Faculty of Computer Science and Engineering

Advanced QA Testing Project

GitHub Link from the Project

Contributors to the project:

Gorjan Bogoevski 221193

Leon Asanovski 221007

What is QA and why is it important?

- Quality Assurance (QA) Testing is a critical phase in the software development lifecycle that ensures applications meet predefined standards of quality, reliability, functionality, and performance before reaching end users. It involves systematically evaluating software through various testing techniques to identify bugs, verify expected behaviors, and validate overall system integrity.

    1. **Ensures Product Reliability**
       QA testing helps catch defects early, ensuring the software behaves consistently under different conditions and use cases.
    2. **Improves User Satisfaction**
       By delivering bug-free and user-friendly applications, QA enhances the end-user experience and builds trust in the product.
    3. **Saves Time and Costs**
       Fixing bugs during development is significantly cheaper and faster than after release. QA reduces rework, downtime, and customer complaints.
    4. **Enables Continuous Improvement**
       QA provides feedback loops for developers, enabling continuous delivery and agile development processes to thrive.

Testing Type	What It Validates
Unit Testing	Individual components or methods (e.g., services)
Integration Testing	Interactions between modules and data flow
API Testing	REST/GraphQL endpoints, response formats, errors
UI Testing	Visual and functional behavior of web/mobile UIs
Performance Testing	Load, stress, and scalability under high usage
Security Testing	Vulnerability scanning and data protection
End-to-End Testing	Full user workflows, simulating real user behavior
Accessibility Testing	Usability for people with disabilities

What have we done on our project?

While we were thinking about what we are going to test, we came up to 4 proofs of concepts we want to realize, so we can cover the learned methods from the course and additionally add up some new things we can see for the first time.

1. **UI Testing** – This concept was done by testing a fake web application that is open source and is open for manipulating. The idea was to test the whole web app., all the functionalities but from User Oriented Aspect.

   Link from the web page -> link | Link from the git folder of UI Testing -> link

2. Load Testing – This concept was meant to be created so we can learn how is it done to test some app under stress and under high usage. We know that we could have done it in Postman, but we wanted something unused, and we chose to test it with JMeter Application. This application is offering creating some tests, creating assertions, timers, etc., and the most important we can do it under different conditions and load, so we can see the speed, the responsiveness and the scaling of the app.

   Link from the web page -> link | Link from the git folder for JMeter -> link

3. API Testing – This concept was meant to test some rest calls and to see its responses, the body, the headers, etc. The tests were made using a fake rest api web page, where there are a lot of endpoints that can be accessed in many ways.

   Link from the web page -> link | Link from the git folder for API tests -> link

4. Application Testing – In this concept we have covered literally every method that we have learned during the course, for example Unit Testing, Parameterized Unit Tests, then we covered Logic Coverage, Graph Coverage, Input Space Partitioning and we covered Mocks too. This application was made during the Laboratory Exercises on the Web Programming course by us, and we have used it, we have upgraded it with several methods so we can fully test all the concepts we need to cover for the project. At the end, we wanted to add something new to the project, and we tested it with Spring MVC Framework Tests, where we have learned additionally a new tool for testing.

   Link from the Application Repo -> link | Link from the Application Folder -> link

Note:

We have provided a detailed Documentation, where there is more description about the tests that are covered and the methods, how have we created the tests etc.

The main link of the repository is provided on the Cover Page, but here it is again.

LINK