gemini-code-assist bot left a comment

Code Review

The pull request introduces a new research agent backend with `/discover` and `/synthesize` endpoints, integrating with Tavily for search, Gemini for LLM interactions, Crawl4AI for web crawling, and Firebase Firestore for audit logging. The changes include adding necessary environment variables, updating configuration, and including the new API routes. Unit tests with mocked dependencies are also provided. Overall, the implementation covers the specified features well, but there are several areas for improvement regarding error handling, configuration flexibility, and data privacy in logging. All original comments remain valid and are included, with additional context provided for logging-related feedback based on repository rules.

☺

dgs-backend/app/api/routes/research.py Outdated Show resolved

dgs-backend/app/api/routes/research.py Outdated

Comment on lines 121 to 124

```
121 +     except Exception as e:
122 +         logger.error(f"Crawling failed: {e}")
123 +         # Continue with whatever we have or fail?
124 +         pass
```

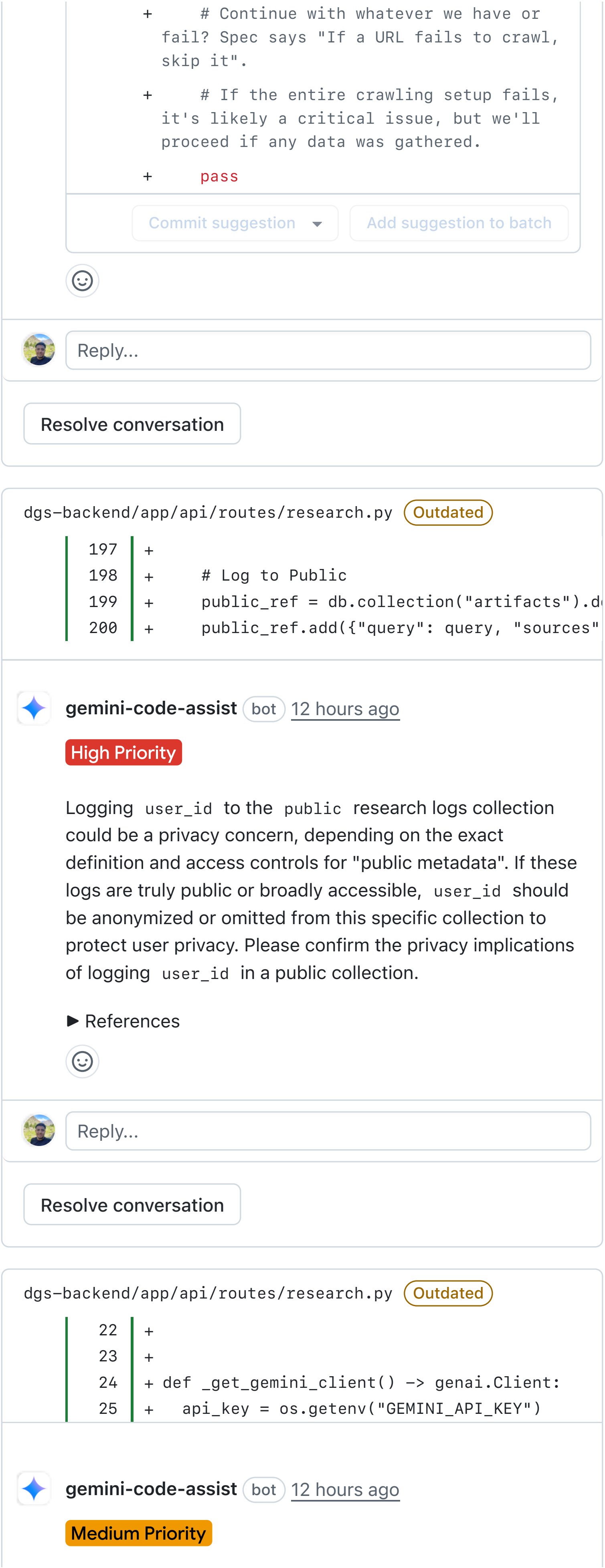
gemini-code-assist bot 12 hours ago

High Priority

The `except Exception as e: pass` block for the entire crawling phase is too broad and can hide critical issues beyond individual URL failures. While the intent is to skip failed URLs, a failure in the `AsyncWebCrawler` initialization or `asyncio.gather` itself would be silently ignored. It's better to catch more specific exceptions or at least log the exception at an `error` level before passing, as is done for individual URL failures.

Suggested change

```
-     except Exception as e:
-         logger.error(f"Crawling failed: {e}")
-         # Continue with whatever we have or fail? Spec says "If a URL fails to crawl, skip it".
-         pass
+     except Exception as e:
+         logger.error(f"Critical crawling phase failed: {e}")
```



```
+     # Continue with whatever we have or fail? Spec says "If a URL fails to crawl, skip it".
+     # If the entire crawling setup fails, it's likely a critical issue, but we'll proceed if any data was gathered.
+     pass
```

dgs-backend/app/api/routes/research.py Outdated

```
+
+     # Log to Public
+     public_ref = db.collection("artifacts").d
+     public_ref.add({"query": query, "sources"
```

gemini-code-assist bot 12 hours ago

High Priority

Logging `user_id` to the `public` research logs collection could be a privacy concern, depending on the exact definition and access controls for "public metadata". If these logs are truly public or broadly accessible, `user_id` should be anonymized or omitted from this specific collection to protect user privacy. Please confirm the privacy implications of logging `user_id` in a public collection.

▶ References

dgs-backend/app/api/routes/research.py Outdated

```
+
+
+ def _get_gemini_client() -> genai.Client:
+   api_key = os.getenv("GEMINI_API_KEY")
```

gemini-code-assist bot 12 hours ago

Medium Priority

For consistency with `_get_tavily_client`, it would be better to retrieve the `GEMINI_API_KEY` from the `settings` object rather than directly from `os.getenv`.

Suggested change

```
-    api_key = os.getenv("GEMINI_API_KEY")
+    api_key = settings.gemini_api_key
```

Commit suggestion ▾   Add suggestion to batch

Reply...

Resolve conversation

dgs-backend/app/api/routes/research.py Outdated

```
41 +
42 + Query: {query}"""
43 +   try:
44 +     response = await client.aio.models.genera
```

gemini-code-assist bot 12 hours ago

Medium Priority

The LLM model name `gemini-2.5-flash` is hardcoded here. It would be more flexible to make this configurable via environment variables and the `Settings` object, similar to how other LLM models are configured in the orchestrator.

Reply...

Resolve conversation

dgs-backend/app/api/routes/research.py Outdated

Comment on lines 47 to 50

```
47 +     for cat in ["General", "Academic", "Secur
48 +       if cat.lower() in category.lower():
49 +         return cat
50 +     return "General"
```

gemini-code-assist bot 12 hours ago

Medium Priority

The fallback logic for query classification is a bit fragile. If the model returns a phrase like "This is a General category query", it will still match "General". Consider using a more robust parsing mechanism (e.g., regex) or refining the prompt to ensure the model returns *only* the category name as requested, to avoid unexpected classifications.

Reply...

Resolve conversation

dgs-backend/app/api/routes/research.py Outdated

```
+   # 3. Search Engine
+   try:
+     # Tavily client is sync, run in threadpoo
+     search_result = await run_in_threadpool(t
```

gemini-code-assist bot 12 hours ago

Medium Priority

The `max_results` parameter for the Tavily search is hardcoded to `5`. This value could be made configurable through the application settings to allow for easier adjustments based on performance or desired search depth.

Open **Implement Research Agent Backend** #32

andesec wants to merge 15 commits into `rbac` from `research-agent-…`

Resolve conversation

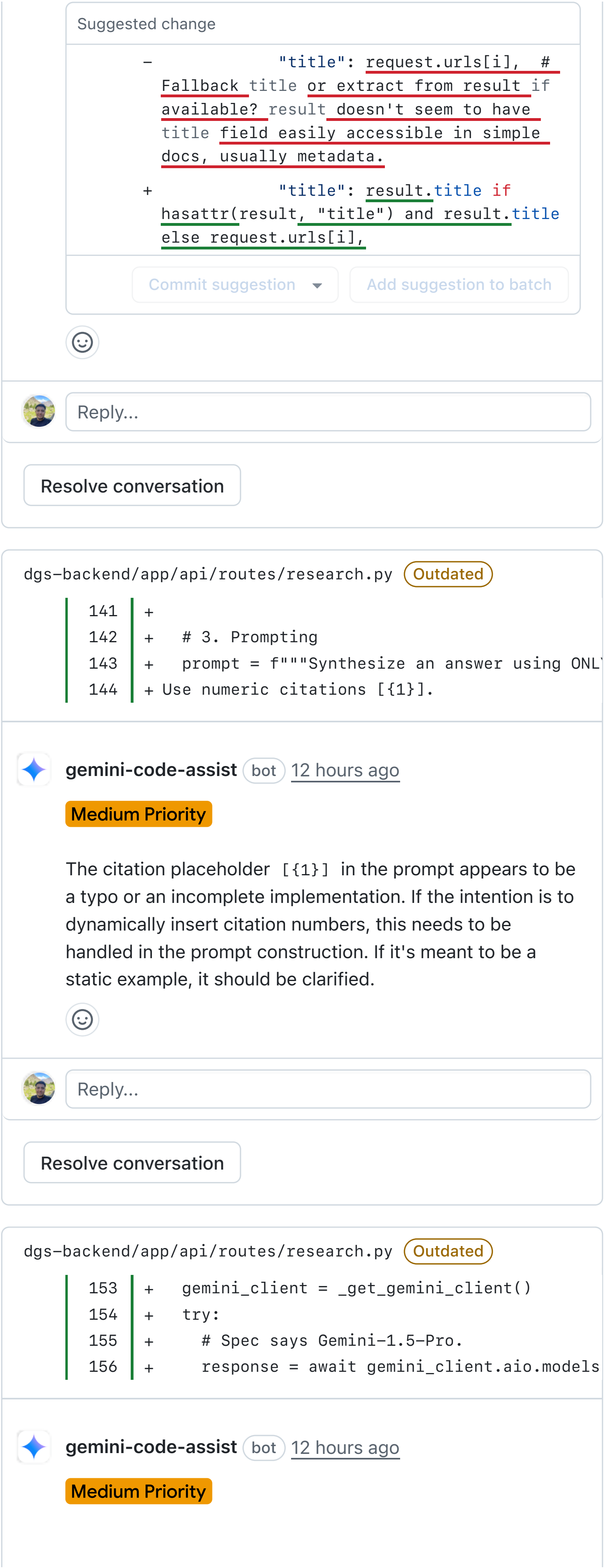
dgs-backend/app/api/routes/research.py Outdated

```
+           {
+             "url": request.urls[i],
+             "markdown": result.markdown,
+             "title": request.urls[i],  # Fall
```

gemini-code-assist bot 12 hours ago

Medium Priority

Using the URL as a fallback title for crawled data is functional but not very user-friendly. The `crawl4ai` result object often contains metadata, including the page title. It would be better to extract the actual page title from `result.metadata` or `result.title` if available, and only fall back to the URL if no title can be found.



Suggested change

```
-            "title": request.urls[i],  # Fallback title or extract from result if available? result doesn't seem to have title field easily accessible in simple docs, usually metadata.
+            "title": result.title if hasattr(result, "title") and result.title else request.urls[i],
```

Reply...

Resolve conversation

dgs-backend/app/api/routes/research.py Outdated

```
+
+   # 3. Prompting
+   prompt = f"""Synthesize an answer using ONL
+ Use numeric citations [{1}].
```

gemini-code-assist bot 12 hours ago

Medium Priority

The citation placeholder `[{1}]` in the prompt appears to be a typo or an incomplete implementation. If the intention is to dynamically insert citation numbers, this needs to be handled in the prompt construction. If it's meant to be a static example, it should be clarified.

Reply...

Resolve conversation

dgs-backend/app/api/routes/research.py Outdated

```
+   gemini_client = _get_gemini_client()
+   try:
+     # Spec says Gemini-1.5-Pro.
+     response = await gemini_client.aio.models
```

gemini-code-assist bot 12 hours ago

Medium Priority

Similar to the classification model, the synthesis LLM model `gemini-1.5-pro` is hardcoded. This should ideally be configurable through the application settings to allow for flexibility in model choice and future upgrades.

Reply...

Resolve conversation

`dgs-backend/app/api/routes/research.py` Outdated

```
+     # Public Metadata
+     # /artifacts/{appId}/public/data/research_
+     # Assuming appId is 'dgs' or similar. Spe
+     app_id = "dgs"  # Default
```

gemini-code-assist bot 12 hours ago

Medium Priority

The `app_id` is hardcoded to "dgs". This value should ideally be retrieved from the application settings (`settings.app_id` or similar) to ensure consistency and configurability across different deployments or environments.

Suggested change

```
-     app_id = "dgs"  # Default
+     app_id = settings.app_id # Assuming app_id is added to settings
```

Commit suggestion ▾ Add suggestion to batch

Reply...

Resolve conversation

google-labs-jules bot and others added 14 commits 11 hours ago

- `feat: refactor research agent with separate provider and agent classes` … ✕ 5273e17
- `feat: enhance research agent logging and config` … ✕ d20ec18
- `fix: set DGS_ALLOWED_ORIGINS in pr-migration-check workflow` … ✕ dd3169d