European Software
Skills Alliance.

Train the Trainer Programme

Annex I
Junior Developer EQF 4/5

30 November 2023

Status: Final version

Train the Trainer Programme – Annex I - Junior Developer EQF 4/5, 2023.

Deliverable 13: "ESSA Train the Trainer Programme & Materials" – ANNEX I

This document is a draft version and is subject to change after review coordinated by the European Education and Culture Executive Agency (EACEA).

Authors: Federica D'Armini (Adecco/Mylia)

Editors/Reviewers: Alessandro Prunesti (Adecco/Mylia), Chiara Longobardi (Adecco/Mylia), Carmel Somers (DTSL), Sabine Boesen – Mariani (Utrecht University of Applied Sciences), Niels Selling (Digital Europe), Ants Slid (BCS Koolitus), Jutta Breyer (Digital Europe), Paul Aertsen(Global Knowledge)

softwareskills.eu

Legal Disclaimer

Co-funded by the
Erasmus+ Programme
of the European Union

About ESSA

The European Software Skills Alliance (ESSA) is a four-year transnational project funded under the EU's Erasmus+ programme. It ensures the skills needs of the rapidly evolving Software sector can be met — today and tomorrow.

ESSA provides current and future software professionals, learning providers and organisations with software needs with the educational and training instruments they need to meet the demand for software skills in Europe.

ESSA will develop a European Software Skills Strategy and learning programmes for Europe. It will address skill mismatches and shortages by analysing the sector in depth and delivering future-proof curricula and mobility solutions; tailored to the European software sector's reality and needs.

Project partners

The ESSA consortium is led by DIGITALEUROPE. It is composed of academic and non-academic partners from the education, training, and software sectors.

View all project partners: ESSA Partners I ESSA Associated Partners

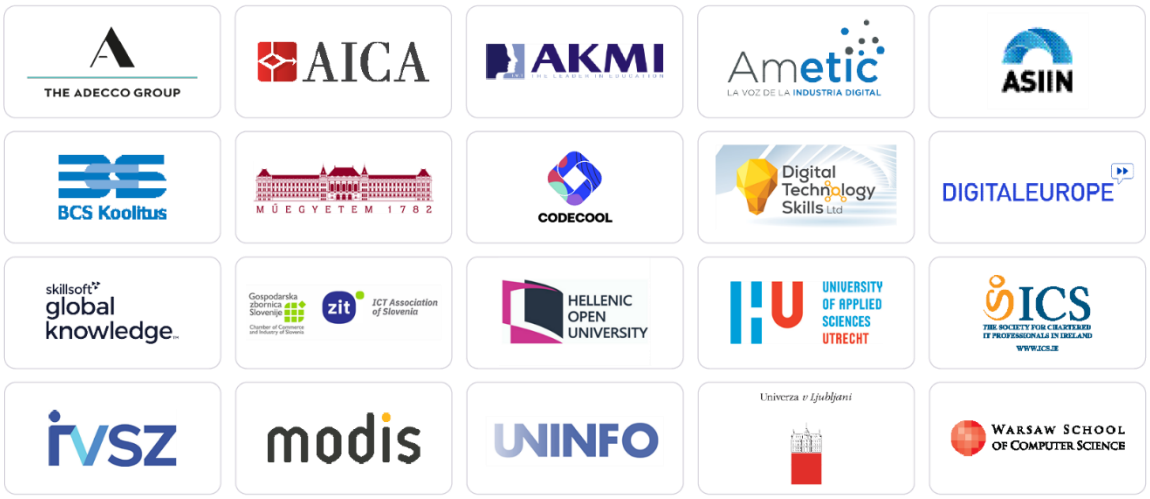

Table of Contents

List of abbreviations and acronyms

Abbreviation	Term
e-CF, EN 16234-1	European e-Competence Framework, European Norm 16234 - Part 1: Framework
ECTS	European Credit Transfer and Accumulation System
EQF	European Qualifications Framework
ESSA	European Software Skills Alliance
LO	Learning Outcome
PLO	Programme Learning Outcome

1 How to use the ESSA Learning Programme for Junior Developer EQF 4/5 Profile

1.1 Introduction

In this Annex trainers, teachers and educators are provided with all information necessary to deliver the ESSA Learning Programmes designed for the Junior Developer EQF 4/5 Educational profile.

The proposed three learning paths follow a modular and flexible structure based on Programme Learning Outcomes (PLOs). Each PLO includes self-consistent Learning Units (LUs) supported by specific learning materials.

In particular, this document provides:

- overall information for Learning Programme - Objective, Total number of Programme Learning Outcomes (PLOs) concerned, Total Learning Units (LUs), Overall duration (hours); Total number of ECTS; Targeted Institutions (learning providers);
- detailed Learning Programme including the Learning Units for each Programme Learning Outcome (PLO).

In this regard, more specifically, the following is provided for each Programme Learning Outcome (PLO):

- overall information (N. of Learning Units, Duration in hours, Total number of ECTS, Recommendations for Micro-credentials, possible integration with studies related to other PLOs, Recommended Didactical Approach, Recommended Delivery methods, etc);
- detailed information for each Learning Unit (Title, Duration in hours, Didactical Approach and delivery method, type of Assessment, Title of the related Learning material proposed, Link to access to the learning material - ESSA Platform).

This Annex is strictly related to the document "*Train the Trainer Programme. DELIVERABLE 13 – ESSA Train the Trainer Programme & Materials*".

As a further support, it is advised to consult the documents indicated in the paragraph "*Sources of reference*" of the Deliverable 13 above mentioned, through the available links.

Learning materials developed to support the delivery of the ESSA Learning Programmes for this Educational Profile are available on the ESSA platform at the following link: https://learn.softwareskills.eu/.

1.2 Target

The three following Learning Programmes address three types of target groups:

- Students with ICT background (students from Technical Institutes, university students and professional committed in upskilling or reskilling paths);
- Unemployed adults and young people aged 16 to 29 who are not currently working or studying;
- Workers in upskilling/reskilling paths.

2 How to deliver the ESSA Junior Developer EQF 4/5 Profile

2.1 Students with an ICT background

2.1.1 Overall Information about the Learning Programme

Objective	*The programme provides participants with the knowledge necessary for software development. Programming languages, techniques and practices for frontend and backend development are in-depth.* *The main support tools, the fundamentals of software testing and project management and team collaboration skills are explored to ensure maximum productivity in business contexts.*
Total number of PLOs concerned	*9*
Total Learning Units (LU)	*14*
Duration	*200 hours*
Total number of ECTS	*8*
Targeted Institutions	*Higher Education and training providers*

2.1.2 Learning Programme PLO 1 – Application Design [e-2]

Overall information PLO 1 - Application Design [e-2]	
N. of Learning Units	*2*
Learning Outcomes	- *Explains and distinguishes basic principles and terminology of software design* - *(e.g., phases in the design process, common techniques, deliverables)* - *Describes principles of user interface design* - *Reads design models and diagrams (e.g., ERD, UML)* - *Interprets a basic database design* - *Interprets a design for an application or software component*
Duration	*20 hours*
Total number of ECTS	*starting from n.0,5 ECTS*
Recommendations for Micro-credentials	*n/a*
Often integrated with studies of PLO	PLO 2 Application Development [e-2]
Recommended Didactical Approach	Virtual Classroom
Additional comments	*-*
Recommended Delivery methods	*Lecture up to 60%* *Coding Training Lab delivered by individual/team project work up to 40%*

Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from reliable sources.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:* • *Specifies a design for an application or software (component), taking into account certain basic constraints/ requirements;* • *Checks whether the design meets requirements/ wishes and, if necessary, makes proposals to adapt the design;* • *Interprets and employs the software (component)/ application design and design patterns.*
Important (new) approaches and technologies to consider	*Before starts to coding, the learner should decide if he/she want to start off with a set of codes and stick with them (deductive coding), or come up with the codes as he/she read what he/she read see in his/her data (inductive), or take a combination approach.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.2.1 Learning Units PLO 1 – Application Design [e-2]

LU1	**Overview of the Main Programming Languages and key differences**
Duration	*8 hours*
Didactical Approach and delivery method	*Virtual classroom, workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the characteristics and differences between the main programming languages, using practical examples, such as viewing and analysing programming code.*
Assessment	*Exam. The candidate must evaluate which programming language is most suitable for the development of specific digital services.*
Title of the Learning material	*Overview of the Main Programming Languages* *Cloud and Virtualization*

LU2	**Principles of UI/UX Design. Adobe XD, Zepling.**
Duration	*12 hours*
Didactical Approach and delivery method	*Virtual classroom, workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main UI/UX design principles, using practical examples, such as viewing and analysing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to build the best User Experience for the web application developed in the previous modules.*

Title of the Learning material	*Principles of UI UX Design*

2.1.3 Learning Programme PLO 2 – Application Development [e-2]

Overall information PLO 2 - Application Development [e-2]	
N. of Learning Units	*7*
Learning Outcomes	- *Explains and distinguishes common software development methods (e.g., waterfall, iterative, agile), techniques (e.g., object-oriented) and tools (e.g., IDE, CASE; multimedia integration tools; app development tools)* - *Describes common programming principles and terminology (e.g., secure programming)* - *Explains concepts and principles of databases, data structures and query languages (e.g., SQL)* - *Participates in a development process and applies a common software development method (e.g., agile)* - *Creates a simple relational database* - *Writes code and related documentation to it, by using a common programming language and applying coding conventions (e.g., Java, Javascript, PHP, Python; clean coding principle)* - *Creates a simple working software component or application, taking into account architecture, design requirements and other possible constraints (e.g., installability) applying relevant tools and techniques (e.g., object-oriented programming; IDE, CASE; editors, compilers, version control tools)* - *Modifies an existing software application or component*
Duration	*112 hours*
Total number of ECTS	*starting from n.5 ECTS*
Recommendations for Micro-credentials	*n/a*
Often integrated with studies of PLO	PLO 1, 3, 4, 5
Recommended Didactical Approach	Virtual Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 60%* *Coding Training Lab delivered by individual/team project work up to 40%*
Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from reliable sources.*
Work Based Learning Task	*After learning the basic principles, terminology, and models of Application Development, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:*

***(If foreseen) and* Follow-up, learning reinforcement**	• *Participates in a development process and applies a common software development method;* • *Writes code and related documentation to it, by using a common programming language and applying coding conventions (e.g., Java, Javascript, PHP, Python; clean coding principle)*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.3.1 Learning Units PLO 2 – Application Development [e-2]

LU1	**Overview of the Main Programming Languages and key differences**
Duration	*8 hours*
Didactical Approach and delivery method	*Virtual classroom, workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the characteristics and differences between the main programming languages, using practical examples, such as viewing and analysing programming code.*
Assessment	*Exam. The candidate must evaluate which programming language is most suitable for the development of specific digital services.*
Title of the Learning material	*Overview of the Main Programming Languages*

LU2	**HTML5, CSS3, BOOTSTRAP**
Duration	*24 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the HTML and CSS code and relative application, using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to create a web application using HTML5, CSS3 and Bootstrap.*
Title of the Learning material	*HTML* *CSS3* *Exercise: JuniorDEVProj_HTML-CSS-JS*

LU3	**Javascript, AJAX, Typescript, GIT**
Duration	*24 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*

Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of Javascript, Ajax, Typescript, using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to develop a web application using frontend coding (Javascript, AJAX, Typescript) and related tools.*
Title of the Learning material	*Java* *Javascript Ajax* *Java at main MaSTERmIKK_JuniorDEVProj · GitHub*

LU4	**Backend – Coding and development tools: Java 11, Spring Boot, Sprint Data, Hibernate, Ex Java: Junit, Mockito**
Duration	*16 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to build and test a backend-side application using Java 11 and related development tools.*
Title of the Learning material	*Backend – Java 11* *Spring at main*

LU5	**Backend – Coding and development tools: PHP, Laravel, Eloquent**
Duration	*16 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to build and test a backend-side application using PHP and related development tools.*
Title of the Learning material	*PHP Laravel Eloquent*

LU6	**Backend – Coding and development tools: Fundamentals of Ruby, Python, NodeJS**
Duration	*16 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.*

	The student is asked to build a backend-side application using Ruby, Python and NodeJS.
Title of the Learning material	*Back End - Objects - Ruby - Phyton -NodeJS* *NodeJS at Main*

LU7	**Agile Project Management, SCRUM and collaboration tools**
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the Agile and SCRUM culture and framework using practical examples and exercises..*
Assessment	*Assignment: practical activity.* *The student is invited to apply the Agile methodology in the development of a web application.*
Title of the Learning material	*Agile PM and SCRUM*

2.1.4 Learning Programme PLO 3 – Component Integration [e-2]

Overall information PLO 3 – Component Integration [e-2]	
N. of Learning Units	*8*
Learning Outcomes	- *Explains and distinguishes common methods, techniques and tools related to efficient integration* - *Describes the interplay between and compatibility of system components* - *Carries out installation and configuration activities, applying common methods, techniques and tools related to efficient integration (e.g., packaging and distribution, virtualisation, containerisation)* - *Monitors and tests the connectivity of integrated systems* - *Writes an installation report*
Duration	*132 hours*
Total number of ECTS	*starting from n.4 ECTS*
Recommendations for Micro-credentials	*n/a*
Often integrated with studies of PLO	PLO 1, 5
Recommended Didactical Approach	Virtual Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 60%* *Coding Training Lab delivered by individual/team project work up to 40%*

Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from authoritative sources.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of Component Integration, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:* • *Participates in a development process and applies a common software development method;* • *Writes code and related documentation to it, by using a common programming language and applying coding conventions;* • *Describes the interplay between and compatibility of system components;* • *Monitors and tests the connectivity of integrated systems.*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.4.1 Learning Units PLO 3 – Component Integration [e-2]

LU1	**Entity-Relationship Model, SQL, MySql**
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Exam. The candidate is asked to set up a database to support a web application.*
Title of the Learning material	*Entity-Relationship Model - SQL - MySql*

LU2	**HTML5, CSS3, BOOTSTRAP**
Duration	*24 hours*
Didactical Approach and delivery method	*Virtual classroom, workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the HTML and CSS code and relative application, using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to create a web application using HTML5, CSS3 and Bootstrap.*

Title of the Learning material	*HTML5, CSS3, Bootstrap* *Exercise: JuniorDEVProj_HTML-CSS-JS*

LU3	**Javascript, AJAX, Typescript, GIT**
Duration	*24 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of Javascript, Ajax, Typescript, using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to develop a web application using frontend coding (Javascript, AJAX, Typescript) and related tools.*
Title of the Learning material	*Javascript Ajax* *Exercise: Java at main*

LU4	**Angular, React. Jest, Mocha, Selenium**
Duration	*24 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to develop a web application using Angular and tools such as: React, Jest, Mocha, Selenium.*
Title of the Learning material	*Angular - React - Jest - Mocha – Selenium*

LU5	**Principles of UI/UX Design. Adobe XD, Zepling**
Duration	*12 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to build the best User Experience for the web application developed in the previous modules.*
Title of the Learning material	*Principles of UI UX Design - Adobe XD – Zepling*

LU6	**Java 11, spring boot, spring data/ Hibernate. Ex. Java: Junit, Mockito**

Duration	*16 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to build and test a backend-side application using Java 11 and related development tools.*
Title of the Learning material	*Backend – Java 11* *Exercise: Java at main*

LU7	**Backend: PHP, Laravel, Eloquent**
Duration	*16 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to build and test a backend-side application using PHP and related development tools.*
Title of the Learning material	*PHP Laravel Eloquent*

LU8	**Introduction to STLC (software testing Life cycle). Unit test, end to end test**
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of STLC using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to test an application developed in the previous modules.*
Title of the Learning material	*Introduction to STLC*

2.1.5 Learning Programme PLO 4 – Testing [e-2]

Overall information PLO 4 – Testing [e-2]	
N. of Learning Units	*1*
Learning Outcomes	- *Explains and distinguishes principles of software testing, common testing methods, techniques, and tools* - *Writes an (automated) test on a piece of code*

	- *Performs common test activities, applying testing and debugging techniques and tools* - *Records and interprets test outcomes and writes test result documentation/ test report*
Duration	*8 hours*
Total number of ECTS	*starting from n.0,5 ECTS*
Recommendations for Micro-credentials	*n/a*
Often integrated with studies of PLO	n/a
Recommended Didactical Approach	Virtual Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 60%* *Coding Training Lab delivered by individual/team project work up to 40%*
Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to software testing procedures, reading websites specialized in STLC, watching online tutorials and downloading materials useful for practical exercises from reliable sources.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of Software Testing, the study should focus on analysing and simulating real work-life-like tasks as, for example:* • *Define and explain appropriate test methods, techniques, and tools.* • *Explain and write (parts of) testing related documentation, such as a test plan, test strategy/approach, test case, test script, test scenario, test conditions.* • *Setup a test environment.*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.5.1 Learning Units PLO 4 – Testing [e-2]

LU1	**Fundamentals of software testing: Introduction to STLC, Unit testing, end to end testing**
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*

Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of STLC using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to test an application developed in the previous modules.*
Title of the Learning material	*Introduction to STLC – Unit test – end to and test*

2.1.6 Learning Programme PLO 5 – Documentation production [e-2]

Overall information PLO 5 – Documentation production [e-2]	
N. of Learning Units	*3*
Learning Outcomes	- *Describes types of technical documentation* - *Provides different (parts of) common technical documents, using appropriate tools (e.g., software documentation tools)*
Duration	*64 hours*
Total number of ECTS	*starting from n.2 ECTS*
Recommendations for Micro-credentials	*n/a*
Often integrated with studies of PLO	PLO 3, 4, 5
Recommended Didactical Approach	Virtual Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 60%* *Training Lab delivered by individual/team project work up to 40%*
Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in software development, watching online tutorials and downloading materials useful for practical exercises from reliable sources.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of Application Development, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:* • *Describes types of technical documentation;* • *Provides different (parts of) common technical documents, using appropriate tools (e.g., software documentation tools).*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.6.1 Learning Units PLO 5 – Documentation production [e-2]

LU1	Entity-Relationship Model, SQL, MySql
Duration	*16 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of those sets of code using practical examples, such as viewing and analyzing programming code.*
Assessment	*Exam. The candidate is asked to set up a database to support a web application .*
Title of the Learning material	*Entity-Relationship Model - SQL – MySql*

LU2	HTML5, CSS3, BOOTSTRAP
Duration	*24 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the HTML and CSS code and relative application, using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to create a web application using HTML5, CSS3 and Bootstrap.*
Title of the Learning material	*HTML5, CSS3, Bootstrap* *Exercise: Html-Css-JS at main*

LU3	Javascript, AJAX, Typescript, GIT
Duration	*24 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of Javascript, Ajax, Typescript, using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to develop a web application using frontend coding (Javascript, AJAX, Typescript) and related tools.*
Title of the Learning material	*Java* *Java at main*

2.1.7 Learning Programme PLO 6 – Problem management [e-2]

Overall information PLO 6 – Problem management [e-2]	
N. of Learning Units	*1*
Learning Outcomes	*Systematically resolves or escalates incidents and problems, resulting in a solved incident e.g., by applying techniques and tools for troubleshooting such as diagnostic tools*
Duration	*8 hours*
Total number of ECTS	*starting from n.0,5 ECTS*
Recommendations for Micro-credentials	*n/a*
Often integrated with studies of PLO	PLO 4
Recommended Didactical Approach	Virtual Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 80%* *Training Lab delivered by individual/team project work up to 20%*
Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in coding and web development, watching online tutorials and downloading materials useful for practical exercises from reliable sources.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of Application Development, the study should focus on analysing and simulating real work-life-like tasks as, for example, the student:* • *Participates in a development process and solve common problems applied to the software development.*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.7.1 Learning Units PLO 6 – Problem management [e-2]

LU1	Fundamentals of software testing: Introduction to STLC, Unit testing, end to end testing
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*

Additional information	*By alternating between theory and practical activities, students will be able to understand the fundamentals of software testing, with reference to software testing life cycle and to Unit Testing, software carried out during the development of an application.*
Assessment	*Assignment: practical activity.* *The student is asked to test a web application developed in the previous modules.*
Title of the Learning material	*Introduction to STLC*

2.1.8 Learning Programme PLO 7 – Professional related competences [EQF5]

Overall information PLO 7 – Professional related competences [EQF5]	
N. of Learning Units	*4*
Learning Outcomes	- *Masters common ICT knowledge* - *Explains the principles, related concepts, advantages and disadvantages of a new technology. Applies and reports on basic methods, techniques and tools related to a new technology.* - *Applies and reports on measures, methods, tools and techniques related to security* - *Applies and reports on measures, methods, tools and techniques related to software lifecycle processes* - *Is aware of basic ethical considerations and issues*
Duration	*28 hours*
Total number of ECTS	*starting from n.1 ECTS*
Recommendations for Micro-credentials	*This PLO should be an integral part of the initial studies for students with no prior knowledge of professional-related competences useful to work in complex organizations embedded in innovative markets.*
Often integrated with studies of PLO	PLO 1, 8, 9
Recommended Didactical Approach	Virtual Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 100%*
Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in innovation, project management and team collaboration.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*n/a*
Important (new) approaches and technologies to consider	• *. Business Process and Business Architecture understanding and mapping tools* • *Product/service design and innovation management*

Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.8.1 Learning Units PLO 7 – Professional related competences [EQF5]

LU1	**Introduction to ICT and Digital Transformation tech enablers**
Duration	*4 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand ICT culture using practical example and storytelling of case histories*
Assessment	*1 test (multiple choice questions) on the characteristics of the main IT technologies.*
Title of the Learning material	*Introduction to ICT and Digital Transformation tech enablers*

LU2	**Agile Project Management, SCRUM and collaboration tools**
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the Agile and SCRUM culture and framework using practical examples and exercises.*
Assessment	*Assignment: practical activity.* *The student is invited to apply the Agile methodology in the development of a web application.*
Title of the Learning material	*Agile PM and SCRUM*

LU3	**Fundamentals of cybersecurity**
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*-*
Assessment	*Assignment: practical activity.* *The student is required to take a computer security test on a series of web applications.*
Title of the Learning material	*Fundamentals of Cybersecurity*

LU4	**Introduction to STLC (software testing Life cycle). Unit test, end to end test.**
Duration	*8 hours*

Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of STLC using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to test a web application developed in the previous modules.*
Title of the Learning material	*Introduction to STLC - Unit tests - end-to-end tests*

2.1.9 Learning Programme PLO 8 – Soft competences [EQF5]

Overall information PLO 8 – Soft competences [EQF5]	
N. of Learning Units	*2*
Learning Outcomes	- *Works together with others in a team* - *Communicates with peers, colleagues, supervisors and/or relevant others, appropriately to the context, using conventions that are relevant to professional practice. Explains and gives instruction.* - *Masters the English language at level B2. Can understand the main ideas of complex text on both concrete and abstract topics, including technical discussions in his/her field of specialisation* - *Distinguishes and analyses fairly complex and unpredictable problems. Solves these problems systematically and in a creative way, using existing procedures and guidelines and own solutions by identifying and using data.* - *Exercises self-management within the guidelines of contexts that are usually predictable, but are subject to change. Is able to cope with limited change and to adapt to a certain level of variety in the workplace. Copes with pressure and stress setbacks and maintains composure. Shows some initiative and carries responsibility for the results of own activities, work and or study. Works correctly and carefully.* - *Realises learning and personal development on request, where necessary with support, through self-reflection and external- and self-evaluation of own (learning) results.*
Duration	*16 hours*
Total number of ECTS	*starting from n.0,5 ECTS*
Recommendations for Micro-credentials	*This PLO should be an integral part of the initial studies for students with no prior knowledge of professional-related competences useful to work in complex organizations embedded in innovative business*
Often integrated with studies of PLO	PLO 9
Recommended Didactical Approach	Virtual Classroom
Additional comments	-

Recommended Delivery methods	*Lecture up to 100%*
Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in programming, watching online tutorials and downloading materials useful for practical exercises from reliable sources.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*n/a*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.9.1 Learning Units PLO 8 – Soft competences [EQF5]

LU1	**Team Collaboration (soft Skill)**
Duration	*8 hours*
Didactical Approach and delivery method	*Formative quizzes* *Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main team collaboration principles and practices, using practical examples, case study analysis and exercises.*
Assessment	*Assignment: practical activity.* *The student is called to work on a development project, creating a work group and collaborating with other people through the use of enterprise social collaboration tools..*
Title of the Learning material	*Team Collaboration and file versioning*

LU2	**Introduction to STLC (software testing Life cycle)**
Duration	*8 hours*
Didactical Approach and delivery method	*Formative quizzes* *Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main applications of STLC using practical examples, such as viewing and analyzing programming code.*
Assessment	*Assignment: practical activity.* *The student is asked to test a web application developed in the previous modules.*
Title of the Learning material	*Introduction to STLC - Unit tests - end-to-end tests*

2.1.10 Learning Programme PLO 9 – Functioning in organisations [EQF5]

Overall information PLO 9 – Functioning in organisations [EQF5]	
N. of Learning Units	*2*
Learning Outcomes	- *Explains the basics of organisation theory and behaviour* - *Describes the relationship between business and IT* - *Works in an organisational context under specific direction with limited autonomy and responsibility e.g., at the level of a trainee, junior or assistant* - *Works in project settings, applies project management methods and tools* - *Writes a report on functioning in the organisation*
Duration	*16 hours*
Total number of ECTS	*starting from n.1 ECTS*
Recommendations for Micro-credentials	*This PLO should be an integral part of the initial studies for students with no prior knowledge of teamwork collaboration.*
Often integrated with studies of PLO	PLO 8
Recommended Didactical Approach	Virtual Classroom
Additional comments	*-*
Recommended Delivery methods	*Lecture up to 100%*
Additional comments	*It is recommended to deepen the topics presented in the Learning Units by reading publications dedicated to the various topics, reading websites specialized in teamwork collaboration and communication & collaboration tools and platforms.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*n/a*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.1.10.1 Learning Units for Learning Units PLO 9 – Functioning in organisation [EQF5]

LU1	**Team Collaboration**

Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand the main team collaboration principles and practices, using practical examples, case study analysis and exercises*
Assessment	*Assignment: practical activity.* *The student is called to work on a development project, creating a work group and collaborating with other people using enterprise social collaboration tools*
Title of the Learning material	*Team collaboration and file versioning*

LU2	**ICT and Digital Transformation tech enablers**
Duration	*8 hours*
Didactical Approach and delivery method	*Workshop and lecture guides*
Additional information	*The didactic approach would be aimed to allows participants to understand ICT culture using practical example and storytelling of case histories*
Assessment	*1 test (multiple choice questions) on the characteristics of the main IT technologies.*
Title of the Learning material	*Introduction to ICT and Digital Transformation tech enablers*

2.2 Unemployed adults and young people aged 16 to 29

2.2.1 Overall Information about the Learning Programme

Role	Junior Developer
EQF Level	*4*
Objective	*The course provides learners knowledge and skills to create a simple working software component or application*
Total number of PLOs concerned	*4*
Total Learning Units (LU)	*13*
Duration	*92 hours*
Total number of ECTS	*4*
Targeted Institutions	*Training providers*

2.2.2 Learning Programme PLO 1 – Application Design [e-2]

Overall information PLO 1 - Application Design [e-2]	
N. of Learning Units	*3*
Learning Outcomes	- *Explains and distinguishes basic principles and terminology of software design* - *(e.g., phases in the design process, common techniques, deliverables)*

	- *Describes principles of user interface design* - *Reads design models and diagrams (e.g., ERD, UML)* - *Interprets a basic database design* - *Interprets a design for an application or software component*
Duration	*16 hours*
Total number of ECTS	*starting from n.0.5 ECTS*
Recommendations for Micro-credentials	*This course is designed as one micro-credential course, and it cannot be split into smaller integral units.*
Often integrated with studies of PLO	-
Recommended Didactical Approach	Virtual Classroom Presence Classroom Blended
Additional comments	-
Recommended Delivery methods	*Lecture up to 20%* *Practical tasks and exercises up to 80%*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example:* • *Designing and creating different webpage examples as real-life-like customer project*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.2.2.1 Learning Units PLO 1 – Application Design [e-2]

LU1	**Basics of programming**
Duration	*4 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Lecture (80%)* *Discussion (20%)*
Assessment	*Report*

Title of the Learning material	*Agile Development*

LU2	**CSS3**
Duration	*8 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Practical exercises*
Assessment	*Practical tasks*
Title of the Learning material	*CSS*

LU3	**Creating User Stories**
Duration	*4 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Teamwork*
Assessment	*Practical tasks*
Title of the Learning material	*User stories*

2.2.3 Learning Programme PLO 2 – Application Development [e-2]

Overall information PLO 2 - Application Development [e-2]	
N. of Learning Units	*6*
Learning Outcomes	- *Explains and distinguishes common software development methods (e.g., waterfall, iterative, agile), techniques (e.g., object-oriented) and tools (e.g., IDE, CASE; multimedia integration tools; app development tools)* - *Describes common programming principles and terminology (e.g., secure programming)* - *Explains concepts and principles of databases, data structures and query languages (e.g., SQL)* - *Participates in a development process and applies a common software development method (e.g., agile)* - *Creates a simple relational database* - *Writes code and related documentation to it, by using a common programming language and applying coding conventions (e.g., Java, Javascript, PHP, Python; clean coding principle)* - *Creates a simple working software component or application, taking into account architecture, design requirements and other possible constraints (e.g., installability) applying relevant tools and techniques*

	(e.g., object-oriented programming; IDE, CASE; editors, compilers, version control tools) - *Modifies an existing software application or component*
Duration	*44 hours*
Total number of ECTS	*starting from n.3 ECTS*
Recommendations for Micro-credentials	*This course is designed as one micro-credential course, and it cannot be split into smaller integral units.*
Often integrated with studies of PLO	-
Recommended Didactical Approach	Virtual Classroom Presence Classroom Blended
Additional comments	-
Recommended Delivery methods	*Lecture up to 15%* *Practical exercises up to 85%*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example:* • *Designing and creating different web application examples as real-life-like customer project*
Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.2.3.1 Learning Units PLO 2 – Application Development [e-2]

LU1	**Basics of programming**
Duration	*4 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Lecture (80%)* *Discussion (20%)*
Assessment	*Discussion*
Title of the Learning material	*Agile Development*

LU2	Agile development methodologies
Duration	*4 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Lecture (80%)* *Discussion (20%)*
Assessment	-
Title of the Learning material	*Agile Development*

LU3	HTML5
Duration	*8 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Practical exercises*
Assessment	*Practical tasks*
Title of the Learning material	*HTML*

LU4	Javascript
Duration	*12 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Practical exercises*
Assessment	*Practical tasks*
Title of the Learning material	*Javascript*

LU5	Php
Duration	*8 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Practical exercises*
Assessment	*Practical tasks*
Title of the Learning material	*Php*

LU6	MySQL
Duration	*8 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Practical exercises*
Assessment	*Practical tasks*
Title of the Learning material	*SQL*

2.2.4 Learning Programme PLO 4 – Testing [e-2]

Overall information PLO 4 – Testing [e-2]	
N. of Learning Units	*3*
Learning Outcomes	- *Explains and distinguishes principles of software testing, common testing methods, techniques, and tools* - *Writes an (automated) test on a piece of code* - *Performs common test activities, applying testing and debugging techniques and tools* - *Records and interprets test outcomes and writes test result documentation/ test report*
Duration	*24 hours*
Total number of ECTS	*starting from n.1 ECTS*
Recommendations for Micro-credentials	*This course is designed as one micro-credential course, and it cannot be split into smaller integral units.*
Often integrated with studies of PLO	-
Recommended Didactical Approach	Virtual Classroom Presence Classroom Blended
Additional comments	-
Recommended Delivery methods	*Case study. Individual/team project, integrated with other modules*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*Designing and creating different web application examples as real-life-like customer project*

Important (new) approaches and technologies to consider	*n/a*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.2.4.1 Learning Units PLO 4 – Testing [e-2]

LU1	**Basics of programming**
Duration	*4 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Lecture (80%)* *Discussion (20%)*
Assessment	*Test*
Title of the Learning material	*Agile Development*

LU2	**Javascript**
Duration	*12 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Practical exercises*
Assessment	*Practical tasks*
Title of the Learning material	*Javascript*

LU3	**Php**
Duration	*8 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Practical exercises*
Assessment	*Practical tasks*
Title of the Learning material	*Php, MySQL*

2.2.5 Learning Programme PLO 8 – Soft competences [EQF5]

Overall information PLO 8 – Soft competences [EQF5]	
N. of Learning Units	*1*
Learning Outcomes	- *Works together with others in a team* - *Communicates with peers, colleagues, supervisors and/or relevant others, appropriately to the context, using conventions that are relevant to professional practice. Explains and gives instruction.* - *Masters the English language at level B2. Can understand the main ideas of complex text on both concrete and abstract topics, including technical discussions in his/her field of specialisation* - *Distinguishes and analyses fairly complex and unpredictable problems. Solves these problems systematically and in a creative way, using existing procedures and guidelines and own solutions by identifying and using data.* - *Exercises self-management within the guidelines of contexts that are usually predictable, but are subject to change. Is able to cope with limited change and to adapt to a certain level of variety in the workplace. Copes with pressure and stress setbacks and maintains composure. Shows some initiative and carries responsibility for the results of own activities, work and or study. Works correctly and carefully.* - *Realises learning and personal development on request, where necessary with support, through self-reflection and external- and self-evaluation of own (learning) results.*
Duration	*8 hours*
Total number of ECTS	*starting from n.0,5 ECTS*
Recommendations for Micro-credentials	*Not applicable*
Often integrated with studies of PLO	-
Recommended Didactical Approach	Blended
Additional comments	-
Recommended Delivery methods	*Teamwork*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*Related with other study units.*

Important (new) approaches and technologies to consider	*Not applicable* .
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.2.5.1 Learning Units PLO 8 – Soft competences [EQF5]

LU1	**Teamwork**
Duration	*8 hours*
Didactical Approach and delivery method	*Blended*
Additional information	*Lecture (80%)* *Discussion (20%)*
Assessment	*Self-reflection report*
Title of the Learning material	General skills

2.3 Workers in upskilling/reskilling paths

2.3.1 Overall Information about the Learning Programme

Objective	*The 22-week programme, crafted for employed professionals looking to reskill and pursue a career in software development, aims to provide a comprehensive foundation in software development. Participants will have the opportunity to engage in both collaborative group work and guided individual learning, ensuring a well-rounded and tailored educational experience.* *It is tailored for entry-level junior developers. It serves as an accelerated pathway for employed individuals seeking to reskill for a career in software development or enhance their existing roles with valuable software development skills. The program focuses on providing a comprehensive foundation in coding and design, equipping participants with the expertise needed in the dynamic field of software development.*
Total number of PLOs concerned	*9*
Total Learning Units	*13*
Duration	*Min 212 hours*
Total number of ECTS	*8*
Targeted Institutions	*Training providers*

2.3.2 Learning Programme PLO 1 – Application Design [e-2]

Overall information PLO 1 - Application Design [e-2]	
N. of Learning Units	*1*
Learning Outcomes	- *Explains and distinguishes basic principles and terminology of software design* - *(e.g., phases in the design process, common techniques, deliverables)* - *Describes principles of user interface design* - *Reads design models and diagrams (e.g., ERD, UML)* - *Interprets a basic database design* - *Interprets a design for an application or software component*
Duration	*16 hours*
Total number of ECTS	*starting from n.0,5 ECTS*
Recommendations for Micro-credentials	*One distinct unit – possibly suitable for micro credentials*
Often integrated with studies of PLO	PLO 2 Application Development [e-2]
Recommended Didactical Approach	Virtual Classroom e-learning Presence Classroom

Additional comments	-
Recommended Delivery methods	· *Lecture* *50%* · *Case study. Individual/team project* *30%* · *e-Learning* *20%*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software application design. These are reinforced through two assignment options.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example:* • *The combination of both assignments provides an opportunity for students to engage in a real-life customer based project.*
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.2.1 Learning Units PLO 1 – Application Design [e-2]

LU1	**Application Design**
Duration	*14 or 18 hours if both assessments are used*
Didactical Approach and delivery method	*Lecture, e-Learning and assignment(s)*
Additional information	*Tutor* *Led*
Assessment	*Design Process Assignment* *Design an Application Assignment*
Title of the Learning material	*ESSA_Junior_Developer_Application_Design* ***Recommended Reading*** **9 Key Elements of Application Design** **https://centogram.com/2022/01/28/9-key-elements-of-application-design/** **Software Architecture and Software Design** **https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3772387** **How has design thinking being used and integrated into software development activities? A systematic mapping** **https://www.sciencedirect.com/science/article/abs/pii/S0164121222000024**

2.3.3 Learning Programme PLO 2 – Application Development [e-2]

Overall information PLO 2 - Application Development [e-2]	
N. of Learning Units	*8*
Learning Outcomes	- *Explains and distinguishes common software development methods (e.g., waterfall, iterative, agile), techniques (e.g., object-oriented) and tools (e.g., IDE, CASE; multimedia integration tools; app development tools)* - *Describes common programming principles and terminology (e.g., secure programming)* - *Explains concepts and principles of databases, data structures and query languages (e.g., SQL)* - *Participates in a development process and applies a common software development method (e.g., agile)* - *Creates a simple relational database* - *Writes code and related documentation to it, by using a common programming language and applying coding conventions (e.g., Java, Javascript, PHP, Python; clean coding principle)* - *Creates a simple working software component or application, taking into account architecture, design requirements and other possible constraints (e.g., installability) applying relevant tools and techniques (e.g., object-oriented programming; IDE, CASE; editors, compilers, version control tools)* - *Modifies an existing software application or component*
Duration	*140 hours*
Total number of ECTS	*starting from n.5 ECTS*
Recommendations for Micro-credentials	*May be suitable for micro-credentials in specific application development skills*
Often integrated with studies of PLO	PLO1 Application Design, PLO3 Component Integration, PLO4 Testing, PLO5 Documentation Production, PLO6 Problem Management, PLO New Technology, PLO7Profession Related Competences, PLO9 Functioning in Organisations).
Recommended Didactical Approach	Virtual Classroom e-learning Presence Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 70%* *Case study. Individual/team project up to 20%* *e-Learning up to 10%*
Additional comments	*Lectures and e-learning are recommended for learning the basic principles and terminology associated with application design. By engaging in one or both assignments, students' have the opportunity to put the learning in this unit into practice.*
Work Based Learning Task	*Through the Industry Related Software Development Project (Software Pathway Project) students' have the opportunity to engage in a coding project which mirrors a software project within a work environment and build on their learning through this unit of application design.*

***(If foreseen) and* Follow-up, learning reinforcement**	
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.3.1 Learning Units PLO 2 – Application Development [e-2]

LU1	**Application Development**
Duration	*30 hours*
Didactical Approach and delivery method	*Lecture* *e-learning*
Additional information	*-*
Assessment	*Python Assignment*
Title of the Learning material	*ESSA_Junior_Developer_application_Development*

LU2	**Application Development**
Duration	*24 hours*
Didactical Approach and delivery method	*Lecture*
Additional information	*-*
Assessment	*-*
Title of the Learning material	*04_HTML5 CSS3 Bootstrap*

LU3	**Application Development**
Duration	*24 hours*
Didactical Approach and delivery method	*Lecture*
Additional information	*-*
Assessment	*-*

Title of the Learning material	*05_Javascript-Ajax*

LU4	**Application Development**
Duration	*16 hours*
Didactical Approach and delivery method	*Lecture*
Additional information	-
Assessment	-
Title of the Learning material	*08_Backend – Java 11*

LU5	**Application Development**
Duration	*16 hours*
Didactical Approach and delivery method	*Lecture*
Additional information	-
Assessment	-
Title of the Learning material	*09_PHP Laravel Eloquent*

LU6	**Application Development**
Duration	*24 hours*
Didactical Approach and delivery method	*Lecture*
Additional information	-
Assessment	-
Title of the Learning material	*10_Back end – Objects – Ruby – Python -NodeJS*

LU7	**Application Development**
Duration	*6 hours*
Didactical Approach and delivery method	*e-Learning*
Additional information	-
Assessment	-

Title of the Learning material and file	*Reading Materials & Supports* *PEP 8 – Style Guide for Python Code* https://peps.python.org/pep-0008/Conduct_one-on-one_interviews_with_users to observe their interactions with the product *How to Write Beautiful Python Code With PEP 8* https://realpython.com/python-pep8/ *Lesson 2. Clean Code Syntax for Python: Introduction to PEP 8 Style Guide* https://www.earthdatascience.org/courses/intro-to-earth-data-science/write-efficient-python-code/intro-to-clean-code/python-pep-8-style-guide/ *Noteworthy 30+ Learning Resources For Developers 2023* https://dev.to/theme_selection/learning-resources-for-developers-165d *3 Must Have Tools for Every Junior Software Developer* https://launchacademy.com/blog/3-must-have-tools-for-every-junior-software-developer Codecademy: **Offers interactive coding courses in various programming languages.** FreeCodeCamp: **Provides free online coding lessons and coding challenges.** Udemy: **Offers a wide range of programming courses taught by industry experts.** Coursera: **Provides online courses from top universities and institutions**

LU8	**Industry Related Software Development Project**
Duration	*4 weeks*
Didactical Approach and delivery method	*Assignment*
Additional information	*Tutor Assigned*
Assessment	*Industry Project*
Title of the Learning material	*ESSA_Software_Pathway_Project_Junior_Developer_(EQF4_7)*

2.3.4 Learning Programme PLO 3 – Component Integration [e-2]

Overall information PLO 3 – Component Integration [e-2]	
N. of Learning Units	*1*
Learning Outcomes	- *Explains and distinguishes common methods, techniques and tools related to efficient integration* - *Describes the interplay between and compatibility of system components* - *Carries out installation and configuration activities, applying common methods, techniques and tools related to efficient integration (e.g., packaging and distribution, virtualisation, containerisation)* - *Monitors and tests the connectivity of integrated systems* - *Writes an installation report*
Duration	*8 hours*
Total number of ECTS	*starting from n.0,5 ECTS*

Recommendations for Micro-credentials	*May be suitable for micro-credential*
Often integrated with studies of PLO	Aligns with end of year Project PLOs 2, 4, 7, 8, 9
Recommended Didactical Approach	Virtual Classroom e-learning Presence Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 50%* *e-learning/self study up to 50%*
Additional comments	*Lectures and e-learning are recommended for learning the basic principles, terminology, and models component integration. These are reinforced through practical in-class assignments and through the Industry Software Project referenced in PLOs 7,8,9.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of software component architectures, the knowledge gained can form part of the final year project*
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.4.1 Learning Units PLO 3 – Component Integration [e-2]

LU1	**Component Integration**
Duration	*8 hours*
Didactical Approach and delivery method	*Virtual Classroom, e-Learning*
Additional information	*Virtual Classroom via Tutor*
Assessment	*Assignment – 10 Questions & Multiple choice quiz*
Title of the Learning material	*ESSA_Junior_Developer_COMPONENT_INTEGRATION* *ESSA_Learning_Programmes_and_Materials_JUNIOR DEVELOPER_(EQF4_7)* ***Recommended Reading:***

	"Software Architecture in Practice" by Len Bass, Paul Clements, and Rick Kazman: This book provides an overview of software architecture, including different architectural styles and patterns. It also covers topics such as architectural design, documentation, and evaluation *"Building Microservices: Designing Fine-Grained Systems" by Sam Newman: This book focuses on microservices architecture, including design principles, implementation strategies, and deployment techniques* *"Enterprise Integration Patterns" by Gregor Hohpe and Bobby Woolf: This book provides an overview of different integration patterns and technologies, including messaging systems, service-oriented architecture, and enterprise application integration* *"Kubernetes in Action" by Marko Luksa: This book provides an introduction to Kubernetes, including installation, deployment, and management of containerized applications* *Docker Deep Dive" by Nigel Poulton: This book provides a comprehensive overview of Docker, including containerization concepts, image creation, and container management* *"Cloud Native Infrastructure: Patterns for Scalable Infrastructure and Applications in a Dynamic Environment" by Justin Garrison and Kris Nova: This book covers the basics of cloud-native infrastructure, including containerization, microservices, and serverless architecture* *"Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl: This book provides a comprehensive overview of service-oriented architecture, including design principles, implementation strategies, and deployment techniques* *"The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations" by Gene Kim, Jez Humble, Patrick Debois, and John Willis: This book covers the principles and practices of DevOps, including continuous integration, delivery, and deployment* *"The Practice of System and Network Administration" by Thomas A. Limoncelli, Christina J. Hogan, and Strata R. Chalup: This book provides a comprehensive guide to system and network administration, including installation and configuration of software component* *"Windows Internals, Part 1: System architecture, processes, threads, memory management, and more" by Mark E. Russinovich and David A. Solomon: This book provides a detailed look at the Windows operating system's internals, including the installation and configuration of software component* *"Linux Administration Handbook" by Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley: This book provides a comprehensive guide to Linux system administration, including the installation and configuration of software components*

	"Cloud Computing: From Beginning to End" by Ray J. Rafaels: This book provides an overview of cloud computing, including the installation and configuration of software components in cloud environments *"Network Warrior: Everything You Need to Know That Wasn't on the CCNA Exam" by Gary A. Donahue: This book provides a practical guide to network administration, including the installation and configuration of software components on networks* *"Software Interoperability: Frameworks for Addressing Challenges" by Arunava Paul, Nilanjan Banerjee, and Sajal K. Das (IEEE Communications Surveys & Tutorials, 2012).* *"A Model-Based Approach to the Design of Interoperable Software Systems" by Hassan Gomaa (IEEE Transactions on Software Engineering, 2004)* *"Interoperability in Healthcare Information Systems: Standards, Approaches, and Challenges" by Miltiadis D. Lytras, Ernesto Damiani, and Patricia Ordóñez de Pablos (IGI Global, 2010)* *"Interoperability of Enterprise Software and Applications" edited by Yingxu Wang, Athanasios Tsadiras, and Richard Hill (Springer, 2005).* *"The Importance of Interoperability in the Age of the Cloud" by Michael Mimoso (Communications of the ACM, 2011).*

2.3.5 Learning Programme PLO 4 – Testing [e-2]

Overall information PLO 4 – Testing [e-2]	
N. of Learning Units	*1*
Learning Outcomes	- *Explains and distinguishes principles of software testing, common testing methods, techniques, and tools* - *Writes an (automated) test on a piece of code* - *Performs common test activities, applying testing and debugging techniques and tools* - *Records and interprets test outcomes and writes test result documentation/ test report*
Duration	*24 hours*
Total number of ECTS	*starting from n.1 ECTS*
Recommendations for Micro-credentials	*Potential to be considered for Micro-credentials*
Often integrated with studies of PLO	PLO2 Application Development, PLO3 Component Integration, PLO7 Profession Related Competencies, PLO8 Soft Competencies and PLO9 Functioning in Organisations.
Recommended Didactical Approach	Virtual Classroom e-Learning Presence Classroom
Additional comments	-

Recommended Delivery methods	*Lecture up to 70%* *Case study. Individual/team project up to 15%* *e-Learning up to 15%*
Additional comments	*Lectures and e-learning are supplemented with two assignments. It is advisable that students' practice software testing to gain competency in automation tools and techniques.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, models and tools associated with software testing, the knowledge gained can form part of the final year project.*
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.5.1 Learning Units PLO 4 – Testing [e-2]

LU1	**Software Testing**
Duration	*24 hours*
Didactical Approach and delivery method	*Lecture* *e-Learning*
Additional information	*Tutor and e-Learning*
Assessment	*Two assignments*
Title of the Learning material and file	*ESSA_Junior_Developer_SOFTWARE_TESTING.pptx* **Optional Additional Reading:** • *Effective Testing with RSpec 3: Build Ruby Apps with Confidence" by Myron Marston and Ian Dees. This book covers the basics of testing with RSpec, a popular testing framework for Ruby applications.* • *"Testing Python: Applying Unit Testing, TDD, BDD and Acceptance Testing" by David Sale. This book covers various testing techniques for Python applications, including unit testing, test-driven development (TDD), behaviour-driven development (BDD), and acceptance testing.* • *"Agile Testing: A Practical Guide for Testers and Agile Teams" by Lisa Crispin and Janet Gregory. This book provides an overview of agile testing, including how it differs from traditional testing methods and best practices for testing in an agile environment.* • *"The Art of Unit Testing: With Examples in .NET" by Roy Osherove. This book covers the basics of unit testing with examples in .NET, including how to write effective unit tests and how to use mocking frameworks to isolate dependencies.*

	• *"Selenium WebDriver Recipes in C#: Second Edition" by Zhimin Zhan. This book provides practical examples of using Selenium WebDriver, a popular automated testing tool, to test web applications in C#.*

2.3.6 Learning Programme PLO 5 – Documentation production [e-2]

Overall information PLO 5 – Documentation production [e-2]	
N. of Learning Units	*1*
Learning Outcomes	- *Describes types of technical documentation* - *Provides different (parts of) common technical documents, using appropriate tools (e.g., software documentation tools)*
Duration	*12 hours*
Total number of ECTS	*starting from n. 0,5 ECTS*
Recommendations for Micro-credentials	*May be suitable for Micro-credentials*
Often integrated with studies of PLO	PLOs 2 Application Development, PLOs 7,8,9
Recommended Didactical Approach	Virtual Classroom e-Learning Presence Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 70%* *Case study. Individual/team project up to 30%*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and models of software development documentation, the knowledge gained can form part of the final year project.*
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.6.1 Learning Units PLO 5 – Documentation production [e-2]

LU1	Software Documentation
Duration	*12 hours*
Didactical Approach and delivery method	*Virtual Classroom*
Additional information	-
Assessment	*Software documentation assignment using software documentation tools*
Title of the Learning material	*ESSA_Juniot_Developer_SOFTWARE_DOCUMENTATION*

2.3.7 Learning Programme PLO 6 – Problem management [e-2]

Overall information PLO 6 – Problem management [e-2]	
N. of Learning Units	*1*
Duration	*12 hours*
Learning Outcomes	*Systematically resolves or escalates incidents and problems, resulting in a solved incident e.g., by applying techniques and tools for troubleshooting such as diagnostic tools*
Total number of ECTS	*starting from n. 0,5 ECTS*
Recommendations for Micro-credentials	*Potentially could form part of a micro-credential*
Often integrated with studies of PLO	PLOs 2, 7 ,8 9
Recommended Didactical Approach	Virtual Classroom Presence Classroom
Additional comments	-
Recommended Delivery methods	*Lecture up to 80%* *Case study. Individual/team project up to 10%* *e-Learning up to 10%*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software problem management. These may be reinforced through practical tasks including class discussion, the case study and a multiple-choice quiz available for this learning Unit.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*After learning the basic principles, terminology, and processes in software problem management the knowledge gained can form part of class projects.*

Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.7.1 Learning Units PLO 6 – Problem management [e-2]

LU1	**Problem management**
Duration	*12 hours*
Didactical Approach and delivery method	*Lecture & eLearning*
Additional information	-
Assessment	*Case study & Multiple-choice quiz*
Title of the Learning material and file	*ESSA_Junior_Developer_Problem_Management.* *ESSA_Junior_Developer_Problem_Solving_Case_Study(EQF4)* *ESSA_Junior_Developer_Problem_Solving_Quiz(EQF4)* *ESSA_New Technologies_Junior Developer* **Additional Reading** *Agile for when things go wrong: the missing piece of your incident response plan* · *https://www.atlassian.com/agile/software-development/incident-response* *Incident Management: A Lost Art in the World of Software Engineering* · *https://betterprogramming.pub/incident-management-a-lost-art-in-the-world-of-software-engineering-f1bcac95a03* *Building an Effective Incident Management Process* · *https://www.infoq.com/articles/effective-incident-management/*

2.3.8 Learning Programme PLO 7 – Professional related competences [EQF5]

Overall information PLO 7 – Professional related competences [EQF5]	
N. of Learning Units	*1*
Learning Outcomes	- *Masters common ICT knowledge*

	- *Explains the principles, related concepts, advantages and disadvantages of a new technology. Applies and reports on basic methods, techniques and tools related to a new technology.* - *Applies and reports on measures, methods, tools and techniques related to security* - *Applies and reports on measures, methods, tools and techniques related to software lifecycle processes* - *Is aware of basic ethical considerations and issues*
Duration	*4 Weeks (across PLOs 7,8,9)*
Total number of ECTS	-
Recommendations for Micro-credentials	*This PLO should be an integral part of the initial studies for students with no prior knowledge of software development as it supports them in seeking employment after the programme and develops transversal skills which will be important in an industry setting.* *The format is likely unsuitable for Micro-credentials as it comprises a range of supports and activities rather formalized learning content*
Often integrated with studies of PLO	PLOs 2 Application Development; 3 Component Integration; 4 Testing; 5 Documentation Production; 6 New Technology; 7 Profession Related Competencies; and 9 Functioning in the Organisation
Recommended Didactical Approach	Virtual Classroom Presence Classroom Industry Mentor
Additional comments	*Professional related competencies are demonstrated through the students' interactions in the classroom, with the Industry Mentor and in the presentation of their final project to a panel consisting of tutors, industry and government representatives.*
Recommended Delivery methods	*Case study. Individual/team project up to 70%* *Industry Mentor up to 30%*
Additional comments	*Additional supports are provided to assist students with their CV and interview preparation, to support their professional competency development.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*The student project - "Software Pathway Project" three PLOs "PLOs 7 Profession Related Competencies; 8 Soft Competencies and 9 Functioning in the Organisation"*
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.8.1 Learning Units PLO 7 – Professional related competences [EQF5]

LU1	**Industry Related Software Development Project**

Duration	*4 weeks*
Didactical Approach and delivery method	*Student Project*
Additional information	*Assignment*
Assessment	*Tutor, Industry and government panel assessment*
Title of the Learning material	*ESSA_Software_Pathway_JUNIOR_DEVELOPER_(EQF4_7).docx* *Note document will need to be updated based on reviews and whether it will be used.*

2.3.9 Learning Programme PLO 8 – Soft competences [EQF5]

Overall information PLO 8 – Soft competences [EQF5]	
N. of Learning Units	*1*
Learning Outcomes	- *Works together with others in a team* - *Communicates with peers, colleagues, supervisors and/or relevant others, appropriately to the context, using conventions that are relevant to professional practice. Explains and gives instruction.* - *Masters the English language at level B2. Can understand the main ideas of complex text on both concrete and abstract topics, including technical discussions in his/her field of specialisation* - *Distinguishes and analyses fairly complex and unpredictable problems. Solves these problems systematically and in a creative way, using existing procedures and guidelines and own solutions by identifying and using data.* - *Exercises self-management within the guidelines of contexts that are usually predictable, but are subject to change. Is able to cope with limited change and to adapt to a certain level of variety in the workplace. Copes with pressure and stress setbacks and maintains composure. Shows some initiative and carries responsibility for the results of own activities, work and or study. Works correctly and carefully.* - *Realises learning and personal development on request, where necessary with support, through self-reflection and external- and self-evaluation of own (learning) results.*
Duration	*22 weeks (duration of the programme)*
Total number of ECTS	-
Recommendations for Micro-credentials	*The format is likely unsuitable for Micro-credentials as it comprises a range of supports and activities rather formalized learning content*
Often integrated with studies of PLO	PLOs 2 Application Development; 3 Component Integration; 4 Testing; 5 Documentation Production; 6 New Technology; 7 Profession Related Competencies; and 9 Functioning in the Organisation
Recommended Didactical Approach	Virtual Classroom Presence Classroom E-Learning

Additional comments	-
Recommended Delivery methods	*Case study. Individual/team project up to 70%* *e-Learning up to 30%*
Additional comments	*Lectures and e-learning are recommended for learning the basic principles, terminology, and models of software development. Soft competencies are gained through the practical experience of engaging in class discussions, engaging in the industry mentoring programme and participating in the Industry Software Project.*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*The student project - "Software Pathway Project" combined with Industry Mentoring addresses three PLOs "PLOs 7 Profession Related Competencies; 8 Soft Competencies and 9 Functioning in the Organisation" in addition to building on the knowledge gained through the other PLOs* *Soft skills are demonstrated through a range of activities over the course of the programme, including group work, active participation in the classroom setting, and attending guest lectures.*
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.9.1 Learning Units PLO 8 – Soft competences [EQF5]

LU1	**Industry Related Software Development Project**
Duration	*4 weeks*
Didactical Approach and delivery method	*Team Project*
Additional information	*Assignment by Tutor*
Assessment	*Tutor, Industry and government panel assessed*
Title of the Learning material	*ESSA_Software_Pathway_Project_JUNIOR_DEVELOPER_(EQF4_7)*

2.3.10 Learning Programme PLO 9 – Functioning in organisations [EQF5]

Overall information PLO 9 – Functioning in organisations [EQF5]	
N. of Learning Units	*1*
Learning Outcomes	- *Explains the basics of organisation theory and behaviour* - *Describes the relationship between business and IT* - *Works in an organisational context under specific direction with limited autonomy and responsibility e.g., at the level of a trainee, junior or assistant*

	- *Works in project settings, applies project management methods and tools* - *Writes a report on functioning in the organisation*
Duration	*4 Weeks (across PLOs 7,8,9)*
Total number of ECTS	-
Recommendations for Micro-credentials	*The format is likely unsuitable for Micro-credentials as it comprises a range of supports and activities rather formalized learning content*
Often integrated with studies of PLO	PLOs 2 Application Development; 3 Component Integration; 4 Testing; 5 Documentation Production; 6 New Technology; 7 Profession Related Competencies; and 9 Functioning in the Organisation
Recommended Didactical Approach	Presence Classroom e-Learning
Additional comments	-
Recommended Delivery methods	*Case study. Individual/team project up to 70%* *e-Learning up to 30%*
Additional comments	*Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software development. These are reinforced through the practical tasks undertaken during the course, participating in the industry mentoring programme and engaging in the final group project (Software Pathway Project)*
Work Based Learning Task *(If foreseen) and* Follow-up, learning reinforcement	*The student project - "Software Pathway Project" addresses three PLOs "PLOs 7 Profession Related Competencies; 8 Soft Competencies and 9 Functioning in the Organisation"*
Important (new) approaches and technologies to consider	*Including a guest speaker from industry or academic would enhance the learner's knowledge and afford them the opportunity to ask questions and gain insights into the role of a software developer in a business context.*
Training facilities (Link to ESSA learning material Platform)	*https://learn.softwareskills.eu/*

2.3.10.1 Learning Units PLO 9 – Functioning in organisation [EQF5]

LU1	**Industry Related Software Development Project**
Duration	*4 weeks*
Didactical Approach and delivery method	*Team Project*

Additional information	*Assignment by Tutor*
Assessment	*Tutor*
Title of the Learning material	*ESSA_Software_ Pathway_Project_JUNIOR_DEVELOPER_(EQF4_7)*

www.softwareskills.eu

Co-funded by the
Erasmus+ Programme
of the European Union