

Assignment 4.

Quality Assurance & Automation

InnoSport Website, Team 9

MEMBER	ROLE	TASKS
Aleksandr Cherkasov	Team-Lead, Backend	<ul style="list-style-type: none">• Project management• Backend development• Organised meeting with the customer and discussed new features
Almir Avkhadiev	Backend, QA	<ul style="list-style-type: none">• Integration tests• Unit tests• Git management
Anton Bugaev	Frontend	<ul style="list-style-type: none">• Fixing bugs• Implementing new frontend features
Dmitriy Ezovskikh	Backend	<ul style="list-style-type: none">• Assignment 4 preparation• Setting up the linter and tester• QA tools analysis
Timur Daminov	Frontend	<ul style="list-style-type: none">• Fixing bugs• Implementing new frontend features

July 13, 2025

Contents

1. Sprint milestone	3
2. Sprint tracking table	3
3. Deployment	3
4. Demo video (in english) with changes since previous week	3
5. User testing	3
5.1. Link to the recording of user testing session with the customer	3
5.2. The defined user tests	3
5.3. Report on the results of user testing	3
6. QA and Tools	4
6.1. Research on vailable QA tools	4
6.2. List of tools that we used	4
7. Tests	4
8. CI pipeline	4
9. Testing reports	4
9.1. Linter reports.	4
9.2. Coverage reports.	5

1. Sprint milestone

Backend: <https://github.com/inno-sport-inh/backend/milestone/2>

Frontend: <https://github.com/inno-sport-inh/frontend/milestone/2>

2. Sprint tracking table

https://docs.google.com/spreadsheets/d/1gCACS5LkCOjsXijKPbO_xkv4q1_KrqgI3_VfKA2161Q/edit?usp=sharing

3. Deployment

Admin panel of the main sport site: <https://t9d.store>

user mail: `test@t9d.store`

user password: `roLouL*6Ei8J`

New front of the sport site: <https://frontend-fawn-six-66.vercel.app>

4. Demo video (in english) with changes since previous week

<https://disk.yandex.ru/i/4fwlmu2YTgAtqg>

5. User testing

5.1. Link to the recording of user testing session with the customer

<https://disk.yandex.ru/i/68Sf2PpOYJ0Yhg>

5.2. The defined user tests

1. Training check-in
2. Reading the sport club information
3. Using the FAQ section

5.3. Report on the results of user testing

As a result, several improvements were suggested by the customer. To make the report more structured and clear, the suggestions will be represented as a list of issues. The same issues can be found on our github.

Suggestions:

1. Add confirmation dialog before canceling a training session
2. Add “collapse/expand” button for past days in the weekly schedule (collapsed by default)
3. Merge training time and club name dialogs into one; unify “enroll” and “book activity” terminology
4. Make club cards clickable (link to club page) beyond just the “Learn more” button
5. Allow training enrollment directly from the club page
6. Improve FAQ search (support multi-word queries, typos, etc.)

6. QA and Tools

6.1. Research on available QA tools

As we use python and React.js, we searched for QA tools for these languages. Here is the results:

1. Python

- pytest
- unittest
- coverage.py
- flake8

2. React.js

- Jest
- ESLint

6.2. List of tools that we used

• Python

We used pytest and flake8 (linter). The reason of this choice is because pytest is very easy and scalable, as well as flake8, which is easy to install and outputs nice report in .txt format.

• React.js

We used Jest for testing, it is the most popular and well documented framework. It is easy to set up too. As a linter, we chose github Super-Linter over ESLint because it requires absolutely no setup. Still, it supports React.js and outputs nice and structured report.

7. Tests

- Unit tests: [link](#)
- Integration tests: [link](#)

8. CI pipeline

Backend: <https://github.com/inno-sport-inh/backend/actions>

Frontend: <https://github.com/inno-sport-inh/frontend/actions>

9. Testing reports

<https://disk.yandex.ru/i/vouzZGDocXhCMQ>

9.1. Linter reports.

Linter reports for python can be found here:

<https://github.com/inno-sport-inh/backend/actions>

- Under each workflow run, there will be linter-report file attached.

Linter reports for React.js can be found here:

<https://github.com/inno-sport-inh/frontend/actions>

- Under each workflow run, there will be “lint-and-test” job attached.
- Inside this job, there will be section “Run Super-Linter”

- Inside the “Run Super-Linter” section you can see the linter output, it will contain different sections such as CSS, HTML, GITLEAKS, JAVASCRIPT_STANDART, e.t.c.

9.2. Coverage reports.

Python coverage reports can be found here:

<https://github.com/inno-sport-inh/backend/actions>

- Under each workflow run, there will be “Run tests” job attached.
- Inside this job, there will be section “Run tests”

- Here, you can see the tester output, at the end of the output you can see how much tests failed and how much passed. Example of the output:

```
===== short test summary info
=====
FAILED api/tests/api/test_profile.py::ProfileAPITests::test_get_profile -
Att...
===== 1 failed, 69 passed, 43 warnings in 32.29s
=====
```

React.js coverage reports can be found here:

<https://github.com/inno-sport-inh/frontend/actions>

- Under each workflow run, there will be coverage-report document attached