

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж

“ДОПУСТИТИ ДО ЗАХИСТУ”
Завідувач кафедри

_____ Жуков І.А.

“ _____ ” _____ 2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

випускника освітнього ступеня “МАГІСТР”
спеціальності 123 «Комп'ютерна інженерія»
освітньо-професійної програми «Комп'ютерні системи та мережі»

на тему: **“Система аналізу даних соціального опитування, побудована за допомогою бібліотек React.js та Apollo GraphQL”**

Виконавець: _____ Аксьонова М.А.

Керівник: _____ Кудренко С.О.

Нормоконтролер: _____ Надточій В.І.

Засвідчую, що у дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань

Аксьонова М.А.

Київ 2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Cybersecurity, Computer and Software Engineering
Computer Systems and Networks Department

“PERMISSION TO DEFEND GRANTED”
The Head of the Department

_____ Zhukov I.A.

“ _____ ” _____ 2020 p.

MASTER’S DEGREE THESIS

(EXPLANATORY NOTE)

Specialty: 123 Computer Engineering

Educational-Professional Program: Computer Systems and Networks

Topic: **“The System for Social Survey Data Analysis Created with React.js
and Apollo GraphQL Libraries.”**

Completed by: _____ Aksonova M.A.

Supervisor: _____ Kudrenko S.O.

Standard’s Inspector: _____ Nadtochii V.I.

Kyiv 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних систем та мереж

Освітній ступінь: «Магістр»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерні системи та мережі»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри

Жуков

І.А.

“ ” 2020 р.

ЗАВДАННЯ

на виконання дипломної роботи

Аксьонові Маргарити Андріївни

(прізвище, ім'я та по-батькові випускника в родовому відмінку)

1. Тема дипломної роботи: “Система аналізу даних соціального опитування, побудована за допомогою бібліотек React.js та Apollo GraphQL” затверджена наказом ректора від 25.09.2020 р. № 1793/ст
2. Термін виконання роботи (проекту): з 1 жовтня 2020 р. до 25 грудня 2020 р.
3. Вихідні дані до роботи (проекту): Система для проектування інтер'єрів
4. Зміст пояснювальної записки: Вступ, огляд теми, огляд існуючих технологій побудови системи аналізу даних соціального опитування, розробка системи аналізу даних соціального опитування, висновки по роботі.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: Графічні матеріали результатів дослідження надати у вигляді презентації у форматах .ppt, .pdf.

NATIONAL AVIATION UNIVERSITY

Faculty of Cybersecurity, Computer and Software Engineering

Department: Computer Systems and Networks

Educational Degree: "Master"

Specialty: 123 "Computer Engineering"

Educational-Professional Program: "Computer Systems and Networks"

"APPROVED BY"

The Head of the Department

Zhukov I.A.

" " 2020 p.

Graduate Student's Degree Thesis Assignment

Aksonova Marharyta Andriivna

1. Thesis topic: "The System for Social Survey Data Analysis Created with React.js and Apollo GraphQL Libraries" approved by the Rector's order of 25.09.2020 p. № 1793/CT

2. Thesis to be completed between 01.10.2020 and 25.12.2020

3. Initial data for the project (thesis): system for interior design

4. The content of the explanatory note (the list of problems to be considered):
Introduction, overview of the topic, review of existing technologies for developing the software module for the system for social survey data analysis, design of the the system for social survey data analysis, conclusions on work

5. The list of mandatory graphic materials: Graphic materials are given in MS Power Point presentation.

6. Календарний план-графік

№ пор.	Завдання	Термін Виконання	Підпис керівника
1	Узгодити технічне завдання з керівником дипломної роботи	1.10.20-8.10.20	
2	Виконати пошук та вивчення науково-технічної літератури за темою роботи	9.10.20-15.10.20	
3	Опрацювати теоретичний матеріал	16.10.20-18.10.20	
4	Огляд існуючих технологій для розробки програмного модуля для системи аналізу даних соціального опитування.	19.10.20-03.11.20	
5	Розробка програмного модуля для системи аналізу даних соціального опитування.	04.11.20-15.12.20	
6	Тестування роботи програмного модуля та виправлення знайдених помилок	16.12.20	
7	Оформлення пояснювальної записки	06.12.20-12.12.20	
8	Оформити графічні матеріали проекту та подати матеріали роботи на антиплагіатну перевірку матеріалів	13.12.20-14.12.20	
9	Отримати рецензію та відгук керівника та надати матеріали роботи на кафедру.	15.12.20-18.12.20	

7. Дата видачі завдання: “1” жовтня 2020 р.

Керівник дипломної роботи _____ Кудренко С.О.
(підпис керівника)

Завдання прийняв до виконання _____ Аксьонова М. А.
(підпис випускника)

6. TIMETABLE

#	Completion stages of Degree Project (Thesis)	Stage Completion Dates	Signature of the supervisor
1	Technical task coordination with the supervisor	1.10.20-8.10.20	
2	Selection and study scientific literature on the topic	9.10.20-15.10.20	
3	Working with theoretical materials	16.10.20-18.10.20	
4	Review of existing technologies for developing the system for social survey data analysis	19.10.20-03.11.20	
5	Designing of software module for the system for social survey data analysis	04.11.20-15.12.20	
6	Testing of the designed software module and fixing the found bugs	16.12.20	
7	Making the explanatory notes	06.12.20-12.12.20	
8	Preparation of the graphical materials and filing materials of work to antiplagiarism checking of materials	13.12.20-14.12.20	
9	Receiving the reviews from reviewer and supervisor and providing the materials to the department	15.12.20-18.12.20	

7. Assignment issue date: 1.10.2020

Diploma Thesis Supervisor _____ Kudrenko S.O.
(Signature)

Assignment accepted for completion _____ Aksonova M.A.
(Student's Signature)

ABSTRACT

The Explanatory Note to on Master's Degree Graduation Project – “The Social Survey Data Analysis System”: 85 pages, 32 figures, 1 table, 19 references.

The Goal of Graduation Project. The goal of the graduation project is to design a system that can help users to analyze the survey answers and made improvements or conclusions based on the achieved results.

Main Tasks. The main tasks which have been set out are to analyze the user's needs, to develop a useful and easy to use system for analyzing the surveys, to protect the system with the security rules to make it secure for each user.

The Subject of Project. Designing, developing and testing the social survey data analysis system.

Practical usage Can be used by each person in to analyze the answers on the survey that was created.

Main Metrics and Results. As a result, the social survey data analysis system was developed using React.js library and MongoDB Database. The program is composed of two parts - the server and the client. All the functionality is covered by unit tests.

The users have access only to their data. The main functionality of the system is to create surveys, share them, get answers and make analyzing. The performance of the system is in a high quality.

CONTENTS

LIST OF SYMBOLS, ABBREVEATIONS, TERMS	10
INTRODUCTION	11
PART 1 REQUIREMENTS COLLECTION TO THE SOCIAL SURVEY DATA ANALYSIS SYSTEM	14
1.1. The Purpose and Scope of the Social Survey Data Analysis System.....	15
1.2. Assignment of the Task	15
1.3. Problem Definition	16
1.4. Analysis of the Existing Developments	17
1.5. Feasibility Study.....	19
1.6. Performance Requirements.....	19
1.7. Functional Requirements	20
1.8. Interface Requirements	20
1.9. Security Requirements.....	23
Conclusions on the First Part.....	24
PART 2 OVERVIEW OF SELECTED TECHNOLOGIES. JUSTIFICATION FOR CHOOSING A TECHNOLOGY STACK. ENVIRONMENT DESCRIPTION	26
2.1. HTML	26
2.2. CSS	29
2.3. Javascript Programming Language	30
2.4. Technologies and Platforms Selected for Creating the System	31
2.4.1. React.js Library	31
2.4.2. Typescript Programming Language	33
2.4.3. GraphQL Query Language	35
2.4.4. D3.js Library	36
2.4.5. React Testing Library	36
2.4.6. Node.js Runtime Environment.....	37
2.4.7. MongoDB Database.....	37

2.5. Webpack Bundler	38
2.6. REST vs. GraphQL	38
2.7. NPM tool.....	38
2.8. Charts. Types for Charts	38
2.9. Web browser. Developer tools and extensions.....	38
2.10. Description of the Code Development Environment, Linters	38
2.10.1. IDE WebStorm	37
2.10.2. ESLint, StyleLint and Prettier code formatter	37
2.11. Git	38
2.11.1. Forks.....	37
2.11.2. Branch vs. Fork	37
2.11.3. Submodules	37
Conclusions on the Second Part	39
PART 3 INTERFACE AND LOGICAL DESIGN, SYSTEM IMPLEMENTATION	41
3.1. Design a User Interface of the System.....	41
3.2. Database Design.....	45
3.3. Purpose of the Testing	48
3.4. Testing Strategy.....	49
3.5. Performance Results.....	50
3.6. Development of the User Manual	51
3.7. Directions of Further Development of the System	53
Conclusions on the Third Part	55
CONCLUSIONS.....	56
REFERENCES	59

LIST OF SYMBOLS, ABBREVEATIONS, TERMS

HTML	– Hypertext Markup Language
CSS	– Cascading Style Sheets
MVC	– Model–view–controller
MVP	– Minimum viable product
UI	– User Interface
UX	– User Experience
SPA	– Single-page Application
DOM	– Document Object Model
URL	– Uniform Resource Locator
REST	– Representational State Transfer
API	– Application Programming Interface
HTTP	– Hypertext Transfer Protocol
HTTPS	– Hypertext Transfer Protocol Secure
AJAX	– Asynchronous JavaScript and XML
ES	– ECMAScript
SVG	– Scalable Vector Graphics
NPM	– Node Package Manager
IDE	– Integrated Development Environment

INTRODUCTION

Currently, sociological research is not uncommon. For the most part, they are designed for a huge number of respondents, and therefore for a huge amount of information. In our time, information is the main factor on which the effectiveness of all spheres of life depends.

Due to the increase in information flows, not all operations can be performed manually. For example, calculating research results. In this case, it becomes necessary to use a computer and computer technology.

All management tasks require accurate analysis. And, since earlier dozens of factors that a person could calculate manually took part in the analysis of the situation, now there are much more of these factors. The same should be mentioned about the great responsibility that is assigned to the one who is directly involved in the results of sociological research. In connection with all this, there is a need to automate the collection and processing of sociological information.

The automation of sociological research processes occurs in several stages:

- automation of input of primary information;
- automation of data processing;
- automation of obtaining the required results.

Before, people wanted to automate various processes of their life, but there was no possibility; these processes required special software and high-quality service. In the modern world, in the age of high technologies, what a person could not afford before has become generally available. It should also be said that automation has the greatest impact on information, and all because of the need for quick search, processing, selection and quick access to it.

The main advantage of the automation process is that it allows to reduce the amount of required memory, reduce the time for data processing, and reduce the number of copies of documents when updating information. Also, as for advantages, it allows to find the information of interest in a short period of time.

The purpose of this diploma project is to develop an information system for automating sociological research processes, namely questionnaires.

The introduction of the automation process will simplify the researcher's work and will allow him to perform his work much faster, more complete and better.

Quantitative methods are classified into:

- polls;
- document analysis;
- observation;

A survey is the collection of primary data when a researcher addresses a respondent or a group with certain questions.

Questionnaire - conducting a survey using a questionnaire. A questionnaire is a specially prepared sheet with questions that must be completed by the respondent.

Respondent is a person who is the object of sociological research.

Document analysis is a set of actions involving the study of documentary sources aimed at collecting information and performing certain research tasks.

The demand for the automated systems for surveys analysis are growing on the information technology market, which means that the supply is also growing. But the functionality of automated control systems is very different. Most of them are not fully free and the functionality is very limited.

One of the goals pursued in the automation of data analysis is the creation of a single information storage. The presence of a single information storage center in order to minimize the functions of individual users is an equally important task when creating automated systems.

Also, one of the requirements for the system being developed is the creation of a convenient user interface that is not overloaded with unnecessary information and functionality.

It provides easy perception and processing of information, providing a free output of information in a more familiar form for employees of the enterprise, such as in document format. The common assignment of the task is to design and develop a project, that will be

suitable and comfortable to work both with the small amount of data required for concrete issue and with a huge amount of data.

In addition, the data analysis system should provide differentiation of the information provided, depending on the role of the user, be resistant to errors and negative input data.

The goal which was achieved is: to solve the problem of individuals or small companies with the analysis of accumulated data statistics, without spending additional funds on purchasing additional software products.

At the moment, in order to analyse received data from surveys there are some software solutions exists. But monitoring their functionality it was noticed that they are large giants capable of performing all possible types of analyses and operations on data. Because of that, they are very voluminous and consume a lot of system resources, and also require special training of personnel to work with them. Using such products in for surveys is not really appropriate.

Therefore, the purpose of the project is to develop a compact software product sharpened for solving a specific problem and working with the minimum amount of data required for specific purpose, and with very large amounts of data.

Having developed a software product that will provide the opportunity for multifunctional analysis and process large amounts of information, users will be able to analyse data from different aspects, which will help users to determine new dependencies and patterns.

A new software product with such capabilities will solve the problem of individuals or small companies with the analysis of accumulated data statistics, without spending additional funds on purchasing additional software products.

In the first part the requirements are described. The system for analyzing social survey data was developed in accordance with the established requirements.

The functional requirements dictated that the system should have the platform menu in the header and check whether users have access to surveys and dashboard.

The interface requirements dictated that the system should have dashboard page, login page, personal account with information about the user, survey page and question analysis page.

As for security requirements, besides fact that the system should be convenient to use, it also has to provide security to users and their private data. Before developing a methodology for protecting a web application from potential threats, the site should be checked for vulnerabilities.

The second section examines in detail the popular concept of the one-page application website and the key elements that make it up. The choice of technology, library, architectural solutions and approaches depends on the tasks to be solved and their complexity. In the second section, frameworks and technologies for creating a system for data analysis were discussed.

The vast majority of the main features of libraries, packages and how to use them were also considered. The structure of the application, the process of creating and running the application are described in detail. Development using these technologies was convenient and comfortable. Their capabilities are quite sufficient to create quite complex applications.

The third section fully described the process of creating a web application using selected technologies. Figures with results of work, examples from the code and results of checking the productivity of system execution were provided. The unit test results also were provided.

To achieve the highest testing quality for each method and component of a tested program, it was chosen the most successful method or set of methods that will provide the desired result.

As part of this project, there is no need for a relational database and was opted for an object database.

MongoDB is an efficient and scalable database for this task. MongoDB is a very successful symbiosis between the familiar relational database and key-value storage.

All documents are grouped in the collections and global relationships are already established between collections.

Unit testing a react component using snapshots allows to write very simple and easily maintainable tests that will help avoid problems in production and make the project code more reliable. For each test case, a snapshot is created, which is considered a reference.

After developing the system, there were added a test that verifies that the component renders correctly. To do that, there were included the react-testing and jest libraries to shallow the components.

Development was carried out using the git version control system. It is saved in a private repository of a GitHub web service for hosting IT projects based on git.

The system was developed using Web Storm IDE. Choosing between a text editor and an IDE, the preference immediately was given to the latter.

Any system can be growth and every system have direction of further development. The potential of the application is quite large, since within the framework of this application it can be placed any database of questions and adapt into any survey format.

This system can be introduced into educational processes to receive feedback from staff and students too. There were also developed the user manual for quick start working with a system.

As the result the desired analysis system was developed. It is secure for users so they can store their information and private data without any doubts. It also has high performance results and beautiful interface.

PART 1

REQUIREMENTS COLLECTION TO THE SOCIAL SURVEY DATA ANALYSIS SYSTEM

1.1. The Purpose and Scope of the Social Survey Data Analysis System

Receiving the results of the survey, it can be online or offline, there are the statistical results of the survey and a data analysis plan, which means it is time to evaluate the information received. The social survey data analysis system is useful to find an explanation for quantitative data from reviewing the answers and highlighting the main research questions and survey objectives, to processing numbers and formulating conclusions.

For better measure the survey results, the system included the functionality to:

- identify the main research questions;
- make a crosstab and filter the results;
- make a calculation;
- draw conclusions.

Benchmarking, Trending and Comparative Data

Comparison is a logical trick necessary in any cognitive activity: at its various stages and levels, outside depending on its object. The comparative method is a narrower concept.

As a special method of research comparison can be performed only when the procedure comparison requires - for its effective implementation - special training and organizations.

This need arises usually when comparing complex objects and phenomena that are described by a large set of widely varying features.

Benchmarking works this way - setting a benchmark and moving forward, it can be seen whether that metric has changed, and if so, how.

Working with numbers

Knowing how many participants completed the survey, but how it can be known whether the answers to this survey can be trusted and can data from answers can be

confidently used to inform future decisions? It is important to be able to consider the quality of the data and to know what constitutes statistical significance.

Drawing conclusions

When the report on the survey results needs to be prepared, analyzation about what the data says should be done. One aspect of data analysis and reporting that need to be considered is causation and correlation.

1.2. Assignment of the Task

In order to provide a transparent and simple interface for interaction between the program modules of the web application, it is necessary to ensure the separation of functionality between the modules.

Since the solution of the problem requires the development of an interactive interface for user interaction, the client side of the web application will need powerful tools to display the interface.

Accordingly, in order to isolate this functionality in one module, it is necessary to concentrate all actions to form the interface display in the client application. The approach to creating a single page application was chosen as a solution.

A single-page application means that most of the logic moves into the client application module: processing data for display, displaying, processing user events, sending data to the server. Interface for the client module consists of methods for receiving and sending raw data without displaying process.

The following technologies were chosen to solve the problems of the client application:

- React library for working with UI rendering;
- GraphQL library as a library for making requests to the back end of a web application and creating an architecture for processing internal actions in the client application.

The main tasks of a server application when implementing a single page web application approach are:

- interaction with the database;
- processing requests of the client application and execution of the business logic of the application.

The following technologies were chosen to solve the problems of the server application:

- Node.js - a platform that allows JavaScript to be executed on the server and provides opportunities for interacting with the server's system resources;
- Express.js - a framework that allows to implement a web server, routing user requests, and decompose program logic into separate subsystems.

The interaction between the database and the web server should be organized following the principle that business logic is in the code of the server application, not in the database.

In this case, the database stores the data and provides direct access to the data, while all the business logic is implemented in the subsystems of the server application. The database allows to perform transactions to perform atomic operations on data.

Isolation of business logic in the subsystems of the server application provides a transparent interface for interaction between the server and the database.

To provide such isolation, the following hierarchical task structure can be defined:

- development of the module for the client part of the web application;
- development of the module for the server part of the web application;
- database module;
- protocol of data exchange between client and server sides;
- designing the interface of interaction between the modules of the client and server parts;
- designing the interface of interaction between the server module and the database.

One of the goals pursued in the automation of data analysis is the creation of a single information storage. The presence of a single information storage center in order to minimize the functions of individual users is an equally important task when creating automated systems.

Also, one of the requirements for the system being developed is the creation of a convenient user interface that is not overloaded with unnecessary information and functionality.

It provides easy perception and processing of information, providing a free output of information in a more familiar form for employees of the enterprise, such as in document format.

The common assignment of the task is to design and develop a project, that will be suitable and comfortable to work both with the small amount of data required for concrete issue and with a huge amount of data.

In addition, the data analysis system should provide differentiation of the information provided, depending on the role of the user, be resistant to errors and negative input data.

1.3. Problem Definition

A prerequisite for the social survey data analysis system creating was the necessity to create a multifunctional analysis system that supports a variety of algorithms for data analysis.

At the moment, in order to analyse received data from surveys there are some software solutions exists. But monitoring their functionality it was noticed that they are large giants capable of performing all possible types of analyses and operations on data. Because of that, they are very voluminous and consume a lot of system resources, and also require special training of personnel to work with them. Using such products in for surveys is not really appropriate.

Therefore, the purpose of the project is to develop a compact software product sharpened for solving a specific problem and working with the minimum amount of data required for specific purpose, and with very large amounts of data.

Having developed a software product that will provide the opportunity for multifunctional analysis and process large amounts of information, users will be able to analyse data from different aspects, which will help users to determine new dependencies and patterns.

A new software product with such capabilities will solve the problem of individuals or small companies with the analysis of accumulated data statistics, without spending additional funds on purchasing additional software products.

1.4. Analysis of the Existing Developments

The demand for the automated systems for surveys analysis are growing on the information technology market, which means that the supply is also growing. But the functionality of automated control systems is very different.

Systems such as 'Foquz', 'SurveyMonkey' and 'WebAsk' were reviewed and explored.

Foquz

One of the main advantages of 'Foquz' is that it is possible to send questionnaires to customers of an online store for a selected event. For example, a trigger can be the fact of making a purchase or ordering a specific product. This functionality allows to automatically collect feedback on the thin areas of customer service, assess customer satisfaction with the quality of the goods and much more. The service is also suitable for conducting traditional surveys and tests.

The service offers to implement customer-oriented marketing. It is tailored to work with complaints and negative comments. For each negative review, manager can track the progress of solving the problem.

Advantages:

- modern visualization;
- automatic mailing by triggers;
- fully functional mailing system;
- implementation of customer-oriented marketing.

Disadvantages:

- the free mode is very limited;
- few ready-made questionnaire templates;
- tools oversaturation.

SurveyMonkey

The service was one of the first to provide services for creating online surveys. It has a huge number of users all over the world. There is a full-fledged multi-language localization.

The finished questionnaire can be published on the web resource page or placed in a pop-up window. The functionality of sending a survey by mail, posting on a social network is available. The system provides many analytical tools for analyzing customer service and the selected niche. Statistics are built in real time.

There are free and professional versions. When using free of charge, the number of surveys is not limited, but each can contain no more than ten questions. Some analytics features are also available. The setup and usage are cumbersome which can be difficult to customers to work with platform.

Advantages:

- lots of possibilities in the free plan;
- there is a mobile application;
- integrates with store automation systems.

Disadvantages:

- complex customization;
- most of the useful features are only available with professional licenses.

WebAsk

A simple online builder for collecting feedback and conducting market research. The product is available in full for free. The platform allows to create questionnaires based on fourteen types of questions.

There is a database of ready-made templates. User can insert media files into polls and carry out branding. The questionnaires are proposed to be published on the website or in the application, the user cannot send polls directly by e-mail – the user can only send a link.

Response statistics are available in real time, but the service does not notify about the arrival of new completed questionnaires. The collected data can be downloaded in different formats for further analysis.

Advantages:

- fully functional free mode;
- adaptive layout;
- the ability to limit the survey by date or number of responses;
- logic and branching support.

Disadvantages:

- weak functionality;
- lack of mailings;
- inflexible polling logic.

1.5. Feasibility Study

A feasibility study is an analysis, calculation and assessment of the feasibility of a specific project, the creation of a new technical facility, modernization and reconstruction of existing processes.

The feasibility study is based on a comparative assessment of costs and benefits, establishing the effectiveness of use. A properly compiled feasibility study shows the effectiveness of developing a new project or revision of existing systems, to understand the market needs for a product and whether there is a need to develop this project.

The feasibility study also helps to select the necessary equipment, select and implement suitable production technologies, and correctly organize development activities.

The feasibility study of the current system is based on the following main sections:

- general conditions for the implementation of the system and its initial data – the authors of the project, initial data for the project;
- design documentation - preliminary determination of the scope of the system, production technology and equipment;
- system timeline planning - estimated timeline for project implementation;

For realization of the social survey data analysis system no special conditions are required. As for equipment, in order to develop the current web-application it is enough to have a powerful computer and fast internet connection.

The estimation for the project realization due to functional, performance and security scope is four months.

1.6. Performance Requirements

Performance requirements determine the effectiveness of the function's execution under the pointed set of conditions. There are a lot of acceptance points, which should be reviewed on the performance issues.

For example, load testing is used for checking the speed of how long the application runs under the expected user loads. The aim of load testing is to find dangerous parts before the application deployed.

The following list of performance characteristics should be taken care of while developing the system:

- user-friendliness: the system functionality and design should be intuitive understandable so new users should not have any problems with onboarding;
- user satisfaction: the system should match the user expectations. If the performance results do not coincide with the requirements, then it should be optimized;
- response time: the system should react to the user's actions as quickly as possible. Ideally, the callback should be executed immediately. For some asynchronous actions that cannot provide feedback in a nutshell, a load UI element must be added during action processing.

1.7. Functional Requirements

Functional requirements define the functionality of the software. They are sometimes called behavioral requirements.

Before developing the system, the functional requirements should be carefully prepared and described. The following requirement are defined for the current system.

The system should have the platform menu in the header and check whether users have access to surveys and dashboard.

The user should have a possibility to see the list of questions and pages which he has in survey. Clicking on the page, user should see detailed statistics for each question on the page. Besides that, there should be a button, clicking on which user can check information for all questions on the page.

User should be able to see general statistics and charts of the survey or for each question. Also, there should be the possibility to sort questions by number, most upvoted insights, most downvoted insights or comments amount.

System will have a functionality to download survey question dashboards and charts with data in .pptx, .jpg, .jpeg, .png formats.

There should be a possibility to export all questions as a .pptx slide deck in a standardized format.

Question block should provide the general information related to a question and a statistical chart that displays survey data. User will have the possibility to sort data bars via drag and drop, change chart axes and keys, add average lines, export chart to .png., add confidence intervals for bars and values and show data as percentages.

For each question user will see the insights list and their information as comment count, upvotes count, downvotes count, as well as all commentary related to that question.

The functionality of displaying questions statistics on chart should be presented. On the chart user will see answers and their percentage to total answers number.

User should be able to see insights list and comments attached to the insight. The functionality of the system should also allow to sort insights by comments, upvotes, downvotes and rating. After clicking on the insight item, user should see all comments attached to current insight.

The system should have the comments list for the insight item which user have chosen. User will be able to send a comment on an insight. Also, the application should have functionality to allow users to edit and delete only those comments that they posted. User should have a possibility to create insights for the questions.

User can add and edit topics for this insight. The functionality to select topics should be included in the system. Then user will see the questions with these topics on the screen. He will be able to change sort of topics by drag and drop.

The functionality to create, delete and update new topic should be considered while developing the system. Adding the topic, user should be able to connect it to the questions from the list.

In order to make a review, there should be a functionality for filtering questions and insights by parameters user has chosen.

1.8. Interface Requirements

When developing a user interface, it necessary to be guided by the UX approach and the tasks that the interface should solve.

UX is a term that describes the degree of user satisfaction from using the product.

Tasks which should be solved by the user interface:

- user authorization;
- granting user access to the resource and personal account;
- possibility to be tested;
- displaying the result.

As a result, the main sections of the application were highlighted:

- dashboard page;
- login page;
- personal account with information about the user;
- survey page;
- question analysis page.

For the direct development, there should be used component approach.

The component approach is a programming paradigm that essentially relies on the concept of a component - an independent module of program code intended for construction. For these purposes, the React.js library is ideal because it supports the component approach.

There should be uniform, synchronized data between the sections and in the application as a whole. For this purpose, the schema of several design patterns will be used. Application model, user interface and user interaction are divided into three separate

components. So that modification of one of the components has minimal impact on the others. This schema forms a MVC binding.

The main purpose of applying this concept is to separate the business logic from its visualization.

This separation increases the reusability. The following tasks are performed:

- it can be attached several views to one section, without affecting the implementation of the model. For example, some data can be presented simultaneously in the form of a spreadsheet, bar chart and pie chart;

- it can be changed the reactions to user actions – clicking on a button, entering data without affecting the implementation of views. To establish that it is enough to use another controller;

- a number of developers specialize in only one of the areas: either develop a graphical interface or develop business logic. Therefore, it is possible to ensure that programmers developing business logic will not be aware of which view will be used at all. Within the scope of this project such a problem cannot exist in the view of the sole developer of the application. But this is a solution to possible problems in the future when the project will be supported by several developers.

1.9. Security Requirements

Besides fact that the system should be convenient to use, it also has to provide security to users and their private data. In order to supply the system safety and reliability the developer should:

- add logic to validate the input value;
- provide blocking wrong user actions which are out of the scope;
- ensure the integrity of stored data;
- exclude the possibility of the unauthorized access to the information.

The peculiarity of hacking resources that it is automated, but not personalized. It carried out in large quantities using special programs. The owner of resource which works

with clients on the model of a web application must be able to protect the project from the most common methods of hacking.

Before developing a methodology for protecting a web application from potential threats, the site should be checked for vulnerabilities. The check is carried out manually or automatically.

The programs will test the application for major risks. These software products exist in two versions: black hat, simulating the actions of hackers, and white hat, systematically revealing all system flaws by scanning.

Some of the most effective tools are:

- scan local networks for vulnerabilities;
- looking for the possibility of injecting malicious code into a web page that steals user account data and other information. The code is injected through vulnerabilities on the user's server or device;
- examine the configuration of a web application and find redundant or malicious code;

Working with web services requires the use of a wide range of security tools. In addition to the main listed methods, the following are often used:

- passwords encryption;
- avoiding cross-site scripting;
- control of uploading files to the server.

The combined use of all available solutions will provide security at the maximum possible level.

Conclusions on the First Part

The purpose of the project is to develop a compact software product sharpened for solving a specific problem and working with the minimum amount of data required for collect and analyse data purpose, and with very large amounts of data.

A new software product with capabilities to collect and analyse data will solve the problem of individuals or small companies with the analysis of accumulated data statistics, without spending additional funds on purchasing additional software products.

The demand for the automated systems for surveys analysis are growing on the information technology market, it means that the supply is also growing. But the functionality of automated control systems is very different and not always match with the individual's needs.

Before developing the system, the functional requirements should be carefully prepared and described. Then, while developing the requirement which were defined for the current system should be followed.

While developing the system the performance issues should also be considered. Performance requirements determine the effectiveness of the function's execution under the pointed set of conditions. There are a lot of acceptance points, which should be reviewed on the performance issues.

When developing a user interface, it necessary to be guided by the UX approach and the tasks that the interface should solve. The main purpose of applying this concept is to separate the business logic from its visualization.

Besides fact that the system should be convenient to use, it also has to provide security to users and their private data. In order to supply the system safety and reliability the developer should add logic to validate the input value, provide blocking wrong user actions which are out of the scope, ensure the integrity of stored data, exclude the possibility of the unauthorized access to the information.

Before developing a methodology for protecting a web application from potential threats, the site should be also checked for vulnerabilities. Some of part of that task can be done manually and another one using special software products.

One more requirement for the system being developed is the creation of a convenient user interface that is not overloaded with unnecessary information and functionality.

It provides easy perception and processing of information, providing a free output of information in a more familiar form for employees of the enterprise, such as in document format.

The common assignment of the task is to design and develop a project, that will be suitable and comfortable to work both with the small amount of data required for concrete issue and with a huge amount of data.

The importance issue is to create secure application without any vulnerabilities. It should be received by combining use of all available solutions to provide security at the maximum possible level.

PART 2

OVERVIEW OF USED TECHNOLOGIES. JUSTIFICATION FOR CHOOSING A TECHNOLOGY STACK

2.1. HTML

HTML - is a standardized markup language for documents on the World Wide Web. The bulk of web pages contain descriptions of markup in HTML.

HTML is interpreted by all browsers which are known. The resulting rich text is displayed on our computer or mobile screens. On the World Wide Web, HTML pages are transmitted to browsers from a server using HTTP or HTTPS protocols in plain text or using encryption.

The HTML language also has its own standards and versions, and at the moment the current version is HTML5. Although not all browsers still support the vast arsenal of features of this version of the language. This version of the language implements many new syntactic features.

There are new elements video, audio, canvas, as well as the ability to use SVG and manipulate graphic and multimedia objects without the need for third-party APIs or plugins. The section, article, header and nav elements have been added to enrich the semantic content of the web page.

Changes have been made to some of previously added elements. This time, the main body of the HTML5 specification is the API and DOM.

2.2. CSS

CSS determines how the document is look. With CSS, it is possible to define page color, font, block placement, and other aspects of how pages are presented.

The existence of CSS helps to separate the creation of the document structure from its presentation.

This separation increases the availability of the document, provides more flexibility and control over its presentation. In addition, CSS allows to present the same document in different styles and output formats.

Styles can be connected to a document in several ways:

- from a separate file using the link element between header tags;
- using the import derivatives between the style tags in the header block. It is also possible to connect an external file;
- a style sheet can be described in the html document itself in style tag in the header block;
- styles is also can be added to each tag directly in the document through the style attribute.

In current project it is taken approach that styles should be separated from the document structure as much as possible, so they will always be placed in a separate file.

Moreover, with a modular approach, each component will have its own file, which will be located directly next to it in the project structure.

Styles are connected to the components using CSS modules, which are imported into components as files and used as an object. Using the webpack builder styles they will be transformed by a number of plugins and added to header between style tags.

2.3. Javascript Programming Language

The structure of the page can be created with HTML, styled and decorated with CSS and with modern CSS3 capabilities. It is even possible to add animation to an element or make it change styles on hovering over them.

But it is impossible to create more complex dynamics using only HTML and CSS. Javascript is used exactly for that purpose.

Javascript was originally created to make the website live. The code written in this language is called a script.

Javascript, by analogy with CSS, connects to HTML and when the page is first launched in the browser then script is triggered.

Javascript language's capabilities are determined by the environment where it runs. When a script is running in a browser, it can manipulate the page, its elements, interact with the user and with the server.

Javascript features:

- creating new HTML tags, deleting existing ones, changing element styles, displaying an element and other manipulations with page elements;
- response to visitor action, handling mouse clicks, tracking the movement of the cursor, clicking on the keyboard and more;
- sending requests to the server and loading data without reloading the page. The AJAX technology;
- receiving and setting cookies, requesting data and displaying messages on the screen.

Javascript is unique due to three main aspects:

- full integration with HTML and CSS;
- simple work with all functionality;
- supported by all browsers.

Javascript is an implementation of the ECMAScript standard. ECMAScript is an embeddable, extensible programming language used as the basis for building other scripting languages.

Browsers are currently supported mainly ECMAScript5. But nothing stops developers from using the more modern ECMAScript6 standard, even if some features are not yet supported by the browser. The well-known Babel transpiler is used for these purposes.

The Babel.js transpiler rewrites the ES-6 code into the previous ES-5 standard, which is more stable.

Babel.js can be divided into next parts:

- a transpiler that rewrites the code;
- a polyfill that adds additional methods.

Babel.js integrates very easily into the project through whatever build system is used. In current case, this is webpack.

2.4. Single Page Application Concept

Single page application (SPA) - is an application that runs in the browser and does not reload the page while it is running. Like any other application, it is designed to help the user with the task at hand, for example: prepare a document or administer a web server.

Any server technology can be used in SPA. Since a significant portion of the web application is moved to the browser, the server requirements can be significantly relaxed.

Figure 2.1 shows the differences between the approaches for implementation of a regular website and a single page application. As it can be seen from the diagram, in SPA, most of the work with data goes from the server to the client.

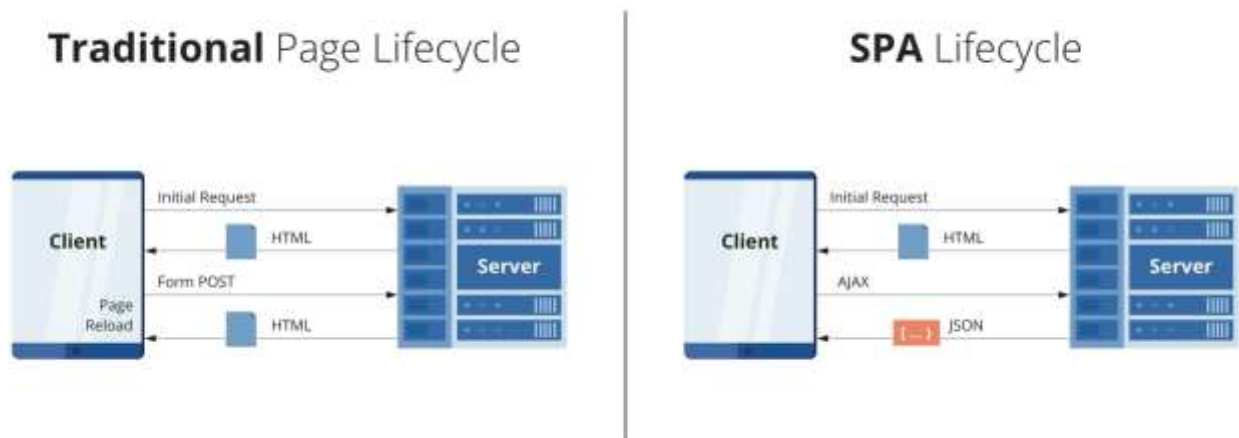


Fig. 2.1 Comparison of the performance of a single page application and the traditional approach

The SPA is rendered as a personal application. It only redraws those parts of the interface that have changed, and only when it is necessary.

In contrast, a traditional site redraws the entire page in response to various user actions, resulting the delays and blinking as the browser has to retrieve the page from the server and draw it on the screen.

If the page is large, the server is busy or the Internet connection is slow, then the delay can be several seconds, and the user can only guess when it will be possible to work with it again. This is a big drawback compared to the fast rendering and instant feedback of the

SPA. SPA can react like a personal application. It minimizes response times by transferring temporary data and part of the processing from the server to the browser.

The SPA has the data and business logic at its disposal to make most decisions locally and therefore quickly. Only user authentication, validation and persistence of data should remain on the server. In the case of a traditional site, most of the application logic resides on the server, so the user must wait until the request-response-redraw cycle is complete to get a response to their actions.

This can take a few seconds, while the SPA's response is almost instantaneous. The SPA can notify the user of its state, just like a personal application. If the SPA does have to wait for a server response, it can dynamically update the progress or busy indicator so that the user is not intimidated by the delay. When working with a traditional site, the user is forced to guess when the page has finished loading and it will be possible to interact with it.

SPA, like the site, is almost always available. Unlike most personal applications, a user can access a SPA with only an Internet connection and a decent browser. Nowadays, it's all on smartphones, tablets, TVs, laptops, and desktops.

SPA, like the site, is updated and distributed instantly. The user doesn't have to do anything at all to benefit from this feature - just restart the browser and everything works. There is no difficulty in maintaining the relevance of numerous programs. A personal application usually needs to be downloaded and then installed with administrator rights, and the interval between release of versions can be several months or even years.

SPA, like the multipage application, works on the different platforms. Unlike for most personal applications, a well-written SPA can run on any operating system that has a modern browser with HTML5 support. This feature is usually considered a developer benefit, but it is equally important for many users with multiple devices, like Windows at work, Mac at home, Linux server, and Android phone.

SPA can offer the best of both worlds - instant response from a personal application along with website portability and accessibility. JavaScript SPA is available on over a billion devices that support modern browsers and do not need third-party plug-ins. With a little effort, you can make it work on desktops, tablets, and smartphones with different operating systems. SPA is easy to update and distribute and usually requires no user action.

2.5. Technologies and Platforms Selected for Creating the System

The choice of technologies for developing an application is an important stage in a project. A well-chosen combination of technologies should ensure comfortable work in the future at all stages of the application's existence:

- support convenience;
- scalability.

The choice of a technology stack must be approached very carefully and responsively. It is necessary to look far ahead into the future and predict the potential development and fate of the project. Obviously, the stack should be easily scalable, functional, correspond the latest market trends. It should meet the most modern features. Most importantly, it has to be easily supported in the future by other developers. The presence of a large community of developers in the world for this or that product in the stack and open source code are huge advantages that should be cared. Also, these technologies should not contradict each other. Their existence together should be harmonious and justified. So, for example, GraphQL is great for generating queries in JavaScript. It also uses a data schema that is easy to use in JavaScript.

Guided by the above factors, as well as listening to the opinions of leading developers of the web technology market, there were identified the following technology stack:

- React.js library;
- GraphQL library;
- Webpack - the utility is great for modular building of a web application;
- npm package manager for managing modules and dependencies.

2.4.1. React.js Library

React.js has a capacious and understandable API. To work with React, it is necessary to understand a number of terms and the differences between them. Elements are JavaScript objects that are HTML elements. Components are elements of React.js that are created by the developer and can have any name. As a rule, they contain their own specific structure and perform a number of functions. React elements and components are built using JSX. JSX is a deep JavaScript syntax that looks like XML.

React creates an analogue of the real DOM tree from components - VirtualDOM and presents it in the browser. The library monitors changes in the virtual tree and, when it changes, updates the real DOM so that the real and virtual tree are the same. (fig. 2.2)

Above are the basic concepts it is needed to know to get started with React.js. Also, there is good to mention about the existence of state in React.

In practice, in the project it will be implementing a state of the application using GraphQL. React and GraphQL is a fairly successful alliance that occurs quite often.

Components have a life cycle such as mount, update and unmount. The library provides the ability to define various points in the life cycles of components and interact with them. When the first use of the component, call the lifecycle methods in this order:

- constructor;
- getDerivedStateFromProps;
- render;
- componentDidMount.

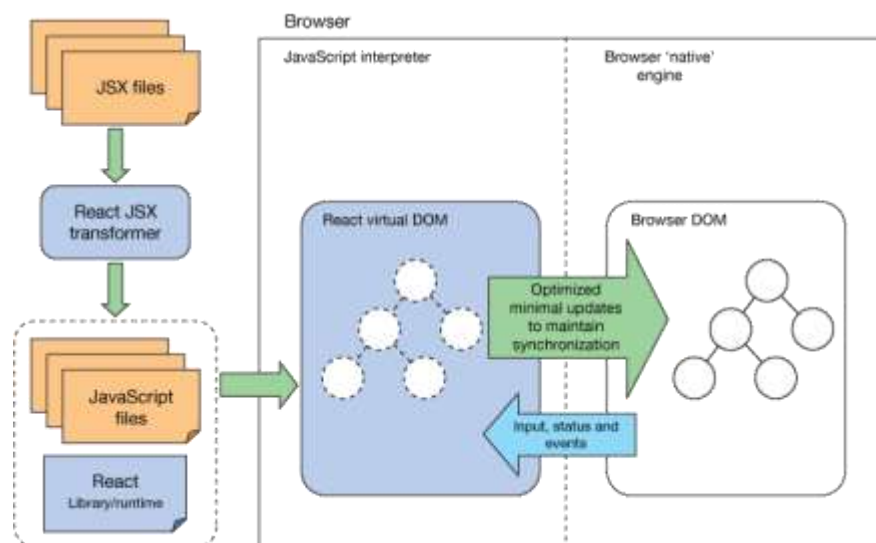


Fig.2.2. React operations

Why React.js Library was integrated:

- The content is referenceable

This is the functionality that makes the difference compared to other frameworks. Thanks to the use of a Node server, the code will be able to be generated on the client side and on the server side. Unlike other traditional JS frameworks which natively execute code only on the client side in the browser.

- React.js is very fast

React.js creates its own virtual DOM where the components are attached. This approach gives a tremendous amount of flexibility and exceptional performance, as React.js calculates which changes in the DOM needs to be made, and just changes the part that needs updating. In this way, React.js avoids expensive operations in the DOM.

- Components are the future of web development

React.js took the concept of Shadow DOM and the Polymer.js framework and took it to the next level.

React.js doesn't use Shadow DOM instead it gives the ability to create the components that can later be reused, combined, and included in the core content. This functionality alone is a guarantee of productivity by the ease of defining and manipulating developer's own components.

- Intelligibility

React.js produces easy to read code, reading it immediately determines what the functionality of the application is. That is essential for the maintenance and expansion of the project over time.

- Javascript is easier to write

ReactJS uses a special syntax called JSX, which allows to mix HTML and JavaScript. This is not required it can still be written the React.js app in native JavaScript but this new syntax allows to write the components very easily. Being able to put a touch of HTML in the rendering functions. And after a while it becomes very natural.

On the image below there is an official logo of React.js library. (fig. 2.3)

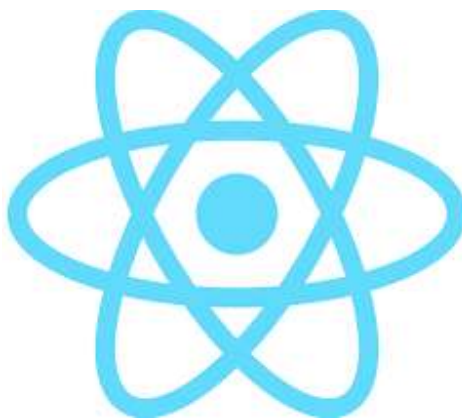


Fig. 2.3 React.js Logo

2.4.2. Typescript Programming Language

The TypeScript language is one of the most popular technologies of recent years, both in Frontend and Backend development. Its popularity continues to grow and it is at the heart of many projects: Angular, NativeScript, Ionic, VS Code, Apollo GraphQL, Babylon.js, RxJS, Nest, TypeORM, etc.

There is currently no alternative, although some technologies share common goals or offer similar benefits. For example, Flow - the type checker developed by Facebook.

TypeScript is an opensource language, created by Microsoft, stackable in JavaScript. On the image below there is the logo of Typescript.js library. (fig. 2.4)



Fig. 2.4 Typescript.js Logo

Its main objectives are:

- support for existing and future EcmaScript proposals;
- the contribution of optional typing to JavaScript;
- the early identification of potentially invalid codes;
- compilation to optimized JavaScript, with a target choice: ES3, ES5, ES6 and next ones.

What TypeScript does not aspire to do, among others:

- imitate existing languages; but rather exploit the nature of JavaScript and the uses of developers as a guide to make the language relevant;
- use a sage type system.

TypeScript's type system is not wise, which means that it allows certain operations that cannot be verified at compile time. This is one of the big major differences with Flow that can be considered.

2.4.3. GraphQL Query Language

GraphQL is a query and data manipulation language for APIs, as well as an environment for making those queries. The language was developed in 2012 by Facebook for the internal needs of the company.

In 2015 it was released to the public, and since November 7, 2018, it is not Facebook that has been working on it, but the GraphQL Foundation. Of course, the project has been developing quite actively since 2012, but it earned particular popularity after it received the opensource status.

Today, GraphQL is used in many popular applications. First, the social network Facebook. The GraphQL is used in products such as Airbnb, GitHub, Pinterest, Shopify, New York Times and many others.

The fact that it was created specifically on Facebook for a project with a large amount of heterogeneous data ensure that working on a product it should not arise with the limitations of the REST architecture.

For example, getting a user's profile, posts and comments does not initially seem difficult. But if it is considered the amount of data in the system and assumed that all this data is stored in different databases (for example, MySQL and MongoDB), it becomes clear that this will require creating several REST endpoints.

Imagining how large the volume of data and heterogeneous data sources are, it becomes clear why it was necessary to develop a new approach to working with the API. This approach is based on the following principle: it is better to have one smart endpoint that will be able to work with complex queries and return data in the exact form and volume that the client needs.

At the heart of any implementation of the GraphQL API is the data schema - this is a description of what data types it can work with and what data types it can return in response to a request - they are described in the GraphQL type system.

To work with any API, the developer needs to know what types of objects can be obtained, what fields to select, what fields are available in internal objects and more. The GraphQL schema is showed on figure 2.5. Working with the GraphQL API, the client does not care at all where the data comes from.

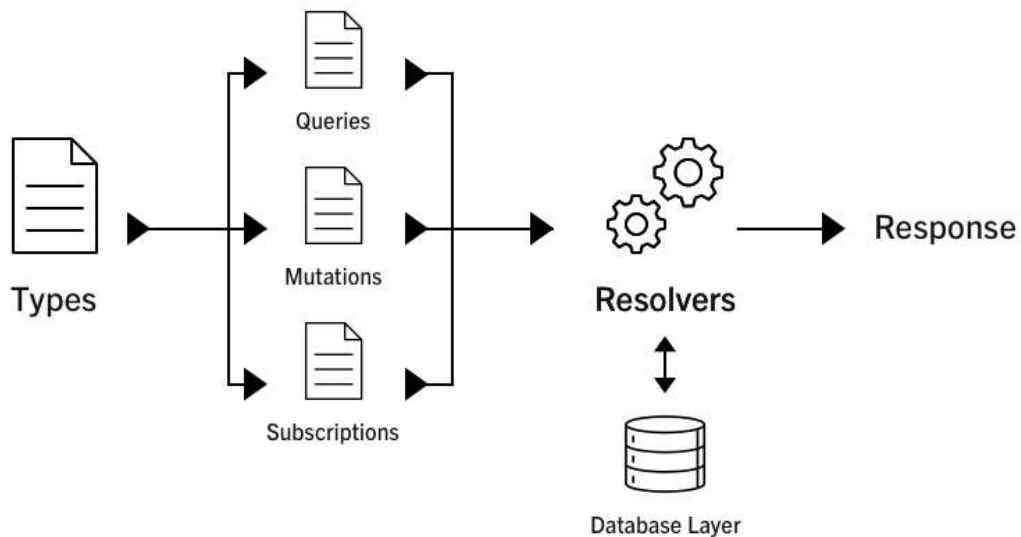


Fig. 2.5 GraphQL schema

It just should be made the request to the extent is needed, and the GraphQL server returns the result. Therefore, it can be concluded that the schema is a contract between the API and the client. Before the client makes any request, it is validated in accordance with the schema of the given API.

Many platforms now support GraphQL: web, Android, iOS and others. The GraphQL client sends a request to receive data or to change it, composed in accordance with the schema, to the GraphQL server.

The GraphQL server is the HTTP server to which the GraphQL schema is associated. It means that all requests received from the client and returned responses are passed through this scheme.

The GraphQL server cannot know what to do with a request unless it is explained to it using special functions. Due to them, GraphQL understands how to get data for the requested fields.

These functions are associated with the corresponding fields and are called resolvers. A response is returned to the client that reflects the data structure requested from the client, usually in JSON format.

It is possible to work with completely different data sources: databases (relational / NoSQL), web search results, Docker, and other. On the image below there is an official logo of GraphQL library. (fig. 2.6)

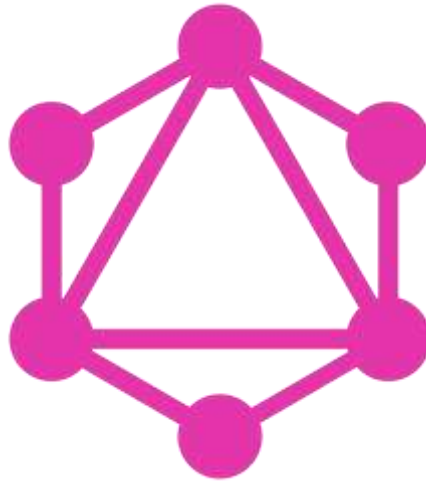


Fig. 2.6 GraphQL Logo

2.4.4. D3.js Library

D3.js is a JavaScript library for data processing and visualization. The name D3 itself stands for Data-Driven Documents and, as it were, focuses on data management, although the key functionality of the library is its powerful visualization capabilities.

The D3.js library is primarily based on JavaScript, SVG and CSS, as opposed to other similar libraries that use the canvas element and its capabilities instead of SVG.

While standard rendering engines like the canvas element rely on pixels, svg uses vectors. Using SVG allows to create graphically rich structures with animation and interoperability. Compared to pixel art, SVG has several advantages.

In particular, SVG is based on xml, which makes it more readable. In addition, SVG code is more lightweight than image files. Largely due to this, D3 is currently one of the most popular frameworks used for graphical data processing and creating all kinds of charts and graphs. On the image below there is an official logo of D3.js library. (fig. 2.7)



Fig. 2.7 D3.js Logo

Unlike other similar JavaScript libraries, D3 does not use jQuery to work with the DOM structure, although at the same time it implements similar concepts for working with elements.

2.4.5. React Testing Library

One of the convenient solutions for unit testing of components is the react-testing-library built on top of react-dom and react-dom/test-utils. It provides utility functions on top of react-dom. Tests run on DOM nodes, not React component instances. The main idea of the library is to bring test scripts closer to using the components the way the user does, which allows for confidence, giving more confidence when the application hits production. It is a lightweight solution for testing React components.

The Jest test framework will be used to run the tests. It is built in such a way that it requires almost no configuration, it will not be difficult to install it in a new project and start using it, moreover, it is part of create-react-app. It is also needed helper library that provides custom DOM mapping for Jest. On the image below there is an official logo of React Testing Library and Jest. (fig. 2.8)



Fig. 2.8 React Testing Library and Jest Logos

The react-testing-library contains all the necessary methods for testing components, simplifies testing and improves test readability.

2.4.6. Node.js Runtime Environment

Node.js is a server-side implementation of the JavaScript programming language, which is a big plus for developers who deal with both the client and server side at the same time. Node.js is a platform based on the V8 engine. The main purpose of Node.js is to create scalable distributed network applications - web servers. Unlike JavaScript scripts, Node.js is implemented on the server, not in the browser.

Node.js includes the npm package installer that makes it easy to install packages using the “npm install” command. It can be used the “npm search” command to find out what packages are available for installation and their brief description. All commands are executed in the console of the computer.

Due to Node.js, it can be implemented following tasks:

- HTTP server;
- routing;
- request handler - to fulfill requests that come to the server and are directed by the router;
- input data processor for input POST requests;
- display logic for sending content received from the request handler to the browser.

For more convenient work with Node.js, there is the Express.js framework, which is widely used in conjunction with Node.js. Express.js provides a rich set of features for mobile and web applications. Using the framework's HTTP service metrics and middleware, it can be quickly and easily created a reliable API.

Node.js has built-in package management support, which uses the NPM tool that is included by default in any Node.js installation. The idea behind NPM modules is in many ways similar to Ruby Gems: they are a set of opensource, reusable components that are easy to install through an online repository. They support versioning and dependency management.

One of these modules is the Express.js framework. In conjunction with it there was decided to create a server.

Node.js with Express.js can be used to create classic web applications on the server side. However, even if it is possible, such a request response paradigm, where Node.js will carry the HTML display, is not typical for this technology.

There are arguments both in favor of this approach and against it. The main points were considered:

- if the application is not computationally intensive, it can be written entirely in JavaScript. Even including a database if it is used an object database (for example MongoDB). This makes development much easier;

- search engine crawlers receive fully rendered HTML in response, which is much more convenient for search engine optimization than, for example, working with single page applications or a web socket application powered by Node.js;
- it is not convenient to use Node.js with a relational database, in this case developer should give preference to another framework, for example, Rails or Django.

2.4.7. MongoDB Database

For a more accurate understanding of the MongoDB database, several concepts can be distinguished:

- MongoDB database can store zero or more data, each of which can be a container for other entities;
- the database can store zero or more collections. Collections in MongoDB are analogous to tables;
- the collections themselves can store zero or more documents. Documents correspond to the concept of string;
- documents consist of fields, which are similar to columns;
- indexes are almost identical to their counterparts in relational databases;
- it is necessary to understand that after making a query in MongoDB, developer get a cursor which we can be performed for arbitrary operations. For example, count data or skip a specified number of records. All this will happen without downloading the data itself.

Unlike traditional DBMSs, MongoDB cannot do joins, in MongoDB terminology - collections. Therefore, MongoDB is suitable for loosely coupled or loosely structured data. That says about the optional strict data structuring.

The table 2.1 represents the differences between MongoDB and the usual RDBMS:

Table.2.1. Differences between MongoDB and RDBMS.

DBMS	MongoDB
1	2
Database, Data Schema	Database

1	2
Table	Collection
Data Row, Tables	Document
Column row	Document field
Cursor	Cursor in Stored Procedures

Considering the fact that nothing binds to specific schemes, it can be gotten by just one collection, which includes many documents of different structures. There are several main strategies for developing databases for MongoDB:

- without embedding. Allocation of a separate collection for each data type. For example, a separate collection for email and a separate one for users;
- everything is built in. Embedding all data in one document;
- partial embedding. The most optimal option. The ability to store data in a separate collection, but if necessary, embed it in a collection that is logically related to this one.

To summarize, MongoDB includes databases, state from collections, which are made up of documents. Each document contains fields. Collections have the ability to be indexed, and as a result, it becomes possible to improve the performance of selection and sorting.

2.6. Webpack Bundler

Webpack is a fairly fast and powerful tool for building web projects. Its main feature is support for a modular system. Developer can organize any logic, any formats, quickly assemble a project, use many other third-party tools together with it.

Webpack building can be called in two ways: through the console and through the NPM scripts. The logic and behavior of how webpack will build the project and what transformations it will perform is described in its configuration file, which must be specified when starting webpack, or it needs to be given a default name of webpack.

In the configuration file, there is described which JavaScript file is the input and into which file there will be converted all JavaScript files. There is described what should be done with files of other formats when importing them into a project. For example, when importing files with the css extension, it is used plugins from the list of the postcss plugins.

Also, it is possible to create various configurations for build development mode and for public version. There will be one or another configuration assigned based on the environment variables.

Webpack has a number of built-in plugins as well as additional plugins that developer can easily install. Many of them help during development, and some of them are also needed public mode.

2.7. REST vs. GraphQL

GraphQL was designed as a more efficient alternative to the REST architecture for developing APIs. Therefore, these two approaches have common features:

- they provide a single interface for working with remote data;
- most often, a JSON response is returned for a request;
- both approaches allow to differentiate between read and write requests.

But GraphQL also has many advantages over REST:

- it is client-oriented, that is, it allows the client to receive the information and exactly in the volume that he needs - no more and no less;
- unlike REST, only one endpoint is needed to work;
- GraphQL is a strongly typed language that allows to evaluate the correctness of a query in advance before the program is executed;
- GraphQL provides the ability to combine queries;
- queries to the GraphQL API always return the expected result that matches the data schema of that GraphQL API;
- using GraphQL reduces the amount of data sent to the client, since the client only requests the data it needs.

Using REST the endpoint determines what data and in what form the client will receive, then when using GraphQL, the client should not request extra data, thereby reducing the server response. (fig.2.9)

In practice, using GraphQL increases the independence of front-end and back-end developers. After agreeing on the schema, the front-end developers will no longer ask to create new endpoints or add new parameters for the existing ones: the back-end developers describe the data schema once, and the front-end specialist creates requests and combines the data as it is necessary.

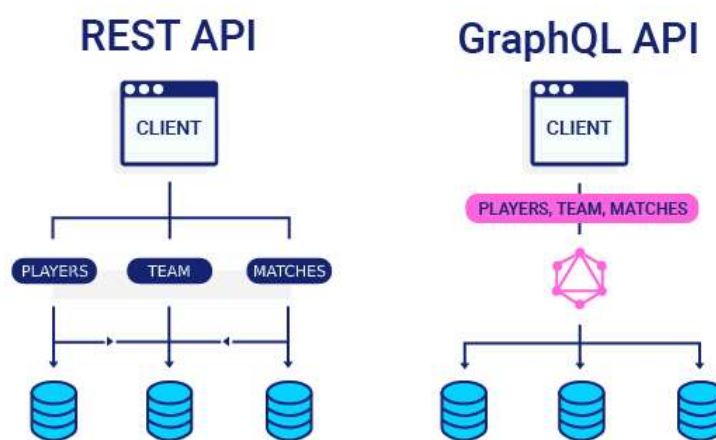


Fig. 2.9 The comparison of REST API and GraphQL API

2.8. NPM tool

NPM - is a tool for managing dependencies in a Node.js project, and it also provides the ability to perform a number of other useful functions. In addition, npm scripts are not a replacement for the bundler, but rather a perfect complement to it. Although it is possible via npm scripts developer can build a project, run a particular package, combine several files into one, transfer files from one folder to another, and a much more. It will be quite difficult to implement this only with npm scripts. This work mostly executes webpack, which was created specifically for the purpose of building a project. NPM scripts are great for starting the server immediately after installing all the necessary packages in the project, or for running a code style check or error using ESLint or StyleLint.

All scripts are described in the package.json configuration file. It contains the description of the project, its name, repository address, a list of all dependencies and information about the author. The scripts themselves are stored in an object called "scripts". To run the script, it should be written the npm run command in the console and the name of the script itself. There are also scripts that run after a certain action on their own.

For example, postinstall, which is called after all packages have been installed.

2.9. Charts. Types of Charts

A chart is a way to visualize data. Showcasing the data with a well-designed chart helps people better understand information and can often speed things up. In particular, diagrams are very useful for visualizing the information contained in large sets of numbers, to see how these sets are related to each other. Creating a diagram, it is possible to identify structure of the process and understand data in common, which is almost impossible to do with just a set of numbers.

The chart help to achieve such goals:

- quickly and unambiguously convey the idea;
- simplify and help to understand numbers.

2.8.1. Column Chart

A column chart is a data visualization where each category is represented by a rectangle, with the height of the rectangle being proportional to the values being plotted. Column charts are also known as vertical bar charts. (fig. 2.10)

A column chart is best for comparing values when it is important to be able to compare each value with each other.

Use case tips:

- used when less than eight values are compared;
- there should be indicated the type of period on the horizontal axis and the value of the period for showing changes over time;
- the start point for values always starts at zero;
- column chart is used to display the trend of a value over time if the values are

no more than twenty and each point has a clear, visible value.

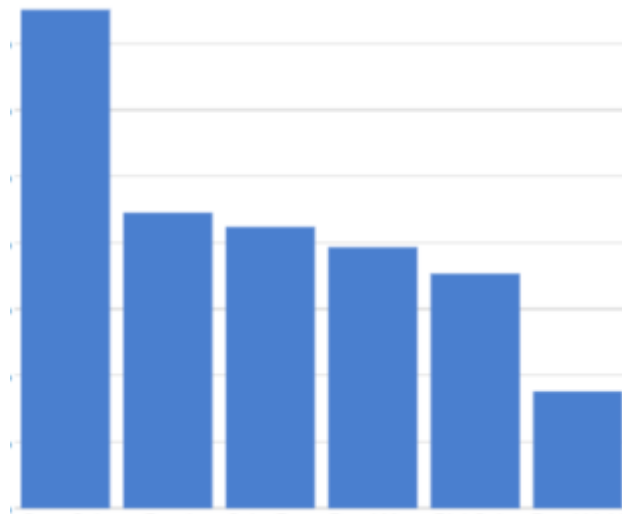


Fig. 2.10 Column Chart

2.8.2. Bar Chart

This type of chart arranges column horizontally. It gives more space for category names. (fig. 2.11)

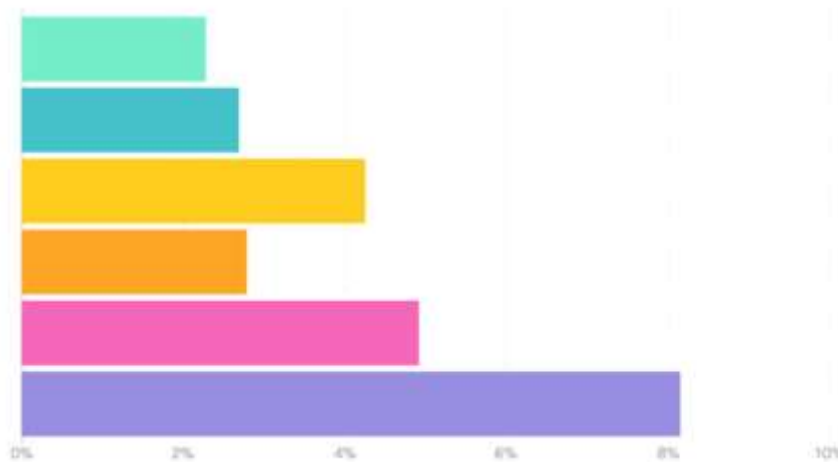


Fig. 2.11 Bar Chart

Bar Chart is good to use in these cases:

- there is a long list of values;
- there are negative values present.

2.8.3. Stacked Bar Charts

Stacked Bar Charts are useful unless there are present only a few variables, constituents, and the emphasis is on composition rather than comparison. (fig. 2.12)

Stacked Bar Charts are not suitable for comparison or analysis of relationships. The only common baseline is located along the left axis of the chart, so it can be reliably compared only the values in the first series and for presenting the sum of all the series.

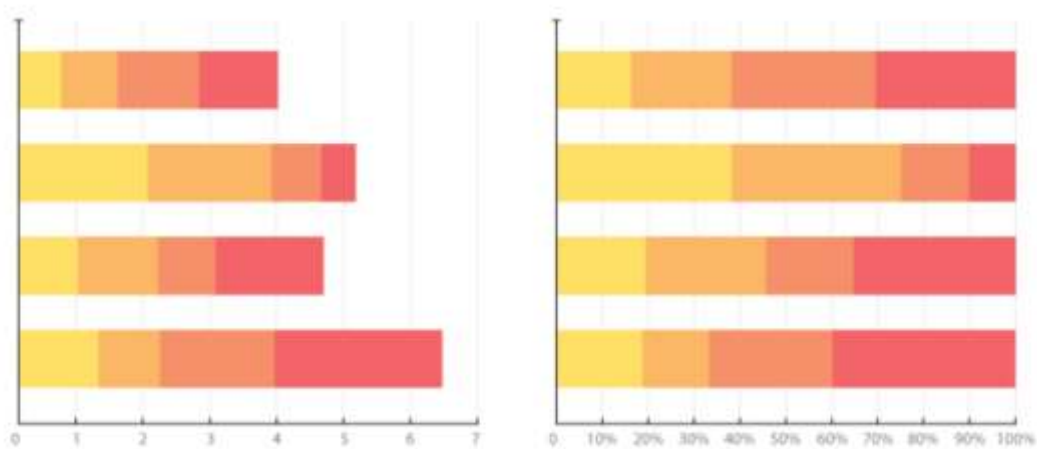


Fig. 2.12 Staked Bar Charts

2.10. Web Browser. Developer Tools and Extensions

The web browser has a number of synonyms: browser, Internet browse and others. The functionality of each browser is different. But most of them have great features: bookmarks, integration of search into the address bar, all possible extensions, and more.

Today, there are several major leading browsers on the market that web developers should focus on: Chrome, Internet Explorer, Edge, Firefox, Safari, Opera, iOS Safari, Opera Mini, Android Browser and Chrome for Android.

Many of them are equipped with developer tools, but the most comprehensive tool in terms of functionality belongs to Chrome Developer tools. Developer tools is equipped with live CSS editing on the fly, console, debugger. This is a small part of the Chrome Developer tool's capabilities. Chrome provides the ability to connect many useful extensions for development, for example:

- ColorPicker - for determining the color on any part of the page;
- PageRuler - to determine the exchange of elements;
- WhatFont - to define fonts;

- XDebug - for debugging server applications;
- React DevTools – for debugging react components;
- Testing Playground – for querying components to test;
- and a limitless number of other useful extensions.

For a number of obvious reasons, Chrome is my primary web browser for developing and debugging web applications.

2.11. Description of the Code Development Environment, Linters

2.10.1. IDE WebStorm

Choosing between a text editor and an IDE, the preference immediately was given to the latter.

IDE (Integrated Development Environment) is a software system used by programmers to develop software. The program includes all the functions necessary for the programmer. Among a number of IDEs for web developers, there is a specialized product from JetBrains - WebStorm.

JetBrains is world famous for its creation of convenient and fast development tools for Java, C#, Ruby, Python, PHP, Objective-C, JavaScript and many others, which definitely inspires confidence.

WebStorm is a JavaScript development environment suitable for both front-end developers and Node.js applications. The main advantage of WebStorm is a convenient and smart editor for JavaScript, HTML and CSS, which so supports many accompanying languages, the syntax of all libraries, frameworks, pre- and post-processors and templates necessary for development.

There are many features in the WebStorm arsenal. The following are main ones:

- built-in console;
- debugging code;
- live editor;
- integration with version control systems;
- integration with bug tracking systems (ESLint, JSHint, JSLint, JSCS, TSLint).

2.10.2. ESLint, StyleLint and Prettier code formatter

A very important aspect in developing and writing code is the style of the code. The style of the code applies to all languages with which we interact - HTML, CSS, JavaScript. Not always in the development process, we can concentrate on the style of the code, or pay attention to certain errors with which the code works properly.

In JavaScript, it could be:

- no semicolon at the end of a line;
- declared, but not used anywhere variable;
- invalid tabulation;
- using an incorrect declaration. Most often it concerns the confusion between “let” and “const”;
- a string exceeding the maximum character value in the line as defined by the code style;
- and many other errors. On the one hand, not critical, but violating the concept of clean code.

In CSS, it is:

- invalid tabulation;
- repeating properties;
- invalid nesting;
- using non-uniform color formats.

To delegate these tasks to the computer, it can be used existing linters that will monitor the style of the code and report possible violations. ESLint is for JavaScript and StyleLint is for styles, respectively. The configuration and code style rules can be determined by developer, guided by the documentation and common sense.

To connect and check code style it can be used the project builder or using npm scripts. In this project, as in many of my other projects, it is used the second method, invoking code style check when trying to submit code to the repository or when trying to create a commit.

Formatting the code consistently is challenging, but modern developer tools allow to automatically maintain consistency in the team's code base.

2.12. Git

Git is the system for controlling versions of the project. In essence, this means that it stores the entire history of project changes. The developer's history of the project and the history of changes in the same project for teammates, who also works on project - everything will have a copy. This is the complete opposite of SVN, where all change history is stored in one place.

Git tracks and stores the content. Most often it is used for program code. The code that is stored in Git can be changed and added. Often several programmers are working on the same project in parallel. The version control system keeps the history of changes and makes it convenient to work with it.

Git is a distributed version control system designed to control the Linux kernel development process. Git is a flexible, distributed version control system that provides many options for software developers who value change history management.

Git allows to share saved data with other users in the repository. Git has a central repository that multiple developers sync with, each provided with a local copy of their entire development history. In doing so, they can download updates from the central repository, make their own edits, and upload them back to the central repository.

Git provides fast splitting and merging of versions, work on a version can go in parallel in several branches, which can then quite easily be partially or completely merged, destroyed, rolled back or grown into new project branches.

The main advantages of this version control system include the following:

- the reliability of the revision comparison and data validation system, due to hashing algorithm;
- high performance;
- the flexibility of the branching and merging system;
- availability of a local repository with full information about changes;
- availability of graphical shells for working with git;
- availability of high-quality documentation;
- the ability to integrate with other version control systems;

- the ability to create checkpoints with fully saved data for quick data recovery;
- flexibility of the system, which allows you to customize it to your needs and create user interfaces based on it.

The main disadvantages of the version control system - Git include the following: focus on Unix-like operating systems; there is a small probability that the hash code will match with different revisions.

In addition, Git has handy things like branches and merges. Git has a remote repository that resides on the server, and a local repository on each developer's machine. This means that the code is not only stored on a central server. Each computer has a complete copy of it.

2.11.1. Forks

In order to get the personal use of any repository that does not belong to other developer, it is needed to use a process called fork. In fact, it works like developer creates a fork in the development of someone's project. Then one of its part appears in the developer's profile.

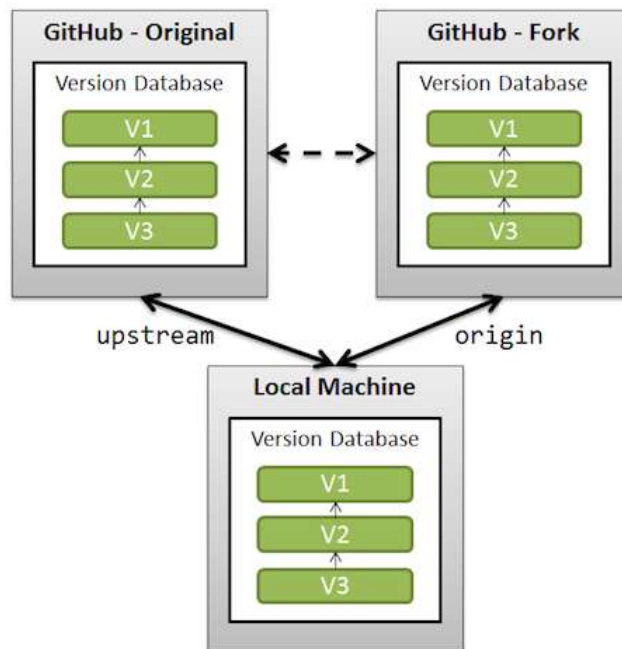


Fig. 2.13. Git Fork Process Schema

A fork is purely the copy of the repository. From the moment developer forked, he gets the current copy, which is then not synchronized with the original in any way. (fig. 2.13) Changes to the fork will not affect the original in any way.

It is possible manually synchronize the fork with the original, thereby updating the copy. Typically, forks are used to prepare changes to the original. Developer cannot create a branch in the original version because it has a different owner. To offer some changes, firstly, it is needed to make a copy for the project, make changes in the copy and only then offer the owner of the original repository the changes. (fig. 2.14)

Sometimes forks are used to create a separate branch that will be completely different from the original: for example, for creating a completely different vision of how the project should look and to rewrite it separately.



Fig. 2.14 Git Fork

2.11.2. Branch vs. Fork

The following list described advantages and disadvantages for forking:

Advantages:

- keeps branches separated by user;
- reduces clutter in the primary repository;
- the team process reflects the external contributor process;

Disadvantages:

- makes it more difficult to see all of the branches that are active;
- collaborating on a branch is trickier (the fork owner needs to add the person as collaborator);
- developer needs to understand the concept of multiple remotes in git;

- requires additional mental bookkeeping;
- it makes the workflow more difficult for people who aren't super comfortable with git.

The following list described advantages and disadvantages for branching.

Advantages:

- keeps all of the work being done around a project in one place;
- all collaborators can push to the same branch to collaborate on it;
- there's only one git remote to deal with.

Disadvantages:

- branches that get abandoned can pile up more easily;
- the team contribution process doesn't match the external contributor process;
- project owner needs to add team members as contributors before they can branch.

The branch, stream and fork work principles are shown on figure 2.15.

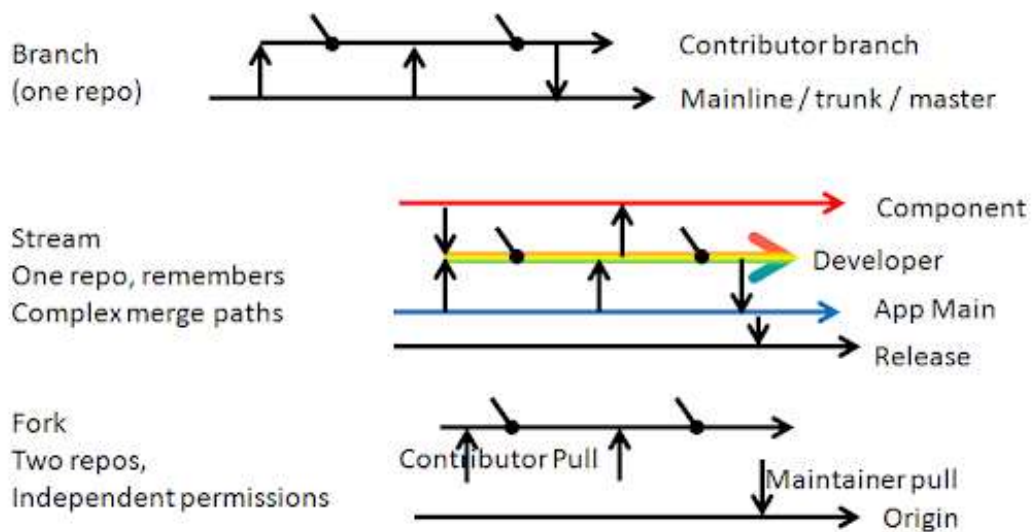


Fig. 2.15 Branch, Stream and Fork work principles

2.11.3. Submodules

During development, it is constantly necessary to involve some other project in the developing project. For example, an additional library that is used independently in several other projects. In order not to produce copies of this library and always have a simple opportunity to update it to the latest version, there is a concept of a submodule in git.

Conclusions on the Second Part

The choice of technologies for developing an application is an important stage in a project. A well-chosen combination of technologies should ensure comfortable work in the future at all stages of the application's existence.

HTML - is a standardized markup language for documents on the World Wide Web. The bulk of web pages contain descriptions of markup in HTML.

CSS determines how the document is look. With CSS, it is possible to define page color, font, block placement, and other aspects of how pages are presented. The existence of CSS helps to separate the creation of the document structure from its presentation.

It is impossible to create more complex dynamics using only HTML and CSS. Javascript is used exactly for that purpose. Javascript was originally created to make the website live. The code written in this language is called a script.

Javascript language's capabilities are determined by the environment where it runs. When a script is running in a browser, it can manipulate the page, its elements, interact with the user and with the server.

It is necessary to look far ahead into the future and predict the potential development and fate of the project. Obviously, the stack should be easily scalable, functional, correspond the latest market trends. It should meet the most modern features. Most importantly, it has to be easily supported in the future by other developers.

React.js has a capacious and understandable API. To work with React, it is necessary to understand a number of terms and the differences between them. Elements are JavaScript objects that are HTML elements. Components are elements of React.js that are created by the developer and can have any name. As a rule, they contain their own specific structure and perform a number of functions.

The TypeScript language is one of the most popular technologies of recent years, both in Frontend and Backend development. Its popularity continues to grow and it is at the heart of many projects.

GraphQL is a query and data manipulation language for APIs, as well as an environment for making those queries. At the heart of any implementation of the GraphQL

API is the data schema - this is a description of what data types it can work with and what data types it can return in response to a request. When working with the GraphQL API, the client does not care at all where the data he requests comes from. It just makes the request to the extent it needs, and the GraphQL server returns the result.

D3.js is a JavaScript library for data processing and visualization. The name D3 itself stands for data driven documents and focuses on data management, although the key functionality of the library is its powerful visualization capabilities.

One of the convenient solutions for unit testing of components is the react-testing-library. The main idea of the library is to bring test scripts closer to using the components the way the user does, which allows for confidence, giving more confidence when the application hits production.

Node.js is a server-side implementation of the JavaScript programming language, which is a big plus for developers who deal with both the client and server side at the same time. Node.js is a platform based on the V8 engine. The main purpose of Node.js is to create scalable distributed network applications - web servers. Unlike JavaScript scripts, Node.js is implemented on the server, not in the browser.

No matter how complex a web application is, it cannot work without data and as a result a database. MongoDB includes databases, state from collections, which are made up of documents. Each document contains fields. Collections have the ability to be indexed, and as a result, it becomes possible to improve the performance of selection and sorting.

Git is the system for controlling versions of the project. In essence, this means that it stores the entire history of project changes. The developer's history of the project and the history of changes in the same project for teammates, who also works on project - everything will have a copy.

PART 3

INTERFACE AND LOGICAL DESIGN, SYSTEM IMPLEMENTATION

Before starting the developing an application, it is necessary to assess the requirements and select the required technology stack. After defining the technology stack, proceed to the allocation of tasks and subtasks, determining the estimated time to complete each task.

3.1. Design a User Interface of the System

The web application interface is the visual part of the application that the user interacts while working with the application. The interface should be simple, intuitive, friendly, visually pleasing to use and consistent.

The visual component is determined by the color scheme and the general theme in which the application is designed. To achieve the goal of a positive perception of the interface of the web application, a colorful, positive design in the style of space and travel across endless spaces was chosen.

During the work with the application, a landscape of space with dynamically moving elements is placed in the background, depending on the location of the cursor. The dynamic component of the site immediately attracts the user's attention and lingers on the application pages.

The correct arrangement of interface elements, logical sequences of web application scenarios and the obvious results of user interaction with the interface determine the UX (experience of interacting with the interface).

3.1.1. Description of the system sections

The application includes following sections:

- sign in and sign up pages;
- dashboard page;
- user personal data page;

- survey page;
- create survey form;
- survey analysis page;
- audience list page.

The next screenshot shows the sign in form for user to login into the system. In order to login user should input email address and password. (fig. 3.1) Also, in further development login with google, facebook and twitter accounts are planned to add.

There also added 'Remember me' checkbox in order to switch off log out after closing the system tab. In case, user check to remember the session, then the session will be saved and there no need to enter email and password again after closing the system.

After submitting the form with clicking on 'Sign in' button, user is redirected to system dashboard route.

Also, there is 'Forgot password' link added. If user cannot remember the credentials for sign in, it is possible to recover the account access.

The link for sign up is present too. For new users, who does not have an account in the system yet, they still can register in the system.

Sign in

Email Address *

Password *

Remember me

SIGN IN

[Forgot password?](#) [Don't have an account? Sign Up](#)

Copyright © Marharyta Aksonova 2020.

Fig. 3.1 Screenshot of Sign in Form

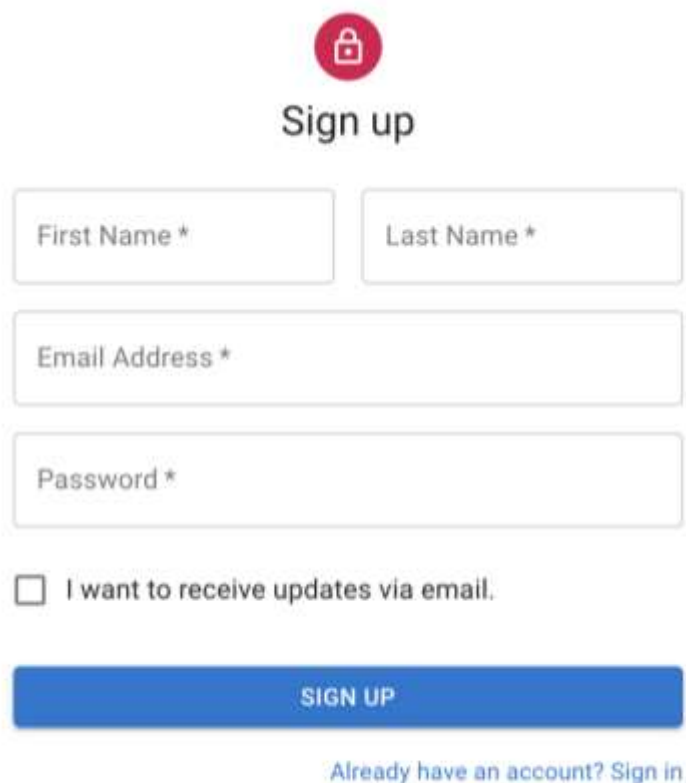
For new users, to register in the system the form for signing up is created. There four inputs which should be filled. (fig. 3.2)

The first name and the last name are needed to know how to name person in the system. The user should enter email and create a password to have the credentials for further login into the system.

The good practice is to create the new password for each system, rather than use the same password for several platforms. It improves the secureness of the access to the system.

Also, there added a checkbox for asking a person if he would not mind achieve letters with description of system updates via email.

There are a lot of features, which can be added and improved in further development, so it nice to have a possibility to tell users, who already actively use the system about the updates which were added or improved.



The screenshot shows a sign-up form with the following elements:

- A red lock icon in a circle at the top center.
- The text "Sign up" below the icon.
- Two input fields: "First Name *" and "Last Name *".
- A single input field: "Email Address *".
- A single input field: "Password *".
- A checkbox with the text "I want to receive updates via email."
- A blue button labeled "SIGN UP".
- A link below the button: "Already have an account? Sign in".

Copyright © Marharyta Aksonova 2020.

Fig. 3.2 Screenshot of Sign up Form

After login, user redirected to dashboard route. On this page there all common analytical information is present. (fig. 3.3) User can see the whole amount of questions which is answered from all surveys.

The total customers number is the amount of people who completed at least one survey. If person asked on questions from several surveys, he is still the one customer.

The tab with survey completed is calculated in percent. Some surveys can be answering right now, or some of them can be canceled. The percentage is calculated by the principle of the correlation of whole amount of started surveys to successfully completed ones.

There also added a possibility to generate report on each survey. When a survey is completed by several people, the system user can create a report with analytical data of that survey.

On the dashboard page, there are two charts present. The chart of the latest activity demonstrates how actively the responders completing the surveys. The grey bar shows how many surveys are started with people by corresponding date.

The blue bar shows how many surveys completed. If the grey bar is higher than blue, that means that some of opened surveys are not completed. If the blue bar higher than grey, that means that some people filled the same survey one time more. The best result is when the grey bar and the blue bar are on the same height.

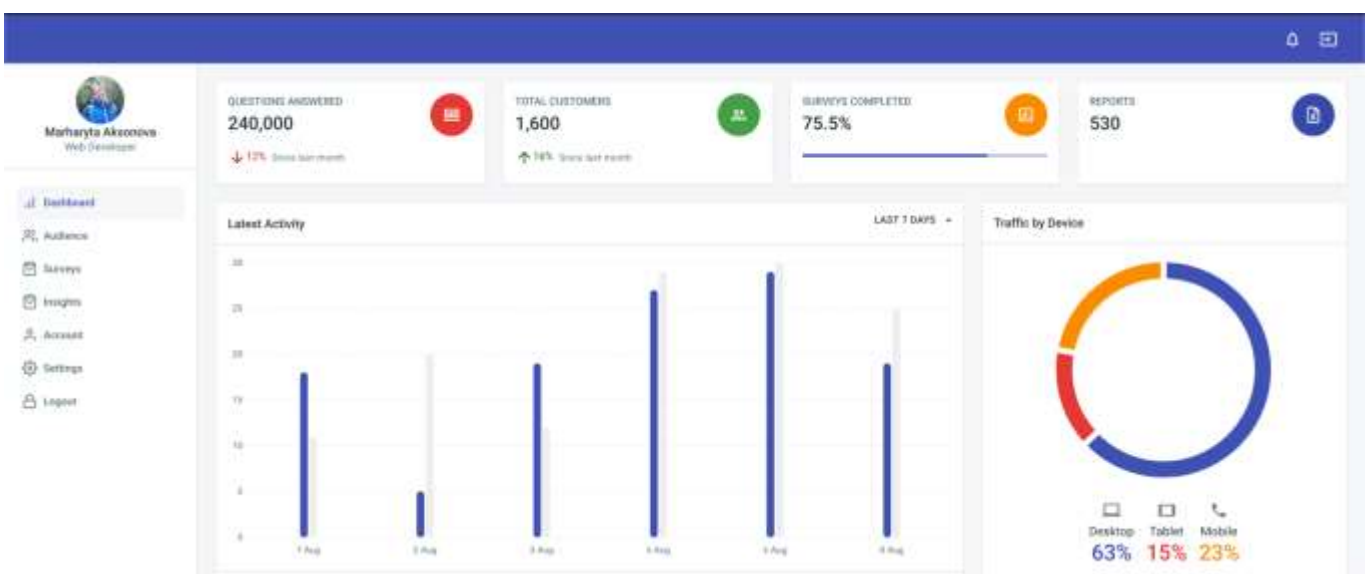


Fig. 3.3 Screenshot of the first part of dashboard page of the system

The screen resolution is checked for understanding the user current device with react-responsive package.

The right chart demonstrates, which device users use to answer the questions of user's survey. There are three possible devices that can be used to complete the surveys: desktop, tablet or mobile. In code, the way to determine if the user on desktop or mobile is showed on figure 3.4.

On the top bar there are only two icons present. The notification icon will show, if there any notification comes. Right now, there is no logic for notification added. It is planning to develop after completing developing and testing the base functionality of the system.

By clicking on the second icon, the user will log out from the system. Also, when the user hovers these icons, the popup with button description appears.

```
import { useMediaQuery } from 'react-responsive'

const Desktop = ({ children }) => {
  const isDesktop = useMediaQuery({ minWidth: 992 })
  return isDesktop ? children : null
}

const Tablet = ({ children }) => {
  const isTablet = useMediaQuery({ minWidth: 768, maxWidth: 991 })
  return isTablet ? children : null
}

const Mobile = ({ children }) => {
  const isMobile = useMediaQuery({ maxWidth: 767 })
  return isMobile ? children : null
}
```

Fig. 3.4 The screenshot of code render with screen resolution check

The figure 3.5 represents the two main data: Latest social media responders and latest activity.

The latest social media responders table lists the social media resource from which people relocate to the system. The simplest way to establish that is to use Google Analytics API to track the link clicks shared with social network. There displayed the social media name, logo and the last target click.

The latest activity table represents the current activity of the users. The first column shows which survey was last completed. The next column points started going through questions of survey.

The third column shows the date of the activity. It is also possible to sort rows either to show oldest first or to show earliest first. The last column represents the status of user action regarding to the survey.

If responder is completed the survey, the status completed is showed. If responder is answering on questions right now, then the status in progress is showed. If responder canceled the survey, then the status canceled is showed.

Clicking on View All button, user appears on the page with the wider list of the correspond table. On those pages user can see the list of more items and detailed information about each of them.

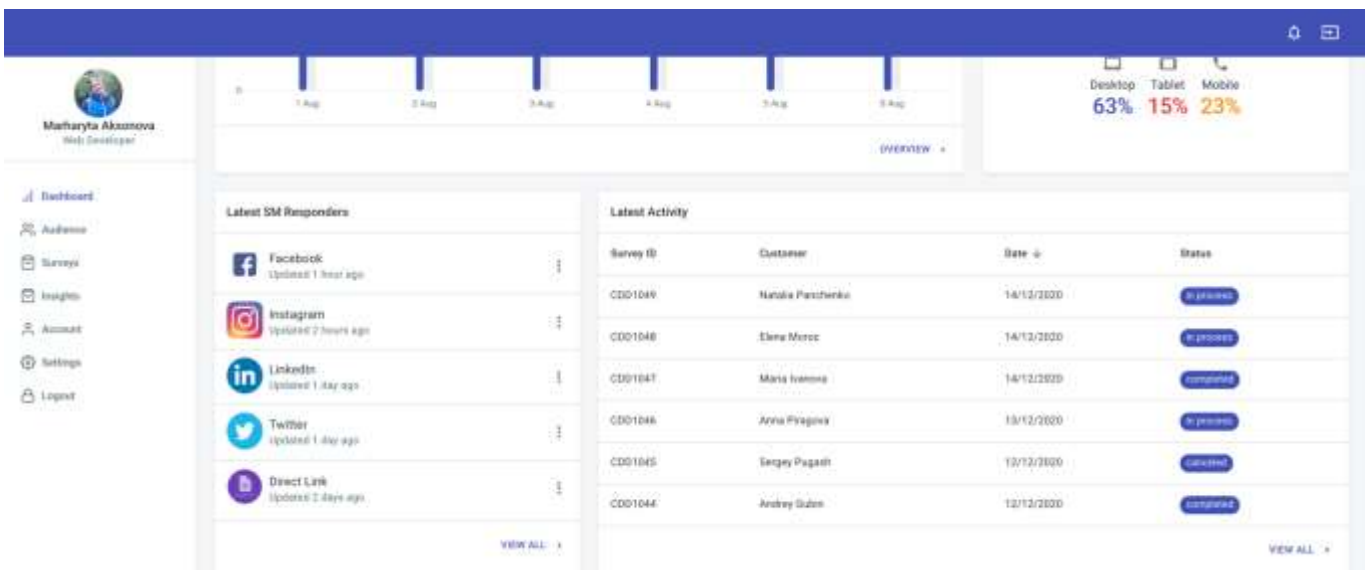


Fig. 3.5 Screenshot of the second part of dashboard page of the system

On the page, which showed the detailed data about the survey analysis there is audience age chart present.

The column stacked bar chart shows the audience age and audience location based on their answers. The bars are divided by different age categories and filled different colors. On each city males and females are represented as different bars.

The chart is a convenient way to see and analyze the data rather than only get it by text, numbers or tables.

Here charts are divided on different colors for better division of data received.

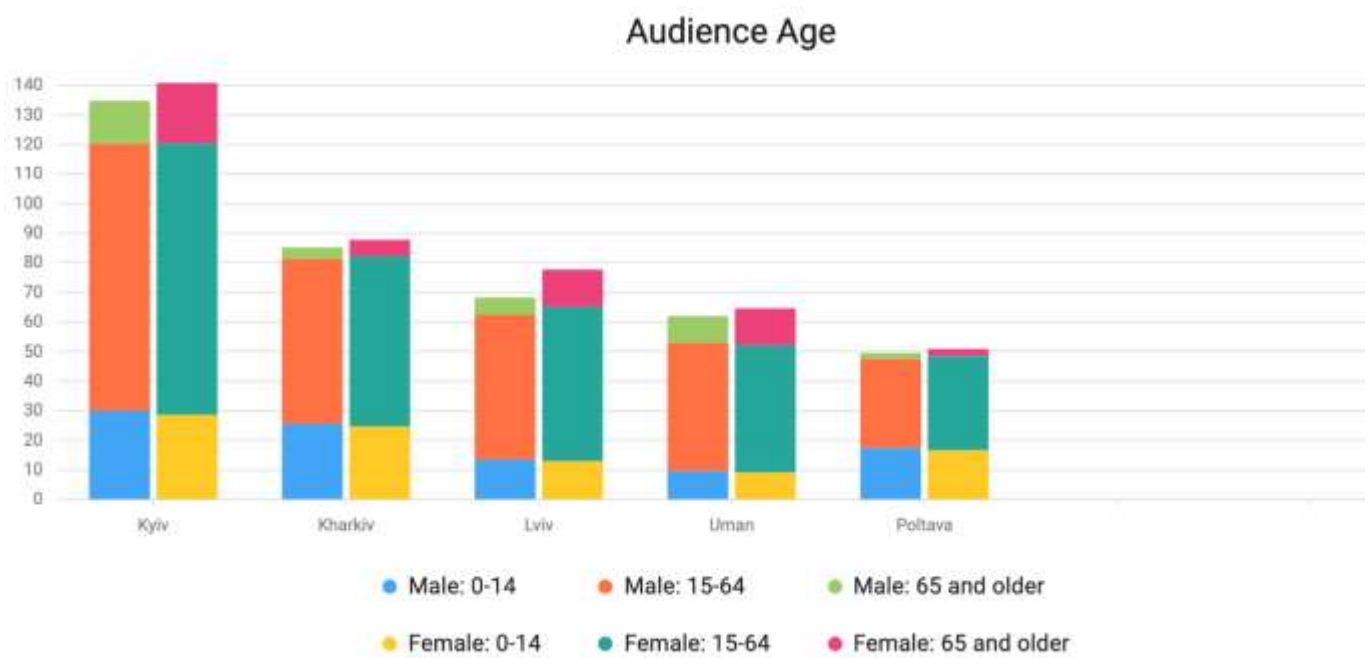


Fig. 3.6. Screenshot of Audience Age Column Stacked Bar Chart

Also, the question block is present on the survey detailed page. (fig.3.7) On the top of the block the question itself is present.

Below it, there is column chart present, which shows the analytics of how audience answered the question.

The horizontal labels represent the options which were suggested to responder to choose. Also, there is skipped label, which shows how many people skipped the question. In some questions there is a possibility to allow responder write custom answer. In case anyone submitted the custom answer the bar should appear.

The vertical labels showed the amount of people answered the question. The numbers are divided on five. In order to see the exact number, it should be hovered on the interested bar.

It is good to have a possibility to sort the bars here. In addition, in future here should be a possibility to filter the answers.

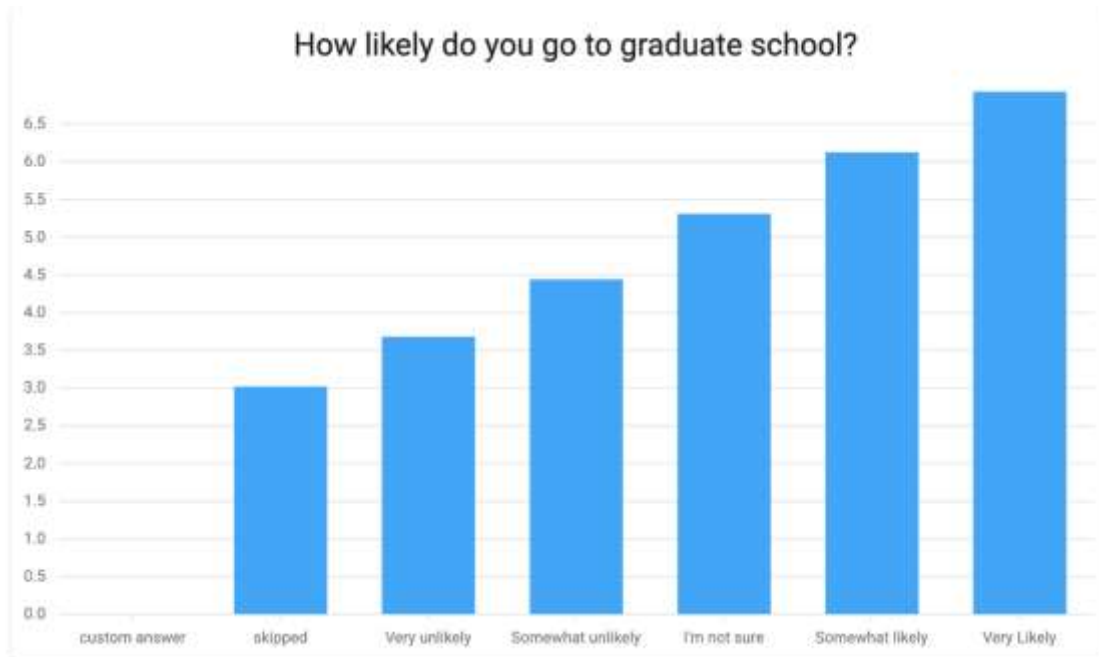


Fig. 3.7. Screenshot of the Question Block

The audience page is showed a list of people who filled the surveys. (fig. 3.8) There is a table present. The first column of the table is a checkbox, clicking on which the user is selected and then there appears a menu to delete or update data for the responder. The second column showed the first and last name of person. The next one displaying his email. The fourth column is present the location of the responder. The city and the country data are indicated here. Then there is a phone number present and registration date. All that data is collected both from registration data and filled surveys data. The list of users and their data are received from database. On this page it is also possible to manually add a customer.

The screenshot shows a dashboard for "Marharyta Alkocova, Web Developer". On the left is a sidebar with navigation options: Dashboard, Audience, Surveys, Insights, Account, Settings, and Logout. The main area features a search bar for customers and buttons for "IMPORT", "EXPORT", and "ADD CUSTOMER". Below is a table with the following data:

<input type="checkbox"/>	Name	Email	Location	Phone	Registration date
<input type="checkbox"/>	Andrey Gulin	andrey@gmail.com	Kyiv, Ukraine	+38 066 123 12 23	13/11/2020
<input type="checkbox"/>	Maria Ivanova	maria.ivanova@gmail.com	Kharkov, Ukraine	+38 066 123 12 26	02/12/2020
<input type="checkbox"/>	Anna Pivagova	anna@gmail.com	Kyiv, Ukraine	+38 066 123 12 24	18/11/2020
<input type="checkbox"/>	Sergey Pugach	pugach-sergey@gmail.com	Lviv, Ukraine	+38 066 123 12 27	08/12/2020
<input type="checkbox"/>	Elena Moroz	elena@ukr.moroz@gmail.com	Poltava, Ukraine	+38 066 123 12 92	14/12/2020
<input type="checkbox"/>	Misha Tkachenko	misha.tkachenko@gmail.com	Kyiv, Ukraine	+38 066 123 12 73	09/12/2020

Fig. 3.8 Screenshot of audience page

Clicking on the export button, the file in json format is downloaded. The import functionality does not work for now. But in future, it is planning to add.

The input for searching customer is added on the top of the table. The filtering works simultaneously based on text field value.

The surveys page is showed below. (fig. 3.9) It is also presented in the table. It was considered that the displaying such kind of data is most convenient through table. The first column of the table is a checkbox, clicking on which the survey is selected and then there appears a menu with additional functionality.

The second column represent the survey title. The third one represents the survey id - it is generated by GraphQL.

Going to the survey page, its id is present in the route too. The third column shows how many questions are in each survey. The functionality for sorting surveys by increasing or decreasing based on questions amount were added. The next column showed information about how many times the survey was responded. The last column showed when the specific survey was created.

<input type="checkbox"/> Surveys	ID	↓ Questions	Completed times	Created
<input type="checkbox"/> Course Satisfaction	CDD1049	25	51	11.12.2020
<input type="checkbox"/> Product Satisfaction	CDD1048	34	87	07.12.2020
<input type="checkbox"/> University	CDD1047	26	65	04.12.2020
<input type="checkbox"/> Cars	CDD1046	24	98	05.12.2020
<input type="checkbox"/> New Year Celebration	CDD1045	18	63	15.12.2020

Rows per page: 5 1-5 of 12

Fig. 3.9 Screenshot of surveys page

Clicking on the checkbox from the header there all surveys are selected. (fig. 3.10) All of them can be removed at once.

12 selected 🗑️

Surveys	ID	Questions	Completed times	Created
<input checked="" type="checkbox"/> Course Satisfaction	CDD1049	25	51	11.12.2020
<input checked="" type="checkbox"/> Product Satisfaction	CDD1048	34	87	07.12.2020
<input checked="" type="checkbox"/> University	CDD1047	26	65	04.12.2020
<input checked="" type="checkbox"/> Cars	CDD1046	24	98	05.12.2020
<input checked="" type="checkbox"/> New Year Celebration	CDD1045	18	63	15.12.2020

Rows per page: 5 1-5 of 12 < >

Fig. 3.10. Screenshot of the selected surveys

When user start survey, he is seeing the list of question. The next screenshot shows the question of the survey. It consists of the title – the question itself and the options to choose. Some of questions allows to answer not only suggested options, but also the custom widely described answer. So, to if user checks custom answers, the input appears. It is multiline text input and it has no restrictions in value length. (fig. 3.11)

How likely do you go to graduate school?

- Very Likely
- Somewhat likely
- I'm not sure
- Somewhat unlikely
- Very unlikely
- Custom answer

Custom answer

Fig. 3.11 Screenshot of the Survey Question

3.2. Server development

The server is built with Express.js. Server file is created at the root of the application, which initiates the server in the static folder. (fig. 3.12)

The port on which the server is created can be determined by the environment variable or directed in the code.

```
const express = require('express');
const { graphqlHTTP } = require('express-graphql');

const app = express();

app.use(
  '/graphql',
  graphqlHTTP({
    schema: SystemGraphQLSchema,
    graphiql: true,
  }),
);

app.listen(4000);
```

Fig. 3.12 The screenshot of basic express server

3.3. Database development

No matter how complex a web application is, it cannot work without data and as a result a database. Before choosing a database, it is necessary to determine what to store in the database and what the site will require.

The goal is to get a general and complete picture of the structure of the database.

As part of this project, there is no need for a relational database and was opted for an object database.

MongoDB is an efficient and scalable database for this task. MongoDB is a very successful symbiosis between the familiar relational database and key-value storage.

All documents are grouped in the collections and global relationships are already established between collections.

The surveys analysis system performs such function – writes the responder's answers to the database and allows review them. Saving surveys questions and other data.

To accomplish this task, it should be created Form Component and a function which is called after the form is confirmed.

To add data to the cloud, the first thing should be done is it should be specified in which document place that data.

For MongoDB it can be done using document reference and the path to the required document.

The path must be indicated with collections and documents. Collections should be written before the documents. The piece of code from the system related to getting surveys is showed on figure 3.13.

```
import {MongoClient, ObjectId} from 'mongodb'
const MONGO_URL = 'mongodb://localhost:27017/blog'
const db = await MongoClient.connect(MONGO_URL)
const Surveys = db.collection('surveys')
const Users = db.collection('users')

Query: {
  post: async (root, {_id}) => {
    return prepare(await Surveys.findOne(ObjectId(_id)))
  },
  posts: async () => {
    return (await Surveys.find({}).toArray()).map(prepare)
  },
  comment: async (root, {_id}) => {
    return prepare(await Users.findOne(ObjectId(_id)))
  },
},
```

Fig. 3.13

3.4. Testing Strategy

To achieve the highest testing quality for each method and component of a tested program, it must be chosen the most successful method or set of methods that will provide the desired result. In this case, it is necessary to find the best correlation between the time that will be spent on testing and the quality of testing. A set of tests should have a minimum loss-making, but at the same time they should cover the maximum of the system functionality.

During the testing of the program, several different testing methods will be used. Conditionally, these methods can be divided into 2 types: white box methods and black box methods.

The difference between these groups of methods is that when using black box methods, testing occurs without access to the code of the program. It means that testator has access only to those capabilities that the user of the program has.

When using white box methods, the testator uses the program code to achieve the desired result. For a linear structure functions a “black box” testing can be applied, using their specifications by the method of equivalent partitions and error assumptions.

Testing methods of classes that have a nonlinear structure (that have conditions, cycles, recursive calls) must be conducted using the solution testing method ("white box"). Also, when testing particularly complex methods, it is necessary to additionally apply the "black box" method, namely putting the hypothesis about an error.

This will increase the reliability of the system and methods, which are especially important for the work. This method is quite well suited for testing the interface part of the program.

The most important part is testing the whole program after all components and methods were tested. It can not be missed because some errors might not arise while testing individual components independently.

So, for testing the developed system the following strategy was developed:

- assumptions about error and conditionality methods were selected for testing;
- testing of the main functions and components should be done by all selected methods;
- testing of other functions and components should be done by the method of error assumption;
- direction of testing is from the bottom to the top. First, all submodules are tested and then the main module which uses all submodules.

Also, it is important to add unit tests to check the correctness of working functionality. There is an opinion exists that the adding units test is just a wasting of time, but to my mind it is wrong.

3.5. Unit and Snapshot tests

Unit testing a react component using snapshots allows to write very simple and easily maintainable tests that will help avoid problems in production and make the project code more reliable.

For each test case, a snapshot is created, which is considered a reference.

After developing the system, there were added a test that verifies that the component renders correctly. To do that, there were included the react-testing and jest libraries to shallow the components. Shallow rendering is needed so that only the component which is testing will render. Internal components are not rendered.

In order to create a snapshot for the component, the assert method should be called. It checks if the component's rendering result matches the reference snapshot. If there is no reference yet, it will be created the first time the test is run.

After launching the test - snapshots folder has appeared in the folder with the component, in which all saved snapshots will be stored. They should be added to the git repository. Snapshots show exactly how it changed during the development process and are additional visual documentation.

For example, on the figure 3.14 there is showed how the button component were tested via snapshot tests.

```
// Global imports
import React from 'react';
import { render, fireEvent, screen } from '@testing-library/react';
import '@testing-library/jest-dom/extend-expect';

// Local imports
import Button from '../../Components/Atoms/Button';

describe('<Button />', () => {
  it('renders and match the snapshot', () => {
    const onClick = jest.fn();
    const { asFragment } = render(
      <Button buttonVariant="secondary" disabled={false} onClick={onClick}>
        Test
      </Button>
    );

    expect(asFragment).toMatchSnapshot();
  });
});
```

Fig. 3.14 Screenshot of snapshot test for Button component

On the figure 3.15 there is showed how the button component is tested via unit tests.

```
it('calls onClick props when clicked', () => {
  const onClick = jest.fn();
  render(<Button onClick={onClick}>Click Me</Button>);
  fireEvent.click(screen.getByTestId('atoms-button'));

  expect(onClick).toHaveBeenCalledTimes(1);
});

it('is disabled when disabled prop equals true', () => {
  render(<Button disabled>Click Me</Button>);
  const button = screen.getByTestId('atoms-button');

  expect(button).toBeDisabled();
});

it('has properly className', () => {
  render(<Button className="secondary">Click Me</Button>);
  const button = screen.getByTestId('atoms-button');

  expect(button).toHaveClass('govuk-button--secondary');
});

it('has properly text content', () => {
  const text = 'Click Me';
  render(<Button>{text}</Button>);
  const button = screen.getByTestId('atoms-button');

  expect(button).toHaveTextContent(text);
});
});
```

Fig. 3.15 Screenshot of unit tests for Button component

3.6. System for version control

Development is carried out using the git version control system. It is saved in a private repository of a GitHub web service for hosting IT projects based on git.

Git allows you to share saved data with other users in the repository. Git has a central repository that multiple developers sync with, each provided with a local copy of their entire development history.

Other developers can download updates from the central repository, make their own edits, and upload them back to the central repository.

The screenshot of project repository is showed on figure 3.16.

File	Commit Message	Time
components	removed ie18 trans from question title	2 hours ago
contexts	prettified	2 hours ago
data	mutation optimization	a day ago
hooks	prettified	last month
modals	added "or" to clean data	last month
routes	prettified	last month
shared	- wysiwyg button & editor dnd functionality	last month
theme	clean data layout fixes	last month
types	prettified	last month
ui	prettified	last month
utils	feature (create pptxExporter)	2 months ago
App.tsx	- wysiwyg button & editor dnd functionality	last month

Fig. 3.16 The screenshot of project repository on the GitHub web service

3.7. Performance Results

When React is in the development mode, it generates the mark for each component and measure events that are part of the User Timing API.

In order to create a performance profile, developer should go to the page on the local host, click on the Start profiling and reload page button in the developer tools.

This will start recording performance information for the current page. Chrome will automatically stop recording data after the page has completely loaded and displayed on the screen. The indicator in the form of a red bar shows that in this place on the time scale there is a serious load on the processor (performing long-term tasks). Perhaps this is exactly what slows down the application, and what we need to explore in more detail.

The colors used in the diagram at the top of the window correspond to different activities. The drop in productivity during the performance of various types of activities has its own causes. There are special ways to fix various problems and means of analyzing them.

The performance diagram of the inventory system is shown on figure 3.10.

React publishes metrics through the User Timing API, which can be used to place time markers in the own code. This allows find out how long a component takes to initialize,

or how long after a certain event. Chrome does not have React debugging tools built in, but support for the generic API that React uses. In addition, all major browsers support the User Timing API, but the Performance tab of the Chrome developer tools went much further than others, which made debugging React applications in Chrome much easier.

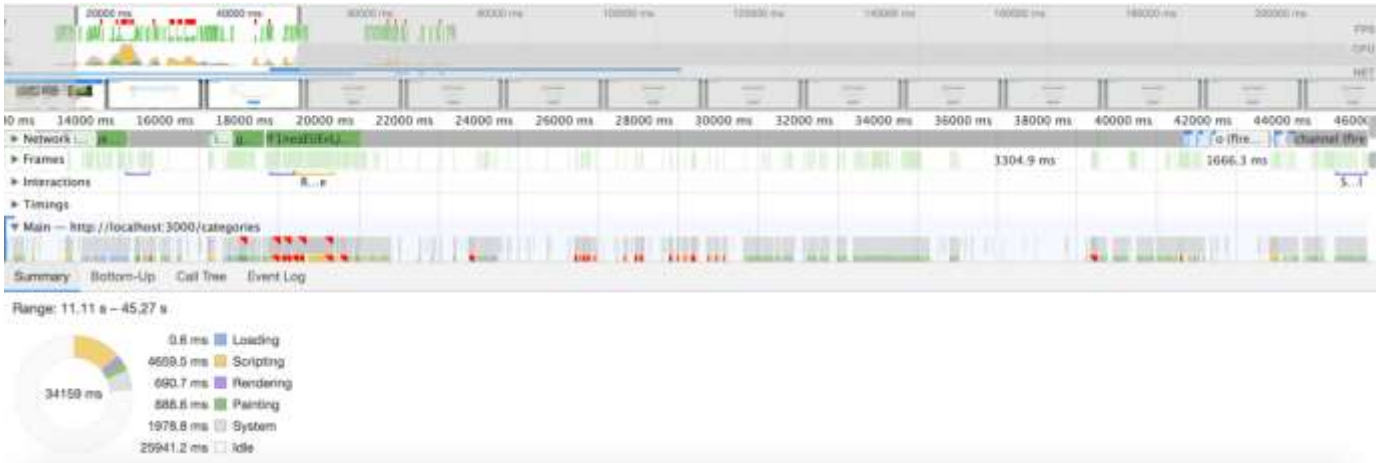


Fig. 3.10. Screenshot of the performance results of the system

3.8. Development of the User Manual

To start working with the system user should register. He can do it by himself, entering his email, password and personal information. After the registry is complete, the whole functions of the system will be available for the user.

Add survey

To add new survey user should go to the page with a list of surveys and click 'Add' button, or to go directly to the link with form.

After that, user should fill out a form, where he must enter the survey name, optionally its description and add list of questions to that survey. After confirming the form the survey will be added to the database.

Edit survey

To edit the survey user should go to the survey list play, determine which survey he wants to edit and click on the edit icon. After that, user goes to page with a form where can update the information and confirm it. Then the old information will be replaced by the new one in the database.

Delete survey

To remove the survey user should go the page with a list surveys, determine which survey he want to delete and click on the delete icon. A confirmation popup appears for user.

In case of the confirmation, the survey will be deleted from the database otherwise, if the user clicks the cancel button, no changes will happen.

There is also a possibility to delete several surveys at once. Simple choose a few surveys and click delete icon.

Search for survey or for user

To search the specific survey or user should input the request in the correspond search text field of the system. After searching will complete, the found items appears to users. If no items will be found, then the correspond message will be shown.

3.9. Directions of Further Development of the System

Each system has a direction where to grow and develop, and this system is not an exception. At this development stage, there was developed the basic features and functionality to make system work as MVP.

In current system there are a lot of functionality that can be added. The notifications are good to be added. The potential of the application is quite large, since within the framework of this application it can be placed any database of questions and adapt into any survey format.

Conclusions on the Third Part

The third section fully described the process of creating a web application using selected technologies. Figures with results of work, examples from the code and results of checking the productivity of system execution were provided.

Before starting the developing an application, it was assessed the requirements and select the required technology stack. After defining the technology stack, proceed to the allocation of tasks and subtasks, determining the estimated time to complete each task.

The correct arrangement of interface elements, logical sequences of web application scenarios and the obvious results of user interaction with the interface determine the UX (experience of interacting with the interface).

The application includes following sections:

- sign in and sign up pages;
- dashboard page;
- user personal data page;
- survey page;
- create survey form;
- survey analysis page;
- audience list page.

No matter how complex a web application is, it cannot work without data and as a result a database. Before choosing a database, it was determined what to store in the database and what the site will require.

The goal was to get a general and complete picture of the structure of the database.

When React is in the development mode, it generates the mark for each component and measure events that are part of the User Timing API.

In order to create a performance profile, developer should go to the page on the local host, click on the Start profiling and reload page button in the developer tools.

To achieve the highest testing quality for each method and component of a tested program, it was chosen the most successful method or set of methods that will provide the desired result. The unit test results also were provided.

For testing the developed system the following strategy was developed:

- assumptions about error and conditionality methods were selected for testing;
- testing of the main functions and components should be done by all selected methods;
- testing of other functions and components should be done by the method of error assumption;
- direction of testing is from the bottom to the top. First, all submodules were tested and then the main module which uses all submodules.

Development was carried out using the git version control system. It is saved in a private repository of a GitHub web service for hosting IT projects based on git.

Any system can be growth and every system have direction of further development. The potential of the application is quite large, since within the framework of this application it can be placed any database of questions and adapt into any survey format.

CONCLUSIONS

During the preparing the diploma project a system for analyzing social survey data was created.

The goal which was achieved is: to solve the problem of individuals or small companies with the analysis of accumulated data statistics, without spending additional funds on purchasing additional software products.

The system for analyzing social survey data was developed in accordance with the established requirements. The functional requirements dictated that the system should have the platform menu in the header and check whether users have access to surveys and dashboard.

The user should have a possibility to see the list of questions and pages which he has in survey. Clicking on the page, user should see detailed statistics for each question on the page. Besides that, there should be a button, clicking on which user can check information for all questions on the page.

User should be able to see general statistics and charts of the survey or for each question. Also, there should be the possibility to sort questions by number, most upvoted insights, most downvoted insights or comments amount.

Question block should provide the general information related to a question and a statistical chart that displays survey data.

User will have the possibility to see answers and their percentage to total answers number. The system should have the comments list for the insight item which user have chosen. User will be able to send a comment on an insight. Also, the application should have functionality to allow users to edit and delete only those comments that they posted.

User should have a possibility to create insights for the questions. User can add and edit topics for this insight.

The functionality to select topics should be included in the system. Then user will see the questions with these topics on the screen. He will be able to change sort of topics by drag and drop.

In order to make a review, there should be a functionality for filtering questions and insights by parameters user has chosen.

The following technologies were chosen to solve the problems of the client application:

- React library for working with UI rendering
- GraphQL library as a library for making requests to the back end of a web application and creating an architecture for processing internal actions in the client application.

The main tasks of a server application when implementing a single page web application approach are:

- interaction with the database;
- processing requests of the client application and execution of the business logic of the application;

The following technologies were chosen to solve the problems of the server application:

- Node.js - a platform that allows JavaScript to be executed on the server and provides opportunities for interacting with the server's system resources;
- Express.js - a framework that allows you to implement a web server, routing user requests, and decompose program logic into separate subsystems.

The interface requirements dictated that the system should have dashboard page, login page, personal account with information about the user, survey page and question analysis page.

As for security requirements, besides fact that the system should be convenient to use, it also has to provide security to users and their private data. Before developing a methodology for protecting a web application from potential threats, the site should be checked for vulnerabilities.

The check is carried out manually or automatically. The programs will test the application for major risks. These software products exist in two versions: black hat, simulating the actions of hackers, and white hat, systematically revealing all system flaws by scanning. Also, in order to supply the system safety and reliability the developer should:

- add logic to validate the input value;
- provide blocking wrong user actions which are out of the scope;
- ensure the integrity of stored data;
- exclude the possibility of the unauthorized access to the information;

The application has to be fast too. The following list of performance characteristics was taken care of while developing the system:

- user-friendliness: the system functionality and design should be intuitive understandable so new users should not have any problems with onboarding.
- user satisfaction: the system should match the user expectations.

If the performance results do not coincide with the requirements, then it should be optimized.

- response time: the system should react to the user's actions as quickly as possible.

The second section examines in detail the popular concept of the one-page application website and the key elements that make it up. The choice of technology, library, architectural solutions and approaches depends on the tasks to be solved and their complexity. In the second section, frameworks and technologies for creating a system for data analysis were discussed.

New frameworks appear every couple of years and greatly simplify the development process by reducing the amount of code, structuring it and adapting known solutions from other platforms.

The React.js library was chosen as the main one for development as the most modern and convenient to use from my perspective.

GraphQL is a query and data manipulation language for APIs, as well as an environment for making those queries.

No matter how complex a web application is, it cannot work without data and as a result a database. Before choosing a database, it was determined what to store in the database and what the system will require.

The goal is to get a general and complete picture of the structure of the database.

As part of this project, there is no need for a relational database and was opted for an object database.

MongoDB is an efficient and scalable database for this task. MongoDB is a very successful symbiosis between the familiar relational database and key-value storage.

All documents are grouped in the collections and global relationships are already established between collections.

Unit testing a react component using snapshots allows to write very simple and easily maintainable tests that will help avoid problems in production and make the project code more reliable. For each test case, a snapshot is created, which is considered a reference.

After developing the system, there were added a test that verifies that the component renders correctly. To do that, there were included the react-testing and jest libraries to shallow the components.

Shallow rendering is needed so that only the component which is testing will render. Internal components are not rendered.

For rendering charts, the D3.js package was chosen. While standard rendering engines like the canvas element rely on pixels, svg uses vectors. Using SVG allows to create graphically rich structures with animation and interoperability. Compared to pixel art, SVG has several advantages.

The vast majority of the main features of libraries, packages and how to use them were also considered. The structure of the application, the process of creating and running the application are described in detail. Development using these technologies was convenient and comfortable. Their capabilities are quite sufficient to create quite complex applications.

The third section fully described the process of creating a web application using selected technologies. Figures with results of work, examples from the code and results of checking the productivity of system execution were provided. The unit test results also were provided.

To achieve the highest testing quality for each method and component of a tested program, it was chosen the most successful method or set of methods that will provide the desired result.

Development was carried out using the git version control system. It is saved in a private repository of a GitHub web service for hosting IT projects based on git.

Any system can be growth and every system have direction of further development. The potential of the application is quite large, since within the framework of this application it can be placed any database of questions and adapt into any survey format. This system can be introduced into educational processes to receive feedback from staff and students too. There were also developed the user manual for quick start working with a system.

As the result the desired analysis system was developed. It is secure for users so they can store their information and private data without any doubts. It also has high performance results and beautiful interface.

REFERENCE LIST

TODO REFERENCE LIST