WAD Student Reader

Cohort 2020-2021 Sprig

Contents

Module information

In this document you can find a week-by-week decomposition of the features you have to work on for the web application you and fellow peer are going to submit for WAD. You can decide as a team what your web application is going to be about, but make sure it does meet the following constraints:

- Must have pages for registered users.
- Revolves around a type of content (indicated as <content> in the rest of the document) which must dynamically be loaded from a MySQL Database.
- Cannot be the web application required for the module PRJ.
- Cannot make use of frameworks

Some examples could be a web application for messages/posts, recipes, movie reviews, quizzes, events, products, etc.

In de rest of the document you can find, for each week, the features you should work on for your web application. Note that the practical will sometimes not follow the same pacing as the lectures with regards to the technologies you will have to use; this is by design but does not mean you cannot already work for futures weeks when you are done (this is actually encouraged).

Our advice, related to way-of-working, is to first try to practice the technologie(s) related to a feature before actually incorporating it into your web application. This will be beneficial for the code quality of your web application (e.g. less experimenting which usually results in messy code).

You can, for example, create two PHP files to practice/experiment how you can create a login form, post request handler and sessions.
After you are comfortable with the technology you should start deciding how to include it in your web application.

Whatever you do, make sure you divide the work as evenly as possible between yourself and your teammate. Besides the features, you can also find links for resources you can use.

During week 9 and 15 you will receive a formative indication for your web application. In Appendix A: Formative assessment matrix you can find the guidelines used to assess what kind of formative indication you could receive at the respective weeks.
Note that the 'complexity' of your chosen <content> and code quality does influence the formative indication.

Expected workload

The approach used in this module is *learning-by-doing,* therefore this course is highly self-study oriented. Besides lectures, an 'average' student will have to spend around 6+ hours of self-study per week; this is in addition to the lectures. The theory lectures introduce what a certain technology is, the purpose of it and the essentials on how to use it. It is the student's task to go deeper and study further in detail such technologies (see practical section) so that he/she is better prepared to tackle the assignment.

Since web development is mainly a collaborative process you will work in groups of two. It is also an incremental and iterative process, so you are expected to work regularly on it and periodically discuss your work with the teacher and incorporate feedback. You must commit your work at least on weekly basis in Git. It is important that you show consistent learning and improvements related to the learning outcomes.

WAD Practical - week 1

Web application features

Nr.	Required	Feature	Pre-request	Remarks
1.1	X	Install the software for your development environment: WAMP (for Windows), MAMP (for MacOS), LAMP (for Linux) or XAMP (for all) Code editor GIT Client		
1.2	X	Create a group of two students and decide on a topic for your web application		Make sure your website topic allows: • registration and login of users to access member pages, • database to store <content> displayed on your webpage.
1.3	X	Initialize a GIT Repository for your WAD group and invite you teacher; make sure your teacher at least has *Reporter* role permission		
1.4	X	Create wireframes for the layout of a: • desktop version of a webpage • mobile version of a webpage		If you are still learning HTML & CSSS we advise you to keep it simple and go for a layout with a header, nav-bar, content, and a footer. See for examples: https://www.w3schools.com/css/css_templates.asp
1.5		Create wireframes for the layout of a: • desktop version of a landing page (/homepage) • mobile version of a landing page (/homepage)		
1.6	X	Submit Ideation document on Canvas before deadline		

Resources

Topic	Links

Fontys GitLab	https://git.fhict.nl
HTML	https://www.w3schools.com/html/default.asp
CSS	https://www.w3schools.com/css/default.asp
Example webpage layouts	https://www.w3schools.com/css/css_templates.asp

WAD Practical - week 2

Web application features

Nr.	Required	Feature	Pre-request	Remarks
2.1	X	Implement the wireframes as a plain HTML page using a grid-layout for the normal webpage	1.4	Using the CSS grid layout makes it easier to create a responsive design. This will allow you to change or hide the cells in a grid depending on the 'version'
2.2		Implement the wireframes as a plain HTML page using a grid-layout for the landingpage	1.5	
2.3	X	Apply a style with CSS for the desktop version	1.6 || 1.7	Tip: if you are new to CSS, we advise you to not be too ambitious about the design and use basic CSS properties; always prioritize UX though Make sure the design is consistent with the mobile version. When you have difficulty with choosing colours you can make use of https://coolors.co/.
2.4	X	Apply a style with CSS for the mobile version	1.6 || 1.7	Decide for yourself what 'threshold' you want to use to switch to a mobile version. Make sure the design is consistent with the mobile version. When you have difficulty with choosing colours you can make use of https://coolors.co/.

Resources

Topic	Links
Fontys GitLab	https://git.fhict.nl
HTML	https://www.w3schools.com/html/default.asp
CSS	https://www.w3schools.com/css/default.asp
Example webpage layouts	https://www.w3schools.com/css/css_templates.asp
Colour schemes generator	https://coolors.co/

WAD Practical - week 3

Web application features

Nr.	Required	Feature	Pre-request	Remarks
3.1	X	If you have not done it already, create a proper folder structure for your web application and copy the HTML and CSS files to the correct folders		Make sure your folder structure stays 'clean' when you continue adding new files for your web application
3.2	X	Create a registration form as an HTML-page		
3.3	X	Create a login form as an HTML-page		
3.4	X	Create a User-class in PHP to represent a logged in user	3.2	Make sure your User-class can at least hold a username, password, and email. Decide for yourself what else is needed
3.5	X	Create another class representing the content you want to save in a database and show in your web application. **Starting from now we refer to this as the <*Content*>**.		Some examples of <*content*>: Movie-class when the website is about movies, Message-class when the website is about posting messages, etc.
3.6	X	Create a PHP page handling the form post of your registration page. When handling the form-post, you will have to make sure valid values are submitted. When the values are valid, create a User-object and then show message like: *Hello <username>, thank you for registering.*	3.2 && 3.4	Make sure you use the User-object when showing the message
3.7	X	Create a PHP page handling the form post of your login page. This page should have an array with dummy users. When handling the form-post, you will have to make sure the credentials are correct by using the array with User-objects. When the credentials are correct, redirect the user to a different page. In all other cases, you display a message like: *Invalid credentials supplied*	3.3 && 3.4	Make sure you use the User-class created for req. 1.3

3.8		Create a PHP page where you create an array with dummy content (i.e. the <*Content*>-objects) and display those on the page	3.5	Make sure you display them with the appropriate HTML-elements and style them via CSS

Resources

Topic	Links
HTML - Forms	https://www.w3schools.com/html/html_forms.asp
PHP	https://www.w3schools.com/php/default.asp https://www.php.net/manual/en/
PHP – Object Oriented	https://www.w3schools.com/php/php_oop_what_is.asp https://www.php.net/manual/en/language.oop5.php
PHP – Super Globals	https://www.php.net/manual/en/language.variables.superglobals.php https://www.w3schools.com/php/php_forms.asp
PHP – Arrays	https://www.php.net/manual/en/language.types.array.php
PHP – Redirects	https://www.php.net/manual/en/function.header.php

WAD Practical - week 4

This week you will have feedback sessions with your teacher. In addition, you should use this week to catch-up and process any feedback you received.

WAD Practical - week 5

Web application features

Nr.	Required	Feature	Pre-request	Remarks
5.1	X	Create a table for the User-class		You are expected to make use of a MySQL database
5.2	X	Create a table for the *<Content>*-class		You are expected to make use of a MySQL database
5.3	X	Create an index.php with the logic for the templating. Also convert all the HTML page to PHP pages, so it makes use of the templating		You can use the example given during the lecture, just be sure to extend it to make it fit you needs!
5.4	X	Change your registration functionality to make use of your database	3.1	You are required to make user of PDO to prevent *SQL Injection*
5.5	X	Change your login functionality to make use of your database	3.1	You are required to make user of PDO to prevent *SQL Injection*
5.6		Add or change your page to display your *<content>*	3.2	You are required to make user of PDO to prevent *SQL Injection*

Resources

Topic	Links
Hera server – MySQL database	https://selfservice.app.fhict.nl
PHP - MySQL	https://www.php.net/manual/en/book.pdo.php https://www.php.net/manual/en/ref.pdo-mysql.php https://www.w3schools.com/php/php_mysql_intro.asp (only the PDO-part!)
PHP – 'Combining' PHP files	https://www.w3schools.com/php/php_includes.asp https://www.php.net/manual/en/function.require.php https://www.php.net/manual/en/function.include.php
SQL Injection	https://www.w3schools.com/sql/sql_injection.asp https://en.wikipedia.org/wiki/SQL_injection

WAD Practical - week 7

Web application features

Nr.	Required	Feature	Pre-request	Remarks
7.1	X	Implement authentication by making use of sessions		Keep the information you store in a session at a minimum!
7.2	X	Include *member* and *admin* roles for registered users		This will require you to change your class and table
7.3	X	Create a profile page with proper authorization		
7.4	X	Include pages to CRUD *<content>*. Make sure you apply proper authorization for certain pages (e.g. what kind of user should be able to create new *<content>*?)		'Read' is not needed if you already created the page in prior weeks.

Resources

Topic	Links
PHP - Sessions	https://www.w3schools.com/php/php_sessions.asp https://www.php.net/manual/en/book.session.php

WAD Practical - week 8

Web application features

Nr.	Required	Feature	Pre-request	Remarks
7.1	X	Refactor your web application to have your codebase in proper layers.		You are expected to at least have a 3 layered design
7.2		Instead of a 3 layered design your code base follow the MVC pattern		Note: you cannot use a framework for this and is a complicated concept. Only attempt this if you are comfortable with web application development!
7.3		Include additional pages that you would expect of a proper web application		For example: landing-page, contact-page, etc.
7.4		Create a CRUD for Users with proper authorization		Only users with an admin-role should be allowed to do this.
7.5		Incorporate a functionality using cookies		Be sure you do not use the cookie to store security or privacy related data!

Resources

Topic	Links
3 Layered design	See lectures of OOD
MVC Information	https://en.wikipedia.org/wiki/Model–view–controller https://www.guru99.com/mvc-tutorial.html (note that the part about MCV Frameworks is not relevant for now and will be covered in weeks 13-15)
PHP - Cookies	https://www.w3schools.com/php/php_cookies.asp https://www.php.net/manual/en/features.cookies.php

WAD Practical - week 9

This week you will have to present your web application. In addition, you should use this week to catch-up and process any feedback you have received.

WAD Practical - week 10

Web application features

Nr.	Required	Feature	Pre-request	Remarks
10.1	x	Catch-up with the practical and incorporate the feedback you received in week 9		
10.2		Any functionality you can think of, as long as it has an added value to your web application		

WAD Practical - week 11

Web application features

Nr.	Required	Feature	Pre-request	Remarks
11.1	x	Include front-end form validation with JavaScript for all the forms in your web application		Note that it is not enough to, for example, only include a required attribute to an input-control. You will (also) have to make use of JavaScript to validate the values in the form.
11.2	x	Include a JavaScript component in your application by using a JavaScript Library		For example, a slideshow, date picker, modal-popups, etc.
11.3		Include additional font-end functionalities with JavaScript		

Resources

Topic	Links
JavaScript	https://www.w3schools.com/js/
jQuery	https://www.w3schools.com/jquery/ https://jquery.com/
jQuery UI	https://jqueryui.com/

WAD Practical - week 13 & 14

Web application features

Nr.	Required	Feature	Pre-request	Remarks
13.1		Include useful AJAX functionalities at at least two places in your web application		Note that we expect each of the group members to have implemented something with AJAX.
13.2		Connect your application with a REST API with AJAX		Note that this counts as 'one place' for 13.1
13.3	X	Finish and polish you web application		
13.4	X	Prepare for your presentation in week 15		
13.5		Any of the optional features from previous weeks		

Resources

Topic	Links
JavaScript - JSON	https://www.w3schools.com/js/js_json.asp
jQuery – AJAX	https://www.w3schools.com/jquery/jquery_ajax_intro.asp
PHP - AJAX	https://www.w3schools.com/php/php_ajax_intro.asp https://www.w3schools.com/php/php_json.asp https://www.php.net/manual/en/book.json.php

WAD Practical – 15

This week you will have to present your web application.

Appendix A: Formative assessment matrix

In weeks 9 and 15 your will receive a formative indication for your website. This is based on the quality of your web applications. Below you can find the guideline your teacher uses to assess your submission.
The overview shows you what your submission should ‘contain’ and to what formative indication it can result to. For example, to be eligible for a *G* you also need to make sure you meet the criteria of *P*, *U* & *S*

Week 9	Requirements for the web application	Week 15
P	Equal work division (backed-up by Fontys GIT Lab repository)	P
	Responsive design by applying an HTML grid element (for desktop & mobile layout)	
	All CSS & JS in separate files	
U	Proper usage of classes in PHP	
	Minimize PHP code, HTML and CSS duplication (usage of include/require)	
	Proper file structure with logical folders and file names	U
	UX/usabilty of the web application (e.g. logical page flow, nav-bar position, 'layout conistency' of pages, user feedback, etc.)	
	Usage of PDO and parameterized queries	
S	MySQL database with appropriate design (e.g. primary key, foreign key, etc.)	
	Usage of the proper SQL queries to CRUD	
	R page(s) for <content>	
	Registration & login functionality (authentication) for users with different roles	
G	C page(s) for <content>	S
	Profile page to edit user information & password with proper authorization	
	Single responsibilty applied	
O	Open/Closed principle	
	Applied a 3 layered design	
	Proper back-end validation for handling user input	G
	JavaScript form validation applied on all forms	
	At least one JavaScript component, such as, slideshow, date picker, modal-popups, etc.	
	Some static pages to complete their web application	
	Additional CRUD pages for <content> with proper authorization for certain pages (e.g. UD only by admin or the user created the <content>, etc.)	O
	Fancy styling/layout	
	Inclusion of usefull AJAX functionalities	
	Additional usefull features	

Figure 1: Last updated at 12-4-2021[1]

[1] Added missing *'R pages(s) for <content>'*