

INSTRUCTIONS TO THE COMPETITOR

Module E – Advanced Web Development

In this module, you are expected to solve three tasks.

Within the media files, you will find three starter kits for each task. You are expected to use these starter kits as a base for your solution. You are not allowed to use any frameworks or libraries for this module except a testing framework for task 1.

Task 1: Writing automated tests

You are given a JavaScript project that has no automated tests. You must write automated unit tests for the project.

A complete test set is expected which covers 100% of the provided code lines and conditionals. A JavaScript testing framework must be used.

These are the assessment criteria for this task:

- The tests are grouped into files logically.
- The tests are written in a way that they are easy to understand.
- The tests pass when running against the original code.
- The tests cover 100% of the provided code lines and conditionals.
- The tests do not pass any logically mutated version of that code (Mutation Testing).
- Each test file is submitted in a separate Pull Request with a clear title and description.
- Commit messages are clear and descriptive.

Task 2: OWASP Top 10 Security Vulnerabilities

You are given a JavaScript project that has code security vulnerabilities. You must find and document these vulnerabilities and suggest improvements. You will document them in an issue tracker. The security vulnerabilities must be documented in a way that they can be understood by a developer with basic knowledge of security issues. You must also provide a summary of the security vulnerabilities you found in a central issue on the issue tracker that links to the individual issues.

These are the assessment criteria for this task:

- The vulnerabilities are documented in a way that they can be understood by a developer with basic knowledge of security vulnerabilities.
- The issues include:
 - o a description of the issue,
 - o the place(s) in the code where the issue is located,
 - o why it is a security issue,
 - o to which OWASP Top 10 category it belongs,
 - o and concretely how to fix it minimally invasive.
- The issues are documented in an issue tracker.

- The issues are grouped logically.
- As many real vulnerabilities as possible are documented.

The OWASP Top 10 documentation is going to be provided in the media files.

Details on the issue tracker will be provided in the competitor handout.

Task 3: CSS Dark Mode Toggle

Your client calls you about a new feature request for a simple static website they already have.

They would like you to implement a dark mode to the existing website with the following requirements:

- The preferred color scheme from the operating system should be applied without JavaScript being needed
- If JavaScript is enabled, a new toggle must be added to the header which switches between light and dark theme, allowing the user to overwrite the system default theme
- If the theme is changed with the manual toggle in the header, the chosen theme must be preserved over page reloads, when the tab is closed and the page opened again, and across browser restarts
- The background color of the page in dark mode should be #121212
- All text on the page should be well readable in the dark mode with a color contrast ratio of at least 7:1 to meet WCAG Level AAA
- If an image has a dark theme variant (provided, next to the light theme variant), it should be used when the dark theme is active, even without JavaScript enabled
- As the colors might have to be tuned later, all colors should be referenced as CSS variables, which are defined in a central place, even colors that are set in the already provided styles
- No CSS preprocessors or frameworks, nor JavaScript frameworks can be used

To track this feature request and to be able to reference it in the future, you must record it in an issue tracker.

Create a new issue summarizing the client's request and list all requirements.

Keep the status of the issue up-to-date as you progress with the implementation, and create a link between the issue and the pull request(s) containing the feature in both directions.

Ultimately, all pull requests must be merged, and the ticket must be closed.