
Click on the Settings button in the Naruhodocs sidebar to open the configuration settings. Here, you can customize your AI and embedding preferences according to your setup and usage needs.



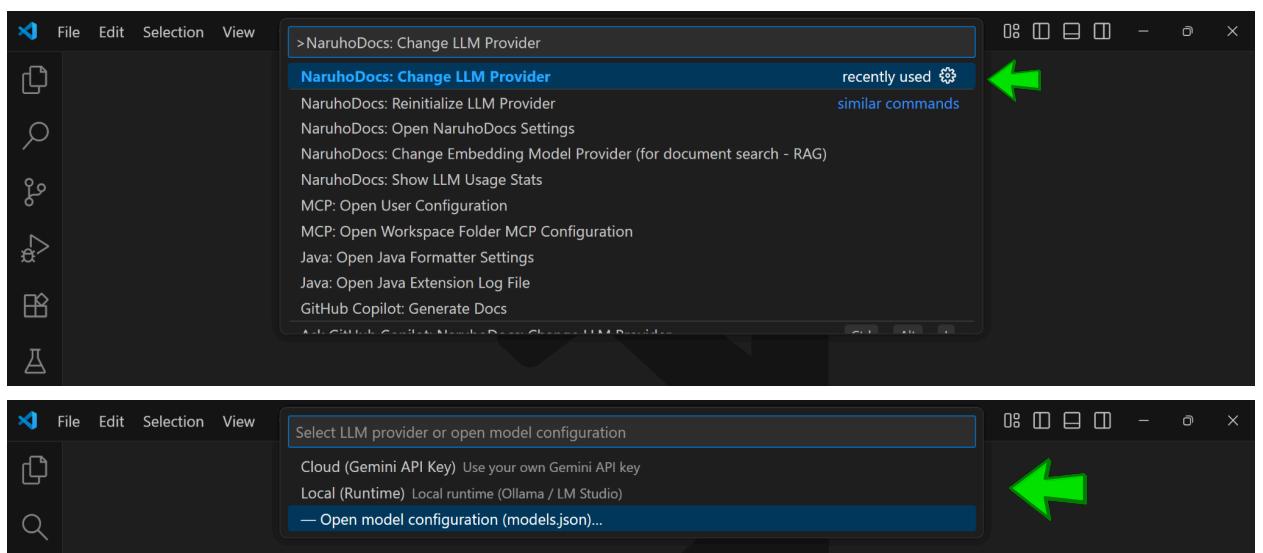
4.1 LLM Provider

This is where you can set up your LLM Provider for all the AI-related tasks (etc. chatbot, documentation tools).

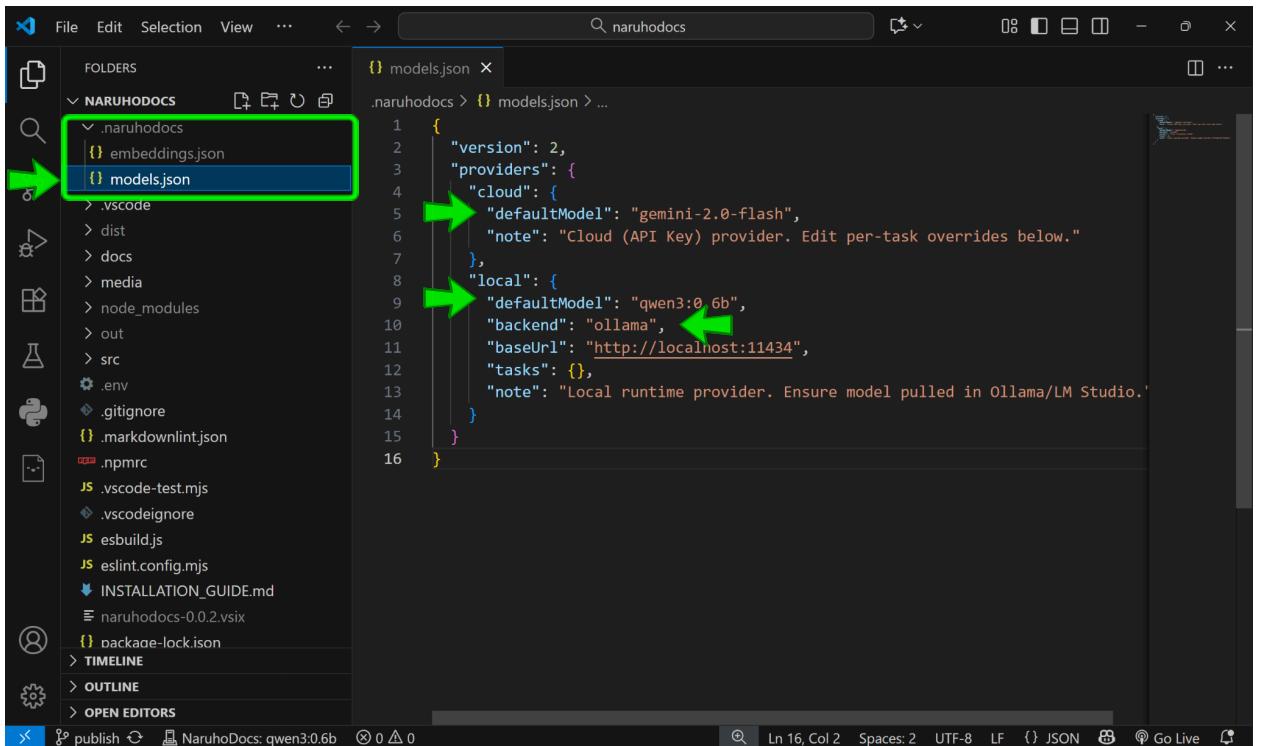
- a. You can select either a Local LLM or a Cloud LLM.
 - i. If you choose **Cloud LLM**, please provide your own **Gemini API key** in the designated field.
 - ii. Get your Gemini API key here: <https://aistudio.google.com/app/api-keys>



- iii. If you are using **Local LLM**, you will need to configure the model settings in **.naruhodocs/models.json** (a **.naruhodocs** folder will be created in your workspace, you might want to add the folder into **.gitignore**).
- iv. You can use Command Palette (Ctrl+Shift+P) and run command "NaruhoDocs: Change LLM Provider" to switch between local and cloud LLMs as well as open the models.json for configuration.



- v. You can modify the default model under the cloud provider to use a different Gemini model.
- vi. For local setups, you can change the backend between Ollama and LM Studio.
- vii. You can also modify the local default model to any model available in your chosen backend. For example: `mistral:7b`, `llama3:8b`.



The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure. A green arrow points to the `models.json` file, which is selected and highlighted in blue.
- Editor:** Displays the `models.json` file content. The content is as follows:

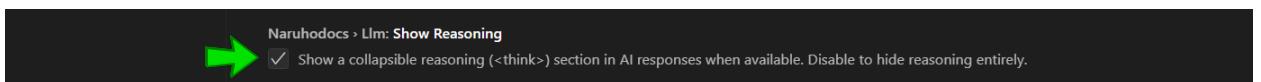

```

1  {
2    "version": 2,
3    "providers": {
4      "cloud": {
5        "defaultModel": "gemini-2.0-flash",
6        "note": "Cloud (API Key) provider. Edit per-task overrides below."
7      },
8      "local": {
9        "defaultModel": "qwen3:0.6b",
10       "backend": "ollama",
11       "baseUrl": "http://localhost:11434",
12       "tasks": {},
13       "note": "Local runtime provider. Ensure model pulled in Ollama/LM Studio."
14     }
15   }
16 }
```

- Bottom Status Bar:** Shows the file path `Naruhodocs > models.json`, line number `Ln 16, Col 2`, and character position `Spaces: 2`.
- Bottom Right:** Includes icons for Publish, Go Live, and other settings.

Configure model settings in `models.json`

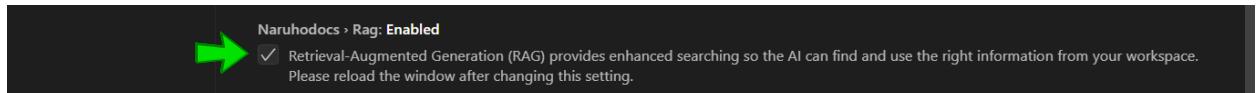
- b. You may also toggle the option to display or hide the reasoning behind the AI's responses if you are using the "Thinking" LLM model.



4.2 Embedding Modes

This is where you can set up your embedding model providers. An embedding model is used for **RAG** (Retrieval-Augmented Generation) to convert text into numerical representations, helping the AI retrieve relevant information more accurately.

- You can **enable or disable the RAG** feature in the settings depending on whether you want the AI to use a more advanced searching method throughout your workspace during AI responses.



*If you do not want RAG feature, you can skip setting up an embedding model.

- To set up the embedding model, you can choose between Local Embedding and Cloud Embedding.
 - If you select **Cloud Embedding**, please insert your **Hugging Face API key**.

naruhodocs

User Workspace

Last synced: 0 secs ago

Naruhodocs > Lim: Show Reasoning
✓ Show a collapsible reasoning (<think>) section in AI responses when available. Disable to hide reasoning entirely.

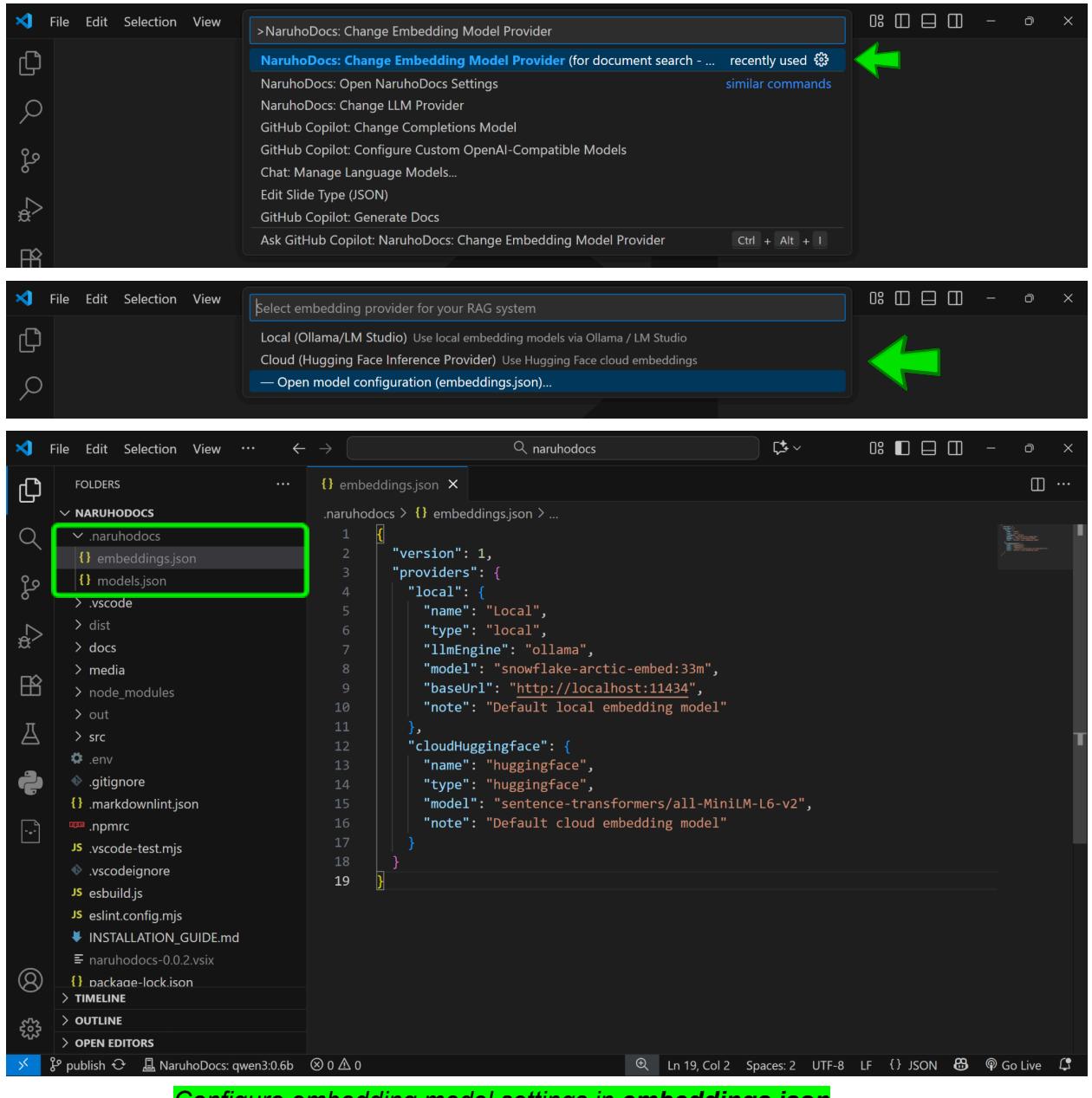
Naruhodocs > Embedding: Provider
Embedding model provider to use for document search (RAG). If you do not want a RAG functionality, you can disable RAG in the settings.
local
local default
cloudHuggingface
Local (Ollama/LM Studio): Use local embedding models via Ollama / LM Studio

Naruhodocs > Features (1)
Notebook (1)
Extensions (11)
NaruDocs (11)
NaruDocs - LL... (3)
NaruDocs - RA... (3)
NaruDocs - Vis... (3)
NaruDocs - Ad... (2)

Naruhodocs > Embedding: Provider
Embedding model provider to use for document search (RAG). If you do not want a RAG functionality, you can disable RAG in the settings.
cloudHuggingface

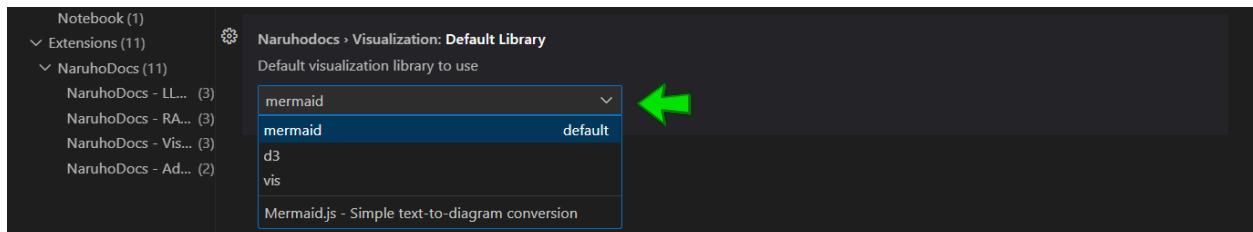
Naruhodocs > Huggingface: Api Key
HuggingFace Inference API key for cloud embeddings.

- ii. If you are using **Local Embedding**, you will need to configure the model settings in **.naruhodocs/embeddings.json** (a **.naruhodocs** folder will be created in your workspace, you might want to add the folder into **.gitignore**).
- iii. You can use Command Palette (Ctrl+Shift+P) and run command “NaruhoDocs: Change Embedding Model Provider” to switch between local and cloud embedding models as well as open the embeddings.json for configuration.



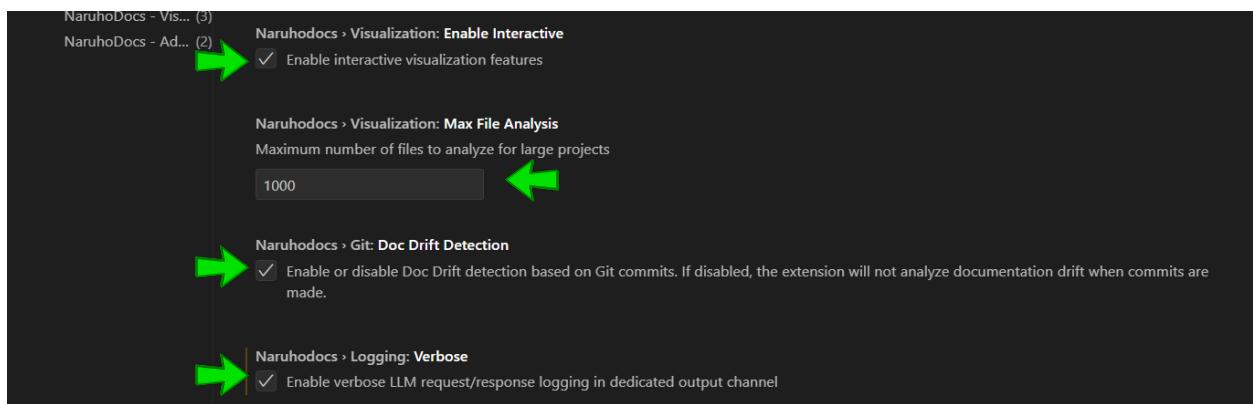
4.3 Visualization Library

- a. You can select your preferred visualization library from Mermaid, D3 or Vis.



4.4 Other Settings

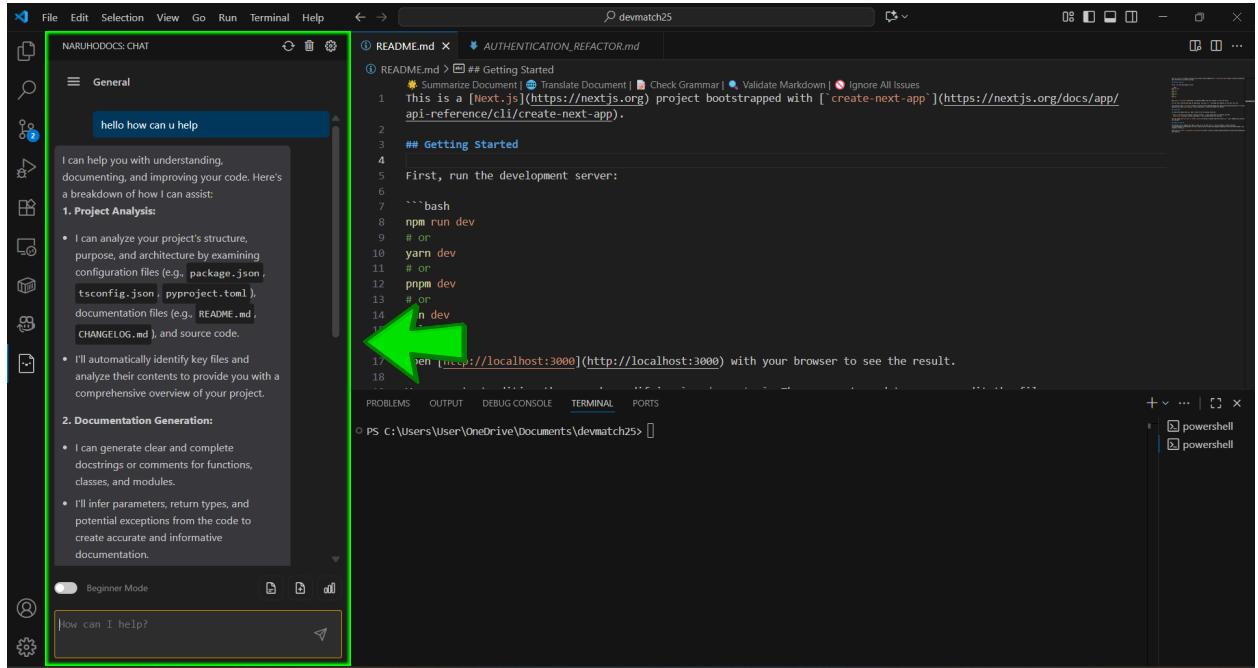
- a. Enable Interactive
 - i. Toggle this option to enable or disable interactive visualization features within NaruhoDocs.
- b. Max File Analysis
 - i. Defines the maximum number of files NaruhoDocs will analyze in large projects.
- c. Doc Drift Detection
 - i. Enables or disables documentation drift detection based on Git commits. This feature compares your current documentation with the latest code changes to identify when the docs are outdated.
- d. Verbose Logging
 - i. When enabled, NaruhoDocs will record detailed LLM request and response logs in a dedicated output channel.



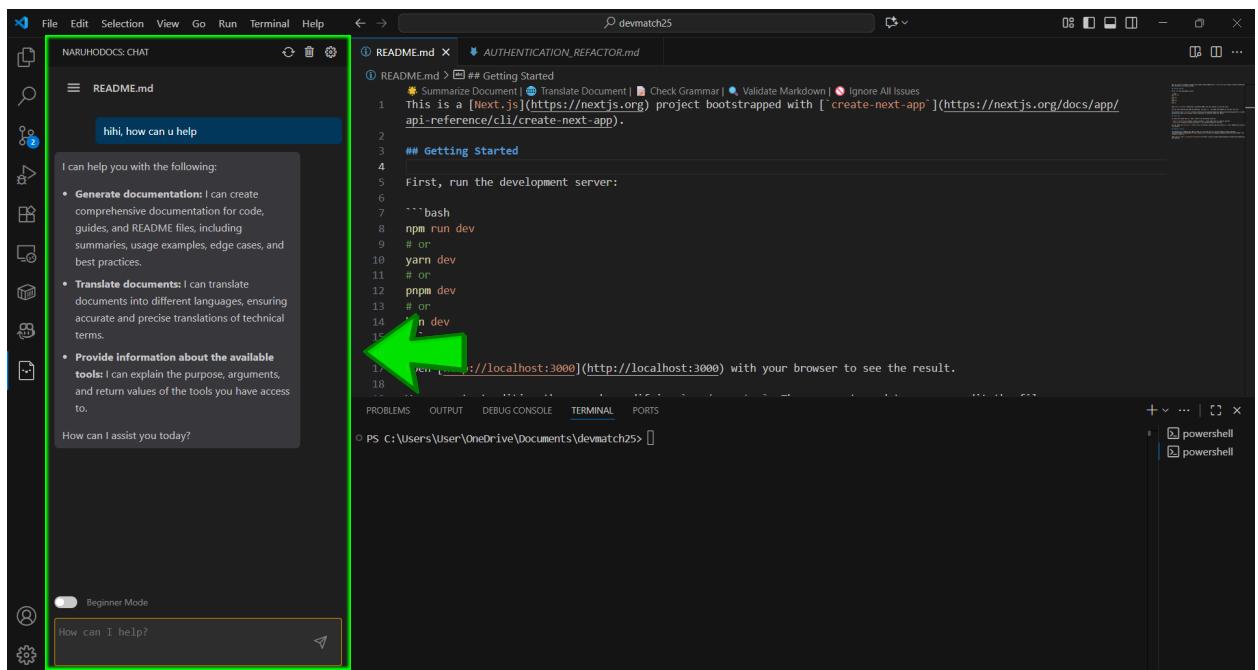
5.0 FEATURES

5.1 AI Chatbot

- You can ask project-level questions through the General Chatbot, which provides insights based on the entire project.



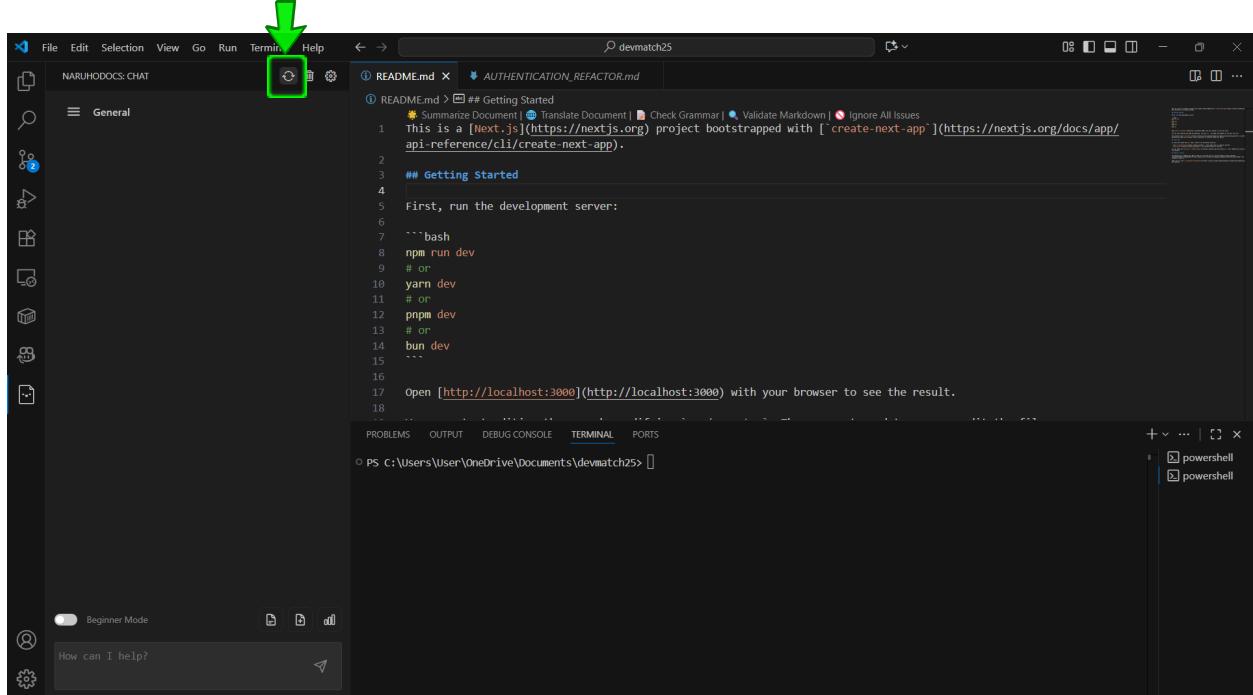
- For document-specific inquiries, you can use the Document Chatbot to receive responses related only to the current file.



5.2 Rebuild RAG Vector Database

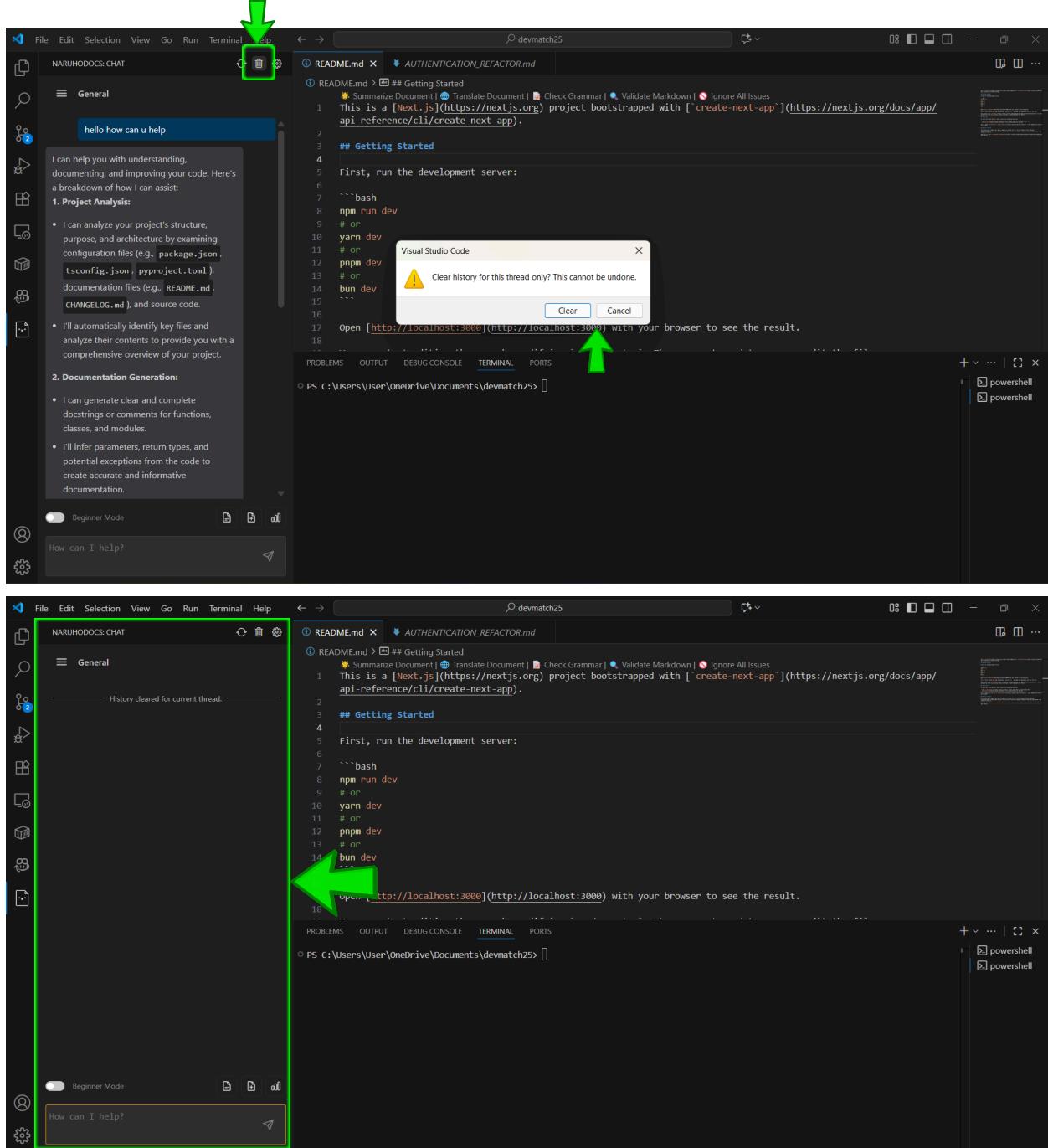
- Click on refresh button to rebuild RAG Vector Database

You may rebuild the RAG Vector Database when your code has updated and you want the LLM to respond based on the latest version of the workspace



5.3 Clear Current Thread History

- To clear the chat history of the current tab, click the bin icon, then select Clear.
- For instance, if you are in the General Tab, this action will delete only the General Tab's chat history, leaving other tabs unaffected.



Chat history cleared

5.4 Beginner and Developer Mode

By default, NaruhodoDocs opens in Developer Mode.

- If you switch to Beginner Mode, the chatbot will provide beginner-friendly explanations that avoid technical jargon.

The screenshot shows the NaruhodoDocs interface in Beginner Mode. On the left, there's a sidebar with icons for file operations like Open, Save, and Print. The main area has a title "NARUHODOCS: CHAT" and a search bar. Below the search bar is a dropdown menu set to "General". A large text box contains an AI-generated response about machine learning, which is clearly explained with bullet points and simple language. At the bottom, a message input field says "How can I help?" and a "Beginner Mode" button is selected. On the right, there's a code editor window titled "AUTHENTICATION_REFATOR.md" showing some code snippets and a terminal window at the bottom.

AI response is easy to understand

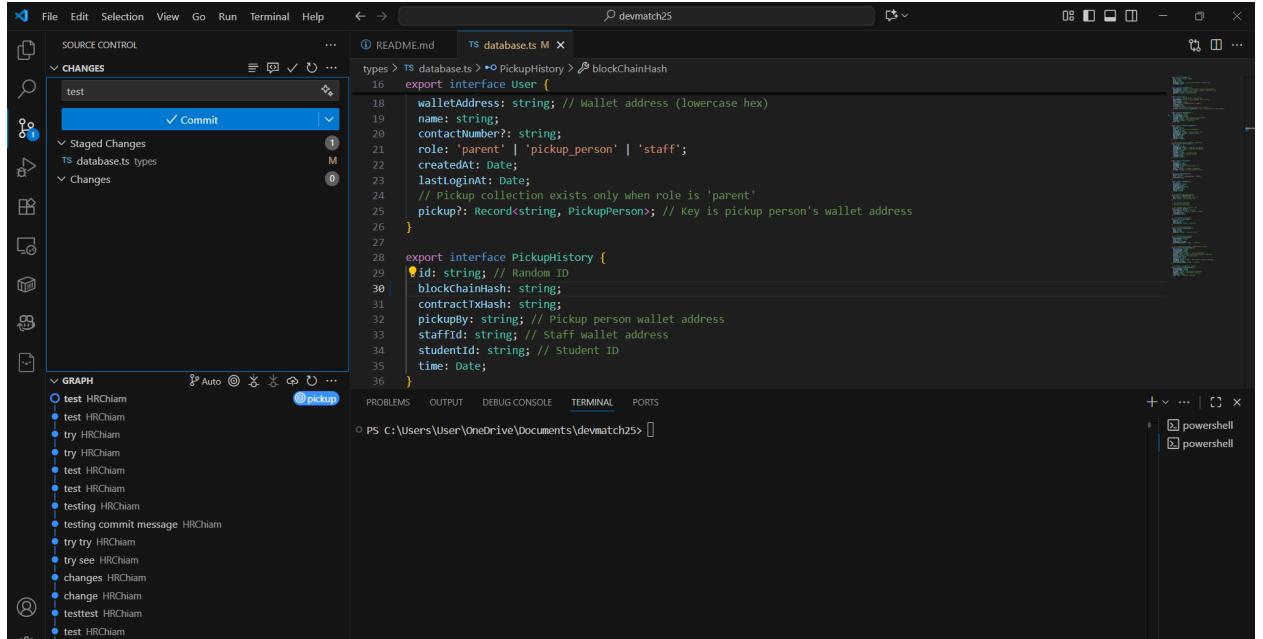
- In Developer Mode, responses are detailed and technical, and assume the user has programming experience.

The screenshot shows the NaruhodoDocs interface in Developer Mode. The layout is identical to the Beginner Mode screenshot, but the AI response is much more detailed and technical. It discusses the "Data Collection and Preparation" step in machine learning, mentioning "Training set", "Validation set", and "Test set". The response is filled with technical terms and concepts, assuming knowledge of these topics. A green arrow points from the developer mode interface towards the detailed AI response.

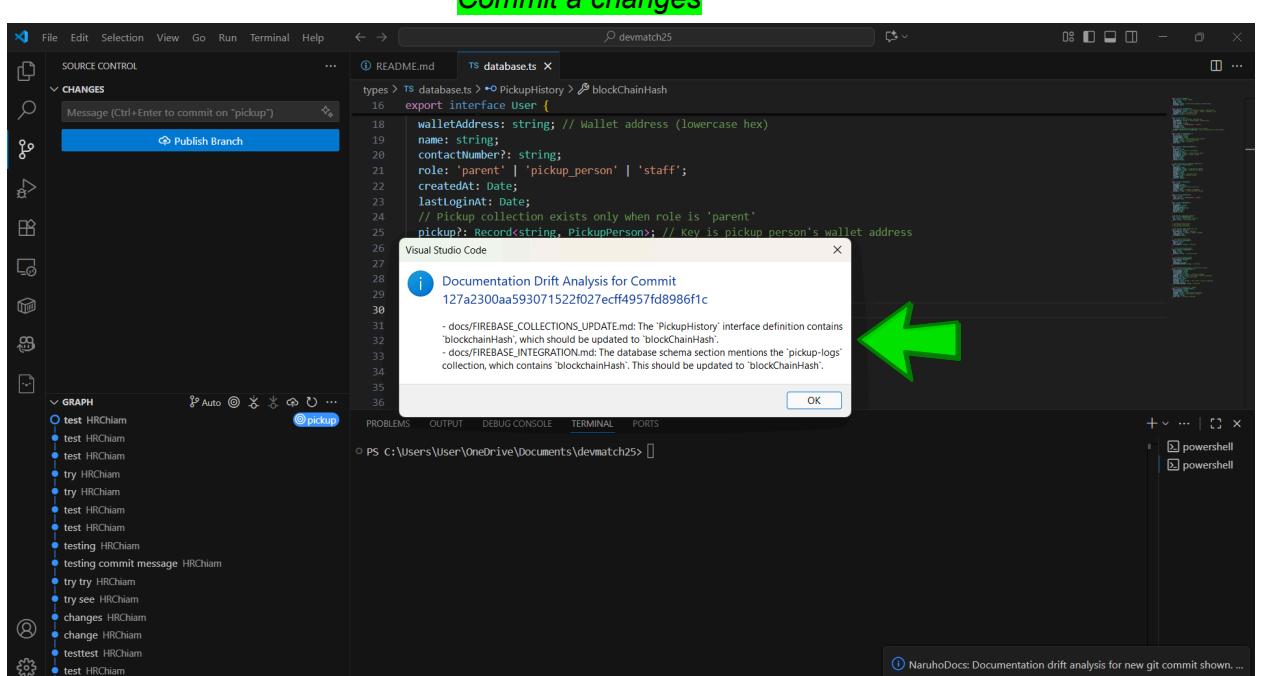
AI response is technical

5.5 Documentation Drift Detection

- a. For each commit, NaruhoDocs automatically detects code changes and provides AI-generated suggestions on other files that may need to be updated accordingly.



The screenshot shows the Visual Studio Code interface with the Source Control panel open. The Changes section displays a commit message: "test". Below the commit message, there is a list of staged changes, including various test files and a "pickup" file. The main editor area shows a TypeScript file named "database.ts" with code definitions for "User" and "PickupHistory" interfaces. A green box highlights the commit message, and the text "Commit a changes" is overlaid at the bottom.

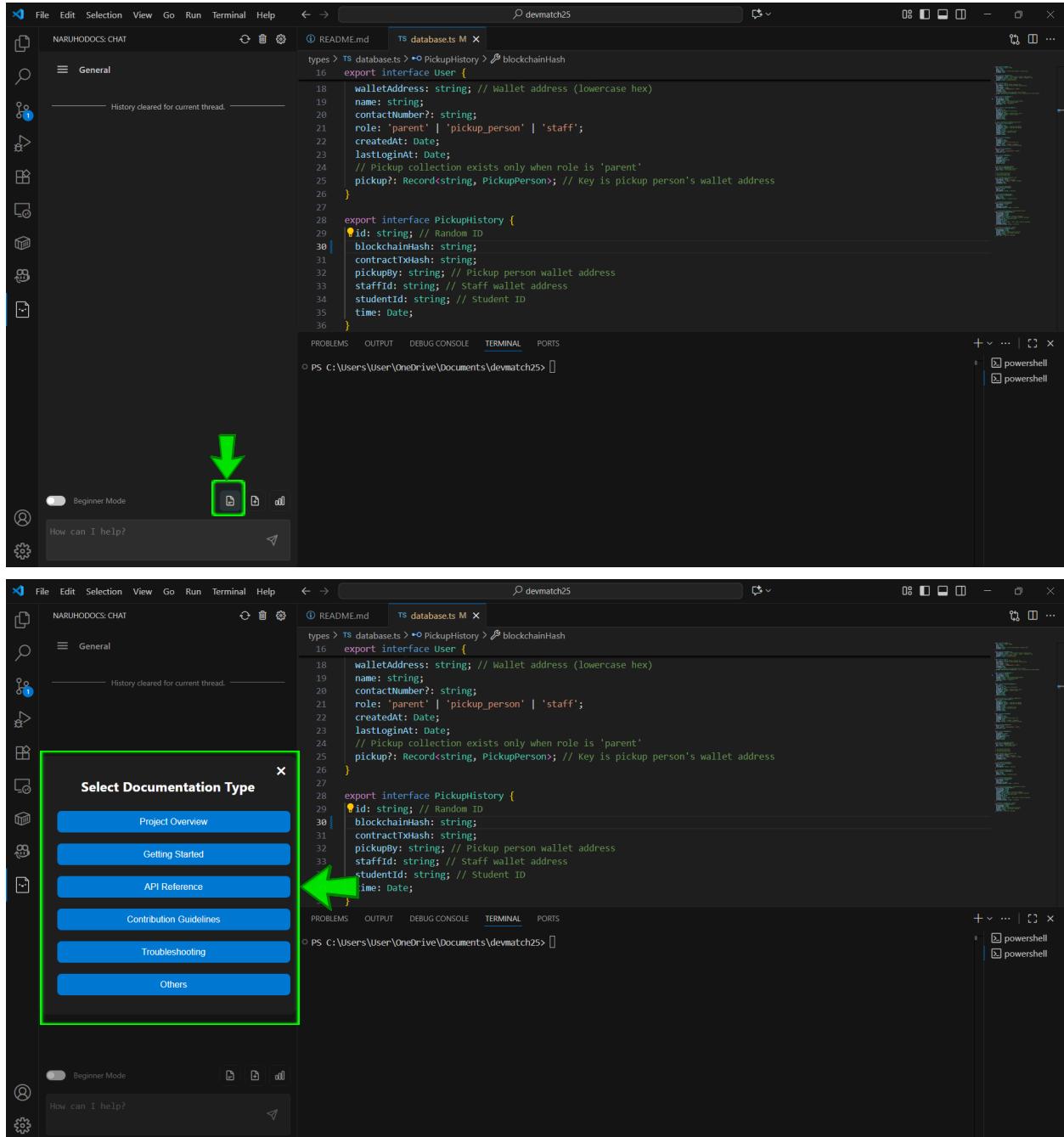


The screenshot shows the Visual Studio Code interface with the Source Control panel open. A dialog box titled "Documentation Drift Analysis for Commit" is displayed, containing a message about drift between "database.ts" and "docs/FIREBASE_COLLECTIONS_UPDATE.md". A large green arrow points from the top screenshot to this dialog box. The text "AI suggest file that need to be updated" is overlaid at the bottom.

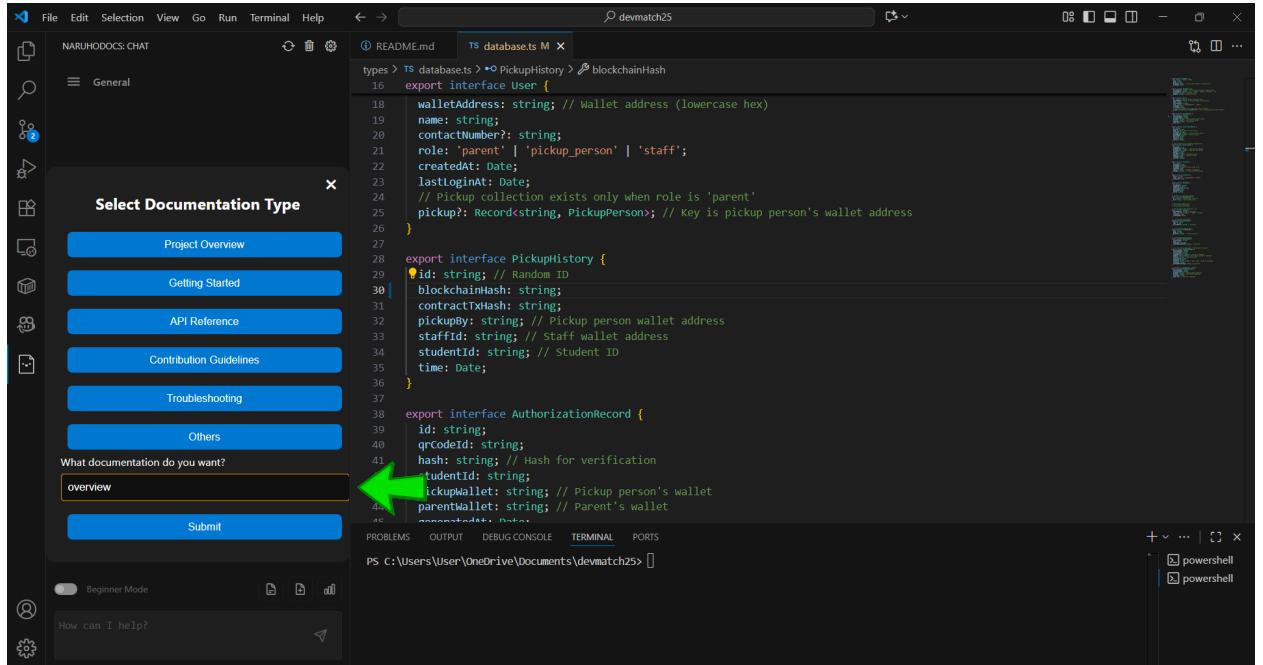
5.6 General Tab Specific Features

5.6.1 Generate Documentation

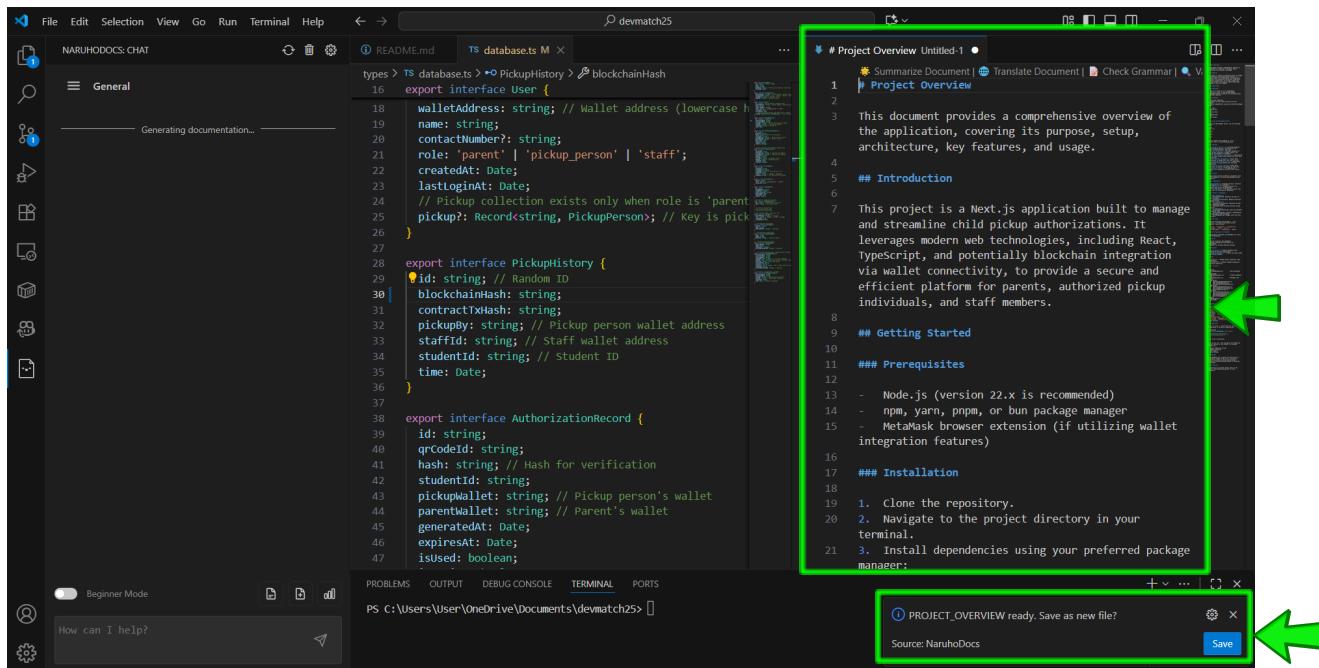
- Click the document-with-lines icon to open a list of documentation suggestions generated by the AI. These are recommended documents that NaruhodoDocs suggests adding to your project.



- b. You can select a document from the suggested list, or choose “Others” to create a custom document by entering your own title.



- c. Once selected, the new document will appear on the right side of the editor. A notification will also appear in the bottom-right corner, allowing you to directly save the file.



The screenshot shows the VS Code interface with the Project Overview file open. The status bar at the bottom right indicates "Saved document to c:\Users\User\OneDrive\Documents\devmatch25..." with a green arrow pointing to it.

```

1 # Project Overview Untitled-1
2
3 ## Project Title
4
5 KidGuard
6
7 ## Description
8
9 KidGuard is a Next.js application designed to streamline and secure the process of child pickup authorization. It provides a role-based dashboard system for parents, authorized pickup individuals, and staff, leveraging technologies like React, TypeScript, QR codes, and potentially blockchain via wallet integration to ensure efficiency and security.
10
11 ## Goals
12
13 * Provide a secure and efficient platform for managing child pickup authorizations.
14 * Offer role-specific dashboards with tailored functionalities for parents, pickup personnel, and staff.
15 * Utilize modern web technologies to ensure maintainability, scalability, and a responsive user experience.
16
17 ## Key Features
18
19

```

The screenshot shows the VS Code interface with the Project Overview file open in the workspace. The status bar at the bottom right indicates "File is saved in the workspace" with a green arrow pointing to it.

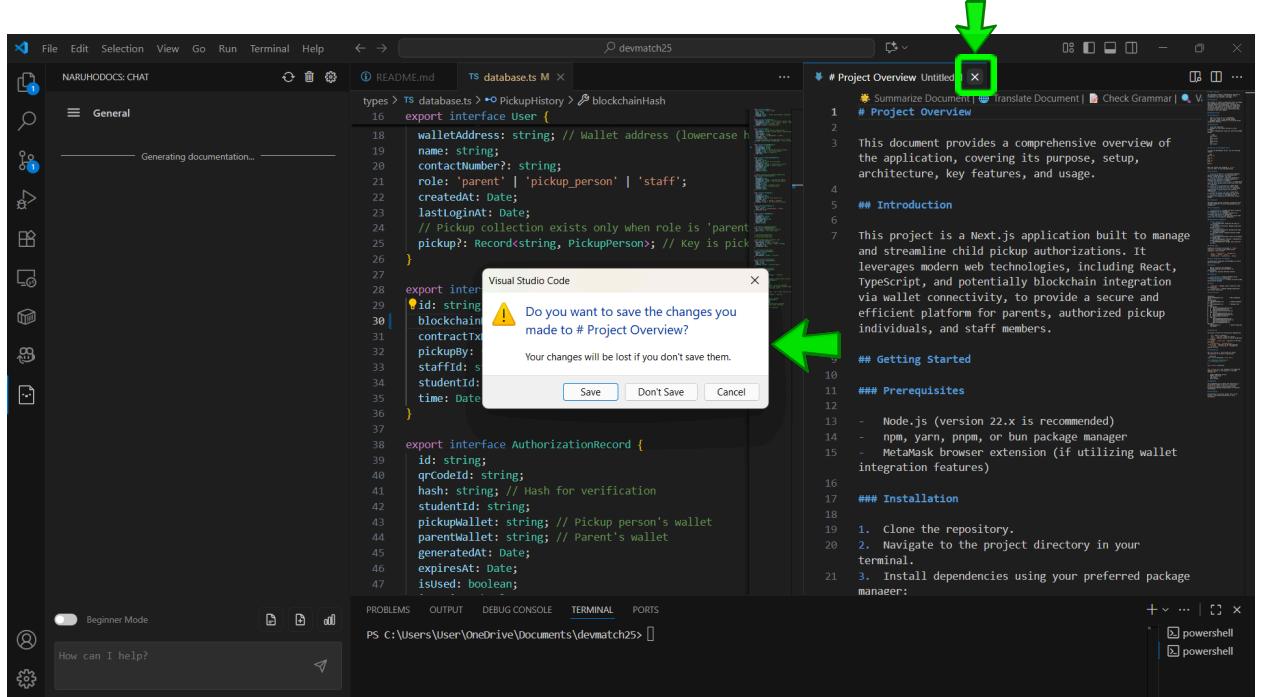
```

1 # Project Overview.md > # Project Overview
2
3 This document provides a comprehensive overview of the application, covering its purpose, setup, architecture, key features, and usage.
4
5 ## Introduction
6
7 This project is a Next.js application built to manage and streamline child pickup authorizations. It leverages modern web technologies, including React, TypeScript, and potentially blockchain integration via wallet connectivity, to provide a secure and efficient platform for parents, authorized pickup individuals, and staff members.
8
9 ## Getting Started
10
11 ## Prerequisites
12
13 - Node.js (version 22.x is recommended)
14 - npm, yarn, pnpm, or bun package manager
15 - Metablock browser extension (if utilizing wallet integration features)
16
17 ## Installation
18
19 1. Clone the repository.
20 2. Navigate to the project directory in your terminal.
21 3. Install dependencies using your preferred package manager:
22
23   ````bash
24   npm install
25   # or
26   yarn install
27   # or
28   npx intellil
29

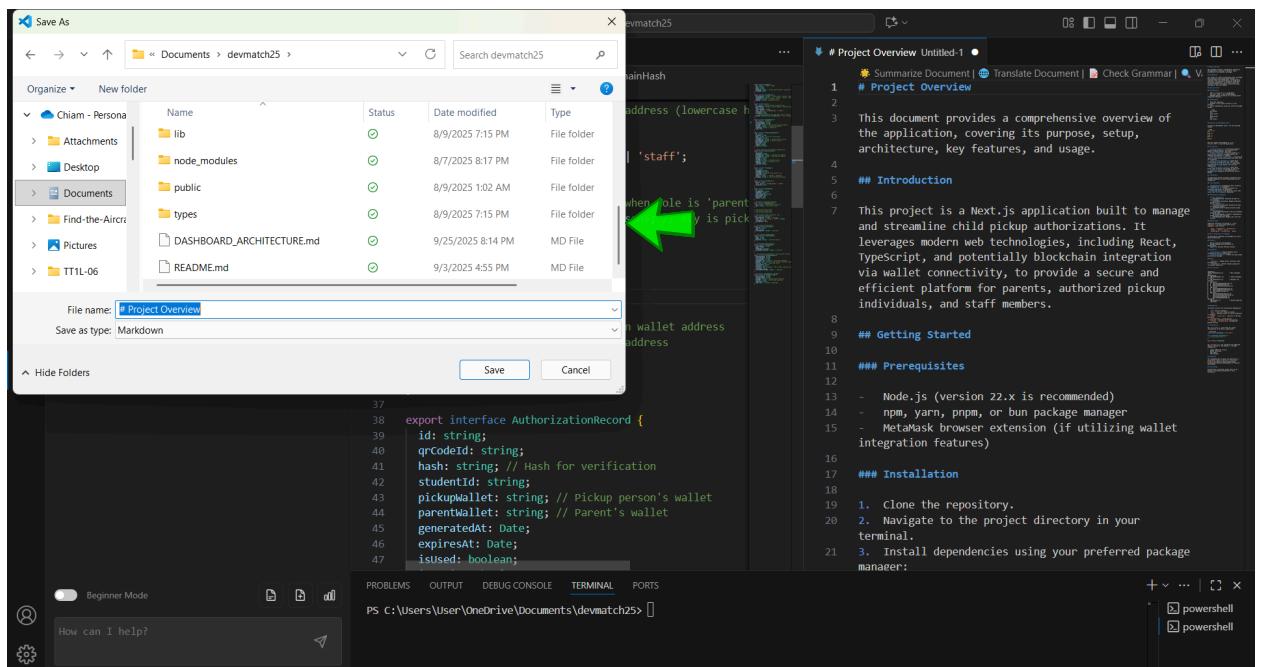
```

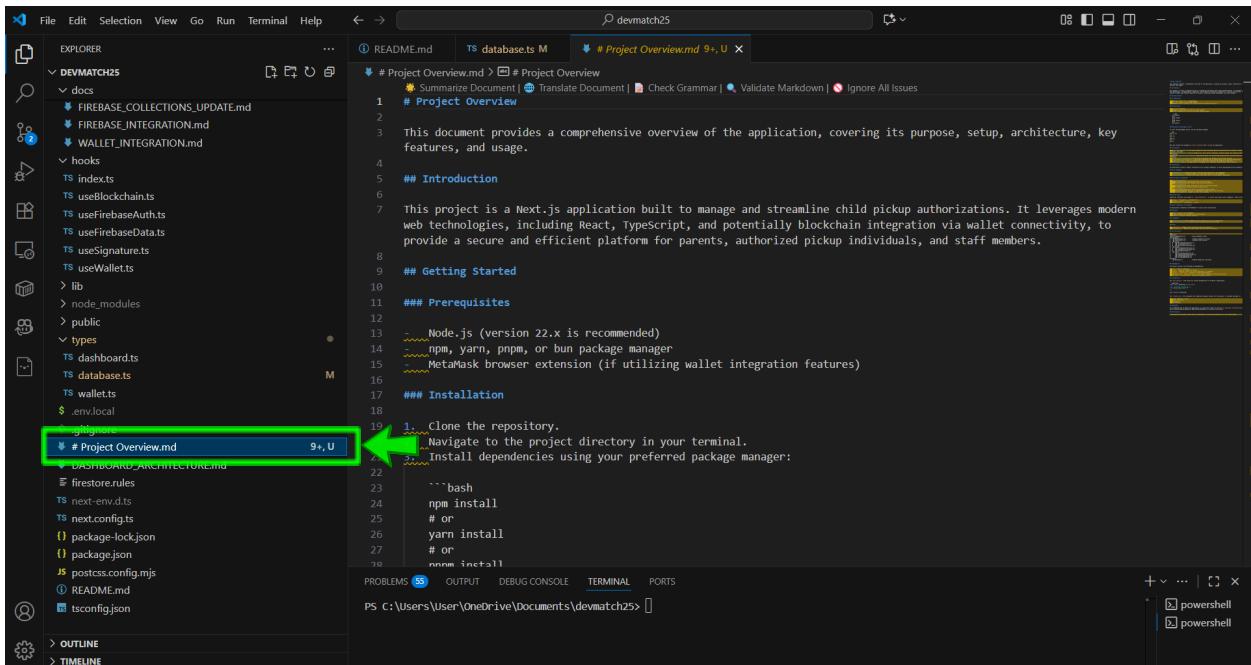
File is saved in the workspace

d. You can also click the icon beside the file name. A menu will appear with the options Save, Don't Save, or Cancel.



e. After selecting Save, choose your preferred save location and rename the file if needed.





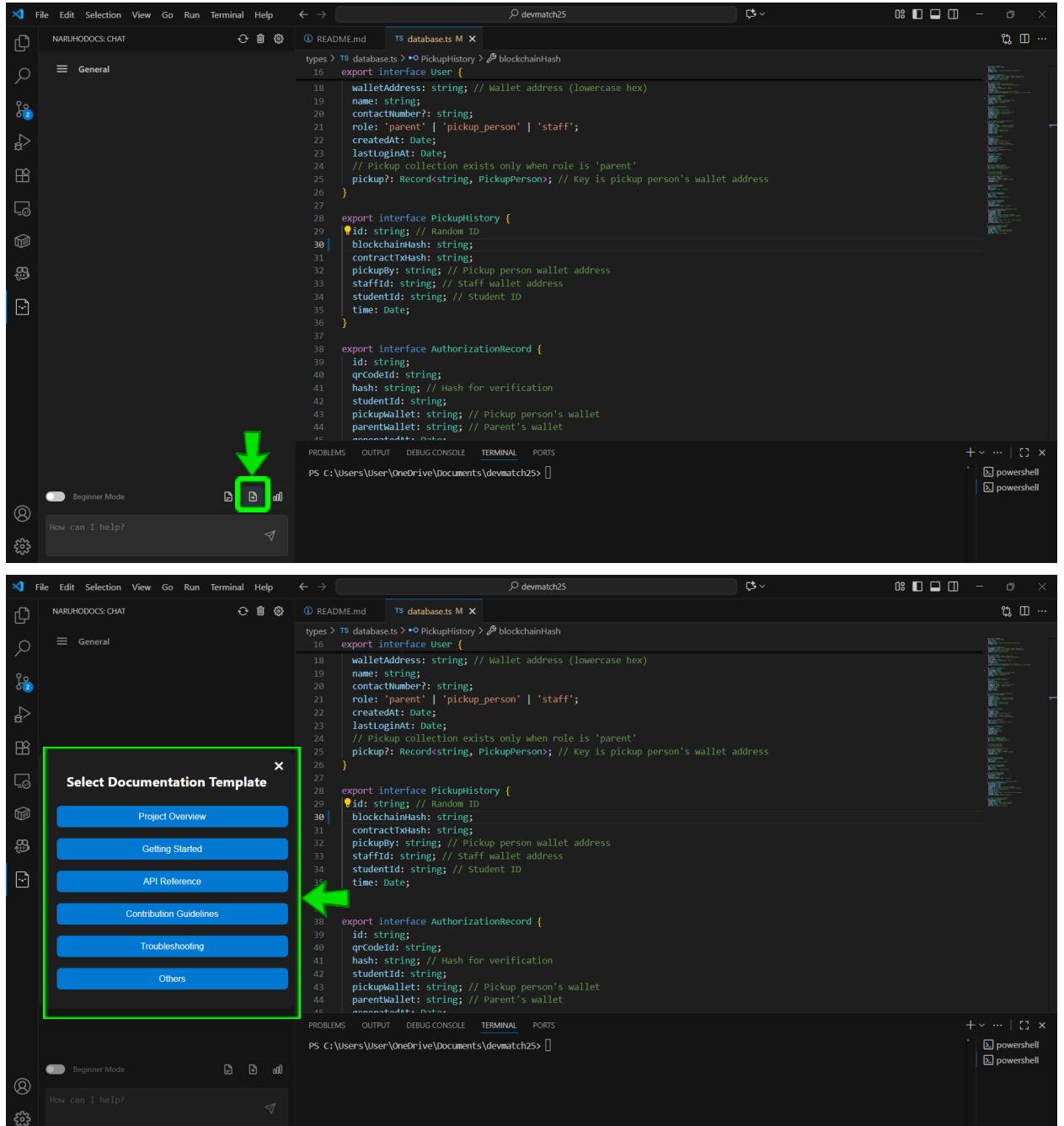
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under 'DEVMATCH25'. Files include README.md, database.ts, Project Overview.md (which is currently selected), FIREBASE_COLLECTIONS_UPDATE.md, FIREBASE_INTEGRATION.md, WALLET_INTEGRATION.md, hooks/index.ts, useBlockchain.ts, useFirebaseAuth.ts, useFirebaseData.ts, useSignature.ts, useWallet.ts, lib, node_modules, public/types/dashboard.ts, database.ts, wallets/.env.local, and .gitignore.
- Editor (Center):** Displays the content of 'Project Overview.md'. The content includes sections like '# Project Overview', '## Introduction', '## Getting Started', '### Prerequisites', and '### Installation'. It also contains code snippets for cloning the repository and installing dependencies.
- Status Bar (Bottom):** Shows the path 'C:\Users\User\OneDrive\Documents\devmatch25>', the terminal tab, and the message 'File is saved in the workspace'.

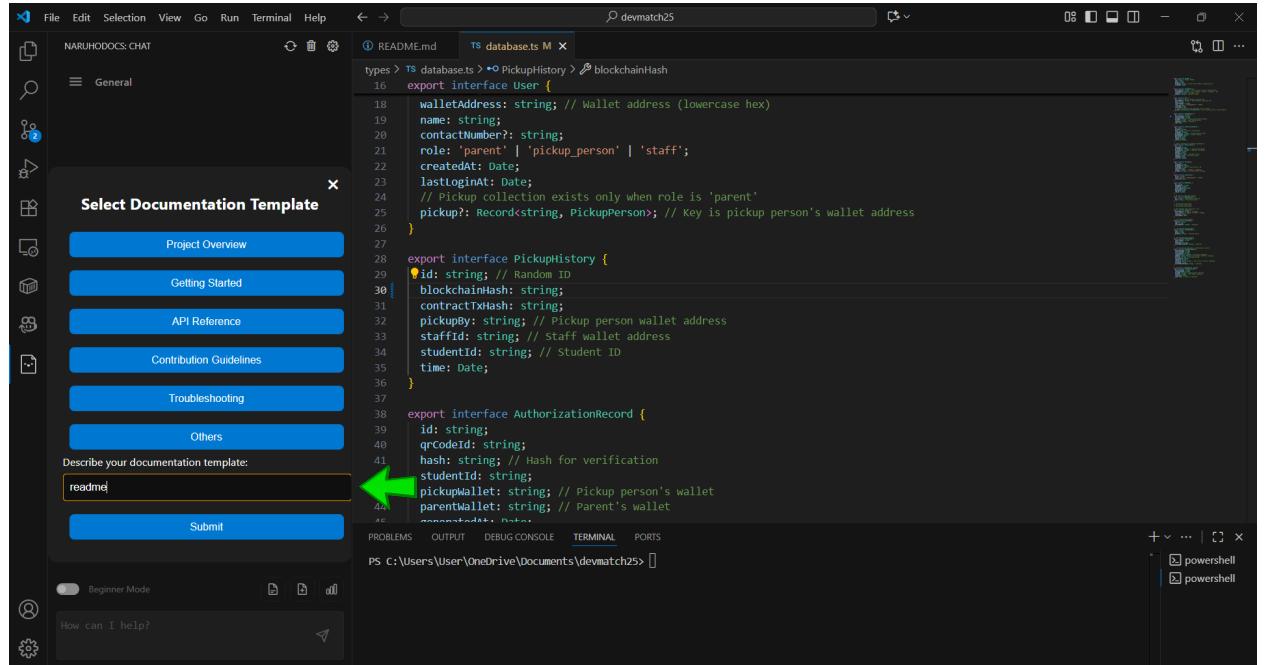
File is saved in the workspace

5.6.2 Suggest Documentation Template

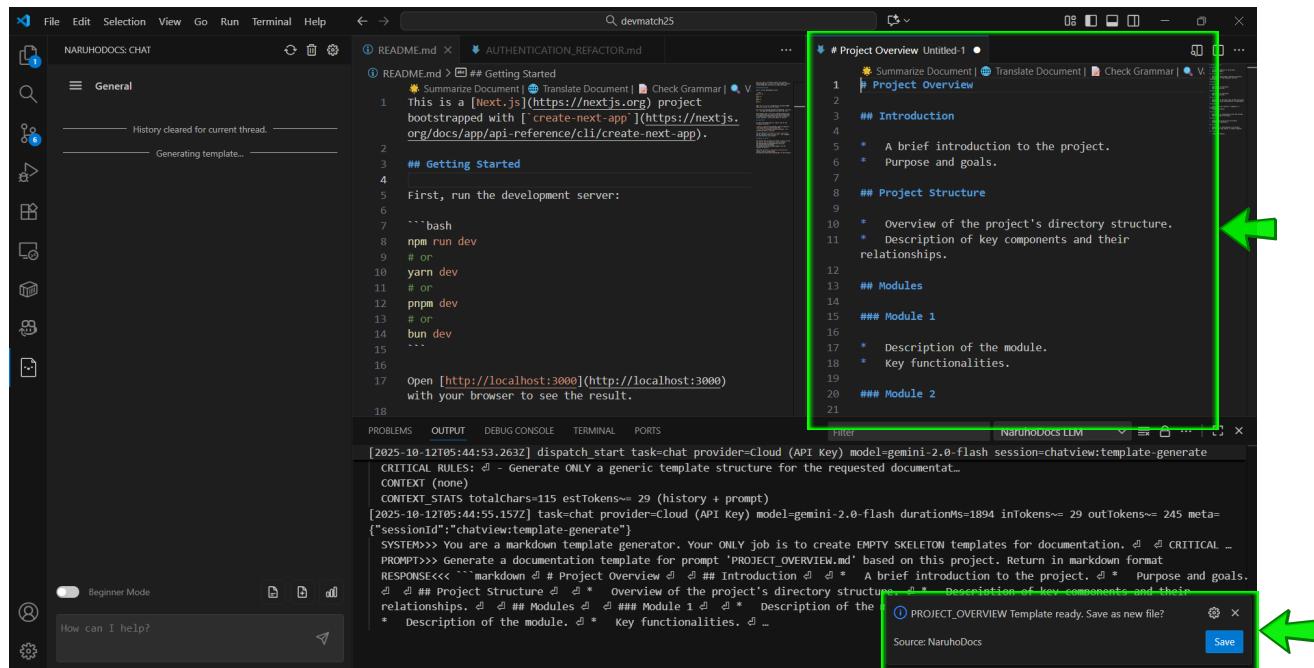
- Click the document-with-plus icon to open a list of documentation suggestions generated by the AI. These are recommended documents that NaruhoDocs suggests adding to your project.



- b. You can select a template from the suggested list, or choose “Others” to create a custom document template by entering your own title.



- c. Once selected, the new template will appear on the right side of the editor. A notification will also appear in the bottom-right corner, allowing you to save the file directly.



The screenshot shows a code editor interface with two main panes. The left pane displays a terminal window with command history and a code editor showing a README.md file. The right pane shows a detailed project overview template. A green box highlights the status bar at the bottom right of the editor, which displays the message "Saved template to c:\Users\User\OneDrive\Documents\devmatch25\docs\PROJECT_OVERVIEW_TEMPLATE.md".

```

# Project Overview Untitled-1
# Project Overview
## Introduction
* A brief introduction to the project.
* Purpose and goals.

## Project Structure
* Overview of the project's directory structure.
* Description of key components and their relationships.

## Modules
### Module 1
* Description of the module.
* Key functionalities.

### Module 2
* Description of the module.
* Key functionalities.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[2025-10-12T05:44:53.263Z] dispatch_start task=chat provider=Cloud (API Key) model=gemini-2.0-flash session=chatview:template-generate

CRITICAL RULES: ⚡ - Generate ONLY a generic template structure for the requested documentat...

CONTEXT (none)

CONTEXT_STATS totalChars=115 estTokens= 29 (history + prompt)

[2025-10-12T05:44:55.157Z] task=chat provider=Cloud (API Key) model=gemini-2.0-flash durationMs=1894 inTokens= 29 outTokens= 245 meta={"sessionId":"chatview:template-generate"}

SYSTEM>>> You are a markdown template generator. Your ONLY job is to create EMPTY SKELETON templates for documentation. ⚡ ⚡ CRITICAL - PROMPT>>> Generate a documentation template for prompt 'PROJECT_OVERVIEW.md' based on this project. Return in markdown format

RESPONSE<<< ````markdown ⚡ # Project Overview ⚡ ⚡ # Introduction ⚡ ⚡ * A brief introduction to the project. ⚡ * Purpose and goals.

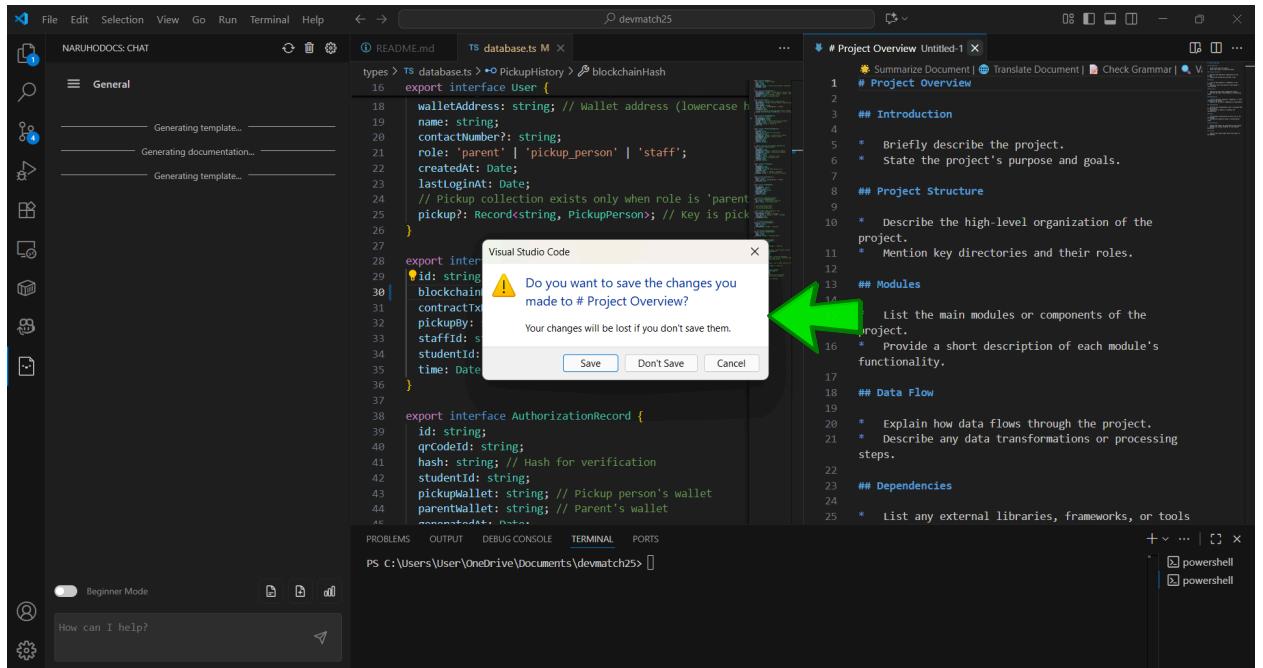
⚡ ⚡ # Project Structure ⚡ ⚡ * Overview of the project's directory structure. ⚡ * Description of key components and their relationships. ⚡ ⚡ # Modules ⚡ ⚡ # Module 1 ⚡ ⚡ * Description of the module. ⚡ * Key functionalities. ⚡ ⚡ # Module 2 ⚡ ⚡ * Description of the module. ⚡ * Key functionalities. ⚡ ...

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

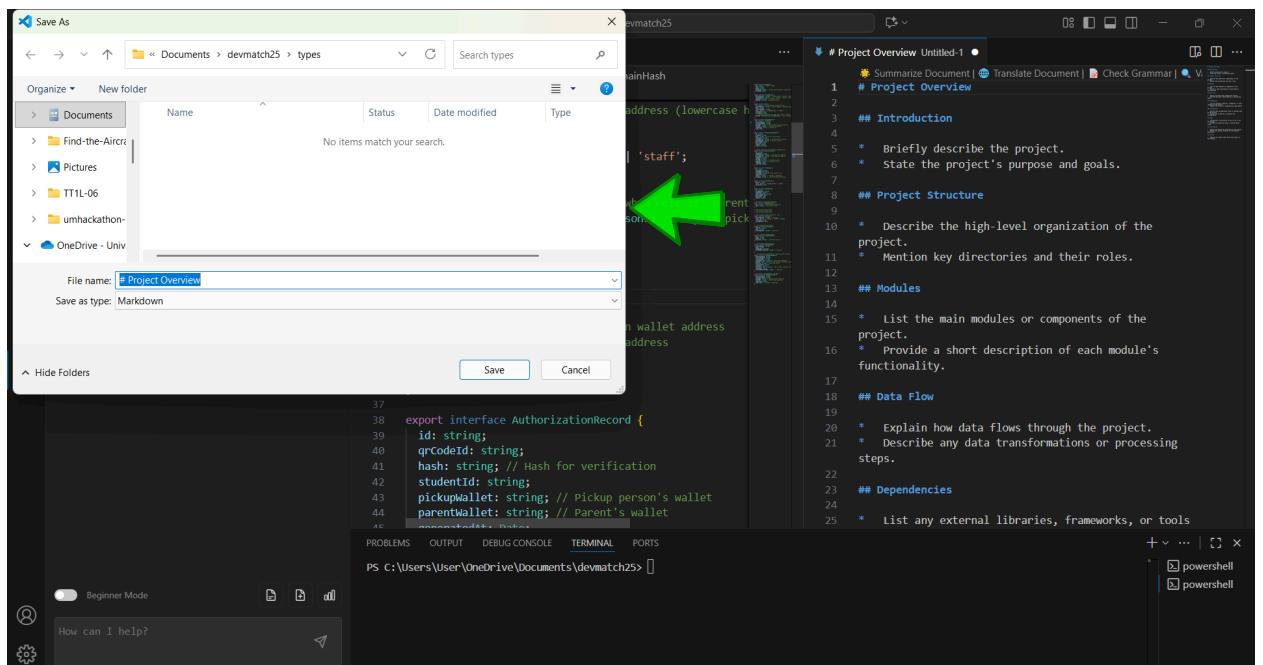
PS C:\Users\User\OneDrive\Documents\devmatch25>

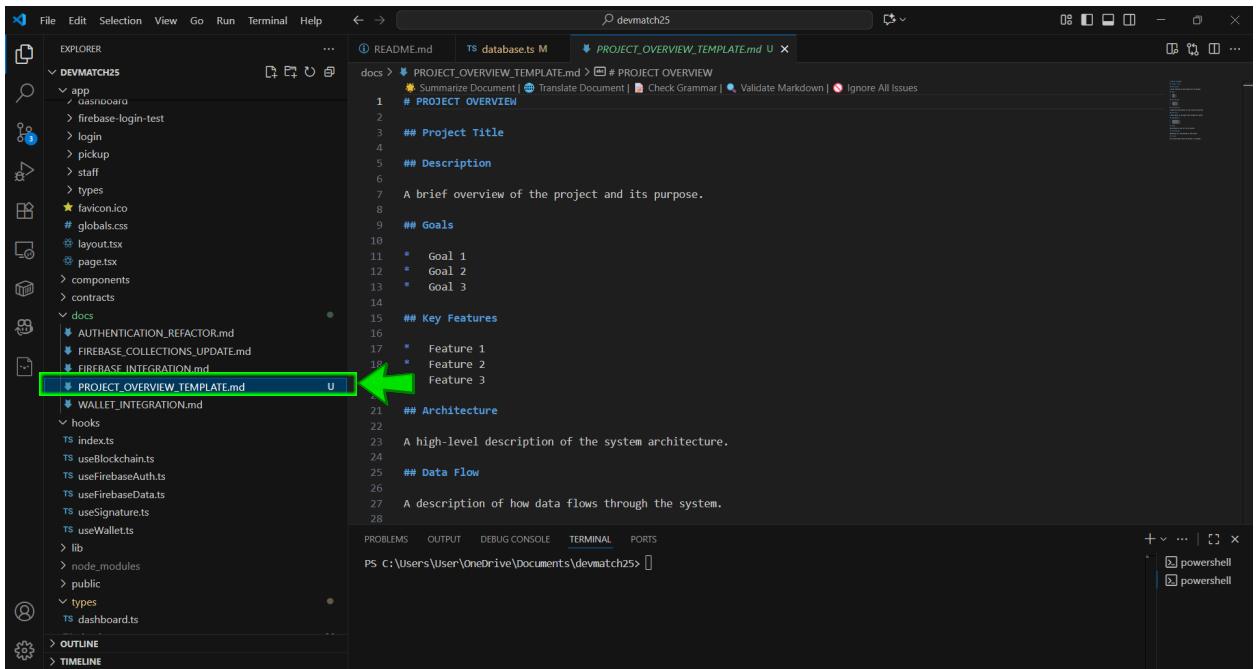
File is saved in the workspace

- d. You can also click the round bullet icon beside the file name. A menu will appear with the options Save, Don't Save, or Cancel.



- e. After selecting Save, choose your preferred save location and rename the file if needed.





The screenshot shows a code editor interface with the following details:

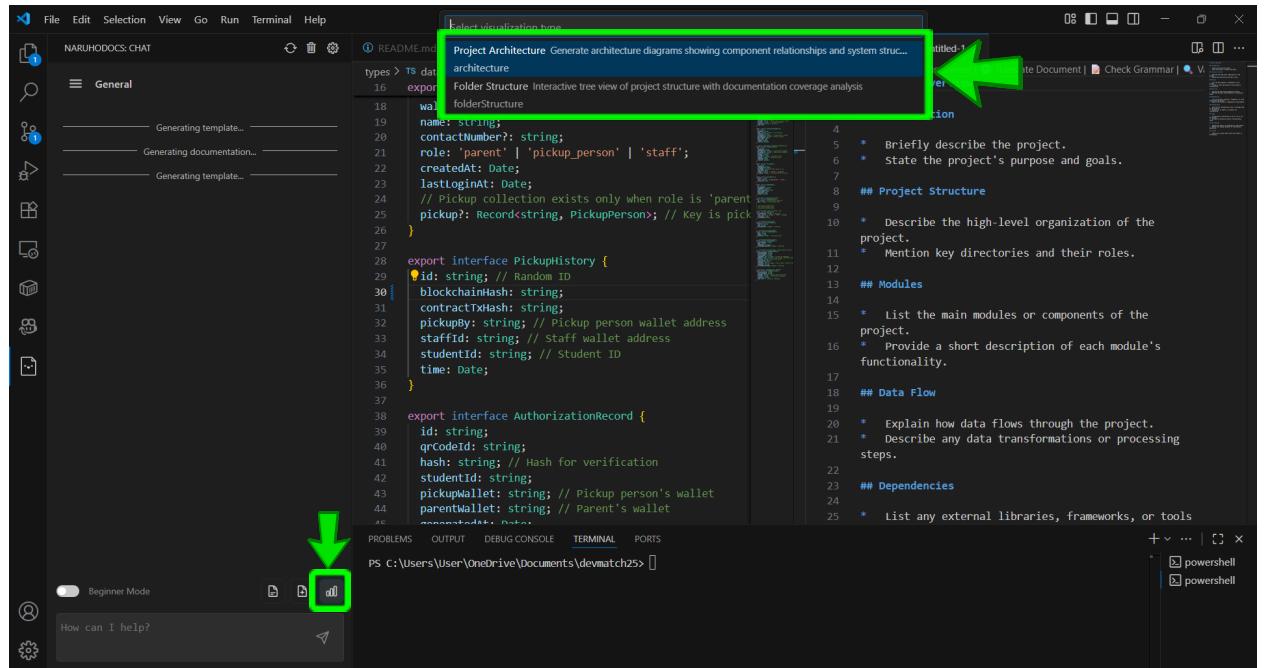
- File Explorer (Left):** Shows a project structure for "DEVMATCH25" with the following files and folders:
 - app
 - firebase-login-test
 - login
 - pickup
 - staff
 - types
 - favicon.ico
 - globals.css
 - layout.tsx
 - page.tsx
 - components
 - contracts
 - docs
 - AUTHENTICATION_REFATOR.md
 - FIREBASE_COLLECTIONS_UPDATE.md
 - FIREBASE_INTEGRATION.md
 - PROJECT_OVERVIEW_TEMPLATE.md
 - WALLET_INTEGRATION.md
 - hooks
 - index.ts
 - useBlockchain.ts
 - useFirebaseAuth.ts
 - useFirebaseData.ts
 - useSignature.ts
 - useWallet.ts
 - lib
 - node_modules
 - public
 - types
 - dashboard.ts
- Editor (Center):** Displays the content of "PROJECT_OVERVIEW_TEMPLATE.md".

```
1 # PROJECT OVERVIEW
2
3 ## Project Title
4
5 ## Description
6
7 A brief overview of the project and its purpose.
8
9 ## Goals
10
11 * Goal 1
12 * Goal 2
13 * Goal 3
14
15 ## Key Features
16
17 * Feature 1
18 * Feature 2
19 * Feature 3
20
21 ## Architecture
22
23 A high-level description of the system architecture.
24
25 ## Data Flow
26
27 A description of how data flows through the system.
28
```
- Bottom Bar:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the command PS C:\Users\User\OneDrive\Documents\devmatch25>.
- Right Side:** Shows a sidebar with "powerShell" listed twice.

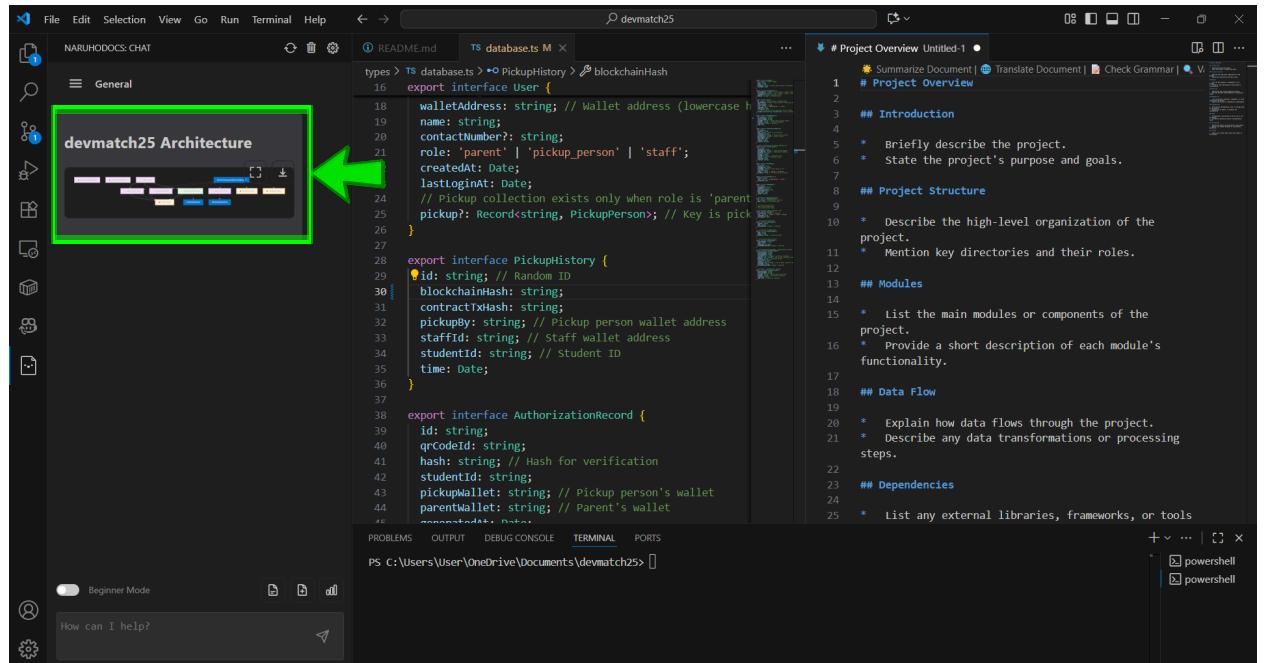
File is saved in the workspace

5.6.3 Visualize Project

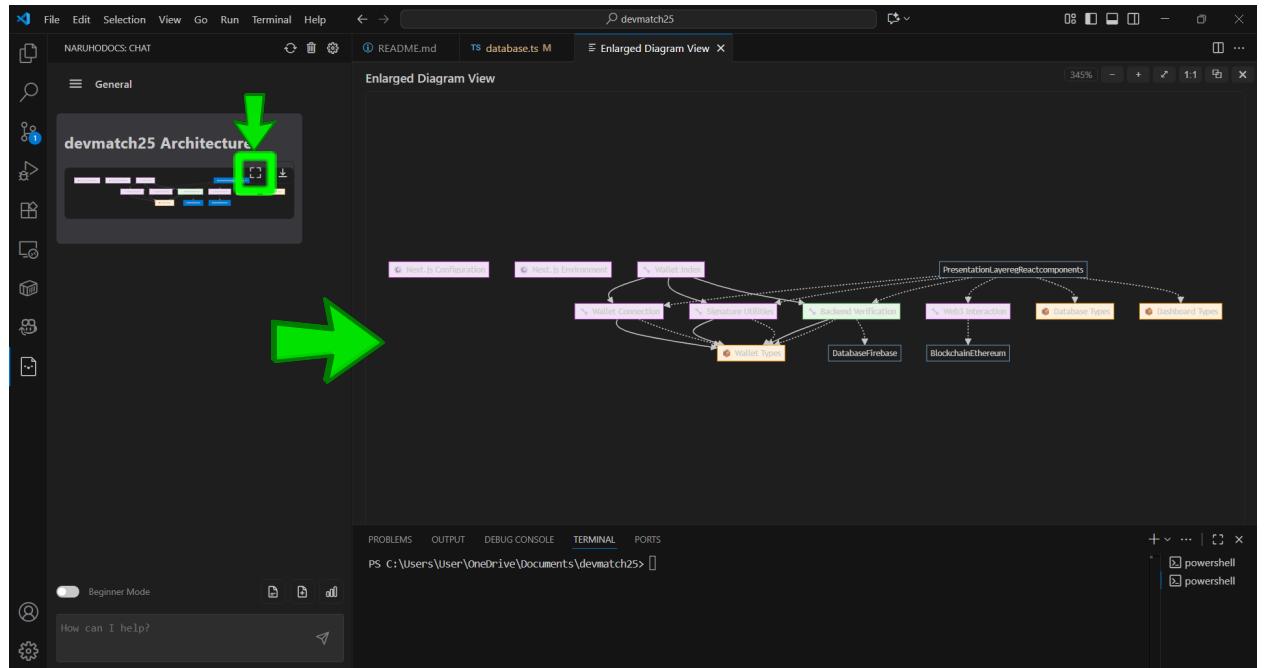
- After clicking the graph icon, you will see two options, Project Architecture and Folder Structure.



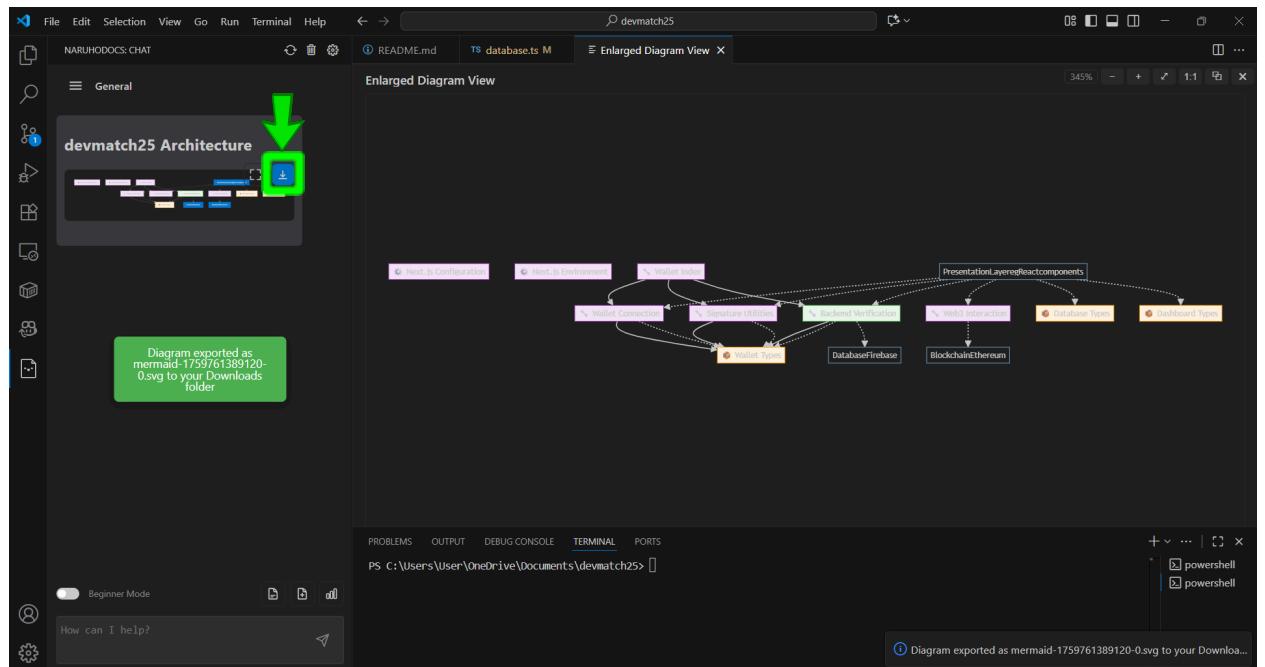
- For Project Architecture, once selected, the diagram will be automatically generated within the chatbot.



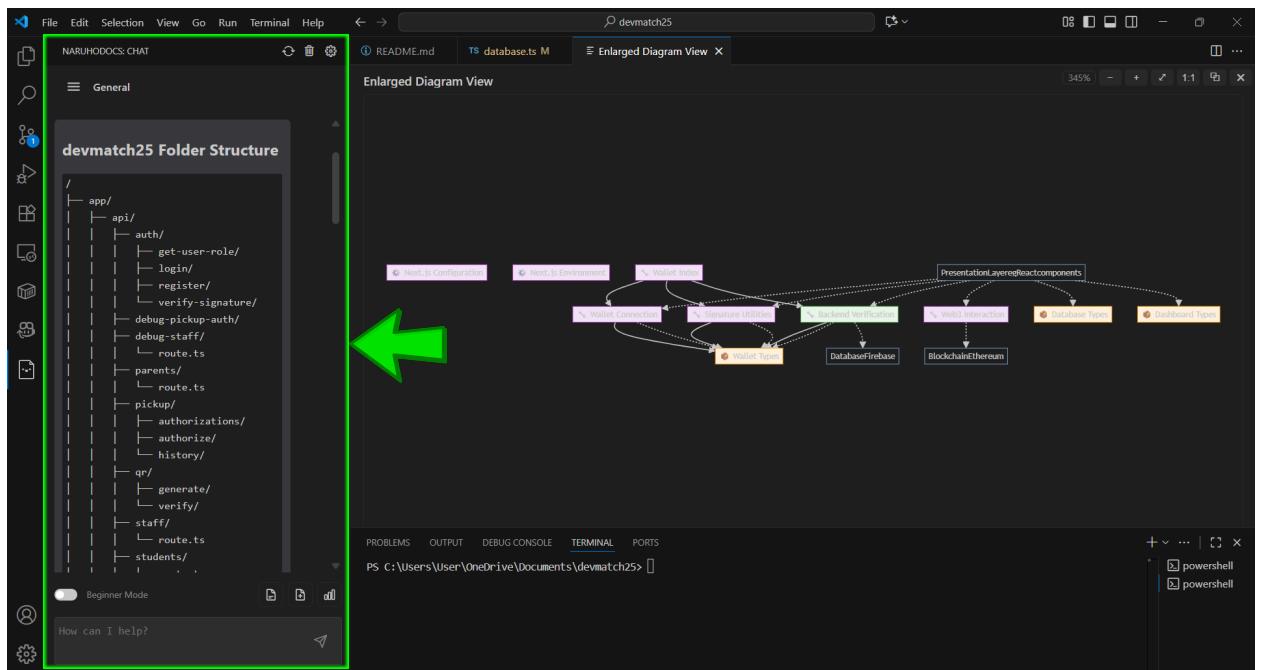
- c. You can choose to enlarge the diagram for a clearer view on the right side.



- d. You can also choose to download it directly—it will be saved to your Downloads folder.



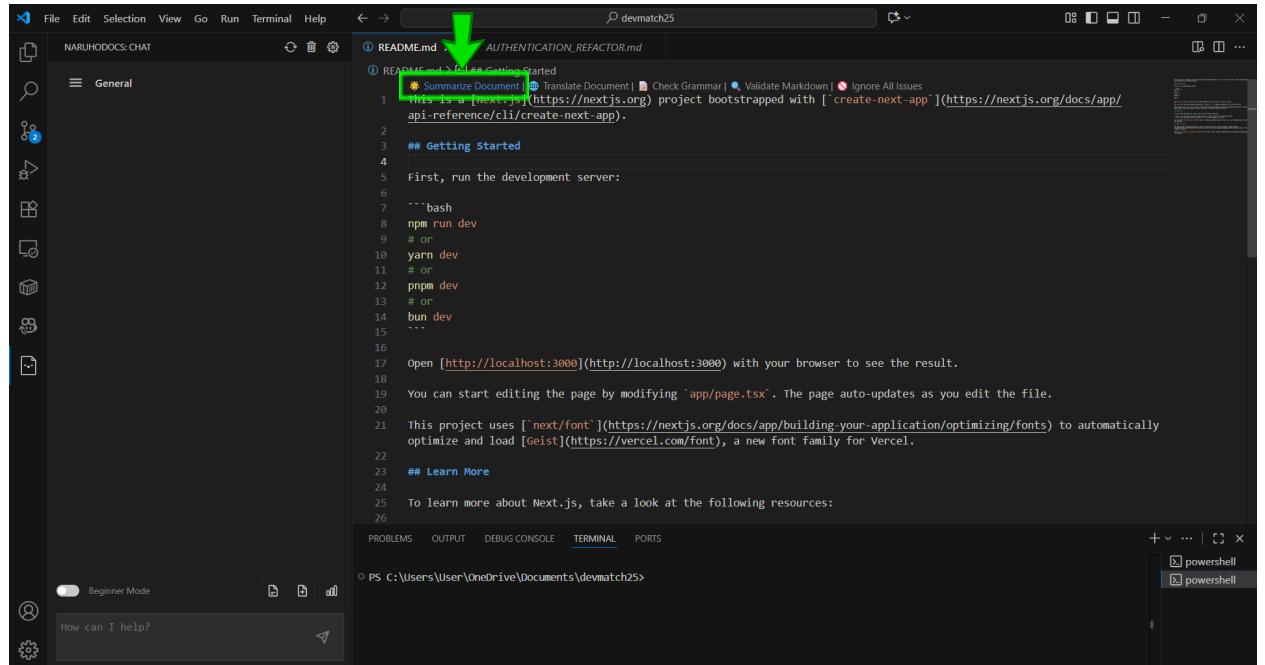
- e. For Folder Structure, the visualization will be generated based on your actual workspace layout, providing an accurate representation of your project's directory structure.



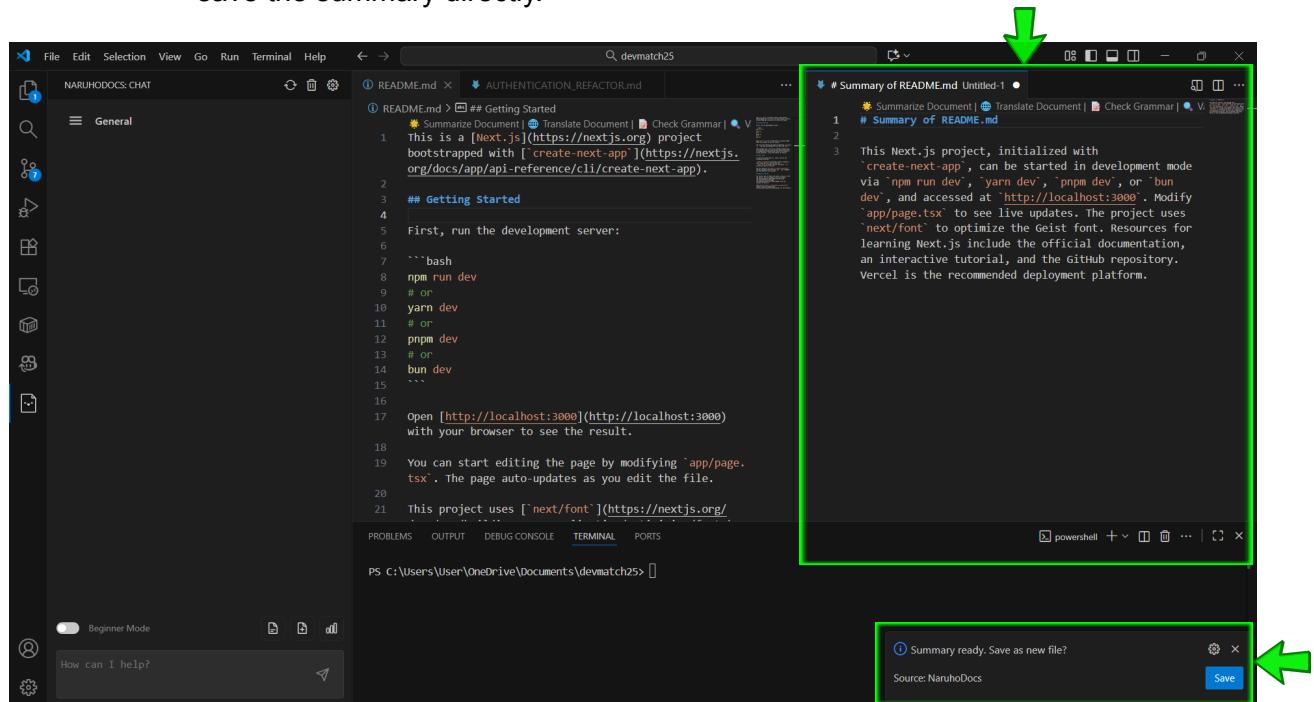
5.7 Document Specific Features

5.7.1 Summarize Document

- Click the “Summarize Document” action button to generate a summary of the selected document.



- Once triggered, the summary will appear on the right side of the editor, and a notification will pop up in the bottom-right corner, allowing you to save the summary directly.



The screenshot shows two instances of the Visual Studio Code (VS Code) interface. The top instance is in 'General' mode, displaying the contents of the `README.md` file. The bottom instance is in 'Dark' mode, showing the file structure in the Explorer sidebar. A green arrow points from the status bar of the top instance to the bottom instance, indicating where the summary file was saved.

Top Instance (General Mode):

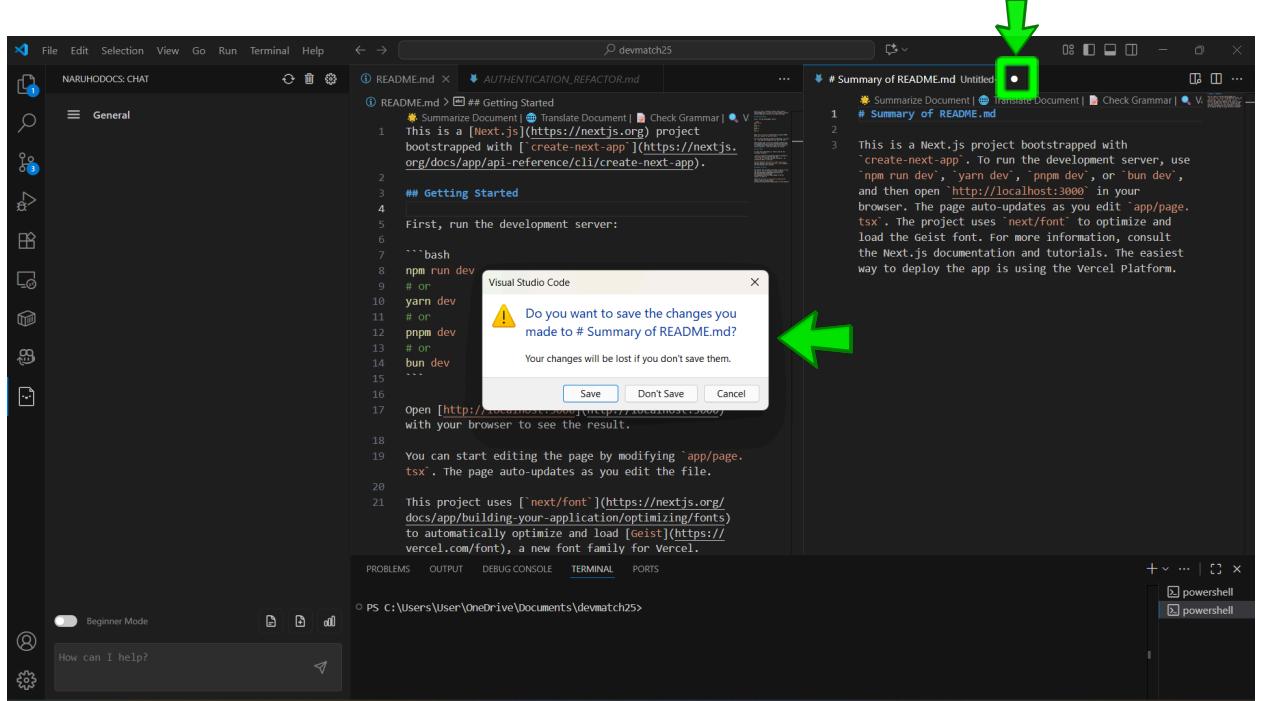
- File menu: File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search bar: devmatch25
- Editor tabs: README.md, AUTHENTICATION_REFATOR.Rmd, # Summary of README.md
- Editor content: The README.md file contains instructions for getting started with a Next.js project. The summary file, '# Summary of README.md', is shown in the right panel.
- Status bar: PS C:\Users\User\OneDrive\Documents\devmatch25> [terminal icon]
- Bottom status bar: Beginner Mode, How can I help?
- Bottom right status bar: Saved summary to c:\Users\User\OneDrive\Documents\devmatch25\d...

Bottom Instance (Dark Mode):

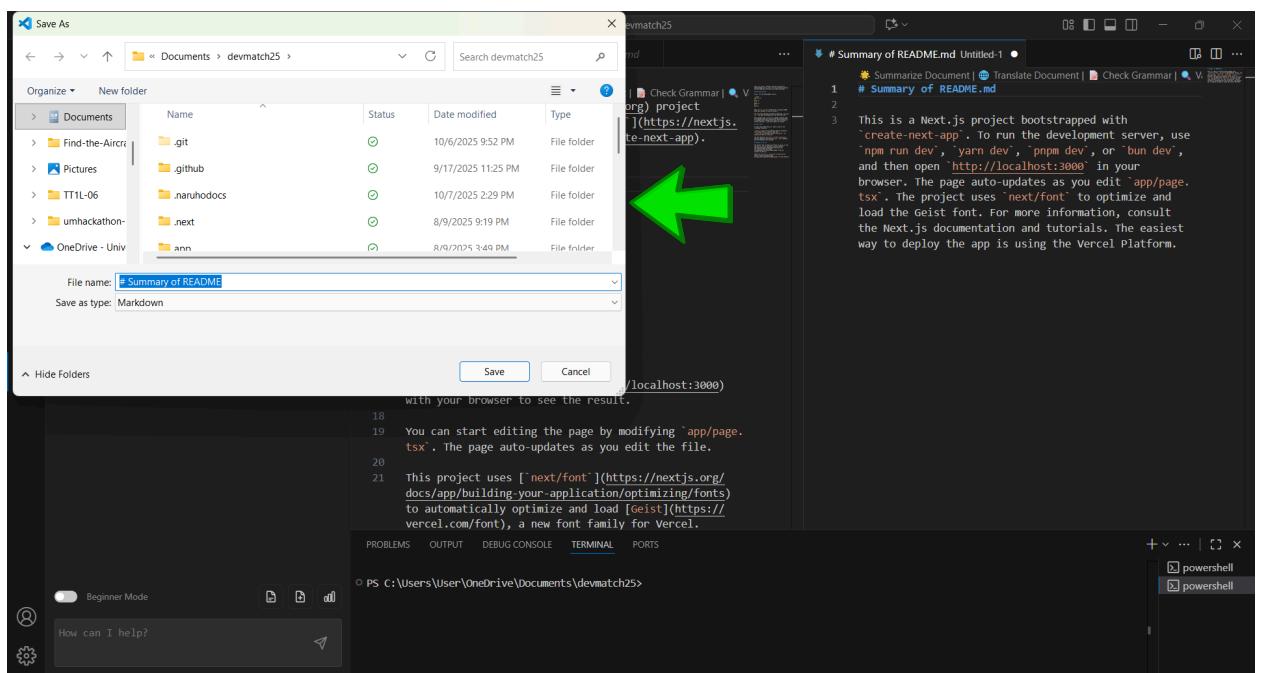
- File menu: File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search bar: devmatch25
- Explorer sidebar: Shows the project structure under 'DEVMATCH25'. The file '# Summary of README.md' is highlighted with a green box and has a green arrow pointing to it from the top instance's status bar.
- Editor tabs: README.md, README_Summary.md, AUTHENTICATION_REFATOR.Rmd
- Editor content: The summary file contains the same content as the original README.md file.
- Status bar: PS C:\Users\User\OneDrive\Documents\devmatch25> [terminal icon]

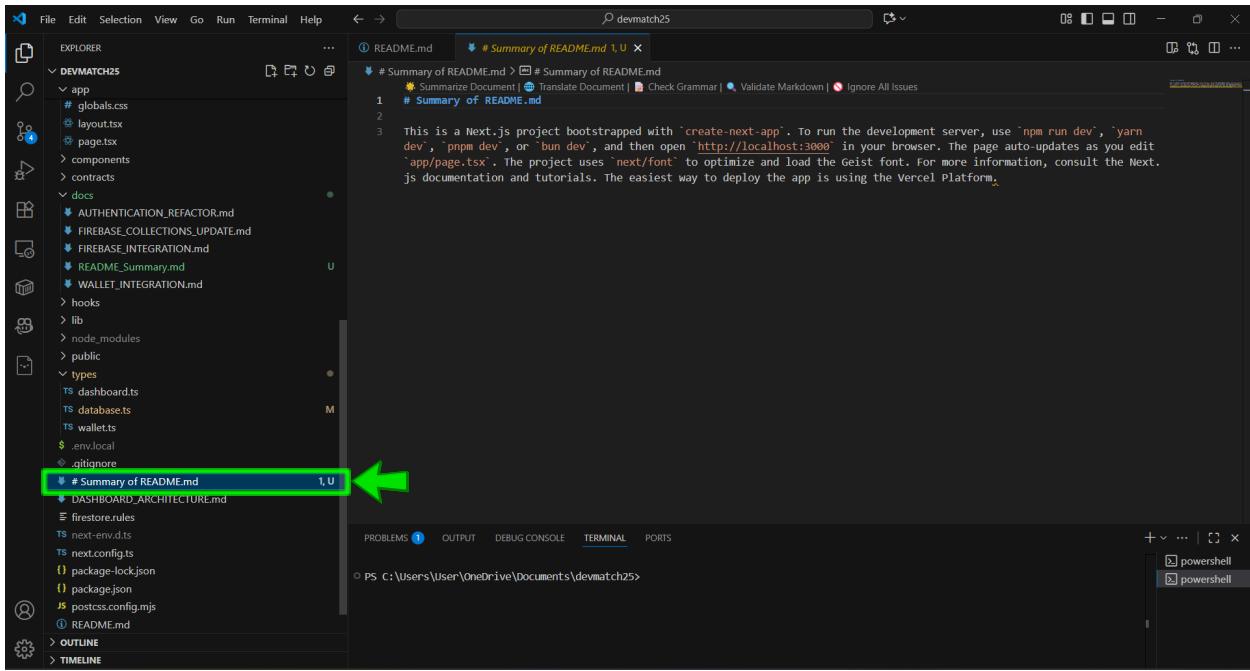
File is saved in the workspace

- c. Alternatively, you can click the icon beside the file name to open a menu with the options Save, Don't Save, or Cancel.



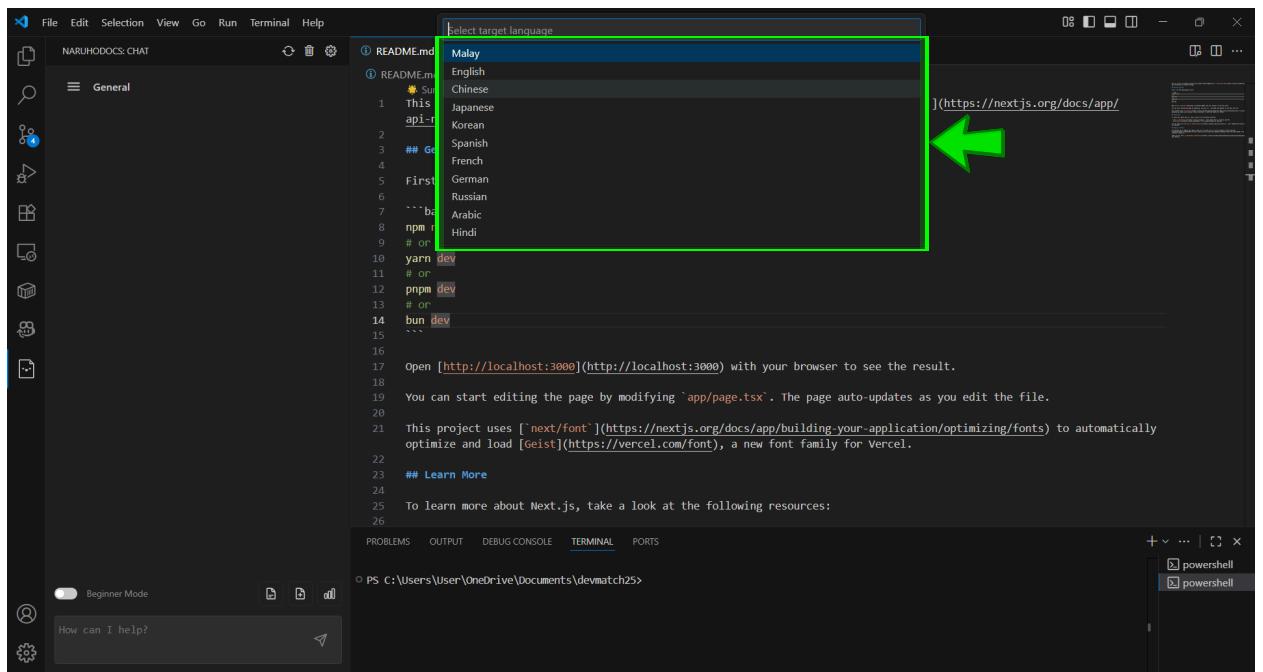
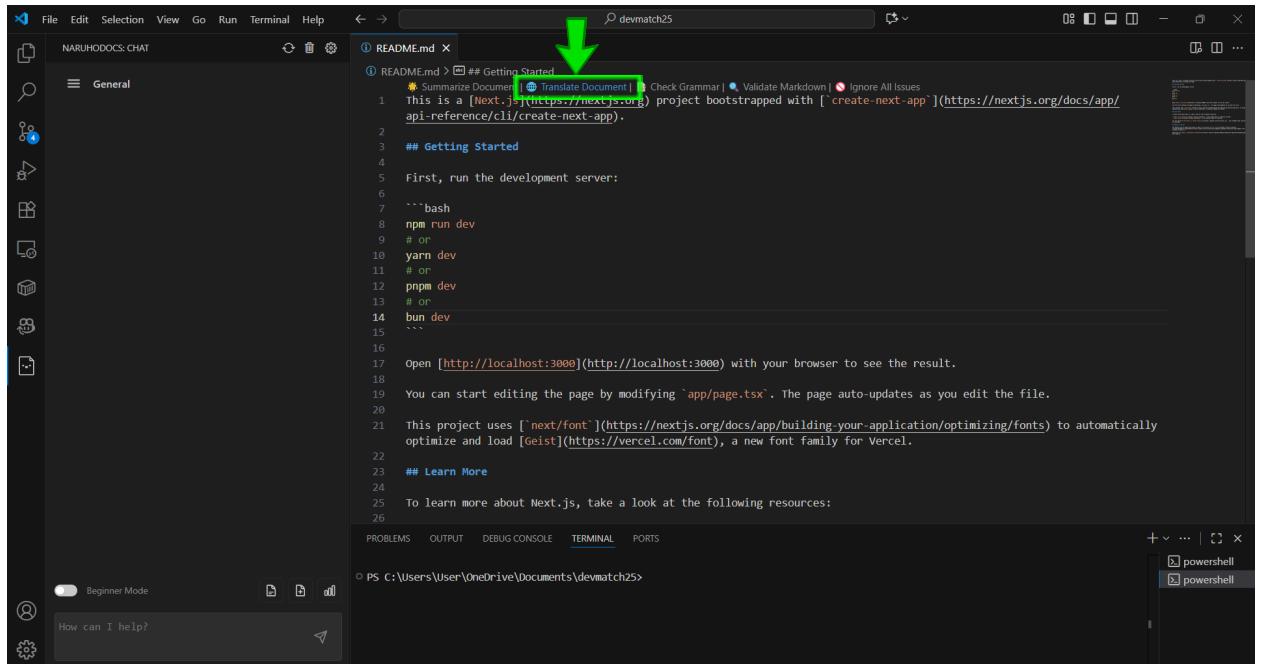
- d. After selecting Save, you can choose your preferred save location and rename the file if desired.





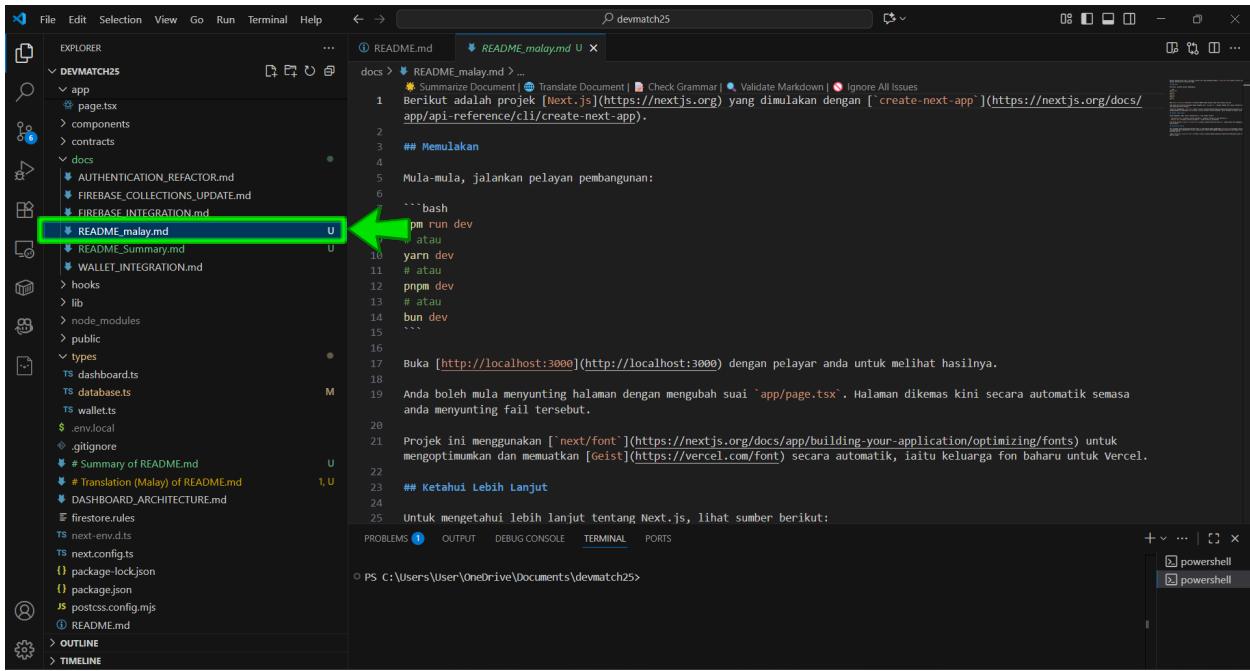
5.7.2 Translate Document

- Click the “Translate Document” action button to open a list of available languages.



- b. Once a language is selected, the translated version of the document will appear on the right side of the editor. A notification will also appear in the bottom-right corner, allowing you to save the translated document directly.

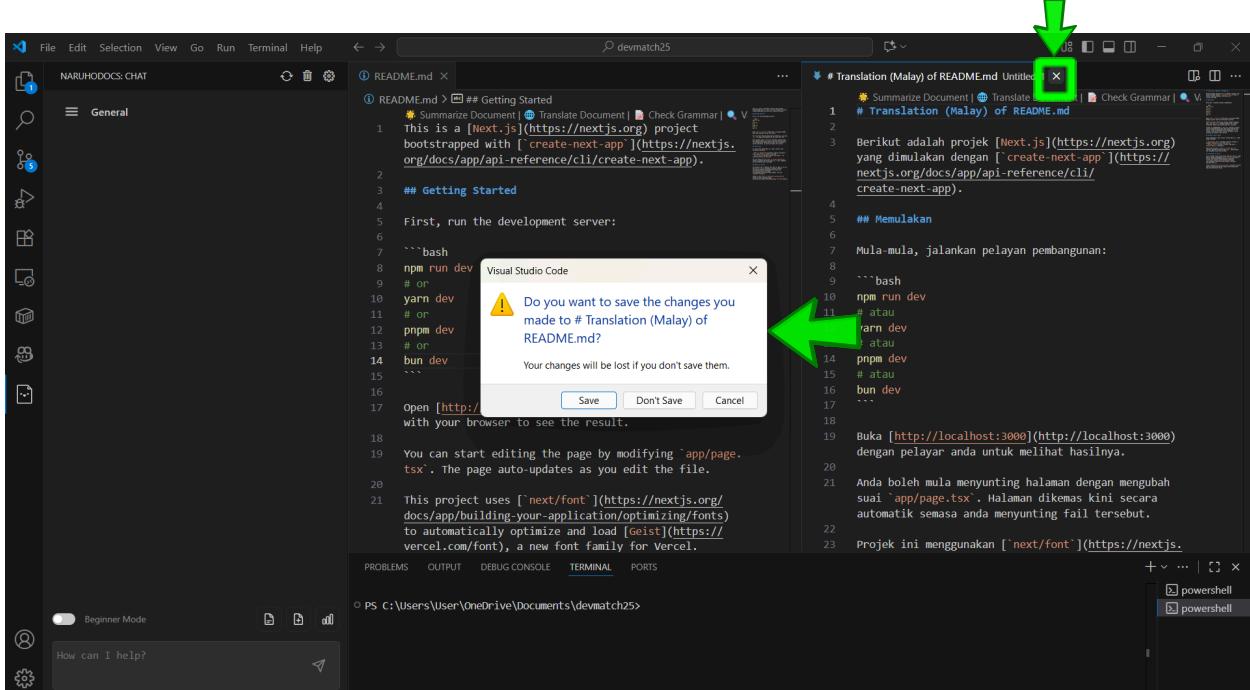
The screenshot illustrates the Microsoft Translator integration in Visual Studio Code. On the left, the original README.md file is shown in English. On the right, the translated content is displayed in Malay. A green arrow points from the top right towards the bottom right corner, where a modal dialog box appears asking if the user wants to save the translation. Another green arrow points from this dialog box towards the bottom right corner, where a message 'Saved translation to c:\Users\User\OneDrive\Documents\devmatch25...' is displayed, indicating the successful saving of the translated file.



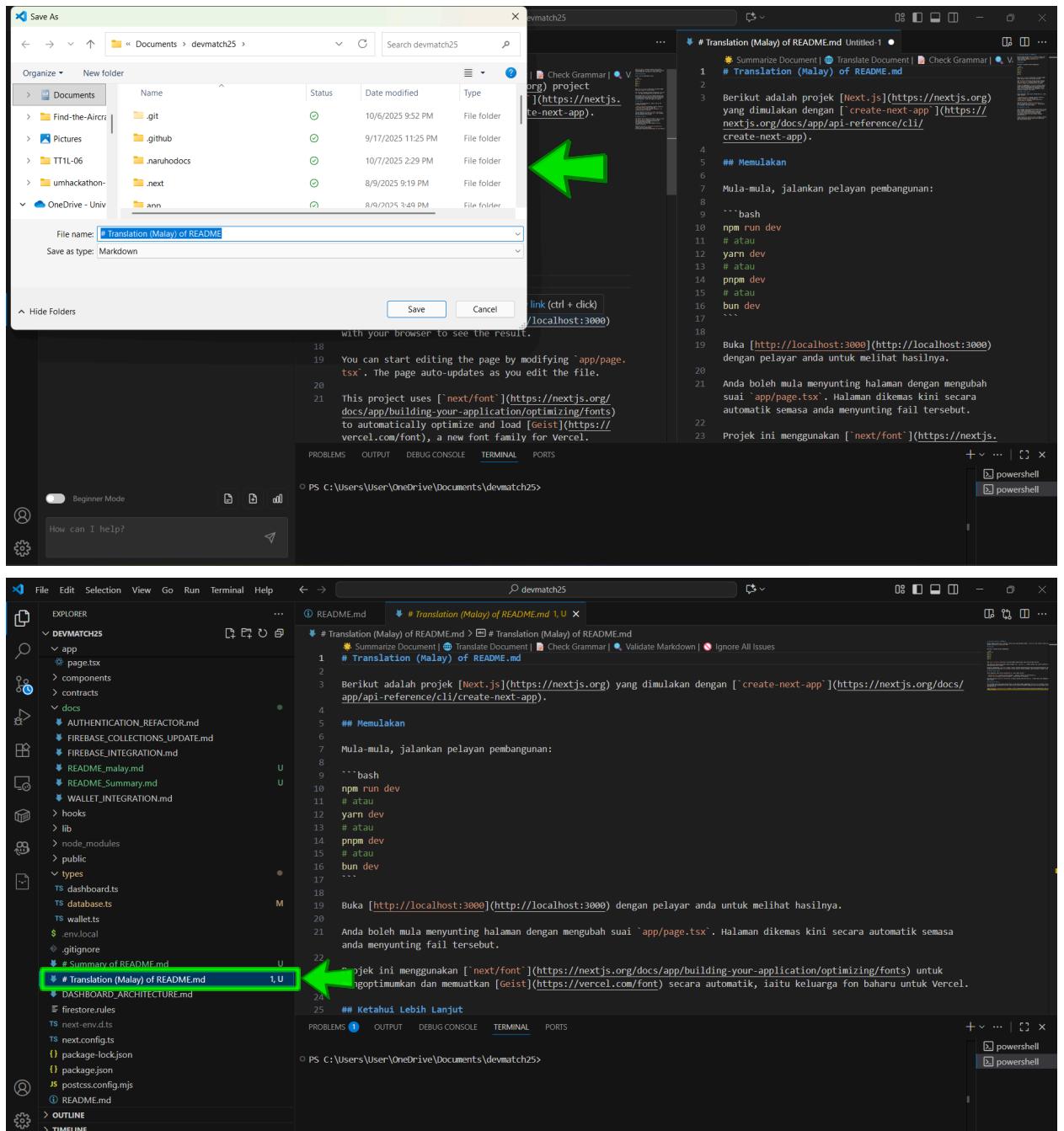
The screenshot shows the Visual Studio Code interface with the workspace titled 'devmatch25'. In the Explorer sidebar, there is a tree view of files and folders. One file, 'README_malay.md', is highlighted with a green border and has a green arrow pointing to its name. The main editor area displays the content of 'README_malay.md', which is a summary of a Next.js project. The status bar at the bottom shows the path 'PS C:\Users\User\OneDrive\Documents\devmatch25>'.

File is saved in the workspace

- Alternatively, you can click the icon beside the file name to open a menu with the options Save, Don't Save, or Cancel.



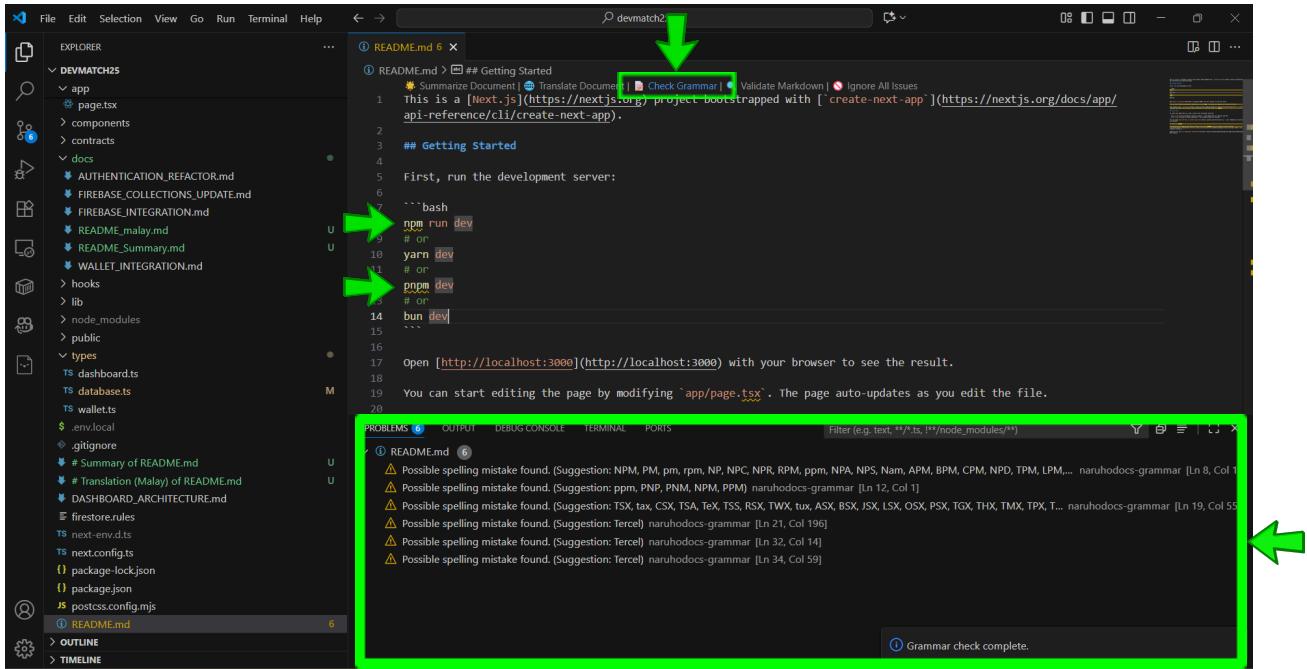
d. After selecting Save, you can choose your preferred save location and rename the file if desired.



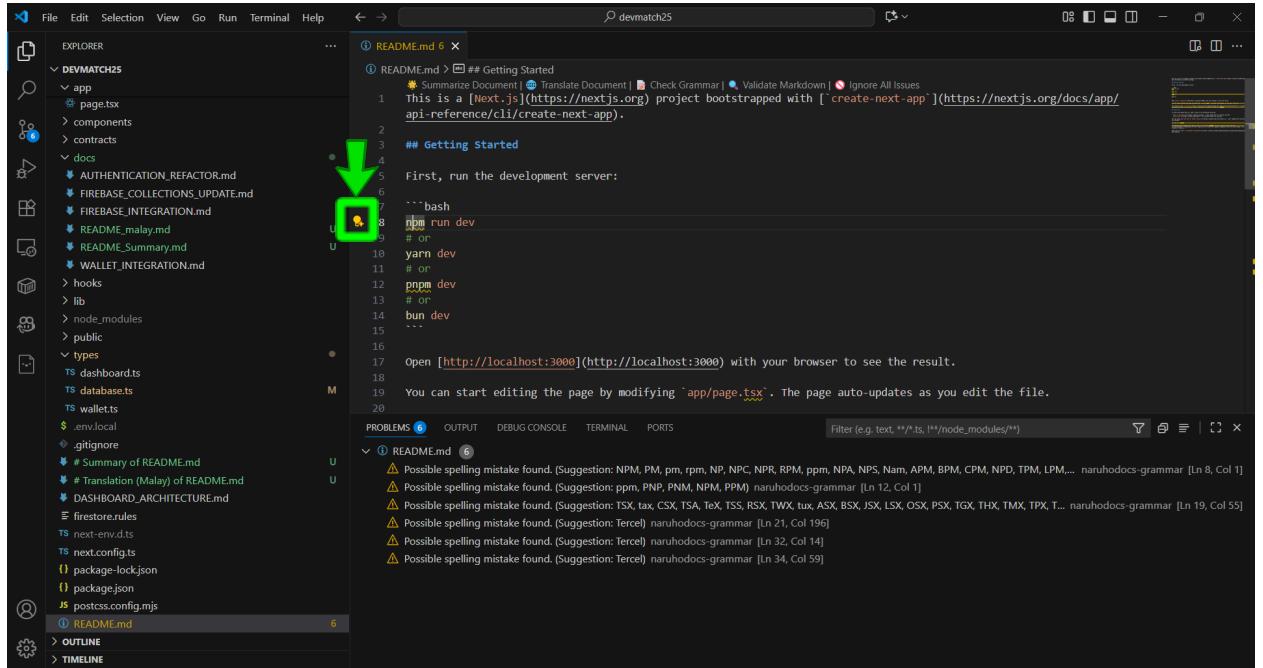
File is saved in the workspace

5.7.3 Check Grammar

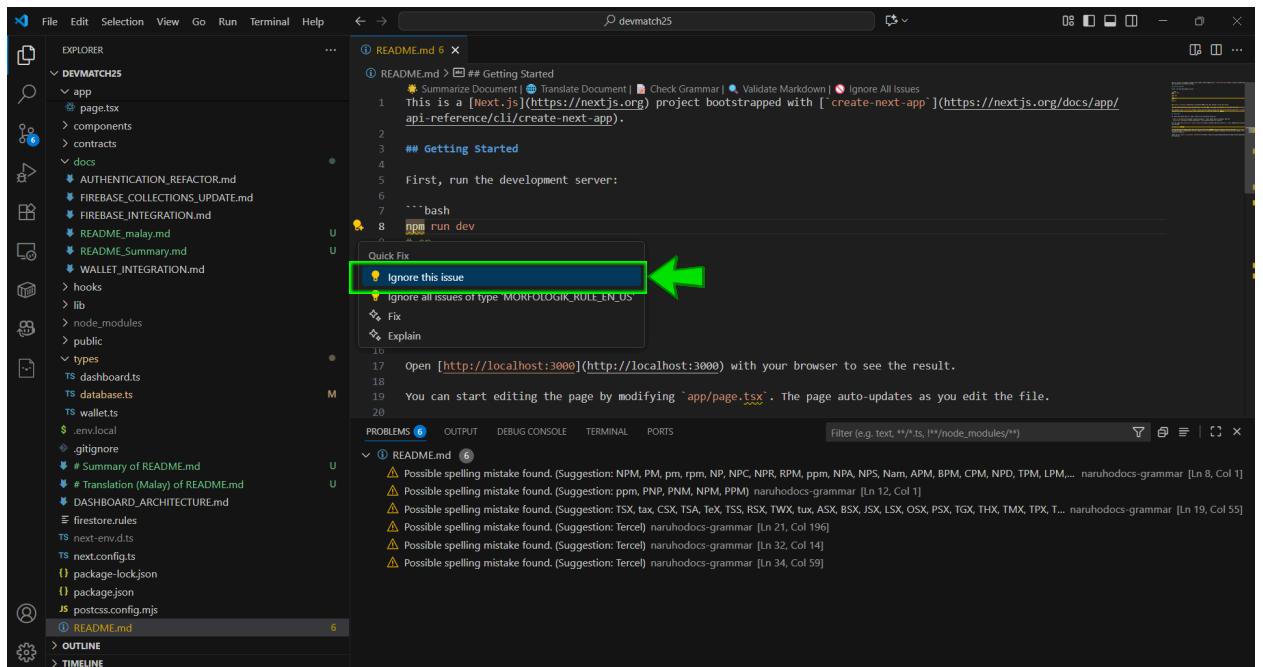
- Click the “Check Grammar” action button to perform a grammar check on the document.
- Detected grammar issues will be underlined with a yellow wavy line and can also be viewed in the Problems panel.

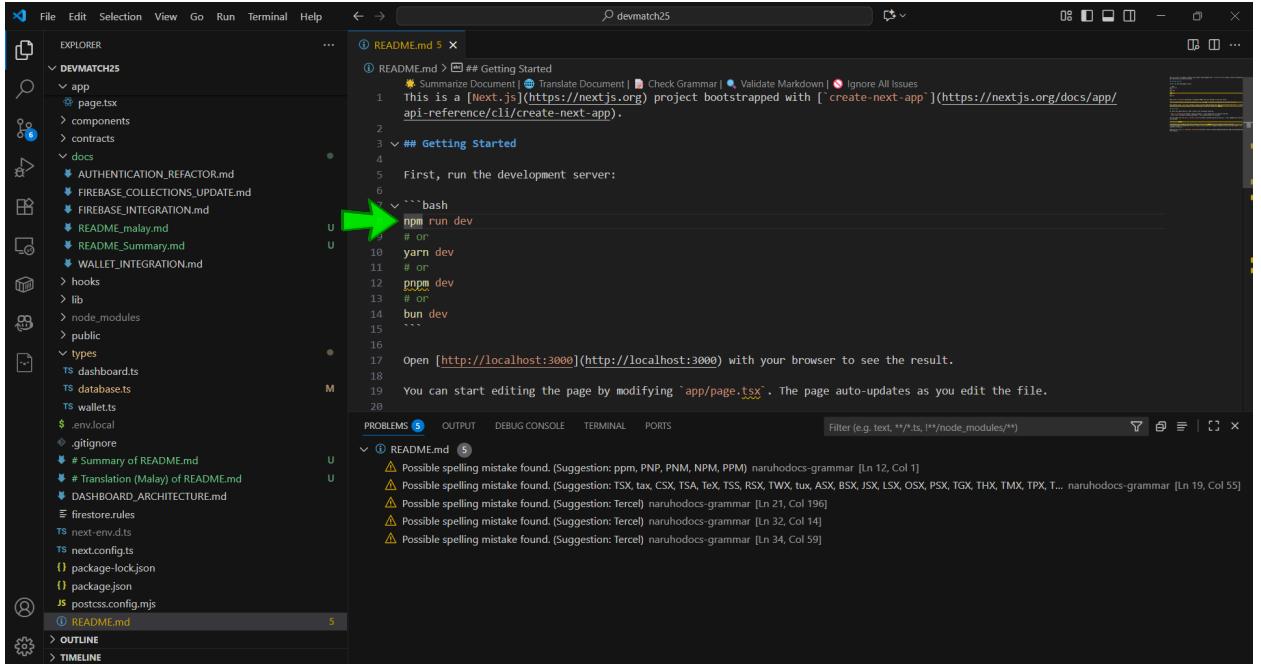


- c. If you believe the underlined word is not an error, click on the word to display a light bulb icon.



- d. Right-click the light bulb to open a menu where you can choose to ignore this specific error.





The screenshot shows the VS Code interface with the file `README.md` open. In the Problems panel, there are several yellow warning icons followed by error messages related to spelling mistakes. A green arrow points to the first error message:

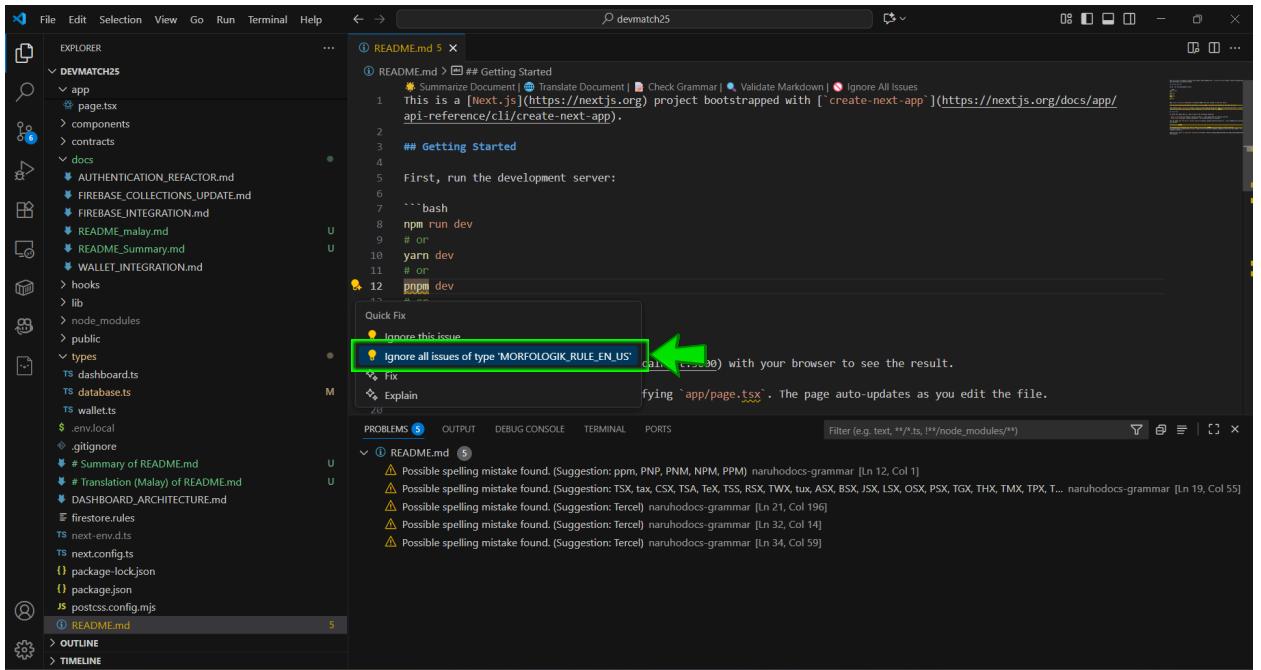
```

1 Possible spelling mistake found. (Suggestion: ppm, PNP, PNM, NPM, PPM) naruhodocs-grammar [Ln 12, Col 1]
2 Possible spelling mistake found. (Suggestion: TSX, tax, CSX, TSA, Tex, TSS, RSX, TWX, tax, ASX, BSX, JSX, LSX, OSX, PSX, TGX, THX, TMX, TPX, T... naruhodocs-grammar [Ln 19, Col 55]
3 Possible spelling mistake found. (Suggestion: Terce) naruhodocs-grammar [Ln 21, Col 196]
4 Possible spelling mistake found. (Suggestion: Terce) naruhodocs-grammar [Ln 32, Col 14]
5 Possible spelling mistake found. (Suggestion: Terce) naruhodocs-grammar [Ln 34, Col 59]

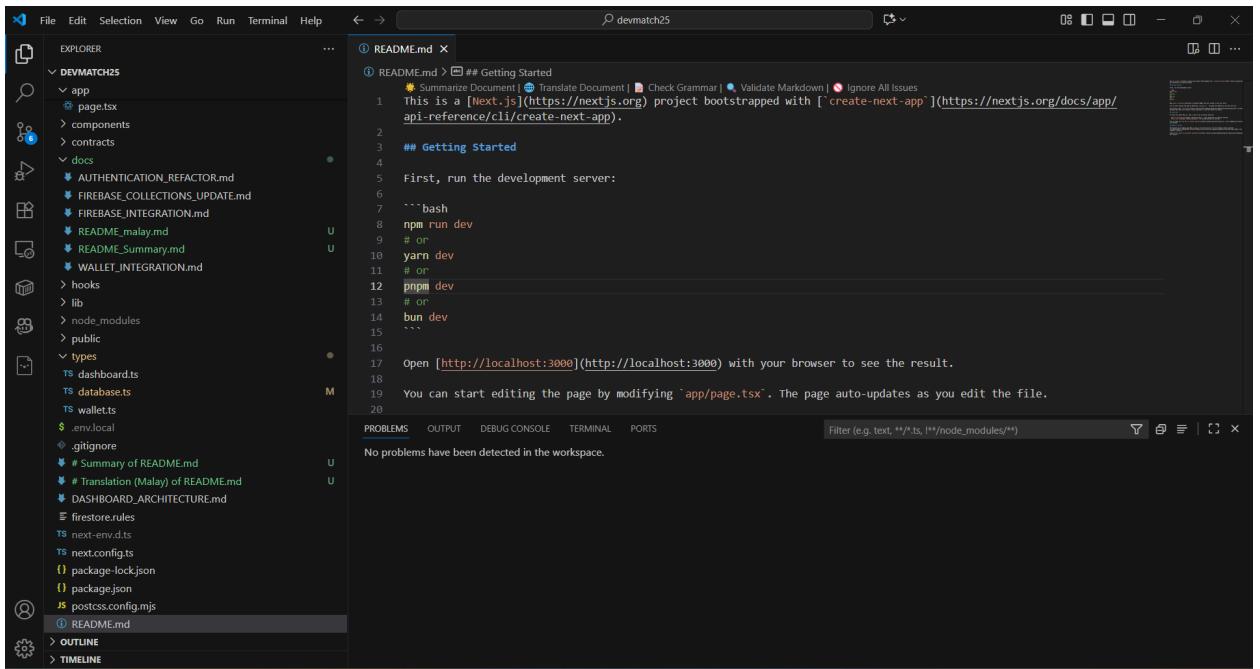
```

The grammar error has been ignored

e. You can also ignore all occurrences of the same error type.



The screenshot shows the VS Code interface with the file `README.md` open. A context menu is open over a yellow warning icon in the Problems panel, with the option `Ignore this issue` highlighted. A green arrow points to this option. The rest of the menu includes `Fix`, `Explain`, and `Quick Fix`.

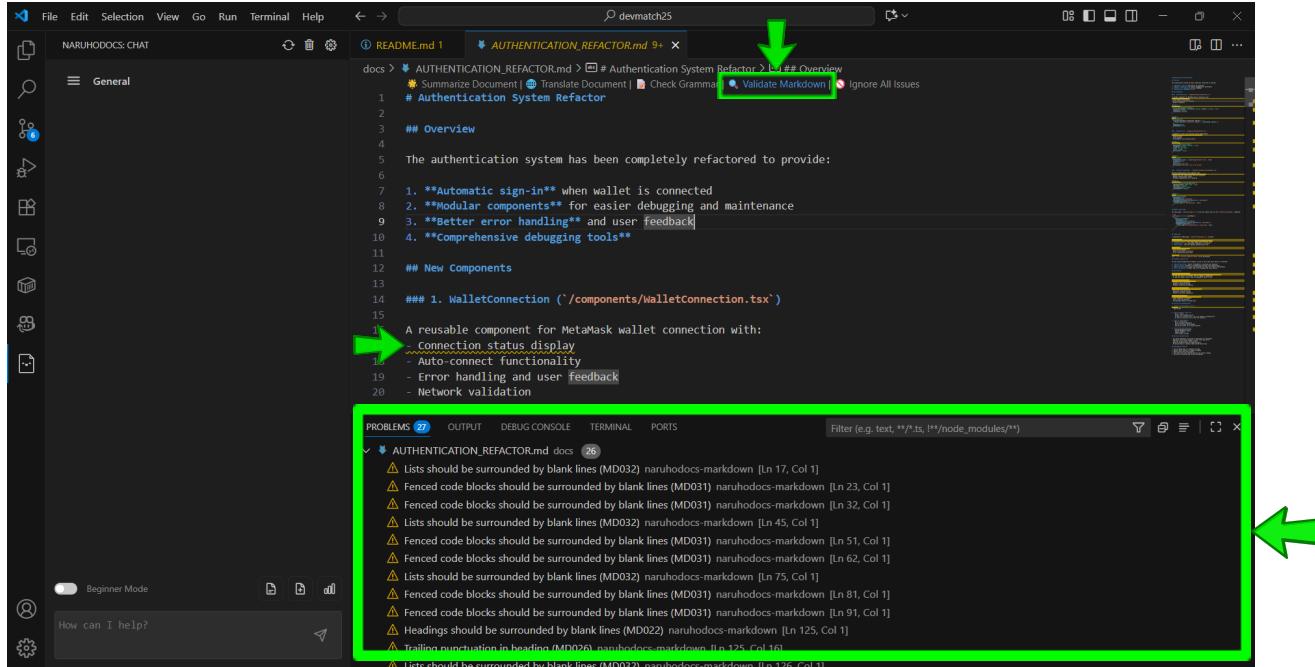


Same type of grammar error has been ignored

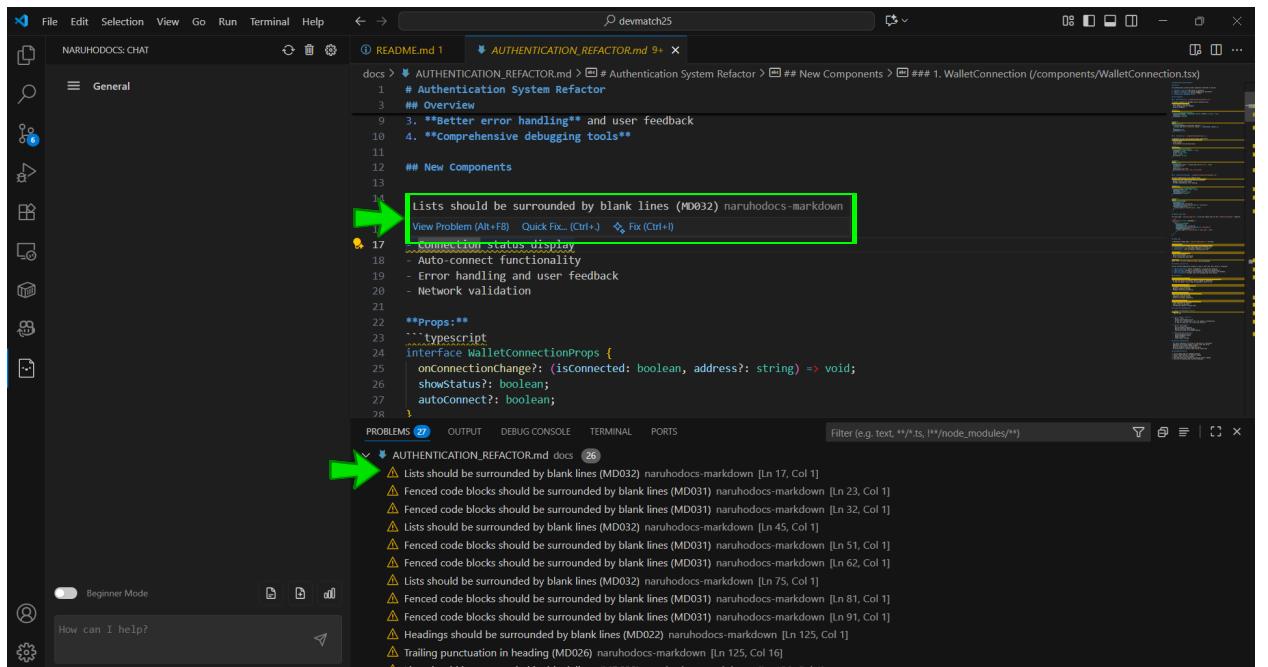
```
① README.md X
① README.md > ## Getting Started
  * Summarize Document | Translate Document | Check Grammar | Validate Markdown | Ignore All Issues
  1 This is a [Next.js](https://nextjs.org) project bootstrapped with [`create-next-app`](https://nextjs.org/docs/app/api-reference/cli/create-next-app).
  2
  3 ## Getting Started
  4
  5 First, run the development server:
  6
  7   ```bash
  8   npm run dev
  9   # or
 10  #   yarn dev
 11  # or
 12  #   pnpm dev
 13  # or
 14  bun dev
 15
 16
 17 Open [http://localhost:3000](http://localhost:3000) with your browser to see the result.
 18
 19 You can start editing the page by modifying `app/page.tsx`. The page auto-updates as you edit the file.
 20
 21
 22
 23
 24
 25
 26
 27
 28
```

5.7.4 Validate Markdown

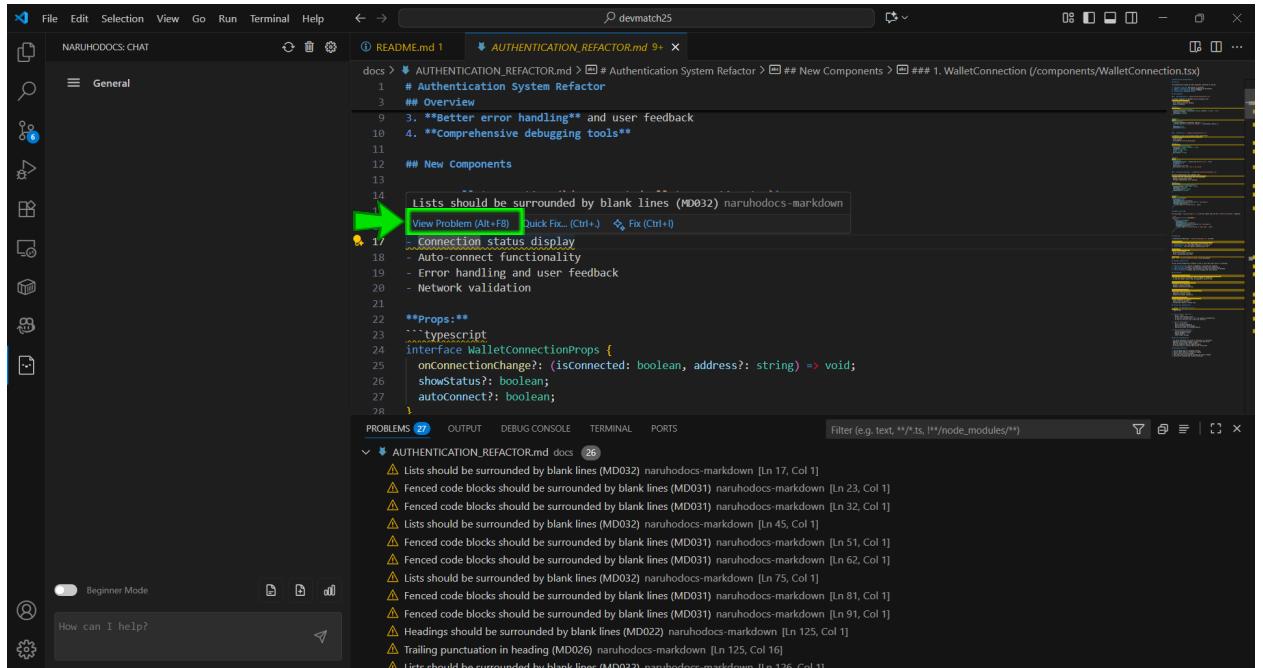
- a. Click the “Validate Markdown” action button to perform a markdown validation check on the document.
- b. Any detected issues will be highlighted with a yellow wavy underline and also listed in the Problems panel.



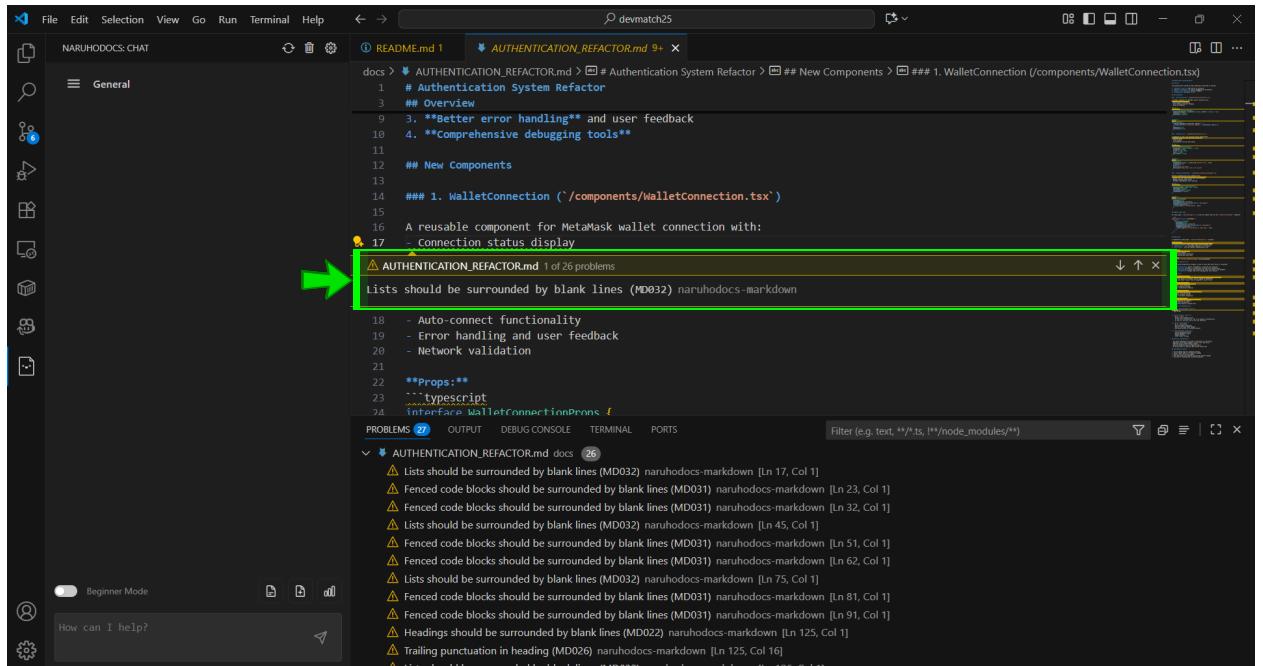
- c. Hover over the underlined text to view detailed error information, which corresponds to the entries shown in the Problems panel.



- d. You can navigate between detected issues by using the “View Problem” option.

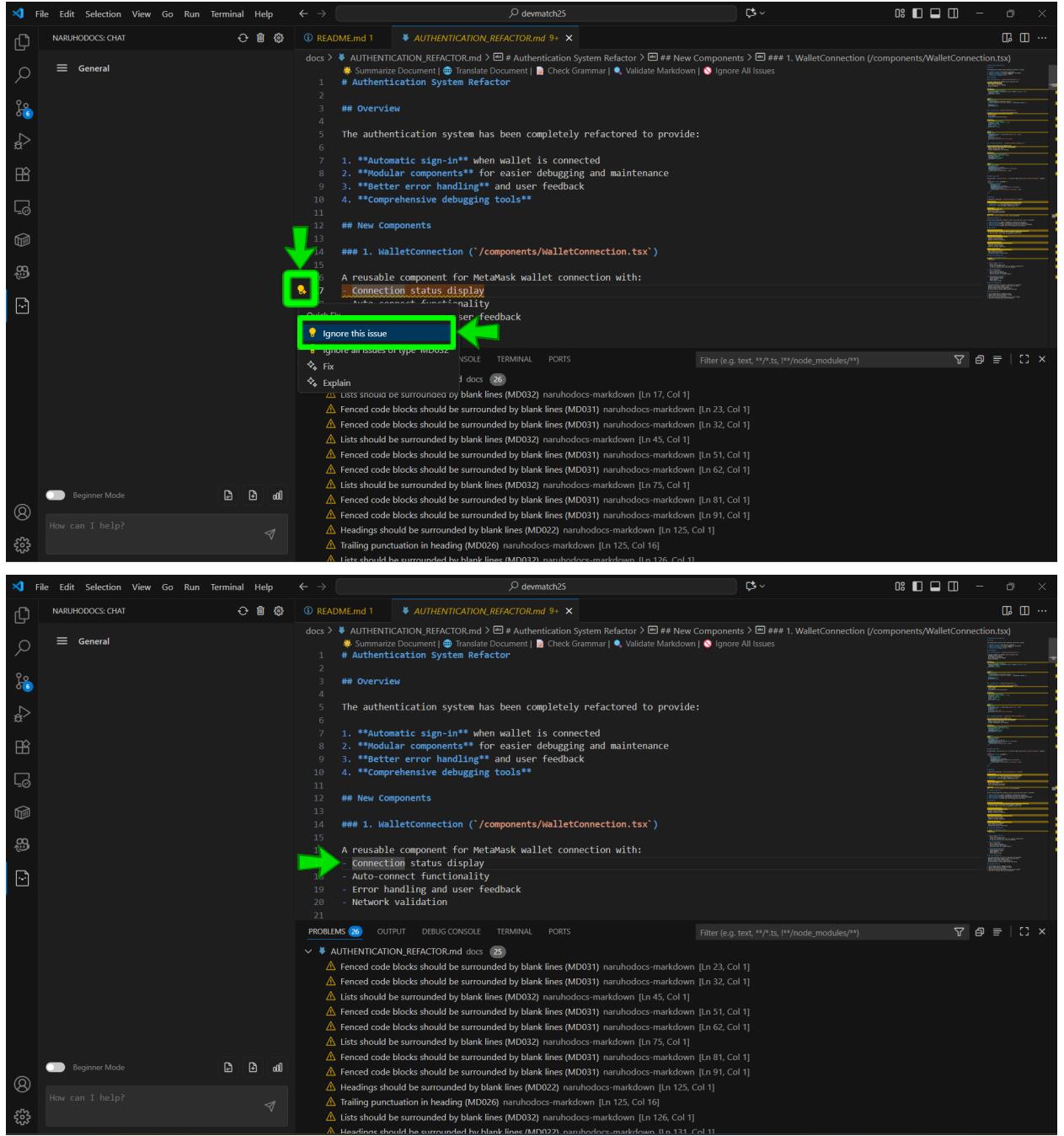


The screenshot shows the VS Code interface with two panels. The main panel displays a Markdown file named `AUTHENTICATION_REFATOR.md`. The bottom panel is the 'PROBLEMS' panel, which lists 26 errors. A green arrow points from the 'View Problem' button in the status bar to the first error in the 'PROBLEMS' panel, which is highlighted with a red border. The error message is: 'Lists should be surrounded by blank lines (MD032) naruhodocs-markdown'. The status bar at the bottom also shows the count '26' next to the 'PROBLEMS' tab.

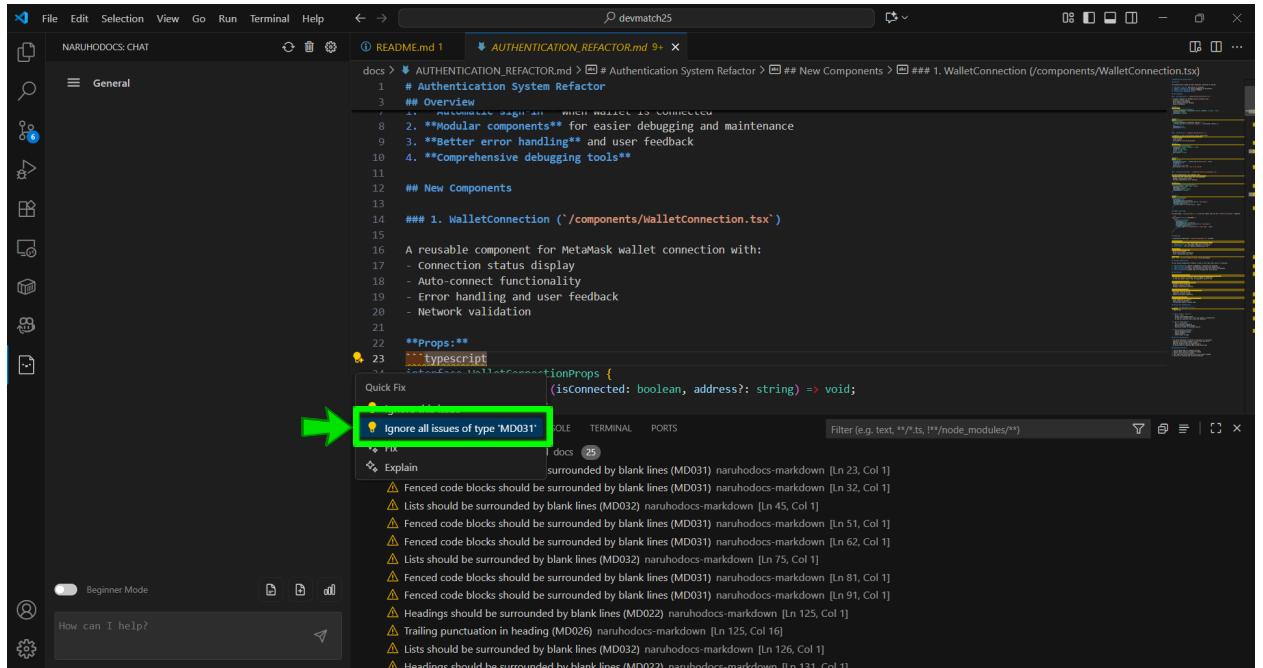


This screenshot is similar to the one above, but the 'PROBLEMS' panel now shows 28 errors. The same red-highlighted error from the previous screenshot is still present. A green arrow points from the 'View Problem' button in the status bar to the same error in the 'PROBLEMS' panel.

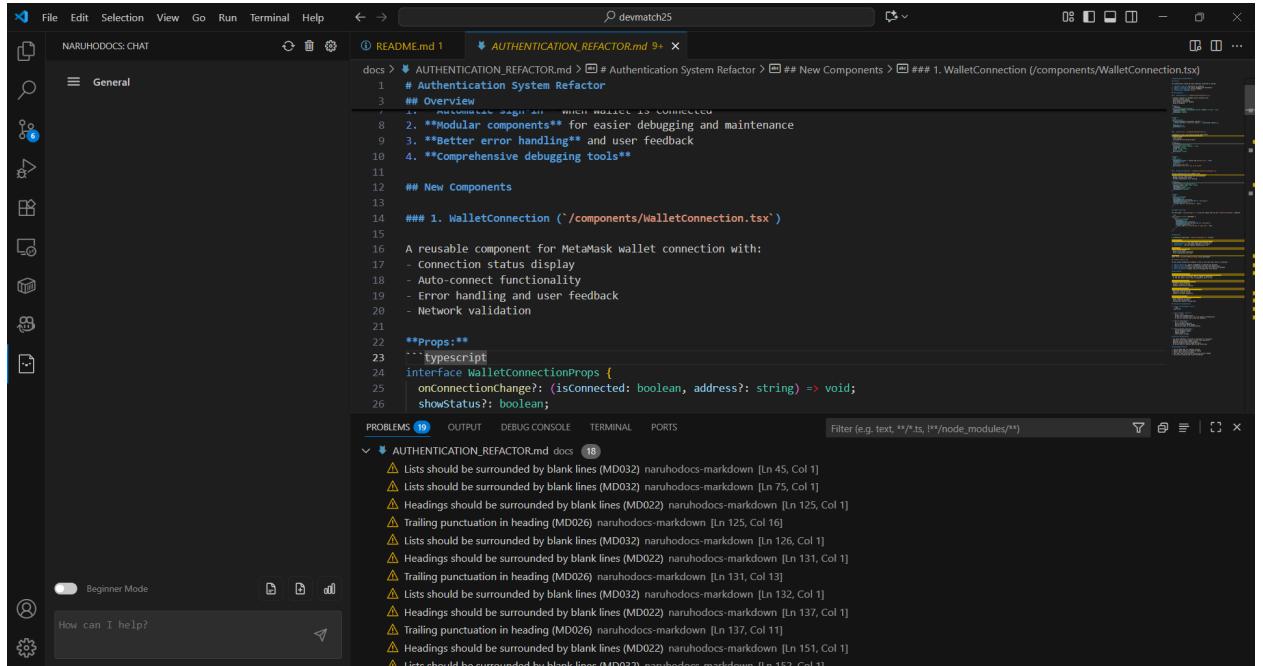
- e. If you believe an underlined word is not an actual error, click on it to display a light bulb icon.
- f. Right-click the light bulb to open a menu where you can choose to ignore this specific error.



g. You can also ignore all occurrences of the same error type.



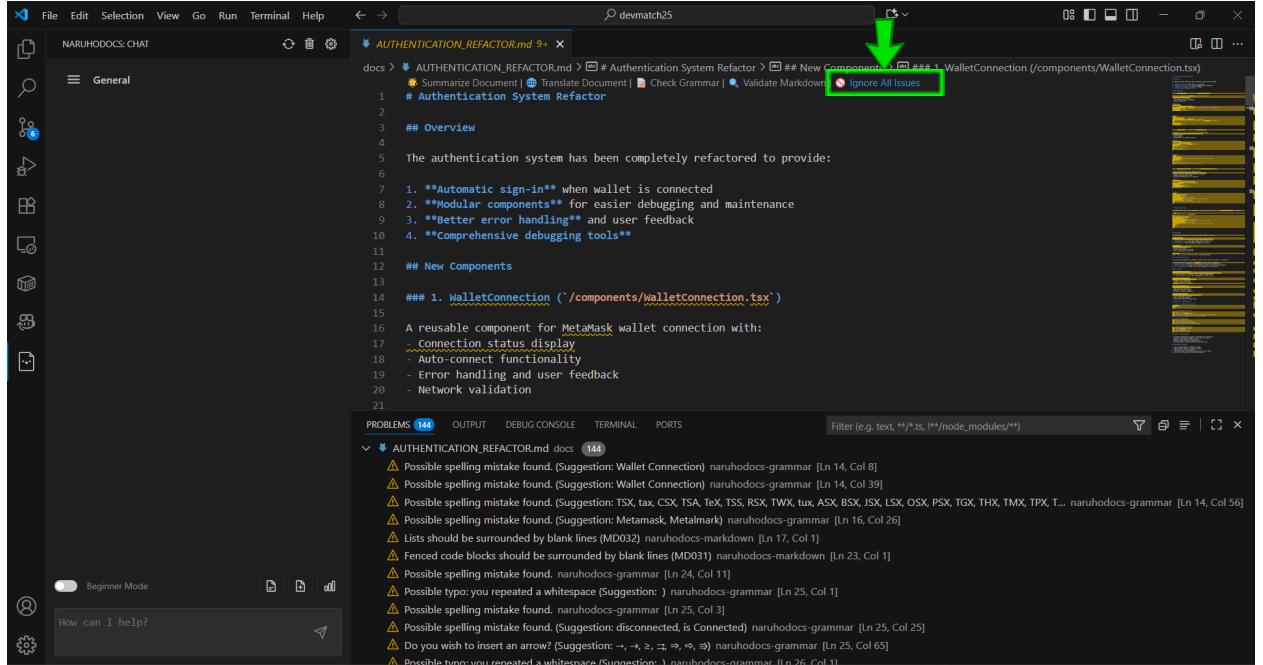
The screenshot shows the VS Code interface with two windows open: 'README.md 1' and 'AUTHENTICATION_REFATOR.md 9+'. The 'AUTHENTICATION_REFATOR.md' window displays several markdown errors, primarily related to list structure. A green arrow points to the 'Ignore all issues of type "MD031"' option in the 'PROBLEMS' view, which is highlighted with a red box. The 'PROBLEMS' view lists multiple errors of type MD031 across the document. After ignoring the type, the screenshot shows the 'PROBLEMS' view again, but the list of errors has been cleared, indicating that all instances of that specific error type have been ignored.



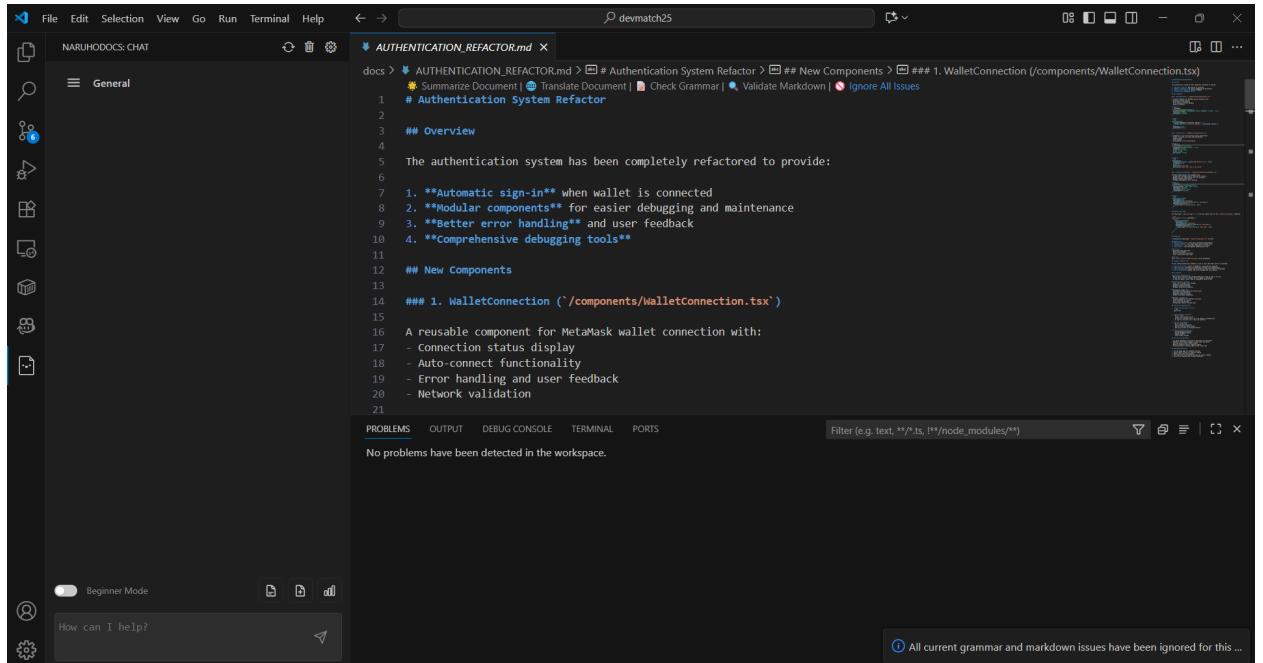
Same type of markdown error has been ignored

5.7.5 Ignore All Issues

- If you believe all underlined words are not actual errors, click the “Ignore All Issues” action button to clear all detected grammar and markdown problems from the document.



The screenshot shows the devmatch25 IDE interface with the file "AUTHENTICATION_REFATOR.md" open. The status bar at the bottom indicates 144 problems. In the top right corner of the code editor, there is a red button labeled "Ignore All Issues". A large green arrow points down to this button, highlighting it. The code editor displays the content of the file, which includes sections like "## Overview" and "## New Components", and a detailed description of a "WalletConnection" component.



The screenshot shows the devmatch25 IDE interface with the file "AUTHENTICATION_REFATOR.md" open. The status bar at the bottom now indicates 0 problems. A message in the bottom right corner states "All current grammar and markdown issues have been ignored for this ...". The code editor displays the same content as the previous screenshot, but the "PROBLEMS" tab is active and shows a message stating "No problems have been detected in the workspace."

All errors have been ignored

6.0 TROUBLESHOOTING

6.1 General Tab Icons Not Visible

- Issue: The icons in the General tab do not appear after opening NaruhoDocs.
- Solution: Click the hamburger menu () and select General to refresh the tab view.

6.2 Newly Opened Document Tab Not Showing

- Issue: A newly opened document tab does not appear in the hamburger menu.
- Solution:
 - Click any existing tab inside the hamburger menu to refresh the tab list. The new document tab should then appear in the list.
 - If the new document tab still does not appear, press Ctrl + Shift + P, search for “NaruhoDocs: Clear All Thread History”, and run the command to restart NaruhoDocs.

6.3 Chatbot Not Responding or Showing an Error Message

- Issue: The AI chatbot does not return any response, or an error message appears.
- Possible Causes:
 - Incorrect or missing API key configuration.
 - API quota for the Cloud LLM has been exhausted.
 - Local LLM model is not installed or not running.
 - Network connection issues (for Cloud LLM).
- Solution:
 - Verify that your LLM provider (Local or Cloud) is correctly selected in the NaruhoDocs settings.
 - If using a Local LLM, ensure the model is properly installed and actively running on your system.
 - If using a Cloud LLM, confirm that your API key is valid and that your usage quota has not been exceeded. Replace the key if necessary.
 - Check your internet connection to ensure stable communication with the Cloud API.
 - Review logs via Output panel → NaruhoDocs LLM for detailed error messages.

6.4 RAG or Embedding Errors

- Issue: “Rebuild RAG Vector Database” fails, or embeddings are not generated.
- Possible Causes:
 - Missing or invalid Hugging Face API key (for Cloud embedding).
 - The local embedding model is not installed or configured properly.
 - Required Python or local dependencies are missing.
 - Cached data corruption or outdated RAG index.
- Solution:
 - Verify your Embedding Mode selection (Local or Cloud) in the settings.
 - If using Cloud Embedding, ensure your Hugging Face API key is valid and active.
 - If using Local Embedding, confirm that the embedding model and dependencies are correctly installed on your system.
 - Try clearing the RAG cache and rebuilding the database using “Rebuild RAG Vector Database.”
 - If the issue persists, review the logs via Output panel → NaruhoDocs LLM for detailed error information.

Still facing issues? Send us feedback [here](#).

7.0 ACKNOWLEDGEMENT

We would like to express our sincere appreciation to everyone who contributed to the development of NaruhoDocs.

This project was made possible through the support of various tools, resources, and collaborative efforts.

Special thanks to:

- Visual Studio Code API, for providing the foundation to build and integrate custom extensions.
- Google AI Studio, Hugging Face and all the open-source LLMs providers, for their powerful language and embedding models that enable intelligent document assistance.
- Mermaid.js, D3.js, and Vis.js, for offering visualization libraries that enhance project understanding.
- Our team members, for their creativity, collaboration, and commitment throughout the hackathon.
- Hackathon organizers and mentors, for their continuous guidance, encouragement, and valuable feedback during the development process.

Together, these contributions helped bring NaruhoDocs to life as an AI-powered documentation assistant within VS Code.

8.0 USER FEEDBACK

We value your input! Please share your feedback, suggestions, or report issues [here](#).