

# Rubrics

## Web Development 1

**Course:** Web Development 1

**Term:** 2.2

**EC:** 3

**Lecturers:** M. de Haan / Dan Breczinski

## Required functionality

The assignment requires a reasonable level of complexity to be present for an opportunity passing grade. The actual grade will be determined by the quality rubrics below.

Fail (0 points, assignment will not be graded)	Pass (grants 1 point)
The application does not have an authentic use case and/or does not go beyond the code discussed in the lectures.  The code is written by AI without convincing demonstration of understanding.	The application is built on an authentic use case, formulated by the student. Knowledge gained during the lectures is expanded upon by creating original functionality.  The application proposal has been communicated clearly (in writing) to the teacher and has been approved.  AI has not been overused. The student understands and is able to convincingly explain their code.

## Grading criteria

The assignment will be graded according to the rubrics below. The application is expected to score at least one point in every rubric. If one rubric is graded as zero, the maximum grade becomes 5 out of 10. Otherwise, the grade is the sum of all assigned points with a maximum of 10. To achieve a higher level rubric, all lower level items must be completed.

*Please note that some grading items require that you document your implementation in the README file with reference to the relevant files.*

Rubric	0 points	1 point	2 points
CSS	The application has not been styled using a CSS framework.	The application has been styled using a CSS framework.  The application has a consistent look & feel with only minor faults.  The application adapts well to smartphone, tablet, laptop and desktop viewport sizes.	The application has a generally professional, polished look.  The application makes use of transitions for state changes.
Sessions	The application does not make use of sessions	The application makes use of sessions for storing and reading simple data, such as login information	

Security	<p>The application is not well protected against malicious JavaScript and SQL code injection.</p> <p>Passwords are stored as plain text.</p> <p>Routes are not protected using authentication and authorization.</p>	<p>The application is well protected against malicious JavaScript and SQL code injection.</p> <p>The application implements parameterized queries and input sanitization correctly.</p> <p>The application sanitizes output to the browser to prevent XSS.</p> <p>The application validates user input on the server to make sure it is the correct type (email, url, etc.) and present before saving.</p> <p>Passwords are only stored in hashed form.</p>	
MVC	<p>The MVC pattern is not or only partially implemented.</p>	<p>All CRUD operations are present in the application.</p> <p>The MVC pattern has been well implemented, with a clear division of responsibilities between controllers, models and views.</p> <p>Routing is present.</p> <p>Repository (or similar) classes are used consistently for data access.</p>	<p>Service and repository layer are included, and interfaces are used when defining integrations between layers.</p> <p>Higher level patterns, used in ASP.net are applied such as, automatic view mapping, use of annotations to define route and controller method binding, dependency inversion to automatically configure class dependencies. These patterns are noted in the README with relevant files referenced.</p> <p>Routing and view templating are present and implemented well.</p> <p>Services and repositories are use to encapsulate business logic and data access.</p> <p>The code is generally well structured and makes good use of object orientation concepts (classes,</p>

			inheritance, encapsulation and polymorphism).
API	There is no API functionality implemented	The application provides one or more API endpoints that allow access to data in JSON format.	
JavaScript	There is no JavaScript functionality implemented	<p>The application makes use of JavaScript to update parts of pages without refreshing, by reading and processing JSON data.</p> <p>The application makes use of JavaScript to communicate with the API endpoint to update parts of pages without refreshing, by reading and processing JSON data.</p>	
Legal / Accessibility	Compliance with WCAG and GDPR have not been considered.	Compliance with WCAG and GDPR have been documented in the README with reference to code that demonstrates fulfillment of these requirements.	