

Team 🥐 Scrum-
ptious 🥐

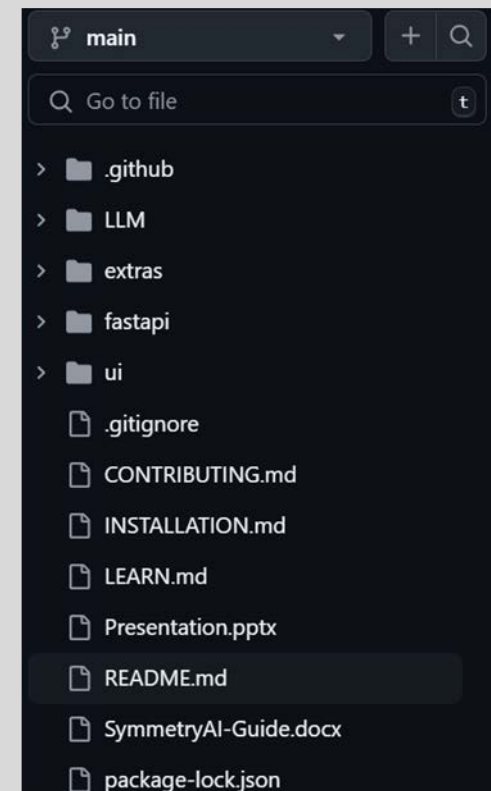
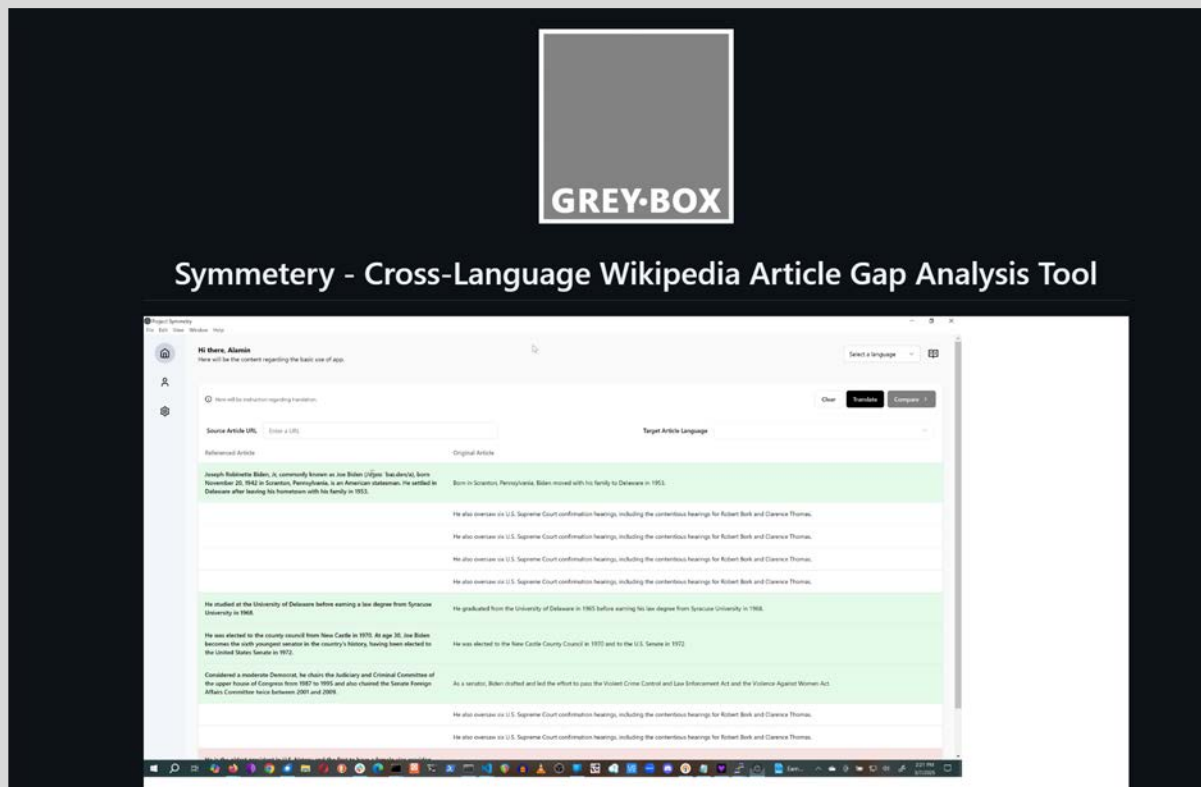
Final Demo



GREY·BOX

Sprint 1

Familiarization



JSON

VS

XML

Consistent with previous infrastructure	Not developer friendly
Smaller Data Payloads	Expensive (tree traversal)
Less resources to parse	Can find hyperlinks easily
Flexible while scaling	Inefficient with multiple models

Sprint 2

Flexible Queries & Error-Handling

s?query=https://en.wikipedia.org/wiki/COVID-19

"detail": "Invalid language code in query parameter: zz .",

OR

"detail": "URL is not a valid Wikipedia request.",

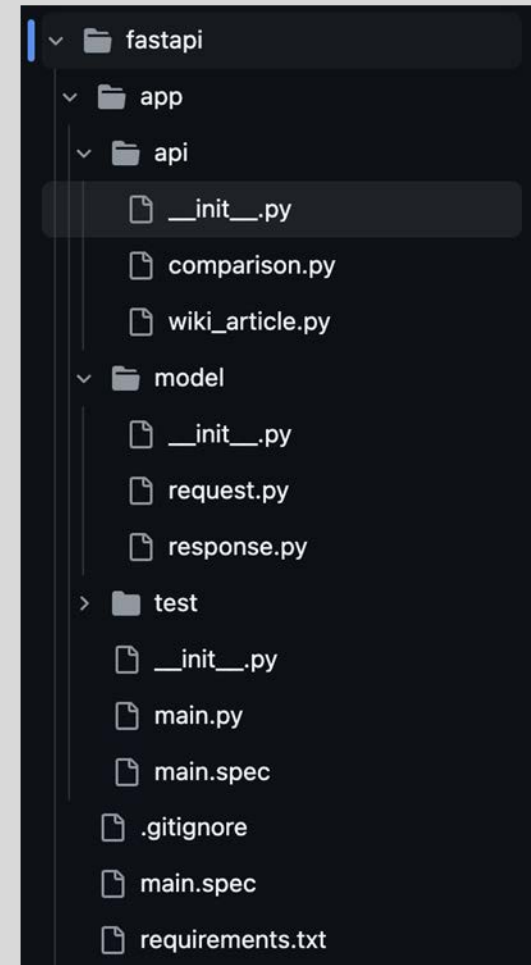
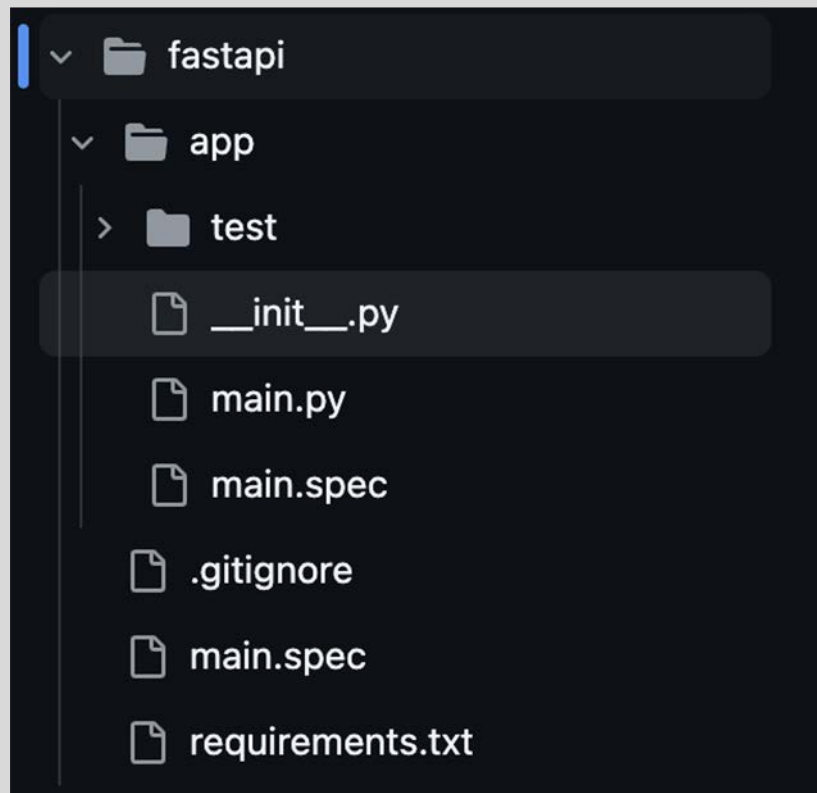
?query=COVID-19&lang=en|

"detail": "Article not found.",

Backend → Frontend Schema

```
{
  "comparisons": [
    {
      "left": "Hello! This is an article.",
      "right": "",
      "highlight": "missing"
    },
    {
      "left": "This information is in both articles. It will display with no highlight.",
      "right": "This information is in both pages. It will display without highlighting.",
      "highlight": "false"
    },
    {
      "left": "",
      "right": "This information is in the second article only.",
      "highlight": "extra"
    }
  ]
}
```

Modular Codespace



Sprint 3

Schema Finalization

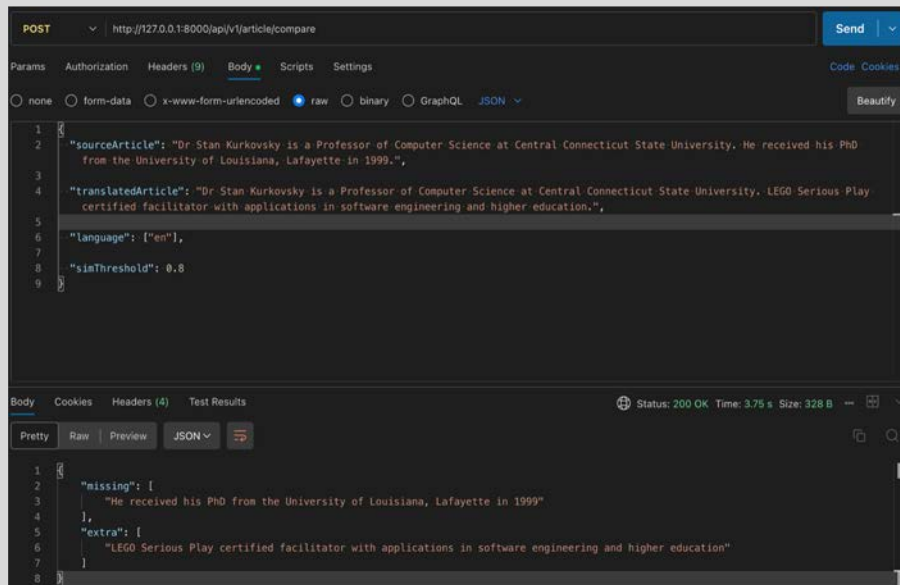
```
{
  "comparisons": [
    {
      "left_article_array": [
        "This is the first sentence",
        "this is the second sentence",
        "this is the third sentence"
      ],
      "right_article_array": [
        "This is the first sentence in a diff language",
        "this is the second sentence in a diff language",
        "this is the third sentence in a diff language"
      ],
      "left_article_missing_info_index": [1, 3, 5, 7, 9, 12, 15, 16, 20],
      "right_article_extra_info_index": [5, 9, 20, 43, 759, 953]
    }
  ]
}
```

Backend → Frontend

```
{
  "article_text_blob_1": "string",
  "article_text_blob_2": "string",
  "article_text_blob_1_language": "string",
  "article_text_blob_2_language": "string",
  "comparison_threshold": 0,
  "model_name": "string"
}
```

Frontend → Backend

Endpoint Configuration & Refactoring



```
# Initialize the router for wiki related endpoints
router = APIRouter(prefix="/symmetry/v1/wiki")

# This caches short language codes existing on Wikipedia.
language_cache: Dict[str, bool] = {}

# GET request method with input validation
@router.get("/articles", response_model=SourceArticleResponse)
```

Debugging Tools & Input Validation

```
1
2
3
4
"detail": "Article not found.",
"stack_trace": "Traceback (most recent call last):\n File \"venv/lib/python3.13/site-packages/starlette/_exception_handler.py\", line 48, in\n receive, sender)\n File \""/Users/erik/Documents/greybox/Project-Symmetry-AI/fastapi/venv/lib/python3.13/site-packages/starlette/routing.py\", line 73, in app\n ~~~~~~\n File \""/Users/erik/Documents/greybox/Project-Symmetry-AI/fastapi/venv/lib/python3.13/site-packages/fastapi/routing.py\", line 301, in app\n (\n ~~~~~~\n ~~~~~~\n Documents/greybox/Project-Symmetry-AI/fastapi/venv/lib/python3.13/site-packages/starlette/_exception_handler.py\", line 48, in\n run_endpoint_function\n return await dependant.call(*args, **kwargs)\n File \""/Users/erik/Documents/greybox/Project-Symmetry-AI/fastapi/venv/lib/python3.13/site-packages/fastapi/routing.py\", line 207, in\n get_article\n raise HTTPException(status_code=404, detail='Article not found.')\n HTTPException: 404: Article not found.\n"
```

```
async def validate_url(url):
    """
    This method is used to validate a Wikipedia article URL.
    The URL should match the general format <language_code>.wikipedia.org/wiki/<article_title>
    It returns the language code included in the URL.
    """

    parsed_url = urlparse(url)

    if not parsed_url.netloc.endswith(".wikipedia.org"):
        logging.info(
            "Invalid domain '%s', only 'wikipedia.org' is allowed.",
            parsed_url.netloc,
        )
        raise HTTPException(status_code=400, detail="Invalid Wikipedia URL.")

    split_url = parsed_url.netloc.split(".")

    if len(split_url) != 3:
        logging.info(
            "Invalid subdomain '%s', only '___wikipedia.org' is allowed.",
            parsed_url.netloc,
        )
        raise HTTPException(status_code=400, detail="Invalid Wikipedia URL.")

    lang = split_url[0]
    if not lang.isalpha() or len(lang) > 2:
        logging.info("Invalid language code '%s'", lang)
        raise HTTPException(status_code=400, detail="Invalid language code in URL.")

    await validate_language_code(lang)

    if not parsed_url.path.startswith("/wiki/"):
        logging.debug("Invalid wiki article path '%s'", parsed_url.path)
        raise HTTPException(status_code=400, detail="Invalid wiki article path.")

    return lang
```

Sprint 4

Caching

Method of Retrieval	Response Time
No caching	300-1500ms
Caching	5-7ms

Up to 300x speed increase!!

Updated Comparison Endpoint

```
2  "comparisons": [  
3    {  
4      "left_article_array": [  
5        "Barack Hussein Obama II is an American politician who served as the 44th president of the United States from 2009 to 2017.",  
6        "Obama previously served as a U.S. senator representing Illinois from 2005 to 2008 and as an Illinois state senator from 1997 to 2004.",  
7        "Obama was born in Honolulu, Hawaii.",  
8        "He graduated from Columbia University in 1983 with a Bachelor of Arts degree in political science and later worked as a community organizer in Chicago.",  
9        "In 1988, Obama enrolled in Harvard Law School, where he was the first black president of the Harvard Law Review.",  
10       "In the 2008 presidential election, after a close primary campaign against Hillary Clinton, he was nominated by the Democratic Party for president.",  
11       "Obama selected Joe Biden as his running mate and defeated Republican nominee John McCain and his running mate Sarah Palin."  
12     ],  
13     "right_article_array": [  
14       "Barack Hussein Obama II is an American politician who served as the 44th president of the United States from 2009 to 2017.",  
15       "A member of the Democratic Party, he was the first African-American president in American history.",  
16       "He graduated from Columbia University in 1983 with a Bachelor of Arts degree in political science and later worked as a community organizer in Chicago.",  
17       "In 1988, Obama enrolled in Harvard Law School, where he was the first black president of the Harvard Law Review.",  
18       "He became a civil rights attorney and an academic, teaching constitutional law at the University of Chicago Law School from 1992 to 2004.",  
19       "In 1996, Obama was elected to represent the 13th district in the Illinois Senate, a position he held until 2004, when he successfully ran for the U.S. Senate.",  
20       "In the 2008 presidential election, after a close primary campaign against Hillary Clinton, he was nominated by the Democratic Party for president."  
21     ],  
22     "left_article_missing_info_index": [  
23       1,  
24       2,  
25       6  
26     ],  
27     "right_article_extra_info_index": [  
28       1,  
29       4,  
30       5  
31     ]  
32   }  
33 ]
```


Documentation

```
"""
This is the API which handles backend. It handles following features
1. Providing source article (with input as URL or Title)
2. Providing available translation languages list
3. Providing translated content
```

Note: You can run this API by cd'ing into 'fastapi' and running 'uvicorn app.main:app --reload' in terminal. Use postman to get response while debugging.

Here is a sample request to request an article using a URL:

```
GET http://127.0.0.1:8000/symmetry/v1/wiki/articles?query=https://
en.wikipedia.org/wiki/covid
```

The response will be a JSON object with the source article and the languages available for translation.

Further documentation on how these requests are concatenated can be found in the wiki_article.py file.

```
"""
```

BACKEND GUIDE

I. RUNNING APP

Refer to [INSTALLATION.md](#) for more in depth configuration details and dependency ins

1. Ensure that your current working directory is `fastapi` (i.e. run `cd fastapi`)
2. If you have not already, set up your virtual environment (`python3 -m venv venv`)
3. Switch to the virtual environment (`source venv/bin/activate` or `venv\bin\activat`)
4. If you have not already, install dependencies (`pip install -r requirements.txt`)
5. Consider enabling debug logging:
 - i. Create `.env` inside the `fastapi` folder
 - ii. Edit it and properties to change the logging level to debug and set the debug

```
LOG_LEVEL=DEBUG
FASTAPI_DEBUG=True
```

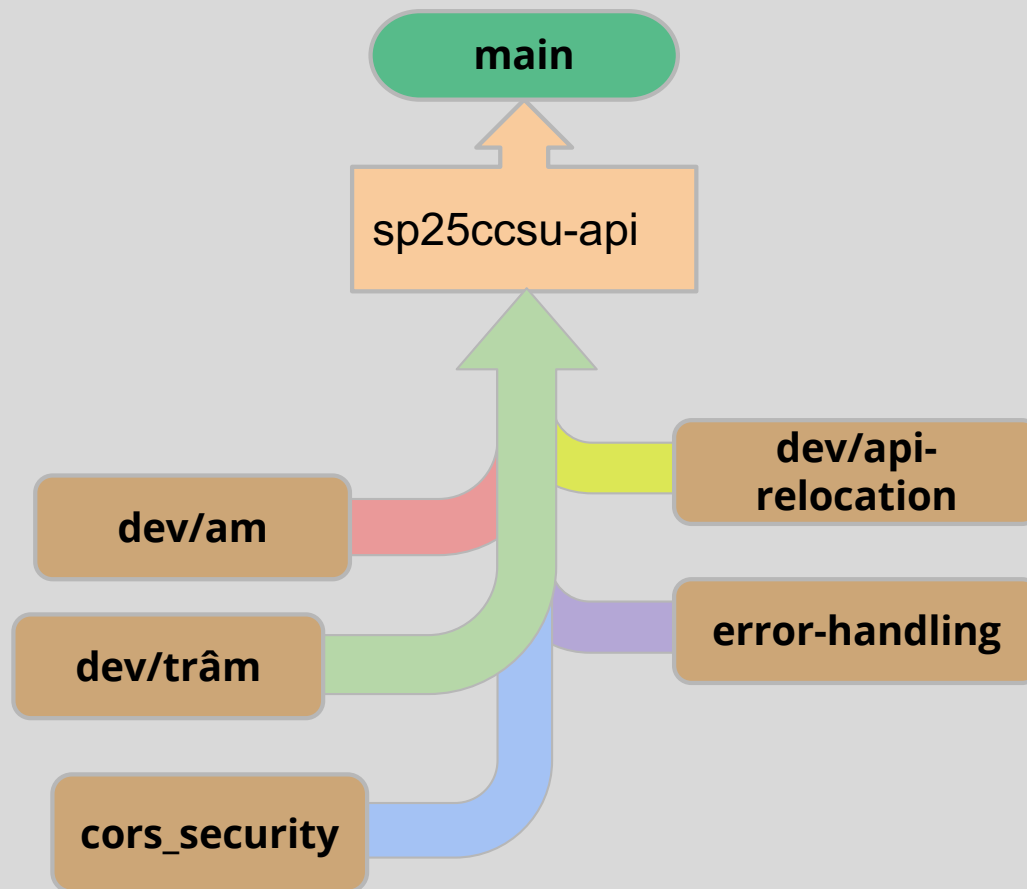
6. Finally, run the app with Uvicorn (`uvicorn app.main:app --reload`)
 - The app runs on port 8000 by default. To use another port, add `--port <number>`

You should see logs similar to the following:

```
INFO: Will watch for changes in these directories: ['/Users/erik/Dc
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to qui
INFO: Started reloader process [623] using StatReload
INFO: Started server process [625]
INFO: Waiting for application startup.
INFO: Application startup complete.
```

To test the endpoints, use a tool like POSTMAN or curl in order to make GET or POST re

Merging



Update dummy response structure and response mo Tram46		5/2/2025 23:33
Implement comparison endpoint and update API mod Tram46*		4/27/2025 22:40
Merge branch 'dev/tram' into sp25ccsu-api	Jikoo	Yesterday 11:04
corrected translate endpoint	Erik Jensen	5/2/2025 23:41
Fix improperly-formatted domains slipping through	Jikoo	5/2/2025 20:41
Move error handlers for whole project to main	Jikoo	5/2/2025 20:23
Slight documentation	Jikoo	5/2/2025 20:22
Important method -> important place	Jikoo	5/2/2025 20:12
Autoformat	Jikoo	5/2/2025 20:09
Fix typing warning	Jikoo	5/2/2025 20:08
Improve logging	Jikoo	5/2/2025 20:07
Fix getting article by title	Jikoo	5/2/2025 18:32
Merge remote-tracking branch 'origin/cors_security' Jikoo		5/2/2025 18:18
Enhancements and refactoring: - Improved input vali bigjoecon*		4/29/2025 16:48
Merge remote-tracking branch 'origin/error-handlin	Jikoo	5/2/2025 17:47
Eliminated redundant commented code	bigjoecon	4/30/2025 18:16
Enhancements and refactoring: - Improved input v bigjoecon		4/29/2025 16:48
Update endpoint usage	Jikoo	4/25/2025 17:11
Use existing project root	Jikoo	4/25/2025 17:05
Clean up merge artifacts	Jikoo	4/25/2025 17:03
Resolved merge conflict in main.py and complete Tram46		4/24/2025 22:04
updated main to initialize with debug paramater Erik Jensen		4/24/2025 14:36
added try-except blocks to cover all edge cases Erik Jensen		4/24/2025 13:32
Delete training_args.bin	origin & main patilraje*	4/23/2025 13:11
Delete tokenizer.json	patilraje*	4/23/2025 13:11
Delete spiece.model	patilraje*	4/23/2025 13:11
Delete special_tokens_map.json	patilraje*	4/23/2025 13:10
Delete generation_config.json	patilraje*	4/23/2025 13:10
Delete config.json	patilraje*	4/23/2025 13:09
Add files via upload	patilraje*	4/23/2025 13:08
Repackage endpoints and schema classes (#2)	Adam*	4/14/2025 21:32
Update requirements.txt	Tram46	4/11/2025 17:11
Add requirements.txt file	Tram46	4/10/2025 15:54
Update GitHub workflow path to fastapi/app	Tram46	4/10/2025 15:11
Add /get_article endpoint to support lookup by URL c Tram46		4/10/2025 13:56
Add /get_article_new endpoint supporting URL or key Tram46		4/10/2025 08:09
Update screen shot on README.md	nathan greyboy	3/7/2025 18:55

What Worked

- Communication
- Delegation
- Mob Programming
- Consistency
- Planning
- Teamwork
- Read then Write (Research and Familiarization)

Lessons Learned

- Time Zones
- Division of Responsibility
- Collaboration
- Compromise
- Version Control
- Documentation

LAST MINUTE WORK!!!!

DEMO!