TALLINN UNIVERSITY OF TECHNOLOGY
Department of Software Science

Pjotr Surkov 220111IAIB

Serverless cloud-based cross-platform learning management system

Bachelor's thesis

Supervisor: Jaak Henno
PhD (Mathematics)

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Tarkvarateaduse instituut

Pjotr Surkov 220111IAIB

Serverivaba pilvepõhine platvormist sõltumatu õpihaldussüsteem

Bakalaureusetöö

Juhendaja: Jaak Henno
PhD (Mathematics)

Tallinn 2022

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Pjotr Surkov

10.01.2022

Abstract

The goal of this thesis is to create a high-quality base (MVP) for LMS based on market demand and focusing on optimization for **change first**, not features or technologies. However, despite this focus, we will still use the latest technologies and good practices in design, code, development process, etc.

Technologies and features might change every day, so, it’s more important to create an adaptable product first. The best product is not the one that uses the best tech stack and has the best features, but the one that **satisfies the market needs** best.

If we think of LEGO, then we will focus on creating high-quality pieces which can be used to easily create different models, not a single solid one. Every part in our MVP will be **modular**, **expandable** and **replaceable**.

We will also compare ourselves with the key competitors and try to make our solution **significantly better** technologically and functionally than others.

As a result, we will have an extendable **game-based** LMS MVP [1]. Its core features:

- Create, import and manage courses (quizzes).
- Test and assess student skills using gamification.
- Review and export statistics.

Source code: https://github.com/Perpetiel/fast-exam

Demo: https://fast-exam.web.app

This thesis is written in English and is 47 pages long, including 22 chapters, 29 figures and 1 table.

Annotatsioon

Selle lõputöö eesmärk on luua kvaliteetne baas (MVP) õpihaldussüsteemiks, mis põhineb turunõudlustel ja keskendub esmalt muudatuste optimeerimisele, mitte funktsioonidele või tehnoloogiatele. Kuid siiski kasutame disainis, koodis, arendusprotsessis jne uusimaid tehnoloogiaid ja häid tavasid.

Tehnoloogiad ja funktsioonid võivad muutuda iga päev, seega on olulisem esmalt luua kohandatav toode. Parim toode ei ole see, mis kasutab parimat tehnoloogiat ja millel on parim funktsionaalsus, vaid see, mis rahuldab kõige paremini turu vajadusi.

Kui mõtleme LEGO-le, siis keskendume kvaliteetsete tükkide loomisele, millest saab luua erinevaid mudeleid, mitte ühte kindlat. Kõik meie mängu osad on modulaarsed, laiendatavad ja vahetatavad.

Lisaks sellele, võrdleme end ka võtmekonkurentidega ning püüame muuta oma lahenduse tehnoloogiliselt ja funktsionaalselt teistest oluliselt paremaks.

Selle tulemusena saame laiendatava mängupõhise LMS-i MVP. Selle põhiomadused:

- Luua, importida ja hallata kursusi (viktoriinid).
- Testida ja hinnata õpilaste oskusi mängulisuse abil.
- Vaadata ja eksportida tulemusi.

Lähtekood: https://github.com/Perpetiel/fast-exam

Demo: https://fast-exam.web.app

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 47 leheküljel, 22 peatükki, 29 joonist, 1 tabel.

List of abbreviations and terms

a11y	Accessibility
AI	Artificial Intelligence
API	Application Programming Interface
CLI	Command Line Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
FP	Functional Programming
HTML	HyperText Markup Language
i18n	Internationalization
IT	Information Technology
JS	JavaScript
KB	Kilobyte
LEAN	Methodology
LEGO	Construction Set
LMS	Learning Management System
LTI	Learning Tools Interoperability - specification
MVP	Minimum Viable Product
OOP	Object-Oriented Programming
PICK	Possible, Implement, Challenge, Kill
PWA	Progressive Web Application
SCORM	Sharable Content Object Reference Model - specifications
SQL	Structured Query Language
SVG	Scalable Vector Graphics
UI	User Interface
UX	User Experience

Table of contents

List of figures

List of tables

1 Introduction

Who of us has never thought of making some learning process more fun or teaching easier? All our life we learn and teach.

When we're young: at school, at university, at work – we always try to improve the ways we learn things. When we're older we get children, new co-workers, maybe a business and now we have to teach others and we try our best to do so. Sometimes it's effective, sometimes not.

There are many tools out there which can help to make our learning and teaching processes more productive. Let's have an overview of some of them, the most popular ones, and see how could we improve them by using the latest technologies in IT.

Let's start with checking some benefits of using digital learning tools over traditional education methods. Based on the study released by McGraw - Hill Education [2]:

- 81% of students say digital learning technology is helping them **boost their grades**.
- 97% students found adaptive learning technology helpful in **retention**.
- 71% students **engage** more with digital course material.
- 84% students say digital will **enhance knowledge**.
- 78% college students confess that digital learning is helping them improve computer science, mathematics, and other **technical skills**.
- 45% students want to learn on **personal** device.
- 79% students prefer **online** quiz, adaptive learning & e-Textbooks in learning.

2 Fast Exam – Learn and teach in the modern way!

One of the ways to improve learning and teaching is to use different tests, and one type of the digital educational tools what can be used for this is quiz apps. Based on the survey we have seen before; 80% students would be happy to use them. And while there are many good quiz apps, most of them are outdated or not so good for both, students and teachers. It's actually quite hard to find a good quiz app. With recent pandemic situation in the world it's even more beneficial to have some good digital tools in our life.

So, let's build the best quiz app out there! And not just quiz app, but multiplayer quiz game app! And not just multiplayer, but **serverless multiplayer quiz game app**!

It will be our base for LMS! Why Fast Exam?

Exam is short for examination, and examination is a formal **test of a person's knowledge** or proficiency in a subject or skill (Google dictionary). And **Fast** means that it allows you do this more **quickly**. Students can perform tests and teachers conduct assessments more swiftly.

To build a good app the first important thing we should do, before even doing something, is to choose the right technologies. The good tech stack can be crucial in building a successful app. Many quiz apps have different limitations just because they are using not the best tech stacks. But before that, let's answer some technical questions to make it easier:

- What features should this app have?
- What user interface should this app have?
- What database should we use?
- What backend technologies should we use?
- What frontend technologies should we use?

2.1 Fast Exam – Feature requirements

- It should be possible to **upload** questions using some file (e.g. spreadsheet) import feature. In this way teachers will not need to manually enter questions. It will also allow to re-use questions in other quiz apps (or re-use existing ones), like Kahoot! or Moodle's Quiz module which have these kinds of import and export features.
- It should be possible to **download** results as a spreadsheet file as well.
- Teachers should be able to do everything by themselves, **without administrators**.
- It should have a good **authentication** and authorization system with different auth methods, as we have several user roles with various **access rights**.
- It should be possible to use **different quizzes** and activate them.
- Both, **single player** and **multiplayer** modes should be available for students.
- Number of questions, answer options, players, points, etc **shouldn't be limited**.
- It can't have fewer extra features than other quiz apps: **power ups**, **images**, etc.
- It should be possible to create different **game types** on top of this quiz app.

2.2 Fast Exam – User Interface (UI) requirements

- Game interface should be in accordance with feature requirements. If we check other apps, e.g. Kahoot! [3] or Trivia Crack [4]:
  - Kahoot! quiz game has only 4 buttons to answer a question. This limits the number of available options.
  - Trivia Crack is designed for 2 players only. It's not possible to get all students to participate in the same game.
- All possible platforms should be supported (cross-platform):
  - Web, Android, iOS, Windows, Mac, Linux, etc
- All possible devices should be supported (responsive design):

    - Desktops, tablets, phone, etc

2.3 Fast Exam – Database requirements

- It should allow to build truly serverless app – everything should work without client-server communication.
- Necessary data should be synced across all clients in real-time – we're creating a multiplayer game.
- It should have offline support – it's an educational app and can be used everywhere (this is not implemented in prototype).

2.4 Fast Exam – Backend and Frontend requirements

Based on features, design and database requirements one of the best tech stack could be:

- Frontend: **Vue.js** [5] (HTML, CSS and JavaScript)
- Backend: **Firebase** [6] [7] [8] [9] [10] [11] [12] [13]

Even without thinking many developers nowadays (September 2021) would choose these two technologies: Vue.js is the most popular frontend framework and Firebase is the most popular backend platform for mobile and web applications.

In 2020 there were over 2.5 million monthly active Firebase apps, which included global businesses like Gameloft and Alibaba [14].

In 2021 Vue.js became Wikimedia Foundation's official choice [15].

3 Fast Exam – Tech Stack

Fast Exam is written using HTML, CSS and JavaScript. However, several other technologies have been used to improve development process and product itself. The used tech stack gave us many benefits over existing solutions (competitors).

3.1 TypeScript vs JavaScript

TypeScript [16] offers all of JavaScript's features, plus type system. The main benefit of TypeScript is that it can highlight unexpected behaviour in your code, lowering the chance of bugs.

Today JavaScript is being used almost everywhere and applications are more complicated than in past years, so, adding types drastically reduces the complexity of making improvements to those apps. TypeScript helps developers feel more confident in their code, and save considerable amounts time in validating that the code works as expected.

3.2 noCSS (Bulma) vs CSS

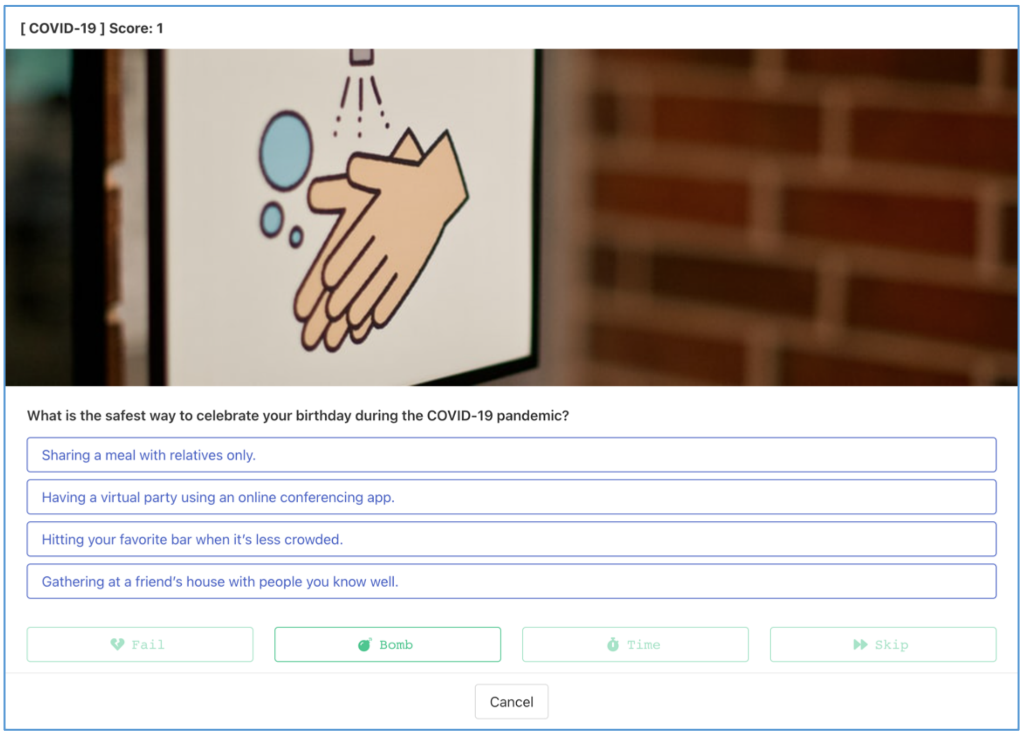
Traditionally, whenever you need to style something on the web, you write CSS. With Bulma [17] [18], you style elements by applying pre-existing classes directly in your HTML.

This approach (utility-first) allows us to implement a completely custom component design without writing a single line of custom CSS!

3.2.1 Bulma Card component

Card [19] is an all-around flexible and composable component. It's used basically in all components and views: Login, SignUp, Upload, Game, Question. (Except Statistics view which utilizes a table component). Card parts:

- Header, including Title and Icon
- Image
- Content
- Footer, including Item


[COVID-19] Score: 1

What is the safest way to celebrate your birthday during the COVID-19 pandemic?

Sharing a meal with relatives only.

Having a virtual party using an online conferencing app.

Hitting your favorite bar when it's less crowded.

Gathering at a friend's house with people you know well.

Figure 1. Card example.

3.2.2 Bulma extensions

Bulma extensions [20] are side projects, two extensions are used in Fast Exam:

- Toast
- Tooltip

3.3 Font Awesome

Font Awesome [21] is the web's most popular icon set and toolkit. It might feel that it is not so important at first, but without icons it's impossible to create a good design for small screen devices. Font Awesome provides two types of icons. Fast Exam uses SVG icons what is a more better option than other type, Web Fonts [22].

3.4 Firebase

Firebase is a platform developed by Google for creating mobile and web applications.

3.4.1 Firestore (Database)

Cloud Firestore [23] is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps at global scale.

3.4.2 Functions (Backend)

Cloud Functions [24] for Firebase allow you to develop a backend and run your code without managing servers.

3.4.3 Authentication (Auth)

Firebase Authentication [25] is simple, free multi-platform sign-in.

3.4.4 Hosting (Publishing)

Firebase Hosting [26] provides you with a fast and secure web hosting.

3.5 Vue.js (Frontend)

Vue.js is a progressive framework for building user interfaces and is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, but Vue is also perfectly capable of powering sophisticated single-page applications when used in combination with modern tooling and supporting libraries.

3.5.1 Vue Router

Vur Router [27] is the official router for Vue.js.

3.5.2 Vuex

Vuex [28] is a state management pattern + library for Vue.js applications. It serves as a centralized store for all the components in an application, with rules ensuring that the state can only be mutated in a predictable fashion.

3.5.3 Vue CLI

Vue CLI [29] is a standard Tooling for Vue.js Development.

3.5.4 Vue Devtools

Vue Devtools [30] is a browser devtools extension for debugging Vue.js applications.

3.5.5 Vue-Screen

Vue-Screen [31] is reactive screen size and media query [32] states for Vue.

3.6 SheetJS

SheetJS [33] is a parser and writer for various spreadsheet formats.

3.7 PWA and Electron

Fast Exam is a progressive web app (PWA) [34], but can be converted to a desktop app using Electron [35].

Electron allows you to build cross-platform desktop apps with web technologies.

If you can build a website, you can build a desktop app. Electron is a framework for creating native applications with web technologies like JavaScript, HTML, and CSS. It takes care of the hard parts so you can focus on the core of your application.

```
vue add electron-builder
npm run electron:build
```

Figure 2. Desktop app with 2 commands [36].

4 Fast Exam – LEAN

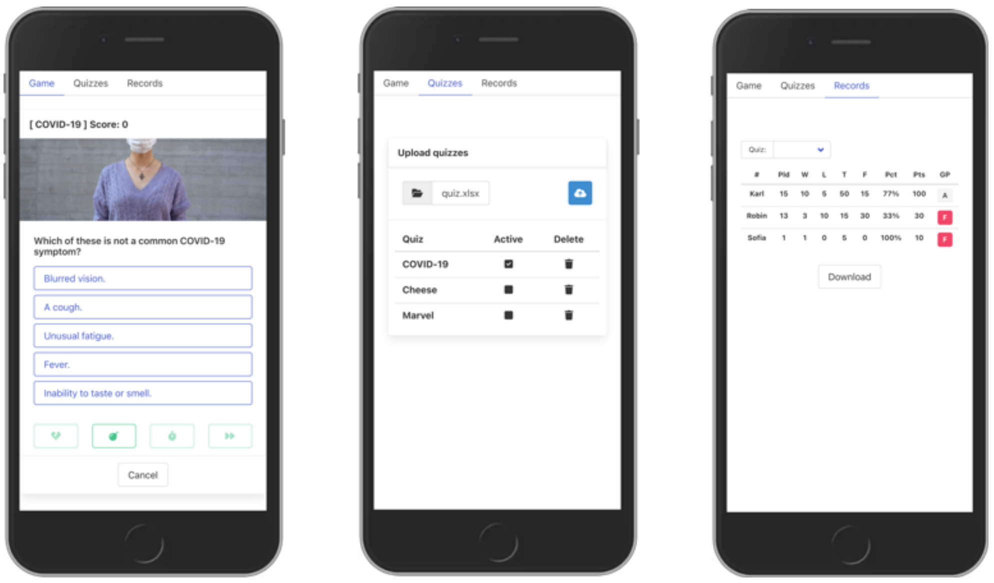
Fast Exam is developed using LEAN good practices: only Minimum Viable Product (MVP) [37] is created and only the core functionality is implemented [38] [Appendix 2].

I's always nice to keep in mind Jidoka and Poka-Yoke techniques when we create a feature or use Ishikawa diagrams and PICK charts to make development faster and easier.

Fast Exam is also optimized for change first while having a good performance as well [Appendix 3].

5 Fast Exam – Mobile-First Design

Fast Exam is implemented for mobile devices first [39], and works well on many devices.



[COVID-19] Score: 0

Which of these is not a common COVID-19 symptom?

- Blurred vision.
- A cough.
- Unusual fatigue.
- Fever.
- Inability to taste or smell.

Upload quizzes

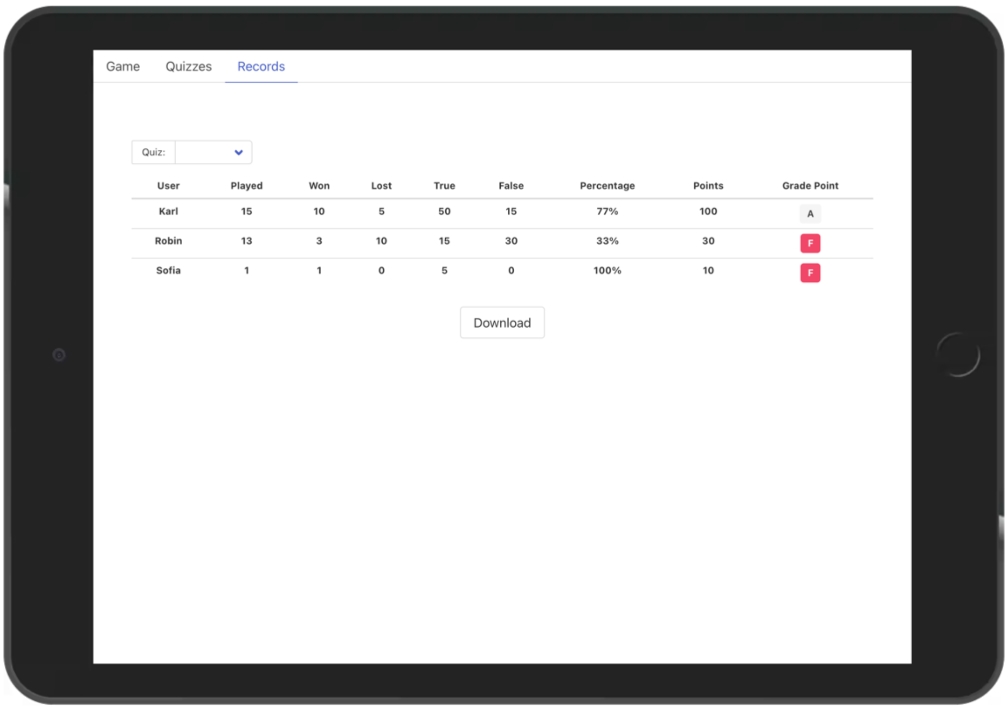
quiz.xlsx

Quiz	Active	Delete
COVID-19	☑	🗑
Cheese	■	🗑
Marvel	■	🗑

Quiz:

#	Pld	W	L	T	F	Pct	Pts	GP
Karl	15	10	5	50	15	77%	100	A
Robin	13	3	10	15	30	33%	30	F
Sofia	1	1	0	5	0	100%	10	F

Figure 3. Mobile views.



Quiz:

User	Played	Won	Lost	True	False	Percentage	Points	Grade Point
Karl	15	10	5	50	15	77%	100	A
Robin	13	3	10	15	30	33%	30	F
Sofia	1	1	0	5	0	100%	10	F

Figure 4. Tablet view.

6 Fast Exam – Clean code

While the tech stack is one of the most important parts of the software design, using this stack the right way during software development is not less important. Let's have a quick review of the good practices which were used during Fast Exam development.

6.1 Functional programming vs object-oriented programming

During my studies at the university (2010 - 2013), OOP was the main way to write software and it had replaced FP what was before it. Nowadays FP is back and is considered as a better option [40]. Fast Exam is written using FP as well while the main competitors: Moodle and Kahoot! are developed using OOP (PHP and Java) what is a big disadvantage.

Like OOP, FP languages supports all 4 concepts:

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

6.2 Software development principles

Here is the list of a good practices every developer should follow nowadays in order to write high quality software [41]:

- **DRY** - Don't Repeat Yourself

Fast Exam is written using small and independent units – Vue composables and SFC.

- **KISS** – Keep It Simple Stupid

Fast Exam is written using the latest JavaScript version (ES6).

- **SOLID**

1. Every component in Fast Exam has single responsibility (Single Responsibility Principle).
2. Every component in Fast Exam is extendable (Open-Closed Principle).
3. Fast Exam is written using TypeScript which allows to create generic functions (Liskov Substitution Principle).
4. Every component in Fast Exam specifies a set of required and optional properties (Interface Segregation Principle).
5. Function (Composable) dependencies are passed in as parameters (Dependency Inversion Principle).

- **YAGNI** – You aren't gonna need it

Only necessary features are implemented in Fast Exam prototype.

- **TDD** – Test Driven Development

Vue Test Utils – the official unit testing utility library for Vue.js allows to write TDD code.

6.3 Tree-shaking

All libraries used by Fast Exam support modularity:

6.3.1 Firebase [42]

```
import { signInWithEmailAndPassword } from 'firebase/auth';
```

Figure 5. Firebase tree-shaking.

6.3.2 Vue.js [43]

```
import { computed, defineComponent, ref } from 'vue';
import { useGrid } from 'vue-screen';
import { useRouter } from 'vue-router';
import { useStore } from 'vuex';
```

Figure 6. Vue.js tree-shaking.

6.3.3 Bulma [44]

```
@import '~bulma/sass/components/card.sass';
@import '~bulma/sass/components/tabs.sass';
```

Figure 7. Bulma tree-shaking.

6.3.4 Font Awesome [45]

```
import {
  faBomb,
  faForward,
  faHeartBroken,
  faStopwatch,
} from '@fortawesome/free-solid-svg-icons';
```

Figure 8. Font Awesome tree-shaking.

7 Fast Exam vs Kahoot!

Kahoot! is a game-based learning platform, used as educational technology in schools and other educational institutions. Its learning games, "kahoots", are user-generated multiple-choice quizzes that can be accessed via a web browser or the Kahoot! app. Kahoot! can be used to review students' knowledge, for formative assessment, or as a break from traditional classroom activities.

7.1 Kahoot! question types [46]

Test knowledge

- Quiz
- True or false
- Type Answer (Premium feature)
- Puzzle (Premium feature)
- Quiz + Audio (Premium feature)

Collect opinions

- Poll (Premium feature)
- Word cloud (Premium feature)
- Open-ended (Premium feature)
- Brainstorm (Premium feature)

Add slide

- Slide (Premium feature)

As we can see only “Quiz” and “True or false” question types are available for free.

For both question types:

- Questions can be up to 120 characters.
- Time limit can range from 5 seconds to 4 minutes.
- Points can be toggled to 0, 1000, or 2000 points.
- Points are awarded based on speed of answer.
- You can add an image or a YouTube video.

Quiz:

- Answers can be text up to 75 characters or an image.
- At least 2 answers are required (4 max).
- At least one must be correct.

True or false:

- Answers are “True” or “False” options and cannot be edited.
- Only one can be correct.

7.2 Fast Exam questions

Fast Exam prototype has only one question type, but that’s enough to create both quiz and true or false questions. There is no need to have different question types.

- No limits for questions – you can write a poem or nothing at all (why not?)
- No limits for answers – you can have one (why not?) or hundred answers
- No limits for timers (no time in prototype)
- No limits for points – custom formula can be used (hard-coded in prototype)
- No limits for correct answers – all answers can be wrong or correct (why not?)
- Images can be added
- Power Ups can be configured and used!

NB! In future there will be no correct answers. Every answer will have the percentage value (0 - 100) of the correctness. E.g.:

- 0 – wrong answer.
- 50 – half-correct answer.
- 100 – correct answer.

7.3 Fast Exam Power Ups

Power Ups are tools that will make it easier for you to answer the questions correctly. Many trivia games have this feature, but not Kahoot!

- **Fail**: get a second shot of guessing the answer.
- **Bomb**: discard 2 wrong answers.
- **Time**: get additional time to answer.
- **Skip**: get a different question.

7.4 Spreadsheets

Both, Fast Exam and Kahoot! allows you to upload questions from a spreadsheet. However, Kahoot! added this feature in 2018 – 6 years after initial release and spreadsheet template has several restrictions [47]:

- Importing questions is for the Quiz game type only!
- Questions have a limit of 95 characters.
- Answers have a limit of 60 characters.
- Each question must have at least two answers.
- Allowable time limit (sec) values are: 5, 10, 20, 30, 60, or 120.

Want to create a "True or false" question or add image? It's not possible.

Fast Exam has no restrictions. Moreover, spreadsheet is the only way to add questions. We don't waste our time on implementing old school features.

8 Fast Exam vs Moodle and Hot Potatoes

Moodle [48] is the world's most popular learning management system and #1 in Estonia. The main issue with this platform that many actions can't be done without administrators. For example, to create a quiz you will need to create a Moodle course before.

To open a new Moodle course, you will need to send an e-mail to moodle@taltech.ee and include following information [49]:

- Course code, title and language
- Teacher name and faculty

Then you can create a quiz [50]. Quizzes can be used in Moodle using the Quiz activity [51], but it is also possible to use quizzes made in Hot Potatoes (which can be imported to the course).

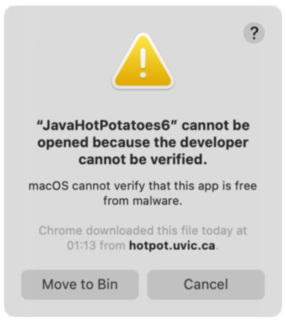
The Quiz activity enables to create 16 types of questions, which may contain not only text but also images, audio and video files, links, etc. Most questions are automatically graded (except for the essay), but it is always possible to reassess the answers, if necessary. It is not possible to take a Moodle quiz anonymously – all results are saved and visible to the teacher.

Fast Exam eliminates the need of administrators and has no limitations what questions you can have or what way assessments are done.

If we assume that we decided to use Moodle, we will also need to create quiz questions somehow. The most popular way to create questions is to use Hot Potatoes [52] and then import them to Moodle. If we use the most popular OS among developers, macOS, then:

- For Mac users, there is a Java version of Hot Potatoes 6. The Java version also runs on Linux and Windows.
- The Java version provides all the features found in the windows version 6, except: you can't export a SCORM object from Java Hot Potatoes. You can create data files in Java Hot Potatoes and open those in Windows Hot Potatoes for upload to hotpotatoes.net or export to SCORM (used in Moodle import).

This means that we will still need Windows to export questions, moreover with the latest macOS you will not be able to run the program and create or update questions:



"JavaHotPotatoes6" cannot be opened because the developer cannot be verified.

macOS cannot verify that this app is free from malware.

Chrome downloaded this file today at 01:13 from **hotpot.uvic.ca**.

Figure 9. Hot Potatoes error.

9 Implementation – Navigation

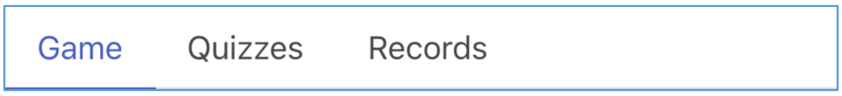


Figure 10. Navigation Tabs.

Fast Exam has 3 views (pages):

- **Game**, where you can take a quiz.
- **Quizzes**, where you can upload, activate and delete quizzes.
- **Records**, where you can see and download statistics.

Before seeing the content of any of these views, user should be logged in.

9.1 Tabs

Based on the modern UX principles: users should be able to access as much of the app's content with as few taps as possible. So far, the best navigation solution for mobile apps is a tab bar [53].

10 Implementation – Sign in and sign up

Fast Exam implements 'session' type of Auth state persistence. This indicates that the state will only persist in the current session or tab, and will be cleared when the tab or window in which the user authenticated is closed.

It means that:

- User should close a tab or sign out in order to be logged out.
- Multiple users can be logged in in the same browser using multiple tabs.

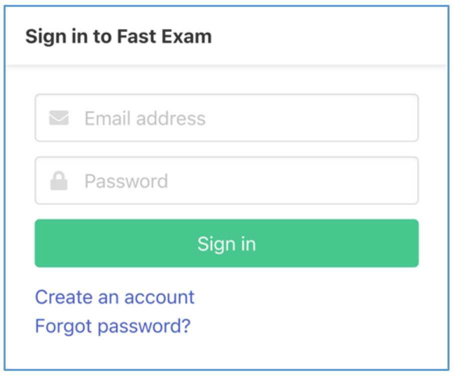


Figure 11. Login.

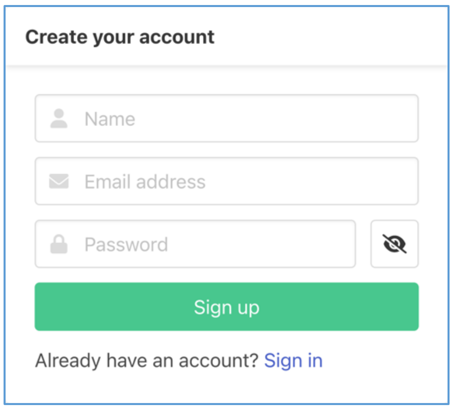


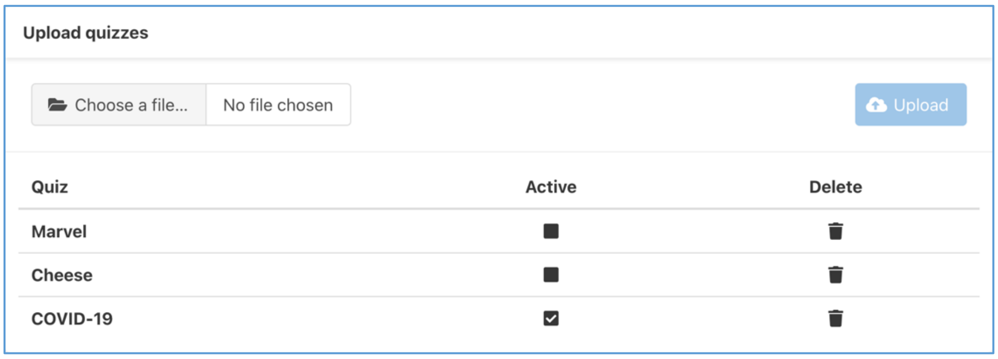
Figure 12. Registration.

10.1 Confirm password field

Based on the modern UX practices, this field is considered as a bad design. Optional password masking can be used instead [54].

11 Implementation – Quizzes

Here you can manage your quizzes: upload, activate and delete them. Quiz names are tab names in spreadsheet file. Multiple quizzes can be uploaded at the same time.


Upload quizzes

Choose a file... No file chosen

Upload

Quiz	Active	Delete
Marvel	■	🗑
Cheese	■	🗑
COVID-19	✓	🗑

Figure 13. Upload.

Fast exam is using spreadsheet import feature. Just choose a file with quiz questions and upload it. Spreadsheet template is quite simple and it's easy to use existing spreadsheets:

Table 1. Spreadsheet template.

question	**imageUrl**	**options**	**answer**

Where:

- **question** is a question text
- **imageUrl** is a question image url
- **options** is a list of available answers
- **answer** is the correct answer

Here is the corresponding Question type declaration (object properties are spreadsheet column names):

```
type Question = {
  text: string;
  imageUrl: string;
  options: string[];
  correctOption: number;
};
```

Figure 14. Question type.

12 Implementation – Game

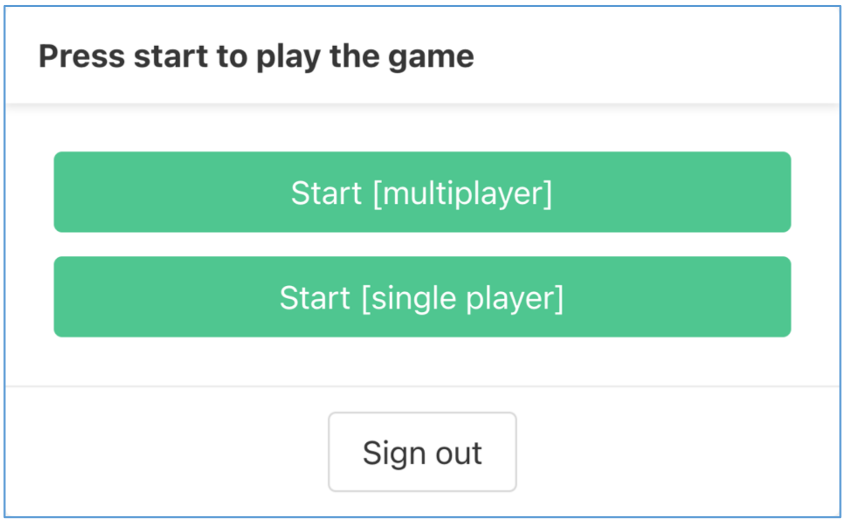


Figure 15. Game start.

It's possible to start a game in two modes: multiplayer and single player. Technically there are no differences between these modes except the number of players in a room:

- Single player mode creates a room with 1 player and immediately starts a game.
- Multiplayer mode creates a room and then waits for other players to join.

```
createRoom(t, {
  full: player.single,
  size: player.single
    ? 1
    : MULTIPLAYER_ROOM_SIZE,
  quizId: quiz.ref.id,
  playerIds: [player.ref.id],
  scores: {
    [player.ref.id]: 0,
  },
  closed: false,
});
```

Figure 16. Create room function call.

13 Implementation – Loading screens

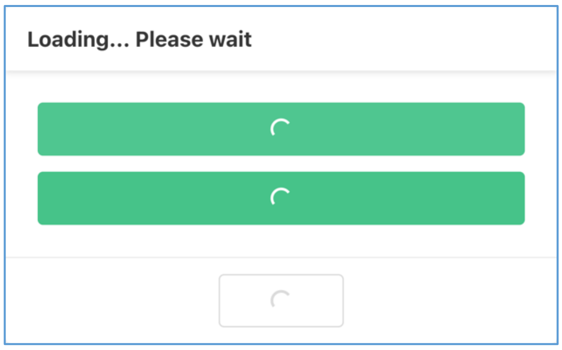


Figure 17. Loading.

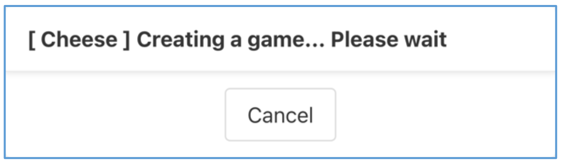


Figure 18. Single player loading.

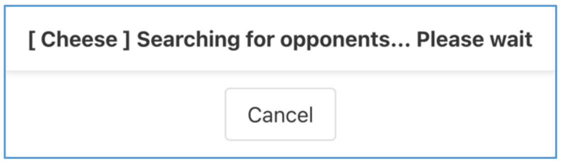
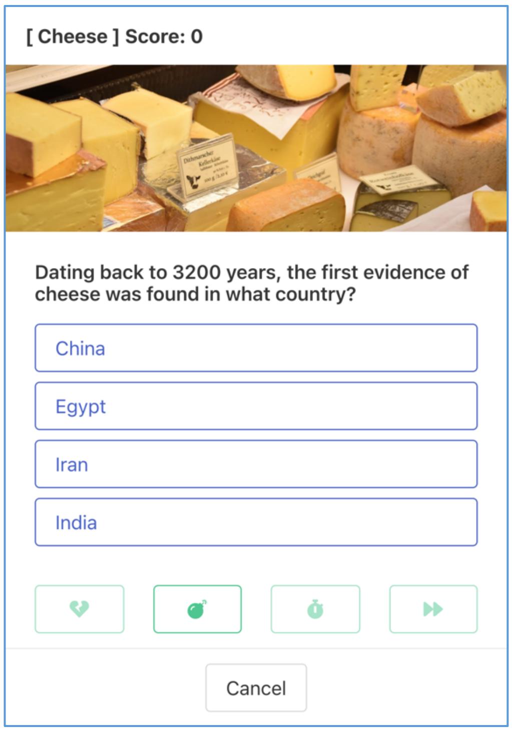


Figure 19. Multiplayer loading.

Loading screens provide a smooth user experience. User can also cancel loading at every moment and return back to home screen (Game Start).

14 Implementation – Questions


[Cheese] Score: 0

Dating back to 3200 years, the first evidence of cheese was found in what country?

China

Egypt

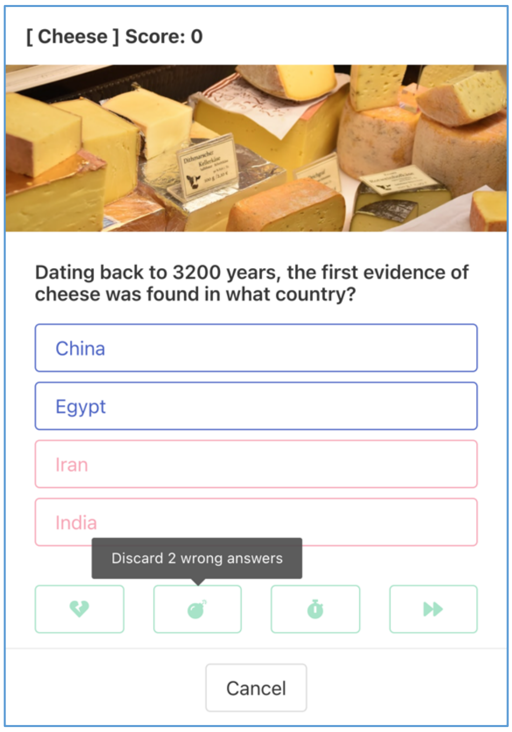
Iran

India

Figure 20. Question.

Every question has optional image, question text, several answer options (the number is not limited) and several power ups (the number is not limited and can be disabled / enabled). User can also cancel a game at every moment and return to the home screen. Excel file with demo quizzes and questions is created as well [55].

15 Implementation – Power Ups


[Cheese] Score: 0

Dating back to 3200 years, the first evidence of cheese was found in what country?

China

Egypt

Iran

India

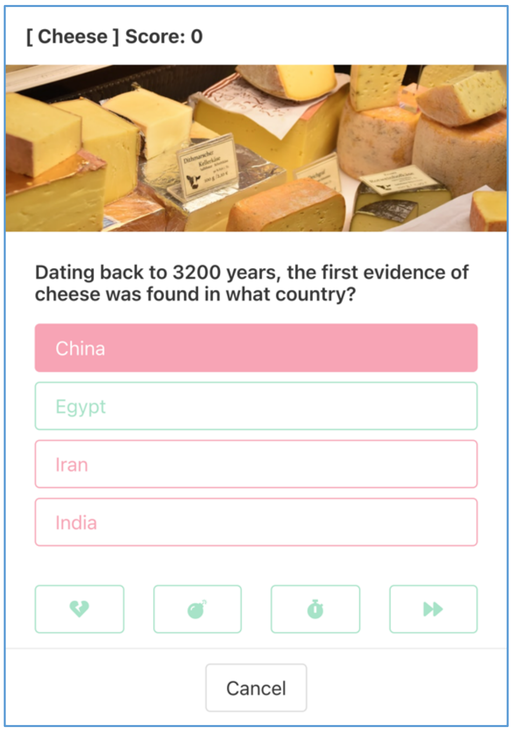
Discard 2 wrong answers

Cancel

Figure 21. Power Up.

Fast Exam prototype has 4 power ups and only one of them is enabled (Bomb). As shown in the screenshot above, power up discarded two wrong answers. If options or power ups don’t fit on the screen, CSS Scroll Snap feature can be used. Tabbing navigation is available as well.

16 Implementation – Answers


[Cheese] Score: 0

Dating back to 3200 years, the first evidence of cheese was found in what country?

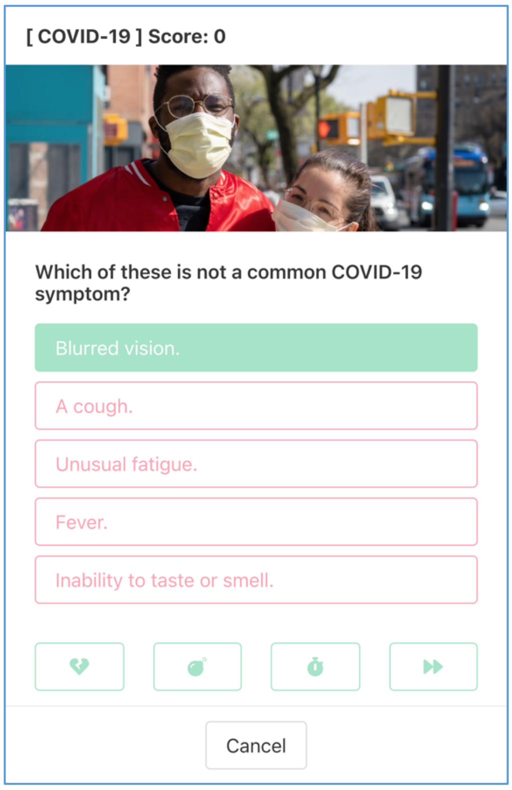
China

Egypt

Iran

India

Figure 22. Wrong answer.


[COVID-19] Score: 0

Which of these is not a common COVID-19 symptom?

Blurred vision.

A cough.

Unusual fatigue.

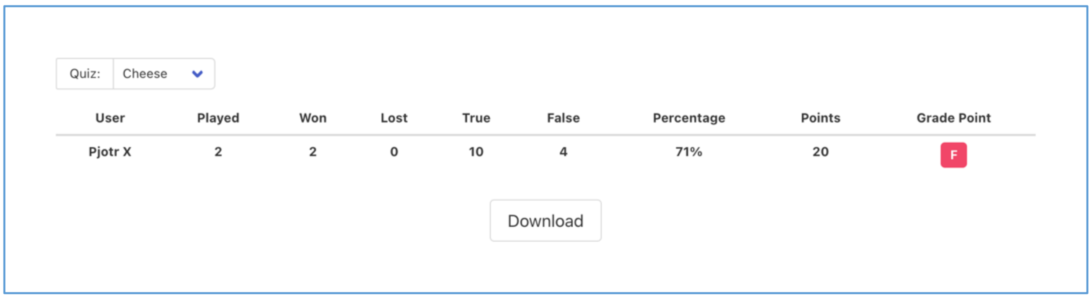
Fever.

Inability to taste or smell.

Figure 23. Right answer.

Wrong and right answers and marked accordingly using green / red colours.

17 Implementation – Statistics



User	Played	Won	Lost	True	False	Percentage	Points	Grade Point
Pjotr X	2	2	0	10	4	71%	20	F

Figure 24. Statistics.

It’s possible to see user per quiz and overall results. True and false are the numbers of right and wrong answers. Grade point is an automatic grade based on the total amount of the gained points:

```
if (value < 51) {
  return 'F';
} else if (value < 61) {
  return 'E';
} else if (value < 71) {
  return 'D';
} else if (value < 81) {
  return 'C';
} else if (value < 91) {
  return 'B';
} else {
  return 'A';
}
```

Figure 25. Grade point calculation.

18 Fast Exam – Serverless multiplayer game

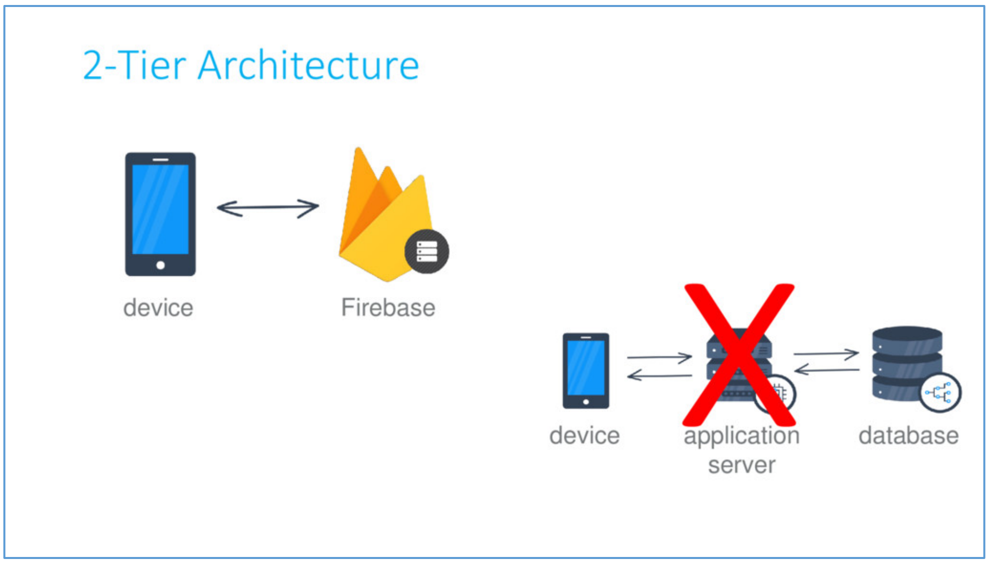


Figure 26. Firebase architecture.

What does serverless mean? It means that we don't have a server and everything works without client – server communication. How it's possible? We're using database triggers and listeners instead. Clients only make changes in synchronized cloud-based database and react to database updates. In addition to that, database reacts to these updates as well, invoking database triggers. Let's see how it works in Fast Exam.

When a first user starts a game:

- user subscribes to question records
- player record is created in database
- playerCreated trigger function is invoked
- room record is created

When a second user starts a game:

- user subscribes to question records
- player record is created is database

- playerCreated trigger function is invoked
- room record is updated (room is full – game starts)
- question records are created

Because both users subscribed to question records in step #1, after questions are created in database both clients can display these questions.

Fast Exam has only 2 mandatory triggers (yes, that's enough to build a multiplayer game!):

- playerCreated
- answerCreated

19 Game Logic

Fast Exam was implemented in the way that it's possible to create different game (exam) types, there are no limits what logic could be you used. Some examples:

1. Players answer random questions until one of the players gets a specific score. E.g. the first player who gets 10 points – wins the game and gets extra points. Trivia Crack type game. In this case it's better to have as much questions as possible, so, they don't repeat.

2. Players answer questions one by one, no winners in this case, but faster students get more points. Kahoot! type game. It's good to have some time limits, so, players don't answer questions all day long.

3. Players could also get different quizzes, e.g. one player might get test A, the other one test B. Classical old school exam type. It's better to have 5 quizzes with 10 questions than 1 quiz with 50 questions. Single player can be used too.

#1 variant is implemented in Fast Exam.

If you think of physical education teachers, they use different type of sports in their classes, similar could be done with Fast Exam. It’s always great to be able to play various games. So, one Quiz Game could be done with time limits like in football, but the other one could be done without timers like running – the fastest student will be the winner.

Fast Exam can also be used to make surveys, questionnaires, quizzes, polls, tests, etc.

20 Demo Questions

Having good questions in any test is quite essential. Fast Exam multiple-choice questions are created based on a checklist:

- Decide exactly what you want to test (done).
- Eliminate as many ambiguities as possible (done).
- Create distractors based firmly on what you want to test (done).
- Give feedback for your distractors which explains why they're wrong.
- Give feedback for the correct answer explaining why it's right.
- Allocate useful correctness values to your distractors.

While it’s not implemented in MVP, there is also a way to make our quiz game more like a conventional test:

- Add a time-limit.
- Show a randomly-selected number of questions.
- Shuffle the order of questions shown every time the page loads.
- Shuffle the order of answers in all the questions every time the page loads.

21 Fast Exam – Possible Improvements

Technical:

- Unit and end-to-end tests.
- Snapshot and screenshot tests.
- Documentation (Helps and manuals).
- UI Documentation (Storybook [56]).
- Error and Performance Monitoring (Sentry [57]).
- Optimizations (Vite [58], PurgeCSS [59], terser [60], etc).

Functional:

- Capacitor and native mobile app.
- Wearables.
- Monetization aspects.
- Psychology in game design.
- LTI support [61] and LMS integrations [62].
- a11y and i18n.

Synchronized spreadsheets:

- Google Sheets [63] (questions & records).
- Google Sheets API [64] (Firebase to Sheets - records).
- Google Apps Script [65] (Sheets to Firebase - questions).
- Zapier [66].

Different question types (MSc topic):

- Custom answers (calculated text field).
- Matching questions.
- Drag & drop (text or image).

Machine Learning (PhD topic):

- Duolingo [67] story [68] [69].
- AI creates questions and answers (great solution), based on some text.
- AI creates wrong answers (simplified version), based on 1 question and 1 answer.

Surveys and feedbacks of using Fast Exam can be conducted as well.

22 Summary

We have succeeded to create a quiz game (base app for LMS) what is technically and functionally far beyond the competitors.

We removed the need to write CSS by using utility classes.

We removed the need of administrators.

While popular quiz apps struggle to deal with different redundant functional limits, by specifying requirements and prioritizing them we removed this kind of limits.

Using TypeScript, we wrote a high quality, bug-less, self-documented and clean code.

Using Vue, we removed the need to write DOM related functionality. DOM updates, two-side binding, etc are done automagically.

Using Firebase, we removed a need to have a server and all server related layers and logic. Client has direct access to database and it’s regulated by powerful security rules.

Despite this, there are many improvements what can be done. But because everything is modular, every part of the app can be easily replaced or expanded. For example,

- Firebase Auth can be replaced with **Smart-ID** login [70].
- Firebase Firestore can be replaced with **MongoDB** [71].
- Firebase Functions can be replaced with **database triggers** [72].
- Firebase Hosting can be replaced with **Amazon Web Services** (AWS) [73].
- Vue.js can be replaced with **React** [74].

References

[1] “LMS,” [Online]. Available: https://trustsourcing.com/Article/how-to-develop-a-learning-management-system-lms.

[2] “Digital Study Trends Survey,” [Online]. Available: https://www.mheducation.com/highered/explore/studytrends.html.

[3] “Kahoot!,” [Online]. Available: https://kahoot.com/.

[4] “Trivia Crack,” [Online]. Available: https://triviacrack.com/.

[5] “Vue.js,” [Online]. Available: https://vuejs.org/.

[6] “Firebase,” [Online]. Available: https://firebase.google.com/.

[7] “Firebase Project,” [Online]. Available: https://firebase.google.com/codelabs/firebase-web.

[8] “Realtime updates,” [Online]. Available: https://firebase.google.com/docs/firestore/query-data/listen.

[9] “Transactions,” [Online]. Available: https://firebase.google.com/docs/firestore/manage-data/transactions.

[10] “Firestore triggers,” [Online]. Available: https://firebase.google.com/docs/functions/firestore-events.

[11] “Security rules,” [Online]. Available: https://cloud.google.com/firestore/docs/security/rules-fields.

[12] “Access control,” [Online]. Available: https://firebase.google.com/docs/auth/admin/custom-claims.

[13] “Authentication State,” [Online]. Available: https://firebase.google.com/docs/auth/web/auth-state-persistence.

[14] “The Firebase Blog,” [Online]. Available: https://firebase.googleblog.com/2020/10/whats-new-at-Firebase-Summit-2020.html.

[15] “Wikimedia mailing-list,” [Online]. Available: https://lists.wikimedia.org/hyperkitty/list/wikitech-l@lists.wikimedia.org/thread/SOZREBYR36PUNFZXMIUBVAIOQI4N7PDU/.

[16] “TypeScript,” [Online]. Available: https://www.typescriptlang.org/.

[17] “Bulma,” [Online]. Available: https://bulma.io/.

[18] “Sass,” [Online]. Available: https://sass-lang.com/.

[19] “Bulma - Card,” [Online]. Available: https://bulma.io/documentation/components/card/.

[20] “Bulma - Extensions,” [Online]. Available: https://bulma.io/extensions/.

[21] “Font Awesome,” [Online]. Available: https://fontawesome.com/.

[22] “SVG,” [Online]. Available: https://fontawesome.com/v6.0/docs/web/dig-deeper/webfont-vs-svg.

[23] “Cloud Firestore,” [Online]. Available: https://firebase.google.com/products/firestore.

[24] “Cloud Functions,” [Online]. Available: https://firebase.google.com/products/functions.

[25] “Firebase Authentication,” [Online]. Available: https://firebase.google.com/products/auth.

[26] “Firebase Hosting,” [Online]. Available: https://firebase.google.com/products/hosting.

[27] “Vue Router,” [Online]. Available: https://router.vuejs.org/.

[28] “Vuex,” [Online]. Available: https://vuex.vuejs.org/.

[29] “Vue CLI,” [Online]. Available: https://cli.vuejs.org/.

[30] “Vue Devtools,” [Online]. Available: https://devtools.vuejs.org/.

[31] “Vue-Screen,” [Online]. Available: https://reegodev.github.io/vue-screen/.

[32] “JavaScript - Media queries,” [Online]. Available: https://blog.logrocket.com/introduction-to-js-media-queries/.

[33] “SheetJS,” [Online]. Available: https://sheetjs.com/.

[34] “Progressive Web App,” [Online]. Available: https://web.dev/progressive-web-apps/.

[35] “Electron,” [Online]. Available: https://www.electronjs.org/.

[36] “Vue Electron Builder,” [Online]. Available: https://nklayman.github.io/vue-cli-plugin-electron-builder/.

[37] “MVP,” [Online]. Available: https://keytotech.com/2020/05/15/poc-prototype-mvp-difference/.

[38] “MVP Features,” [Online]. Available: https://dzone.com/articles/how-to-build-an-mvp-the-best-feature-prioritizatio.

[39] “Mobile First Design,” [Online]. Available: https://www.browserstack.com/guide/how-to-implement-mobile-first-design.

[40] “Functional Programming,” [Online]. Available: https://www.journaldev.com/8693/functional-imperative-object-oriented-programming-comparison.

[41] “Acronyms,” [Online]. Available: https://thefullstack.xyz/dry-yagni-kiss-tdd-soc-bdfu.

[42] “Firebase - Treeshaking,” [Online]. Available: https://firebase.google.com/docs/web/modular-upgrade.

[43] “Vue - Treeshaking,” [Online]. Available: https://v3.vuejs.org/guide/migration/global-api-treeshaking.html.

[44] “Bulma - Treeshaking,” [Online]. Available: https://bulma.io/documentation/overview/modular/.

[45] “Font Awesome - Treeshaking,” [Online]. Available: https://fontawesome.com/v5.15/how-to-use/javascript-api/other/tree-shaking.

[46] “Kahoot! - Question types,” [Online]. Available: https://support.kahoot.com/hc/en-us/articles/115002308428-Question-types.

[47] “Kahoot! - Spreadsheet template,” [Online]. Available: https://support.kahoot.com/hc/en-us/articles/115002812547-Import-questions-from-a-spreadsheet.

[48] “Moodle,” [Online]. Available: https://moodle.org/.

[49] “TalTech - E-learning,” [Online]. Available: https://taltech.ee/en/e-learning.

[50] “Moodle - Create Quiz,” [Online]. Available: https://sisu.ut.ee/juhendid/creating-quiz-moodle.

[51] “Moodle - Quiz activity,” [Online]. Available: https://docs.moodle.org/311/en/Quiz_activity.

[52] "Hot Potatoes," [Online]. Available: https://hotpot.uvic.ca/.

[53] "Navigation," [Online]. Available: https://medium.muz.li/3-good-reason-why-you-might-want-to-remove-that-hamburger-menu-from-your-product-69b9499ba7e2.

[54] "Confirm Password," [Online]. Available: https://uxmovement.com/forms/why-the-confirm-password-field-must-die/.

[55] "COVID-19 Quiz," [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/test-your-knowledge-covid-19.

[56] "Storybook," [Online]. Available: https://storybook.js.org/.

[57] "Sentry," [Online]. Available: https://sentry.io/welcome/.

[58] "Vite," [Online]. Available: https://vitejs.dev/.

[59] "PurgeCSS," [Online]. Available: https://purgecss.com/.

[60] "terser," [Online]. Available: https://terser.org/.

[61] "LTI," [Online]. Available: https://www.edu-apps.org/code.html.

[62] "Moodle and LTI," [Online]. Available: https://docs.moodle.org/311/en/LTI_and_Moodle.

[63] "Google Sheets," [Online]. Available: https://www.google.com/sheets/about/.

[64] "Sheets API," [Online]. Available: https://developers.google.com/sheets/api.

[65] "Google Apps Script," [Online]. Available: https://developers.google.com/apps-script.

[66] "Zapier," [Online]. Available: https://zapier.com/.

[67] "Duolingo," [Online]. Available: https://www.duolingo.com/.

[68] "Duolingo - AI," [Online]. Available: https://venturebeat.com/2020/04/30/duolingos-english-test-ai-serve-and-score-questions/.

[69] "Duolingo - AI," [Online]. Available: https://venturebeat.com/2020/08/18/how-duolingo-uses-ai-in-every-part-of-its-app/.

[70] "Smart-ID," [Online]. Available: https://www.smart-id.com/.

[71] "MongoDB," [Online]. Available: https://www.mongodb.com/.

[72] "Database Triggers," [Online]. Available: https://docs.mongodb.com/realm/triggers/database-triggers/.

[73] "Amazon Web Services," [Online]. Available: https://aws.amazon.com/.

[74] "React," [Online]. Available: https://reactjs.org/.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I, Pjotr Surkov

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Serverless cloud-based cross-platform multiplayer educational game”, supervised by Jaak Henno.
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

10.01.2022

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Use case diagram

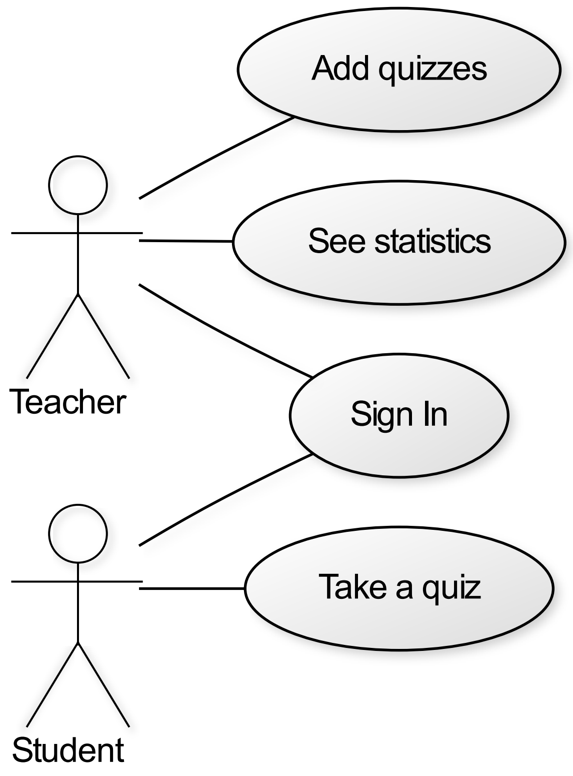
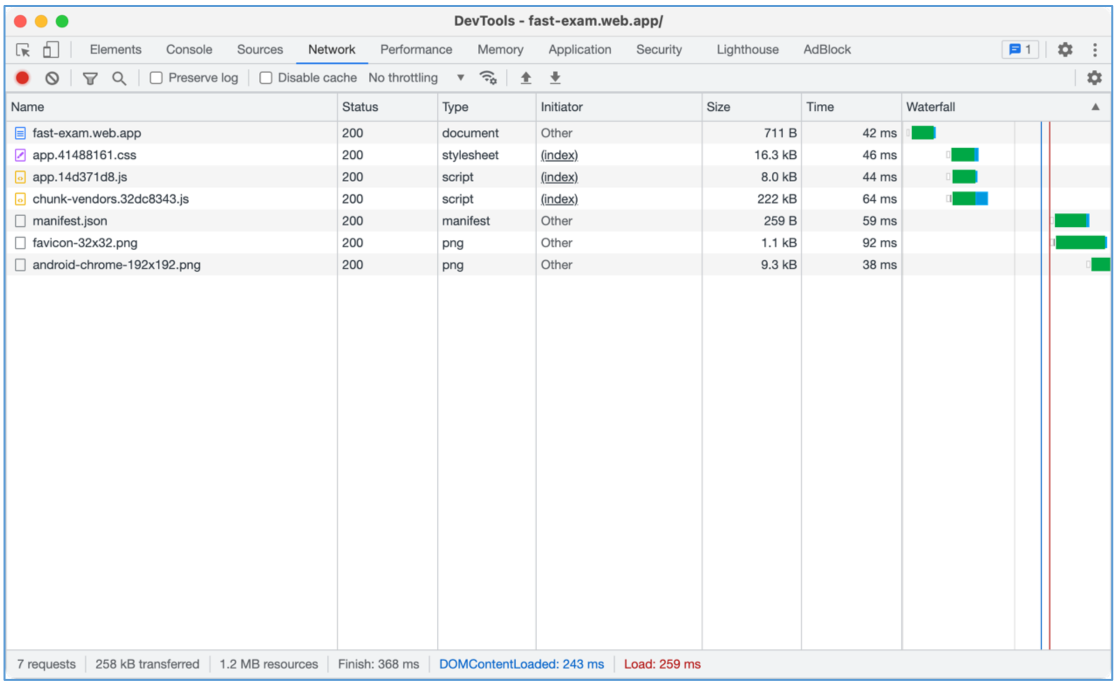


Figure 27. Use case diagram.

Appendix 3 – Performance and bundle size



Name	Status	Type	Initiator	Size	Time	Waterfall
fast-exam.web.app	200	document	Other	711 B	42 ms	
app.41488161.css	200	stylesheet	(index)	16.3 kB	46 ms	
app.14d371d8.js	200	script	(index)	8.0 kB	44 ms	
chunk-vendors.32dc8343.js	200	script	(index)	222 kB	64 ms	
manifest.json	200	manifest	Other	259 B	59 ms	
favicon-32x32.png	200	png	Other	1.1 kB	92 ms	
android-chrome-192x192.png	200	png	Other	9.3 kB	38 ms	

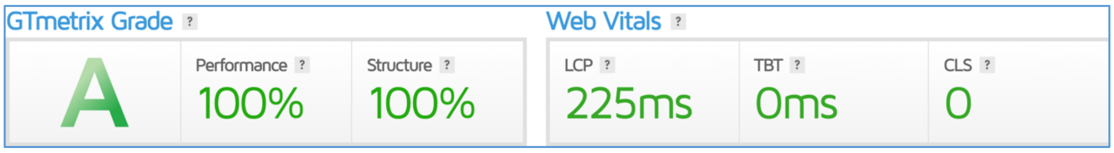
7 requests | 258 kB transferred | 1.2 MB resources | Finish: 368 ms | DOMContentLoaded: 243 ms | Load: 259 ms

Figure 28. Network activity.

Fast Exam page size is 252 KB, in which:

- JavaScript – 225 KB.
- CSS – 16 KB.

Unique hashes allow JS and CSS files to be browser cached for 1 year.



GTmetrix Grade ?

A

Performance ? 100%

Structure ? 100%

Web Vitals ?

LCP ? 225ms

TBT ? 0ms

CLS ? 0

Figure 29. Performance test.

Fast Exam has the highest score in GTmetrix (A) and PageSpeed Insights (100).