Back-end Development 1

Vocational Education
Professional Degree

Level 5.1 in the Norwegian Qualification Framework

Version: 1.0
Date: 10.02.21

Table of Contents

Introduction

Back-end Developers serve a vital role in any organisation concerned with offering services via the web. Back-end Developers are integral to the creation and maintenance of all manner of server-based software solutions. These services may be required to cooperate with Front-end solutions or independently, this necessitates that a competent Back-end Developer is highly skilled in developing complete robust server solutions and able to integrate with modern Front-End Technologies. Candidates for this program will learn technical programming skills which are pivotal to the profession, they will also learn supplementary skills which will enable them to work both independently and as a productive member of a software development team.

Target Group

The programme is aimed towards people who want to create and deploy database driven web solutions by starting from the foundations of a web server and build up to basic user interface functionality. The programme is open to candidates with no prior programming experience as all relevant skills are part of the described curriculum, but candidates should have a strong interesting in technical, analytical, or mathematic subjects. No formal mathematics is required beyond high school level.

Educational Profile

The programme focuses on practical skills and applicable solutions to the field of web server development. All aspects of development of web solutions are considered and solutions to deployment of webservices using industry accepted technologies are presented. The programme incorporates numerous practical based assignments which ensure candidates are provided with a means to practise and apply the relevant theory presented within the programme. The candidates are able to grow their competence through professional workflows and software methodologies which are commonly used by professional back-end developers. The academic structure of the programme consists of selected theory supported by practical work with daily exercises, larger compulsory assignments and major projects. Several subjects and assignments can be facilitated for collaboration with fellow students across disciplines at Noroff.

Teaching and Exam Language

Teaching, curriculum and syllabus is in English and Norwegian. The examination language is in English.

Candidates learning resources and literature will mainly be in English. Most learning resources and terminology come from English-language sources, such as manuals for equipment and software, lecture videos and articles from the internet.

Career Possibilities

Candidates will be able to work in both national and international companies in need of Back-end developers. Back-end developers form a crucial role in business technology solutions and there are numerous job listings relevant to skilled back-end developers. Candidates will be able to pursue positions as both in-house developers for business aiming to developer their own solutions or candidates may work for specialized software development firms which cater for the bespoke needs for their clients. The software developer realm is a fast paced ever changing field, but the core tenets of the role of Back-end developer will persist throughout these changes.

The table below shows examples of opportunities for students graduating from Back-end Development

Profession	Competence	Possible duties at work
Back-End JavaScript Developer	- Experience with one or more development programming languages - Understanding of development lifecycle - Familiarity with front-end languages (e.g. HTML, JavaScript and CSS) - Analytical, problem solving, and time management skills - Teamwork and collaboration skills	- Participate in the entire application lifecycle, focusing on coding and debugging - Write code to develop functional server solutions - Troubleshoot and debug existing solutions - Review unit tests and peers code submissions - Collaborate with Front-end developers - Gather and address technical and design requirements - Build reusable code and libraries

Admission Requirements

Candidates can be accepted into this programme either by general admissions or recognition of prior learning or work experience. The level of qualification for admission is set by the National Qualifications Framework (NQF) and extensive information on the admission requirements for the school is in our Regulations available at www.noroff.no.

Formal Admission Requirements

For general admission to this programme it is required to document the following criteria:

- General University Admissions Certification

Admission by Prior Experience in the Field

Prior experience in the field means knowledge, skills and general competence that a person has acquired, either through

- formal learning, such as foreign education, higher education, folk high school or adult education
- non-formal learning, such as courses or other structured training that is not part of the education system
- informal learning, such as care work, organizational experience, leisure activities, volunteer work or work experience

Applicants wishing to be recognized for prior learning or work experience must be at least 23 years of age. For further information, please see the school's regulations (§2.3.).

Overall Objectives

The NQF levels are formulated on the basis of what a person knows, can do and is capable of doing as a result of a learning process. The outcomes of the completed learning process are described in the categories "knowledge", "skills" and "general competences".
The categories describing learning outcomes include:

Knowledge: Understanding of theories, facts, principles, procedures in the discipline, subject area and/or occupations.
Skills: Ability to utilize knowledge to solve problems or tasks (cognitive, practical, creative and communication skills).
General Competence: Ability to utilize knowledge and skills in an independent manner in different situations.

Learning Outcome Descriptions – Back-End Development

After graduation the candidates possess the following learning outcomes:

Knowledge
The Candidate...

- has knowledge of primary concepts, core ideas and general methods that are used in programming
- has knowledge of the modules, packages, and frameworks that are used within the JavaScript ecosystem to build server-based solutions
- has knowledge of development and debugging processes and tools that are used in developer enabled web browsers and extensions
- has knowledge of processes, principles and tools that are used to build and maintain databases
- has knowledge of REST APIs and solutions that are used to create fast accessible data on the web

- has knowledge of software development lifecycle stages, frontend technologies management processes, and tools that are used in the field of web and back-end development
- has insights into relevant GDPR standards, regulations and quality requirements for back-end development
- has knowledge of the software development industry and is familiar with the associated work and developer roles
- can update his/her knowledge of the JavaScript programming language and in software development
- understands the importance of back-end development in a societal and value-creation perspective

Skills
The Candidate...

- can apply knowledge of programming principles to efficiently develop web and back-end solutions
- can apply knowledge of JavaScript to develop and maintain back-end and software solutions
- masters relevant tools, materials and techniques using the JavaScript ecosystem to build server-based solutions
- masters relevant tools and techniques to write, edit and debug JavaScript programs
- can apply knowledge of databases to enhance the functionality and performance of data-centric solutions
- masters relevant tools and techniques to create and enhance the functionality of static web pages
- can find information and material that is relevant to a problem through purpose-built JavaScript software
- can find information and material about GDPR regulations that are relevant to a project
- can study a project and identify the software requirement needs of a JavaScript solution, and what measures need to be implemented

General Competence
The Candidate...

- understands the ethical principles that apply to development and maintenance of web solutions intended for public use
- understands the ethical principles that apply to GDPR compliance in software development
- has developed an ethical attitude in relation to back-end and software development
- can carry out programming, development, and maintenance tasks on JavaScript based solutions
- can build relations with his/her peers, also across disciplines and with external software developers
- can develop projects and solutions of relevance to the software development industry

Organisation and Structure

Available as a Campus Programme	Yes
Available as an Online Programme	Yes
Duration Full-time	2 semesters
Duration Part-time	4 semesters
Number of Credits	60

Course Structure

The table below shows how the courses are distributed across semesters.

C = Number of Credits for the course.
1-2 = How the points are allocated across semesters.

Course Code	Name	C	1	2
FI1BDPF05	Programming Foundations	5	X	
FI1BDFT10	Front-End Technologies	10	X	
FI1BDPO05	Programming with Objects	5	X	
FI1BDPM25	Project Methodology	2.5	X	
FI1BDP175	Semester Project	7.5	X	
FI1BDJS05	JavaScript Servers	5		X
FI1BDDB05	Databases	5		X
FI1BDRA75	REST APIs	7.5		X
FI1BDSD05	Server Deployment	5		X
FI1BDP275	Exam Project	7.5		X
	Total credits	**60**	**30**	**30**

Academic Progression Table

Semester 1	**Academic Weeks**				
Course Name	1-3	4-8	9-11	12-13	14-17
Programming Foundations	126				
Front-End Technologies		210			
Programming with Objects			126		
Project Methodology				84	
Semester Project					168
Semester 2	**Academic Weeks**				
Course Name	18-21	22-25	26-31	32-35	36-39
JavaScript Servers	168				
Databases		168			
REST APIs			252		
Server Deployment				168	
Exam Project					168

The academic weeks show the duration and progression of each course. The number in each cell indicate hours per course. Minor changes in the overview can occur.

Academic Content

Semester 1

Programming Foundations

The course provides knowledge of and skills with programming and the general core ideas, concepts and principles within the field. JavaScript is used to explore basic programming principles by building small and simple programs. Through practical tasks, candidates learn about syntax, program structure, control structures, data types and variables. A portion of this course is used dedicated to a small, guided project intended to let candidate s explore the new concepts they have recently learnt.

Front-End Technologies

The course provides the candidate with understanding and practical knowledge in coding with HTML and CSS, and an introduction to using JavaScript as a Front-end language. The candidates will be introduced to the concepts of user interaction and interface design. While covering the fundamentals of the JavaScript language, the candidates will learn the basics of DOM manipulation and how to interact with REST APIs to retrieve data. This Front-end course will draw distinctions between the roles of Front-end developers and Back-end developers, ensuring that candidates are able to distinguish the roles easily. They will also learn how the two roles cooperate within the realm of web development. The course will also cover the usage of Git as a version control tool for coding.

The course builds on competence from *Programming Foundations.*

Programming with Objects

The course will cover more advanced topics within the JavaScript language (object orientation). These advanced topics will delve further into the JavaScript language specifications and will include topics such as *closures* and *higher order functions*. The candidates will be taught all aspects of working with JavaScript Object Notation (JSON), and how data can be sent, receive, stored, and updated from existing REST APIs using JSON.

The course builds on competence from *Programming Foundations.*

Project Methodology

The course provides knowledge of concepts, processes and tools that are used in planning and executing a software project. The course will extensively cover project planning and documentation, with emphasis given to the roles within a software team, and the delegation of responsibilities to those roles. Candidates will be taught the principles of Kanban for project management. The course will expand on the usage of Git as a means to collaborate on a project through the usage of feature branches. The candidates will be taught the purposes and usage of continual integration and continual delivery (CI/CD) in their projects.

Semester Project

The semester concludes with a graded project where the candidate must demonstrate practical skills and competence from courses in the first semester. Candidates work independently or in a group on a project, which must be planned, documented and executed according to project criteria. The aim is to carry out a practical project consisting of elements from the previous

courses to develop a web-based solution to a provided business case using a provided API. The candidate also prepares a project plan and an individual reflection report documenting the process and choices made along the way.

The course builds on competence from *Programming Foundations, Front-End Technologies, Programming with Objects* and *Project Methodology*.

Semester 2

JavaScript Servers

The course provides knowledge of concepts, processes and tools of conventions and solutions for setting up a development environment for building server-based JavaScript solutions and deploying those servers on a local computer. Candidates will learn to use the Node runtime and the associated Node Package Manager (NPM). Through NPM they will explore the JavaScript ecosystem of tools and package management.

The focal point of these packages will be the use of external packages to build JavaScript servers. Candidates will be taught how to create, launch and maintain these servers locally on their own machines. These server solutions will encompass the concepts of routing, path variables, parameters and simple endpoints.

The course builds on competence from *Programming with Objects.*

Databases

The course provides knowledge of techniques, principles and tools used build, maintain and utilise a relational database. Candidates will learn the basic operations needed to interact with a relational database (create, read, update, and delete). They will also be taught how to use normalization to improve their database designs. The candidates will be taught he principles and skills needed to include these databases in their server-based solutions.

The course builds on competence from *Programming Foundations*.

REST APIs

The course provides knowledge of concepts, processes and tools that are used to build and test Representational State Transfer (REST) APIs. REST APIs are fast, reliable and expandable solutions to creating accessible data on the web. This course will greatly expand on the use of the JavaScript server solutions and will require candidates to build software that interacts with the databases solutions such as those that they created previously with SQL. Candidates will learn the skill necessary to use the Postman software (and opensource alternatives) to test and document their created APIs.

Candidates will be tasked a project requiring that they create a working REST API using a database built to meet provided specification.

The course builds on competence from *JavaScript Servers* and *Databases.*

Server Deployment

The course provides knowledge of concepts, processes and tools that relate to deploying their server software to hosted servers. Candidates will be able to authenticate users to their solutions and store session information on the server for returning users. The course will familiarise the candidates with the implications of GDPR on their solutions.

Candidates will learn the concepts, processes and tools necessary to ensure that their software meets customer specifications by using common JavaScript testing frameworks, and the principle of Test-Driven Development (TDD).

Candidates will learn the concepts, processes and tools necessary to deploy their solutions to simple container services (Heroku) and other basic web hosts that support Node server solutions.

The course builds on competence from *JavaScript Servers* and *Databases.*

Exam Project

This is a major project that reflects competence the candidates have acquired during the academic year.

The candidate must solve the assignment independently, or as part of a group, from a given practical problem. Candidates should be encouraged to form small groups based on the tasks/challenges of the given problem. Internship projects are encouraged and the project challenges candidates to find a real-world project to acquire practical experience in a professional setting. The candidate is responsible for all aspects of the project, in accordance with the supervisor through the internship, if applicable. If the candidate is not working on a real-world project, an alternative case project will be presented by the academic staff. The completed project will be presented to the teacher, sensor, fellow students and if applicable, the customer.
If the candidate is working on a real-world project, the candidate is required to provide documentation for the completed project even if it is not within the customer specifications.

The course builds on competence from all courses during the academic year.

Timetable for the Academic Year

The table below shows the total workload for candidates divided into course duration indicated by total number of weeks, hours per week (H/W), Teaching (T), Teacher-Supported Work (TSW) and Self-Study (SS).

Academic Year						
Course	**Weeks**	**H/W**	**T**	**TSW**	**SS**	**Total**
Programming Foundations	3	42	30	30	66	126
Front-End Technologies	5	42	50	50	110	210
Programming with Objects	3	42	30	30	66	126
Project Methodology	2	42	20	20	44	84
Semester Project	4	42	1	12	155	168
JavaScript Servers	4	42	40	40	88	168
Databases	4	42	40	40	88	168
REST APIs	6	42	60	60	132	252
Server Deployment	4	42	40	40	88	168
Exam Project	4	42	1	12	155	168
Total	**39**	**420**	**312**	**334**	**992**	**1638**

Teaching, Learning and Work Methods

The foundation of Noroff's pedagogic concept is that the candidate learns best by actively participating in industry-leading learning activities. Therefore, teaching time applies practical learning methods to accumulate knowledge, skills and general competence. The teacher acts as a proactive supervisor and instructor. Close monitoring of individual learning needs is crucial for the candidates' achievements. Our teaching methodology is founded on a problem-based learning model, which ensures that the candidates have independently acquired theoretical and technical knowledge through available online lessons, syllabus and course materials prior to lectures and teaching hours. The instruction is tailored to the needs of the candidate and it is important that he/she follows the course material to be well prepared for the scheduled lessons. Through observation, testing and industry-relevant tasks, course evaluations and guidance, teachers review and adapt their teaching method to the greatest extent possible. Teachers adjust the course materials and remain differentiated based on the candidates' goal achievement.

A regular teaching day often consists of two separate sessions:

1. Teaching and Teacher-Supported Work

Teaching consists of various learning activities where the teacher lecture, guides and instructs. The methods used are mainly exercises, lectures, demonstrations and student presentations. Skills are developed through course specific exercises that run over one or more days. The teachers serve as active and present mentors. Members of the academic staff from other departments are also available for questions and professional follow-up through the school's LMS. The raw material used for assignments are made available on the LMS for all students regardless of the campus affiliation or if they are online students.

We aim to take advantage of the expertise in the professional community, at a national level. Lectures and demonstrations are conducted locally on each campus or as streamed video lessons transferred from other departments in Noroff. All students have access to the online lessons through our LMS. Regardless of the method, the campus students will have a teacher physically available for guidance, supervision and professional follow-up. They will also always have access to a contact teacher within the field of study throughout the programme and across courses. In this way we always offer adequate follow-up and provide up to date industry-relevant instruction adapted to digital developments and new educational technology.

This first session is often summarized from day to day with student presentations or concluded by the teacher in the form of simple presentations based on academic needs, either in separate groups or in plenary. The methods used in this session are adapted as far as possible to student groups and individual student learning achievement. Flexible teaching methods are therefore an important criterion for the success of customized instruction.

2. Self-study

The time set for self-study is used for assignments and projects related to the described learning outcome for the course. The teacher is available for questions and follow-up as needed. The assignments are usually completed over extended periods and students apply their knowledge and skills to acquire technical, theoretical or creative skills, individually or in groups. Compulsory assignments are continuous throughout the academic year and are often carried out in parallel with the learning activities from day to day. After submission, assignments are assessed and graded in the school's LMS based on the assessment criteria in the assignment description and the learning outcomes of the course.

Noroff's teaching model sets the student's responsibility for self-learning in focus and this requires a clarification of expectations and motivation for the students from the first day of school. In today's working life graduates must solve various problems by using all available aids. The students independent use of resources in the form of assignments descriptions, video explanations, online resources and similar will, therefore, be crucial for their learning outcomes and achievements throughout the year.
The students' presentation skills are also an important part of the education and several student presentations are conducted throughout the academic years.

Through fixed daily tasks, assignments, projects, course assessments and tests the school's academic personnel can quickly determine whether students need follow-up or guidance. The compulsory assignments are normally delivered by the end of the week in the final week of the course. During the following week, the students receive feedback from teachers and fellow students on completed assignments and will be given a pass or fail based on the assessment criteria in the assignment description. Students who fail to pass or don't deliver are immediately followed up and given the ability to re-deliver according to the examination regulations. All assignments and projects must be submitted and passed to obtain a vocational diploma.

Through written reports delivered on Noroff's online learning management system we can further control the variation in learning outcomes between the different students.
The system also allows teachers to easily have an overview of average grades, attendance and submissions. Discrepancies are reported and followed up by the teacher and course leader.

Online Delivery

This study programme is offered as an online study, in addition to campus. Noroff enrols four times a year for online studies, the start-ups are in January, March, August and October. Online students can access lectures, material, resources, activities and demonstrations through Noroff's LMS. This enables students to study at their own leisure. Online students perform similar learning activities as campus students. Just like campus students, online students can work with activities and assignments individually or in groups, depending on the learning activity that is carried out. There will be additional guidance for collaboration groups, where needed.
Resources like literature, lessons, audio-visual material, demonstrations and more is available in the LMS. Online students can also participate in streamed lectures and webinars, when given and where it fits into the progression plan. Streamed lectures are recorded and distributed in the LMS shortly after the lecture has ended.

The school's LMS provides supervision and follow-up of grades, submissions, chat, forums, activity reports and student activity logs through embedded systems. The system also tracks activity in forums (e.g. posts made, and threads viewed) and students' access to learning resources (e.g. activity logs to see if a lesson has been read or just browsed). It is also possible to check for significant differences in academic needs in relations to enrolment groups and whether an enrolment groups size is of a decisive factor. This provides an overview of individual student activity and is a proprietary tool for tracking and follow-up on individual students, enrolment groups and collaborative groups. Enrolment groups are online students starting together at a specific date, either as full-time or part-time students. A collaborative group is a selection of students within an enrolment group or a selection of students across enrolment start-ups.

The academic staff is available through e-mail and interactive tools in the LMS, such as forums and instant messaging at given intervals.

There are no mandatory gatherings on this programme. Online students have the opportunity to attend campus gatherings and events. Costs related to travel and accommodation are covered by the candidate.

Individual Student Follow-up

Students are followed up both on their participation in the LMS, and through their work requirements. By evaluating a student's participation, the teacher can assess individual academic needs.

Regular submissions will quickly reveal if a student is struggling. Compulsory assignments must be submitted and verified. If a student has not delivered at a given time; they will be contacted. After submission students receives feedback on the work that has been delivered.

Groups and Collaboration

In addition to the follow-up capabilities mentioned above, group activity and collaboration areas is another function in the LMS. Collaborative work can be group assignments and projects where students use dedicated forums, chats or wikis. This allows each collaborative group to have their own work area for different assignments similar to campus students. The groups can be supervised, mentored and participated in by the academic staff.

Groups can collaborate and work across enrolments, as all students have access to the same communication tools regardless of students' individual progression. This in contrast to campus-based studies, where learning happens in an auditorium or classroom and everyone must be on the same course according to the progression of the study programme. This enables new students to collaborate and learn from the more experienced students, by mentoring or supporting on tasks and learning activities that they have mastered.

Online Support Services

- Academic staff is available on forums and chat, at predefined times
- Administrative support is available on e-mail and phone, at predefined times
- Learning Management System
- Online content from external providers
- Webinars, if applicable
- Streamed lectures and demonstrations, where applicable
    - Recordings will be distributed in the LMS, if applicable

Examination and Assessment

The learning outcomes of students are assessed with a set of compulsory assignments and a practical letter graded exam project. The table below indicates how assessment is conducted for each course.

- Projects are assessed as an examination project.
- Module and Course Assignments are assessed as compulsory assignments.
    - Course Assignments are final assignments for each course, while Module Assignments are smaller assignments during a course.

Compulsory assignments are assessed as Passed or Failed, after which verbal or written feedback is provided from the examiner indicating the level and the submitted works strengths and weaknesses. The purpose of the assessment of assignments is to determine whether the student has achieved the learning outcomes. Feedback provides the student with means to strengthen their weaknesses leading up to the graded projects.

For certain courses, specified in the course descriptions, the work requires students to assess their fellow students work along the way during the assignment period. This is the case for both online and campus students. The course work assessments are carried out in writing or verbally over time and are compulsory. The minimum requirement for passing an assignment is assessed based on the learning outcomes for each course. If the learning outcomes are not achieved the assignment is failed. Assessment is made individually for each course. For examination projects a summative assessment is conducted and given a letter grade ranging from A-F, where E is the minimum passing grade. External examiners from the profession will participate in these assessments.

All assessment of compulsory assignments and the achieved grade on projects are included on grade transcripts and the diploma. Student's must pass all compulsory assignments and achieve the passing grade on the projects to obtain the diploma.

Assessment Table

CA = Compulsory Assignment (assessed with Passed / Failed)
EP = Examination Project (assessed with a letter grade ranging from A-F)

Semester 1	CA	EP	Semester 2	CA	EP
Programming Foundations	X		JavaScript Servers	X	
Front-End Technologies	X		Databases	X	
Programming with Objects	X		REST APIs	X	
Project Methodology	X		Server Deployment	X	
Semester Project		X	Exam Project		X

A	Outstanding	Outstanding achievement that clearly stands out. Demonstrates a great degree of autonomy.
B	Very good	Very good performance that is above average. Demonstrates independent capabilities.
C	Good	Average performance that is satisfactory in most areas.
D	Fairly good	Performance below average, with some significant deficiencies.
E	Sufficient	Performance that meets the minimum requirements, but no more.
F	Fail	Performance that does not meet the minimum requirements.

The table shows the grade scale and description of each letter grade.

Equipment Requirements

Listed below are the equipment requirements for this programme. The responsible party for the procurement of the equipment and software is also detailed in the table. It is also stated in which semester the equipment is required.

Online students must initially procure and maintain their own equipment and software, as indicated in the table. The student fees for online studies are significantly lower and this therefore compensates for these expenses.

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Reading List

In a discipline in continuous growth and change, the reading list must be current. Final reading lists are made available to students in advance of each academic year. The following list and those used in the course descriptions are therefore to be considered as guidelines for the proposed reading list.

Required Reading

None assigned.

Recommended Reading

Eloquent JavaScript: A Modern Introduction to Programming, *Marijn Haverbeke*, No Starch; 3rd edition (2018), ISBN: 9781593279509

Clean Code : A Handbook of Agile Software Craftsmanship, *Robert C. Martin*, Prentice Hall; (2009), ISBN: 9780132350884

The Pragmatic Programmer, 20th Anniversary Edition: your journey to mastery , *David Thomas, & Andrew Hunt*, Addison-Wesley Professional; (2019), ISBN: 9780135957059

Refactoring: Improving the Design of Existing Code, *Martin Fowler*, Addison-Wesley Professional; 2nd edition (2019), ISBN: 9780134757599

Course Description: Programming Foundations

Course Code	FI1BDPF05
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Programming Foundations
Norwegian	Grunnleggende programmering
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	5
Length in Weeks	3
Teaching Hours	30
Teacher Supported Hours	30
Self-study Hours	66
Total Hours	**126**

The course can be taken as a single course	Yes

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	No prerequisite required.
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course provides knowledge of and skills with programming and the general core ideas, concepts and principles within the field. JavaScript is used to explore basic programming principles by building small and simple programs. Through practical tasks, candidates learn about syntax, program structure, control structures, data types and variables. A portion of this course is used dedicated to a small, guided project intended to let candidate s explore the new concepts they have recently learnt.

Course Relevance

Knowledge and skills acquired during this course will be extended when the student specializes in programming with JavaScript or learns new programming languages. The student gains a broader perspective of their own field, which provides a basis for interdisciplinary collaboration with other areas of the discipline and backend developers.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of primary concepts, core ideas and general basic methods within programming
- has knowledge of program syntax, program structure, control structures, data types, and variables as used in the JavaScript language
- has knowledge of development and debugging within a browser
- has knowledge of industry relevant software used for writing JavaScript code
- can update his/her knowledge of basic programming concepts

Skills

The Candidate…

- can apply knowledge of program syntax, program structure, control structures, data types, and variables to complete simple tasks using the JavaScript language
- can apply knowledge of browsers and development tools to set up a computer for use in developing and debugging small JavaScript programs
- masters the use of relevant tools for writing, editing and debugging JavaScript source code

General Competence

The Candidate…

- can carry out simple programming tasks with the JavaScript language

Learning Methods
Teaching varies between online based lessons, lectures, software training, activities and guidance.

Examination and Assessment
The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination
It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements
Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources
Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Course Description: Front-End Technologies

Course Code	FI1BDFT10
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Front-End Technologies
Norwegian	Front-end teknologi
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	10
Length in Weeks	5
Teaching Hours	50
Teacher Supported Hours	50
Self-study Hours	110
Total Hours	**210**

The course can be taken as a single course	Yes, if the required prerequisite knowledge is covered.

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	Programming Foundations
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course provides the candidate with understanding and practical knowledge in coding with HTML and CSS, and an introduction to using JavaScript as a Front-end language. The candidates will be introduced to the concepts of user interaction and interface design. While covering the fundamentals of the JavaScript language, the candidates will learn the basics of DOM manipulation and how to interact with REST APIs to retrieve data. This Front-end course will draw distinctions between the roles of Front-end developers and Back-end developers, ensuring that candidates are able to distinguish the roles easily. They will also learn how the two roles cooperate within the realm of web development. The course will also cover the usage of Git as a version control tool for coding.

The course builds on competence from *Programming Foundations.*

Course Relevance

Through this course, the candidates will learn to construct simple static web pages; that is web pages consisting of only HTML, CSS, and JavaScript. They will learn about the DOM and how it is reflected int eh browser rendering of a web page. They will then use JavaScript to manipulate the DOM and this create a website that "reacts" to a user. They will use JavaScript to retrieve data from a provided REST service and once again manipulate the DOM based on the retrieved data.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of the concepts, processes, and tools that are used to build static web pages
- has knowledge of concepts, processes and tools that are used in development in the JavaScript language
- has knowledge of concepts of the Document Object Model (DOM)
- has knowledge of the concepts, processes, and tools that are used to interact with RESTful services
- has knowledge of concepts, processes, and tools of git version control software
- has insight into the HTML/CSS and ECMAScript specifications
- can update his/her vocational knowledge in HTML/CSS
- can update his/her vocational knowledge in JavaScript language

Skills

The Candidate…

- can apply vocational knowledge of JavaScript to manipulate the Document Object Model (DOM) and to retrieve and display data from existing REST services in static web pages
- masters relevant vocational tools, materials, and techniques to create static web pages using HTML and CSS
- masters JavaScript tools, materials, techniques, and styles used to enhance functionality in static web pages both as inline scripting and as external script files
- masters Git source control tools, materials, techniques, and styles that are used to create and work with a version-controlled project
- can find informational and material that is relevant to extending JavaScript functionality in static web pages through the use of third-party libraries

General Competence

The Candidate…

- understands the ethical principles that apply when creating JavaScript web solutions intended for public use
- can carry out work based on the needs of a Junior JavaScript developer

Learning Methods

Teaching varies between online based lessons, lectures, software training, activities, guidance and presentations.

Examination and Assessment

The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination

It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements

Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources

Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Recommended Reading

Eloquent JavaScript: A Modern Introduction to Programming, *Marijn Haverbeke*, No Starch; 3rd edition (2018), ISBN: 9781593279509

Course Description: Programming with Objects

Course Code	FI1BDPO05
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Programming with Objects
Norwegian	Objektorientert programmering
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	5
Length in Weeks	3
Teaching Hours	30
Teacher Supported Hours	30
Self-study Hours	66
Total Hours	**126**

The course can be taken as a single course	Yes, if the required prerequisite knowledge is covered.

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	Programming Foundations
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course will cover more advanced topics within the JavaScript language (object orientation). These advanced topics will delve further into the JavaScript language specifications and will include topics such as *closures* and *higher order functions*. The candidates will be taught all aspects of working with JavaScript Object Notation (JSON), and how data can be sent, receive, stored, and updated from existing REST APIs using JSON.

The course builds on competence from *Programming Foundations.*

Course Relevance

Working with data is the cornerstone of developing web services, and Programming with Objects is the means by which data is modelled and manipulated in a modern software solution. In the majority of cases data accessed from external sources will be delivered in the JSON format and developers must be familiar with how to convert retrieved data into usable information. This course will also ensure that candidates are familiar with advanced JavaScript topics which will enable them to produce more secure and robust solutions.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of concepts, processes and tools that are used in Object Orientated programming and JavaScript Object Notation
- has knowledge of concepts, and processes of functional programming as an alternative to Object Orientated programming
- has knowledge of concepts, processes and tools that are used with existing REST APIs
- can update his/her REST API and advanced JavaScript language knowledge
- has knowledge of industry and the JavaScript Developer role with the associated responsibilities

Skills

The Candidate…

- can apply vocational knowledge of Object Orientation concepts to practical and theoretical problems
- masters JavaScript Object Notation techniques, and styles
- masters advanced JavaScript tools, materials, techniques, and styles
- can find information and material that is relevant to a using existing REST APIs
- can find information and materials this is needed to building general purpose JavaScript solutions to relevant problems

General Competence

The Candidate…

- can carry out work based as a Junior JavaScript developer

Learning Methods
Teaching varies between online based lessons, lectures, software training, activities, guidance and presentations.

Examination and Assessment
The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination
It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements
Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources
Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Recommended Reading

Eloquent JavaScript: A Modern Introduction to Programming, *Marijn Haverbeke*, No Starch; 3rd edition (2018), ISBN: 9781593279509

Course Description: Project Methodology

Course Code	FI1BDPM25
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Project Methodology
Norwegian	Prosjektmetodikk
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	2,5
Length in Weeks	2
Teaching Hours	20
Teacher Supported Hours	20
Self-study Hours	44
Total Hours	**84**

The course can be taken as a single course	Yes, if the required prerequisite knowledge is covered.

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	Programming Foundations
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course provides knowledge of concepts, processes and tools that are used in planning and executing a software project. The course will extensively cover project planning and documentation, with emphasis given to the roles within a software team, and the delegation of responsibilities to those roles. Candidates will be taught the principles of Kanban for project management. The course will expand on the usage of Git as a means to collaborate on a project through the usage of feature branches. The candidates will be taught the purposes and usage of continual integration and continual delivery (CI/CD) in their projects.

The course builds on competence from *Programming Foundations.*

Course Relevance

Project planning is a vital part of the subject area. Time management, progress plans and knowledge of lean project methods are essential knowledge the candidates must possess in order to function as part of a larger project. With an understanding of version control systems in teams, the value and principles are supported by concepts such as "working code", "done" and "sprint", which are central in the context of agile development.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of the concepts, processes and typical stages of a software project lifecycle
- has knowledge of concepts and processes for version control with Git and CI/CD tools
- has knowledge of the principles of Kanban and Agile Development methods
- has knowledge of the industry and is familiar with the roles present in a software development team

Skills

The Candidate…

- can apply vocational knowledge of general-purpose planning techniques and tools to a software project
- can apply vocational knowledge of the principles of Kanban and Agile Development as a means to facilitate project management
- masters CI/CD principles and tools in a software project
- masters Git tools and process to all stages of the project lifecycle

General Competence

The Candidate…

- can carry out project planning based on the needs of a selected target groups or a given brief
- can develop project plans for a software development project

Learning Methods
Teaching varies between online based lessons, lectures, software training, activities, guidance and presentations.

Examination and Assessment
The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination
It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements
Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources
Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Recommended Reading

Eloquent JavaScript: A Modern Introduction to Programming, *Marijn Haverbeke*, No Starch; 3rd edition (2018), ISBN: 9781593279509

Course Description: Semester Project

Course Code	FI1BDP175
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Semester Project
Norwegian	Semester prosjekt
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	7,5
Length in Weeks	4
Teaching Hours	1
Teacher Supported Hours	12
Self-study Hours	155
Total Hours	**168**

The course can be taken as a single course	No

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	All previous courses
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The semester concludes with a graded project where the candidate must demonstrate practical skills and competence from courses in the first semester. Candidates work independently or in a group on a project, which must be planned, documented and executed according to project criteria. The aim is to carry out a practical project consisting of elements from the previous courses to develop a web-based solution to a provided business case using a provided API. The candidate also prepares a project plan and an individual reflection report documenting the process and choices made along the way.

The course builds on competence from *Programming Foundations, Front-End Technologies, Programming with Objects* and *Project Methodology*.

Course Relevance

The project challenges candidates to use and combine accumulated competence from the previous courses and showcase how they can complete larger projects, either individually or across disciplinary boundaries. Candidates are challenged to think and work holistically, which provides a solid platform for further learning and understanding of the developer field.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of the industry and is familiar with the workflow and methodologies that apply for the back-end development field
- can update his/her knowledge of the back-end development through the subjects and topics in the first semester
- understands the importance of back-end development in a societal and value-creation perspective

Skills

The Candidate…

- can apply vocational knowledge of general-purpose planning techniques and tools to a software project
- can apply vocational knowledge of Git tools and process to all stages of the project lifecycle
- masters the use of JavaScript code as means to build a complete solution for a specified task
- can find information and material that is relevant to accessing and display data from existing REST service
- can study and document his/her own work and identify programming related issues in a larger project, and correctly identify and design means to resolve these issues

General Competence

The Candidate…

- can carry out work based on the needs of a selected target groups

Learning Methods
Teaching consists of a single lecture to introduce the project, guidance, self-study, production and presentation.

Examination and Assessment
The learning outcomes are evaluated accordingly:
Examination projects are assessed with a letter grade from A-F.

Re-Examination
It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements
1. All projects must be approved by the teacher before the work begins to ensure the learning outcomes and that the project scope is relevant.
2. The examination project must be delivered according to the assignment description criteria and be evaluated with at least the passing grade.
3. Candidates are required to present his/her submitted work.

Resources
There is no specific reading list for this course, reference is made to previous courses reading lists.

Course Description: JavaScript Servers

Course Code	FI1BDJS05
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	JavaScript Servers
Norwegian	JavaScript servere
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	5
Length in Weeks	4
Teaching Hours	40
Teacher Supported Hours	40
Self-study Hours	88
Total Hours	**168**

The course can be taken as a single course	Yes, if the required prerequisite knowledge is covered.

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	Programming with Objects
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course provides knowledge of concepts, processes and tools of conventions and solutions for setting up a development environment for building server-based JavaScript solutions and deploying those servers on a local computer. Candidates will learn to use the Node runtime and the associated Node Package Manager (NPM). Through NPM they will explore the JavaScript ecosystem of tools and package management.

The focal point of these packages will be the use of external packages to build JavaScript servers. Candidates will be taught how to create, launch and maintain these servers locally on their own machines. These server solutions will encompass the concepts of routing, path variables, parameters and simple endpoints.

The course builds on competence from *Programming with Objects.*

Course Relevance

The Node runtime is a pivotal component of the JavaScript developer's toolkit; it is used in many frameworks as part of the development process, but is also used as a standalone production server. As projects grow in scope and complexity they begin to rely more on the use of external packages, and NPM provides a central tool for managing these dependencies within a given project.

Using their prior knowledge and the topics covered in this course, candidates will be able to create server solutions to meet the needs of simple single-purpose applications.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of concepts, processes, and tools that are used in Node based JavaScript projects and solutions
- has knowledge of concepts, processes, and tools that are used in Node Package Manger based project and solutions
- has knowledge of concepts, processes, and tools of server-based JavaScript solutions
- has knowledge of the industry of server-based JavaScript solutions and is familiar with the field of work
- can update his/her vocational knowledge of server-based JavaScript solutions

Skills

The Candidate…

- can apply vocational knowledge server-based JavaScript solutions to practical and theoretical solutions
- masters the use of Node as a tool to develop, deploy, and maintain general purpose JavaScript solutions
- masters the use of Node as a tool to develop, deploy, and maintain server-based JavaScript solutions
- masters the use of Node Package Manager
- masters the use of a server framework for building Node based web servers
- can find information and material that is relevant to selecting appropriate Node libraries for use in JavaScript solutions

General Competence

The Candidate…

- understands the ethical principles that apply when developing, deploying, and maintaining server-based JavaScript solutions
- has developed an ethical attitude in relation to the role of JavaScript server developers and maintainers
- can carry out work as a JavaScript server developer and maintainer

Learning Methods

Teaching varies between online based lessons, lectures, software training, activities, guidance and collaborative work.

Examination and Assessment

The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination

It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements

Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources

Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Course Description: Databases

Course Code	FI1BDDB05
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Databases
Norwegian	Databaser
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	5
Length in Weeks	4
Teaching Hours	40
Teacher Supported Hours	40
Self-study Hours	88
Total Hours	**168**

The course can be taken as a single course	Yes, if the required prerequisite knowledge is covered.

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	Programming Foundations
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course provides knowledge of techniques, principles and tools used build, maintain and utilise a relational database. Candidates will learn the basic operations needed to interact with a relational database (create, read, update, and delete). They will also be taught how to use normalization to improve their database designs. The candidates will be taught he principles and skills needed to include these databases in their server-based solutions.

The course builds on competence from *Programming Foundations*.

Course Relevance

In previous courses candidates have used external data sources, by learning how to use databases effectively they can create and manage their own data storages. The most common means of doing so is the sue of a relational SQL database, which is covered in this course.

SQL a separate language from JavaScript, but it is a simple language by comparison and a developer learning it will find many uses for it throughout his/her career.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of the concepts of relational databases and non-relational databases
- has knowledge of standard SQL and the elements that make up SQL statements
- has knowledge of the principles of CRUD (create, read, update, and delete)
- has insight into the impact of normalization on database efficiency
- has knowledge of the process of normalization up to 3rd normal form
- has knowledge of the steps necessary to integrate a database into a server-based solution
- has knowledge of the techniques, principles and tools used to provide basic security to a database

Skills

The Candidate…

- can apply vocational knowledge of normalization (up to 3rd normal form) to create or improve the design of a relational database
- masters writing simple SQL statements for use with a relational database
- can apply SQL language knowledge to create and maintain SQL statements that can create, read, update, and delete data
- can find information and material needed to integrate SQL database access into a server-based solution
- can apply vocational knowledge of database security to design or asses databases access systems

General Competence

The Candidate…

- can carry our work as an SQL programmer and maintainer

Learning Methods
Teaching varies between online based lessons, lectures, software training, activities, guidance and presentations.

Examination and Assessment
The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination
It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements
Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources
Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Course Description: REST APIs

Course Code	FI1BDRA75
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	REST APIs
Norwegian	REST API´er
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	7,5
Length in Weeks	6
Teaching Hours	60
Teacher Supported Hours	60
Self-study Hours	132
Total Hours	**252**

The course can be taken as a single course	Yes, if the required prerequisite knowledge is covered.

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	JavaScript Servers and Databases
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course provides knowledge of concepts, processes and tools that are used to build and test Representational State Transfer (REST) APIs. REST APIs are fast, reliable and expandable solutions to creating accessible data on the web. This course will greatly expand on the use of the JavaScript server solutions and will require candidates to build software that interacts with the databases solutions such as those that they created previously with SQL. Candidates will learn the skill necessary to use the Postman software (and opensource alternatives) to test and document their created APIs.

Candidates will be tasked with a project requiring that they create a working REST API using a database built to meet provided specification.

The course builds on competence from *JavaScript Servers* and *Databases.*

Course Relevance

Through this course candidates learn how to make data available to others; where as before they used external data sources (*JavaScript Servers*) and then internal (*Databases*), this course teaches them how to expose their data to external parties.

Modern web services rely more and more on REST APIs as the *de facto* means for transporting data, even when building enterprise level Java/.Net solutions the end result is still a REST API that is shares many commonalities between those that the candidates will study in this course.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of industry standard JavaScript frameworks that can be used to build RESTful services
- has knowledge of techniques, principles and tools used to implement API authentication
- has insight into REST principles and the features of a RESTful service
- can update his/her vocational knowledge of industry standard JavaScript REST frameworks

Skills

The Candidate…

- can apply vocational knowledge of RESTful principles to describe the features that identify a service as being RESTful
- masters creating RESTful solutions in JavaScript that make use of common HTTP methods
- masters the Postman software (or opensource alternatives) to test and document a REST API
- can apply vocational knowledge to implement API authentication in new and existing RESTful solutions
- can study an existing Database solution and design an appropriate REST service to expose desired data

General Competence

The Candidate…

- can develop RESTful solutions from database integration all the way to completed REST endpoints
- can carry out work as a REST API developer, maintainer, or tester

Learning Methods
Teaching varies between online based lessons, lectures, software training, activities, guidance and presentations.

Examination and Assessment
The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination
It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements
Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources
Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Course Description: Server Deployment

Course Code	FI1BDSD05
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Server Deployment
Norwegian	Server deployment
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	5
Length in Weeks	4
Teaching Hours	40
Teacher Supported Hours	40
Self-study Hours	88
Total Hours	**168**

The course can be taken as a single course	Yes, if the required prerequisite knowledge is covered.

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	JavaScript Servers and Databases
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						
External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

The course provides knowledge of concepts, processes and tools that relate to deploying their server software to hosted servers. Candidates will be able to authenticate users to their solutions and store session information on the server for returning users. The course will familiarise the candidates with the implications of GDPR on their solutions.

Candidates will learn the concepts, processes and tools necessary to ensure that their software meets customer specifications by using common JavaScript testing frameworks, and the principle of Test Driven Development (TDD).

Candidates will learn the concepts, processes and tools necessary to deploy their solutions to simple container services (Heroku) and other basic web hosts that support Node server solutions.

The course builds on competence from *JavaScript Servers* and *Databases.*

Course Relevance

This final subject course is when candidates need to understand the impact that their solutions will have on the world at large. GDPR is an extremely important consideration for all business today, and developers need to be aware about how their code can affect the rights of the user, and vice versa.

This course also deals with compliance of solutions to predefined testing criteria, these criteria will need to be 'passed' in the process of deploying a solution into containers and then onto a hosting server. Candidates will also need to be familiar with the associated maintenance of a hosted solution.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of concepts, process and tools that are used when packaging JavaScript server solutions for use in containers
- has knowledge of concepts, process and tools that are used to deploy container solutions into hosted containers
- has knowledge of concepts, processes, and tools that are used for authentication and session management in server-based JavaScript solutions
- has knowledge of frameworks used for creating unit tests in JavaScript
- has knowledge of the concepts, processes and tools that are used in Test Driven Development
- has knowledge of concepts, and processes of environment variables and server secrets
- has insight into GDPR as a it applies from the perspective of both users and developers

Skills

The Candidate…

- can apply vocational knowledge of authentication and sessions to create secure server solutions
- can apply vocational knowledge of coding style guides and standards to his/her own solutions and review of others solutions
- can apply vocational knowledge of environment variables and server secrets during server deployment

- masters building and deploying container-based JavaScript solutions such as those hosted through Heroku or other similar services
- masters the concepts of Unit Testing and can apply this knowledge to Test Driven Development
- can find information and material that is relevant to understanding the impact of GDPR regulations on a given software solution

General Competence

The Candidate…

- understands the ethical principles that apply to GDPR compliance
- has developed an ethical attitude in relation to the responsibilities of a server-based JavaScript solution that is intended to store sensitive user information
- can carry out deployments of containerized JavaScript solutions
- can carry out code review based on Unit Tests as part of solution deployment

Learning Methods

Teaching varies between online based lessons, lectures, software training, activities, guidance and presentations.

Examination and Assessment

The learning outcomes are assessed in compulsory assignments as Passed/Not Passed.

Re-Examination

It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements

Compulsory assignments must be submitted and evaluated with a passing grade. There will be given verbal guidance or written feedback together with the grade.

Resources

Teaching materials and essential resources will be shared on the LMS, in addition to the software user manuals.

Course Description: Exam Project

Course Code	FI1BDP275
Version	1.0
Language of Instruction	Norwegian and English
Language of Assessment	Norwegian and English
Course Name	
English	Exam Project
Norwegian	Eksamensprosjekt
Level in The Norwegian Qualifications Framework for Lifelong Learning	
Level 5: Tertiary Vocational Training 1	
Scope	
Credits	7,5
Length in Weeks	4
Teaching Hours	1
Teacher Supported Hours	12
Self-study Hours	155
Total Hours	**168**

The course can be taken as a single course	No

Required Prerequisite Knowledge	
Compulsory Prerequisite Knowledge	All previous courses.
Recommended Prerequisite Knowledge	-

Equipment

Required Equipment	Responsibility for procuring equipment				Semester	
	Campus		Online		1	2
	Campus	Student	Campus	Student		
PC		X		X	X	X
Internet Connection	X			X	X	X
Software						
Office Tools		X		X	X	X
Integrated development environment		X		X	X	X
Software for project file sharing, collaboration and presentations		X		X	X	X
Software for source code management		X		X	X	X
Recommended Equipment						

External storage		X		X	X	X
Headset and extra monitor		X		X	X	X

Academic Content

This is a major project that reflects competence the candidates have acquired during the academic year.

The candidate must solve the assignment independently, or as part of a group, from a given practical problem. Candidates should be encourage to form small groups based on the tasks/challenges of the given problem. Internship projects are encouraged and the project challenges candidates to find a real-world project to acquire practical experience in a professional setting. The candidate is responsible for all aspects of the project, in accordance with the supervisor through the internship, if applicable. If the candidate is not working on a real-world project, an alternative case project will be presented by the academic staff. The completed project will be presented to the teacher, sensor, fellow students and if applicable, the customer.
If the candidate is working on a real-world project, the candidate is required to provide documentation for the completed project even if it is not within the customer specifications.

The course builds on competence from all courses during the academic year.

Course Relevance

The aim of the course is to provide candidates with the ability to make independent choices and deliver a comprehensive product from start to finish with professional standards, deadline compliance and focus on efficient workflows. The candidate must demonstrate his/her ability to make reasonable and efficient choices though the use of software tools and detailed documentation, in addition to communicating with professional terminology and expressions during a project.

Learning Outcome

Knowledge

The Candidate…

- has knowledge of the web development industry and is familiar with the associated work
- has knowledge of presentation concepts and tools to present software solution to customers or project stakeholders
- can update his/her knowledge of JavaScript as a problem-solving tool and the roles present in a software development team

Skills

The Candidate…

- can apply vocational knowledge of JavaScript to solve a given problem
- can apply vocational knowledge of project management to describe a given project as a set of measurable tasks
- can apply vocational knowledge to present software solution to customers or project stakeholders
- masters contributing to a software project through industry-standard collaboration tools
- can find information and material that is relevant to a solving a given problem through purpose-built JavaScript software

- can study a given project and identify the software requirement needs of a JavaScript solution

General Competence

The Candidate…

- understands the ethical principles that apply in the field of web development industry
- has developed an ethical attitude in relation to the practicing of back-end development
- can carry out back-end development project based on the given project brief
- can carry out the task of documenting his/her source code contributions to a project
- can build and maintain relations with his/her peers, clients or other target groups
- can develop JavaScript projects of relevance to the software development industry

Learning Methods

Teaching consists of a single lecture to introduce the project, guidance, self-study, production and presentation.

Examination and Assessment

The learning outcomes are evaluated accordingly:
Examination projects are assessed with a letter grade from A-F.

Re-Examination

It is possible to re-sit this course after the guidelines clearly outlined in the school's examination rules.

Course Work Requirements

1. All projects must be approved by the teacher before the work begins to ensure the learning outcomes and that the project scope is relevant.
2. The examination project must be delivered according to the assignment description criteria and be evaluated with at least the passing grade.
3. Candidates are required to present his/her submitted work.

Resources

There is no specific reading list for this course, reference is made to previous courses reading lists.