Software Engineering Semester 1 Year 3

U5.1 Software Engineering II N/I:

	Microservices Architecture
Module designation	Software Engineering II
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.1
Subtitle, if applicable	
Courses, if applicable	Microservices Architecture
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Ms. Jihen Hadhli
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester 21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/semester (35 hours of Self-Study/semester)
Credit points	2
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Strong programming skills are a prerequisite for this advanced microservices course, ensuring students can efficiently implement microservices solutions.
Module objectives/intended learning outcomes	 Learning Outcomes: Upon completion of this teaching module, the student will be able to: By course completion, students will possess the skills to design, develop, and deploy microservices architectures for scalable and efficient distributed systems. They will be able to make informed architectural decisions, implement security measures, and handle data in a distributed, microservices-based ecosystem. The course ensures students grasp industry best practices, fostering a deep understanding of real-world microservices applications. Ultimately, graduates will emerge as proficient microservices engineers capable of contributing to mitrice advancement of the inclusion.

Content	Chapter 1: Introduction to Microservices
	1. Introduction
	2. Fundamentals of Microservices
	3. Principles of Microservices
	4. Microservices architecture
	Practical Work:
	TP1: Build a simple microservices-based application and assess its benefits and challenges.
	Chapter 2: Design and Development of Microservices
	1. Microservices Design Principles
	2. Implementing Microservices
	4. Testing strategies for microservices
	5. Continuous integration and continuous deployment (CI/CD)
	Practical Work:
	TP2: Design and implement a microservices architecture with CI/CD for a real-world problem.
	Chapter 3: Microservices Communication and Orchestration
	1. Communication Protocols
	2. Containerization and Orchestration
	3. Service discovery and load balancing
	Practical Work:
	TP3: Containerize and orchestrate a microservices app with Docker and Kubernetes, including service
	discovery and load balancing.
	Chapter 4: Data Management in Microservices
	1. Data Storage in Microservices
	2. Event-Driven Microservices
	3. Implementing microservices using Apache Kafka
	Practical Work:
	TP4: Develop a microservices system using event sourcing and CQRS with Apache Kafka integration.
	Chapter 5: Advanced Topics and Case Studies
	1. Microservices Security and Scaling

	2. Monitoring and Troubleshooting
	3. Scalability and performance optimization
	Practical Work:
	TP5: Strengthen microservices security, optimize performance, and create a monitoring and troubleshooting plan.
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%) (Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection
Reading list	1. Fowler, M. (2014). "Microservices: a definition of this new architectural term."
	 Newman, S. (2015). "Building Microservices: Designing Fine-Grained Systems." O'Reilly Media.
	3. Wiggins, R., & Jones, N. (2018). "Mastering Microservices." Packt Publishing.
	4. Tilkov, S., & Vinoski, S. (2015). "Microservices: From Design to Deployment." O'Reilly Media

DevOps	
Module designation	Software Engineering II
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.1
Subtitle, if applicable	
Courses, if applicable	DevOps
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Ms. Belgacem Hajji
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/semester (35 hours of Self-Study/semester)
Credit points	2
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	 Knowledge of programming languages and software development principles.
	• Familiarity with version control systems (e.g., Git).
	• Proficiency in using the command line in Linux/Unix environments.
	\circ Understanding of system and network administration concepts.

U5.1 Software Engineering II

Module objectives/intended learning outcomes	Learning Outcomes:
	Upon completion of this teaching module, the student will be able to:
	• Understanding of devops Principles.
	• Use version control systems like Git to manage source code and collaborate effectively.
	• Set up CI pipelines to automate code integration and testing.
	• Implement CI best practices to ensure code quality.
	• Deploy applications and infrastructure changes automatically using CD pipelines.
	• Understand containerization concepts and technologies like Docker
	 Deploy and manage containerized applications in Kubernetes clusters.
	• Write infrastructure as code using tools like Terraform or Ansible.
	• Set up monitoring and alerting systems to track the health and performance of applications and infrastructure

Content	Chapter 1: Introduction to DevOps
	1. Understand the basics of DevOps, its principles, and benefits.
	2. Explore the history and evolution of DevOps.
	3. Read case studies and real-world examples of DevOps success.
	Chapter 2: Source Code Management
	1. Learn about version control systems (e.g., Git).
	2. Explore branching, merging, and best practices in source code management.
	3. Set up your Git repository, practice basic commands.
	Chapter3: Continuous Integration (CI)
	1. Study CI concepts and its importance in DevOps.
	2. Hands-on experience with popular CI tools like Jenkins.
	3. Create a simple CI/CD pipeline for a sample application.
	Chapter 4: Containerization and Orchestration
	1. Learn about Docker containers and their benefits.
	2. Explore container orchestration tools such as Kubernetes.
	3. Deploy a containerized application on a Kubernetes cluster.
	Chapter 5: Configuration Management
	1. Understand the importance of infrastructure as code (IaC).
	2. Learn tools like Ansible management.
	3. Write playbooks or manifests for managing server configurations.
	Chapter 6: Monitoring and Logging
	 Study the importance of monitoring and logging in DevOps.
	2. Explore tools like Prometheus, Grafana, ELK Stack for monitoring and logging.
	3. Set up monitoring and logging for your sample

	application.
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%) (Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT) Video projection
Reading list	 <u>https://www.docker.com/</u> <u>https://kubernetes.io/</u> <u>https://docs.ansible.com/</u> <u>http://git.com/</u>

U5.2 Advanced Web Programming

Python Web Framework (Django)

Module designation	Advanced Web Programming
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.2
Subtitle, if applicable	
Courses, if applicable	Python Web Framework (Django)
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Mr. Imed Hammadi
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/semester (35 hours of Self-Study/semester)
Credit points	3
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Web technologies such as HTML and CSS Python programming
Module objectives/intended learning outcomes	 Course Objective: The objective of this course is to provide students with the skills to build dynamic, secure, and scalable web applications using the Django framework. Students will learn how to structure projects, interact with databases, manage templates and forms, implement authentication, and deploy applications following best practices. Learning Outcomes: By the end of this course, students will be able to: Understand Django's architecture and development workflow Create dynamic web applications with URL routing, views, and templates Build and manage databases using Django's ORM Handle user input securely through forms Implement user authentication and authorization Customize the Django admin for content management

Content	Chapter 1: Introduction to Django
	1. MVC/MVT architecture
	2. Django installation and project structure
	Chapter 2: URL Routing and Views
	1. URL patterns and dynamic routing
	2. Function-based and class-based views
	Chapter 3: Templates and Static Files
	1. Template language, inheritance
	2. Static files and media management
	Chapter 4: Models and ORM
	1. Defining models and relationships
	2. ORM queries, migrations
	Chapter 5: Forms and User Input
	1. Django forms, ModelForms
	2. Form validation and handling
	Chapter 6: Authentication and Authorization
	1. User registration, login, logout
	2. Permissions and access control
	Chapter 6: Admin Interface and Customization
	1. Managing content through Django admin
	2. Admin customization
	Chapter 7: Testing and Debugging
	1. Unit tests with Django's testing framework
	2. Debugging tools
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%) (Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection

Reading list	 Adrian Holovaty & Jacob Kaplan-Moss – The Django Book (Online version)
	2. Antonio Mele – Django 4 by Example, Packt Publishing
	 William S. Vincent – Django for Beginners, Django for APIs, Django for Professionals
	4. Django Documentation – https://docs.djangoproject.com

.NET Technologies	
Module designation	Advanced Web Programming
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.2
Subtitle, if applicable	
Courses, if applicable	.NET Technologies
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Mr. Imed Ben Boukhatem
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester
	21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/semester (35 hours of Self-Study/semester)
Credit points	3
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Web technologies such as HTML and CSS Java programming

U5.2 Advanced Web Programming

Module objectives/intended learning outcomes	Course Objective: The objective of this course is to provide students with practical and theoretical knowledge to build secure, scalable, and modern web applications using ASP.NET Core. The course focuses on the core components of the .NET platform, including MVC architecture, Entity Framework Core, dependency injection, middleware, and API development.
	Learning Outcomes:
	By the end of this course, students will be able to:
	• Understand the architecture and components of ASP.NET Core
	 Develop interactive web applications using the MVC pattern
	 Integrate and manage databases using Entity Framework Core
	 Apply dependency injection and middleware in application development
	 Implement secure authentication and authorization systems
	• Build and document RESTful APIs for client consumption

Content	Chapter 1: Introduction to .NET and ASP.NET Core
	1NET ecosystem overview
	2. Project structure and lifecycle
	Chapter 2: MVC Architecture in ASP.NET Core
	1. Controllers, views, and models
	2. Routing and middleware pipeline
	Chapter 3: Razor Pages and Blazor
	1. Razor syntax and view rendering
	2. Overview of component-based web development
	Chapter 3: Entity Framework Core and Data Access
	1. Code-first approach
	2. Migrations, LINQ queries
	Chapter 4: Dependency Injection and Services
	1. Registering and injecting services
	2. Scoped, transient, and singleton lifetimes
	Chapter 5: Authentication and Authorization
	1. Identity framework integration
	2. Role-based and policy-based access control
	Chapter 6: Building RESTful APIs
	1. API controllers
	2. Serialization, versioning, and Swagger documentation
	Chapter 7: Testing and Debugging
	1. Unit testing with xUnit
	2. Logging and diagnostics
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%) (Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection

Reading list	 Andrew Lock – ASP.NET Core in Action, Manning Publications
	2. Adam Freeman – Pro ASP.NET Core MVC, Apress
	3. Microsoft Docs – ASP.NET Core – https://docs.microsoft.com/en-us/aspnet/core
	 Jon Galloway & Contributors – Introduction to ASP.NET Core (Microsoft eBook)

Module designation	Advanced Web Programming
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.2
Subtitle, if applicable	
Courses, if applicable	Front-end Framework
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Ms. Salma Bouazizi
Language	English
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester
	21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/semester (35 hours of Self-Study/semester)
Credit points	3
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Web technologies such as HTML and CSS
	JavaScript programming
Module objectives/intended learning outcomes	Course Objective: The objective of this course is to enable students to build dynamic, modern, and maintainable single-page applications (SPAs) using Angular. The course covers the core concepts of Angular including components, services, routing, forms, and HTTP communication, with a strong focus on modular design, reusability, and best practices for front-end development. Learning Outcomes: By the end of this course, students will be able to: • Set up and structure Angular applications using Angular
	 CLI Develop reusable and dynamic components Manage data flow using services and observables
	• Implement client-side routing and navigation

Front-end Framework

Content	Chapter 1: Introduction to Angular	
	1. What is Angular and why use it	
	2. Angular CLI and project setup	
	Chapter 2: Components, Templates, and Directives	
	1. Creating and nesting components	
	2. Data binding and event handling	
	3. Structural and attribute directives	
	4. Custom directives and pipes	
	Chapter 3: Services and Dependency Injection	
	1. Creating services	
	2. Injecting and sharing data across components	
	3. Router module, route parameters	
	4. Lazy loading and route guards	
	Chapter 4: Reactive and Template-driven Forms	
	1. Form validation and dynamic forms	
	2. FormBuilder and observables	
	Chapter 5: HTTP Client and APIs	
	1. Sending requests and handling responses	
	2. Error handling and interceptors	
	3. Managing application state	
	4. Basic use of RxJS	
	Chapter 6: Deployment and Best Practices	
	1. Building and optimizing Angular apps	
	2. Hosting options and production deployment	
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%) (Report for each workshop/Project required)	
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)	
	Video projection	

Reading list	1. Minko Gechev – Angular: Up and Running, O'Reilly
	2. Stephen Radford – Beginning Angular with Typescript, Apress
	3. Angular Official Documentation – https://angular.io/docs
	4. Fireship.io – Angular crash courses and advanced tips

U5.3]	Mobile	and Java	Programn	ning
--------	--------	----------	----------	------

Module designation	Mobile and Java Programming
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.3
Subtitle, if applicable	
Courses, if applicable	Mobile Programming
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Ms. Hamza Mejri
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/semester (35 hours of Self-Study/semester)
Credit points	3
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Object Oriented Programming (Java) Object Oriented Design
Module objectives/intended learning outcomes	 Objectives: Learn to program mobile apps for the Android program using the Kotlin programming language and an IDE. Work individually and in groups to develop mobile apps. Learning outcomes: Recognize the capabilities and limitations of mobile devices Design mobile applications with more than one user interface and more than one system component. Develop mobile applications on a popular mobile development platform.

Mobile Programming

Content	Chapter 1: Introduction to Android Development
	1. State of the Art of Mobile Applications
	2. Specificities of mobile development
	3. App monetization
	4. Android platform architecture
	Practical Work:
	• Android Studio installation and setup
	Chapter 2: KOTLIN Basics
	1. Introduction
	2. Variables and constants
	3. Functions
	4. Conditional structures
	5. Loops
	6. Arrays
	7. Classes
	Practical Work:
	• Developing a simple management application using KOTLIN.
	Chapter 3: User Interface Design
	1. Android layouts and views
	2. XML layout design
	3. User interface best practices
	4. Adapters and Dynamic Layout
	5. Fragments
	6. Jetpack compose
	Practical Work:
	• Create a simple discount calculation application using basic graphic elements (constraintLayout, button, seekBar, EditText) and DataBinding
	• Create the same application using Jatpack Compose

	technique
Chapt	er 4: Activity and Intent
1.	Activities and their lifecycle
2.	Implicit and explicit intents
3.	Passing data between activities
4.	Activity navigation
5.	Inflate menu
Practi	cal Work:
•	Create an application that allows to create a quiz game containing multiple choice question regarding the flags of different countries
Chapt	er 5: Data persistence
1.	Shared Preferences
2.	Internal Storage
3.	External Storage
4.	SQLite Databases
5.	Network storage
6.	Broadcast, ContentProvider
Practi	cal Work:
•	Create a directory application allowing management and backup information (last name, first name, email, telephone number, address, etc.) concerning people.
Chapt	er 6: Networking and Web Services
1.	HTTP requests with HttpURLConnection or Retrofit
2.	Parsing JSON data
3.	Handling network operations asynchronously
4.	Error handling and network security
Practi	cal Work:
•	Develop the MeteoApp application which uses the free REST web services of the OpenWeatherMap.org site to get the weather forecast

	Chapter 7: Services and Threads
	1. Types of Services
	2. Service Life Cycle
	3. Threads on Android
	4. Coroutines
	Chapter 8: Working with Sensors
	1. Introduction to Android libraries
	2. Integrating Google Maps API
	3. Using third-party libraries
	Practical Work:
	• Creating a Location-Based Service App
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%) (Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection
Reading list	1. Documentation and tutorials on developer.android.com
	 Michael Fazio, Kotlin and Android Development featuring Jetpack, The Pragmatic Programmers, 2021

U5.3 Mobile and Java Programming

Advanced Java Programming (OCA Certification)

Module designation	Mobile and Java Programming
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.3
Subtitle, if applicable	
Courses, if applicable	Advanced Java Programming (OCA Certification)
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Mr. Hamza Mejri
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours Lecture/ semester21 hours practical workshop in Lab/ semester
Workload	Total 77 hours/semester (35 hours of Self-Study/semester)
Credit points	3
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Basic Java Programming
Module objectives/intended learning outcomes	Course Objective: This course aims to equip students with advanced Java programming skills, focusing on exception management, generics, functional programming with streams, multithreading, and network communication using Java APIs. It prepares learners for real-world applications requiring robust, efficient, and concurrent Java solutions.
	Learning Outcomes:
	By the end of this course, students will be able to:
	Create robust applications using advanced exception handling techniques
	• Design reusable and type-safe code using generics
	• Implement multithreaded applications with proper synchronization
	Build networked Java applications using socket programming
	• Work with files, serialization, and Java I/O APIs effectively

Content	Chapter 1: Java Basics recall
	Chapter 2: Advanced Exception Handling
	1. Custom exceptions
	2. Multi-catch, try-with-resources
	3. Exception chaining and best practices
	Chapter 3: Generics in Java
	1. Generic methods and classes
	2. Wildcards (? extends, ? super)
	3. Type bounds and type erasure
	Chapter 4: Java Collections Framework
	1. Comparator vs Comparable
	2. NavigableMap, LinkedHashSet, PriorityQueue
	3. Performance tuning and immutability
	Chapter 5: Java Streams and Functional Programming
	1. Stream API, pipelines, and collectors
	2. Lambdas and method references
	3. Optional and functional interfaces
	Chapter 6: Multithreading and Concurrency
	1. Threads, Runnable, and ExecutorService
	2. Synchronization, locks, and race conditions
	 Concurrency utilities (Semaphore, CountDownLatch, etc.)
	Chapter 7: Networking in Java
	1. Sockets (TCP/UDP), ServerSocket
	2. Client-server architecture
	3. Java HTTP client API
	Chapter 8: File I/O and Serialization
	1. NIO vs IO
	2. File streams, buffering

	3. Object serialization/deserialization
	Chapter 9: Testing and Debugging Techniques
	1. Unit testing with Junit
	2. Logging with Log4j/SLF4J
	3. Java debugging tools (jdb, breakpoints, etc.)
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection
Reading list	1. Joshua Bloch – Effective Java, Addison-Wesley
	 Brian Goetz – Java Concurrency in Practice, Addison- Wesley
	 Cay S. Horstmann – Core Java Volume II – Advanced Features, Pearson
	4. Oracle Java Documentation – https://docs.oracle.com/javase

Blockchain Technology		
Module designation	Information Technologies	
Module level, if applicable	Year 3, Semester 1	
Code, if applicable	U5.4	
Subtitle, if applicable		
Courses, if applicable	Blockchain Technology	
Semester (s) in which the module is taught	Semester 5	
Person responsible for the module	Dept Head	
Lecturer	Mr Mahmoud Somrani	
Language	French	
Relation to curriculum	Compulsory module	
Type of teaching, contact hours	21 hours Lecture/ semester	
Workload	Total 42 hours/semester (21 hours of Self-Study/semester)	
Credit points	2	
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams	
Recommended prerequisites	None	
Module objectives/intended learning outcomes	 Learning Outcomes: Upon completion of this teaching module, the student will be able to: Understand the fundamental concepts of blockchain technology, including decentralized ledger, cryptographic hashing, and consensus mechanisms. Analyze the benefits and challenges of blockchain technology in various applications. Design and implement basic blockchain applications using smart contracts and decentralized applications (DApps). Evaluate security considerations and best practices in blockchain development. Critically assess real-world blockchain use cases and their impact on different industries. 	

U5.4 Information Technologies

Content	Chapter 1: Introduction to Blockchain
	1. What is Blockchain?
	2. History and Evolution of Blockchain Technology
	 Key Terminologies: Blocks, Decentralization, Consensus Mechanisms
	4. Use Cases and Importance of Blockchain
	Chapter 2: Blockchain Architecture
	1. Blockchain Components: Nodes, Transactions, Blocks
	2. Public vs. Private Blockchains
	3. Blockchain Protocols: Bitcoin, Ethereum, Hyperledger
	4. How Mining Works
	Chapter 3: Cryptography in Blockchain
	1. Role of Cryptography in Blockchain
	2. Public and Private Keys
	3. Digital Signatures
	4. Hash Functions
	Chapter 4: Consensus Mechanisms
	1. Proof of Work (PoW)
	2. Proof of Stake (PoS)
	3. Delegated Proof of Stake (DPoS)
	4. Practical Byzantine Fault Tolerance (PBFT)
	Chapter 5: Smart Contracts
	1. Understanding Smart Contracts
	2. Introduction to Solidity (Ethereum)
	3. Writing and Deploying a Simple Smart Contract
	4. Use Cases for Smart Contracts
	Chapter 6: Blockchain Challenges and Limitations
	1. Scalability and Performance Issues
	2. Security Concerns

	3. Regulatory and Legal Challenges
	4. Environmental Impact
	Chapter 7: Real-world Applications
	1. Blockchain in Finance and Banking
	2. Supply Chain Management
	3. Healthcare and Pharmaceuticals
	4. Voting and Governance Systems
	5. Cross-Industry Use Cases
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection
Reading list	 D. Drescher, Blockchain Basics: A Practical Approach, 2018
	2. P. Vigna, The Truth Machine: The Blockchain and the Future of Everything, 2018
	3. S. Vyasl, Blockchain Technology Exploring Opportunities, Challenges, and Applications, 2022

U5.4 Information Technologies

Information Technology Management (ERP, CRM)

Module designation	Information Technologies
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U5.4
Subtitle, if applicable	
Courses, if applicable	Information Technology Management (ERP, CRM)
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Mr. Imed Hammadi
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours Lecture/ semester
Workload	Total 42 hours/semester (21 hours of Self-Study/semester)
Credit points	2
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Database and Software Engineering
Module objectives/intended learning outcomes	Objectives: The objective of this course is to provide the student with knowledge related to the implementation of integrated management software packages, ERP. The course deals with the roles and issues of integrated systems in a project management context and will allow the student to evaluate the technological needs of a company during the implementation and configuration of integrated systems. The student should understand what an ERP and/or CRM is, and what issues they address for the company

Content	Course elements:
	1. The company and the management information system - importance of the IS
	2. Definition and role of an ERP/CRM.
	3. Methodology and selection criteria.
	4. Presentation of the different modules of an ERP
	5. Functioning and integration process.
	6. Configuring a mini case in an open-source ERP.
	7. Administration and advanced concepts.
	Content of the Practical Work:
	• PW 1: Installation and configuration of an open-source ERP: Odoo
	• PW 2: Database initialization (Third party, Products) and ERP handling
	• PW 3: Workflows and business processes
	• PW 4: Realization of a commercial workflow (Quotes, orders, invoices).
	• PW 5: Realization of a manufacturing workflow (PO, Bill of Materials).
	• PW6: Discovery of the CRM functionalities
	• PW 7: Initialization of accounts, contacts, prospects
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
2 <i>1 1</i>	
Reading list	Mainly training materials and official ERP documentation on the web. (www.odoo.com)

U5.5 Advanced Databases

Database Administration		
Module designation	Advanced Databases	
Module level, if applicable	Year 3, Semester 1	
Code, if applicable	U5.5	
Subtitle, if applicable		
Courses, if applicable	Database Administration	
Semester (s) in which the module is taught	Semester 5	
Person responsible for the module	Dept Head	
Lecturer	Mr. Houssem Mahmoudi	
Language	French	
Relation to curriculum	Compulsory module	
Type of teaching, contact hours	21 hours Lecture/ semester21 hours practical workshop in Lab/ semester	
Workload	Total 63 hours/semester (21 hours of Self-Study/semester)	
Credit points	2	
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams	
Recommended prerequisites	Database fundamentals, SQL	
Module objectives/intended learning outcomes	Course Objective: This course aims to provide students with the knowledge and skills required to install, configure, manage, monitor, and secure relational database systems. It emphasizes practical database administration tasks such as user management, performance tuning, backup and recovery, and ensuring data integrity and availability.	
	Learning Outcomes:	
	By the end of this course, students will be able to:	
	• Install and configure a relational database system	
	• Manage users, permissions, and security settings	
	 Perform and automate regular backups and recovery procedures 	
	Monitor and optimize database performance	
	• Ensure data availability through replication and high- availability solutions	
	• Apply auditing and compliance techniques in a secure environment	

Databas Administrati

Content	Chapter 1: Introduction to Database Administration
	1. DBA roles and responsibilities
	2. Types of DBMS and architectures
	3. Setting up DBMS (e.g., MySQL, PostgreSQL, Oracle)
	4. Configuration files and system requirements
	Chapter 2: User and Security Management
	1. Creating and managing users and roles
	2. Access control, privileges, and authentication
	Chapter 3: Backup and Recovery Strategies
	1. Logical vs physical backups
	2. Recovery models and restore techniques
	Chapter 4: Storage and Space Management
	1. Tablespaces, partitions, and datafiles
	2. Monitoring disk usage
	Chapter 5: Performance Tuning and Optimization
	1. Indexing, query optimization
	2. Monitoring tools and performance metrics
	3. Logs, alerts, and health checks
	4. Automating tasks with schedulers or cron jobs
	Chapter 6: High Availability and Replication
	1. Replication types and configurations
	2. Clustering and failover strategies
	Chapter 7: Data Integrity and Auditing
	1. Constraints, transactions, and logging
	2. Auditing user activity
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection

Reading list	 Oracle Documentation – Oracle Database Administrator's Guide
	2. PostgreSQL Docs - https://www.postgresql.org/docs/
	 Christian Antognini – Troubleshooting Oracle Performance, Apress
	4. Grant Fritchey – SQL Server Execution Plans, Red Gate
	5. Linux and Shell Basics – for database server maintenance and automation

NoSQL Databases		
Module designation	Advanced Databases	
Module level, if applicable	Year 3, Semester 1	
Code, if applicable	U5.5	
Subtitle, if applicable		
Courses, if applicable	NoSQL Databases	
Semester (s) in which the module is taught	Semester 5	
Person responsible for the module	Dept Head	
Lecturer	Mr. Houssem Mahmoudi	
Language	English	
Relation to curriculum	Compulsory module	
Type of teaching, contact hours	21 hours Lecture/ semester	
	21 hours practical workshop in Lab/ semester	
Workload	Total 63 hours/semester (21 hours of Self-Study/semester)	
Credit points	2	
Requirements according to the	Minimum attendance rate: 80% of the total contact hours	
examination regulations	>20 % of nonattendance = elimination for exams	
Recommended prerequisites	Big Data	
Module objectives/intended learning outcomes	 Learning Outcomes: Upon completion of this teaching module, the student will be able to: Define and explain the foundational concepts of NoSQL databases. Distinguish between various NoSQL database types, including document, key-value, column-family, and graph databases. Design, model, and implement database solutions using NoSQL technologies. Analyze the advantages, use-cases, and limitations of NoSQL databases over traditional RDBMS. Apply consistency, availability, and partition tolerance (CAP theorem) principles in the context of NoSQL databases. Evaluate security and performance considerations specific to NoSQL databases. Implement scaling and replication strategies for NoSQL databases 	

U5.5 Advanced Databases

Content	Chapter 1: Introduction to NoSQL
	1. What is NoSQL?
	2. History and evolution of NoSQL
	3. Why NoSQL?
	Chapter 2: Types of NoSQL Databases
	1. Document databases (e.g., MongoDB)
	2. Key-Value stores (e.g., Redis)
	3. Column-family stores (e.g., Cassandra)
	4. Graph databases (e.g., Neo4j)
	Chapter 3: NoSQL Database Design
	1. Data modeling for NoSQL
	2. Schema-less design considerations
	3. Use case Design task
	Chapter 4: CAP Theorem
	1. Consistency
	2. Availability
	3. Partition tolerance
	Chapter 5: Performance and Scaling
	1. Scaling strategies
	2. Replication and sharding
	3. Advanced case
	Chapter 6: Security in NoSQL
	1. Access control
	2. Encryption techniques
	Chapter 7: Real-world Applications and Integration
	1. Use cases for NoSQL databases
	2. Integration in modern web applications
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)

Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT) Video projection
Reading list	 "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" by Martin Fowler and Pramod Sadalage."
	 Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement" by Luc Perkins, Eric Redmond, and Jim R. Wilson. "D. SOL for March 1." In Description.
	3. "NoSQL for Mere Mortals" by Dan Sullivan.

U5.6 Entrepreneurship and Marketing

Entrepreneurship and Innovation

Entrepreneurship and Marketing
Year 3, Semester 1
U5.6
Entrepreneurship and Innovation
Semester 5
Dept Head
Ms. Neila Mouihbi
French
Compulsory module
21 hours of Supervision on Campus/ semester
Total 31.5 hours/semester (10.5 hours of Self-Study/semester)
1.5
Minimum attendance rate: 80% of the total contact hours $>20\%$ of nonattendance = elimination for exams
None
 Course Objective: The course aims to equip students with the entrepreneurial mindset and tools necessary to identify opportunities, develop innovative solutions, and launch successful ventures. It blends theory and practice to foster creativity, critical thinking, and business model development in dynamic environments. Learning Outcomes: By the end of this course, students will be able to: Recognize and evaluate business opportunities in various sectors Develop innovative and viable business models

Content	Module 1: Introduction to Entrepreneurship
	Module 2: Opportunity Identification and Market Research
	Module 3: Innovation and Creativity
	Module 4: Business Models and Value Proposition
	Module 5: Building a Startup Strategy
	Module 6: Financial Planning and Funding
	Module 7: Pitching and Communication Skills
	Final Project: Startup Pitch
	Team-based project: pitch a new venture idea
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT)
	Video projection
Reading list	1. Eric Ries – The Lean Startup, Crown Business
	 Alexander Osterwalder & Yves Pigneur – Business Model Generation, Wiley
	3. Steve Blank – The Startup Owner's Manual, K&S Ranch
	4. Guy Kawasaki – The Art of the Start 2.0, Penguin
	 MIT OpenCourseWare – Entrepreneurship and Innovation Series

U5.6 Entrepreneurship and Marketing

Module designation Entrepreneurship and Marketing	
Module level if applicable	Vegr 3 Semester 1
Code, if applicable	05.6
Subtitle, if applicable	
Courses, if applicable	Digital Marketing
Semester (s) in which the module is taught	Semester 5
Person responsible for the module	Dept Head
Lecturer	Ms. Neila Mouihbi
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours of Supervision on Campus/ semester
Workload	Total 31.5 hours/semester (10.5 hours of Self-Study/semester)
Credit points	1.5
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours
	>20 % of nonattendance = elimination for exams
Recommended prerequisites	None
Module objectives/intended learning outcomes	 Objective: In this module, student will be introduced to digital marketing. Learning Outcomes: Upon completion of this course, students should be able to: develop and execute a marketing plan understand the role of digital media in marketing establish a marketing analytics and quantitative evaluation of the marketing environment

Content	1. Introduction to Digital Marketing
	2. Search Engine Optimisation (SEO)
	3. Search Engine Marketing
	4. Social Media Marketing
	5. Content Marketing & Strategy
	6. Web Analytics
	7. Google Tag Manager
	8. Display Advertising
	9. Web Remarketing
	10. Email Marketing/ Mobile Marketing
	11. E-Commerce / Online Reputation Management
	12. Adsense, Blogging, and Affiliate Marketing
	13. Analytics and Reporting
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT) Video projection
Reading list	1. Maccoby, M. (2000). Understanding the Difference
	between Management and Leadership, p. 57.
	 Rosenbach, W. E., Taylor, R. L., & Youndt, M. A. (2012). Contemporary Issues in Leadership, Leadership, Chapter 1, Transcendent Leadership, chapter 4, Summit Leadership, Chapter 5.
	3. Walker, R., & Aritz, J. (2014). Leadership Talk: A Discourse Approach to Leader Emergence, Chapter 1.