Assignment 4

Quality Assurance & Automation

Project: Workshops Check-in for InNoHassle

Team Members and Their Contributions:

- Insaf Garifullin – Team lead, organized the work in the team, worked on a CI/CD
- Evgenii Moskal – Frontend developer, continued writing the frontend, rewrited the project using Tailwind CSS
- Kirill Karaev - Frontend developer, continued writing the frontend, rewrited the project using Tailwind CSS
- Khasan Sadykov - Backend developer, continued work on the backend
- Ilnur Ramazanov - Backend developer, continued work on the backend, managed notes during the meeting with customers
- Ranis Haertdinov - Designer, worked on assignment

Non-participants: None

June 29, 2025

Backlog

Link to the Backlog

Sprint Milestone

Link to the Sprint milestone

Sprint Tracking Table

Link to the Sprint Tracking Table

Deployment and Demo

- **Link to deployed product:** `https://workshops.innohassle.ru/workshops`
- **Short Video Demo (2 mins):** Link

User Testing

- **User Testing Recording:** Link
- **Defined User Tests:** Link
- **Testing Report:**
  - Total users: 1
  - Key feedback: Everything works well, but there are minor bugs when editing workshops which need to be fixed.
  - Adjustments made: Further investigation required to resolve these issues promptly.

QA Tools Summary

Used Tools

- **Black**: for automatic style checking (linting).
- **Pytest**: for unit testing and test coverage analysis.
- **GitHub Actions**: for continuous integration and deployment (CI/CD).

CI/CD Process Description

We have implemented a CI/CD pipeline that triggers on each push or pull request. The CI/CD process includes the following steps:

1. Linting with **Black**, ensuring adherence to coding style guidelines before proceeding further.

2. Testing:

   a.) Running all tests via **Pytest** to ensure functionality works as expected.

   b.) Coverage measurement to identify areas needing more test cases.

3. Deployment: Final step where verified and tested code is automatically deployed to production servers.

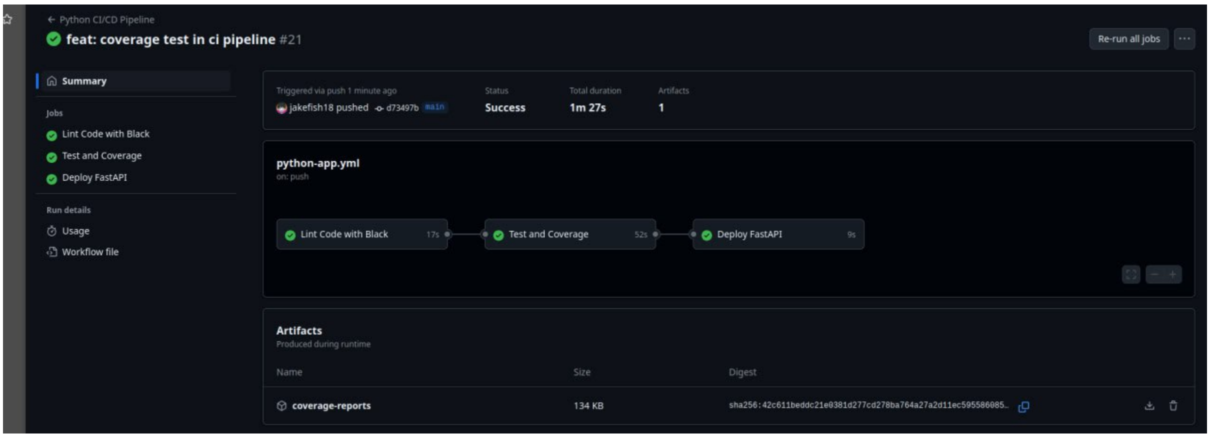
This approach ensures fast feedback loops, enhances development quality, and minimizes risks associated with introducing bugs into live environments.

Testing Artifacts

- **Unit Tests:** Link
- **Integration Tests:** Link
- **CI Pipeline:** Link

Reports and Coverage

- **Coverage Report:**



feat: coverage test in ci pipeline #21

Jobs: Lint Code with Black; Test and Coverage; Deploy FastAPI

Triggered via push 1 minute ago: jakefish18 pushed d73497b main

Status: Success

Total duration: 1m 27s

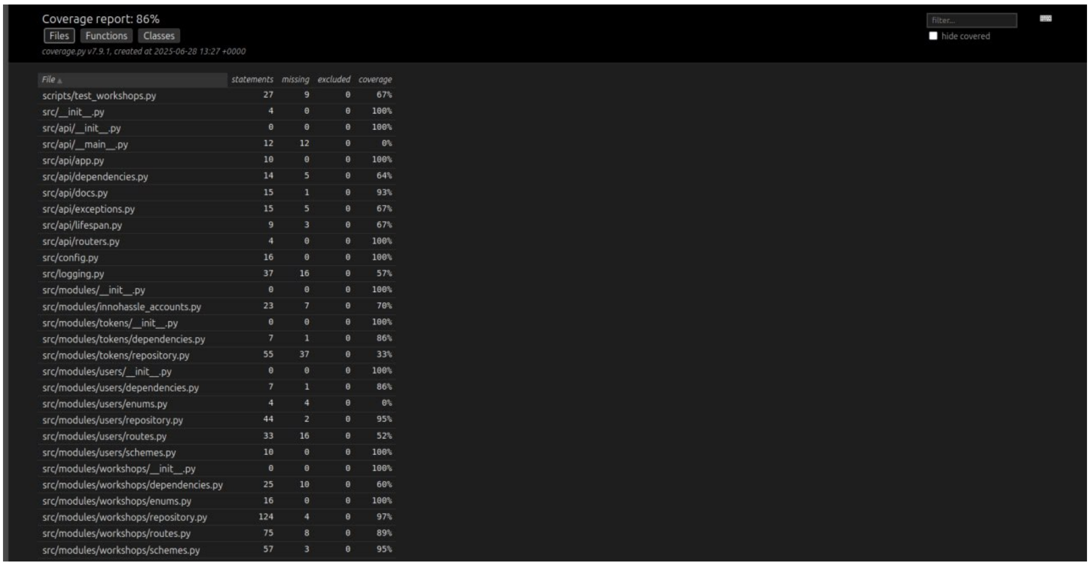
Artifacts: 1

python-app.yml
on: push

Lint Code with Black 17s → Test and Coverage 52s → Deploy FastAPI 9s

Artifacts
Produced during runtime

Name	Size	Digest
coverage-reports	134 KB	sha256:42c611beddc21e0381d277cd278ba764a27a2d11ec595586085…



Coverage report: 86%

coverage.py v7.9.1, created at 2025-06-28 13:27 +0000

File ▲	statements	missing	excluded	coverage
scripts/test_workshops.py	27	9	0	67%
src/__init__.py	4	0	0	100%
src/api/__init__.py	0	0	0	100%
src/api/__main__.py	12	12	0	0%
src/api/app.py	10	0	0	100%
src/api/dependencies.py	14	5	0	64%
src/api/docs.py	15	1	0	93%
src/api/exceptions.py	15	5	0	67%
src/api/lifespan.py	9	3	0	67%
src/api/routers.py	4	0	0	100%
src/config.py	16	0	0	100%
src/logging.py	37	16	0	57%
src/modules/__init__.py	0	0	0	100%
src/modules/innohassle_accounts.py	23	7	0	70%
src/modules/tokens/__init__.py	0	0	0	100%
src/modules/tokens/dependencies.py	7	1	0	86%
src/modules/tokens/repository.py	55	37	0	33%
src/modules/users/__init__.py	0	0	0	100%
src/modules/users/dependencies.py	7	1	0	86%
src/modules/users/enums.py	4	4	0	0%
src/modules/users/repository.py	44	2	0	95%
src/modules/users/routes.py	33	16	0	52%
src/modules/users/schemes.py	10	0	0	100%
src/modules/workshops/__init__.py	0	0	0	100%
src/modules/workshops/dependencies.py	25	10	0	60%
src/modules/workshops/enums.py	16	0	0	100%
src/modules/workshops/repository.py	124	4	0	97%
src/modules/workshops/routes.py	75	8	0	89%
src/modules/workshops/schemes.py	57	3	0	95%