TECHNISCHE HOCHSCHULE LÜBECK

Module: Web Services

Level	Bachelor	**Short Name**	WebSvc
Responsible Lecturers	Nane Kratzke, Prof. Dr.		
Department, Facility	Electrical Engineering and Computer Science		
Course of Studies	International Track		
Compulsory/elective	Compulsory	**ECTS Credit Points**	5
Semester of Studies	(Unspecified)	**Semester Hours per Week**	4
Length (semesters)	1	**Workload (hours)**	150
Frequency	(Flexible)	**Presence Hours**	60
Teaching Language	German/English	**Self-Study Hours**	90

The following section is filled only if there is **exactly one** module-concluding exam.

Exam Type	Project Work	**Exam Language**	German/English
Exam Length (minutes)		**Exam Grading System**	One-third Grades

Learning Outcomes

- Students understand the basic concepts, architectures and technologies required for developing and using web services and web-scale systems (including but not limited to SOAP, REST, JSON, XML and WSDL).
- Students understand how web services exchange, process, and store data. They also learn how to handle different data formats and the importance of data modelling and serialization.
- Students can design and implement RESTful web services and APIs that are clear, consistent, secure, and comply with REST principles.
- Students can design simple event-driven web services and integrate them using messaging systems such as Kafka, RabbitMQ, or similar systems.
- Students can integrate existing web services into their applications, including third-party APIs and are aware of the importance of API keys, OAuth and other mechanisms for integrating and authenticating within web services.
- Students can apply the concepts and techniques of asynchronous implementation of web services, including the use of promises, async/await and callbacks. They understand the benefits of asynchronous processing for improving the scalability and performance of web applications.
- Students can design and implement web services that efficiently and asynchronously handle I/O operations and enable non-blocking communication.
- Students have gained practical experience with modern development tools and frameworks for web services, such as Postman, Swagger (OpenAPI), Spring Boot for Java, and FastAPI for Python (or comparable technologies).

	• Students can design web services for microservice architectures that consider the 12-factor rules and deploy and manage them automatically in a Kubernetes environment. • Students understand the advantages of using containers for the provision and scaling services, the configuration of Kubernetes pods, services and deployments and how to ensure resilience and load balancing.
Participation Prerequisites	A successful completion of the Web and Cloud Computing Project module is strongly recommended.

The previous section is filled only if there is **exactly one** module-concluding exam.

Consideration of Gender and Diversity Issues	✔ Use of gender-neutral language (THL standard) ✘ Target group specific adjustment of didactic methods ✘ Making subject diversity visible (female researchers, cultures etc.)
Applicability	The module deepens the content of the web and cloud computing project's content for students particularly interested in web-scale systems.
Remarks	

TECHNISCHE HOCHSCHULE LÜBECK

Module Course: Web Services (Lecture)

(of Module: Web Services)

Course Type	Lecture	**Form of Learning**	Online supported with presence hours
Mandatory Attendance	no	**ECTS Credit Points**	1
Participation Limit	24	**Semester Hours per Week**	1
Group Size	12	**Workload (hours)**	30
Teaching Language	German/English	**Presence Hours**	15
Study Achievements ("Studienleistung", SL)		**Self-Study Hours**	15
SL Length (minutes)		**SL Grading System**	

The following section is filled only if there is a course-specific exam.

Exam Type		**Exam Language**	
Exam Length (minutes)		**Exam Grading System**	
Learning Outcomes			
Participation Prerequisites			

The previous section is filled only if there is a course-specific exam.

Contents	• The basics of web services and web-scale systems: An introduction to SOAP, REST, JSON, XML, and WSDL. Understanding the different architectures and technologies. • Data management in web services: dealing with data formats, data modelling, and serialization; methods for data exchange, processing, and storage. • Design of RESTful web services: Design and implementation of RESTful APIs, ensuring clarity, consistency, and security. • Event-driven web services: Designing and integrating event-driven web services with messaging systems such as Kafka and RabbitMQ. • Integration and use of third-party APIs: Use of API keys, OAuth, and other mechanisms for integration and authentication. • Development tools and frameworks: Practical experience with tools such as Postman, Swagger (OpenAPI), Spring Boot, and FastAPI development (or similar technologies). • Microservice architectures and the 12-factor app: This includes designing web services for microservices, applying the 12-factor rules, and automating deployment and management in Kubernetes. • Containerisation and Kubernetes: advantages of containers, configuration of Kubernetes pods, services, deployments, reliability and load balancing.

Literature	Nane Kratzke, Cloud-native Computing, Hanser Verlag, 2nd Edition, 2024
Remarks	The lecture will take place online via screencasts on the above-mentioned topics. Face-to-face lectures might be given on selected topics. Small weekly Moodle tests will assess the content of the screencasts. These test results will be considered as a participation component in the grade of the project work.

Module Course: Web Services (Labs + Project)

(of Module: Web Services)

Course Type	Practical Training	**Form of Learning**	Presence
Mandatory Attendance	yes	**ECTS Credit Points**	4
Participation Limit	24	**Semester Hours per Week**	3
Group Size	12	**Workload (hours)**	120
Teaching Language	German/English	**Presence Hours**	45
Study Achievements ("Studienleistung", SL)		**Self-Study Hours**	75
SL Length (minutes)		**SL Grading System**	

The following section is filled only if there is a course-specific exam.

Exam Type		**Exam Language**	
Exam Length (minutes)		**Exam Grading System**	
Learning Outcomes			
Participation Prerequisites			

The previous section is filled only if there is a course-specific exam.

Contents	The content taught in the lecture is deepened and practiced in accompanying labs. During the lecture-free period, students demonstrate their understanding through a project-based transfer task. As part of this project, students may be required to deepen their understanding of additional content on their own responsibility and to learn about it independently. The project work can be organized as individual or group work.
Literature	See lecture
Remarks	See lecture Participation during the lecture and the labs is included in the grade for the project work.