**THE EKA UNIVERSITY OF APPLIED SCIENCES**

The Study Programme "Information Technologies"

**Meiirzhan Baltabayev**

**DEVELOPMENT OF A WEBSITE WITH A BUILT-IN CHAT ASSISTANT**

**Bachelor thesis**

Adviser:

Docent, Mg.sc.comp. A.Liepiņš

**Rīga 2024**

**ABSTRACT**

**Meiirzhan Baltabayev.** Development of a website with a built-in chat assistant. – Riga: Higher School of Economics and Culture, 2024.

The Bachelor Thesis is written in English. The volume of the work is 62 pages (not counting appendices). The thesis contains 3 annexes on 15 pages. The Bachelor's thesis consists of an Introduction, an Analytical Review section, a case study section, a Research Results section, a Conclusions and Recommendations section, and a List of references. The work contains 3 tables, 19 figures, 3 annexes and 31 sources of literature.

**Research relevance:** Small and medium-sized entrepreneurs have severe resource constraints, skills and knowledge that limit their potential for growth and competitiveness. For this reason, developing a website for entrepreneurs with the ability to work with statistics and a built-in chat assistant that gives advice on doing business and taxes, as well as analytics of business documents, should help automate the daily tasks of entrepreneurs and improve the management of business processes, which will increase the quality of their business management and competitiveness in the business environment.

**Research goal:** develop an innovative solution that can provide SMEs with effective management and support for their business with a built-in chat assistant based on the ChatGPT API, which can provide real-world support. for SMEs in this dynamic environment.

**Research methods:** In the bachelor's thesis, such methods as the collection of case stages for the analysis of competitive companies were used, as well as such methods of data processing as descriptive statistics, regression analysis and qualitative content analysis are also used.

**Main findings:** As a result of the study, the criteria for integrating digital technologies into small and medium-sized enterprises were determined. A website has been developed with the ability to manage business documentation, statistical analysis, as well as real-time support from a chat system based on OpenAI GPT.

*Keywords:* MySQL; flask; chat-assistant; Vue.js; SME

**ANOTĀCIJA**

**Meiirzhan Baltabayev.** Tīmekļa vietnes izstrāde ar iebūvētu tērzēšanas palīgu. – Rīga: Ekonomikas un kultūras augstskola, 2024.

Bakalaura darbs ir rakstīts angļu valodā. Darba apjoms ir 62 lappuses (neskaitot pielikumus). Bakalaura darbam ir 3 pielikumi uz 15 lapām. Bakalaura darbs sastāv no ievada, analītiskā pārskata sadaļas, gadījumu izpētes sadaļas, pētījumu rezultātu sadaļas, secinājumu un ieteikumu sadaļas un atsauču saraksta. Darbā ir 3 tabulas, 19 attēli, 3 pielikumi un 31 literatūras avoti.

**Pētījuma aktualitāte:** Mazo un vidējo uzņēmumu uzņēmēji izjūt būtisku resursu ierobežojumu un profesionālo prasmju un zināšanu trūkumu, kas viņiem rada vairākas grūtības augt un būt konkurētspējīgiem. Specializētas tīmekļa vietnes izstrāde ar spēju strādāt ar statistiku un ar iebūvētu tērzēšanas palīgu, kas sniedz viņiem padomus par uzņēmējdarbības veikšanu un nodokļiem, kā arī ar uzņēmējdarbības dokumentu analīzi, palīdzēs automatizēt uzņēmēju ikdienas uzdevumus un uzlabot biznesa procesu vadību, kas palielinās viņu biznesa kvalitāti un konkurētspēju.

**Pētījuma mērķis:** izstrādāt inovatīvu risinājumu, kas var nodrošināt MVU efektīvu pārvaldību un atbalstu viņu uzņēmējdarbībai ar iebūvētu tērzēšanas palīgu, kura pamatā ir ChatGPT API, kas var nodrošināt reālu atbalstu. MVU šajā dinamiskajā vidē.

**Pētījuma metodes:** Bakalaura darbā tika izmantotas tādas metodes kā gadījuma posmu vākšana konkurētspējīgu uzņēmumu analīzei, kā arī tiek izmantotas arī tādas datu apstrādes metodes kā aprakstošā statistika, regresijas analīze un kvalitatīvā satura analīze.

**Sasniegtie rezultāti:** Pētījuma rezultātā tika noteikti kritēriji digitālo tehnoloģiju integrēšanai mazos un vidējos uzņēmumos. Ir izstrādāta vietne ar iespēju pārvaldīt biznesa dokumentāciju, statistisko analīzi, kā arī reāllaika atbalstu no tērzēšanas sistēmas, kuras pamatā ir OpenAI GPT.

*Keywords:* MySQL; flask; chat-assistant; Vue.js; SME

**TABLE OF CONTENTS**

[INTRODUCTION 5](#_Toc167745224)

[1. ANALYTICAL REVIEW OF LITERATURE 8](#_Toc167745225)

[1.1. Concepts and meaning of Digitalization and Automation 8](#_Toc167745226)

[1.2. The role and impact of digital technologies on small and medium-sized businesses 10](#_Toc167745227)

[1.3. Artificial intelligence and chatbots in supporting business processes. 18](#_Toc167745228)

[2. CASE STUDY 24](#_Toc167745229)

[2.1. Website requirement definition. 24](#_Toc167745230)

[2.2. Selection of Technologies and their rationale 31](#_Toc167745231)

[2.3. Website development 37](#_Toc167745232)

[3. RESULTS 41](#_Toc167745233)

[3.1. Frontend website development 41](#_Toc167745234)

[3.2. Backend website development 49](#_Toc167745235)

[CONCLUSIONS AND RECOMMENDATIONS 57](#_Toc167745236)

[BIBLIOGRAPHY 59](#_Toc167745237)

[ANNEXES 62](#_Toc167745238)

# 

# INTRODUCTION

In the most developed countries, small businesses account for 60-70% of GNP. It is small and medium-sized businesses that are generators of innovative processes in industry in developed and developing countries, capable of dynamically responding to changes in consumption patterns (IFAC 2019). Due to the rapid growth of information technology and an unstable business environment, small and medium-sized businesses have become the focus of government attention as the main driver of economic growth and innovation. In this regard, the creation and development of small and medium-sized businesses is the main task of the economic policy of countries in the normal functioning of the market mechanism.

Also, the rate of economic growth, the quality of the gross national product, as well as its structure are determined by small and medium-sized businesses. The most significant thing for small and medium-sized businesses is their participation in the export of products. Experts have identified the share of small and medium-sized enterprises in industrial exports of a few developed countries. Thus, in Germany and the Netherlands this share is about 50%, in Italy – about 30-35% (Natalya Zakharova 2017). Their participation in the formation of the economic structure and share in product exports emphasize the need for modern tools for effective management and business development.

Given this need and the need of entrepreneurs for quick and affordable access to information, it is worth considering creating a business-oriented website with a built-in chat assistant based on the ChatGPT API. This innovative approach allows small and medium-sized businesses to not only manage their operations effectively, but also to receive recommendations and advice from artificial intelligence based on extensive experience and data from millions of projects around the world.

Also, with the high growth of Internet users and demand for online resources, the development of such a project is an important tool for small and medium-sized businesses seeking to strengthen their position in the market and have a chance to develop their business.

Due to the above reasons, in the context of the rapid development of information technology and the changing business environment and the vulnerability of small and medium-sized businesses in the modern market economy, the relevance of the research on the development of a business website with a built-in chat assistant based on ChatGPT API becomes undeniable. Small and medium-sized enterprises, which are the engines of the economy, need modern tools to effectively manage and develop their business. As well as consulting with artificial intelligence, which has experience and information about millions of projects around the world, will give these small entrepreneurs a new impetus for their further development.

The object of the research is the process of creating and implementing a business website designed for small and medium-sized businesses, and the subject of the research is the functionality of statistical analysis and a chat assistant based on the ChatGPT API built into this web application.

The goal of this bachelor’s thesis is to develop an innovative solution capable of providing small and medium-sized enterprises with the effective management and support of their business with the help of a built-in chat assistant based on ChatGPT API, which can provide real support to SMEs in this dynamic environment. The following tasks will be used to achieve the following goals:

1. Analysis of the requirements and needs of small and medium-sized businesses in the field of digitalization and automation of management processes.
2. Development and implementation of statistical analysis functionality on the website for convenient tracking of key business indicators.
3. Development and implementation of a database into the website.
4. ChatGPT API-based chat assistant integration to provide users with real-time advice and support.

Method of data collection. During the study, the method of collecting case studies will be used. Analysis of specific companies that have productively integrated digital technologies, and also work with a chat assistant. The purpose of the analysis is to collect as much data as possible about successful and unsuccessful cases of companies of different sizes.

Data processing methods.

1. Descriptive statistics. The collected statistics will be summed up into averages, various kinds of deviations, as well as minimums and maximums to describe quantitative data on the impact of digital technologies on business.
2. Regression analysis. This method of data collection will be used to identify connections, assess the impact of the level of digital adoption on the company's performance.
3. Qualitative analysis of the content. An in-depth analysis of specific materials about companies will be carried out .

Limitations of the study include limited access to chatbot APIs due to their fee-for-service policies, as well as tight deadlines for writing a bachelor's thesis.

A brief description of the main parts of the study includes an analysis of current trends in the development of business technologies, the process of developing and implementing a business website, as well as testing the effectiveness of using a chat assistant based on ChatGPT API in practice.

# ANALYTICAL REVIEW OF LITERATURE

## Concepts and meaning of Digitalization and Automation

Digitalization is the process of using digital technologies to transform a business model and create new revenue streams and value creation opportunities. This involves integrating digital tools and systems into various aspects of the business, from management and communication to production and customer service. In addition, digitalization is the process of simplifying and improving the efficiency of workflows and processes using digitized information. (Ribeiro‑Soriano 2023). Dealing with digitalization is key to business growth and survival, especially in a technology-driven world.

It is an approach to automating various business processes, workflows, and tasks of a company with minimal human intervention. Workflow automation uses software to create a series of automated actions to implement steps in a business process, helping teams complete tasks efficiently and consistently. (Goldberg 2012).

**Current trends in digitalization and automation.**

Within the framework of the fourth technological order (1985-2035), which is currently active, the world economy is increasingly integrating achievements in the field of artificial intelligence (AI). This trend predicts even more significant changes in the sixth technological paradigm (2035-2045), where the emphasis is on "unmanned" technologies. It is already visible how artificial intelligence and robotic systems are contributing to the domestic, industrial, medical and public sectors and are becoming an integral part of social and economic life. (Allam H 2022).

The impact of AI on various sectors of the economy. By 2025, the revenue from home robots will reach $33.2 billion, accounting for 50 percent of the robotics market, and manufacturing robots will capture 25 percent of the market or $16.6 billion. Healthcare and consumer services, along with the public sector, are showing significant growth, confirming the general trend of automation and digitalization. (cram 2024).

AI as a driver of business processes for small and medium-sized businesses. Artificial intelligence plays a key role in the digitization and automation of business processes. AI-enabled web platforms provide small and medium-sized businesses with the tools they need to analyze big data, allowing them to remain competitive in a rapidly changing market. These platforms help streamline processes by providing guidance and automated solutions for everyday tasks.

Examples of successful integration of AI into business. One example is intelligent robotic systems, which are being actively implemented in industry and everyday life. Working in synergy with artificial intelligence, these systems are able to perform a wide range of tasks, from manufacturing to healthcare, significantly increasing their efficiency and reducing human error.

Potential risks and prospects. The transition to the sixth technological mode does not mean the complete removal of the technologies of the fifth mode, as can be seen in the example of the United States, where elements of the fifth mode still dominate in the integration of elements of the sixth. According to experts, the world community, which missed fifth place, is important to actively participate in the development and implementation of new technologies in order not to remain on the periphery of the world economy.

Artificial intelligence is already transforming business processes, showing enormous potential, especially for small and medium-sized businesses. The introduction of AI into web platforms not only improves the efficiency of enterprises, but also opens up new opportunities for growth and development. This is especially true in the context of global technological changes, for which all countries must be especially prepared. In the near future, information technologies of automation and digitalization, including artificial intelligence, will shape the economic landscape of states, and the implementation and availability of digital infrastructure in the enterprise will become a key factor for success in the business arena.

**The difference between digitalization and automation.**

Digitalization and automation differ in that digitalization focuses on converting processes and data into a digital format, whereas automation focuses on using technology to perform specific tasks.

Other key differences between digitization and automation are:

1. Goal: The goal of digitalization is to increase efficiency, improve the quality of service and create new opportunities. The goal of automation is to increase efficiency and reduce costs.
2. Field of application: Digitalization can be used in any field of activity, while automation is most often used in business and production.
3. Technology: Digitization uses a wide range of technologies, including computers, software, networks, and sensors. Automation can use robots, software, or other technologies. (Moiseev 2023).

Digitalization and automation are often used together. Digitalization can create the basis for automation by transferring data and processes digitally. While automating business processes can help companies increase their efficiency and significantly reduce costs, which can lead to further progress in digitalization.

Examples of combined use of digitalization and automation:

1. In business: Digitalization is used to create online stores, which in the future need to be automated for order processing, delivery of goods and customer consultation.
2. In healthcare: digitalization is used to store medical records in electronic form, which will later be automated for access to information and provision of services.
3. In the field of education: digitalization is used to create online courses that will need to be automated to deliver content and measure results. (Moiseev 2023).

Digitalization and automation are important trends that will increasingly affect all areas of life. As these technologies advance, people can expect new and innovative ways to use them.

## The role and impact of digital technologies on small and medium-sized businesses

Nowadays, there are relatively few studies that analyze the relationship between digitalization and innovation in SMEs. In this regard, in this analytical review, the author will take statistics on the German innovation ecosystem due to the greater amount of information on the country itself and the EU.  The German innovation ecosystem supports the digitalization of SMEs through various policies and programs. Competence Centers have been established since 2015 to give SMEs a holistic approach to digitalization and to support companies in taking the first steps towards digitally connected businesses. As for big data analytics, 18% of German enterprises have implemented it, which is higher than the EU average 14%. As for artificial intelligence technologies, 28% of German enterprises use them, which is higher than the EU average 25%, and 57% have medium or high intensity of environmental action through ICTs compared to 66% on average in the EU. (Radicic 2023).

The introduction of digital technologies can affect any stage of innovation processes, from innovation costs, such as R&D spending and other innovative investments, to innovative outcomes, such as product and process innovations. (Meffert 2019). Digitalization contributes to the development and acquisition of new skills, competencies and knowledge, which in turn can lead to the creation of new products and processes. Moreover, the ability to assimilate, as an element of the management of the firm's economic processes, critically depends on access to the internal network and control over knowledge.

Digital technologies that enable the collection, storage and processing of data, such as artificial intelligence, the Internet of Things and blockchain, offer new opportunities for SMEs. These opportunities include, among others, market entry, participation in global value chains and international trade. Moreover, companies are using digital technologies to transform their supply chain models and identify new ways to deliver value to customers.

Based on several studies on the impact of digitalization on SMEs, recently identified four reasons that explain why digital transformation is slow to be adopted by SMEs. First, small companies with their specific focus are less exposed to the need for rapid digitalization. Second, small companies often lack the resources and management vision to fully understand the impact of digital transformation on company performance. Thirdly, SMEs usually take a gradual approach to digitalization, unlike larger companies. Finally, investments in digitalization within companies of this type rely heavily on the financial resources of firms, which are limited, especially in micro and small firms.

**Analysis of companies that have implemented digital technologies.**

**DHL**. The COVID-19 pandemic has intensified the company's efforts to go digital. Investing more than $2 billion in digital transformation projects between 2021 and 2025. Here are some of the most prominent cases that have accelerated digital transformation.

1. Advanced Quality Control Center (AQCC). Millions of parcels are delivered around the world through countless logistics services, many suppliers and hundreds of thousands of touchpoints. What does that mean? DHL has invested in a state-of-the-art quality control center. Its main functions are the interpretation of big data on delivery issues, real-time problems in cargo/air transportation. Thanks to artificial intelligence (AI) and machine learning, AQCC is constantly adapting and improving.
2. Improving the customer experience through digitalization. Round-the-clock chatbots providing real-time information about the location of packages, on-demand delivery that allows users to schedule the delivery time of their contents, as well as physical contact. (DHL 2024).

**Wolt.** Wolt started its journey as a small sized business by implementing digitalization in the food delivery industry. They developed a platform using mobile applications and route optimization algorithms.

1. Order growth: Since the start of digitalization, Wolt has achieved impressive growth in orders on its platform. Over the past year, the number of orders has increased by 50%, which indicates that users appreciate the convenience and speed of the ordering process through the mobile app.
2. Optimization of delivery. Delivery optimization. The company reduced delivery times by 30 percent using route optimization algorithms. Reducing delivery times by 30% resulted in increased customer satisfaction with deliveries and increased likelihood of repeat orders.
3. Revenue growth: Digitalization of order delivery increased the company’s revenue by 40% in 2022. This result exceeded the forecasts of investment firms and business analysts, which indicates the success of the company’s digitalization strategy. (WOLT 2024).

**Problems and challenges of business during digitalization.**

Digital transformation is not only about the introduction of new software, technologies, and processes that are more efficient and automated than traditional business practices and processes; It's a whole new, innovative way to do what business is. A company's digital transformation is challenging for several reasons.

1. Software and technology complexity. Enterprise lost collateral can be challenging in the initial stages of implementation. This is a major challenge for organizations undergoing digital transformation, both with data access points and end-user observation points.
2. Staff training. New processes of digitalization and automation of work may be negatively received by full-time employees who cannot or do not want to switch to a new way of working. For this reason, when implementing new software.
3. Lack of necessary IT skills. In order for digitalization and automation of business processes to be successful, the company must have a qualified and high-performance team of IT specialists.
4. Security problems Security issues. The challenges facing many organizations that handle mission-critical data are privacy and cybersecurity concerns. Most often, the digital transformation of such companies involves abandoning local solutions and moving to the cloud and integrating all company data into one centralized system for ease of control.
5. Isolated organizational structure. In many organizations, departments or teams operate in silos, leading to poor collaboration and communication, ineffective resource allocation, duplication of work, and fragmented approaches that undermine the success of digital transformation initiatives. Poor organizational structure and lack of collaboration between teams can hinder the flow of ideas and innovation, and hidden data can negatively impact decision making. (Olmstead 2022).

**Interaction of digitalization and R&D in innovative activity**

As part of the analytical work on the state and prospects of digitalization of small and medium-sized businesses, the author's work is based on valuable empirical data provided in a study conducted by Dragan Radicich and Saša Petkovich.

In particular, they study how investments in digital technologies and internal research and development (R&D) correlate with a company's ability to innovate. The research methodology developed by Radičić and Petkovic includes careful data analysis, the use of correlation matrices to assess relationships between variables, and the use of two-dimensional probit models for Explore the impact of digitization on product and process innovation. The correlation matrix can be found in Table 1.

Table 1. Correlation matrix (Radicic 2023).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Independent variables** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Digital prod\_logistics | 1.000 |  |  |  |  |  |  |  |
| Digital value chain | 0.444 | 1.000 |  |  |  |  |  |  |
| Big data | 0.229 | 0.216 | 1.000 |  |  |  |  |  |
| R&D | 0.209 | 0.099 | 0.143 | 1.000 |  |  |  |  |
| Productivity | 0.074 | -0.001 | 0.031 | 0.085 | 1.000 |  |  |  |
| Exports | 0.088 | 0.003 | 0.036 | 0.366 | 0.277 | 1.000 |  |  |
| Graduates | 0.226 | 0.136 | 0.118 | 0.262 | 0.162 | 0.186 | 1.000 |  |
| Firm size | 0.172 | 0.036 | 0.086 | 0.204 | 0.189 | 0.233 | 0.265 | 1.000 |

The icons \*\*\*, \*\*, \* indicate the level of statistical significance (p <0.01, p <0.05, p <0.1 respectively).

The table shows that digital productivity and logistics are associated with the successful use of digital value chains, big data, and developments in the company. Indicates that companies that use digital technologies in logistics are more likely to develop or use innovations in other areas. Also, the data show us that R&D has a strong correlation with export activities, which indicates that companies that are actively engaged in research often face the need to enter the international market, that is, expand. Firm size correlates with virtually all of the variables studied, emphasizing that large companies have the resources to integrate and embrace a variety of technological and innovative approaches.

Based on these data, firms and enterprises can draw conclusions on the aspects of their professional activities. That is, to understand what improvements should be made to achieve greater innovation and competitiveness. The results of empirical research can be seen in Table 2.

Table 2. Empirical Results of Two-Dimensional Probit Models. (Radicic 2023).

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Micro firms** | **Small firms** | **Medium Firms** |
| Digital prod\_logistics | Processes: 0.394\*\*\* | Processes: -0.007 | Processes: 0.336\*\*\* |
| Digital value chain | Groceries: \*0.284\*\*  Processes: \*0.330\*\* | Groceries: 0.134 Processes: 0.176 | Groceries: 0.133  Processes: 0.180 |
| Big data | Groceries: 0.028  Processes: -0.051 | Processes: \*0.270\* | Groceries: \*0.300\* |

\*\*\*, \*\*, \* indicate the level of statistical significance (p <0.01, p <0.05, p <0.1, respectively).

Table three examines the impact of various technological variables on the likelihood of innovation in products and processes in firms.

In micro firms, digital productivity and logistics increase the likelihood of process innovation, while the digital value chain moderately drives innovation in both products and processes. The table shows that for small firms, big data has a positive impact on the chances of product innovation. In medium-sized firms, digital productivity and logistics drive process innovation. While big data to a lesser extent but also increases the probability in products.

Based on this data, it can be concluded that investments in digital technologies and automation of business processes can be different in effectiveness depending on the size of the enterprise and the type of innovative activity.

Looking at Figure 1, which shows the results for micro firms, in the top left graph, R&D activity increases the probability of product innovation by 28.2 percentage points when digitalization = 0, and this increase is 30.6 percentage points. When digitization = 1 (although this difference is not statistically significant at any conventional significance level). For process innovation (top right graph), R&D activity increases the likelihood of process innovation by 7.5 percentage points. increases. With digitalization = 0, this increase is 6.9 bps. When digitized = 1 (the difference is statistically insignificant). For complex innovators (bottom chart), R&D increases the likelihood of both types of innovation by 24.4 percentage points. increases. With digitalization = 0, this increase is 28.2 bps. When digitized = 1 (but again the difference is not statistically significant).

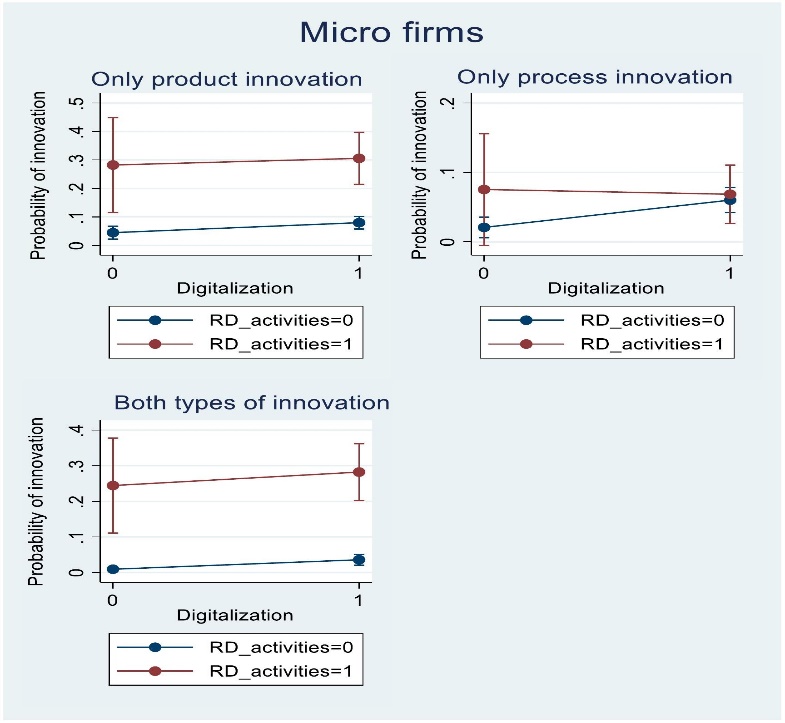


Figure 1. The Impact of Digitalization and R&D on Innovation in Micro Firms.

(Radicic 2023).

Figure 2, which presents the results for small firms, shows that R&D activity increases the likelihood of product innovation by 28.9 percentage points. shows that it is increasing. With digitalization = 0, this increase is 27.7 bps. When digitized = 1 the difference is statistically insignificant. For process innovation top right graph, R&D activity increases the likelihood of process innovation by 11 points. When digitization = 0, this increment is 12.2 bits. When digitized = 1 . For complex innovators, R&D increases the likelihood of both types of innovation by 37.9 percentage points. increases. With digitalization = 0, this increase is 34.6 bps. When digitized = 1.

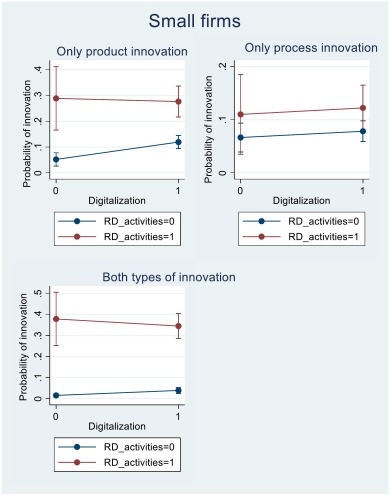


Figure 2. The Impact of Digitalization and R&D on Innovation in Small Firms.

(Radicic 2023).

Figure 3 shows the results for medium-sized firms. The top left graph shows that R&D activity increases the likelihood of product innovation by 20.8 percentage points. shows that it is increasing. With digitalization = 0, this increase is 21.9 percentage points. When digitized = 1. For process innovation, R&D activity increases the likelihood of process innovation by 18.5 percentage points. increases. With digitalization = 0, this increase is 17.2 bps. When digitized = 1. For complex innovators, R&D increases the likelihood of both types of innovation by 24.4 percentage points. increases. With digitalization = 0, this increase is 37.6 bps. When digitization = 1. As explained in footnote 8, the difference in effect is statistically significant (p < 0.05). This result shows that R&D enhances the effect of digitalization only in medium-sized companies that are experienced innovators. A potential explanation for this finding may be related to the greater financial and human resources possessed by medium-sized firms compared to micro and small firms. In addition, complex innovators have higher innovation potential than firms that focus only on new products or new production processes.

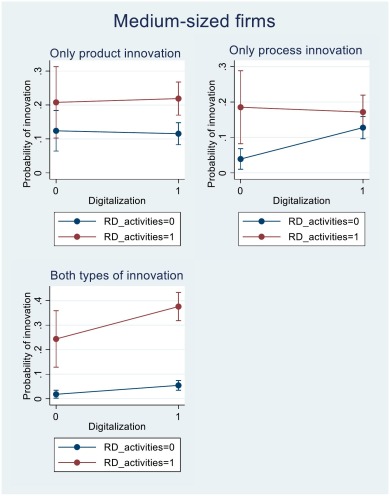


Figure 3. The Impact of Digitalization and R&D on Innovation in Medium-sized Firms.

(Radicic 2023).

An analysis of the graphs for the three categories of firms highlights that R&D is an important driver of innovation in all size groups. Digitalization is amplifying this effect, especially in medium-sized firms, which may indicate their greater potential to implement comprehensive innovation strategies. The lack of in-house R&D does not deprive firms of the opportunity to innovate through digitalization, but with a lower growth rate in the likelihood of innovation.

## Artificial intelligence and chatbots in supporting business processes.

A chatbot imitates human speech by performing repetitive, automated actions based on predefined triggers and algorithms. A bot is designed to communicate with a human using a chat interface or voice messages in a web or mobile application, just as a user would. (IBM 2024)Chatbots, a type of conversational artificial intelligence, are similar to virtual assistants. Thanks to advanced natural language processing, custom AI chatbots, especially those based on GPT-3.5 and GPT-4, speak to the user in a manner that closely mimics human speech.

The most important ability of these chatbots is their ability to work with specific business documentation or specialized files that a specific enterprise need.

Essentially, these chatbots can automate routine tasks, offer support to internal teams, and even help with data analysis, freeing up valuable human resources to focus on more strategic initiatives. (OPEN AI 2023)

**Review of existing AI models used in chatbots.**

In this part of the literature analysis, the author will give examples of existing AI models used in chatbots.

**ChatGPT** is an AI chatbot from OpenAI that can successfully mimic human conversation when communicating with a user. It is based on the revolutionary GPT-3 language model based on huge datasets with over 175 billion parameters. (KUMAR 2023)

A chatbot can not only conduct meaningful conversations in the question-answer mode, but also create unique emails, essays, descriptions of goods and services, poems, scientific articles, resumes, as well as by adding code based on the relevant voice or text requests (prompts).

ChatGPT Features.

1. Uses OpenAI's advanced GPT-3.5 or GPT-4 NLP models (paid version of ChatGPT Plus).
2. Capable of performing a wide range of generative AI tasks, from writing text and code to solving mathematical problems.
3. Thanks to its intuitive interface, it is convenient and easy to use, even for those who do not have technical experience.
4. Searches the web just as efficiently as popular search engines. The only significant difference is that the results are returned in the form of ready-made answers, without references to the original sources.
5. Effectively copes with regular compilation of text for mailings and automatically generates meta tags.
6. Helps users generate new content and ideas, making it a useful tool for overcoming writer's deadlock syndrome and blank page.
7. ChatGPT's incoming "message" is limited to 500 words.
8. The chatbot does not have an internet connection and cannot use its channels in real-time. The knowledge base of the ChatGPT language model is limited to the fall of 2021. The developers explain this decision by the desire to preserve the security and privacy of user data.
9. ChatGPT can remember previous messages in a conversation and use this context to generate appropriate responses. It takes into account the entire chat history and uses it to generate relevant responses. (ChatGPT 2024)

**Bing -** is a new version of the search engine with chatbot integration based on the latest NLP model GPT-4, which appeared in early February 2023. (Harrison 2023).

Bing Features.

1. Ability to write code in C++ and Python to help solve various programming problems.
2. Assistance in search engine optimization (SEO), providing personal recommendations for websites and blogs.
3. Ability to create articles for blogs and websites. But it is necessary to check the relevance of the texts.
4. Providing tips and recommendations, examples on choosing gifts or interesting ideas.
5. Creating stories or poems thanks to the ability to work in a creative mode.
6. Generating requests for Midjourney (platform for testing chatbots).
7. Planning trips, collecting information from various sources.
8. Help you find relevant information in real time.

**Bard** - is based on the Google Transformer neural network architecture, which has also become the basis for other generative AI tools, such as the GPT-3.5 language model used by ChatGPT. At this stage, the Neural Network is available by appointment to citizens of the United States and the United Kingdom. (INSTAGANTT 2024).

Bard's Opportunities.

1. Based on a lightweight version of the larger Google LaMDA (Language Model for Conversational Applications) language model.
2. In addition to being a member of the government, Bard requires a personal Google account and must be 18 years old.
3. Bard supports the following web browsers: Chrome, Chromium-based Edge, Firefox, Opera, and Safari.
4. The bard does not know how to write code. This feature may be added later.
5. Built-in security features. The chatbot will not answer questions that are ethically incorrect or beyond its capabilities.
6. As with Bing, when you use Bard, you'll feel like you're using a regular search engine, only enhanced by AI capabilities.

**Chatsonic** - AI chatbot from Writesonic. It is a product with many features similar to ChatGPT, but it is based on the latest NLP GPT-4 model and has access to the internet and the latest information. In addition to generating text, Chatsonic is capable of creating unique images based on suggestions received from the user. (What'sthebigdata 2024).

Chatsonic Features.

1. Based on OpenAI's large GPT-4 language model.
2. Supports 25 languages.
3. Chatsonic has internet access and Google search integration. It helps you write the most relevant content based on relevant information obtained in real-time.
4. ChatSonic AI has built-in integration with Stable Diffusion and Dall-E generative neural networks, which allows you to supplement texts with unique AI-generated images.
5. The voice input feature, similar to Siri and Google Assistant, is a great time-saver when working with Chatsonic.
6. Quick integration with external tools using the Chatsonic API.
7. Built-in Chatsonic Chrome extension.

* "Personality" mode, which allows you to "program" the transponder signal based on 16 individual patterns.
* Sonic Editor, an online text editor, analogous to Google Docs.
* Saving information from chats.

AI chatbots such as ChatGPT, Bing, Bard, and Chatsonic are based on language models (GPT-3, GPT-4) developed by OpenAI. These chatbots can perform a wide range of tasks, from writing text to integrating with search engines and creating images, making them useful in a variety of fields, from education to marketing. All models have advanced natural language processing capabilities and ease of use, making them accessible to a wide audience of users around the world.

**Analysis of the impact of chatbots on improving the efficiency of business processes.**

The use of AI chatbots has grown significantly since their inception. 58% [iv] of B2B companies actively use chatbots. In fact, by 2027, AI chatbots are predicted to become the primary customer service channel for a quarter of businesses. Citing Zendesk's 2023 Customer Experience Trends report, 72 percent of entrepreneurs said using artificial intelligence and chatbots in customer service is their top priority in the next 12 months. (Saini 2023).

AI-powered chatbots can help improve customer service capabilities, increase operational efficiency, and provide personalized customer service. You can also use this scalable solution to efficiently manage customer inquiries and support needs.

The use of chatbots and artificial intelligence in marketing has been revolutionary for businesses in recent years. These technologies have the potential to transform the way businesses interact with their customers, increasing efficiency and effectiveness while reducing costs.

One of the main benefits of chatbots and artificial intelligence is the ability to deliver personalized experiences at scale. Using data about individual customers, these technologies learn and tailor interactions to specific needs and preferences, thereby improving the customer experience.

Another key benefit is the potential for advanced analytics and insight. Chatbots and artificial intelligence can collect vast amounts of data about customer behaviour, preferences, and pain points. This information is used to enhance marketing strategies and improve product or service offerings. Among other benefits, sentiment analysis algorithms identify patterns in customer feedback and identify areas for improvement. Moreover, automation technology allows marketers to test campaigns and optimize their performance based on real-time results quickly and accurately.

AI can also improve companies' agility and ability to respond to changes in the competitive business environment by supporting critical business functions such as automating business processes, gaining insights through data analytics, and interacting with customers and employees. And analyse data, chatbots provide customized offers and solutions to customer requests while providing a seamless and satisfying experience. In addition, chat bots that work with AI facilitate customer interaction by effectively solving problems and releasing employees to focus on more complex tasks and improving customer satisfaction. (McGee 2024).

**Analysis of companies that have successfully implemented chat bots.**

1. Marketing and sales. Sephora, the largest cosmetics retailer, uses AI chatbots to increase customer engagement and sales. They have integrated a chatbot for booking so that customers can book appointments with in-store beauty professionals. (Saini 2023).

In addition, their chatbot provides an augmented reality feature where users can upload photos to find products of the right colour. These chat services allow Sephora to engage in a dialogue with consumers and offer personalized experiences that ultimately increase sales.

2. Customer support. On the operational front, Starbucks has used artificial intelligence to improve the productivity and efficiency of its stores. The deployment of AI-enabled coffee makers and other smart equipment in stores has simplified beverage preparation processes, reduced wait times and improved the customer experience. This automation extends to workforce allocation and inventory management, where AI analytics predict customer flow and product demand, allowing managers to optimize staffing levels and inventory accordingly. This not only keeps stores running smoothly during peak hours, but also helps reduce waste and improve profitability. The integration of these AI solutions reflects Starbucks' commitment to combining technology with human expertise to create a unified and efficient operating model. (subscribed.fyi 2024).

3. Operations: To ensure a safe and seamless boarding experience for drivers and passengers, Uber's data scientists created One-Click Chat, a mobile UberChat feature that delivers fast, dynamic and personalized smart responses.

Lyft's live assistance features chat features powered by bots powered by artificial intelligence and machine learning. AI is used to predict and anticipate questions a customer might have and then personalize the passenger experience, working through conversation to solve the problem.

# CASE STUDY

## Website requirement definition.

Methodology 5W Brand Sherrington. It is based on five questions, each of which begins with the letter W. The method was invented by Mark Sherrington, the founder of the Added Value consulting firm, known throughout the world. The marketing model is based on a simple rule: every product has its own buyer. (inSales 2022).

The main task of 5W is to identify separate segments of the audience with similar characteristics, and then create an individual proposal for each group.

1. Who? - The website is designed to serve small and medium-sized enterprises (SMEs), including startups and entrepreneurs who want to strengthen their market position or grow their business. My research has shown that this audience is looking for affordable and effective ways to manage and grow their business.
2. A what? - The main feature of the site is assistance through a chatbot, which uses artificial intelligence to give advice on business planning, current changes in legislation, and tax issues. The bot also provides sample documents, making it an indispensable tool for entrepreneurs.
3. Where is? - The analysis showed that the target audience is actively looking for information on specialized forums and search engines, and also often turns to legal advisers. The site will provide convenient and centralized access to the necessary information, reducing the time for search and consultations.
4. When? - The greatest user activity is expected at the beginning of the financial year, when SMEs plan their budgets and business strategies. In addition, as the analysis showed, important factors in accessing the site are moments of business crisis, for example, problems with scaling or difficulties in launching a new project.
5. Why? - The choice of our site was due to several main factors: an intuitive interface, the reliability of the system supported by the OpenAI corporation, the relevance of the information provided, and the availability of document templates. These factors make the site not only convenient, but also an extremely useful resource for entrepreneurs.

This method helped the author create a foundation for further development of the site, ensuring that every aspect of the platform would meet the real and current needs of users. This structured study not only allows you to accurately determine the requirements for the functionality of the site, but also provides the basis for its unique positioning in the market in comparison with competitors.

After understanding the needs of the website and our users, by using the 5W method, it is also important to look outside to those who already offer similar or alternative solutions on the market. Analyzing and understanding how other companies address similar user needs and what solutions they have implemented to solve them will provide critical data for the formation of a unique website.

Competitor Analysis – also known as competitive analysis and competitive analysis, it is the process of researching similar brands in your industry to gain insight into their offerings, branding, sales and marketing approach. Below is the reason of conducting a competitive analysis.

1. Understanding of industry standards to ensure project compliance with standards.
2. The opportunity to occupy empty market niches.
3. Differentiation of products and services.
4. Satisfy users' desires and solve their problems better than competitors.
5. Brand differentiation
6. Height measurement Competitor Summary Analysis: PepperVA and DoNotPay

PepperVA - Pepper Virtual Assistants provides expert virtual assistance services to businesses and individuals around the world. Since its inception in 2009, Pepper VA has helped numerous businesses streamline their processes and scale their operations while saving time and money. (PepperVA 2024).

1. Primary activity: Providing virtual assistants for small businesses, for solving administrative and client services.
2. Strengths: Personalization of services, flexibility of the approach focused on the individual business, quick adaptation to the requirements of the company, close interpersonal interaction.
3. Weaknesses: Dependence on the human factor, which can lead to errors and unstable quality of service.

DoNotPay - is an artificial intelligence chatbot that helps consumers fight corporations, beat red tape, find hidden money and protect their rights. The chatbot was created by British entrepreneur Joshua Browder in 2015 to help people appeal parking fines. It has since expanded to help users solve more than 1,000 consumer issues and challenges.

DoNotPay uses artificial intelligence and machine learning to automate email creation, phone calls, form filling, and other tasks to help customers solve complex customer service problems and get the results they deserve. A chatbot acts as a virtual assistant on your behalf dealing with big companies, unfair fees and confusing government processes.

1. Main activity: Automation of solving bureaucratic, legal and administrative tasks through an AI chatbot using an interactive platform.
2. Strengths: High degree of automation, efficient solution of everyday tasks, use of process optimization technologies.
3. Weaknesses: Possible low-quality information in complex legal situations, dependence on the accuracy of user data. (DoNotPay 2024)

Comparative analysis

1. Technological innovation: The advantage of DoNotPay is the use of the latest technology to solve problems, which can lead to higher overall efficiency compared to PepperVA, where the focus is on the human factor.
2. Service Personalization: PepperVA surpasses DoNotPay in its ability to provide personalized services that are highly valued in niche and specialized market segments.
3. Market positioning: DoNotPay may be more appealing to users looking for simple solutions to everyday tasks, while PepperVA may appeal to customers looking for deeper and more personalized support.

Table 3 will be provided below, which will illustrate the main weaknesses of competitors PepperVA and DoNotPay, as well as potential directions for the development of the project, to maximize the use of the identified vulnerabilities.

Table 3. Competitors' Weaknesses PepperVA and DoNotPay. (Source: author's summary)

|  |  |  |  |
| --- | --- | --- | --- |
| Criterion/Competitor | PepperVA | DoNotPay | Potential possibilities |
| Technological Innovation | Limited use of AI and automation. | High level of automation but may not be effective in complex situations. | Development of advanced AI solutions for complex tasks. |
| Personalization | Heavy reliance on human interaction, which can lead to inconsistent quality. | Low level of personalization in service. | Improving AI algorithms to deliver personalized services. |
| Complex legal/bureaucratic tasks | Does not specialize in complex tasks, which may reduce their appeal to a certain audience. | Efficiency is reduced in complex legal cases that require detailed analysis. | Development of modules for in-depth analysis and processing of complex requests. |
| Customer Experience | Human errors and inconsistencies in service are possible. | The lack of deep personalization can be a turn-off for users looking for specific solutions. | Create an intuitive user interface with detailed customization. |
| Scalability | Scaling services can be difficult due to the dependency on the quality of individual task performance. | The model may not be suitable for scaling in the face of intense demand growth. | Build a scalable architecture that can adapt to changes in supply and demand. |

This analysis provides a framework for designing a website that will meet both the basic user requirements but also outperform competitive products in key areas such as user interface, functionality, and unique offerings.

The author decided to create a full-fledged website for entrepreneurs, where they will be able to consult with a chat bot, which will have the latest data on Latvian business legislation, data on successful Latvian enterprises and their business structure in the country. Also, the chat assistant will focus specifically on helping entrepreneurs, without philosophizing about other topics in the dialogue. In addition to the chat bot, users will be able to work with statistics on their business sales. And store users’ personal business documents, such as tender documents, documents with business data tables in Excel form, and other important documents. The author’s goal, after analyzing competitors, is to create a platform that entrepreneurs will use as often as CRM systems. The future functionality of the site will be described below.

**Functional part of the website.**

Uploading and downloading documents.

1. Downloading documents using the download interface.
2. Uploading documents using the upload interface, by clicking on a specific document to upload.
3. View a list of all user documents.

Authorization.

1. Login to the system with verification of the user's mail and password, according to the data from the Database.
2. Persistent user session using the local database for tokens.

Sign up.

1. Register in the system after filling out a list of registration fields such as first name, last name, email address, password and repeat password.
2. Feedback on registration errors.

Communication with artificial intelligence.

1. Receiving and processing the user's request to display responses from the bot chat.
2. Practical advice from the chatbot about the business vision and business-related issues.
3. User document analytics.
4. Scroll to the last message in the interface.

Timetable and statistics.

1. Special fields for entering information about sales for a certain period.
2. Display the average sales of the user, based on his entered data.
3. Displaying and updating the graph based on the user's entered data.

Work with video materials.

1. Integrate and show videos from the YouTube platform related to building a business environment and general educational content on entrepreneurship.
2. Ability to view videos on the website.
3. Adaptability of the layout to different user screens.

Navigation.

1. Navigating from pages using VueRouter to work with a specific page.
2. Content loading, dynamic, without the need to reload the page.

User Content Management.

1. Displaying additional content, based on the user's authentication status.
2. Modifying the elements of the website and its visibility, focusing on the user's actions and authentication.

Error notifications.

1. Visualize the dashboard with an error message for failed actions, such as logging in or registering.
2. Notification of successful or unsuccessful operations.

**Non-functional part of the website.**

Website security.

1. Implementation of encryption for the user's personal data, from transmission using HTTPS and during storage.
2. Implement an authentication and authorization mechanism to prevent unauthorized login and leakage of user data.

Website usability for users.

1. Development of an intuitive interface, according to WCAG regulations
2. Implementation of the interface, so that all the main functions of the site are available using three clicks.
3. Provide help documentation and supporting tips.

Website compatibility.

1. The website should be displayed in the same way on different browsers.
2. Adapt the design of the site so that the functionality and layout of the site are displayed on different screens.
3. The system must successfully interact with external systems, such as the YouTube platform and OpenAI. YouTube for watching videos by transmitting data via JSON, using open standards like HTTP/HTTPS and OAuth 2.0 authentication mechanisms. OpenAI to work with a chatbot, by creating a gateway to process the API from OpenAI in the server code.
4. Use the classic JSON method of data exchange, and open standards for working with APIs.

Website maintenance.

1. The codebase of a website should be well organized and documented for future modernization and debugging.
2. Use modular architecture for easy upgrade and maintenance.

Use Cases

1. User registration: The user fills out the registration form and submits the data for verification, the server checks the uniqueness of the email, then the password is verified. If the email is unique, the system sends a notification about the successful registration of the user, after successful registration, the user goes to the login page.
2. Log in: The user enters his email with a password, the server checks the entered data against the database, after successful verification, the server creates an authentication token and stores it in the local server, and the user automatically goes to the main page. In case of unsuccessful authorization, the user receives an error notification about the incorrectness of the entered data.
3. Documents: When the user clicks on the download button, the protrude window opens where he can select the file to download. After successful uploading, the user has a list of his personal documents, which he can download by clicking on the download button,
4. Using a chatbot: A user enters a request or question in the input field in the chatbot interface, after sending a message, the chatbot analyzes the user's message, and sends a response that is displayed in the user interface.
5. View statistics. The user enters sales data in the euro currency, after sending the data, the server processes the data and updates the database, after which a schedule is created based on the entered data. Also, the user has the opportunity to see his average number of sales.

## Selection of Technologies and their rationale

**Front-end technologies**

The author decided to use the Vue.js framework. Vue.js is a lightweight front-end JavaScript framework. Vue.js is a model-view-view (MVVM) JavaScript framework for building user interfaces (UIs) and single-page applications. Vue.js was created by Evan Yu, a former Google engineer who quickly became disillusioned with the use of AngularJS in Google projects. He decided to take the best features of Angular and create something incredibly lightweight.

The MVVM architecture Vue.js ensures that the presentation logic is separate from the business logic that represents the view and the model, respectively. The view model then acts as an intermediary in preparing the data objects for representation. Also, a feature of the Vue.js is its directives. Directives extend HTML with HTML attributes, extending the functionality of HTML applications. These directives are built-in, but can also be defined by the user. Essentially, they allow you to manipulate the Document Object Model (DOM). The DOM is an interface, specifically an API for HTML and XML documents, that instructs machines on how to structure text. The concept of directives exists in both Angular and Vue.js, but Vue.js provides the ability to add this concept to any part of the server application and manually configure a more interactive version of the interface for consumers. At its core, this feature allows HTML elements to encapsulate reusable code. JavaScript frameworks like Vue.js promote more organization when it comes to front-end development. Vue.js can take a web page and break it down into reusable components, each with its own HTML, CSS, and JavaScript elements. Also, Vue.js has two tools, Vuex and Vue-Router. They work with the Vue.js's core software to provide. (VueJS 2024).

1. Stateful Control with Vuex – UI controls such as text boxes, OK buttons, etc., may need to be managed outside of the current page the user is visiting.
2. Routing with Vue-Router – This is the process that occurs when you need to synchronize URLs with views in your application.

The decision to develop the front-end part of the project in this framework was made after analyzing a number of its advantages, which were important when choosing a tool for the future version of the site.

1. Lightness: Vue.js, for all its features, has the smallest JavaScript code size among its competitors Angular and React. For comparison, the latest version of React weighs only 7 kB in minified form, but a mandatory dependency in the form of react-dom will bring another 121 kB to the JavaScript bundle. A clear example of the Vue.js site can be seen in figure 4.

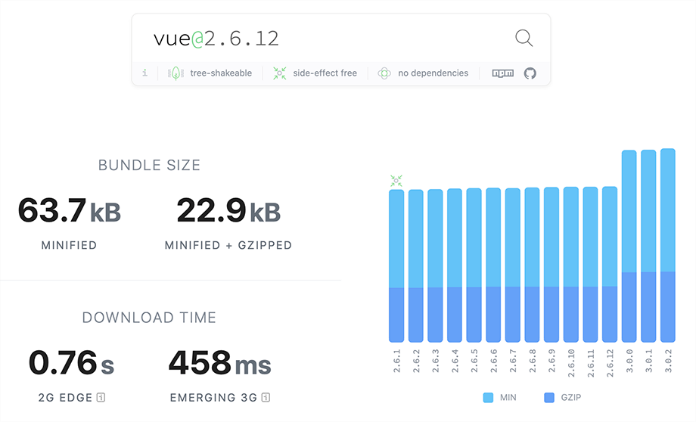


Figure 4. File size Vue.js. (bundlephobia 2024)

1. Easy to learn: Vue.js has excellent documentation that is also translated into several languages. At the same time, it not only provides access to a description of the main features of the framework and its API, but also contains various examples of real configurations and implementations of applications, recommendations, links to various training materials, etc.
2. Toolkit: Vue.js offers a rich set of testing and development tools, including built-in debugging and state management tools.
3. Community: Vue.js has an active and useful online community, and the platform is independent of large companies like Facebook and Google.
4. DOM Management: Vue.js ensures smooth integration of the existing HTML markup returned by the server, known as the base application template. Simply initialize the application in the root element of the DOM and the system will automatically apply the view. This feature sets Vue.js apart from other popular tools by offering a simpler interface.
5. Custom Vue.js directives make it easier to interact with elements in the DOM tree at a low level. They allow you to easily customize the listening and handling of events, as well as make changes to the approaches to the DOM components.

After analyzing different frameworks for creating a website and determining the needs of users, the author chose the Vue.js framework for front-end development. The choice was made due to the compactness, efficiency of the MVVM architecture, ease of learning, and a powerful set of state management and routing tools. These characteristics make Vue.js the preferred solution for building a website with a built-in UI chatbot.

**Technology Back-End.**

For the development of the backend, the Flask framework based on python was chosen.

Flask is a micro framework for developing web applications in the Python programming language. It provides a set of tools and extensions that allow developers to quickly create simple and scalable web applications. Flask is based on two main components: Werkzeug, a toolkit for working with WSGI, and Jinja2, a blockage template for generating HTML. (Python Tutorial 2021).

According to Stack Overflow, Flask is one of the twenty most popular frameworks by developers. Below are the statistics from the Stack Overflow website in figure 5.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Figure 5. Flask Framework Rate. (stackoverflow 2020)

The choice of Flask to develop a website with a built-in chatbot using OpenAI's API is based on its ability to adequately handle the dynamic and scalable designs that are essential for an effective user experience. The choice of backend implementation in Flask came from its convenience in developing websites. Because Flask is a Web Server Gateway Interface framework. It's basically a way for web servers to pass requests to web applications or platforms. Below will be additional reasons to choose this framework.

1. Scalability. Flask is a micro framework, which allows it to be used for the rapid development of a technical project, such as web applications. It is convenient when creating a website that will start with a small format, but at the same time will have the potential for growth in those areas that have not yet been fully developed. Its ease of use and low dependencies allow it to run smoothly even at scale.
2. Flexibility. This is the main feature of Flask and one of its biggest advantages. This is useful in terms of the fact that the project can be easily moved in a different direction, and the structure of the project will not collapse if any part of it changes.
3. Easy to use. The micro framework is easy to understand and helps a web developer to quickly navigate and concentrate only on writing code, with minimal attention spent on breaking down processes.
4. Simplicity. Flask supports modular programming, in which its functionality can be divided into several interchangeable modules. Each module acts as an independent building block that can perform some part of the functionality. In general, all the components of the structure are flexible, movable and testable in themselves.
5. Documentation. The Flask framework has a large number of structured examples and best practices. This creates good conditions for preliminary training, analyzing various aspects and capabilities of the tool.

**Database.**

For the work and processing of user data, the author chose the MySQL Database Management System.

MySQL is a database management system (DBMS) that allows you to store, organize, and manipulate large amounts of data. MySQL is developed and maintained by Oracle and is available in two versions: commercial (MySQL Enterprise Edition) and free (MySQL Community Edition). The most common is the community version.

MySQL provides powerful data processing tools. This software uses the standard SQL language for data processing, and the corresponding product allows the user to create tables, run queries, insert, update, and delete data without knowing the SQL query language. MySQL combines good performance, high reliability, and simplicity, making it a popular choice among website builders and administrators who require data processing. (ORACLE 2022).

In the context of developing a website with a built-in chatbot based on artificial intelligence from OpenAI, the choice of MySQL DBMS is due to a number of critical characteristics that ensure the stability and efficiency of the platform. The use of MySQL in this project allows you to solve the following tasks.

1. Data storage: MySQL provides reliable mechanisms for structured storage of large amounts of data, which is critical for the functioning of a chatbot that processes user requests and stores the history of their interactions. This creates the basis for analyzing and improving the quality of service.
2. Data Management: MySQL has advanced capabilities for creating, modifying, and managing tables and indexes, allowing you to efficiently organize and optimize data processing. This provides quick access to the information you need and improves overall system performance.
3. Security. MySQL allows you to enforce password policies by requiring non-privileged users to set a new password when they enter their current password. This feature can protect the database from several threats, such as attackers who compromise the host computer and try to access user database sessions through the web shell. If Change Password Policy is enabled, attackers cannot lock the target user out of the compromised account unless the user has an existing password. You can also use TLS/SSL encryption to protect MySQL connections between networks. Communication on a secure network may not require encryption. MySQL also allows data encryption at rest to protect stored data in the event of a server crash.
4. Open-source code and community: Thanks to its openness and wide developer community, MySQL provides access to many resources and tools, which helps to speed up development and simplify website maintenance.

Thus, the use of MySQL in this project allows you to create a powerful, reliable, and scalable database capable of supporting the complex and dynamic functions of a website whose centrepiece is an AI-based chatbot. This choice is driven not only by the technical characteristics of the system, but also by the needs of the project, confirming its feasibility in terms of strategic and operational data management.

AI Integration.

To create an AI chatbot, the author will use the ChatGPT API platform.

ChatGPT is a natural language processing chatbot powered by generative AI that allows you to have human-like conversations to perform various tasks. For example, an AI tool can answer questions and help you with tasks such as composing emails, coding, and pitching programs.

The OpenAI API allows users to access large OpenAI language models and harness the power of generative artificial intelligence. The OpenAI API helps users create more reliable and controlled outputs from LLMs. This can be achieved by efficiently designing input hints and using LLM hyperparameters to regulate the deterministic behavior of the output.

The OpenAI API gives users access to a variety of large language models, including GPT-4, GPT-3.5, and text-davinci-003. Newer models such as GPT-3.5 and GPT-4 can be accessed through the chat/completion endpoint, and older models such as text-davinci-003 can be accessed through the completion endpoint. One of the important factors in measuring the performance of artificial intelligence models is the number of parameters they have. Parameters refer to the configured values used by neural networks in language models. These parameters play a crucial role in allowing models to process given inputs and generate the desired output. (geeksforgeeks 2023).

Below are the main reasons why this particular API was chosen.

1. Natural Language Processing (NLP). ChatGPT has advanced comprehension and text generation capabilities, which are crucial for creating a chatbot that is capable of having a natural and meaningful conversation with the user. This improves the usability of the website and provides a high quality of service.
2. Manage multiple conversations. The API supports conversation context management, which allows the chatbot to manage the conversation while maintaining consistency and consistency in responses, even to complex user requests. The illusion of natural communication with a live interlocutor is created.
3. Customization and Customization: ChatGPT provides customization and financial customization options that allow you to tailor the model to the specific needs and challenges of a particular industry or business. This gives you the flexibility to use APIs to meet the unique requirements of your project.
4. Scalability. The ChatGPT API is capable of handling high volumes of requests without sacrificing performance, which is critical for business applications with high traffic and responsiveness requirements.
5. Security and privacy. OpenAI ensures a high level of security for the data used and generated by the API, ensuring that users' personal information is protected and that data privacy regulations are respected.
6. Easy integration. Integration with Python and use of libraries provided by OpenAI make it easy to integrate ChatGPT functionality into existing or new web platforms. This reduces development time and simplifies technical support of the project.

Choosing the ChatGPT API to integrate a chatbot into a website is a decision based on the project’s needs to create effective, secure and scalable means of communication with the user.

This choice is supported by both the technical characteristics of the API and strategic considerations of improving the user experience and increasing the competitiveness of the project.

## 2.3. Website development

**System architecture.**

The front-end part of the site is implemented using the Vue.js framework, chosen for its flexibility and powerful capabilities for creating interactive user interfaces. Vue.js supports the dynamic creation of single-page applications (SPAs), ideal for modern web services that require high interactivity without reloading pages. The Vuex state management system is used to provide a reactive display of data, allowing users to see real-time updates without triggering page refreshes. Vue-Router is integrated to manage navigation in the SPA, supporting deep browser history integration and SEO optimization without sacrificing performance. Below is the wireframe version of the website, i.e. its main page and the login and registration page in Figure 6.

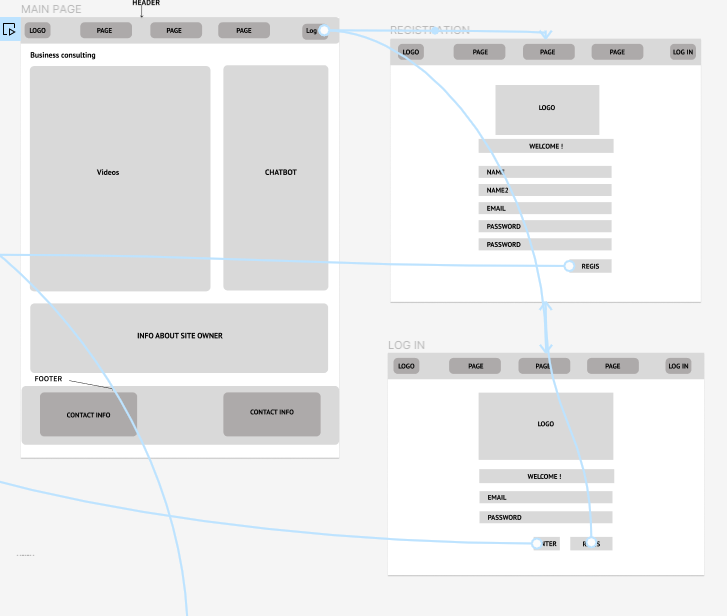


Figure 6. Wireframe version of the website.

For the backend, we chose Flask, a high-level Python framework that facilitates the rapid development of reliable and efficient web applications. An important advantage of Flask is its built-in support for administration, user authentication, and security tasks, which are required for commercial applications.

OpenAI's ChatGPT API integration is done through an API gateway developed by Flask that handles requests to the chatbot. This solution allows you to isolate the chatbot logic from the main business logic of the site, providing flexibility in managing changes and scaling individual components of the system. The ChatGPT API is used to extend the site's capabilities by providing text generation capabilities that are optimized for user queries, which significantly increases the level of engagement. The architecture of the page with the chatbot is under figure 7.

Изображение выглядит как снимок экрана, текст, дизайн

Автоматически созданное описание

Figure 7. Wireframe version of website pages.

**Implementation of functionality**

The development process begins with the prototyping of the interface, followed by the coding of the components. Using Vue.js, developers can create scalable and easily maintainable components that are important for the long-term stability of the project. Backend development aims to create a RESTful API that interacts with the frontend, processes requests, and returns data in JSON format.

Given the potential growth of the audience and the load on the system, special attention is paid to an architecture capable of horizontal scaling. With cloud services and containerization with Docker, you can easily scale the number of running application instances based on your current resource requirements.

**Problems and solutions.** When developing an interactive Web application, there is often a problem with synchronizing the state of the various components. Given the dynamic nature of data and the need to keep it up to date in real time, the challenge of managing state has become critical. Vuex was chosen as the solution to this problem because of its ability to centrally manage all application states in the Vue.js ecosystem. This made it possible.

1. Provide a single source of truth for application health information, which is critical for features that require high data consistency, such as a chatbot that needs to monitor the context of a user's conversation.
2. Simplify debugging because the state of the application can be monitored and modified from a single location, greatly reducing the time it takes to find and fix errors in the code.

When a web application deals with user data, its security becomes important. And for this reason, a comprehensive security strategy was developed and implemented.

1. All data stored in the database and transmitted between the client and server is encrypted using modern cryptographic algorithms using HTTPS.
2. Regular security checks and code reviews are conducted to help identify and eliminate potential application security vulnerabilities.

# 3.RESULTS

## Frontend website development

The frontend of the website was developed using Vue.js. The author used its architecture and reactivity, which is component-driven to design a dynamic and responsive user interface. A website combines various functions and services of the visual part of a website, such as:

1. User Document Management
2. User Authentication
3. User Registration
4. Real-Time Chat with Artificial Intelligence
5. Sales data with the form of a graph
6. A main page with the ability to watch videos about the development of the business environment and entrepreneurship in general.

The author used each individual component to perform certain functions and services, which gave the site modularity and ease of maintenance. Also, each individual component implements @media queries for website adaptability on different devices and screen sizes.

Additional technologies that were used during the development:

1. Vuex – Has been used as a technology to monitor the state of the application, which has implications for managing user authorization state and data between components. (See Annex 1).
2. Vue Router – has been used to navigate between pages in a website, allowing you to work from a single page without additional reloading. (See Annex 1).
3. Axios was used to make HTTP requests from the browser to the server written in python. It also supports uploading and sending files, setting request headers. (See Annex 1).
4. Chart.js – Used to visualize user data, it provides interactive graphs for analytics. (See Annex 1).

**Authentication and User Experience**

AuthService is a function for authenticating users on a website. The service makes asynchronous requests to an API connected to the server to authenticate and retrieve user data. After the response from the server, the authorization service processes the results, stores the user's access token, and sends the user information back for further use.

The main functionality of the service is the login(credentials) and loadUserInfo() methods.

1. 1. The login(credentials) method is responsible for authorizing the user into the system, checking for the presence of specific data in the database. If the user is successfully authenticated, this method stores the resulting token in the browser's local database. Also, if authentication is negative, the service logs an error and sends it as a console error.
2. 2. The loadUserInfo() method is responsible for loading information about the current user from the server API. This method also processes successful server responses, and if authentication fails, logs errors and sends them to the server.

The UserService service is a functionality for working with user data. The service makes requests to the API to get a list of users or add a new user. The results of operations are processed asynchronously, and any errors that occur when executing requests are logged and passed to application components for further processing.

The main functionality of the service is the getUsers(), addUser(userData) methods.

1. 1. The getUsers() method receives data about authorized users in the form of a list from the API and returns the result in the form of successful authorization.
2. 2. The addUser(userData) method adds a new user to the database and returns data in the form of a successful addition message.

**Header and Footer**

To ensure the same appearance and functionality of the header and footer on all pages of the website, the author decided to import the components of these parts into the main App.vue file, which allows them to be used on all websites without the need to re-import. Libraries such as Bootstrap or Tailwind CSS were also used for styling and cross-browser compatibility.

The header component implements navigation links, as well as the website logo. It is divided into a right part with links to various pages of the site, as well as a left part with a logo, also depending on the user's authorization status, a link to log in or log out of the user's account.

The footer component is implemented by adding various data about the author: his contact details, links to social networks and links to the support service in the form of the author of the site.

**Main page of the web site**, on this page there are three blocks of content, on the right side general educational videos about the implementation of business systems and settings in SMEs are displayed, and on the right side there is a mini-interactive chat bot with which the user can communicate, and under They contain brief information about the author of the site. When a user is authorized, the block with information about the owner is removed and in its place a block with business statistics is added, by clicking on which you can go to the business statistics page. All pages of the website adapt to the size of the user’s screen using media queries, which ensures high-quality and correct display on different devices.

The block with a chat bot has a standard set of functions such as typing, a send button, the ability to scroll a message, as well as a button to close the chatbot. The logic of the chatbot will be described in the sections, Web page for communication with chat bot.

The block with educational video materials works with the help of styles for correct display on users' screens, and the videos themselves are taken from the YouTube application, which provides the ability to remove videos with its HTML code. And with the help of the iframe command and embedding HTML code in it, the website loads and calls up video material from the YouTube platform itself, and this command also allows you to watch the video on the website page itself. The videos themselves are under absolute positioning, which allows them to be correctly integrated into the block design. The visual part of the site can be viewed in Figure 8 below.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Figure 8. Main Page (Source: author's summary)

**Web page for registration and authorization**

The registration page is responsible for the registration of new users through a special component on the website.

The main functionality of the component is the registerUser(), UserService.addUser() methods.

1. The registerUser() method controls and processes the correctness of the data entered in the registration form, verifies the identity of the password in the re-entry field, and sends a request for a new user to register via UserService.addUser() to the server. In case of unsuccessful registration, a registration error message is displayed to the user through the browser notification system.

The login page, responsible for authenticating existing users and managing sessions.

The main functionality of the component is the login(), AuthService.login() methods.

1. The method, login(), controls and processes the user's login form data, sends an authentication request via AuthService.login(), and manages the user's tokens and sessions. It is also used by Vuex to manage the state of the website. Also, in case of unsuccessful authorization, an authorization error message is displayed to the user through the browser notification system.

After registration, the user gets the opportunity to save his documents, the ability to work with statistics, as well as the ability to analyse personal documents using a chat bot.

The security of user data is maintained by storing tokens that are stored in localStorage, which allows user sessions to be maintained between browser sessions, as well as the use of https and client-side and server-side validation for input data. The visual part of the registration page can be viewed below in the figure 9.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Figure 9. Registration Page (Source: author's summary)

**Web page for working with documents,** is a page on which an authorized user can download and upload documents. Component operation logic. When the handleFileUpload(event) state changes, the file is saved in the local state, after submitFile() the file is added to formdata and sent to the server, and the list is updated. After this, the page uses fetchDocuments() to update the current list of documents based on information from the server, then the user can use downloadFile(documentId), which opens window.open to download the document by ID according to the information from the server, or the user can use deleteFile(documentId), which deleting a document by id. The visual part of this page with users’ documents can be seen in Figure 10.

The main functionality of the component – methods submitFile(), fetchDocuments(), downloadFile(documentId), handleFileUpload(event) methods.

1. handleFileUpload(event) is needed to recognize the user’s choice of document.
2. submitFile() is responsible for sending the document to the server side, using methods such as axios and formdata.
3. fetchDocuments() is responsible for querying the authorized user's list of documents from the server.
4. downloadFile(documentId), this method is needed to download a document by its id.
5. deleteFile(documentId), this method is used to delete documents by id from the database.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Figure 10. Document Page (Source: author's summary)

**Web page for communication with chat bot,** on this page the user can interact with the chatbot in real time by sending a message as text and send a document for analysis from the chatbot.

Implementation of the component, when a user sends a message, the message data is intercepted from the input field and sent to the server using ApiService.sendMessageToBot(), after which the response from the server is added to the list of the chatbot message, and the interaction interface automatically scrolls to the new message. The visual part of the page with the chat bot can be viewed below in figure 11.

The main functionality of the component is the methods sendMessage(), sendFile(), sendMessageOrFile(), ScrollToEnd(), toggleSidebar(), handleFileUpload().

1. sendMessage(): This method is designed to process the user request and the bot response.
2. sendFile(): This method is responsible for uploading the document to the server and processing the analysis provided by the bot.
3. sendMessageOrFile(): Determines whether the user is sending a message or a file and calls the appropriate method.
4. ScrollToEnd(): is responsible for automatically scrolling the chat to the last user request after each new message.
5. toggleSidebar(): Allows you to hide and show the sidebar.
6. handleFileUpload(): Handles a file upload by the user and stores it for later upload.

User Experience.

1. The interface is easy to learn and intuitive, with specific divisions of the elements of interacting with the chat bot and sending messages.
2. Interactivity, implemented visual feedback when responding to a chatbot and sending a message with scrolling through the text field, for smooth interaction.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Figure 11. ChatBot Page (Source: author's summary)

**Web page for working with business statistics**, the main part of the page is used to enter data, display this data, and analyze the entered data in the form of a graph. Component logic, the user can enter the amount of his sales in the currency of the euro, as well as the date of filling, in the input field. After that, sales data are displayed in the form of a list, with a date and with the amount of money, after which these values with earnings are summed up and their average value is displayed. And with the help of this data, the line graph is visualized. The user also has a button to delete sales data.

Example of input data.

1. Sales Amount: 150.00 (one numeric value representing the sales amount)
2. Date: 05/15/2024 (one date value indicating the date the sale was recorded)

The main functionality of this component is the validateInput(), addSale(), and Sales().

1. validateInput(), this method validates the data before adding it against criteria such as a positive sales amount and today's date.
2. addSale(), after validating via validateInput(), addSale() processes this data and sends it to the server via ApiService.post().
3. Sales(), based on this array, is calculated using functions such as totalSales and averageSales. Errors during the addition of sales data are processed and displayed in the developer console.
4. deleteSale(saleID) this function deletes sales data from a user interface and database by their id.

The process of building a graph was implemented using Chart.js features, for this a separate component with a line graph from canvas was created, which is imported into the component with statistics of the website. The chart is constantly updated when the sales array changes. The visual part of the statistics page can be viewed below in figure 12.

Изображение выглядит как текст, диаграмма, программное обеспечение, число

Автоматически созданное описание

Figure 12. Statistic Page (Source: author's summary)

The development of the visual part of the website was carried out using the Vue.js framework, which made it possible to create an adaptive and dynamic interface. The ability to work with the component architecture and reactivity of Vue.js made it possible to implement the main functions of the website, such as authorization and authentication, registration, user document management, communication with AI in real time, and the ability to visualize business data through graphs.

## 3.2. Backend website development

With the help of the python programming language, the backend of the website was developed, which interacts with the user through the user interface, and has the ability to process user requests using artificial intelligence from OpenAI. The server provides users with the ability to authorize, register, work with text messages with artificial intelligence, download files, upload files, as well as manage personal data. (See Annex 2).

Technologies used.

For the development of the server, such technologies as.

1. Python 3.8, This is a high-level programming language that was chosen because of its large standard library, which interacts well with OPENAI's API, as well as easy integration with other programming languages.
2. Flask, this lightweight web framework, was chosen because of its easy syntax, user-friendly templates, with the help of which it became possible to create a reliable and simple backend.
3. MYSQL, this kind of database management system (DBMS), was chosen for its speed, free license, and robust security.
4. Bcrypt, a library for encrypting passwords by hashing them, providing secure storage of user passwords.
5. Dotenv, a small package that reads key-value pairs from a file. env, and loads the environment variables required by the application.
6. The OpenAI API is an application programming interface that provides access to artificial intelligence technologies developed and maintained by the OpenAI Research Lab.
7. Redis, this is an open-source server that is used to store the user session

Server-side libraries.

1. Flask is the main framework for working with the server.
2. Flask\_cors is a special extension for the framework that is needed to support Cross-Origin Resource Sharing (CORS), which allows the server's API to securely interact with other sources.
3. Bcrypt – to hash the user's passwords.
4. Mysql.connector – is a driver for connecting to a database on MYSQL.
5. Dotenv – Needed to work with the .env file.
6. Openai – to work with the OPENAI API.
7. Os – Library for working with the functionality of the operating system.

The Flask and CORS configuration was configured, which made it possible to process requests from a standard local Vue.js server <http://localhost:8000>.

Also, to check the activation of the server, the function @app.route('/') was written, which shows a welcome message in the local server <http://127.0.0.1:5000>.

**Website database.**

When the server works with the front-end of the site, three tables are created in the MYSQL database: a user table to store user data, a document table to store documents with files, and a sales table with sales data. Additionally, the Documents and Sales table is linked to the Users table through a foreign key called user\_id. (See Annex 2).

The user table has the following fields.

1. Id, to uniquely identify the user, has the int type as the primary key, this field is needed to interact with other users.
2. firstName, to store the username, is of type varchar (255).
3. lastName, to store the user's last name, is also of type varchar (255).
4. Email, used to store the user's email, has the varchar type (255), is also a unique identifier for logging in to the system.
5. Password\_hash, stores the user's encrypted password, uses the char (60) type.
6. Create\_at, stores the exact time of account creation, in fact, automatically records the time of the string creation, is of the Timestemp type.
7. Last\_login, stores the time the last time the user visits to the system, essentially updates each time the user logs in, also uses the Timesyamp type.
8. Sales\_statistics, stores the user's sales statistics, JSON type, is used to store structured information about the user's sales statistics.

The user’s table is used to register and authorize users, as well as store their personal data and track their activities on the site. The contents of the users table can be seen in figure 13.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Figure 13. Users table (Source: author's summary)

The documents table has the following fields.

1. Id, as a unique identifier for a document, uses the int type.
2. User\_id, is needed as an identifier for the user working with files, type int.
3. Create\_at, the specific time the file is uploaded, automatically updated.
4. File\_data, stores file-specific data, type mediumblob, stores data in binary form.
5. File\_name, file name, works with type varchar (255)

The documents table is used to provide the functionality of downloading, storing, and uploading files. The contents of the documents table can be seen in figure 14.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Figure 14. Documents table (Source: author's summary)

The sales table has the following fields.

1. id, as the user's sales identifier, has type int as the primary key.
2. user\_id, required as a user identifier for working with sales, works with the int type.
3. amount, stores sales data with high precision, can have a decimal number with a maximum total precision of 10 digits. Two of them are after the decimal point and eight are before it. works with type decimal(10,2).
4. date, needed to save the date of sale, works with the date type, which allows you to store data in a standard format (YYYY-MM-DD).

The contents of the documents table can be seen in figure 15.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Figure 15. Sales table (Source: author's summary)

On the server side, functions were implemented to process HTTP requests from the front-end during user authorization and registration. (See Annex 2).

1. @app.route('/login', methods=['POST']) – is needed to process the user's login to the account, processing is carried out by checking credentials from the database, and establishes a session upon successful authorization.
2. @app.route('/user-info', methods=['GET']) – returns information about the user.
3. @app.route('/users', methods=['POST']) – adds a new user to the database. During implementation, there is a call hash\_password function, which is responsible for hashing the user's password.

You can see the health of the database in figure 16.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Figure 16. Users Table Contents (Source: author's summary)

Section with user documents. It has been implemented with three functions, for uploading, downloading, and processing documents respectively. You can see the health of the database in figure 17.

1. @app.route('/upload', methods=['POST']) – works through POST requests. is written so that it would load the file, but first check it against requirements such as extension, size.
2. @app.route('/documents', methods=['GET']) – works through GET requests, the function is needed for the user to return his personal list of documents.
3. @app.route('/download/<int:document\_id>', methods=['GET']) – works through GET requests, designed to download files by identifiers from the database.
4. @app.route('/documents/<int:document\_id>', methods=['DELETE']) – gives the user the opportunity to delete the sale.

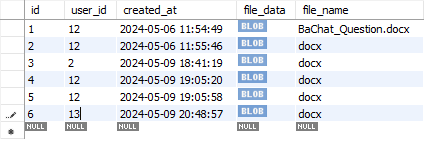


Figure 17. Document Table Contents (Source: author's summary)

Part of the server with user sales data. Implemented using three functions for adding data, obtaining a list of sales and deleting sales if entered incorrectly. You can see the health of the database in figure 18.

1. @app.route('/sales', methods=['POST']) – sends new information about the user's sales to the database, and updates the user's sales statistics.
2. @app.route('/sales', methods=['POST']) – returns the user's personal sales data.
3. @app.route('/sales/<int:sale\_id>', methods=['DELETE']) – gives the user the opportunity to delete the sale.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Figure 18. Sales Table Contents (Source: author's summary)

**Implementation and training by Artificial Intelligence.**

Artificial intelligence is called BaChat, which stands for business assistance chat. BaChat is configured on the GPT - 3.5 Turbo model, it is implemented to solve business problems, as well as advise entrepreneurs on tax laws and advise on building a business and business processes at the enterprise. The model training process was carried out on many documents with tax information from the baltikon website, as well as with documents with information from the business cases of Latvian companies. An AI behaviour model was also written so that it would only answer questions related to business in Latvia.

BaChat instruction.

1. Role - Business Assistant in Latvia.
2. The main tasks are to respond to user queries about business, management, taxation, financial control, and laws related to the business environment. Help with the preparation and creation of business documentation. Answer questions related to taxation from the attached documentation, as well as provide data on laws and articles.
3. Restriction - Provide information and answer only questions related to the business environment in Latvia. Warn users about your restrictions if the question is not about business.
4. The goal is to provide accurate and high-quality information on issues related to business in Latvia.

The process of integrating the Chat Assistant API into the server.

A separate .env file has been created, which contains a variable OPENAI\_API\_KEY with the chatbot's API key. After that, a python file was created with the logic of working with user messages and a chat bot. This file contains the EventHandler class, which implements the logic for collecting user messages and responses from the chatbot. It also contains a handle\_user\_message function that is implemented to handle the user's message through the EventHandler class.

On the server side, the dotenv library was used, with the help of which information from the .env file with the Api key was loaded, as well as a file with the chatbot logic was imported. The api\_handle\_user\_message function was also implemented, which is needed for the server to interact with the API file. This feature works through POST requests and allows users to send messages through a web interface.

The upload\_file() function has also been added, for working with documents from the authorized user. This function is needed to upload a file to the server, save the same file in the database and process the document using the GPT-3.5 Turbo model. If the document does not allow it to be decoded as text, the function returns an error message. The answer of the chatbot to the uploaded document and the question about this document can be seen in figure 19.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Figure 19. Chat Bot Communication (Source: author's summary)

During the development of the project, all the tasks were achieved to provide quality services for entrepreneurs.

1. The author has successfully integrated artificial intelligence into his website, which provides entrepreneurs, i.e. users, with a powerful system for consulting and analyzing their business projects and enterprise.
2. An intuitive and responsive user interface has been created that meets basic user requests, regardless of their level of technical savvy.
3. The server ensures the security and confidentiality of data.
4. The website is easily scalable since it is written using a modular architecture, which will allow you to adapt and add functionality in the future, without much time.

The project shows that the implementation of IT solutions for working with small and medium-sized businesses are effective and affordable tools for business support and development.

# CONCLUSIONS AND RECOMMENDATIONS

**Conclusions**

1. The main goal. The main goal of developing a website by an entrepreneur from small and medium-sized businesses, to help with business management, business data management, as well as assistance from the chat assistant was achieved.
2. Key requirements. The author analysed and identified important aspects of SMEs' work with digitalization and automation in their business. Aspects such as the need for a user-oriented interface to see business problems, working with data analytics and the availability of consulting support.
3. Statistical analysis. A system has been developed in which the user can analyse the statistics of his business by entering sales data. In addition to showing the average value of the business, the website visualizes this data in the form of a graph for the convenience of the user.
4. Database. In my bachelor’s thesis, a MySQL database was successfully created and integrated. A database works to store and manage business data by creating two tables by using a join key value.
5. Chat assistant. The author has successfully integrated and trained a chat assistant based on the ChatGPT API. The chat assistant helps entrepreneurs resolve issues related to taxes, business, financial control, and preparing or analyzing business documents. Training and instructions for the chat assistant were implemented in the OpenAI platform, by attaching documents related to business management and legislation in Latvia for analysis from artificial intelligence.
6. Oriented design. The front-end part of the website was focused on user convenience by using the Vue.js framework. An interactive interface that changes depending on the size of the screen was created, as well as an intuitive navigation system.

**Recommendations**

1. Platform analysis. Regularly review and update the platform based on testing, as well as competitor analysis. The platform analysis process will make the functionality of the site more relevant and improve the user experience.
2. Surveys and research. Regularly conduct surveys and research among potential users to identify new problems to solve in the website. Conducting surveys will help in identifying new directions for improving the website, and will also meet the needs of users.
3. Improve statistical analysis functionality. It is recommended to improve the page with the analysis of sales statistics. By implementing machine learning-based forecasting techniques, such as cluster analysis and time series. The development of these methods will help to analyse as well as predict future trends, as well as identify key indicators of business growth. It is also recommended to implement tools for analyzing seasonal fluctuations.
4. Database. It is recommended to optimize the existing database by indexing frequently used fields, optimizing server queries, and setting up regular backups of data from the database.
5. Chat assistant. Regularly update the chat assistant according to the new OpenAI standards. And constantly train the chat assistant by working with new data and examples of work. It is necessary to regularly update the chatbot database with new legal and tax regulations and current business practices.
6. Integration. It is recommended to use third-party platforms to help entrepreneurs, such as CRM systems, accounting systems, and project management tools. Integration with these third-party systems will improve the overall quality of the website for users.

# BIBLIOGRAPHY

1. Allam, H., Aboul, E., Hassanien, A. (2022) *The Fourth Industrial Revolution: Implementation of Artificial Intelligence for Growing Business Success.* Berlin.
2. Bundlephobia. (2022). *vue@3.4.27.* Electronic resource [date of treatment 05/01/2024]. Available: https://bundlephobia.com/package/vue@3.4.27.
3. ChatGPT. (2024). *Get answers. Find inspiration. Be more productive.* Electronic resource [date of treatment 17/09/2024]. Available: https://openai.com/chatgpt/.
4. Cram. (2024). *Artificial Intelligence, "how to think about the AI impact on the economy".* Electronic resource [date of treatment 17/03/2022]. Available: https://cpram.com/rou/en/individual/publications/megatrends/artificial-intelligence-how-to-think-about-the-ai-impact-on-the-economy.
5. DHL. (2024). *About us.* Electronic resource [date of treatment 04/11/2024]. Available: https://group.dhl.com/en/about-us.html.
6. DoNotPay. (2022). *Your AI Consumer Champion.* Electronic resource [date of treatment 11/09/2024]. Available: https://donotpay.com/.
7. geeksforgeeks. (2023). *What is ChatGPT API?* Electronic resource [date of treatment 27/03/2023]. Available: https://www.geeksforgeeks.org/what-is-chatgpt-api/.
8. Goldberg, H., Kenneth, Y. (2012). *What is Automation ?* Berkeley: ResearchGate.
9. Harrison, M. (2023). *The Microsoft Bing Search Engine and ChatGPT-4.* Electronic resource [date of treatment 23/05/2023]. Available: https://bullseyestrategy.com/microsoft-bing-search-engine-chatgpt/.
10. IBM. (2022). *What is a chatbot?*. Electronic resource [date of treatment 10/12/2024]. Available: https://www.ibm.com/topics/chatbots.
11. IFAC. (2019). *The Foundation for Economies Worldwide Is Small Business.* Electronic resource [date of treatment 26/06/2019]. Available: https://ifac.org/knowledge-gateway/contributing-global-economy/discussion/foundation-economies-worldwide-small-business-0.
12. InSales. (2022). *Mark Sherrington's 5W technique: features of the segmentation model.* Electronic resource [date of treatment 09/12/2022]. Available: https://www.insales.ru/blogs/university/metodika-5w-marka-sherringtona.
13. INSTAGANTT. (2024). *What is Bard (Google AI)? Everything you need to know.* Electronic resource [date of treatment 09/02/2024]. Available: https://www.instagantt.com/project-management/what-is-bard-google-ai.
14. KUMAR, N. (2022). *How GPT-3 is Revolutionizing AI Language Models.* Electronic resource [date of treatment 23/02/2023]. Available: https://www.linkedin.com/pulse/how-gpt-3-revolutionizing-ai-language-models-nitin-kumar-sharma/.
15. McGee, J. (2024). *Impact Of Chatbots On Business Productivity And Efficiency.* Electronic resource [date of treatment 18/01/2024]. Available:. https://ai-techpark.com/impact-of-chatbots-on-business-productivity-and-efficiency/.
16. Meffert, V., Kulagin, A., Sukharevski, Y. (2019). *Digital @scale Настольная книга по цифровизации бизнеса.* Moscow: альпина паблишер.
17. Moiseev, Andrey. (2023). *Цифровизация vs. автоматизация: в чем разница и как они связаны?.* Electronic resource [date of treatment 07/12/2023]. Available: https://www.linkedin.com/pulse/%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F-vs-%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F-%D0%B2-%D1%87%D0%B5%D0%BC-%D1%80%D0%B0%D0%B7%D0%BD%D0%B8%D1%86%D0%B0-.
18. Natalya, Z., Alexander, L. (2017). *Small & Medium Entrepreneurship in European Countries: Main Tendencies of the Development.* Sankt Petersburg: Administrative Consulting.
19. Olmstead, Levi. (2022) *11 Critical Digital Transformation Challenges to Overcome.* Electronic resource [date of treatment 02/12/2022]. Available: https://whatfix.com/blog/digital-transformation-challenges/.
20. OPEN AI. (2023). *GPT-4.* Electronic resource [date of treatment 14/03/2022]. Available: https://openai.com/index/gpt-4-research/.
21. ORACLE. (2024). *What is MySQL?* Electronic resource [date of treatment 13/01/2022]. Available: https://www.oracle.com/mysql/what-is-mysql/.
22. PepperVA. (2023) *Our Story.* Electronic resource [date of treatment 23/11/2024]. Available: https://peppervirtualassistant.com/hire-a-virtual-assistant/.
23. Python Tutorial. (2024). *What is Flask Python.* Electronic resource [date of treatment 04/11/2022]. Available: https://pythonbasics.org/what-is-flask-python/.
24. Radicic, D. (2023). *Impact of digitalization on technological innovations in small and medium-sized enterprises (SMEs).* Electronic resource [date of treatment 29/03/2023]. Available: https://www.sciencedirect.com/science/article/pii/S0040162523001592.
25. Ribeiro, S., Esther, S., (2023). *The role of digitalization in business and management:.* Berlin:Springer.
26. Saini, K. (2023). *Enhance Customer Support and Drive Operational Efficiency With AI Chatbots.* Electronic resource [date of treatment 31/07/2023]. Available: https://www.grazitti.com/blog/enhance-customer-support-and-drive-operational-efficiency-with-ai-chatbots/.
27. stackoverflow. (*2020). Developer survey.* Electronic resource [date of treatment 09/12/2020]. Available: https://survey.stackoverflow.co/2020#technology-most-loved-dreaded-and-wanted-web-frameworks-loved2.
28. subscribed.fyi. (2024). *Starbucks Chatbot Integration: Enhancing Customer Experience.* Electronic resource [date of treatment 24/04/2024]. Available: https://subscribed.fyi/blog/starbucks-chatbot-integration-enhancing-customer-experience/.
29. VueJS. (2024). *Getting Started.* Electronic resource [date of treatment 20/01/2024]. Available: https://012.vuejs.org/guide/.
30. What'sthebigdata. (2023) *Chatsonic AI.* Electronic resource [date of treatment 16/08/2024]. Available: https://whatsthebigdata.com/tool/chatsonic-ai/.
31. WOLT. (2024). *ABOUT US.* Electronic resource [date of treatment 10/12/2023]. Available:https://explore.wolt.com/en/lva/about?\_gl=1\*1tspbgk\*\_ga\*MTk3MTMwNTczOC4xNzE1NTk5MzIy\*\_ga\_CP7Z2F7NFM\*MTcxNTU5OTMyMi4xLjEuMTcxNTU5OTM0OC4zOC4wLjA.\*\_gcl\_au\*MTE3NjA3NDk0OS4xNzE1NTk5MzI2.

# ANNEXES

Annex 1.

**Front end of website**

**Vuex**

*//stoore.js*

import { createStore } from 'vuex';

export default createStore({

  state: {

    isLoggedIn: false

  },

  mutations: {

    setLoggedIn(*state*, *status*) {

      state.isLoggedIn = status;

    }

  },

  actions: {

    login({ *commit* }, *token*) {

      localStorage.setItem('authToken', token);

      commit('setLoggedIn', true);

    },

    logout({ *commit* }) {

      localStorage.removeItem('authToken');

      commit('setLoggedIn', false);

    }

  }

});

**Vue Router**

*//router.js*

import Router from 'vue-router';

Vue.use(Router);

*const* router = new Router({

    mode: 'history',

    routes: [

        {

            path: '/',

            name: 'home',

            component: Home,

        },

        {

            path: '/enter',

            name: 'enter',

            component: EnterPage,

            meta: { hideFooter: true },

        },

        {

            path: '/registr',

            name: 'registr',

            component: RegisterPage,

            meta: { hideFooter: true },

        },

    ],

});

export default router;

**Axios**

import axios from 'axios';

*const* apiClient = axios.create({

  baseURL: 'http://127.0.0.1:5000',

  withCredentials: true,

  headers: {

    'Content-Type': 'application/json',

    'Accept': 'application/json',

  }

});

*const* ApiService = {

  init() {},

  post(*endpoint*, *body*) {

    return apiClient.post(endpoint, body);

  },

  get(*endpoint*) {

    return apiClient.get(endpoint);

  },

  login(*credentials*) {

    return apiClient.post('/login', credentials);

  },

  loadUserInfo() {

    return apiClient.get('/user-info');

  },

  sendMessageToBot(*message*) {

    return apiClient.post('/api/send-message', { message });

  },

  getChatHistory() {

    return apiClient.get('/api/chat-history');

  },

  uploadDocument(*file*) {

*let* formData = new FormData();

    formData.append('file', file);

    return apiClient.post('/api/upload-document', formData, {

      headers: {

        'Content-Type': 'multipart/form-data'

      }

    });

  },

  fetchDocuments() {

    return apiClient.get('/documents');

  },

  downloadDocument(*documentId*) {

    return apiClient.get(**`**/download/${documentId}**`**, {

      responseType: 'blob'

    });

  },

  deleteDocument(*documentId*) {

    return apiClient.delete(**`**/documents/${documentId}**`**);

  },

  addSale(*sale*) {

    return apiClient.post('/sales', sale);

  },

  fetchSales() {

    return apiClient.get('/sales');

  },

  deleteSale(*saleId*) {

    return apiClient.delete(**`**/sales/${saleId}**`**);

  }

};

export default ApiService;

**Chart.js**

export default {

  name: 'SalesChart',

  props: {

    sales: Array

  },

  data() {

    return {

      chart: null

    };

  },

  mounted() {

*this*.createChart();

  },

  methods: {

    createChart() {

*const* ctx = *this*.$refs.salesChart.getContext('2d');

*this*.chart = new Chart(ctx, {

        type: 'line',

        data: {

          labels: *this*.sales.map(*sale* *=>* sale.date),

          datasets: [{

            label: 'Sales in EUR',

            data: *this*.sales.map(*sale* *=>* sale.amount),

            backgroundColor: 'rgba(255, 99, 132, 0.2)',

            borderColor: 'rgba(255, 99, 132, 1)',

            borderWidth: 1

          }]

        },

        options: {

          scales: {

            y: {

              beginAtZero: true

            }

          }

        }

      });

    }

  },

  watch: {

    sales: {

      deep: true,

      handler() {

*this*.chart.destroy();

*this*.createChart();

      }

    }

  }

};

</script>

**Enter Page**

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Annex 2.

**Back end of website**

**Server (app.py)**

from flask import Flask, request, jsonify, Response, session, make\_response, redirect

from flask\_session import Session

from flask\_cors import CORS

import redis

import bcrypt

import mysql.connector

import os

from dotenv import load\_dotenv

import openai

from werkzeug.utils import secure\_filename

import docx

app = Flask(\_\_name\_\_)

CORS(app, *supports\_credentials*=True, *origins*=["http://localhost:8080"])

app.config['SECRET\_KEY'] = '31423142314231423142'

app.config['SESSION\_TYPE'] = 'redis'

app.config['SESSION\_PERMANENT'] = False

app.config['SESSION\_USE\_SIGNER'] = True

app.config['SESSION\_KEY\_PREFIX'] = 'session:'

app.config['SESSION\_REDIS'] = redis.StrictRedis(*host*='localhost', *port*=6379, *db*=0)

app.config['SESSION\_COOKIE\_NAME'] = 'your\_cookie\_name'

app.config['SESSION\_COOKIE\_HTTPONLY'] = True

app.config['SESSION\_COOKIE\_SECURE'] = True

app.config['SESSION\_COOKIE\_SAMESITE'] = 'None'

Session(app)

dotenv\_path = 'doc\_2024-04-30\_19-13-36.env'

load\_dotenv(dotenv\_path)

api\_key = os.getenv("OPENAI\_API\_KEY")

openai.api\_key = api\_key

db = mysql.connector.connect(

*host*="127.0.0.1",

*user*="root",

*password*="3142758609",

*database*="bachat\_users"

)

@app.after\_request

*def* after\_request(*response*):

    response.headers['Access-Control-Allow-Headers'] = 'Content-Type,Authorization'

    response.headers['Access-Control-Allow-Methods'] = 'GET,PUT,POST,DELETE,OPTIONS'

    response.headers['Access-Control-Allow-Origin'] = 'http://localhost:8080'

    response.headers['Access-Control-Allow-Credentials'] = 'true'

    return response

UPLOAD\_FOLDER = 'uploads'

ALLOWED\_EXTENSIONS = {'txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif', 'doc', 'docx'}

MAX\_CONTENT\_LENGTH = 16 \* 1024 \* 1024  *# 16 MB*

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

app.config['MAX\_CONTENT\_LENGTH'] = MAX\_CONTENT\_LENGTH

if not os.path.exists(UPLOAD\_FOLDER):

    os.makedirs(UPLOAD\_FOLDER)

*def* allowed\_file(*filename*):

    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

*def* read\_docx(*file\_path*):

    doc = docx.Document(file\_path)

    full\_text = []

    for para in doc.paragraphs:

        full\_text.append(para.text)

    return '\n'.join(full\_text)

*# ChatBot*

@app.route('/api/send-message', *methods*=['POST'])

*def* api\_handle\_user\_message():

    data = request.json

    message = data.get('message')

    if message:

        try:

            response = openai.ChatCompletion.create(

*model*="gpt-3.5-turbo",

*messages*=[{"role": "user", "content": message}]

            )

            responses = response.choices[0].message['content'].strip()

            return jsonify({"response": responses}), 200

        except *Exception* as e:

            app.logger.error(*f*"Error processing the message: {e}")

            return jsonify({"error": "Failed to process your message"}), 500

    else:

        return jsonify({"error": "No message provided"}), 400

@app.route('/api/upload-document', *methods*=['POST'])

*def* upload\_file():

    if 'file' not in request.files:

        return jsonify({"error": "No file part"}), 400

    file = request.files['file']

    if file.filename == '':

        return jsonify({"error": "No selected file"}), 400

    if file and allowed\_file(file.filename):

        filename = secure\_filename(file.filename)

        file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

        file.save(file\_path)

        if os.path.getsize(file\_path) > MAX\_CONTENT\_LENGTH:

            os.remove(file\_path)

            return jsonify({"error": "File too large"}), 413

        try:

            cursor = db.cursor()

            with open(file\_path, 'rb') as f:

                file\_data = f.read()

            cursor.execute("INSERT INTO documents (user\_id, file\_data, file\_name) VALUES (%s, %s, %s)",

                           (session.get('user\_id'), file\_data, filename))

            db.commit()

            cursor.close()

            try:

                if filename.endswith('.docx'):

                    file\_text = read\_docx(file\_path)

                else:

                    file\_text = file\_data.decode('utf-8')

                response = openai.ChatCompletion.create(

*model*="gpt-3.5-turbo",

*messages*=[{"role": "system", "content": "Analyze the following document:"}, {"role": "user", "content": file\_text}]

                )

                analysis = response.choices[0].message['content'].strip()

            except (*UnicodeDecodeError*, docx.opc.exceptions.PackageNotFoundError):

                analysis = "The file could not be decoded as text."

            os.remove(file\_path)  *# Удаляем файл после обработки*

            return jsonify({ "analysis": analysis}), 201

        except *Exception* as e:

            return jsonify({"error": *str*(e)}), 500

    else:

        return jsonify({"error": "File type is not allowed"}), 400

*# Flask*

@app.route('/')

*def* index():

    return "Welcome to the Flask API!", 200

*def* hash\_password(*password*):

    salt = bcrypt.gensalt()

    return bcrypt.hashpw(password.encode('utf-8'), salt)

*def* verify\_password(*password*, *hashed*):

    if isinstance(hashed, *str*):

        hashed = hashed.encode('utf-8')

    return bcrypt.checkpw(password.encode('utf-8'), hashed)

*# Registration and Login*

@app.route('/login', *methods*=['POST'])

*def* login():

    user = request.json

    cursor = db.cursor(*dictionary*=True)

    cursor.execute("SELECT id, password\_hash FROM users WHERE email = %s", (user['email'],))

    result = cursor.fetchone()

    cursor.close()

    if result:

        if bcrypt.checkpw(user['password'].encode('utf-8'), result['password\_hash'].encode('utf-8')):

            session['user\_id'] = result['id']

            app.logger.info(*f*"User {result['id']} logged in successfully. Session: {session['user\_id']}")

            response = make\_response(jsonify({"message": "Logged in successfully"}))

            response.set\_cookie(

                app.config['SESSION\_COOKIE\_NAME'],

                session.sid,

*httponly*=True,

*secure*=app.config['SESSION\_COOKIE\_SECURE'],

*samesite*='None'

            )

            return response, 200

        else:

            app.logger.warning(*f*"Invalid credentials for email: {user['email']}")

            return jsonify({"error": "Invalid credentials"}), 401

    else:

        app.logger.warning(*f*"User not found for email: {user['email']}")

        return jsonify({"error": "User not found"}), 404

@app.route('/logout', *methods*=['POST'])

*def* logout():

    user\_id = session.get('user\_id')

    if user\_id:

        try:

            cursor = db.cursor()

            cursor.execute("DELETE FROM documents WHERE user\_id = %s", (user\_id,))

            db.commit()

            cursor.close()

        except mysql.connector.Error as error:

            app.logger.error(*f*"Error deleting user's documents: {error}")

        session.pop('user\_id', None)

    return jsonify({"message": "Logged out successfully"}), 200

@app.route('/user-info', *methods*=['GET'])

*def* user\_info():

    app.logger.info(*f*"Session data: {session.items()}")

    if 'user\_id' not in session:

        app.logger.info("User not authenticated")

        return jsonify({"error": "User not authenticated"}), 401

    user\_id = session['user\_id']

    app.logger.info(*f*"Authenticated user: {user\_id}")

    return jsonify({"user\_id": user\_id, "info": "Some user info"})

@app.route('/users', *methods*=['POST'])

*def* add\_user():

    data = request.get\_json()

    if not data:

        return jsonify({"error": "No data provided"}), 400

    try:

        cursor = db.cursor()

        password\_hash = hash\_password(data['password'])

        cursor.execute(

            "INSERT INTO users (firstName, lastName, email, password\_hash, sales\_statistics) VALUES (%s, %s, %s, %s, %s)",

            (data.get('firstName'), data.get('lastName'), data.get('email'), password\_hash, '[]')

        )

        db.commit()

        cursor.close()

        return redirect('/')

    except mysql.connector.Error as error:

        app.logger.error(*f*"Error when trying to register user: {error}")

        return jsonify({"error": "Registration failed"}), 500

*# Document*

@app.route('/documents', *methods*=['GET'])

*def* get\_documents():

    user\_id = session.get('user\_id')

    if not user\_id:

        return jsonify({"error": "User not authenticated"}), 401

    cursor = db.cursor(*dictionary*=True)

    cursor.execute("SELECT id, file\_name, created\_at FROM documents WHERE user\_id = %s", (user\_id,))

    documents = cursor.fetchall()

    cursor.close()

    return jsonify(documents)

@app.route('/download/<int:document\_id>', *methods*=['GET'])

*def* download\_file(*document\_id*):

    user\_id = session.get('user\_id')

    if not user\_id:

        return jsonify({"error": "User not authenticated"}), 401

    cursor = db.cursor(*dictionary*=True)

    cursor.execute("SELECT file\_data, file\_name, user\_id FROM documents WHERE id = %s", (document\_id,))

    doc = cursor.fetchone()

    cursor.close()

    if doc and doc['user\_id'] == user\_id:

        file\_data = doc['file\_data']

        file\_name = doc['file\_name']

        return Response(file\_data, *mimetype*="application/octet-stream", *headers*={

            "Content-Disposition": *f*"attachment;filename={file\_name}"

        })

    else:

        return jsonify({"error": "Document not found or access denied"}), 404

@app.route('/documents/<int:document\_id>', *methods*=['DELETE'])

*def* delete\_document(*document\_id*):

    user\_id = session.get('user\_id')

    if not user\_id:

        return jsonify({"error": "User not authenticated"}), 401

    cursor = db.cursor()

    cursor.execute("DELETE FROM documents WHERE id = %s AND user\_id = %s", (document\_id, user\_id))

    db.commit()

    cursor.close()

    if cursor.rowcount == 0:

        return jsonify({"error": "Document not found or access denied"}), 404

    return jsonify({"message": "Document deleted successfully"}), 200

*# Sales - Statistic*

@app.route('/sales', *methods*=['POST'])

*def* add\_sale():

    if 'user\_id' not in session:

        return jsonify({"error": "User not authenticated"}), 401

    data = request.get\_json()

    if not data or 'amount' not in data or 'date' not in data:

        return jsonify({"error": "Invalid data"}), 400

    try:

        cursor = db.cursor()

        cursor.execute("INSERT INTO sales (user\_id, amount, date) VALUES (%s, %s, %s)",

                       (session['user\_id'], data['amount'], data['date']))

        db.commit()

        sale\_id = cursor.lastrowid

        cursor.close()

        return jsonify({"id": sale\_id, "amount": data['amount'], "date": data['date']}), 201

    except mysql.connector.Error as error:

        return jsonify({"error": *str*(error)}), 500

@app.route('/sales', *methods*=['GET'])

*def* get\_sales():

    if 'user\_id' not in session:

        return jsonify({"error": "User not authenticated"}), 401

    start\_date = request.args.get('startDate')

    end\_date = request.args.get('endDate')

    query = "SELECT id, amount, date FROM sales WHERE user\_id = %s"

    params = [session['user\_id']]

    if start\_date and end\_date:

        query += " AND date BETWEEN %s AND %s"

        params.extend([start\_date, end\_date])

    cursor = db.cursor(*dictionary*=True)

    cursor.execute(query, params)

    sales = cursor.fetchall()

    cursor.close()

    return jsonify(sales), 200

@app.route('/sales/<int:sale\_id>', *methods*=['DELETE'])

*def* delete\_sale(*sale\_id*):

    if 'user\_id' not in session:

        return jsonify({"error": "User not authenticated"}), 401

    cursor = db.cursor()

    cursor.execute("DELETE FROM sales WHERE id = %s AND user\_id = %s", (sale\_id, session['user\_id']))

    db.commit()

    cursor.close()

    if cursor.rowcount == 0:

        return jsonify({"error": "Sale not found or access denied"}), 404

    return jsonify({"message": "Sale deleted successfully"}), 200

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(*debug*=True, *host*='127.0.0.1', *port*=5000)

**chatbot**

*# bachat2.py*

from openai import OpenAI, AssistantEventHandler

import requests

*class* EventHandler(*AssistantEventHandler*):

*def* \_\_init\_\_(*self*):

*super*().\_\_init\_\_()

*self*.responses = []

*self*.current\_message = ""

*def* on\_text\_delta(*self*, *delta*, *snapshot*):

        text\_value = delta.value

        print(*f*"Received delta: {text\_value}")

*self*.current\_message += text\_value

        if text\_value.endswith(('.', '?', '!')):

*self*.responses.append(*self*.current\_message.strip())

*self*.current\_message = ""

*def* on\_tool\_call\_created(*self*, *tool\_call*):

        print(*f*"Tool call created: {tool\_call.type}")

*def* on\_tool\_call\_delta(*self*, *delta*, *snapshot*):

        print("Received tool call delta")

*def* handle\_user\_message(*client*, *thread\_id*, *assistant\_id*):

    event\_handler = EventHandler()

    with client.beta.threads.runs.stream(

*thread\_id*=thread\_id,

*assistant\_id*=assistant\_id,

*event\_handler*=event\_handler,

    ) as stream:

        stream.until\_done()

    responses = event\_handler.responses

    return responses

Annex 3.

**Relational Database Diagram**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**SQL command to create tables**

CREATE TABLE users (

id INT PRIMARY KEY,

firstName VARCHAR(255),

lastName VARCHAR(255),

email VARCHAR(255),

password\_hash CHAR(60),

created\_at TIMESTAMP,

last\_login TIMESTAMP,

sales\_statistics JSON

);

CREATE TABLE documents (

id INT PRIMARY KEY,

user\_id INT,

created\_at TIMESTAMP,

file\_data MEDIUMBLOB,

file\_name VARCHAR(255),

FOREIGN KEY (user\_id) REFERENCES users(id)

);

CREATE TABLE sales (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

amount DECIMAL(10,2),

date DATE

);

**Statement**

I hereby confirm that Bachelor thesis “development of a website with a built-in chat assistant” has been developed independently, it does not allow violation of intellectual property rights of other persons or plagiarism - further expression of the results of another person's creative activity in my own name. The thesis does not use text generated by artificial intelligence tools. The works, definitions and quotations taken from other sources are cited in the thesis. Used research results of other authors and data sources are indicated in the references. The thesis has never been published and is being presented for the first time for defense at the State Final Examination Commission.

As the sole author of the Thesis, I allow the EKA University of Applied Sciences to publish my thesis in full (including the Thesis files enclosed in the complete Work package).

I certify that the texts of the PDF version of the Thesis uploaded to the EKA Moodle system with my electronic signature and the MS Word version for plagiarism checking are identical.

*This Statement is an integral part of the Thesis and the electronic signature of the Thesis containing a time stamp is also the electronic signature of this Statement.*