Assessment Element 2 – Build a Customer Relationship Management / E-Commerce system Using Laravel[1]

This coursework will focus on continuing the development of the Customer Relationship Management (CRM) system from SSP1 using the Laravel Framework. You must understand the core features of server-based applications, including security, encryption, token authentication and usage of APIs to expose data endpoints and then apply this knowledge to expand your own CRM. The coursework will include middleware authentication, data encryption, use of a 3rd party API and completion of an e-commerce system with a purchase flow.

Learning Outcomes

- Demonstrate a critical understanding of the functionality used in server-side web application frameworks.
- Design, implement, test and demonstrate a flexible, robust and secure server-side web application solution.
- Apply appropriate web application testing strategies and explain the importance of their use.
- Demonstrate a critical understanding of web applications' security issues and implement an appropriate strategy to counter potential risks.

Important

All assignment work is to be completed individually.
University regulations on academic conduct[2] and exceptional circumstances[3] apply. Please ensure that you are familiar with these regulations.

Schedule

Element	Issue Date	Submission	Method	%
1	TBA	TBA	Zipped copy submitted to LMS	80%
1	TBA	TBA	Demonstration of your code	20%

Submission and Assessment

Submission is to LMS. A zipped copy of your code (not a RAR file) should besubmitted immediately following your demonstration.

Submission **will not be accepted** by alternative means (such as email), so you should ensure that your submission is made well before the deadline to avoid last-minute problems.

There will be a scheduled demonstration slot for you to show your work. **Failure to demonstrate at the time of your allocated demo assessment element in zero marksfor this element of the assessment regardless of whether work has been submitted**.

Necessary! - Your assessment will be made against the marking criteria at the end of this document.

You should ensure that you are fully aware of these criteria and that you understand them and what they require of you, and that you follow them closely tomaximise your mark.

---

[1] https://laravel.com/
[2] https://www.staffs.ac.uk/students/course-administration/academic-policies-and-regulations/academic- conduct-procedure
[3] https://www.staffs.ac.uk/students/course-administration/academic-policies-and-regulations/exceptional- circumstances-procedure

Scenario

Based on the Business Model selected for the SSP 1 CRM assessment, you should expand the system to make your application a production-ready version of a CRM.

This version of the application should have the following modules.

- Product
    - Stocks
    - Categories
    - Locations
    - Attributes
- Customer
    - Profile
    - Address Book
    - History (Purchase history)
    - Analytics
- API
    - CRUD operation for Product
    - CRUD operation for Customer

Constraints

Before starting the development, you must discuss your data structure and architectural approach for the CRM with your lecturer and get his feedback and approval to continue development. *

The CRM must be developed using Laravel, Bootstrap, and AlpineJS or by using a preferred JavaScript library.

The database must be based on either SQLite or MySQL. If SQLite is used, the DB file must be attached; if MySQL is used, a SQL dump must be attached in the ZIP submission file.

All the latest versions of packages and technologies must be used.

Task

Design the data model for the system.

Create a full mind map / ER diagram for the design.

Develop the user interface.

Develop the user interfaces based on Bootstrap and AlpineJS.

Develop the database layer.

Create the database (Eloquent) based on Laravel concepts.

Testing the application for functionality, usability, and performance.

Write at least 10 test cases to test the system and functionality.

Document the process, features, and objectives of the system.

Write a complete document with the details, including screenshots of the system.

Create a quality assurance review process and test cases.

Write a complete document with the review process and explain the test cases, why it was created, and a deep analysis.

Create a future upgrade plan to ensure the system is future-proofed.

Write a min 500-word section explaining the future plan to expand this application to be marketed/sold as a SaaS solution.

The Demonstration

You are required to demonstrate your implementation over the course of 5 to 10 minutes.You will be asked to show or talk about any or all of the following:

- The required ‘base’ functionality (show only)
- An explanation of any of the code that you produce.

Marking Criteria

Base Mark (50% of the mark for Task 1)

Successfully implementing the functionality described by any criterion below will get the full marks otherwise, the mark will be zero.

Criteria	**Mark**
Design the data model for the system.	12
Develop the user interface.	8
Develop the database layer.	10
Testing the application for functionality, usability, and performance.	5
Document the process, features, and objectives of the system.	5
Create a quality assurance review process and test cases.	5
Create a future upgrade plan to ensure the system is future-proofed.	5
Total	**50**

Enhanced Mark (50% of the mark for Task 1)

The enhanced mark relies on your attempt on the base mark criteria above. Your base mark will limit the impact your assessor can award you based on their judgement of your performance guided by the standards below.

Criteria	**Limit to achievable mark**	**Assessor Guidance**	**Mark (fixed)**	**Weighted Total**
Working code	0 in Base Mark	No Base Mark criteria met.	0	
	Less than 30 in Base Mark limits maximum achievable mark to 3	Code could be made to work (good attempt made at functionality) but currently user data is not saved to the database.	1	
		User data saved to database but there were numerous errors/bugs evident.	2	
		Working code but one or two errors evident.	3	
	Above 30 in Base Mark required for up to 4 marks	Very minor suggestions / debugging required for working code.	4	
	40 in Base Mark required for maximum mark	Code was error free.	5	25
Understanding	0 in Base Mark	No Base Mark criteria met.	0	
	Less than 30 in Base Mark limits maximum achievable mark to 3	Unable to explain code when prompted.	1	
		Unable to explain code without being prompted.	2	
		Code explained with only a little prompting.	3	
	Above 30 in Base Mark required for up to 4 marks	Code explained well with no prompting.	4	
	40 in Base Mark required for maximum mark	Code explained clearly with justification of the approach taken.	5	25
Total				**50**