Rubrics

Web Development 1

Course: Web Development 1
Term: 2.2
EC: 3
Lecturer: M. de Haan

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 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.

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.

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 and look & feel with only minor faults.	The application has been styled using a CSS framework. The application has a consistent and look & feel. The application has a generally pleasing aesthethic. The application adapts well to smartphone, tablet, laptop and desktop viewport sizes.
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	The application is not well protected against malicious JavaScript and SQL code injection. Passwords are stored as plain text.	The application is well protected against malicious JavaScript and SQL code injection. The application implements parameterized queries and input sanitization correctly.	

		Passwords are only stored in hashed form.	
MVC	The MVC pattern is not or only partially implemented.	All CRUD operations are present in the application. The MVC pattern has been implemented, with a clear division of responsibilities between controllers, models and views. Routing is present. Repository (or similar) classes are used consistently for data access.	All CRUD operations are present in the application. The MVC pattern has been implemented, with a clear division of responsibilities between controllers, models and views. Routing and view templating are present and implemented well. Services and repositories or similar classes are used consistently for data access. The code is generally well structured and makes good use of basic object orientation concepts (classes, 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	The application makes use of JavaScript to update parts of pages without refreshing, by reading and processing JSON data.	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. The application makes use of JavaScript to send data to the server without refreshing.