

A)

```
Help ← → nikhilkhanchandani
.env x
LeadGenerationResearch > .env
1 GOOGLE_CLOUD_PROJECT="nikhil-ai-agents-19137"
2 GOOGLE_CLOUD_LOCATION="us-central1"
3
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
python + - ×

Error: Invalid value for 'AGENT': Directory 'LeadGenerationResearch/LeadGenerationResearch' does not exist.
nikhilkhanchandani@cloudshell:/LeadGenerationResearch (nikhil-ai-agents-19137)$ # Option B (some repos wire the package root differently)
poetry run adk run LeadGenerationResearch
Log setup complete: /tmp/agents_log/agent.20251109_211135.log
To access latest log: tail -F /tmp/agents_log/agent.latest.log
/home/nikhilkhanchandani/.cache/pypoetry/virtualenvs/leadgenerationresearch-Iowux8hR-py3.12/lib/python3.12/site-packages/google/adk/cli/cli.py:134: UserWarning: [EXPERIMENTAL]
InMemoryCredentialService: This feature is experimental and may change or be removed in future versions without notice. It may introduce breaking changes at any time.
    credential_service = InMemoryCredentialService()
/home/nikhilkhanchandani/.cache/pypoetry/virtualenvs/leadgenerationresearch-Iowux8hR-py3.12/lib/python3.12/site-packages/google/adk/auth/credential_service/in_memory_credential_service.py:33: UserWarning: [EXPERIMENTAL] BaseCredentialService: This feature is experimental and may change or be removed in future versions without notice. It may introduce breaking changes at any time.
    super().__init__()
Running agent InteractiveLeadGenerator, type exit to exit.
[user]: Find fintech leads in Thailand
[InteractiveLeadGenerator]: How many successful companies in your target market would you like to analyze to identify investment patterns?
[user]: 5

[InteractiveLeadGenerator]: I'm sorry, but I was unable to identify any investment patterns from the companies I analyzed. This is because all five of the companies I looked at were not suitable for this analysis. They were either local Thai companies, the investment information was outdated, or the sources could not be verified.

Would you like me to try again?

[user]: [user]: Proceed with 5 companies for discovery, then find 5 leads
[InteractiveLeadGenerator]: I am sorry, but I was unable to identify any investment patterns in the top 5 successful fintech companies in Thailand. This is because all the companies I analyzed were local Thai companies, and I need to analyze foreign companies to find investment patterns.

Would you like me to try again with a different number of companies? Or would you like to change the country or industry?

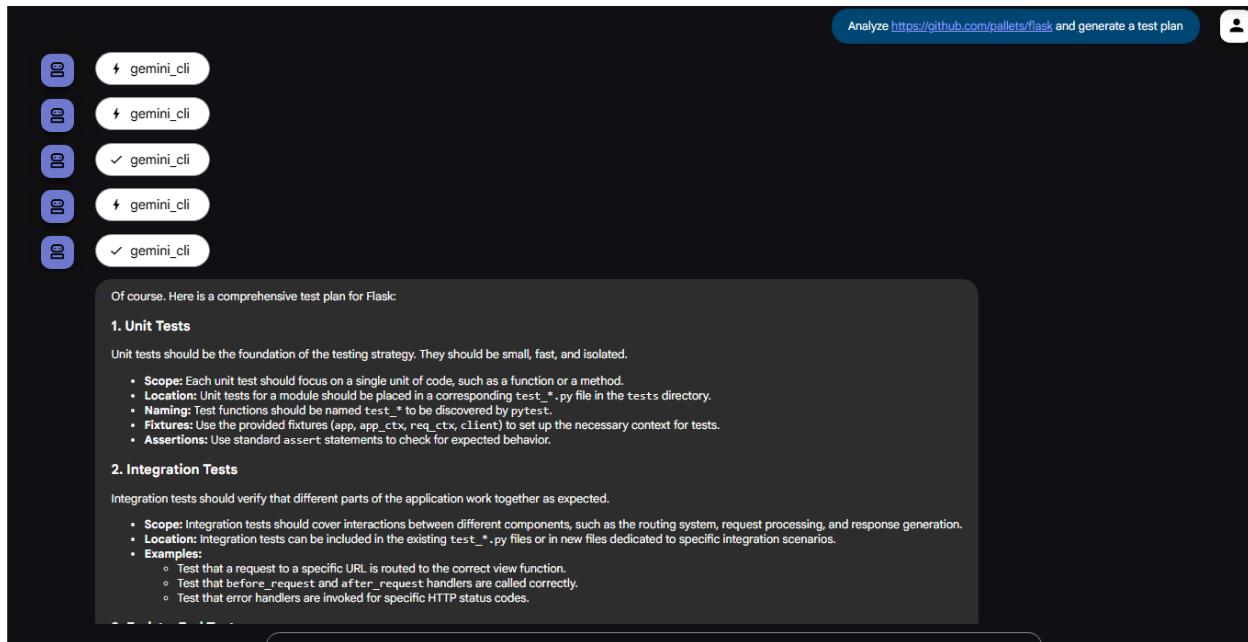
[user]: Please try analyzing 5 foreign fintech companies
[InteractiveLeadGenerator]: I was able to find one company that matched your criteria, Siam Validus. While I can't identify a pattern from a single company, I can share some of the key activities Siam Validus took before their investment, which could be helpful indicators for potential leads.

Before their investment in April 2023, Siam Validus:
1. Secured significant capital and financial partnerships.
2. Grew through strategic acquisitions to expand their capabilities and assets.
3. Laid the foundational regulatory and market groundwork in Thailand.

Would you like me to find leads based on these activities, or would you like to start over?
[user]:
```

B)

Analyze <https://github.com/pallets/flask> and generate a test plan 



Of course. Here is a comprehensive test plan for Flask:

1. Unit Tests

Unit tests should be the foundation of the testing strategy. They should be small, fast, and isolated.

- **Scope:** Each unit test should focus on a single unit of code, such as a function or a method.
- **Location:** Unit tests for a module should be placed in a corresponding `test_*.py` file in the `tests` directory.
- **Naming:** Test functions should be named `test_*` to be discovered by `pytest`.
- **Fixtures:** Use the provided fixtures (`app`, `app_ctx`, `req_ctx`, `client`) to set up the necessary context for tests.
- **Assertions:** Use standard `assert` statements to check for expected behavior.

2. Integration Tests

Integration tests should verify that different parts of the application work together as expected.

- **Scope:** Integration tests should cover interactions between different components, such as the routing system, request processing, and response generation.
- **Location:** Integration tests can be included in the existing `test_*.py` files or in new files dedicated to specific integration scenarios.
- **Examples:**
 - Test that a request to a specific URL is routed to the correct view function.
 - Test that `before_request` and `after_request` handlers are called correctly.
 - Test that error handlers are invoked for specific HTTP status codes.

3. End-to-End Tests

End-to-end tests should simulate real user scenarios and verify the application's behavior from the user's perspective.

- **Scope:** End-to-end tests should cover complete workflows, such as user authentication, form submission, and API interactions.
- **Location:** End-to-end tests can be placed in a separate directory, such as `tests/e2e`.
- **Tools:** The existing `client` fixture can be used for end-to-end testing.
- **Examples:**
 - Test that a user can register, log in, and log out.
 - Test that a user can submit a form and the data is processed correctly.

4. Regression Tests

Regression tests should be added for every bug fix to prevent the bug from reoccurring.

- **Scope:** Each regression test should specifically target the bug it is fixing.
- **Location:** Regression tests should be added to the `tests/test_regression.py` file.

5. Running Tests

Tests should be run using the following command:

```
tox
```

This will run the full test suite against all supported Python versions. To run tests for a specific Python version, use:

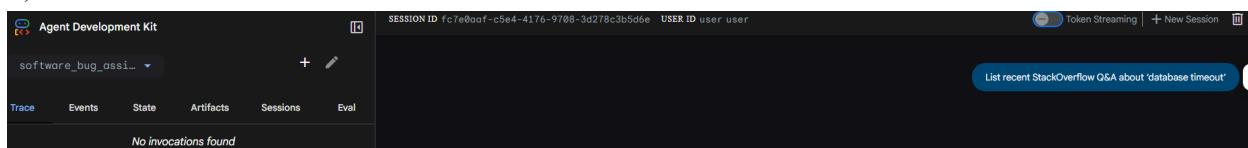
```
tox run -e py<python_version>
```

For example, to run tests for Python 3.11:

```
tox run -e py311
```

This test plan provides a comprehensive approach to testing the Flask project, ensuring code quality and stability.

C)



Agent Development Kit

SESSION ID fc7e00af-c5e4-4176-9708-3d278c3b5d66 USER ID user user

Token Streaming | + New Session

software_bug_assi... ▾

No invocations found

List recent StackOverflow Q&A about 'database timeout'

D)

```
    return False
[CodeAnalyzer]: I'm sorry, but there was a syntax error in the code you provided, so I couldn't fully analyze its structure.

**Syntax Error Detected:**
*  **Location:** Line 1
*  **Message:** "expected an indented block after function definition on line 1"

This means that the function definition `def dfs_search_v1(graph, start, target):` is incomplete because it's missing an indented code block (the function's body).

Please provide the complete and syntactically correct Python code for analysis.
[StyleChecker]: ## Style Analysis Results
- Style Score: 98/100
- Total Issues: 1
- Assessment: Excellent style compliance!

## Top Style Issues
- Line 1, Col 41: W292 no newline at end of file
```

E)

