

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ярославский государственный университет имени П. Г. Демидова»

Кафедра компьютерных сетей

Сдано на кафедру

«_____» _____ 2022 г.

Заведующий кафедрой,
д. ф.-м. н., профессор

_____ С. Д. Глызин

Выпускная квалификационная работа

**Разработка сервиса
централизованного управления паролями**

по направлению
01.03.02 Прикладная математика и информатика

Научный руководитель
к. ф.-м. н., доцент

_____ С. В. Алешин

«_____» _____ 2022 г.

Студент группы ИВТ-41БО

_____ Н. С. Батогов

«_____» _____ 2022 г.

Ярославль, 2022

Реферат

Объем 16 с., 5 гл., 0 рис., 0 табл., 9 источников, 2 прил.

Ключевые слова: **управление паролями, пароль, генерирование пароля, хранение паролей, безопасность компании, кибербезопасность, защита данных.**

Объект исследования - централизованное корпоративное управление паролями.

Цель работы - создание сервиса для удобного, безопасного и централизованного управления паролями в компании.

В результате работы был реализован и протестирован сервис, который позволяет решить данную задачу.

Область применения - небольшая компания, которой нужно удобно организовать пароли для почты, сайтов, серверов, карт и других ресурсов.

Содержание

Введение	4
1. Технологический стек	6
1.1. Серверная часть	6
1.2. Клиентская часть	7
2. Оформление элементов текста	8
2.1. Рисунки и таблицы	8
2.2. Заголовки и приложения	9
2.2.1. Заголовки	9
2.2.2. Приложения	9
2.3. Нумерация страниц	9
3. Формулы	10
4. Рисуем с помощью TikZ	11
5. Псевдокод	12
Заключение	13
Список литературы	14
Приложение А. Исходный код программы на C++	15
Приложение Б. Исходный код программы на Python	16

Введение

В настоящее время разработка веб-приложений является одной из самых перспективных отраслей современного ИТ. Веб-приложение - это полноценная программа, которую пользователи используют через браузер. Именно поэтому данный вид разработки получил такой большой интерес в последнее время, ведь браузеры есть у каждого. Многие предприниматели переносят свой бизнес в диджитал сферу и выбирают для этого именно веб-приложение. Почти любое веб-приложение можно реализовать в короткие сроки по сравнению с обычным десктопным приложением, а так же любое такое приложение будет кроссплатформенным, ведь оно работает только в браузере клиента. С помощью такого метода программирования можно с легкостью реализовывать даже самые сложные бизнес-процессы и задумки. В рамках данной работы мы будем использовать термин "сервис".

Современный Интернет невозможно представить без паролей, систем шифрования, аутентификации и авторизации пользователей. Одна из самых важных частей каждого человека в сети - это его пароли, но, к сожалению, достаточно много людей не хотят уделять внимание их безопасности. Когда же дело касается безопасности корпоративных паролей, то разговор идет не только о надежной защите данных конкретного пользователя, но и о сохранении имиджа и репутации всей компании. Одновременно с этим, многие компании в своей работе используют незащищенные и неудобные инструменты управления и хранения паролей. Не стоит забывать, что от правильной тактики хранения паролей зависит и уровень защиты от кибератак. Для компании критически важно, чтобы у администратора системы хранения паролей было понимание того, у кого есть тот или иной доступ к информации. Многие используют слишком примитивные способы хранения паролей, например текстовые файлы на рабочем столе компьютера или документы в облаке. Для того, чтобы избежать таких способов и грамотно управлять корпоративными паролями требуется специальный инструмент, заточенный под конкретные нужды компании. Такие инструменты называют менеджерами паролей. В большинстве случаев, компании предпочитают рациональный, надежный и удобный подход к внутреннему приложению собственной команды. Следовательно, наилучшим способом реализации сервиса для централизованного управления паролями можно считать именно веб-приложение.

Объектом исследования является централизованное корпоративное управление паролями.

Предметом исследования является сервис для безопасного, централизованного и удобного управления паролями.

Целью выпускной квалификационной работы является создание сервиса

для удобного, безопасного и централизованного управления паролями в компании. Данный сервис позволит небольшим компаниям грамотно работать со своими паролями, а так же не переживать за безопасность их хранения.

Для достижения этой цели необходимо решить следующие задачи:

- произвести аудит существующих сервисов для хранения паролей(менеджеров паролей)
- создать клиентскую часть(интерфейс) сервиса
- создать серверную часть сервиса

???

1. Технологический стек

1.1. Серверная часть

В качестве языка программирования мы выбрали TypeScript - это язык программирования для веб-разработки, основанный на JavaScript, который позволяет писать статически типизированный код и следовать лучшим ООП практикам.

Для написания backend-части сервиса использовался Nest.js - это мощный и гибкий фреймворк для написания эффективного серверного кода. Он построен на Express.js - самом популярном решении для создания серверного кода для платформы Node.js. Использование фреймворков помогает нам писать меньше шаблонного кода и сосредоточиться на решении конкретных бизнес-задач. Преимущества решения:

- 1) обеспечивает заранее масштабируемую и чистую архитектуру, требует от программиста писать хороший код
- 2) поддерживает множество инструментов “из коробки”
- 3) присутствуют мощные инструменты командного интерфейса для быстрой разработки модулей
- 4) хорошая реализация инъекции зависимостей(Dependency Injection) из коробки
- 5) поддержка сообщества

Создатели данного фреймворка вдохновлялись Angular и многие вещи написаны с использованием декораторов. Декораторы - это способ “оборачивания” функций, они позволяют модифицировать поведение классов.

Для хранения данных в сервисе использовалась СУБД PostgreSQL. Для связи базы данных и объектов в приложении использовалась технология ORM(Object-Relational Mapping - объектно-реляционное отображение) и ее конкретная реализация - Sequelize ORM.

Для решения аутентификации, авторизации и контроля доступа использовался продукт Keycloak. Преимущества решения:

- 1) единый вход в приложение(SSO)
- 2) LDAP/Active Directory интеграция и быстрый перенос пользователей
- 3) хороший и документированный API

Документация конечных точек API для серверной части задокументирована с помощью OpenAPI 3.0 Swagger.

1.2. Клиентская часть

Для клиентской части приложения использовался язык программирования JavaScript и фреймворк Vue.js 2.

Для интерфейсов средней сложности это идеальный выбор, потому что данное решение легко масштабировать, у него прекрасная документация, а так же он очень производительный.

В роли менеджера состояния у нас выступает библиотека и одновременно паттерн управления состоянием Vuex 3. Это самое популярное решение для Vue, потому что данные хранятся с правилами, которые гарантируют изменение только предсказуемым образом. По мере роста приложения возникает потребность передавать данные между компонентами и это начинает приносить неудобства в использовании данных. Vuex призван решить эту проблему. В нем данные хранятся централизованно.

Управление маршрутами происходит с помощью Vue Router - официального роутера для Vue.js.

В качестве UI библиотеки для написания компонентов интерфейса мы использовали библиотеку Vuetify 2.6. Данное решение хорошо документировано и позволяет писать масштабируемые компоненты.

2. Оформление элементов текста

2.1. Рисунки и таблицы

Иллюстрации (фотографии, рисунки, чертежи, графики, диаграммы и т. п.) обозначаются сокращенно словом «Рис.», которое пишется под иллюстрацией с прописной буквы и выделяется полужирным шрифтом. Нумеруются иллюстрации арабскими цифрами. Нумерация сквозная по всему тексту ВКР. Пример — рис. 1. Под рисунком по центру размещаются его наименование и поясняющие надписи. Иллюстрации располагают сразу же после ссылки на них в тексте ВКР.



Рис. 1 — Название рисунка

Таблицы нумеруются в рамках раздела арабскими цифрами. Слово «Таблица» и ее номер пишется вверху, с правой стороны над таблицей. Ниже слова «Таблица» посередине строки помещают ее название. Название таблицы должно отражать ее содержание, быть точным и кратким. Название таблицы записывается с прописной буквы и выделяется полужирным шрифтом. Заголовки строк и столбцов выделяются полужирным шрифтом. Пример — табл. 2.1.

Таблица 2.1

Название таблицы

Тип работы	Число глав		
	Одна	Две	Три
Курсовая	3	2	1
Работа бакалавра	2	4	3
Диплом	1	5	6
Магистерская диссертация	0	4	5

2.2. Заголовки и приложения

2.2.1. Заголовки

Заголовки должны четко и кратко отражать содержание соответствующих разделов, подразделов, пунктов. Заголовки печатают, отделяя от номера пробелом, с заглавной буквы. Точка в конце заголовка не ставится. Заголовки выделяют полужирным шрифтом. В заголовках следует избегать сокращений (за исключением общепризнанных аббревиатур). В заголовке не допускается перенос слова на следующую строку и подчеркивание слов. Выравнивание заголовков выполняется по левому краю или по центру строки (единообразно во всей работе) без абзацного отступа. Расстояние между названием глав и последующим текстом должно равняться двум межстрочным интервалам. Такое же расстояние выдерживается между заголовками главы и параграфа.

2.2.2. Приложения

В виде приложений оформляется материал, дополняющий основную часть ВКР. Приложения обозначают прописными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ. Каждое приложение начинается с новой страницы. При этом в верхнем правом углу страницы приводят слово «Приложение», записанное строчными буквами с первой прописной, с указанием номера приложения. Название приложения располагается ниже его обозначения на отдельной строке по центру строчными буквами с первой прописной и выделяется полужирным шрифтом. Приложения должны иметь общую с основной частью документа сквозную нумерацию страниц. В тексте ВКР должны быть даны ссылки на все приложения. Ссылки на приложения в тексте ВКР должны быть организованы в строго нумерационном порядке. Пример оформления — приложение А.

2.3. Нумерация страниц

Все страницы текста ВКР, включая его иллюстрации и приложения, должны иметь сквозную нумерацию. Титульный лист считается страницей № 1, но номер на нем не проставляется. Номера страниц проставляются арабскими цифрами внизу страницы в ее правом углу или по центру. В случае необходимости номер на некоторых страницах может быть проставлен вручную.

3. Формулы

Издательская система \LaTeX [1] предлагает широкий спектр средств для набора математических формул. Подробно эта тема освещается в многочисленных книгах (см., например, [2, 3]). Ниже приводится лишь несколько простых примеров.

Если параметр a равен нулю, а $b \neq 0$, то уравнение $ax = b$ не имеет корней. Определим функцию $\text{sgn} : \mathbb{R} \rightarrow \mathbb{N}$ следующим образом:

$$\text{sgn}(x) = \begin{cases} 1, & \text{при } x \geq 0, \\ -1, & \text{иначе.} \end{cases} \quad (1)$$

Из формулы (1) следует, что $x \text{sgn}(x) = |x|$.

Множество рациональных чисел:

$$\mathbb{Q} := \left\{ \frac{n}{m} \mid n \in \mathbb{Z}, m \in \mathbb{N} \right\}.$$

Вероятность события A при условии, что событие B произошло: $P(A \mid B)$. Вот так выглядит $(n \times k)$ -матрица:

$$M = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1k} \\ m_{21} & m_{22} & \dots & m_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \dots & m_{nk} \end{pmatrix}$$

4. Рисуем с помощью TikZ

Пакет TikZ предлагает удобные инструменты для рисования диаграмм, блок-схем, графов, графиков функций и т. п [4, 5]. При этом рисунки сохраняются в векторной графике, а для надписей используется тот же шрифт, что и в основном тексте. Простейшие примеры использования этого пакета изображены на рис. 2 и 3.

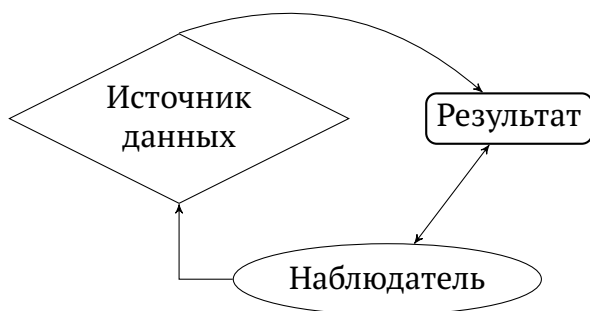


Рис. 2 — Блок-схема

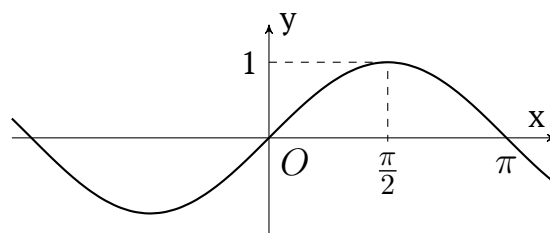


Рис. 3 — График функции $y = \sin(x)$

5. Псевдокод

Пакет `algorithm2e` предлагает широкий спектр инструментов для создания и оформления псевдокода. Также имеется возможность делать ссылки на строки кода. Например, в строке 16 алгоритма 1 целиком содержится цикл типа `do-while`.

Алгоритм 1. Быстрая сортировка

Вход : массив A , индексы начала b и конца e сортируемого фрагмента

Выход : массив A , отсортированный по возрастанию

```
1 Procedure QuickSort( $A, b, e$ )
2   if  $b < e$  then
3      $m \leftarrow \text{Partition}(A, b, e)$ 
4     QuickSort( $A, b, m$ )
5     QuickSort( $A, m + 1, e$ )
6   end
7 end

8 Function Partition( $A, b, e$ )
9    $v \leftarrow A[b]$ 
10   $i \leftarrow b - 1$ 
11   $j \leftarrow e + 1$ 
12  loop
13    do
14       $i \leftarrow i + 1$ 
15      while  $A[i] < v$ 
16      do  $j \leftarrow j - 1$  while  $A[j] > v$ 
17      if  $i \geq j$  then
18        return  $j$ 
19      end
20      поменять местами  $A[i]$  и  $A[j]$ 
21    endloop
22 end
```

Заключение

В заключении подводятся итоги выполненной работы, рассказывается о том, что удалось и что не удалось сделать, описываются перспективы продолжения исследований.

Список литературы

- [1] TeX в ЯрГУ [Электронный ресурс]. URL: <http://www.tex.uniyar.ac.ru> (дата доступа: 20.05.2017).
- [2] Oetiker T., Partl H., Hyna I., Schlegl E. The Not So Short Introduction to $\LaTeX 2_{\epsilon}$ [Электронный ресурс]. URL: <https://tobi.oetiker.ch/lshort/lshort.pdf> (дата доступа: 01.06.2017).
- [3] Котельников И. А., Чеботаев П. З. $\LaTeX 2_{\epsilon}$ по-русски. 3-е изд., перераб. и доп. Новосибирск : Сибирский хронограф, 2004. 496 с.
- [4] Tantau T. PGF — Create PostScript and PDF graphics in \TeX [Электронный ресурс]. URL: <https://www.ctan.org/pkg/pgf> (дата доступа: 17.05.2017).
- [5] Кирютенко Ю. А. TikZ & PGF. Создание графики в $\LaTeX 2_{\epsilon}$ -документах. Ростов-на-Дону, 2014. 277 с. URL: <https://open-edu.sfedu.ru/files/pgf-ru-all-method.pdf>
- [6] Cook S. A. The complexity of theorem-proving procedures // Proceedings of the third annual ACM symposium on Theory of computing. ACM, 1971. P. 151–158.
- [7] Пупырев С. Н., Тихонов А. В. Визуализация динамических графов для анализа сложных сетей // Модел. и анализ информ. систем. 2010. Т. 17, № 1. С. 117–135.
- [8] Кузьмин И. Г. Некоторые проблемы государственных финансов в современной России // Российские предприятия в системе рыночных отношений : материалы научно-практич. конф. Ярославль, 17–18 окт. 2000 г. / отв. ред. Л. Б. Парфенова. Ярославль, 2000. С. 86–90.
- [9] ГОСТ Р 517721-2001. Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования. Введ. 2002-01-01. М. : Изд-во стандартов, 2001. IV, 27 с.

Исходный код программы на C++

```

1  #include <iostream>
2  #include <string>
3  #include <list>
4  using namespace std;
5
6  // Copy data from container 'f' to container 't'
7  template <typename Tfrom, typename Tto>
8  inline void copydata(Tfrom &f, Tto &t)
9  {
10     for(typename Tfrom::iterator it=f.begin(); it!=f.end(); it++)
11         t.push_back(*it);
12 }
13
14 // Sort string 'source'
15 string sorts(string &source)
16 {
17     list<char> tmp;
18     copydata (source, tmp);
19     tmp.sort();
20     copydata (tmp, source);
21     return source;
22 }
23
24 int main()
25 {
26     string source;
27     cout << "Print_something\n";
28     getline (cin, source); // Get data from command line
29     cout << "Your_string:_\" << source << "\"'\n";
30     cout << "After_sort:_\" << sorts(source) << "\"'\n";
31     return 0;
32 }

```

Исходный код программы на Python

Пример кода на Python 3 (взят с официального сайта), реализующий симулятор машины Тьюринга для сложения унарных чисел (типа $11 + 111$). Листинг позволяет делать (автоматическую) ссылку на какую-нибудь строку. Например, на строку 15 с командой `print(tape)`.

```

1  # prog is indexed by the current tape symbol (0 or 1)
2  # and then by state (a kind of instruction pointer)
3  # to get an 'instruction' comprising:
4  #   symbol to write on current tape position,
5  #   head action (-1 = move left, +1 = move right)
6  #   next state (like a goto jump).
7
8  #           symbol 0      symbol 1
9  prog = [[(1, +1, 1), (1, +1, 0)],           # state 0
10          [(0, -1, 2), (1, +1, 1)],         # state 1
11          [(0, +1, 2), (0, +1, 9)]]          # state 2
12 tape = [1,1,0,1,1,1,0,0,0]                 # The data tape
13 head = 0                                   # head position on tape
14 state = 0                                  # instruction pointer
15 print(tape)
16 while state != 9:                           # while not halt:
17     symbol = tape[head]                       # read current tape symbol
18     symbol, dir, state = t = prog[state][symbol] # lookup instruction
19     print('_' * (head * 3 + 1) + '^__' + str(t)) # display progress
20     tape[head] = symbol                       # write new symbol on tape
21     print(tape)
22     head = head + dir                         # move tape head

```