<table>
<tr><td>Course Title</td><td colspan="5">Web Engineering</td></tr>
<tr><td>Course Code</td><td colspan="5">WSS502</td></tr>
<tr><td>Course Type</td><td colspan="5">Compulsory (Both Specializations)</td></tr>
<tr><td>Level</td><td colspan="5">MSc (Level 2)</td></tr>
<tr><td>Year / Semester</td><td colspan="5">1 / 1</td></tr>
<tr><td>Teacher’s Name</td><td colspan="5">Achilleas Achilleos, PhD</td></tr>
<tr><td>ECTS</td><td>10</td><td>Lectures / week</td><td>3</td><td>Laboratories/week</td><td>0</td></tr>
<tr><td>Course Purpose</td><td colspan="5">The aim of this course is to provide students with critical understanding on how to productively engineer Web Applications. The course addresses the concepts, methods, technologies and tools for developing advanced Web applications. Topics covered include evolution of the Internet and the Web, Web technologies and the basic models and architectures in a Web environment. It kicks-off with a theoretical examination of web engineering and its phases. It proceeds to introduce client-side technologies (HTML5, CSS, JavaScript, jQuery, BootStrap.js) and server-side technologies (PHP, JSP, Java JAX-RS) that support the development of static web sites, dynamic web applications and rich internet applications. It concludes with an overview of cutting-edge concepts, technologies and frameworks (MEAN stack – MongoDB, Express.js, Angular.js, Node.js) that support full stack web development. The course has a theoretical underpinning, but the focus is on practical examples that reveal how to engineer Web applications using the introduced implementation methods, techniques and tools.</td></tr>
<tr><td>Learning Outcomes</td><td colspan="5">Upon successful completion of the course students will be able to:
• Describe and explain the concepts, principles and software processes that are used for engineering web applications.
• Outline the stages of the web evolution and describe the conceptual and technological differences of Web 1.0, Web 2.0 and Web 3.0.
• Gain theoretical knowledge and analytical skills to develop dynamic web sites and advanced internet applications.
• Utilise different Web technologies and programming languages in theory and through practical exercises.
• Apply these concepts, processes and technologies (HTML, CSS, JavaScript, jQuery, BootStrap, PHP, JSP, JAX-RS) to develop Rich Internet Applications.
• Understand the differences and benefits of Full Stack Web Development in comparison to the above web concepts, processes and technologies.
• Outline and explain the purpose and functionality of the MEAN stack
– MongoDB, Express.js, Angular.js and Node.js – frameworks.</td></tr>
<tr><td>Prerequisites</td><td colspan="2"><b>None.</b></td><td colspan="2">Corequisites</td><td><b>None.</b></td></tr>
</table>

*One hour out of the three is normally devoted to laboratory-based programming exercises.

*One hour out of the three is normally devoted to laboratory-based programming exercises.

Course Content	**1. Introduction to Web Engineering and Web 1.0 Technologies (2 Weeks)** - Engineering Web Applications. Web Application Features. Processes and Models. Concepts, principles and methods of Web engineering. Web Evolution Challenges and Drivers. Static Web Sites: Enabling Content (HTML5) and Design (CSS). Dynamic Web Applications: Client-Side Scripting (JavaScript, jQuery). Building Web Applications with HTML5, CSS, JavaScript and jQuery. **2. Developing Rich Internet Applications: Dynamic Languages (3 Weeks)** - Basic overview of 3-Tier Architecture and Technologies: Server-Side Scripting (PHP) and the Model View Controller Pattern (MVC). Developing RIAs with PHP, MVC and MySQL. Basic Overview of 4-Tier Architecture and Technologies: Java Servlets, Java Server Pages (JSP) and Active Server Pages (ASP). Enterprise Application Architectures. Introduction to PHP/MVC Frameworks (e.g., Laravel, CakePHP). **3. Web 2.0 and Web 3.0: Web Services, Cloud Computing and the Internet of Things (5 Weeks)** - The XML Meta-Language. AJAX (Asynchronous JavaScript and XML). JavaScript and AJAX. Web Services. The WS*- Stack Web Services Model. The REST Architectural Pattern. Developing Rich Internet Applications with AJAX and RESTful Services. Introduction to Cloud Computing. Benefits of Cloud Computing. Installing OpenStack Cloud on a Linux Server. Deploying a Virtual Machine with XAMPP on OpenStack Cloud. Developing and Deploying Rich Internet Applications on the OpenStack Cloud. Web 3.0: The Semantic Web and the Internet of Things. **4. Introduction to Full Web Stack Development (3 Weeks)** - Web Frameworks. Evolution of Web Technologies. Introduction to Front-End Web UI Frameworks (Bootstrap.js). Introduction to Client-Side JavaScript Frameworks (Angular.js). Introduction to NoSQL Databases (MariaDB). Introduction to Back-End JavaScript Frameworks (Node.js, Express.js). Developing a Web Application using the MEAN (MongoDB, Express, AngularJS, and Node.js) stack.		
Teaching Methodology	The methodology followed in this course is structured around lectures and laboratory exercises, so that students gain theoretical knowledge as well as practical skills. The taught part of course is delivered to the students with the help of computer presentations. Presentations are available through the e-learning system for students to use in combination with the textbooks. Furthermore, theoretical principles are explained by means of specific examples and solution of specific		

*One hour out of the three is normally devoted to laboratory-based programming exercises.

	problems using practical examples. The code for these programming examples is also made available in the e-learning system. Lectures are supplemented with supervised computer laboratories, which include demonstrations of taught concepts and experimentation with related technologies to solve specific problems via exercises. Hence, during laboratory sessions, students apply their gained knowledge and identify the principles taught in the lecture sessions by means of working on different tasks and solving domain-specific problems. The course includes a midterm test that involves both theoretical and critical thinking questions, as well as practical programming exercises. The midterm test is undertaken using the e-learning system. Also, a course project is assigned to the students since this is a practical-oriented web programming course. Finally, the course assessment is completed by means of a three-hours final exam at the end of the semester.
Bibliography	**Textbooks:** 1. Sven Casteleyn, Florian Daniel, Peter Dolog, Maristella Matera, “Engineering Web Applications”, Book: Data-Centric Systems and Applications, 2009, Springer-Verlag Berlin Heidelberg, ISBN: 9783540922001. 2. Bill Burke, “RESTful Java with JAX-RS 2.0: Designing and Developing Distributed Web Services”, Paperback: 392 pages, Publisher: O'Reilly Media; Second edition (December 2, 2013), Language: English, ISBN-10: 144936134X, ISBN-13: 978-1449361341 **References:** 1. Robin Nixon, “Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5”, 5th Edition, Series: Learning PHP, MYSQL, Javascript, CSS & HTML5, Paperback: 832 pages, Publisher: O'Reilly Media; 5 edition (June 8, 2018), Language: English, ISBN-10: 1491978910, ISBN-13: 978-1491978917. 2. Nicholas S. Williams, “Professional Java for Web Applications”, 1st Edition, Paperback: 936 pages, Publisher: Wrox; 1 edition (March 10, 2014), Language: English, ISBN-10: 1118656466, ISBN-13: 978-1118656464. 3. Simon Holmes, Clive Herber, “Getting MEAN with Mongo, Express, Angular, and Node”, 2nd Edition, Paperback: 504 pages, Publisher: Manning Publications; 2 edition (May 10, 2019), Language: English, ISBN-10: 1617294756, ISBN-13: 978-1617294754, Book examples code: LINK. 4. A series of entry-level tutorial on various Internet Technologies: www.w3schools.com. 5. The official site of the World Wide Web Consortium. Various references, RFCs and interesting reading material on the Internet development: www.w3c.org.

*One hour out of the three is normally devoted to laboratory-based programming exercises.

Assessment	- Midterm Test: 20% - Course Project: 30% - Final Exam: 50%
Language	English.

*One hour out of the three is normally devoted to laboratory-based programming exercises.