Module SYLLABUS

Organizational unit name	The Polytechnic Institute – Department of Computer Science
Field of study	Computer science
Module name	Internet application technologies and programming

Module code	POWER.IP.2	**Erasmus code**	11.3
ECTS	3	**Module type**	Optional
Year of study	4	**Semester**	7

Form of classes	**Hours total**	**Form of assessment**
Project classes	30	Graded credit

Coordinator teacher	mgr inż. Dariusz Piwko		
Academic teacher	mgr inż. Dariusz Piwko		
Language of instruction	English		
Basic courses	No	**Open course / course at he another field of study**	No
Profile of education	Practical profile	**Study level**	First-cycle level

Prerequisites and additional requirements
1. Basic knowledge of Internet technologies. 2. Basic knowledge of HTTP protocol. 3. Knowledge of MVC pattern for the architecture of applications. 4. Knowledge of using languages HTML/CSS 5. Knowledge of client-side programming. 6. Knowledge of server-side programming. 7. Knowledge of projecting databases.

Learning outcomes for module

No.	Student after module completion has the knowledge/knows how to/is able to Learning outcome code	Learning outcome type	Method of learning outcomes verification	Form of cla-sses Proj ect
1.	The Student is able to create web application based on MVC pattern.	Skills	project	Y
2.	The student is able to create web application using known programming languages and technologies/libraries.	Knowledge Skills	project	Y

3.	The student is able to use technical documentation of libraries and packages. He is also able to prepare documentation for the specific project.	Skills	technical documentation	Y
4.	The student is aware of how to cooperate in a project team. S/he can plan and coordinate the work of the team.	Social competence	project	Y

Didactic methods

Class form:
Classes have project character. The main goal is to create a web application using specific tools.

Didactic methods:
The progress of the application is verified during a consultation. The project is split into a several stages for better motivation.

Rules of assessment

1. Every student has to prepare a web application. Each project has to be done independently.
2. The project has to be created using object-oriented programming language and MVC architectural pattern.

Module content (brief)

1. Discuss project life cycle.
2. Choose and discuss project scope.
3. Specify functional and non-functional requirements.
4. Define system architecture.
5. Develop basic diagrams.
6. Create system.
7. Develop tests for the system.
8. Create technical documentation.
9. Improve technical vocabulary.

Module content (comprehensive)

Project classes consist of general rules of software development in IT:

1. Discuss project life cycle. Presentation of all phases and stages how to design project IT. Discuss project planning including determining and documenting a list of specific project goals.
2. Choose and discuss project scope, documenting a list of specific project tasks. Use frameworks/libraries based on MVC design pattern.
3. Specify functional and non-functional requirements. Prepare list of functionalities that solves particular problems for users.
4. Define a high-level system architecture split into client application and server application with database.
5. Prepare the architecture diagrams showing integration between frontend and backend and internal technical structure. Diagrams UML which describes use cases, ERD describes database and description of Application Interface (API).
6. System implementation using specific frameworks or libraries (e.g. Angular, ReactJS, VueJS, Spring Boot, Symfony, etc.). System consists of client part and server application which cooperate with database (PostgreSQL, MySQL, MongoDB etc.).
7. Specify testing application flow. The main goals are integrated tests which eliminate a lot of common errors.
8. Preparing technical documentation which describes each phase of the created application.
9. Improve technical and IT vocabulary.

Recommended literature and teaching resources

1. Craig Walls: Spring in Action, Fifth Edition, Manning Publications, 2018.
2. Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich: Node.js in Action, Manning Publications, 2014.
3. Nathan Murray, Felipe Coury, Ari Lerner, Carlos Taborda: ng-book: The Complete Guide to Angular, CreateSpace Independent Publishing Platform, 2018.
4. Olga Filipoca: Learning Vue.js 2, Packt Publishing, 2016.
5. PostgreSQL, MySQL, MongoDB - technical documentation available in Internet.
6. Symfony 4.2 Documentation - available in Internet.

Connection with area of study	engineering sciences
Student workload (ECTS credits balance)	
Student workload form	Student workload [hours]
Participation in project classes	30
Completion of a project	40
Individual consultations and final project presentation	10
Summary student workload	75
Module ECTS credits	
Workload of the direct assistance of the academic teacher	**1.6**
Workload of the practical classes	**3**

Annotation:
1 hour = 45 minutes