.Program Content

Semester	**III**		
Course Code:	**IT3406**		
Course Name:	**Web Application Development II**		
Credit Value:	**4**		
Core/Optional	**Core**		
Hourly Breakdown	Theory	Practical	Independent Learning
	45 hrs.	30 hrs.	125 hrs.

Course Aim:

- This module on web application development provides an insight to the server side web development technologies along with the advances features, methods and tools needed to add interactivity to produce rich internet applications.

Intended Learning Outcomes:

After successfully completing this course, students should be able to

- Describe and apply the fundamental and advanced concepts of PHP.
- Describe the MVC architecture.
- Employ Advanced features of client-side programming using JavaScript and Ajax to add interactivity to web pages.
- Employ JavaScript libraries in web pages.
- Describe and employ fundamental features of web security.

Course Content: (Main Topics, Sub topics)

Topic	Theory (Hrs)	Practical (Hrs.)
1. Server Side Web Development using PHP & MySQL	15	10
2. Fundamentals of Asynchronous JavaScript and XML (AJAX)	15	8
3. Advanced Client Side Development	9	8
4. Introduction to web application security	6	4
Total	45	30

1. Server Side Web Development using PHP & MySQL (15 hours)

Theoretical Aspects

1.1. Explain basic features of PHP [Ref 1: Pg. (303-304)] [Ref 10: Pg. (271-316)]
1.2. Articulate and Explain MVC architecture [Ref 1: Pg. (683)] [Ref 3: Pg. (1-4)]
1.3. Differentiate available PHP frameworks [Ref 3: Pg. (345-441)]/[Ref. 4: Pg. (83-105)]
1.4. Consider server-side security threats [Ref 1: Pg. (375)] [Ref 10: Pg. (425-436)]
  1.4.1. Data spoofing [Ref 1: Pg. (376)]
  1.4.2. Invalid data [Ref 1: Pg. (379-380)]
  1.4.3.Unauthorized file access [Ref 1: Pg. (382)]
1.5. Vulnerability solutions [Ref 1: Pg. (384-389)]

1.5.1.Data sanitizing [Ref 1: Pg. (384)]
1.5.2.Data validation [Ref 1: Pg. (389)]

****Guided Practicals**

1.1. Install PHP in a windows/ Linux environment [Ref 10: Pg. (40-54)]
1.2. Explain MVC architecture and differentiate available PHP frameworks [Ref 1: Pg. (683)] [Ref 3: Pg. (1-4)] [Ref 3: Pg. (345-441)]/[Ref. 4: Pg. (83-105)]
1.3. Explain basic features of PHP [Ref 1: Pg. (310-323, 325-343)] [Ref 10: Pg. (271-316)]
  1.3.1. PHP syntax and semantics
    1.3.1.1. Variables [Ref 1: Pg. (311)] [Ref 10 : Pg. (288-296)]
    1.3.1.2. Constants [Ref 10 : Pg. (287)]
    1.3.1.3. Conditional statements [Ref 1: Pg. (325-330)]
    1.3.1.4. Loops [Ref 1: Pg. (331-334)]
    1.3.1.5. Functions [Ref 1: Pg. (336)]
  1.3.2. Arrays and data processing with arrays [Ref 1: Pg. (206)] [Ref 10 : Pg. (296-305)]
  1.3.3. Handling HTML forms with GET and POST operations [Ref 1: Pg. (343)]
  1.3.4. Form validation fields (including URLs and email address) and required fields [Ref 10: Pg. (574-585)]
  1.3.5. Filtering inputs (validate and sanitize external inputs) [Ref 1: Pg. (384-389)] [Ref 10: Pg. (432)]
  1.3.6. Session control and cookies (create and retrieve a cookie) PHP [Ref 1: Pg.(419-435)][Ref 10: Pg. (437-446)]
  1.3.7. File handling (Open, read, create, write operations with files, upload files) PHP [Ref 10: Pg. (366-368)]
  1.3.8. Sending emails using PHP [Ref 11]
  1.3.9. Object Orientation with PHP [Ref 1. Pg. (395-418)]
1.4. Use web services with PHP [Ref 10: Pg. (541-553)]
1.5. Develop a web application with PHP [Ref 9: Pg. (604-636)]
1.6. Setting up MySQL [Ref 1: Pg. (470-474)] [Ref 10: Pg. (56-74)]
1.7. Connect to MySQL database [Ref 1: Pg. (513)] [Ref 10: Pg. (515-527)] [Ref 2: Pg. (165)]
1.8. MYSQL database operations - Read/modify/delete/search operations [Ref 1: Pg. (497-513)] [Ref 2: Pg. (206)]
  1.8.1.Processing forms (Create, Read/Retrieve, Update, and Delete operations) [Ref 2: Pg. (235-250)]
1.9. Use of PHP unit testing tools for testing automation of front end web applications and manual testing of applications. [Ref 1: Pg. (616)]
1.10. Consider server-side security threats eg: data spoofing, invalid data, unauthorized file access [Ref 1: Pg. (375-382)][Ref 10: Pg. (425-436)]
1.11. Handle vulnerability solutions eg: data sanitizing and validation [Ref 1: Pg. (384-389)]

At the end of the section, students are guided to complete following **mini project**

- Create a basic PHP form that contains the following:
  - Consist of GET and POST operations.
  - Contains basic field validation, required field validation and validate email address, mobile number and URLs,.. etc
  - Once user submit button is pressed, how to keep the values of input field.

- Process form by combining MySQL database to perform (Create, Read/Retrieve, Update, and Delete operations).
- Test the front end of the PHP application using Selenium PHP unit testing automation tool.
- Write test cases and carry out manual testing.

2. Fundamentals of Asynchronous JavaScript and XML (AJAX) (15 hours)

Theoretical Aspects

2.1. Explain, and employ the traditional technologies used in web Application development [Ref 9: Pg. (1-10)]
2.2. Discuss and differentiate AJAX and Non-AJAX Applications [Ref 9: Pg. (405-420]
  2.2.1. Discuss serialized data in a structured formats eg: XML/JSON both synchronous and asynchronous
2.3. Discuss industry standard tools and technologies for web development [Ref 3: Pg. (43-65)]
2.4. Discuss pros and cons of development frameworks for web development [Ref 1: Pg. (695-710)] [Ref 3: Pg. (83-109)]

****Guided Practicals**

2.1. Explain and differentiate the traditional technologies used in web Application development [Ref 9: Pg. (1-10)]
2.2. Creating a Simple AJAX application [Ref 1: Pg. (619-643)] [Ref 9: Pg. (408-411)] [Ref 5] [Ref 8]
  2.2.1. Differenticate AJAX and Non-AJAX applications [Ref 9: Pg. (405-420]
  2.2.2. Basic AJAX connection [Ref 12: Pg. (227-233)]
  2.2.3. Using jQuery AJAX library [Ref 1: Pg. (629-634)]
  2.2.4. Using XML in PHP [Ref 1: Pg. (636)]
  2.2.5. Using XML in JavaScript [Ref 1: Pg. (640)]
  2.2.6. Dynamic Hypertex Markup Language (DHTML)
2.3. Basic AJAX functionalities
  2.3.1. AJAX XMLHttp [Ref 12: Pg. (230)] [Ref 12]
  2.3.2. AJAX Request [Ref 9: Pg. (491)] [Ref 13]
  2.3.3. AJAX Response [Ref 14]
2.4. Develop a webpage employing PHP, and AJAX [Ref 9: Pg. (408-417)]
2.5. Discuss industry standard tools and technologies for web development [Ref 3: Pg. (43-65)]
2.6. Use collaboration tools such as GitHub to work with a team on web [Ref 3: Pg. (1-21)]

At the end of the section, students are guided to complete following **mini project**

- To the previously created PHP form,
  - show how a web page can communicate with a web server while a user type characters in an input field.
  - Fetch information from a database with AJAX.
  - Interactive communication with an XML file.
  - To handle interactive serches (live search where results are getting live while user types).
  - Demostrate poll results without reloading the page.

3. Advanced Client Side Development (9 hours)

Theoretical Aspects

3.1. Use JavaScript to add new features to make more richer and compelling user interface on web pages[Ref 1: Pg. (195)]
3.2. Use JavaScript libraries to Manipulate Document Object Model [Ref 1: Pg. (223)]
3.3. Employ AJAX and JavaScript together in a website[Ref 1: Pg. (619)]
  3.3.1. Understand Single Page Application development
3.4. Consider client-side security threats [Ref 9: Pg. (317-318)]
3.5. Evaluate design and architecture of a web considering issues such as design pattern (including MVC), and tradeoff between redundancy, scalability, state management and search engine optimizations [Ref 1: Pg. (683-694)]

**Guided Practicals

3.1. Understanding JavaScript basics [Ref 1: Pg. (195-222)] [Ref 2: Pg. (329-346)] [Ref 10: Pg. (187-190)]
  3.1.1. Understanding Document Object Model [Ref 1: Pg. (223)] [Ref 10 Pg:(208-217)]
    3.1.1.1. Working with nodes and elements [Ref 1: Pg. (223-233),(259-261)]
    3.1.1.2. Working with the Document Object Model [Ref 1: Pg. (238)]
3.2. Use JavaScript Libraries [Ref 10 Pg: (219-240)]
  3.2.1. Introduction to JavaScript libraries [Ref 10 Pg: (219-240)]
  3.2.2. Working with Events and Event Listeners [Ref 10 Pg: (241-254)]
  3.2.3. Understanding events [Ref 10 Pg: (241-254)]
  3.2.4. jQuery fundamentals [Ref 1 Pg: (243-247)] [Ref 10 Pg: (219-232)]
    3.2.4.1. Adding jQuery, reacting to events with JavaScript and jQuery [Ref 10: Pg. (219-259)]
  3.2.5. Working with Forms [Ref 10 Pg: (242-247)]
  3.2.6. Keyboard events [Ref 10 Pg: (247-254)]
3.3. Building a JavaScript program [Ref 10: Pg. (191-214)]
  3.3.1. Employ AJAX and JavaScript together in a website[Ref 1: Pg. (619)]
3.4. Handle client-side security threats [Ref 9: Pg. (317-318)]
  3.4.1. Validate data inputs on the client side [Ref 2: Pg. (371-387)] [Ref 9: Pg. (381-402)]
3.5. Express constraints involved in state management (cookies, query string, sessions) in the web [Ref 2: Pg. (287-304)] [Ref 9: Pg. (301-320)]
3.6. Employing development framework such as jQuery, Angular, ASP.NET MVC, WordPress etc. [Ref 1: Pg. (651-677,695-713)] [Ref 10: Pg. (219-240)]

At the end of the section, students are guided to complete following **mini project**

- To the previously created PHP form in section 2,
  - Add HTML form validations via Javascript.

4. Introduction to web security (6 Hrs)

Theoretical Aspects

4.1. Produce secure web application by utilizing authentication, secure certificate, encryption, hashing, cookies and sessions. [Ref 2: Pg. (367-391)] [Ref 1: Pg. (419-428)]
- 4.1.1.Setting, accessing and destroying a cookie [Ref 1: Pg. (419-428)]
- 4.1.2. HTTP Authentication - Storing Usernames and passwords [Ref 4: Pg. (53-84)]
- 4.1.3. Using sessions – Starting and ending sessions, session security and timeout [Ref 1: Pg. (430-435)] [Ref 9: Pg. (432-446)]

4.2. Able to discuss on common types of vulnerabilities and attacks in web [Ref 4: Pg. (11-19)]
- 4.2.1.Injection [Ref 4: Pg. (13)]
- 4.2.2.Cross-site scripting [Ref 4: Pg. (13)]
- 4.2.3.Broken authentication and session management [Ref 4: Pg. (14)]
- 4.2.4.Security misconfiguration [Ref 4: Pg. (16)]
- 4.2.5.Insecure cryptographic storage [Ref 4: Pg. (16)]
- 4.2.6.Failure to restrict URL access [Ref 4: Pg. (17)]
- 4.2.7.Unvalidated redirects and forwards [Ref 4: Pg. (19)]

4.3. Differentiate client security and server security [Ref 9: Pg. (317-318)] [Ref 10: Pg. (425)]
- 4.3.1.Securing server and client machines [Ref 10: Pg. (425)]
- 4.3.2.Securing client application and apache web server [Ref 10: Pg. (425-426)]
- 4.3.3.Configure PHP securely [Ref 10: Pg. (425-426)]
- 4.3.4.Handling errors safely [Ref 10: Pg. (429-431)]
- 4.3.5.Sanitizing variables [Ref 10: Pg. (432-434)]

****Guided Practicals**

4.1. Illustrate browser security models including same-origin policy and threat models in web security [Ref 4: Pg. (35-44),Ref 4: Pg. (149-155)]

4.2. Employ how authentication, secure certificates, secure communication - SSL can be used in web sessions [Ref 1: Pg. (437-447)] and common type of vulnerabilities [Ref 4: Pg. (11-19)]
- 4.2.1.Setting, accessing and destroying a cookie [Ref 1: Pg. (419-428)]
- 4.2.2.HTTP Authentication - Storing Usernames and passwords [Ref 4: Pg. (53-84)]
- 4.2.3.Using sessions – Starting and ending sessions, session security and timeout [Ref 1: Pg. (430-435)] [Ref 9: Pg. (432-446)]

4.3. How sessions, cookies, encryption and hashing provide means of secure web application [Ref 2: Pg. (287-290,290-296,299-304)] [Ref 4: Pg. (367-391)]
- 4.3.1. Using Cookies in PHP [Ref 2: Pg. (287-290)]
  - 4.3.1.1. Setting a Cookie [Ref 2: Pg. (289)]
  - 4.3.1.2. Accessing a Cookie [Ref 2: Pg. (290)]
  - 4.3.1.3. Destroying a Cookie [Ref 2: Pg. (290)]
- 4.3.2.HTTP Authentication - Storing Usernames and passwords [Ref 2: Pg. (290-296)]
- 4.3.3.Using Sessions [Ref 2: Pg. (299-304)]
  - 4.3.3.1. Starting a Session [Ref 2: Pg. (299)]
  - 4.3.3.2. Ending a Session [Ref 2: Pg. (299)]
  - 4.3.3.3. Setting a Timeout [Ref 2: Pg. (303)]
  - 4.3.3.4. Session Security [Ref 2: Pg. (304)]
- 4.3.4.Hashing by using SHA1, MD5 [Ref 6,7]

4.4. Difference between http and https roles in web [Ref 4: Pg. (18-19)]

4.5. Handle client and server security [Ref 9: Pg. (317-318)] [Ref 10: Pg. (425)]

4.5.1.Configure PHP securely [Ref 10: Pg. (425-426)]
4.5.2.Handling errors safely [Ref 10: Pg. (429-431)]
4.5.3.Sanitizing variables [Ref 10: Pg. (432-434)]

At the end of the section, students are guided to complete following **mini project**

- To the previously created PHP form in section 2,
  - Include sessions, cookies and suitable hashing techniques to make the system more secure.

Teaching /Learning Methods:
You can access all learning materials and this syllabus in the VLE: http://vle.bit.lk/, if you are a registered student of the BIT degree program.

Assessment Strategy:

Continuous Assessments/Assignments:

The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). The maximum mark for a question is 10 and the minimum mark for a question is 0 (irrespective of negative scores). Final assignment mark is calculated considering both assignments, and students will have to obtain at least 40% for each assignment. Students are advised to complete online assignments before the given deadline. It is compulsory to pass all online assignments to qualify to obtain the Level II, Higher Diploma in IT (HDIT), certificate.

In the course, case studies/Lab sheets will be introduced, and students have to participate in the learning activities.

Final Exam:
Final examination of the course will be held at the end of the semester. The course is evaluated using a two hour question paper which consists of 25 MCQ (1 hour) and 2 Structured Questions (1 hour).

References/ Reading Materials:

Main references

- **Ref 1.** PHP, MySQL, & JavaScript All-in-One For Dummies Richard Blum, 2017 (Online source : https://www.pdfdrive.com/php-mysql-javascript-all-in-one-for-dummies-e90592496.html)
- **Ref 2.** Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, 5th Edition, O'Reilly Media, Inc. 2018 (online source : https://www.pdfdrive.com/learning-php-mysql-javascript-with-jquery-css-html5-e188490793.html) .

Supplimentry references

- **Ref 3.** PHP Development Tool Essentails, Chad Russell,2016 (online source: https://ikamy.ch/public/img/books/PHP+Development+Tool+Essentials.pdf)
- **Ref 4**. Web Application Security, A Beginner's GuideMcGraw-Hill Education;by Bryan Sullivan and Vincent Liu, 1st Edition (2011)
- **Ref 5.** https://www.w3schools.com/php/php_ajax_intro.asp

- **Ref 6.** https://www.w3schools.com/php/func_string_md5.asp
- **Ref 7.** https://www.w3schools.com/PHP/func_string_sha1.asp
- **Ref 8.** https://www.w3schools.com/php/
- **Ref 9.** Learning PHP, MySQL, JavaScript, and CSS, 2nd Edition, O'Reilly Media, Inc. 2012.
- **Ref 10.** PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies , John Wiley & Sons, Inc. 2013
- **Ref 11.** https://www.w3schools.com/php/php_form_url_email.asp
- **Ref 12** JavaScript and AJAX For Dummies,Andy Harris, 2009
- **Ref 13.** https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp
- **Ref 14.** https://www.w3schools.com/xml/ajax_xmlhttprequest_send.asp
- **Ref 15.** https://www.w3schools.com/xml/ajax_xmlhttprequest_response.asp