

## Overall Testing Coverage Report

### Testing Suite

- Unit tests were written for the following containers: datapipeline, api-service, frontend-simple, models. We generated github actions that ensure the unit tests are run if there are any pushes to the specified container.
- An integration test was written in tests/test\_integration.py that runs integration tests across the exposed API on every pull request.
- Additionally if there are merges to main we deploy updates to the Kubernetes cluster

**Overall Coverage:** 72%

### Datapipeline

Coverage: 60%

```
.././tests/test_datapipeline_finetuning.py .... [ 30%]
.././tests/test_datapipeline_preprocess_data.py ... [ 53%]
.././tests/test_datapipeline_preprocess_rag.py ..... [100%]

----- coverage: platform linux, python 3.12.7-final-0 -----
Name                Stmts   Miss  Cover
-----
__init__.py           0      0   100%
finetuning_data.py    97     63    35%
preprocess_data.py    33      9    73%
preprocess_rag.py     79     12    85%
-----
TOTAL                 209     84    60%
Coverage HTML written to dir htmlcov
```

### API Service

Coverage: 85%

```
.././.././tests/test_utils_chat.py ... [ 42%]
.././.././tests/test_utils_llm_rag.py .... [100%]

----- coverage: platform linux, python 3.12.7-final-0 -----
Name                Stmts   Miss  Cover
-----
chat_utils.py        54     11    80%
llm_rag_utils.py     51      5    90%
-----
TOTAL                 105     16    85%
Coverage HTML written to dir htmlcov
```

## Models

Coverage: 70%

```
../../tests/test_models_model_rag.py ... [ 60%]
../../tests/test_models_train_model.py .. [100%]

----- coverage: platform linux, python 3.12.7-final-0 -----
Name                Stmts   Miss  Cover
-----
model_rag.py         54     19    65%
train_model.py       51     13    75%
-----
TOTAL                105     32    70%
Coverage HTML written to dir htmlcov
```

## Frontend Simple

Since there are no python files in frontend-simple we did not generate a coverage report for these tests.

```
► Run pipenv run pytest ../../tests/test_frontend*
===== test session starts =====
platform linux -- Python 3.12.7, pytest-8.3.3, pluggy-1.5.0
rootdir: /home/runner/work/AC215_GitGirls/AC215_GitGirls
plugins: cov-6.0.0
collected 5 items

../../tests/test_frontend_simple.py ..... [100%]

===== 5 passed in 1.40s =====
```

## Integration Tests

These tests check that the url for the site works on every pull request. We check that the homepage is correct and that the chat functionality works and returns a message when prompted.

```
► Run pipenv run pytest ../../tests/test_integration.py
===== test session starts =====
platform linux -- Python 3.12.7, pytest-8.3.4, pluggy-1.5.0
rootdir: /home/runner/work/AC215_GitGirls/AC215_GitGirls
plugins: anyio-4.6.2.post1, cov-6.0.0
collected 2 items

../../tests/test_integration.py .. [100%]

===== 2 passed in 4.36s =====
```

**Modules that lack testing**

- We did not write any tests for the deployment container but we did manual testing to ensure that we can generate a website that can be interacted with and that the deployment uses the correct Kubernetes cluster for scaling.