00 – Organization

Web Technology Project (International Computer Science)
Summer semester 2025
Prof. Dr. Felix Schwägerl

Personal introduction

Prof. Dr. Felix Schwägerl

Professor of Global Software Engineering
felix.schwaegerl@oth-regensburg.de
Office K232 (Galgenbergstr. 32, Building K)
Office hours: Mon, 14:30-15:30 (registration required)

since 2023	Professor of Global Software Engineering at OTH Regensburg
2018 – 2023	Senior Software Engineer and Team Manager at MID GmbH in Nürnberg
2012 – 2018	Scientific assistant at University of Bayreuth (chair of Software Engineering)
2007 – 2012	Bachelor and Master studies in Computer Science

Prerequisites

- 2nd study stage (30 ECTS from 1st and 2nd semester, check HISinONE)
- Programming skills on PG2 level, preferably in Java
- Recommended: (Global) Software Engineering, Databases, Computer Networks
- Interest in the technical foundations of web-based applications
- Willingness to work on a personal project in the 2nd half of the semester

Goals

- You understand how the internet works.
- You obtain an overview of the web standards HTTP, HTML, CSS, and JavaScript.
- You get familiar with technical frameworks for backend development (here: Spring Boot) and frontend development (here: React)
- You demonstrate the application of the acquired knowledge by the implementation of a full-stack web application as a personal project

What is Web Technology?

Standards

Frameworks

Facts

- Nowadays, most software is being developed using web technology.
- Web development is a rapidly evolving field around a few basic standards and protocols.
- To become a good web developer, you should learn the underlying standards and one representative of each type of framework (database, backend, frontend, style, deployment).

Phase 1: Seminar and exercises

- In the *seminar*, we discuss the foundations of web development.
- Practical *exercises* are offered to deepen the understanding of the material discussed in the seminar.
  - "Voluntary" (= highly recommended) submission
  - Important preparation for phase 2: You will practically apply all the basics required for starting off with your personal project from day one.

Phase 2: Project

- Individual project (no teams), common specification
  - Everyone works individually on the same topic based on the same project specification provided during the project kick-off session
  - The minimum requirements for passing the project with success as well as the evaluation criteria will be communicated in the kick-off session, too.
- The project makes up 100% of the final grade of the module WTP.
  - There is no exam!

Detailed Events

Seminar (4 SWS 50% of semester)

- Fri, 10:00-13:15, K007
- 21 March until 9 May
- Voluntary presence

Exercises (2 SWS 50% of semester)

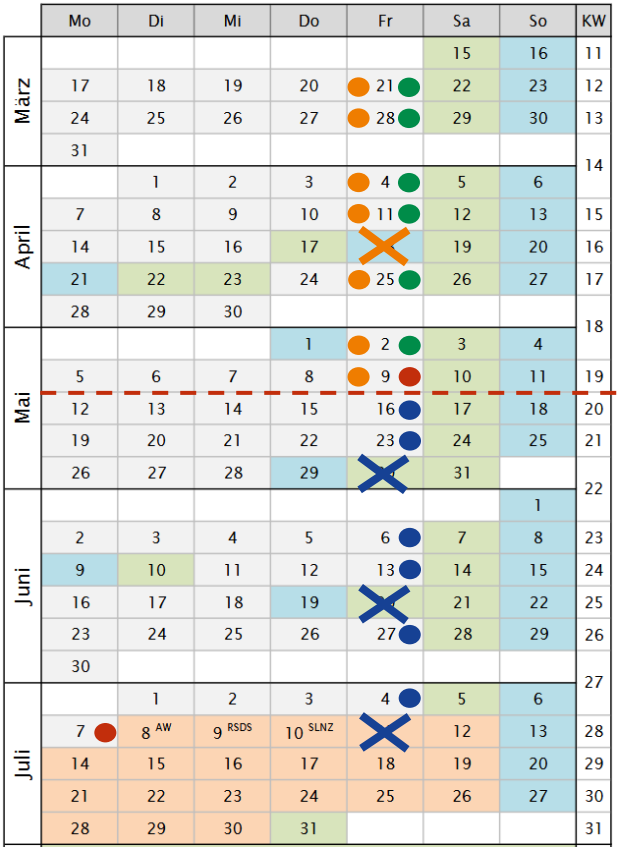
- Fri, 13:45-15:15, K007
- 21 March until 2 May
- Voluntary presence

Project Kick-off and Deadline

- Kick-off: Fri 9 May, 13:45-15:15, K007
  - Mandatory presence
- Project deadline: 7 July
  - No final meeting / presentation

Progress Meetings

- Individual 5-minute time slot every week
- Fri, 10:00 – 15:15, K007
- 16 May until 4 July
- Success requirement: Attend 4 meetings



Requirement for passing the module: Successful project

Requirements for passing the module

- Attend at least 4 out of 6 offered *progress meetings*
- Your submission must be available on OTH GitLab (as required by specification)
- The *minimum requirements* of the *specification* must be fulfilled:
  - Functional requirements (use cases offered by your application)
  - Minimum code quality / test coverage (as required by specification)
  - Project must be buildable and executable in the standard environment (Java 23, Node.js v22, SQL database)
- The *Git version history* must indicate that you were constantly working on the project, gathering and incorporating *feedback* from the project meetings.

Evaluation criteria impacting the final grade

- (more details will be provided with project specification)
- Usability of the user interface
- Design of the API
- Code quality / style
- Additional features implemented (beyond the scope of specification)
- Bugs

Project registration and submission deadline

- Registration deadline via HISinOne: 22 until 29 April 2025
  - Deregistration period: 3 until 12 June 2025 (after project kick-off)
  - No registration / deregistration is allowed after the deadlines.
  - If you are registered but don't submit a project, a failed attempt will be registered.
- Submission deadline: Monday, 7 July 2025, 11:59 pm
  - The project repositories will be locked after this deadline.

Zero tolerance for plagiarism

- Failing grade 5 is registered immediately if you
  - Copy a solution, or parts thereof, from another student
  - Make your Git repository available to other students
  - Copy large parts of source code from the internet (which includes generative AI tools such as ChatGPT)
- If you are repeatedly not able to explain your own source code during the project meetings, I must assume plagiarism.

Failing and repeating the project

- If you fail the project, you have to take the next possible repeat opportunity.
  - Summer semester 2026 (different project specification!)

General Progress Meeting rules

- K007 is available for individual project work from Fri, 10:00 – 15:15 (2nd half of semester)
- According to the handbook, you *should* be present for ~2 SWS per week, but I don't track.
- I will assign individual 5-minutes time slots within the official period (Fri, 10:00 – 15:15)
  - Presence is *mandatory* during your personal time slot.
  - You may skip 2 out of 6 meetings, e.g., in the case of illness.
  - In exceptional cases, the progress meeting can be carried out via Zoom during my office hours (Monday afternoon). Registration is strictly required.

Use your time slot to report progress and gather feedback.

- Your report should be structured like a Scrum daily:
  - What did you achieve in the past 7 days?
  - What are you planning to achieve in the next 7 days?
  - Where are you stuck?
- You must have your source code available in the IDE (CIP PCs or own laptop).
- Your current progress must be committed and pushed to your Git project repository
  - Use branch main for stable, additional branches for unstable versions.
- For conceptual discussions, please prepare adequate sketches.
  (I do not expect lengthy PowerPoint presentations!)
- The reports may impact your final grade (positively or negatively).

WTP workload calculation

- Module credits: 5 ECTS = 5 × 30h = **150h**
  - 1 *SWS* = 15h (assuming a 15-week semester)
- *Presence* workload: **60h**
  - Seminar: 4 *SWS* × 50% = **30h**
  - Exercise: 2 *SWS* × 50% = **15h**
  - Presence-based project work: 2 *SWS* × 50% = **15h**
- *Self-study* workload: **90h**
  - Solve exercise sheets: 4 *SWS* × 50% = **30h**
  - Additional project work: 8 *SWS* × 50% = **60h**

➢ In the second phase of the semester, you are expected to work on the project for about *10 hours per week for 7.5 weeks*.
  - Official distribution: 2 hours presence, 8 hours self-study
  - Presence minimization: If you only attend your own project meeting, you are supposed to spend 10 hours with self-study.
  - Presence maximization: If you spend the full 6 hours in the classroom, you have 4 hours of “homework” left.

- Your actual workload is evaluated at the end of the semester. Please make honest indications.

Web Technology Project (ICS summer 2025)

- https://elearning.oth-regensburg.de/course/view.php?id=6617
- Please register ASAP!

Provided material

- Important announcements (make sure you receive notification mail)
- Forum (use it for asking questions!)
- Lecture slides
- Exercise sheets (>1 week before deadline)
- Project specification (uploaded on Fri, 9 May during the project kick-off)
- Personal time slot assignment for progress meetings

Literature and on-line documentation

I don't expect you to read additional literature for WTP.

The seminar partly builds on:

- Juha Hinkula: Full Stack Development with Spring Boot 3 and React, Packt, 2022
- Mike Amundsen: Design and Build Great Web APIs, The Pragmatic Programmers, 2020
- Ben Frain: Responsive web design with HTML5 and CSS, 3rd edition, Packt, 2020

Recommended on-line documentation:

- Mozilla Developer Network (including HTTP, HTML, CSS, JavaScript): https://developer.mozilla.org/en-US/docs/Web
- OpenAPI documentation: https://learn.openapis.org/
- Spring Boot documentation: https://docs.spring.io/spring-boot/index.html
- React documentation: https://react.dev/
- PicoCSS documentation: https://picocss.com/docs
- Thymeleaf documentation: https://www.thymeleaf.org/documentation.html
- Docker documentation: https://docs.docker.com/

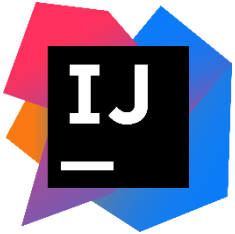
Not recommended:

- General search engines like Google: The result will mostly return outdated StackOverflow pages not considering the current best practices.
- Usage of LLMs like ChatGPT without reflecting the result. The same outdated StackOverflow pages have been used as training data.

This slide deck re-uses material from previous courses:

- Prof. Dr. Markus Heckner: Webtechnologien (OTH Regensburg)
- Victoria Kirst: Web Programming Fundamentals (Stanford University)

Integrated Development Environment (IDE): IntelliJ

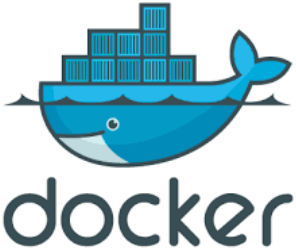


- Primarily Java IDE, but also includes plug-ins for frontend development
- Pre-installed on Windows and Linux in all PC labs of the P and K buildings
  - Use **U:** drive for persistent personal storage!
- Download for your own PC: https://www.jetbrains.com/idea/

Node.js / Node Package Manager (npm)

- Build system and runtime environment for frontend
- Pre-installed on Windows and Linux in all computer labs
- https://nodejs.org/en/download/

Docker



- Container management platform simplifying the test-deployment of components (e.g., database)
- Pre-installed on Windows and Linux in all computer labs
- https://www.docker.com/

MariaDB

- Open-source relational SQL database
- Not pre-installed in PC labs, but can be executed started via docker (see chapter 02)
- https://mariadb.org/
- If you prefer a different database (e.g., PostgreSQL), you can use it with only slight differences in configuration.

Version Control System (VCS): Git

- Used for submitting exercises and your project.
- Pre-installed on Windows and Linux in all computer labs
- https://git-scm.com/

Source code hosting: OTH GitLab

- https://gitlab.oth-regensburg.de/scf38786/ics-wtp-seminar
- You receive access by solving exercise 1

<12 inch laptops or tablets are entertainment devices!

- You should work with an *ergonomic* set-up you are comfortable to work with on programming exercises for several hours per week.
  - Sufficiently large screen, suitable keyboard, external mouse
- Recommendation: The **CIP Pools** in P and K buildings are available for you when there are no lessons. Check WebUntis for free capacities!
  - In the case of problems with the installed software, please contact me immediately.

Seminar chapters (tentative)

01 – HTTP and REST
02 – Backend Development
03 – HTML
04 – CSS
05 – JavaScript
06 – Frontend Development
07 – Containerization of Web Applications