

UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Tsotne Kekelia

**Development of a Course on Web-Application
Development in Software Engineering**

Master's Thesis (30 ECTS)

Supervisor: Dietmar Pfahl, PhD

Tartu 2020

Development of a Course on Web-Application Development in Software Engineering

Abstract:

This thesis describes the process of development and delivery of an updated version of the course “Web-Application Development” in the Institute of Computer Science of University of Tartu.

Main focus of the course is to familiarize bachelor’s students with the core concepts of web development, such as web servers, multi-tier architectures, development and deployment flows, front-end, back-end and their respective major technologies and tools. This thesis explains learning objectives and educational considerations that were used to model this course.

The thesis goes into details of materials and topics of the course. This paper also analyzes the student feedback and suggests future possible improvement ideas and considerations.

Keywords:

Web development, teaching, Web-Application

CERCS: P170 Computer science, numerical analysis, systems, control

Kursuse „Veebirakenduste loomine“ väljatöötamine tarkvaratehnikas

Lühikokkuvõte:

Käesolev uurimistöö kirjeldab kursuse “Veebirakenduste loomine” uuendatud versiooni väljatöötamise ja edasiandmise protsessi Tartu Ülikooli Arvutiteaduse Instituudis.

Kursuse peamiseks eesmärgiks on tutvustada bakalaureusetudengitele veebiarenduse põhilisi kontseptsioone nagu veebiserverid, mitmetasandiliste arhitektuuride loomine, juurutamine, front-end ja back-end arenduse olulisemad tehnoloogiad ja tööriistad. Uurimistöös selgitatakse õppe eesmärkide ja õppimist soodustavate tegurite arvestamist kursuse väljatöötamisel.

Uurimistöös kirjeldatakse detailselt kursuse materjale ja teemasid. Ühtlasi analüüsitakse tudengite tagasisidet ning tuuakse välja ettepanekud ja ideed kursuse täiustamiseks tulevikus.

Võtmesõnad:

Veebiarendus, õpetamine, veebirakendused

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1	Introduction	5
2	Course Design	6
2.1	Learning Objectives.....	6
2.2	Related Work.....	7
2.2.1	Previous Versions of the Course.....	7
2.2.2	Learning Resources Available Online	7
2.3	Schedule	8
2.4	Didactic Considerations.....	9
2.5	Grading Scheme	9
3	Course Materials	10
3.1	Types of Materials Created	10
3.1.1	Lecture Slides	10
3.1.2	Practice Session Guidelines	10
3.1.3	Homework	10
3.1.4	Exam	11
3.2	Topics	11
3.2.1	Introduction to the Course / Multi-tier Architecture / HTTP.....	11
3.2.2	HTML	12
3.2.3	CSS	12
3.2.4	JavaScript Basics	12
3.2.5	JavaScript in Depth.....	12
3.2.6	DOM and jQuery	13
3.2.7	AJAX / JSON.....	13
3.2.8	Vue.js Part 1.....	14
3.2.9	Vue.js Part 2.....	14
3.2.10	Vue.js Part 3	14
3.2.11	Back-end: Introduction to NodeJS	14
3.2.12	Back-end: Databases and Authentication.....	15
3.2.13	Testing.....	15
3.2.14	Where to go from here: Roadmap	15
4	Results and Course Evaluation.....	16
4.1	Analysis of Feedback	16
4.2	Improvement Ideas	17
5	Conclusion.....	18

6	References	19
	Appendix	21
	I. Exams	21
	II. Feedback Questions	38
	III. License	42

1 Introduction

Web-development is an increasingly large and complex field of study, since it is highly applicable in the real-world applications and undergoes unceasing change due to constant evolution of the industry.

In past years thanks to the advances in hardware and increased demand from the industry, myriads of new technologies (libraries, frameworks, programming languages and other pieces of technology) appeared on the market to help businesses to achieve their technical goals. Due to these rapid advancements in technology modern web-developers need to be familiar with an array of different technologies and keep themselves up to date with the new industry standards. This continuous evolution makes it very difficult for newcomers to navigate their professional paths.

Web-development generally consists of three major parts:

- Client facing product, so called Front-end
- System to handle business logic, so called Back-end
- Deployment and delivery infrastructure, this consists of all the hardware and networking needed to run the front-end and back-end software and fulfil business requirements

Today's market is oversaturated with different so-called “Stacks”. “Stack” is a bundle of technologies and tools that usually consist of some kind of front-end framework, back-end programming language and framework, operating system, web server technology or a platform and a database technology. Just to name a few:

- LAMP (Linux, Apache, MySQL and PHP)
- MEAN (MongoDB, Express.js, Angular, NodeJS)
- MERN (MongoDB, Express.js, React, NodeJS)
- MEVN (MongoDB, Express.js, Vue.js, NodeJS)

In order to keep up with the progress in the industry of the recent years it was decided to update the contents of the “Web-application development” course with code LTAT.05.004, This course is aimed at third year bachelor’s students.

This thesis describes work that went into updating the contents of this course and justify decisions made. Section 2 is dedicated to explaining the learning objectives of the course and lists related work that was used as a reference and foundation for this course. Section 3 provides a detailed overview of different kinds of materials that were created for this course and provided to students, it also lists the topics that were introduced to students. Lastly, Section 4 analyses student feedback and also provides detailed results for homework and exams. Possible future improvements are also suggested.

2 Course Design

This section describes decisions that were made during designing the course. In Section 2.1 the learning objectives of the course are described, taking into consideration past versions and their feedback and different learning resources available online, which are discussed in Section 2.2. Section 2.3 is dedicated to the course's schedule. Section 2.4 discusses didactic considerations that were taken into account and helped to shape the final form of the course and finally Section 2.5 describes the grading scheme of the course.

2.1 Learning Objectives

The purpose of this course is to help students understand and familiarize them with core concepts, technologies and tools of web-application development. The subject of web-application development is too wide to cover fully in one semester course, so decision was made to focus on the following aspects:

- **Multi-tiered Architecture** - Modern web-applications largely consist of multiple different logical or physical tiers and layers, thus It was deemed necessary to educate students about such types of software and it's benefits and detriments.
- **Hypertext Transfer Protocol (HTTP)** - Before providing students with more information about actual development of web-applications it was decided that they would benefit from learning about underlying mechanisms of web-applications and how web browsers communicate with the software that they would develop in the future.
- **Hypertext Markup Language (HTML)** - HTML is a cornerstone of Web-application development and this course went into great detail of its core concepts and usages. Decision was made to familiarize students with the newest version HTML, HTML5
- **Cascading Style Sheet (CSS)** - Another crucial technology for web development, this course touched upon most useful parts of CSS and students from the beginning were introduced to newest version CSS3
- **JavaScript** - It is practically unimaginable that a modern web-application doesn't use this language at some capacity, therefore it was decided that big portion of this course would be dedicated to teaching students JavaScript syntax, and it's capabilities with heavy focus on more modern additions such as lambda functions, generators and promises to name a few.
- **Package Management** - Managing different libraries and modules from various vendors is a tedious and painful process, it is important for modern web developers to leverage power that modern package managers offer, to this end Node Package Manager (NPM) was selected.
- **Front-end Frameworks** - It was deemed necessary to familiarize students with the modern approach of enterprise level web-application development techniques. Out of three most popular JavaScript frameworks, Angular, React and Vue.js latter was deemed most appropriate for this course due to its very low learning curve [1]
- **Back-end Frameworks** - This course is also heavily focused on back-end technologies to present full scope of web-application development process to students, to this

end decision was made to introduce students with Express.js framework built on top of NodeJS platform, this would ensure that students are familiar with core back-end concepts and are comfortable with the syntax since NodeJS uses JavaScript language. Concern was raised that, if we had introduced technologies or frameworks built on other languages such as Java, C#, PHP, Python or Ruby students would be more concerned with learning new syntax rather than focusing on back-end concepts and architecture.

- **REST API** - Another industry standard that is essential part of a modern web-developer toolkit. REST API concepts and techniques were presented to students and during the course they had to build and utilize their own REST APIs.
- **Testing** - Crucial part of any type of software development, students were presented with advantages and necessity of rigorous automated testing.

2.2 Related Work

This section describes how an already existing version of this course and other online materials helped to shape it.

2.2.1 Previous Versions of the Course

“Web-application development” course with code LTAT.05.004 in previous years, touched upon multi-tier architecture and its concepts, these materials were used as a basis for updated lectures on the same topic.

Previous version of the course heavily relied on XML technology as a communication tool between client-side(front-end) and back-end parts of a web-application. However, in recent years industry heavily favours JavaScript Object Notation (JSON) as a standard for data exchange and communication between different tiers of multi-tiered web-applications. Therefore, it was decided that students this year would be familiarized with JSON instead of XML.

2.2.2 Learning Resources Available Online

There are a multitude of learning resources available online, books, articles, online courses, tutorials. For this course students were encouraged but not required to use materials available at Scrimba [2]. This web-site allows students to follow comprehensive video tutorials and complete coding tasks presented in video interactively in the same environment, moreover this website covers almost all major parts of the learning objectives set out for this course.

A book by Roxane Anquetil [18] provides good insight for complete beginners into the topic of HTML5L, CSS3 and JavaScript. This book also offers information about various topics such as: history of the “World Wide Web”, Search Engine Optimization (SEO), hosting and domain name management and more. It needs to be noted that SEO and domain name management were not in the scope of the updated course on web-application development.

Simply Web Development: An Introduction to the World of Web Development by Anna Voice and Ray Voice [19] provides detailed overview of the fundamental concepts of front-end technologies such as: HTML, CSS, JavaScript, ReactJS and also touches upon back-end side of web-applications, using NodeJS as a demonstration for web server capabilities and MongoDB as a database engine. Approaches used in this book to describe back-end concepts were adapted for our course.

The online education platform Coursera provides a free online course titled *HTML, CSS, and JavaScript for Web Developers* [22] offered by Johns Hopkins University. This online course consists of approximately 28 hours of video lectures on the topics of HTML5, CSS3 and JavaScript.

Node JS Training: Learn and Understand Node JS [21] is an online course offered by an online education platform Udemy. This course offers extensive overview and detailed insight of inner workings and concepts of NodeJS platform. Topics of theory and motivation behind NodeJS were adapted from this course. However, since the scope of our course was somewhat limited we did not incorporate some of the topics discussed in this online course such as V8 the JavaScript runtime engine and the concept of NodeJS pipes.

The online course titled *The Web Developer Bootcamp* [20] offered by Udemy, had the biggest influence on shaping of our course. This is a bootcamp style video course aimed at beginners. It contains all the topics offered by our course and in addition it offers tutorials on Bootstrap4 and web-application deployment. The approach of bootcamp style teaching and providing most up-to-date information to students makes this course extremely popular. It has well over 500 000 students enrolled and 4.6 (out of 5) star rating with over 170 000 reviews.

2.3 Schedule

The course was developed according to the European Credit Transfer and Accumulation System (ECTS) with a score of 6 ECTS.

Duration of the course was sixteen weeks. The course consisted of two parts, lectures and practical sessions, every week there was one ninety minute lecture presented by Tsotne Kekelia and five practical sessions, each of them ninety minutes long, two of them on Mondays and Wednesdays held by Tair Vaher and three of them on Mondays, Tuesdays and Wednesdays held by Tsotne Kekelia.

The learning objectives were broken down into weekly topics and are listed below:

- Week 1 (September 2nd, 2019): No lecture, practical session was used to introduce students to the course
- Week 2 (September 9th, 2019): Multi-tier architecture / HTTP
- Week 3 (September 16th, 2019): HTML
- Week 4 (September 23rd, 2019): CSS
- Week 5 (September 30th, 2019): JavaScript basics
- Week 6 (October 7th, 2019): JavaScript in depth
- Week 7 (October 14th, 2019): DOM and jQuery
- Week 8 (October 21st, 2019): AJAX / JSON
- Week 9 (October 28th, 2019): Vue.js Part 1
- Week 10 (November 4th, 2019): Vue.js Part 2
- Week 11 (November 11th, 2019): Vue.js Part 3
- Week 12 (November 18th, 2019): Back-end: Introduction to NodeJS

- Week 13 (November 25th, 2019): Back-end: Databases and Authentication
- Week 14 (December 2nd, 2019): Testing
- Week 15 (December 9th, 2019): Where to go from here: Roadmap
- Week 16 (December 16th, 2019): Preparation for the exam

The last step of the course was an exam. Two exam dates and one resit exam date were selected: January 6th, 2020 and January 13th, 2020, resit exam was scheduled on January 27th, 2020. A detailed description of the materials presented in each lecture and practical session as well the exam variants are presented in Section 3.

2.4 Didactic Considerations

The course includes sixteen in-class lectures and practical sessions, each ninety minutes long. During the lectures, the instructor presented to the participants the topic to be covered in that session. Depending on the character of the particular topic, the historical information, theoretical knowledge, motivation and respective tools and software were introduced.

During the practical sessions, the participants were provided with a set of hands-on tasks on the same topic, which illustrated the practical usage of the concepts introduced. The practical sessions were held in class and were led by the instructors so that participants could receive help, advice and clarifications if needed.

All of the course materials, lecture slides, practical session exercises, homeworks and exams were developed and presented by Tsothe Kekelia in accordance with the advice and guidance from Marlon Dumas, PhD.

The last lecture was used to review the course and prepare for the exam. Sample exam questions were provided, and students were encouraged to ask questions about the topics of the course or the topics that were not covered but related to the course.

The Students also had to fulfil four homeworks given to them on week 3, week 6, week 10 and week 14. Students were split into groups of 3-4 and they had to present the final product to the instructor in the practical session of the next week.

2.5 Grading Scheme

Total possible amount of points for students was 100. 50 points were dedicated to four homeworks, first one was worth 5 points, second - 10 points, third - 15 points and last homework was worth 20 points. Other 50 points were dedicated to the exam, which was multiple choice quiz, 34 questions total, with 21 questions giving students 1 point each, 10 questions giving students 2 points each and 3 questions worth 3 points each.

100 points were mapped to a differentiated grading system, where students needed at least 51 points to pass the course and get the grade E, students that failed this criteria or failed to sign up or show up for the exam and a resit got the grade F.

3 Course Materials

A variety of study materials and resources were created for the course. They formed the structure of the in-class activities and provided students necessary information and online resources. The materials can be reused in the future iterations of the course.

This section is dedicated to providing a detailed view of the materials created for the course. Firstly, different types of the materials and resources that were created are discussed. Secondly all topics are explained in a great detail.

3.1 Types of Materials Created

Central repository for course materials was necessary for students to easily access all course materials, to this end a course webpage was created in the Institute of Computer Science courses environment [3]. This repository hosted the Portable Document Format (PDF) versions of lecture slides, URLs to classwork tasks for each practical session and homework submission page.

Active communication between instructors and students was essential for this course, for this purpose Slack channel was created where students could directly communicate with the instructors, each other or subscribe to announcements channel, which was used by instructors to deliver important information to participants.

A Google form [7] document [4] was created for students to register their and their homework groupmate names.

A grading website [5] was developed to grant students access to their progress of the course. This website also had an administrative part where instructors could grade each student's homework.

3.1.1 Lecture Slides

Set of lecture slides was used by the instructor to present relevant historical, theoretical and technical facts about each topic. Slides include necessary references to the resources which would help students study each topic in more detail.

The slides were developed using the Google Slides [6] software. The slides also were exported as a Portable Document Format (PDF) and uploaded to the course webpage, this way, they could be easily accessed and downloaded by the students.

3.1.2 Practice Session Guidelines

Every week necessary materials for practice sessions were uploaded to the course website. Practice sessions consisted of two parts, firstly instructors would present a challenge, based on that week's topic, explain key concepts, showcase end results and give necessary instructions depending on tasks at hand. Secondly, students were given a number of tasks to be completed independently. Students were encouraged to ask questions or seek help and further explanations from the instructors.

3.1.3 Homework

Homework consisted of several tasks to be completed in groups of 3 or 4. Students were required to use online repository hosting service GitHub [16] to host their solutions. One student from the group was required to present the solution to the instructor and provide proof of work in the form of git commits.

There were four homeworks given to the students during the entire course. Homework grades were instantly entered in the grading website [5]. First homework was given to the students in week 3 and was worth 5 points. This homework was presented by students in week 4's practical sessions. Second homework was given to the students in week 6 and was worth 10 points. This homework was presented by students in week 7's practical sessions. Third homework was given to the students in week 10 and was worth 15 points and it was presented by students in week 11's practical sessions. Lastly, fourth homework was given to the students in week 14 and was worth 20 points and it was presented by students in week 16's practical sessions.

3.1.4 Exam

Originally instead of exam it was decided that students would have to develop a final project that they would have to incorporate all the knowledge they gained during the course but due to number of the students registered to the course, time constraints and the fact that there were only two instructors available, decision was made in favour of a final exam.

Format of the exam was a multiple-choice quiz. There were two dates designated to exams January 6th, 2020 and January 13th, 2020. Each exam had two variants A and B. Each exam variant consisted of 34 questions. 21 questions that gave students 1 point, 10 questions with 2 points and 3 questions with 3 points each.

All the exam questions are presented in the Appendix.

3.2 Topics

Materials for topics to be covered were mostly sourced from official documentations and specifications of each subject. Materials were mostly gathered from online resources such as Mozilla Developer Network (MDN) [8], Vue.js official documentations [9], NodeJS official documentations [10] and Express.js official documentations [11].

3.2.1 Introduction to the Course / Multi-tier Architecture / HTTP

This lecture consisted of three major parts: Introducing students to the course, covering multi-tiered architecture and explaining working mechanics of HTTP. In the first part students were introduced with the course's learning objectives, instructors, structure and grading scheme. Required tools and software were also introduced to students.

In the second part students were given an overall overview of the web-application development process as well as concept of web-application itself. Students were familiarized with the ideas and concepts of multi-tiered architecture and different layers that this architecture consists of. Client-server communication concepts were also covered during this lecture.

In the third part of the lecture students were described what HTTP is and how it works. They were presented with explanations of technical terms such as URL and URI and most commonly used HTTP methods and their characteristics.

Practical session for this week consisted mostly with helping students set up their work environments and get familiar with the necessary pieces of software for the course such as version control system (GIT) [12] and API development environment Postman [13]. In the practical session students were also tasked with making different types of HTTP requests to mocked endpoints to understand the flow of the communication between client and server.

3.2.2 HTML

In this lecture students were presented with core concepts and history of HTML. The syntax and components of HTML language were explained in great detail. Students were shown how marked up text files are interpreted by web browsers to visualize web-application structure. Most common formatting, structural and media HTML elements were explained in great detail with examples.

Lecture went into great details of working mechanisms of user input elements that HTML provides for developers. Students were also introduced to the concept of responsive web applications and the importance of accessibility features that HTML provides.

In the practical session students were instructed to clone the pre-existing GitHub repository, that was prepared before session by the instructor, and fulfil sets of tasks that demonstrated in practical terms all the concepts that were presented during the lecture.

3.2.3 CSS

During this lecture, syntax structure and working mechanisms of CSS were discussed. Instructor presented the history and motivation behind CSS creation and its need in web-applications. Students were shown all possible selector types, combinators and pseudo classes and elements.

Lecturer also went into technical details and specifications of cascading and precedence which should provide students with a deep understanding of CSS.

Students were familiarized with the units and values specific to CSS and up to date technologies that modern CSS3 introduced.

During practical sessions students were tasked to continue on the project created in the previous session and apply the concepts of the lecture practically. Students were also tasked with independent tasks one of which was to adapt existing web-application for a mobile device use.

In this week students were also given the first homework, which they had to complete in groups of 3 or 4 and had to present the results in week 4. Homework combined concepts taught in the first three weeks. Students were required to work in a collaborative manner and had to present a proof of done work in the form of GIT commits.

3.2.4 JavaScript Basics

This lecture was dedicated to presenting to students the history and basic core concepts of JavaScript language. First part of the lecture was dedicated to explaining how to integrate JavaScript files into the HTML application. In the second part syntax of basic functionality was explained such as variable types and their declaration, operators, conditional and loop statements. In the Third part of the lecture error types and debugging techniques were discussed.

This week was dedicated to homework grading; thus, no practical tasks were given to students. Results for this homework will be presented in Section 4.

3.2.5 JavaScript in Depth

This lecture went into depth of JavaScript concepts. Important structures such as functions were discussed. Concept of global and local scopes were explained and demonstrated to students.

Next part of the lecture was dedicated to one of the most used features of JavaScript, events. Events were discussed in great length with examples, common problems and their solutions in the form of capturing and bubbling were demonstrated to students.

After that, the lecture went into explaining Object Oriented Programming (OOP) approach in JavaScript explaining structures such as classes and objects. New additions to JavaScript such as promise, iterators and generators were also presented to students.

In the practical session it was decided to give students a task that would incorporate the knowledge of all the previous weeks, to this end students were tasked with building a calculator software, this would ensure that they would have to use knowledge they gained about HTML, CSS and JavaScript and how to combine these technologies together to achieve end result.

3.2.6 DOM and jQuery

In this lecture students were familiarized with the concept of Document Object Model (DOM) and its possible manipulations that JavaScript allows. History and structure of DOM was also discussed.

Students were also familiarized with different types of local storage technologies that modern web browsers offer such as: Local Storage, Cookies and Indexed DB.

One of the most popular and used JavaScript libraries, jQuery was introduced to the students, concern was raised that jQuery as a technology is outdated by modern standards but argument for including jQuery in this course suggested that, students need to understand natural progression and evolution of web-application development throughout years and the market requirements that made jQuery and other similar libraries outdated in favour of more robust and extensive frameworks. This way understanding the importance of the frameworks would be easier for students that have little to no experience in web-application development.

Practical session for this week required students to apply OOP principles of JavaScript and use jQuery library to add functionality to the pre-existing project.

This week second homework was also introduced. Students were tasked to build and style a simple web page that would display student information, courses they have taken and grades. Web page was supposed to have an interactive interface where users could add new data and newly entered data needed to be reflected on the web page.

3.2.7 AJAX / JSON

The crucial technology in developing dynamic web-applications, Asynchronous JavaScript And XML (AJAX) was introduced to students in week 7. Students were presented with history and motivation behind creating AJAX. Students were also shown in detail the sequence of actions AJAX uses to achieve its goals. Security issues and best practices concerning AJAX use were also discussed.

Second half of this lecture was dedicated to the modern standard in data transportation, JavaScript Object Notation (JSON). JSON and its syntax were discussed in great detail using examples. Comparison between JSON and XML was extensive in order to underline the advantages that JSON possess over XML.

Practical session in this week was dedicated to grading homework that was given to the students in week 6.

3.2.8 Vue.js Part 1

This lecture consisted of three parts. Firstly, students were introduced to NodeJS platform. Advantages and technical details as well as possible pitfalls were discussed, and appropriate examples were provided. Concept of Node modules were introduced to students that tied with the theme of the second part of this lecture.

Secondly, the importance of package managing in modern web-application development was highlighted. General working principles and concepts of package versioning, management and maintenance were discussed using Node Package Manager (NPM) as an example.

Finally, Main ideas and core concepts, as well as advantages and disadvantages of using frameworks were presented to the students, this led to introduction of Vue.js framework and its main building block, Vue component. Importance of componentization of large front-end applications was underlined with real-life examples.

In the practical session students were required to install NodeJS platform on their computers and use NPM to initialize a blank Vue.js project. This project was used by instructors to familiarize students with the file structure and unique characteristics of Vue.js framework.

3.2.9 Vue.js Part 2

Lecture in week 9 was dedicated to presenting students with an in-depth overview of the inner mechanisms of Vue.js framework. Templating specifications were presented in great detail with examples. Important notions such as properties, methods and data state of a Vue.js component were explained extensively. Lifecycle hooks of a Vue.js component, which is one of the cornerstone concepts of this framework, were presented in great detail.

The other part of this lecture was used to explain to students the concept of Vue.js directives and how to create new directives and use built-in ones.

In this week's practical session students were tasked to extend an already existing project that was provided to them by instructors, using the newly acquired knowledge.

This week third homework was given to students. In this homework students were required to refactor the web page created in the homework 2, into Vue.js application. This would help them apply Vue.js concepts in practice and would also highlight advantages of using frameworks in comparison to conventional JavaScript and jQuery approach.

3.2.10 Vue.js Part 3

Data binding is a key concept of Vue.js framework, it is one of the main reasons this framework exists, week 10's lecture was dedicated entirely to this technique.

Students were presented with the history and motivation behind two-way data binding and its necessity in modern web-applications. Syntax and enhancements of this technique were also discussed.

Practical session was dedicated to grading the homework 3.

3.2.11 Back-end: Introduction to NodeJS

Even though NodeJS was introduced as a platform to students in week 8, in week 11 lecture the instructor went into great detail of the structure and inner mechanisms of non-blocking I/O features of NodeJS.

General ideas and concepts of back-end applications such as Model View Controller (MVC) architecture were also discussed during the lecture. This led to the introduction of REST

API and principles that define it. Advantages, standards and naming conventions of REST API were presented to students.

In the practical session students were tasked with creating a simple to-do back-end application, which would have capability of:

- 1) Retrieving existing tasks from a file
- 2) Ability to create new task and store it in a file

3.2.12 Back-end: Databases and Authentication

In this lecture students were presented with an extensive overview of different database technologies. Importance of data persistence as a concept was underlined. Detailed comparison between NoSQL and SQL type databases were presented to students, detailing each one's specifics and highlighting similarities and differences.

Second half of this lecture was dedicated to the mechanisms of authentication and authorization. Students were presented with several different authentication mechanisms such as: Basic Auth, JSON Web Token (JWT) and sessions.

Practical session task of this week was based on the previous week's project and students had to refactor the code so that instead of persisting data in a file on the local disk drive, data would be stored in a database. To this end SQLite was selected, due to its light-weightness.

3.2.13 Testing

Modern web-application development is unimaginable without rigorous testing. To this end, this lecture was entirely dedicated to highlight the importance, necessity and advantages of properly implemented tests. Students have been familiarized with different types of testing and how to approach each one of them.

Practical session was dedicated to one single task: to cover Vue.js application that was created in previous practical sessions with tests.

In that week last homework was given to students as well. This homework required students to cover pre-existing application with tests, the minimum requirement of the homework was to get the statements coverage at least to 70%.

3.2.14 Where to go from here: Roadmap

It was deemed necessary to provide students with resources and guidance to help them plan their future careers. To this end, in this lecture students were presented with the developer roadmap [14], the resource which helps up and coming web developers navigate through technologies and build their career paths.

In the practical sessions sample exam quiz was presented to the students to help them understand the format and scope of the final exam.

4 Results and Course Evaluation

This section is dedicated to present results of the course by analysing student feedback. Student feedback is gathered from the University of Tartu information system [15].

4.1 Analysis of Feedback

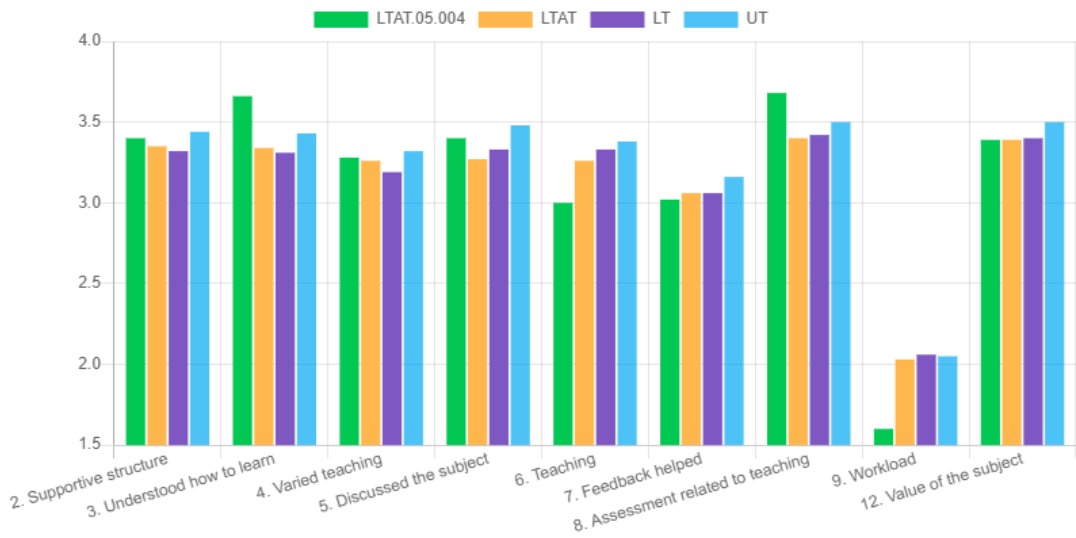


Figure 1. Feedback comparison of LTAT.05.004 with LTAT, LT and UT [17]

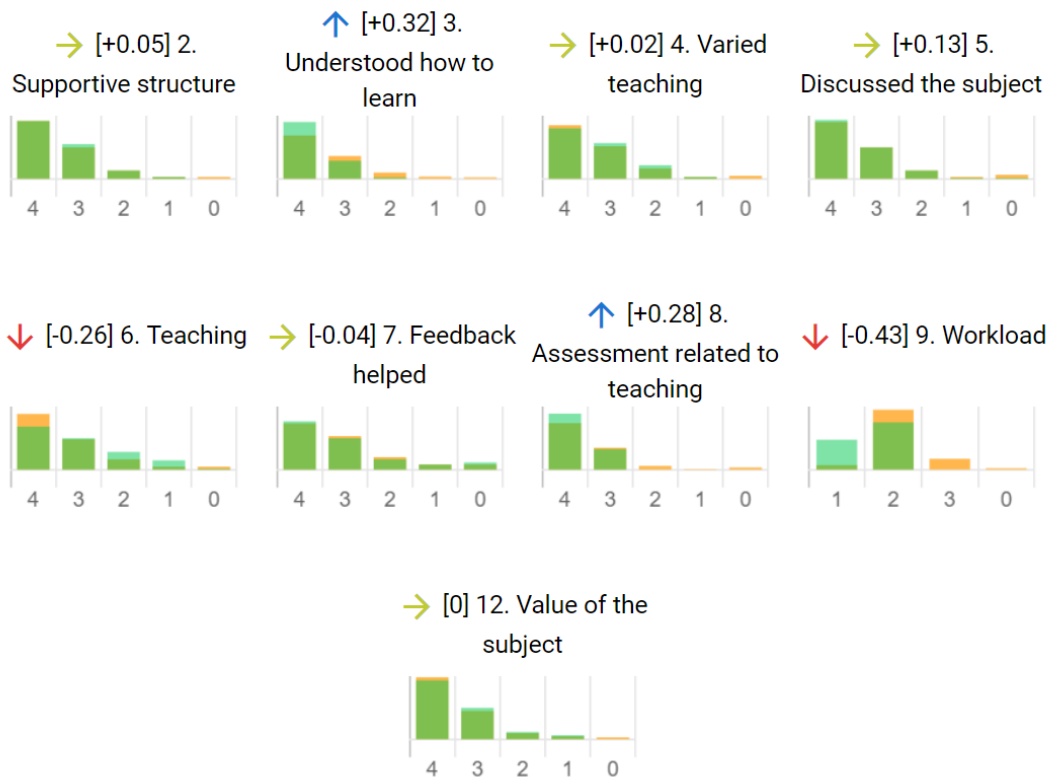


Figure 2. Comparison with institute's average [17]

Analysis of Figure 1 and Figure 2 shows notable improvements on two dimensions compared to the university and institute average:

- “Understood how to learn” (+0.32)
- “Assessment related to teaching” (+0.28)

On the other hand, the course was evaluated with the lower scores in two other dimensions:

- “Teaching” (-0.26)
- “Workload” (-0.43)

Lower score in “Teaching” dimension can be explained by inexperience of the instructors. For both of them this was first time teaching and they lacked necessary experience. As for the lower score in “Workload” can be explained by the fact that there were only two instructors and 153 registered students for the course. It was challenging to balance workload with necessary time for instructors to evaluate work done by students and providing them with relevant feedback. Improvement ideas for this problem can be found in Section 4.2.

Detailed list of the questions of the feedback questionnaire with their respective results can be found in Appendix, in the section II. Feedback Questions.

4.2 Improvement Ideas

The student feedback shows that student satisfaction with the course was moderately high. Only one dimension, “workload” was evaluated lower than that of institute average. To this end it would be beneficial to change homework structure and instead of offering four homeworks per semester, offer students weekly homeworks, of course increasing the number of instructors would be necessary to manage this workload.

Another improvement idea would be to offer students the system for individual work in the form of a platform where they would be proposed with a number of coding challenges and automated tests would evaluate their solutions. This system would improve their skills and help practically apply learned materials.

To provide students with the full picture of the web-application development, it would be extremely helpful to introduce them to the topic of deployment and concepts of domain names and name servers.

The last suggestion would be to introduce the final project instead of an exam. This will help students practice collaborative and coding skills that are essential for success in the industry.

5 Conclusion

This thesis describes the process of development and delivery of an updated version of the course “Web-Application Development” in the Institute of Computer Science of the University of Tartu.

Main goal of the course is to familiarize bachelor’s students with the core concepts of web development, such as web servers, multi-tier architecture development and deployment flows, front-end, back-end and their respective major technologies and tools. This thesis details learning objectives and educational considerations that were used to shape this course.

In Section 2 learning objectives are discussed. Previous iteration of the course and other online resources are also described in this section. Section 2 also gives an overview of the course schedule, didactic considerations that were taken into account and explains the grading scheme.

Several types of learning materials were created for this course, which include lecture slides, practical session tasks, homeworks and exams are detailed in Section 3. Topics presented to students each week are also explained and detailed.

Finally, Section 4 describes, and analyses student feedback gathered from the University of Tartu information system [15]. Possible future improvements based on the student feedback are also proposed in this section.

All of the resources that were created for this course are accessible from the course web page in the courses environment of the Institute of Computer Science of the University of Tartu [3].

6 References

- [1] “Comparison with Other Frameworks,” [Online]. Available: <https://vuejs.org/v2/guide/comparison.html#Learning-Curve>. [Accessed 13 06 2020].
- [2] “Learn to Code with Interactive Tutorials | Scrimba,” [Online]. Available: <https://scrimba.com>. [Accessed 13 06 2020].
- [3] “Web application development - Courses - Institute of Computer Science,” [Online]. Available: <https://courses.cs.ut.ee/2019/wad/fall>. [Accessed 14 06 2020].
- [4] “Web Application Development 2019,” [Online]. Available: https://docs.google.com/forms/d/e/1FAIpQLScmualWQ-IbP8R8toORU_NbMuVYC0Vm9fqoq1imXVpWT98_Fw/viewform. [Accessed 14 06 2020].
- [5] “Web Application Development,” [Online]. Available: <https://wad-2019.herokuapp.com>. [Accessed 14 06 2020].
- [6] “Google Slides: Free Online Presentations for Personal Use,” [Online]. Available: <https://www.google.com/slides/about>. [Accessed 14 06 2020].
- [7] “Google Forms: Free Online Surveys for Personal Use,” [Online]. Available: <https://www.google.com/forms/about>. [Accessed 14 06 2020].
- [8] “MDN Web Docs,” [Online]. Available: <https://developer.mozilla.org/en-US>. [Accessed 14 06 2020].
- [9] “Vue.js,” [Online]. Available: <https://vuejs.org>. [Accessed 14 06 2020].
- [10] “Docs | NodeJS,” [Online]. Available: <https://nodejs.org/en/docs>. [Accessed 14 06 2020].
- [11] “Express - NodeJS web application framework,” [Online]. Available: <https://expressjs.com>. [Accessed 14 06 2020].
- [12] “Git,” [Online]. Available: <https://git-scm.com>. [Accessed 14 06 2020].
- [13] “Postman | The Collaboration Platform for API Development,” [Online]. Available: <https://www.postman.com>. [Accessed 14 06 2020].
- [14] “Developer Roadmaps,” [Online]. Available: <https://roadmap.sh>. [Accessed 15 06 2020].
- [15] “ÕIS II,” [Online]. Available: <https://ois2.ut.ee>. [Accessed 15 06 2020].
- [16] “The world’s leading software development platform · GitHub,” [Online]. Available: <https://github.com>. [Accessed 15 06 2020].
- [17] “ÕIS II,” [Online]. Available: <https://ois2.ut.ee/#/courses/LTAT.05.004/version/lt-2019-autumn-fulltime-c6-en/feedback>. [Accessed 15 06 2020].
- [18] R. Anquetil, Fundamental Concepts for Web Development: HTML5, CSS3, JavaScript and Much More! for Complete Beginners!, Independently Published, 2019.
- [19] R. Voice and A. Voice, Simply Web Development: An Introduction to the World of Web Development, Independently Published, 2019.
- [20] “The Web Developer Bootcamp: Learn HTML, CSS, Node, and More! | Udemy,” [Online]. Available: <https://www.udemy.com/course/the-web-developer-bootcamp/>. [Accessed 18 06 2020].

- [21] “Node JS Training: Learn and Understand Node JS | Udemy,” [Online]. Available: <https://www.udemy.com/course/understand-nodejs/>. [Accessed 18 06 2020].
- [22] “HTML, CSS, and Javascript for Web Developers | Coursera,” [Online]. Available: <https://www.coursera.org/learn/html-css-javascript-for-web-developers>. [Accessed 18 06 2020].

Appendix

I. Exams

Exam 06.01.2020: Variant A

1. **Syntax of style attribute of an HTML tag is? (1 point)**
 - a. `<style="property;">`
 - b. `<tagname style="property">`
 - c. `<style="value;">`
 - d. `<tagname style="property:value;">`
2. **Which of these CSS selectors will select all div tags with “example“ class? (1 point)**
 - a. `div.example`
 - b. `div#example`
 - c. `.example div`
 - d. `div .example`
3. **\$("#p") What does it select? (1 point)**
 - a. All p tags
 - b. First p tag
 - c. Last p tag
 - d. All div tags
4. **What is the correct jQuery code to set the background color of all p elements to red? (1 point)**
 - a. `$("#p").css("background-color", "red");`
 - b. `$("#p").manipulate("background-color", "red");`
 - c. `$("#p").change("background-color", "red");`
 - d. `$("#p").background("color", "red");`
5. **\$("#div p") What does it select? (1 point)**
 - a. All div elements with class p
 - b. All p elements that are inside div element
 - c. All p elements with id div
 - d. All p elements with class div
6. **\$("#p#intro") What does it select? (1 point)**
 - a. p tag with class intro
 - b. p tag with id intro
 - c. intro tag with id
 - d. tag with id intro that is child of a p tag

7. **When we need to create an ordered list which markup we need to use? (1 point)**
- ``
 - ``
 - ``
 - `<nl>`
8. **Which of the following table tags is used to create a table row? (1 point)**
- `<th>`
 - `<td>`
 - `<tr>`
 - `<table>`
9. **Which of the following HTML tags is not valid? (1 point)**
- `<h1>`
 - `<h8>`
 - `<h4>`
 - `<h5>`
10. **`<h6>` is the biggest header tag. (1 point)**
- True
 - False
11. **What HTML form input should we use to present multiple options, but to select only one? (1 point)**
- `<input type="text">`
 - `<input type="radio">`
 - `<input type="checkbox">`
12. **Which is the correct syntax to use a javascript file? (1 point)**
- `<script src="xxx.js">`
 - `<script href="xxx.js">`
 - `<script path="xxx.js">`
 - `<script link="xxx.js">`
13. **How can we execute the function “myFunction” in javascript? (1 point)**
- `invoke(myFunction)`
 - `call(myFunction)`
 - `myFunction()`
14. **Which is the correct way to use comments in Javascript? (1 point)**
- `{# ... #}`
 - `<!-- --!>`
 - `//`
 - `>> ...`
15. **Which is the correct way to declare an array in Javascript? (1 point)**
- `let names = ['John', 'James', 'Bob']`
 - `let names = array('John', 'James', 'Bob')`
 - `let name = 'John', 'James', 'Bob'`
 - `let names = [0]=>'John', [1]=>'James', [2]=>'Bob'`

16. **Which is correct?** (1 point)
- a. `i =+ 1;`
 - b. `i += 1;`
 - c. `i = i++1`
 - d. `+i+`
17. **How can we get the DOM element with id?** (1 point)
- a. `window.getElementById(...)`
 - b. `document.getElementById(...)`
 - c. `page.getElementById(...)`
 - d. `document.innerHTML.getElementById(...)`
18. **What does npm stand for?** (1 point)
- a. Node Package Manager
 - b. Node Property Manager
 - c. New Project Manager
 - d. New Package Manager
19. **Which sign does jQuery use as a shortcut for jQuery?** (1 point)
- a. #
 - b. \$
 - c. !
 - d. %
20. **Which data binding interpolation is also known as “Mustache” syntax?** (1 point)
- a. `[...]`
 - b. `<<...>>`
 - c. `{{...}}`
 - d. `<!--...-->`
21. **In the following code snippet, what value is given for the bottom margin:** (1 point)
- ```
margin: 10px 8px 15px 9px;
```
- a. 10px
  - b. 8px
  - c. 15px
  - d. 9px
22. **What property is used to change the text color of an element?** (2 points)
- a. color
  - b. text-color
  - c. font-color
  - d. font:color
23. **Which is the correct CSS syntax?** (2 points)
- a. `a {font-weight: bold; }`
  - b. `{a:font-weight: bold; }`
  - c. `a {font:weight: bold }`
  - d. `a :font:weight = bold;`

24. **How do you change the bottom margin of an element?** (2 points)
- margin-bottom
  - bottom-margin
  - margin
  - margin:bottom
25. **What symbol indicates an HTML tag?** (2 points)
- Curved brackets e.g. {,}
  - Commas e.g. ','
  - Exclamation marks e.g. !
  - Angle brackets e.g. <,>
26. **Which directive is used to make two-way binding to data?** (2 points)
- None of these
  - v-inline
  - v-model
  - v-on
27. **Which directive is used to attach event listeners that invoke methods?** (2 points)
- v-for
  - v-on
  - v-model
  - v-bind
28. **What is a shortcut for v-bind directive?** (2 points)
- #
  - %
  - :
  - \*
29. **What is the correct way to create a new instance in vue.js?** (2 points)
- var text = new Vue({ // options })
  - var text = new vue(){ // options }
  - var text = new object({ // options })
  - var text = new text({ // options })
30. **All are life cycle hooks in Vue except ...?** (2 points)
- created
  - mounted
  - beforeMount
  - didMount
31. **Which of the following are valid keyboard events shipped with the Vue.js core?**  
(2 points)
- @keyup, @keydown
  - @inputup, inputdown
  - @keyboardup, @keyboarddown
  - @buttonup, @buttondown



32. Consider this element from a template: (3 points)

```
<input type="checkbox" v-model="done">
```

Assume the checkbox is initially unchecked and the user clicks to check it.

What is the value of “done” after checking the box?

- a. false
- b. "on"
- c. "yes"
- d. true

33. Consider the code:

```
<Example :list="data"/>
```

What is the list? (3 points)

- a. a property
- b. a data
- c. a method
- d. a watcher

34. What is the proper way to import Test module from a test package? (3 points)

- a. use Test from “test”
- b. import Test from “test”
- c. extract Test from “test”
- d. read Test from “test”

Exam 06.01.2020: Variant B

1. Syntax of style attribute of an HTML tag is? (1 point)

- a. <while>
- b. <picture>
- c. <head>
- d. <array>

2. Which of these CSS selectors will select all div tags with “example“ id? (1 point)

- a. div.example
- b. div#example
- c. #example div
- d. div .example

3. \$("span") What does it select? (1 point)

- a. All span tags
- b. First span tag
- c. Last span tag
- d. All input tags

4. What is the correct jQuery code to set the text color of all span elements to green? (1 point)

- a. \$("span").css("color", "green");
- b. \$("span").manipulate("color", "green");
- c. \$("span").style("text-color", "green");
- d. \$("span").text("color", "green");

5. `$("#div > p")` What does it select? (1 point)
  - a. All div elements with class p
  - b. All p elements that are immediate children of div element
  - c. all p elements with class div
  - d. All p elements with attribute div
6. `$("#button[intro='test']")` What does it select? (1 point)
  - a. button tag with class intro
  - b. button tag with id intro
  - c. intro tag with child element button
  - d. button element that has attribute intro with value test
7. When we need to create an unordered list which markup we need to use? (1 point)
  - a. `<li>`
  - b. `<ol>`
  - c. `<ul>`
  - d. `<nl>`
8. Which of the following table tags is used to create a table data (cell)? (1 point)
  - a. `<th>`
  - b. `<td>`
  - c. `<tr>`
  - d. `<table>`
9. Which of the following HTML tags is not valid? (1 point)
  - a. `<h1>`
  - b. `<h7>`
  - c. `<h4>`
  - d. `<h5>`
10. `<h1>` is the biggest header tag. (1 point)
  - a. True
  - b. False
11. What HTML form input should we use to present multiple options and be able to select multiple options? (1 point)
  - a. `<input type="text">`
  - b. `<input type="radio">`
  - c. `<input type="checkbox">`
12. Which is the correct syntax to use javascript file? (1 point)
  - a. `<script src="xxx.js">`
  - b. `<script href="xxx.js">`
  - c. `<script path="xxx.js">`
  - d. `<script link="xxx.js">`
13. How we can declare function "myFunction" in javascript? (1 point)
  - a. `create(myFunction){};`
  - b. `begin myFunction(){ }end;`
  - c. `function myFunction(){};`
14. Which is the correct way to use comments in HTML? (1 point)
  - a. `{# ... #}`
  - b. `<!-- .... --!>`
  - c. `// ...`
  - d. `>> ...`
15. Which is the correct way to declare an object in Javascript? (1 point)

- a. `let names = ['name' : 'James', 'age' : 21]`
  - b. `let name = { 'name' : 'James', 'age' : 21 }`
  - c. `let names = ['name' to 'James', 'age' to 21]`
  - d. `let names = { 'name' -> 'James', 'age' -> 21 }`
16. **Which is the correct?** (1 point)
- a. `i -= 1;`
  - b. `i = i--1`
  - c. `i -= 1;`
  - d. `-i-`
17. **How can we get the DOM elements with class?** (1 point)
- a. `window.getElementsByClassName(...)`
  - b. `document.getElementsByClassName(...)`
  - c. `page.getElementsByClassName(...)`
  - d. `document.innerHTML.getElementsByClassName(...)`
18. **What does npm stand for?** (1 point)
- a. Node Property Manager
  - b. New Project Manager
  - c. New Package Manager
  - d. Node Package Manager
19. **Which sign does jQuery use as a shortcut for jQuery?** (1 point)
- a. \*
  - b. \$
  - c. @
  - d. %
20. **Which of the following is correct?** (1 point)
- a. jQuery is a Javascript framework
  - b. jQuery is a Javascript library
  - c. jQuery is a PHP library
  - d. None of these
21. **In the following code snippet, what value is given for the top padding:** (1 point)
- `padding: 10px 8px 15px 9px;`
- a. 8px
  - b. 10px
  - c. 15px
  - d. 9px
22. **What property is used to change the background color of an element?** (2 points)
- a. `bg-color`
  - b. `background-color`
  - c. `background.color`
  - d. `background:color`
23. **Which is the correct CSS syntax?** (2 points)
- a. `a {font-weight: bold; }`
  - b. `{a:font-weight: bold; }`
  - c. `a {font:weight: bold }`
  - d. `a :font:weight = bold;`
24. **How do you change the left margin of an element?** (2 points)
- a. `margin-left`
  - b. `left-margin`

- c. margin-l
  - d. margin:left
25. **How to use for loop in Vue.js?** (2 points)
- a. vFor
  - b. v-loop
  - c. v-for
  - d. \*ngFor
26. **Which directive is used to make two-way binding to data?** (2 points)
- a. None of these
  - b. v-inline
  - c. v-model
  - d. v-on
27. **Which directive is used to make two-way binding to props?** (2 points)
- a. v-for
  - b. v-on
  - c. v-model
  - d. v-bind
28. **What is a shortcut for v-on directive?** (2 points)
- a. #
  - b. @
  - c. :
  - d. \*
29. **What is the correct way to create new instance in vue.js?** (2 points)
- a. var text = new Vue({ // options })
  - b. var text = new vue(){ // options }
  - c. var text = new object({ // options })
  - d. var text = new text({ // options })
30. **All are life cycle hooks in Vue except ...?** (2 points)
- a. created
  - b. mounted
  - c. beforeMount
  - d. didCreate
31. **Which of the following are valid mouse events shipped with the Vue.js core?** (2 points)
- a. @mousedown, @mouseup
  - b. @pointerup, @pointerdown
  - c. @cursorup, @cursordown
32. **What is axios package responsible for?** (3 points)
- a. Listening to events
  - b. Rendering HTML
  - c. Connecting to database
  - d. Making HTTP requests
33. **Consider the code:**
- ```
<div v-for="[1, 2, 3] as number">{{ number }}</div>
```
- How many divs will be printed as a result of this loop?** (3 points)
- a. 1
 - b. 2
 - c. 3
 - d. None, error will be thrown

34. Consider the following code:

```
<h4>{{ name | uppercase }}</h4>
```

What is “uppercase”? (3 points)

- a. property
- b. method
- c. filter
- d. watcher

Exam 13.01.2020: Variant A

1. **Syntax of id attribute of an HTML tag is?** (1 point)

- a. <tagname="id">
- b. <tagname id="test">
- c. <id="tagname;">
- d. <tagname #id="test">

2. **Which of these CSS selectors will select all div tags with “example“ id?** (1 point)

- a. div.example
- b. div#example
- c. #example div
- d. div .example

3. **\$("#p") What does it select?** (1 point)

- a. All p tags
- b. First p tag
- c. Last p tag
- d. All div tags

4. **What is the correct jQuery code to set the text color of all span elements to green?** (1 point)

- a. \$("span").css("color", "green");
- b. \$("span").manipulate("color", "green");
- c. \$("span").style("text-color", "green");
- d. \$("span").text("color", "green");

5. **\$("#div p") What does it select?** (1 point)

- a. All div elements with class p
- b. All p elements that are inside div element
- c. All p elements with id div
- d. All p elements with class div

6. **\$("#button[intro='test']") What does it select?** (1 point)

- a. button tag with class intro
- b. button tag with id intro
- c. intro tag with child element button
- d. button element that has attribute intro with value test

7. **When we need to create an ordered list which markup we need to use?** (1 point)

- a.
 - b.
 - c.
 - d. <nl>
8. Which of the following table tags is used to create a table header cell? (1 point)
- a. <th>
 - b. <td>
 - c. <tr>
 - d. <table>
9. Which of the following HTML tags is not valid? (1 point)
- a. <h1>
 - b. <h8>
 - c. <h4>
 - d. <h5>
10. <h1> is the biggest header tag. (1 point)
- a. True
 - b. False
11. What HTML form input should we use for users to enter text information? (1 point)
- a. <input type="text">
 - b. <input type="radio">
 - c. <input type="checkbox">
12. Which is the correct syntax to use a javascript file? (1 point)
- a. <script src="xxx.js">
 - b. <script href="xxx.js">
 - c. <script path="xxx.js">
 - d. <script link="xxx.js">
13. How can we declare the function “myFunction” in javascript? (1 point)
- a. create(myFunction){};
 - b. begin myFunction(){ }end;
 - c. function myFunction(){};
14. Which is the correct way to use comments in CSS? (1 point)
- a. {# ... #}
 - b. <!-- ... --!>
 - c. /* ... */
 - d. // ...
15. Which is the correct way to declare an array in Javascript? (1 point)
- a. let names = ['John', 'James', 'Bob']
 - b. let names = array('John', 'James', 'Bob')
 - c. let name = 'John', 'James', 'Bob'
 - d. let names = [0]=>'John', [1]=>'James', [2]=>'Bob'
16. Which is correct? (1 point)

- a. `i += 1;`
- b. `i += 1;`
- c. `i = i++1`
- d. `+i+`

17. **How can we get the DOM element with a tag name?** (1 point)

- a. `window.getElementsByTagName(...)`
- b. `document.getElementsByTagName(...)`
- c. `page.getElementsByTagName(...)`
- d. `document.getElementByTagName(...)`

18. **What is the newest version of HTML being developed?** (1 point)

- a. HTML 4
- b. HTML 4.1
- c. HTML 5
- d. HTML doesn't have versions

19. **What does CSS stand for?** (1 point)

- a. Cascading Style Sheet
- b. Complex Styled Syntax
- c. Complete Source Solution

20. **Which data binding interpolation is also known as “Mustache” syntax?** (1 point)

- a. `[...]`
- b. `<<...>>`
- c. `{{...}}`
- d. `<!--...-->`

21. **In the following code snippet, what value is given for the bottom margin:** (1 point)

```
margin: 10px 8px 15px 9px;
```

- a. 10px
- b. 8px
- c. 15px
- d. 9px

22. **What property is used to change the background color of an element?** (2 points)

- a. `bg-color`
- b. `background-color`
- c. `background.color`
- d. `background:color`

23. **Which is the correct CSS syntax?** (2 points)

- a. `a {align: center;}`
- b. `a {align-center;}`
- c. `a {text-align: center;}`
- d. `a {text: center;}`

24. **How do you change the bottom margin of an element?** (2 points)

- a. margin-bottom
 - b. bottom-margin
 - c. margin
 - d. margin:bottom
25. **What symbol indicates an HTML tag?** (2 points)
- a. Curved brackets e.g. {,}
 - b. Commas e.g. ','
 - c. Exclamation marks e.g. !
 - d. Angle brackets e.g. <,>
26. **Which directive is used to make two-way binding to data?** (2 points)
- a. None of these
 - b. v-inline
 - c. v-model
 - d. v-on
27. **Which directive is used to attach event listeners that invoke methods?** (2 points)
- a. v-for
 - b. v-on
 - c. v-model
 - d. v-bind
28. **What is a shortcut for v-bind directive?** (2 points)
- a. #
 - b. %
 - c. :
 - d. *
29. **What is the correct way to create a new instance in vue.js?** (2 points)
- a. `var text = new Vue({ // options })`
 - b. `var text = new vue(){ // options }`
 - c. `var text = new object({ // options })`
 - d. `var text = new text({ // options })`
30. **All are life cycle hooks in Vue except ...?** (2 points)
- a. created
 - b. mounted
 - c. beforeMount
 - d. didMount
31. **Which of the following are valid mouse events shipped with the Vue.js core?** (2 points)
- a. @mousedown, @mouseup
 - b. @pointerup, @pointerdown
 - c. @cursorup, @cursordown
32. **What is the axios package responsible for?** (3 points)
- a. Listening to events
 - b. Rendering HTML

- c. Connecting to database
- d. Making HTTP requests

33. Consider the code:

```
<Example :list="data"/>
```

What is the list? (3 points)

- a. a property
 - b. a data
 - c. a method
 - d. a watcher
- 34. What is the proper way to import Test module from a test package?** (3 points)
- a. use Test from “test”
 - b. import Test from “test”
 - c. extract Test from “test”
 - d. read Test from “test”

Exam 13.01.2020: Variant B

1. **Syntax of id attribute of an HTML tag is?** (1 point)
 - a. <tagname="id">
 - b. <tagname id="test">
 - c. <id="tagname;">
 - d. <tagname #id="test">
2. **Which of these CSS selectors will select all div tags with “example” id?** (1 point)
 - a. div.example
 - b. div#example
 - c. #example div
 - d. div .example
3. **\$("p")** **What does it select?** (1 point)
 - a. All p tags
 - b. First p tag
 - c. Last p tag
 - d. All div tags
4. **What is the correct jQuery code to set the text color of all span elements to green?** (1 point)
 - a. \$("span").css("color", "green");
 - b. \$("span").manipulate("color", "green");
 - c. \$("span").style("text-color", "green");
 - d. \$("span").text("color", "green");

5. `$("#div p")` What does it select? (1 point)
- All div elements with class p
 - All p elements that are inside div element
 - All p elements with id div
 - All p elements with class div
6. `$("#button[intro='test']")` What does it select? (1 point)
- button tag with class intro
 - button tag with id intro
 - intro tag with child element button
 - button element that has attribute intro with value test
7. When we need to create ordered list which markup we need to use? (1 point)
- ``
 - ``
 - ``
 - `<nl>`
8. Which of the following table tags is used to create a table data (cell)? (1 point)
- `<th>`
 - `<td>`
 - `<tr>`
 - `<table>`
9. Which of the following HTML tags is valid? (1 point)
- `<h6>`
 - `<h7>`
 - `<h8>`
 - `<h9>`
10. `<h1>` is the biggest header tag. (1 point)
- True
 - False
11. What HTML form input should we use for user to enter text information? (1 point)
- `<input type="text">`
 - `<input type="radio">`
 - `<input type="checkbox">`
12. Which is the correct syntax to use javascript file? (1 point)
- `<script src="xxx.js">`
 - `<script href="xxx.js">`
 - `<script path="xxx.js">`
 - `<script link="xxx.js">`
13. How we can declare function "myFunction" in javascript? (1 point)
- `create(myFunction){};`
 - `begin myFunction(){ }end;`
 - `function myFunction(){};`

14. **Which is the correct way to use comments in HTML?** (1 point)
- `{# ... #}`
 - `<!-- ... --!>`
 - `// ...`
 - `>> ...`
15. **Which is the correct way to declare an array in Javascript?** (1 point)
- `let names = ['John', 'James', 'Bob']`
 - `let names = array('John', 'James', 'Bob')`
 - `let name = 'John', 'James', 'Bob'`
 - `let names = [0]=>'John', [1]=>'James', [2]=>'Bob'`
16. **Which is the correct?** (1 point)
- `i -= 1;`
 - `i = i--1`
 - `i -= 1;`
 - `-i-`
17. **How can we get the DOM element with tag name?** (1 point)
- `window.getElementsByTagName(...)`
 - `document.getElementsByTagName(...)`
 - `page.getElementsByTagName(...)`
 - `document.getElementByTagName(...)`
18. **What does npm stand for?** (1 point)
- Node Property Manager
 - New Project Manager
 - New Package Manager
 - Node Package Manager
19. **Which sign does jQuery use as a shortcut for jQuery?** (1 point)
- *
 - \$
 - @
 - %
20. **Which of the following is correct?** (1 point)
- jQuery is a Javascript framework
 - jQuery is a Javascript library
 - jQuery is a PHP library
 - None of these
21. **In the following code snippet, what value is given for the top padding:** (1 point)
- ```
padding: 10px 8px 15px 9px;
```
- 8px
  - 10px
  - 15px
  - 9px

22. Which data binding interpolation is also known as “Mustache” syntax? (1 point)
- a. [...]
  - b. <<...>>
  - c. {{...}}
  - d. <!--...-->
23. Which is the correct CSS syntax? (2 points)
- a. a {font-weight: bold; }
  - b. {a:font-weight: bold; }
  - c. a {font:weight: bold }
  - d. a :font:weight = bold;
24. How do you change the left margin of an element? (2 points)
- a. margin-left
  - b. left-margin
  - c. margin-l
  - d. margin:left
25. How to use for loop in Vue.js? (2 points)
- a. vFor
  - b. v-loop
  - c. v-for
  - d. \*ngFor
26. Which directive is used to make two-way binding to data? (2 points)
- a. None of these
  - b. v-inline
  - c. v-model
  - d. v-on
27. Which directive is used to make two-way binding to props? (2 points)
- a. v-for
  - b. v-on
  - c. v-model
  - d. v-bind
28. What is a shortcut for v-on directive? (2 points)
- a. #
  - b. @
  - c. :
  - d. \*
29. What is the correct way to create new instance in vue.js? (2 points)
- a. var text = new Vue({ // options })
  - b. var text = new vue(){ // options }
  - c. var text = new object({ // options })
  - d. var text = new text({ // options })
30. All are life cycle hooks in Vue except ...? (2 points)
- a. created

- b. mounted
- c. beforeMount
- d. didCreate

31. **Which of the following are valid keyboard events shipped with the Vue.js core?**

*(2 points)*

- a. @keyup, @keydown
- b. @inputup, inputdown
- c. @keyboardup, @keyboardevent
- d. @buttonup, @buttondown

32. **Consider this element from a template: (3 points)**

```
<input type="checkbox" v-model="done">
```

**Assume the checkbox is initially unchecked and the user clicks to check it.**

**What is the value of “done” after checking the box?**

- a. false
- b. "on"
- c. "yes"
- d. true

33. **Consider the code:**

```
<div v-for="number in [1, 2, 3]">{{ number }}</div>
```

**How many divs will be printed as a result of this loop? (3 points)**

- a. 1
- b. 2
- c. 3
- d. None, error will be thrown

34. **What is the proper way to import Test module from test package? (3 points)**

- a. use Test from “test”
- b. import Test from “test”
- c. extract Test from “test”
- d. read Test from “test”

## II. Feedback Questions

1. The structure of the course supported my learning.

Choice answer	Amount	%
I agree (4)	66	54%
I somewhat agree (3)	41	33%
I somewhat disagree (2)	10	8%
I disagree (1)	3	2%
Not applicable (0)	1	0%

2. I understood what and how I am expected to learn on this course.

Choice answer	Amount	%
I agree (4)	87	71%
I somewhat agree (3)	29	23%
I somewhat disagree (2)	3	2%
I disagree (1)	1	0%
Not applicable (0)	1	0%

3. The teaching was varied (different kinds of methods and tasks were employed).

Choice answer	Amount	%
I agree (4)	58	47%
I somewhat agree (3)	43	35%
I somewhat disagree (2)	16	13%
I disagree (1)	3	2%
Not applicable (0)	1	0%

4. Students were given the chance to discuss the subject matter.

Choice answer	Amount	%
I agree (4)	69	57%
I somewhat agree (3)	37	30%
I somewhat disagree (2)	11	9%
I disagree (1)	2	1%
Not applicable (0)	2	1%

5. The course was intellectually challenging.

Choice answer	Amount	%
I agree (4)	50	41%
I somewhat agree (3)	36	29%
I somewhat disagree (2)	21	17%
I disagree (1)	12	9%
Not applicable (0)	2	1%

6. The feedback helped me to understand which knowledge and/or what skills I should develop further.

Choice answer	Amount	%
I agree (4)	56	46%
I somewhat agree (3)	36	29%
I somewhat disagree (2)	13	10%
I disagree (1)	7	5%
Not applicable (0)	9	7%

7. The assessment was closely related to the teaching.

Choice answer	Amount	%
I agree (4)	86	71%
I somewhat agree (3)	33	27%
I somewhat disagree (2)	1	0%
I disagree (1)	0	0%
Not applicable (0)	1	0%

8. Please evaluate how well your workload matched the number of ects credits in this course (1 ects credit = 26 hours of studying).

Choice answer	Amount	%
Workload is too low (1)	47	38%
Workload is adequate (2)	73	60%
Workload is too high (3)	0	0%
Not valid (0)	1	0%

9. All in all, the course was valuable for me.



Choice answer	Amount	%
I agree (4)	68	56%
I somewhat agree (3)	38	31%
I somewhat disagree (2)	9	7%
I disagree (1)	5	4%
Not applicable (0)	1	0%

### **III. License**

#### **Non-exclusive licence to reproduce thesis and make thesis public**

I, Tsothe Kekelia

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, Course on Web-Application Development in Software Engineering supervised by Dietmar Pfahl, PhD

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

*Tsothe Kekelia*

*16/06/2020*