University of Staffordshire

COMP50016
Server Side Programming - I

Module Learning Outcomes (LO)
On completion of this module, you will be able to:

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

Deadline	Weighting
Dates available on SAIS	20%

University Regulations

- University Regulations regarding exceptional circumstances and academic misconduct will apply.
- Please ensure that you are familiar with these regulations.
- https://www.staffs.ac.uk/students/course-administration/academic-policies-and-regulations/home

Submission rules

- late submissions attract 0 marks for that section
- failure to submit on Blackboard may forfeit your opportunity to present or demonstrate your work
- failure to attend the presentation or demonstration on time may result in 0 marks for that component of assessed work
- during the presentation/demonstration, if asked, you must be able to explain in detail your work, otherwise this may result in 0 marks for that component of assessed work

Any questions about the assessment should be directed to the module leader

Assignment Specification

Assessment Requirements:
You need to submit the following via LMS:

- A zip file containing your code (the database, the PHP theme files(if used), the PHP files and the database for PHP).
- An SRS Report (detailed below)
- You need to attend a 10-minute demonstration.

Scenario

You will need to come up with an Ecommerce Application based on a business's needs. You will need to identify the business's needs, the challenges the system must address, and the desired product.

Next, you should develop an SRS that includes the project plan, detailing the scope and timeline of the project.

The timeline should clearly outline the required tasks and the timeline for completion. All tasks should be tracked and monitored, and **any changes to the plan should be communicated to the lecturer**.

Once the development tasks are complete, you should conduct a quality assurance review.

This review should include functionality testing, usability testing, performance testing and UI/UX (User experience testing based on the mobile UI and layout). Once the system meets the desired objectives, documentation must be created to explain the system's process, features and objectives.

Constraints

You should **discuss your Ecommerce business model with the lecturer and get his / her approval before starting the initial architecture**.

The business models can vary from

- Retail and consumer goods
- Health and wellness
- Food and Beverages
- Real estate
- Luxury Goods
- Pet Supplies
- Art and collectibles

You have the freedom to choose a suitable business model for your product.

The selected business model must have the following essential components.

- **A product or service.**
- **A revenue model.**
- **A requirement to store customer details for future reference.**

Task 1

You are required to write up an SRS that meets the requirements stated in the table below:

Criteria	Marks
Business Scenario outcomes (Product or service, revenue model etc)	5
Wireframes and UI Mock-ups	5
UML Diagrams (Use case, Activity), ER Diagram	5
Test Cases	5
TOTAL	20

Marking Allocation for Task 1 (Based on 5 Marks)

	0 – 1 Unsatisfactory	2 satisfactory	3 good attempt	4 very good attempt	5 excellent attempt
Business Scenario outcomes (Product or service, revenue model etc))(5)	• **0-1 marks**: Unclear or Incomplete Business Scenario: The project lacks a clear and comprehensive business scenario, including the definition of the product or service offered and the revenue model. Key aspects such as target market, value proposition, and competitive analysis are missing or poorly defined.	Basic Business Scenario: The project provides a basic outline of the product or service offered and the revenue model. However, key details such as target market segmentation, pricing strategy, and revenue generation channels may be vague or underdeveloped. There's room for improvement in defining the business's value proposition and differentiation from competitors.	Adequate Business Scenario: The project outlines the product or service offered and the revenue model with reasonable clarity. Key aspects such as target market, pricing strategy, and revenue generation channels are defined, but may lack depth or specificity. The business's value proposition is clear, but there's room for further refinement in defining competitive advantages.	Well-Defined Business Scenario: The project presents a well-defined business scenario, including a clear description of the product or service offered, target market segmentation, pricing strategy, and revenue generation channels. The business's value proposition is compelling, and there's evidence of thorough market research and competitive analysis. The revenue model is well-structured and aligned with the business's objectives.	Exceptional Business Scenario: The project exceeds expectations in defining the business scenario, demonstrating a deep understanding of the product or service offered, target market needs, and competitive landscape. The revenue model is innovative and well-suited to capitalize on market opportunities. The business's value proposition is compelling and effectively differentiates it from competitors. Overall, the business scenario presents a clear and compelling vision for the project's success.

Wireframes and UI Mock-ups	• 0-1 marks: Inadequate or Missing Wireframes and Mock-ups: The project lacks wireframes and UI mock-ups, or they are incomplete, inconsistent, or of poor quality. There's little evidence of planning or consideration given to the user interface design.	Basic Wireframes and Mock-ups: The project includes basic wireframes and UI mock-ups, but they lack detail, polish, or consistency. They may provide a general idea of the layout and structure but fail to capture key design elements or user interactions effectively.	Adequate Wireframes and Mock-ups: The project presents adequate wireframes and UI mock-ups that effectively convey the layout, structure, and functionality of the application. They demonstrate a reasonable level of detail and consistency in design elements, allowing stakeholders to visualize the user experience.	Well-Developed Wireframes and Mock-ups: The project includes well-developed wireframes and UI mock-ups that demonstrate a thoughtful approach to user interface design. They capture key design elements, user interactions, and visual aesthetics effectively, providing stakeholders with a clear understanding of the intended user experience.	Exceptional Wireframes and Mock-ups: The project exceeds expectations in wireframes and UI mock-ups, showcasing exceptional attention to detail, creativity, and usability. They demonstrate innovative design solutions, intuitive user interactions, and visually appealing aesthetics, elevating the overall quality of the user experience.

UML Diagrams	• **0-1 marks**: Inadequate or Missing UML Diagrams: The project lacks UML diagrams altogether, or they are incomplete, inaccurate, or of poor quality. There's little evidence of planning or consideration given to system architecture or database design.	Basic UML Diagrams: The project includes basic UML diagrams, but they lack detail, clarity, or consistency. Use case diagrams may have missing actors or unclear relationships, while activity diagrams may lack sequence details or decision points. The ER diagram may be missing essential entities, relationships, or attributes.	Adequate UML Diagrams: The project presents adequate UML diagrams that effectively capture system requirements, user interactions, and database design. Use case diagrams clearly define actors, use cases, and relationships. Activity diagrams illustrate system processes and user workflows logically. The ER diagram identifies relevant entities, relationships, and attributes, providing a solid foundation for database design.	Well-Developed UML Diagrams: The project includes well-developed UML diagrams that demonstrate a thorough understanding of system architecture and database design. Use case diagrams are comprehensive, capturing all relevant actors, use cases, and relationships accurately. Activity diagrams provide detailed representations of system processes, including decision points and loops. The ER diagram reflects a well-designed database schema, with clear entity definitions, relationships, and attribute specifications.	Exceptional UML Diagrams: The project exceeds expectations in UML diagrams, showcasing exceptional clarity, completeness, and coherence in system architecture and database design. Use case diagrams present a comprehensive overview of system functionality, including alternate and exception flows. Activity diagrams offer detailed insights into system processes, highlighting complex interactions and dependencies effectively. The ER diagram demonstrates a well-structured database schema, with optimized entity relationships and attribute definitions. Overall, stakeholders are impressed by the professionalism and accuracy of the UML diagrams, contributing to a clear and compelling

					understanding of the project requirements and design.
Test Cases	• 0-1 marks: Inadequate or Missing Test Cases: The project lacks test cases altogether, or they are incomplete, inaccurate, or poorly defined. There's little evidence of testing strategy or consideration given to system validation.	Basic Test Cases: The project includes basic test cases, but they lack detail, coverage, or clarity. Test cases may overlook critical scenarios or fail to address edge cases adequately. There's limited evidence of systematic testing procedures.	• Adequate Test Cases: The project presents adequate test cases that cover essential functionality and user scenarios. Test cases are reasonably detailed and provide coverage for common use cases. There's evidence of systematic testing procedures, such as positive and negative testing.	Well-Developed Test Cases: The project includes well-developed test cases that demonstrate a thorough understanding of system requirements and user workflows. Test cases are comprehensive, covering a wide range of scenarios, including edge cases and exceptional flows. There's evidence of systematic testing procedures, such as boundary value analysis and equivalence partitioning.	Exceptional Test Cases: The project exceeds expectations in test cases, showcasing exceptional clarity, completeness, and coherence in testing strategy. Test cases are meticulously designed to cover all aspects of system functionality, including complex interactions and dependencies. There's evidence of advanced testing techniques, such as stress testing and usability testing. Overall, stakeholders are impressed by the professionalism and effectiveness of the test cases, providing confidence in the quality and reliability of the system.

Task 2

You are required to build the application using **PHP**, **MySQL/ SQLite**, and **Tailwind** that meet the requirements stated in the table below.

Criteria	**Marks**
has a database to hold information	10
Meets specific criteria given by the business scenario:	**60**

• Crud operations based on scenario (40)
  ○ Ability to create (10)
  ○ Ability to update (10)
  ○ Ability to Delete (10)
  ○ Ability to view (10)
• Authenticated Routes(20)
  ○ Separate auth pages based on user roles(10)
  ○ Secure pages displayed with content based on the logged in user(10)
• Landing page(10)	

create an additional class using Tailwind's @layer components and uses it for styling(such as adding new fonts) & configure the application by changing the config file to customize the theme colour	5
Discretionary mark:	5
-Secure pages displayed with content based on the logged in user	
- suitable content for the web application	
- uses Tailwind to style the web application for both mobile and desktop (responsive)	
- your use of **MVC** design pattern in the entire project	
- your understanding of the PHP and Tailwind development process	
TOTAL	80

Mark allocations for Task 2 (Out of 10 marks)

	0 – 3 Unsatisfactory	4 satisfactory	5 good attempt	6 very good attempt	7-10 excellent attempt
has a database to hold information	• **0**: No Database: The project lacks any implementation of a database. There's no integration with a database management system, and therefore, no data persistence. • **1-2**: Database Present, but Incomplete: While there might be some indication of database usage, it's not fully implemented. Perhaps there are tables created, but they lack relationships or constraints. Or maybe there's an attempt at integrating a database, but it's not functional or incomplete. • **3**: Has Tables but No Relationships or Constraints: The database exists, and there might be tables created, but there are no established relationships between tables or constraints to maintain data integrity. This	Basic Database Setup: The project includes a functional database with tables, relationships, and constraints. However, the implementation might be simplistic or lack optimization. It fulfills the basic requirement of data storage but may not be robust or scalable.	Adequate Database Design: The database design demonstrates a solid understanding of relational database concepts. Tables are appropriately structured, relationships are established where necessary, and constraints are in place to maintain data integrity. The database adequately supports the requirements of the application, but there may be room for optimization or refinement.	Well-Designed Database: The database design is well-thought-out and optimized for efficiency and scalability. Tables are normalized to reduce redundancy, relationships are properly defined, and constraints are utilized effectively. The database architecture aligns closely with the requirements of the application and demonstrates a high level of proficiency in database management.	Exceptional Database Implementation: The database implementation exceeds expectations and sets a high standard for quality and efficiency. It demonstrates advanced techniques for optimization, such as indexing, partitioning, or denormalization where appropriate. The database design is robust, flexible, and scalable, capable of handling large volumes of data without sacrificing performance. Overall, it showcases a mastery of database management principles and significantly enhances the functionality and reliability of the application.

	0 – 3 Unsatisfactory	4 satisfactory	5 good attempt	6 very good attempt	7-10 excellent attempt
	setup is prone to data inconsistencies and might not fully meet the needs of the application.				
Crud operations based on scenario	• **0**: No CRUD Functionality: The project lacks any implementation of CRUD operations. There's no functionality to create, read, update, or delete records. • **1-2**: CRUD Partially Implemented: While there might be some attempt at implementing CRUD operations, it's not fully functional or complete. Some operations may be missing or only partially implemented. • **3**: Basic CRUD Functionality: CRUD operations are implemented at a basic level, but there are significant deficiencies or errors in functionality. For example, records may be created but not updated or	Basic CRUD Implementation: CRUD operations are implemented with basic functionality. Users can create, read, update, and delete records, but the implementation may lack robust error handling or validation. The functionality meets minimum requirements but may not be user-friendly or efficient.	Adequate CRUD Functionality: CRUD operations are implemented effectively and meet the requirements of the scenario. Users can perform create, read, update, and delete operations with appropriate validation and error handling. The functionality is reliable and user-friendly, but there may be some room for improvement in terms of efficiency or additional features.	Well-Executed CRUD Operations: CRUD operations are implemented efficiently and with attention to detail. The functionality is user-friendly, with intuitive interfaces for creating, updating, deleting, and viewing records. Error handling and validation are robust, ensuring data integrity and security. The implementation demonstrates a high level of proficiency in CRUD development.	Exceptional CRUD Implementation: The CRUD functionality exceeds expectations and sets a high standard for quality and usability. The implementation showcases advanced features such as batch operations, data pagination, or real-time updates where applicable. User interfaces are polished and responsive, providing a seamless experience for users. Overall, the CRUD functionality significantly enhances the usability and effectiveness of the application, demonstrating mastery in application development.

	0 – 3 Unsatisfactory	4 satisfactory	5 good attempt	6 very good attempt	7-10 excellent attempt
	deleted properly, or viewing records may not display accurate data.				
Authenticated Routes	• 0: No Authentication Implemented: The project lacks any implementation of authentication. There are no mechanisms in place to authenticate users or restrict access to certain pages or features. • 1-2: Authentication Partially Implemented: While there might be some attempt at implementing authentication, it's not fully functional or secure. Authentication mechanisms may be incomplete or poorly implemented, leading to vulnerabilities or inconsistencies. • 3: Basic Authentication Implementation: Authentication is implemented at a basic level, but there are significant	Basic Authentication Setup: Authentication mechanisms are in place to verify user identities and restrict access to authenticated users. However, the implementation may lack robustness or flexibility. User roles may not be clearly defined, and security measures such as encryption may be minimal.	Adequate Authentication Functionality: Authentication mechanisms are implemented effectively and meet the requirements of the scenario. Users can log in with appropriate credentials, and access to pages or features is restricted based on user roles. Security measures such as password hashing and session management are implemented, but there may be room for improvement in terms of usability or additional security features.	Well-Executed Authentication System: The authentication system is well-designed and implemented with attention to security and usability. User roles are clearly defined, and access permissions are enforced accurately. Security measures such as two-factor authentication or password policies may be implemented to enhance security further. The authentication system demonstrates a high level of proficiency in user authentication and access control.	Exceptional Authentication Implementation: The authentication system exceeds expectations and sets a high standard for security and usability. Advanced features such as role-based access control (RBAC), OAuth integration, or single sign-on (SSO) may be implemented to provide a seamless and secure user experience. Security measures are comprehensive, addressing common vulnerabilities such as cross-site scripting (XSS) or cross-site request forgery (CSRF). Overall, the authentication system significantly enhances the security and usability of the application, demonstrating mastery in

	0 – 3 Unsatisfactory	4 satisfactory	5 good attempt	6 very good attempt	7-10 excellent attempt
	deficiencies or security risks. For example, there may be no distinction between user roles, or authentication may rely on insecure practices such as storing passwords in plain text.				authentication and access control.
Landing page	• **0**: No Landing Page: The project lacks any implementation of a landing page. Users are not greeted with any introductory or informative content upon visiting the application. • **1-2**: Landing Page Partially Implemented: While there might be some attempt at implementing a landing page, it's incomplete or poorly designed. The landing page may lack essential elements such as navigation, branding, or relevant content. • **3**: Basic Landing Page: A basic landing page is present, but it lacks polish and	Basic Landing Page Setup: The landing page provides essential information about the application and its features. It includes elements such as a brief introduction, navigation options, and calls to action. However, the design may be simplistic, and content may lack depth or customization.	Adequate Landing Page Design: The landing page is well-designed and effectively communicates the purpose and benefits of the application. Content is engaging and relevant, capturing the user's attention and encouraging further exploration. Visual elements such as images or videos may be used to enhance the presentation.	Well-Executed Landing Page: The landing page is visually appealing and professionally designed, leaving a positive first impression on users. Content is comprehensive and tailored to the target audience, addressing their needs and pain points effectively. The layout is intuitive, guiding users to key features or actions seamlessly.	Exceptional Landing Page Implementation: The landing page exceeds expectations and serves as a powerful marketing tool for the application. It incorporates advanced design techniques, such as animation or interactive elements, to captivate users' attention and encourage engagement. Content is optimized for conversion, leading users towards specific actions such as signing up or exploring key features. Overall, the landing page significantly enhances the user experience and contributes

	0 – 3 Unsatisfactory	4 satisfactory	5 good attempt	6 very good attempt	7-10 excellent attempt
	fails to engage users effectively. Content may be sparse or irrelevant, and the design may be uninspiring or inconsistent with the rest of the application.				to the success of the application, demonstrating mastery in web design and user engagement.

Mark allocations for Task 2 (Out of 5 marks)

	0 – 1 Unsatisfactory	2 satisfactory	3 good attempt	4 very good attempt	5 excellent attempt
create an additional class using Tailwind's @layer components and uses it for styling(such as adding new fonts) & configure the application by changing the config file to customize the theme colour	• **0**: No Tailwind Customization: The project lacks any customization of Tailwind CSS. There's no evidence of using @layer components or configuring the application's theme color. • **1**: Basic Tailwind Customization: There's some attempt at customization, but it's minimal or incomplete. Perhaps there's a basic customization of colors or fonts, but it's not implemented using @layer components or configuration files.	Adequate Tailwind Customization: Tailwind CSS is customized to some extent, with evidence of using @layer components or configuration files. For example, there might be a new font added using @layer components, or the theme color might be changed through the configuration file	Well-Executed Tailwind Customization: Tailwind CSS is customized effectively, with attention to detail and aesthetic appeal. The @layer components are utilized creatively for styling, such as adding new fonts or defining custom utility classes. The application's theme color is customized in the configuration file, enhancing the visual coherence of the UI.	Exceptional Tailwind Customization: The Tailwind CSS customization exceeds expectations, showcasing advanced techniques and creativity. The @layer components are leveraged innovatively to implement complex styling requirements, such as typography adjustments or responsive layouts. The configuration file is utilized extensively to customize various aspects of the application's design, resulting in a visually	Outstanding Tailwind Customization: The Tailwind CSS customization is exemplary, setting a high standard for design and aesthetics. Innovative use of @layer components and configuration files demonstrates mastery in Tailwind CSS customization, with attention to detail and consistency across the application. The customized styling significantly enhances the user experience and contributes to the overall quality and

	0 – 1 Unsatisfactory	2 satisfactory	3 good attempt	4 very good attempt	5 excellent attempt
				stunning and cohesive user interface.	appeal of the project.
Discretionary Mark	• 0: No Implementation or UnderstandingThe project lacks implementation or understanding in multiple aspects. There's either no evidence of secure pages, relevant content, responsive styling, MVC architecture, or understanding of PHP and Tailwind development processes. The project fails to meet basic requirements in these areas. • 1: Basic Implementation or Understanding There's a basic attempt at implementation or understanding in some aspects, but it's limited or incomplete. For example, there might be secure pages, but content relevance is lacking, or responsive styling is implemented but not fully optimized. MVC architecture may be partially implemented, but there's room for improvement. Understanding of PHP and Tailwind development processes is basic.	Adequate Implementation and Understanding There's evidence of adequate implementation or understanding in most aspects. Secure pages are implemented with some level of content personalization. Content relevance is reasonable, and responsive styling is present, though it may not be perfect. MVC architecture is implemented adequately, and there's a decent understanding of PHP and Tailwind development processes.	Adequate Implementation and Understanding: There's evidence of adequate implementation or understanding in most aspects. Secure pages are implemented with some level of content personalization. Content relevance is reasonable, and responsive styling is present, though it may not be perfect. MVC architecture is implemented adequately, and there's a decent understanding of PHP and Tailwind development processes..	Well-Executed Implementation and Understanding: The project demonstrates a well-executed implementation and understanding across all aspects. Secure pages are implemented effectively with personalized content. Content is highly relevant and engaging, and responsive styling is well-executed across various devices. MVC architecture is implemented correctly, contributing to code organization and maintainability. There's a strong understanding of PHP and Tailwind development processes.	Exceptional Implementation and Understanding: The project exceeds expectations in implementation and understanding across all aspects. Secure pages are implemented flawlessly, with personalized and secure content delivery. Content is highly relevant, engaging, and tailored to user needs. Responsive styling is exceptional, providing a seamless user experience across all devices. MVC architecture is implemented with excellence, resulting in a well-structured and maintainable codebase. There's a deep understanding of PHP and Tailwind development processes, leading to innovative and

	0 – 1 Unsatisfactory	2 satisfactory	3 good attempt	4 very good attempt	5 excellent attempt
					efficient solutions.

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 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 the system during your allocated timeslot will automatically give you zero marks, regardless of whether the work has been submitted.**

The database you choose to build the application on must be based on either SQLite or MySQL. If SQLite is used, the DB file must be attached; if MySQL is used, an SQL dump must be attached to the ZIP submission file.