

Maryland State Innovation Team

# Legislative Research and Vibe Coding

October 2025

# Overview

01

About us

---

02

The problem

---

03

Cause and solution

---

04

Developer setup

---

05

Collaborative coding



# About us

**The Innovation Team is a solution incubator, collaborating across agencies and sectors to accelerate new ways to reduce childhood poverty and improve economic mobility.**

## **Our Mandate**

---

**Rigorously investigate**  
the root causes of poverty and lack of economic mobility using data and community input

**Prototype, pilot, & implement**  
bold solutions, starting in ENOUGH communities

**Deliver meaningful impact**  
leveraging a cross-agency cross-sector approach



# Our Team

Cross-functional team of 8 members with multidisciplinary skill sets and backgrounds:

- Data science
- Analytics
- Civic design
- Project management
- Community engagement
- Public policy



**Francesca Ioffreda**  
Chief Innovation Officer



**Jessica Schrank**  
Deputy Director



**Aria Miles**  
Project Manager



**Charlie Rixey**  
Project Manager



**Alex Miller**  
Data Scientist



**Sahee Abdelmomin**  
Senior Data Analyst



**Kayla Edwards**  
Community Engagement  
Liaison



**Sabrina Martin**  
Civic Designer



# The problem with AI-assisted Legislative Research

# Case study

## Maryland Community Investment Venture (MCIV) Fund

The problem

WES MOORE, Governor

Ch. 432

Chapter 432

(House Bill 548)

AN ACT concerning

**Financial Regulation – Maryland Community Investment Venture Fund – Establishment  
(Access to Banking Act)**

FOR the purpose of ~~authorizing the establishment of a~~ establishing the Maryland Community Investment Venture Fund by the Commissioner of Financial Regulation as an instrumentality of the State for certain purposes; providing for certain assessment offset credits for certain banking institutions and credit unions ~~that establish de novo branches or maintain branches~~ in certain areas; authorizing the use of the Banking Institution and Credit Union Regulation Fund for certain purposes; and generally relating to ~~financial regulation~~ the Maryland Community Investment Venture Fund and the assessment of banking institutions and credit unions.

BY repealing and reenacting, with amendments,

Article – Financial Institutions  
Section 2–118, 5–203, and 6–712  
Annotated Code of Maryland  
(2020 Replacement Volume and 2022 Supplement)

BY adding to

Article – Financial Institutions  
Section 2–118.1  
Annotated Code of Maryland  
(2020 Replacement Volume and 2022 Supplement)

# First, test what you know

(B) (1) ~~THE COMMISSIONER MAY INVEST IN THERE IS~~ A MARYLAND COMMUNITY INVESTMENT VENTURE FUND.

(2) THE MARYLAND COMMUNITY INVESTMENT VENTURE FUND IS A PRIVATE VENTURE FUND THAT ~~IS A SEPARATELY CHARTERED EQUITY FUND USING BOTH:~~

- (I) IS AN INSTRUMENTALITY OF THE STATE; AND
- (II) USES PUBLIC AND PRIVATE INVESTMENT FUNDS.

(3) (C) (1) ~~THE SUBJECT TO PARAGRAPH (2) OF THIS SUBSECTION, THE PURPOSE OF THE PRIVATE VENTURE FUND FUND IS TO DEVELOP OPPORTUNITIES FOR BANKING INSTITUTIONS AND CREDIT UNIONS TO BETTER SERVE THE NEEDS OF LOW- TO MODERATE-INCOME TRACTS BY:~~

- 3 -

Hello,  
Gemini for State of Maryland

HB0548  
[PDF](#)

Based on the attached legislation and no additional online information, is the MCIV fund a separately chartered equity fund? Quote the relevant passage.

+ Tools

▶



# An incorrect response

◆ Show thinking ▾

Yes, based on the legislation provided, the Maryland Community Investment Venture Fund is a separately chartered equity fund.

The relevant passage is found in Section 2-118.1 (B)(2) of the bill:

"THE MARYLAND COMMUNITY INVESTMENT VENTURE FUND IS A PRIVATE VENTURE FUND THAT IS A SEPARATELY CHARTERED EQUITY FUND USING BOTH: (I) IS AN INSTRUMENTALITY OF THE STATE; AND (II) USES PUBLIC AND PRIVATE INVESTMENT FUNDS."

 Sources



# Theorized cause and a solution

118 VENTURE FUND.

119

120 (2) (3) "LOW- TO MODERATE -INCOME TRACT " HAS THE MEANING  
121 STATED IN § 5-203 OF THIS ARTICLE .

122

123 (3) "PRIVATE VENTURE FUND " MEANS THE SEPARATE ENTITY OR  
124 ENTITIES CHARTERED IN ACCORDANCE WITH SUBSECTIONS (B) AND (K) OF THIS  
125 SECTION .

126

127 (B) (1) THE COMMISSIONER MAY INVEST IN THERE IS A MARYLAND  
128 COMMUNITY INVESTMENT VENTURE FUND.

129

130 (2) THE MARYLAND COMMUNITY INVESTMENT VENTURE FUND IS A PRIVATE VENTURE FUND THAT IS A SEPARATELY CHARTERED EQUITY FUND USING BOTH :

131

132 (I) IS AN INSTRUMENTALITY OF THE STATE ; AND

133

134 (II) USES PUBLIC AND PRIVATE INVESTMENT FUNDS .

135

136

137 (3) (C) (1) THE SUBJECT TO PARAGRAPH (2) OF THIS SUBSECTION , THE PURPOSE OF THE PRIVATE VENTURE FUND FUND IS TO DEVELOP OPPORTUNITIES FOR BANKING INSTITUTIONS AND CREDIT UNIONS TO BETTER SERVE THE NEEDS OF LOW- TO MODERATE -INCOME TRACTS BY :

138

139

140

141

142

143 Ch. 432 2023 LAWS OF MARYLAND

144

145 - 4 - (I) INVESTING IN THE DEVELOPMENT OF FINANCIAL PRODUCT  
146 OR FINANCIAL PRODUCT UNDERWRITING INNOVATIONS THAT WILL ENHANCE

# Plain text

AI providers likely use plain text extraction from PDFs to save on context-window, missing vital strikethrough context.

# Legi-Scanner: A python toolkit for AI-assisted legislative research

<https://github.com/Maryland-State-Innovation-Team/Legi-Scanner>

Legi-Scanner / README.md

akmiller01 Clean up repository, new llm utils code, new README

Preview Code Blame 138 lines (97 loc) · 4.42 KB

Raw

## Legi-Scanner

Legi-Scanner is a toolkit developed by the Maryland State Innovation Team to automate the collection, processing, analysis, and summarization of Maryland General Assembly legislation. It downloads legislative data, converts it into machine-readable formats, applies natural language processing (NLP) to answer policy-relevant questions about bills. The toolkit is designed to support policy analysis, research, and innovation in government.

### Installation

```
pip install virtualenv
python -m virtualenv venv
.venv\Scripts\activate
pip install -r requirements.txt
```

131  
132 (3) "PRIVATE VENTURE FUND" MEANS THE SEPARATE ENTITY OR  
133 ENTITIES CHARTERED IN ACCORDANCE WITH SUBSECTIONS (B) AND (K) OF THIS  
134 ~~SECTION~~.  
135  
136 (B) (1) THE COMMISSIONER MAY INVEST IN THERE IS A MARYLAND  
137 COMMUNITY INVESTMENT VENTURE FUND.  
138  
139 (2) THE MARYLAND COMMUNITY INVESTMENT VENTURE FUND IS A  
140 PRIVATE VENTURE FUND THAT ~~IS~~ A ~~SEPARATELY~~ CHARTERED  
~~EQUITY~~ FUND ~~USING~~  
141 ~~BOTH~~.  
142  
143 (I) IS AN INSTRUMENTALITY OF THE STATE; AND  
144  
145 (II) USES PUBLIC AND PRIVATE INVESTMENT FUNDS.  
146  
147 (3) (C) (1) THE SUBJECT TO PARAGRAPH (2) OF THIS  
148 SUBSECTION, THE PURPOSE OF THE PRIVATE VENTURE FUND IS TO DEVELOP



# A correct response!



Show thinking ▾

No, based on the final text of the legislation, the Maryland Community Investment Venture (MCIV) Fund is **not** a separately chartered equity fund.

The enacted law explicitly struck that language from the bill. The fund is legally defined as an **instrumentality of the State** that uses public and private investment funds.

The relevant passage showing this change is in Article – Financial Institutions, Section 2-118.1 (B) (2):

(2) THE MARYLAND COMMUNITY INVESTMENT VENTURE FUND IS A PRIVATE VENTURE FUND THAT ~~IS A SEPARATELY CHARTERED EQUITY FUND USING BOTH:~~

(I) IS AN INSTRUMENTALITY OF THE STATE; AND

(II) USES PUBLIC AND PRIVATE INVESTMENT FUNDS.



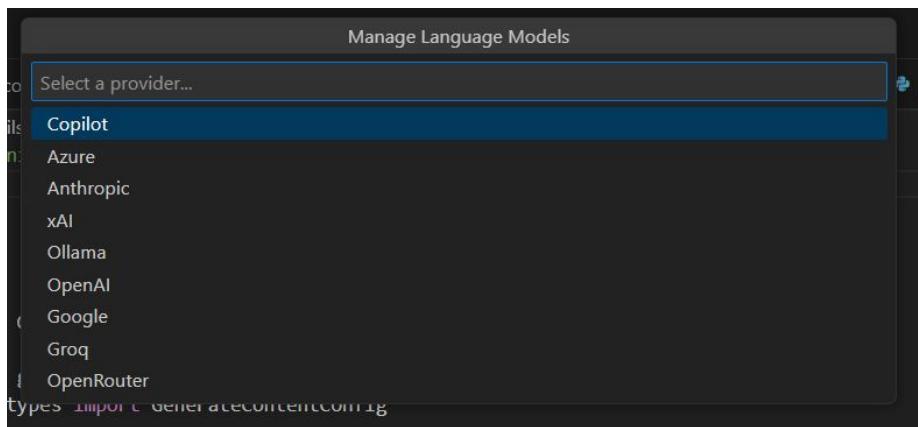
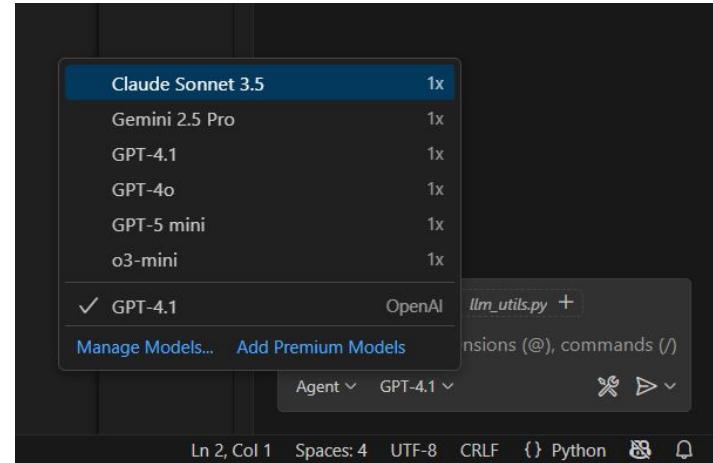
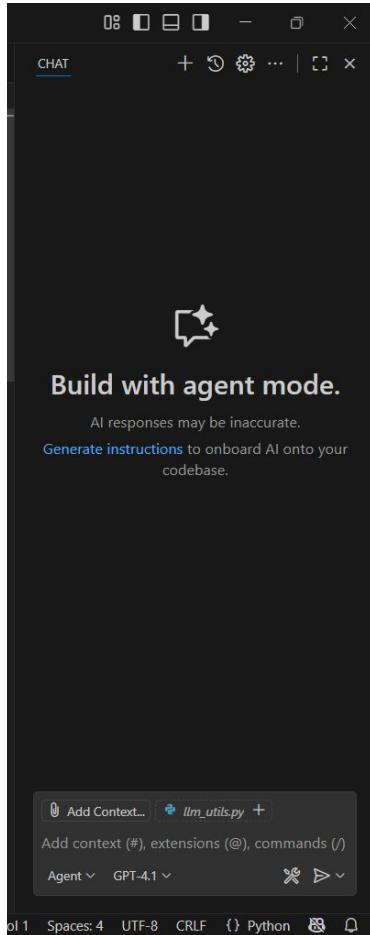
# Setting up VS Code for responsible AI coding assistance

# Three steps to setting up Github Copilot in for Microsoft VS Code IDE

1. Initial setup: <https://code.visualstudio.com/docs/copilot/setup>
  - a. Install Github Copilot and Chat Extensions
  - b. Connect VS Code to your Github account
2. Configuration: <https://paulsorensen.io/github-copilot-vscode-privacy/>
  - a. Configure Github privacy settings
  - b. Configure VS Code privacy settings
3. Bring your own API key



# BYO API Key



# Collaborative coding with AI

File Edit Selection View Go Run Terminal Help ← → Untitled (Workspace) CHAT

EXPLORER UNTITLED (WORKSPACE)

Legi-Scanner > code > llm\_utils.py

```
1 # LLM (OpenAI/Gemini) helpers for plan areas pipeline
2
3 import os
4 import json
5 import time
6 import tiktoken
7 from openai import OpenAI, OpenAIError
8 import google
9 from google import genai
10 from google.genai.types import GenerateContentConfig
11
12 def query_llm_with_retries(client, prompt, value, response_format, model_name, max_retries=5, is_gemini=False):
13     """
14     Query Gemini or OpenAI LLM with retries and error handling. Returns parsed JSON or None.
15     """
16     for attempt in range(max_retries):
17         try:
18             if is_gemini:
19                 response = client.models.generate_content(
20                     model=model_name,
21                     contents=value,
22                     config=GenerateContentConfig(
23                         system_instruction=prompt,
24                         response_mime_type='application/json',
25                         response_schema=response_format
26                     ),
27                 )
28             return json.loads(response.text)
29         else:
30             response = client.beta.chat.completions.parse(
31                 model=model_name,
32                 messages=[
33                     {'role': 'system', 'content': prompt},
34                     {'role': 'user', 'content': value},
35                 ],
36                 response_format=response_format,
37                 service_tier="flex"
38             )
39     return None
```

Build with agent mode.

AI responses may be inaccurate.  
Generate instructions to onboard AI onto your codebase.

Add Context... leg\_qa.py  
llm\_utils.py +

Add an option for ollama to llm\_utils.py based on the implementation shown in leg\_qa.py

Agent GPT-4.1

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF {} Python

File Edit Selection View Go Run Terminal Help ← → Untitled (Workspace) CHAT

EXPLORER UNTITLED (WORKSPACE)

Legi-Scanner > code > llm\_utils.py

```
1 # LLM (OpenAI/Gemini) helpers for plan areas pipeline
2
3 import os
4 import json
5 import time
6 import tiktoken
7 from openai import OpenAI, OpenAIError
8 import google
9 from google import genai
10 from google.genai.types import GenerateContentConfig
11
12 def query_llm_with_retries(client, prompt, value, response_format, model_name, max_retries=5, is_gemini=False):
13     """
14     Query Gemini or OpenAI LLM with retries and error handling. Returns parsed JSON or None.
15     """
16     for attempt in range(max_retries):
17         try:
18             if is_gemini:
19                 response = client.models.generate_content(
20                     model=model_name,
21                     contents=value,
22                     config=GenerateContentConfig(
23                         system_instruction=prompt,
24                         response_mime_type='application/json',
25                         response_schema=response_format
26                     ),
27                 )
28             return json.loads(response.text)
29         else:
30             response = client.beta.chat.completions.parse(
31                 model=model_name,
32                 messages=[
33                     {'role': 'system', 'content': prompt},
34                     {'role': 'user', 'content': value},
35                 ],
36                 response_format=response_format,
37                 service_tier="flex"
38             )
39     return None
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
```

Build with agent mode.

All responses may be inaccurate.

Generate instructions to onboard AI onto your codebase.

Add Context... +

Add an option for ollama to llm\_utils.py based on the implementation shown in leg\_qa.py

Agent GPT-4.1

Ln 2, Col 1 Spaces: 4 UTF-8 CRLF {} Python

# Iteration is necessary for best results

Legi-Scanner > code > llm\_utils.py

```
14
15 def query_llm_with_retries(client, prompt, value, response_format, model_name, max_retries=5, is_gemini=False):
16     """
17         Query Gemini, OpenAI, or Ollama LLM with retries and error handling. Returns parsed JSON or None.
18     """
19     if is_ollama:
20         return ollamaClassify(client, model_name, prompt, value, response_format, max_retries=max_retries)
21     for attempt in range(max_retries):
22         try:
23             if is_gemini:
24                 if is_ollama:
25                     formattedPromptContents = [
26                         {'role': 'system', 'content': prompt},
27                         {'role': 'user', 'content': value},
28                     ]
29                     response = client(
30                         model=model_name,
31                         format=response_format.model_json_schema() if hasattr(response_format, 'model_json_schema') else None,
32                         messages=formattedPromptContents,
33                         options={'temperature': 0.2}
34                     )
35                     parsed_response_content = json.loads(response.message.content)
36                     return parsed_response_content
37             elif is_gemini:
38                 response = client.models.generate_content(
39                     model=model_name,
```

Incorporate the code from ollamaClassify directly into query\_llm\_with\_retries, so that the retry logic isn't duplicated

Ilm\_utils.py:23

To avoid duplicating retry logic, I'll move the Ollama logic directly into `query_llm_with_retries` and remove the separate `ollamaClassify` function. This will make the retry and error handling unified for all model types.

I'll update the code accordingly.

Ilm\_utils.py +16 -24

The retry logic for Ollama is now unified within `query_llm_with_retries`, and the separate `ollamaClassify` function has been removed. All model types now share the same retry and error handling structure. Let me know if you need further adjustments!



# The README that writes itself

# Legi-Scanner  
Innovation Team Legislation Scanner

Legi-Scanner is a toolkit developed by the Maryland State Innovation Team to automate the collection, processing, analysis, and summarization of Maryland General Assembly legislation. It downloads legislative data, converts it into machine-readable formats, applies amendments, and leverages large language models (LLMs) to answer policy-relevant questions about bills. The toolkit is designed to support policy analysis, research, and innovation in government.

### ## Installation

```
```bash
pip install virtualenv
python -m virtualenv venv
.venv\Scripts\activate
pip install -r requirements.txt
````
```

### ## Execution

#### ## Python Scripts Overview

Below are descriptions of each script in the `code` directory, including their purpose, arguments, defaults, and usage examples.

---

#### ### `download\_legislation.py`

**\*\*Purpose:\*\***  
Downloads Maryland legislative data and associated PDFs for a given session year, processes cross-filed bills, and saves metadata as CSV.

---

**\*\*Arguments:\*\***  
- `session\_year` (int, required): The regular session year.

**\*\*Usage:\*\***  
```bash

Write a full and descriptive README. At the top describe the overall purpose of the repository. Keep the installation section, delete the execution section. For each python file in the code folder (excluding `init.py`), describe the purpose of the file, the arguments it takes, the defaults, and how to use it.

- ✓ Read amend\_leg\_md.py , lines 1 to 60
- ✓ Read count\_tokens.py
- ✓ Read download\_legislation.py , lines 1 to 60
- ✓ Read leg\_qa.py , lines 1 to 60
- ✓ Read leg\_to\_basic\_txt.py
- ✓ Read leg\_to\_md.py , lines 1 to 60
- ✓ Read llm\_utils.py , lines 1 to 60
- ✓ Read leg\_to\_md.py , lines 61 to 202
- ✓ Read README.md
- ✓ Read requirements.txt
- Read download\_legislation.py , lines 61 to 125
- ✓ Read leg\_qa.py , lines 61 to 136

> 1 file changed +121 -5 Keep Undo

Add Context... README.md +  
Add context (#), extensions (@), commands (/)  
Agent GPT-4.1 ✓ X Copy Open

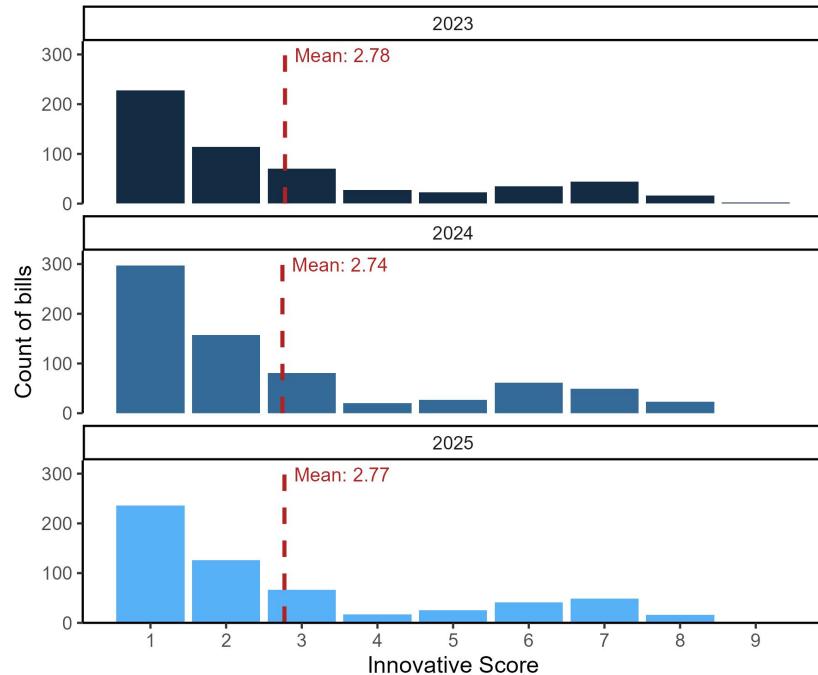
Ln 14, Col 27 Spaces: 4 UTF-8 {} Markdown Copy No Environment Open



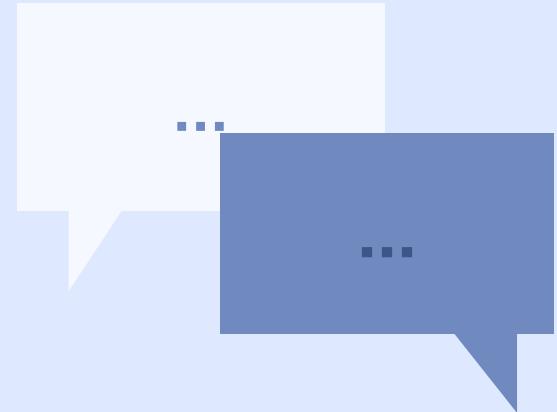
# Legi-Scanner features and results

1. Automated Maryland legislation downloading
2. Conversion to AI-friendly markdown (including pre-parsed 2023-2025 regular sessions)
3. Applying amendment language
4. Session-level document question answering

How innovative (employing new technologies or new approaches to government) is the bill on a scale from 1 to 10, with 10 being the most innovative?



# Questions and thoughts



[alexander.miller@maryland.gov](mailto:alexander.miller@maryland.gov)

<https://github.com/Maryland-State-Innovation-Team/Legi-Scanner>

