

Национальный исследовательский Университет ИТМО
Мегафакультет информационных и трансляционных технологий
Факультет инфокоммуникационных технологий

Инфокоммуникационные системы и ТЕХНОЛОГИИ

Лабораторная работа №1

Работу
выполнил:
С. А. Григорян
Группа: К3140
Преподаватель:
Н. Н. Горлушкина

Санкт-Петербург
2022

Содержание

Постановка задачи	3
1. Общее описание	4
2. Интерфейсы системы	4
3. Выбор архитектуры системы	5
4. Обоснование выбора технологий и программных средств	5
5. Листинг	7
Заключение	9
Список использованных источников	10

Введение

Целью разрабатываемого программного продукта является обеспечение простого и удобного ведения собственной информационной базы, включающей в себя систему ведения заметок и категоризированных закладок.

Каждый день человек сталкивается со множеством информации, которую необходимо запомнить и использовать в дальнейшем, но вследствие человеческого фактора данная информация может быть забыта или утеряна. Для того, чтобы выносить ключевые моменты из обработанного материала, необходима система, которая бы позволяла просто и удобно реализовывать эту задачу. В большинстве случаев разработанный продукт будет актуален студентам и работникам различных научных направлений, так как именно в перечисленных сферах чаще всего возникает потребность в постоянном поиске и обработке информации.

Для разработки системы необходимо выполнить следующие задачи:

- анализ предметной области;
- формирование функциональных требований к системе;
- серверное веб-приложение;
- клиентское веб-приложение;
- плагин для браузера;
- тестирование разработанной системы.

1. Общее описание

Назначением разработанного проекта является помощь человеку разгрузить свою «оперативную память», предоставляя возможность вести список заметок и закладок, к которым пользователь может обратиться в любой момент с любого устройства.

Большинство приложений с функционалом создания заметок и закладок, требуют от пользователя знаний и умений работы в конкретном сервисе. Однако это отнимает много времени, а необходимость в записи информации может возникнуть в любой момент. Начинаящий пользователь может запутаться или записать в заметках телефона без желания опробовать интерфейс. Можно сделать вывод о том, что крайне не хватает приложения с удобным интерфейсом и направленным функционалом. Поэтому наша команда с удовольствием взялась за разработку такого решения и поставила следующую цель: обеспечить простое и удобное ведение собственной базы данных, с возможностью ведения заметок и добавления закладок.

2. Интерфейсы системы

Разработанный сервис позволяет создать закладку используя такие данные, как URL сайта и его название. Закладку можно удалить, изменить и скопировать. С помощью нажатия на кнопку «Перейти» вы сможете перейти на сайт из закладки. Также реализована возможность сортировки закладок по какому-либо признаку и поиска, что позволяет легко работать с большим объемом информации.

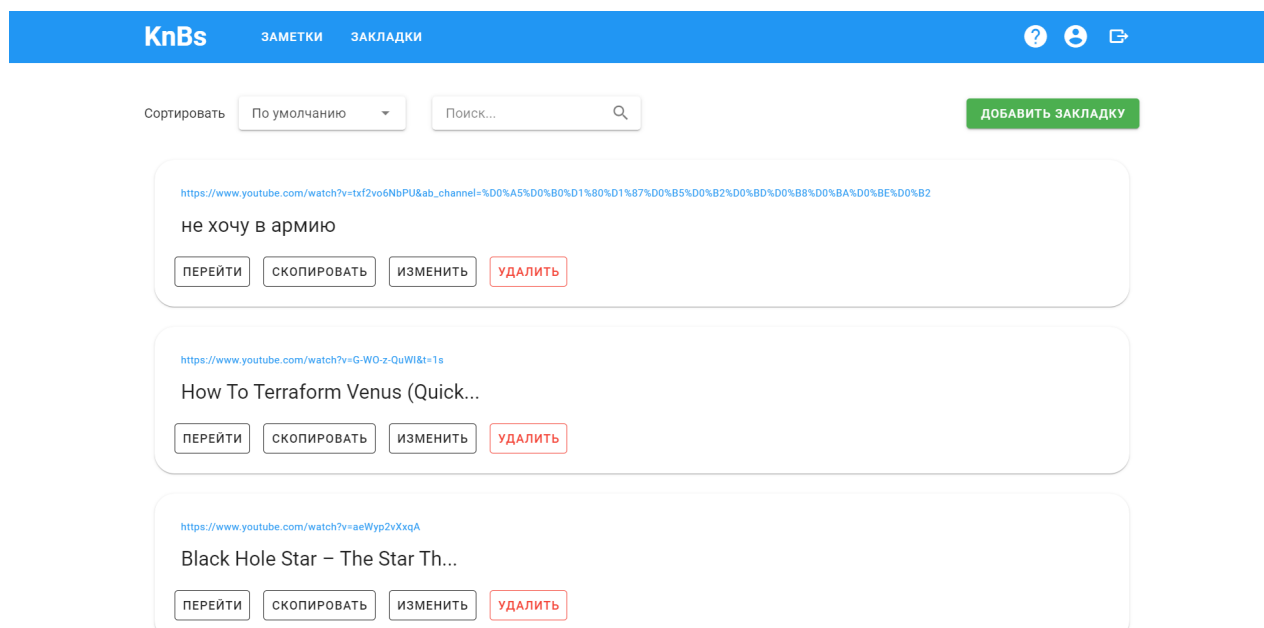


Рисунок 2.1. Страница закладок

На странице «Заметки» отображается весь список добавленных заметок. Заметки можно изменять, а также удалять не нужные, используя необходимые кнопки. Для добавления необходимо ввести название и нужную вам информацию, после чего нажать на кнопку «Добавить», и заметка появится в общем списке. Для заметок был расширен функционал сортировки, в отличии от закладок. Заметки можно сортировать по умолчанию, по дате создания, по дате редактирования, по названию. Также, как и в создании закладок, реализована возможность поиска.

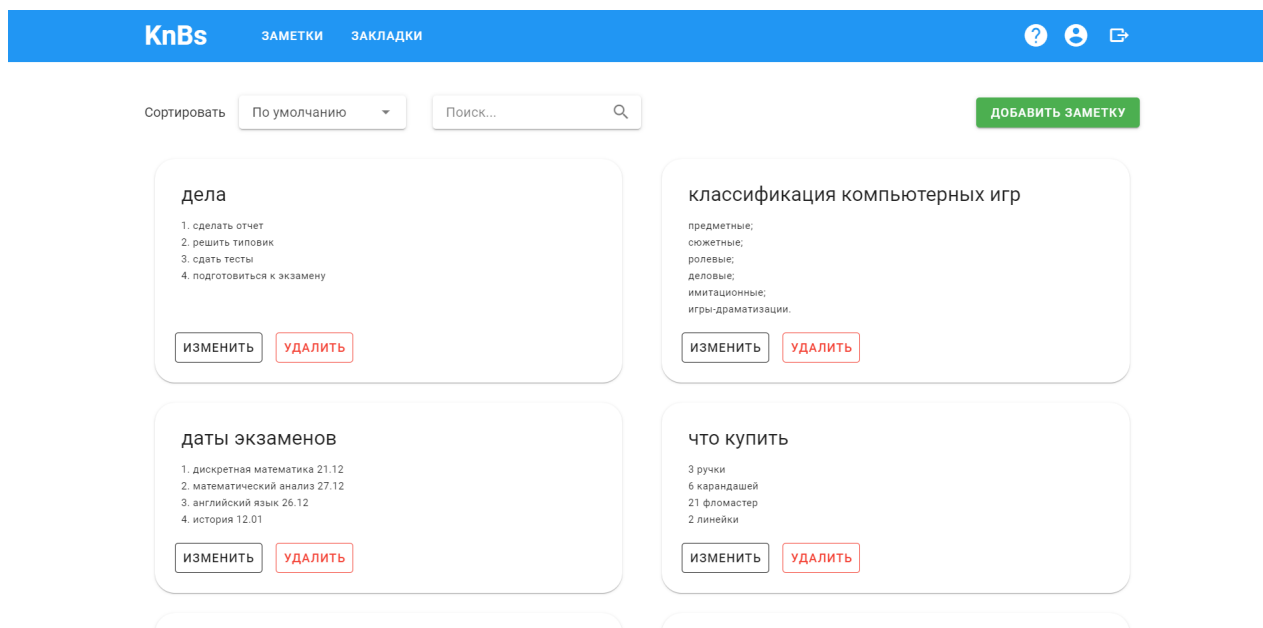


Рисунок 2.2. Схема неструктурированной сети

Наше приложение может быть запущено как на удаленном сервере, так и локально на компьютере пользователя, что дает возможность работать с ним даже без доступа в интернет. Для подключения к серверу и сохранения закладок необходимо ввести его адрес в соответствующее поле в окне расширения. Также пользователь должен ввести свой ключ доступа к приложению, с помощью которого происходит авторизация для того, чтобы получить доступ к созданию закладок.

3. Выбор архитектуры системы

Веб-приложение – клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одно из преимуществ такого подхода это независимость клиентов от конкретной операционной системы. Рекомендовано ознакомиться с сайтом [1]

4. Обоснование выбора технологий и программных средств

Для разработки веб-сервиса был выбран язык программирования JavaScript – является одним из самых популярных языков программирования.

Именно в области Frontend задействовано огромное число наработок, основанных на Javascript. Наиболее активно используется примерно 25-30 библиотек и фреймворков. Эти готовые шаблоны и решения для стандартных задач существенно экономят время. Они упрощают процесс web-разработки, ускоряют его, снижая стоимость проектов.

Рекомендовано ознакомиться с книгой [2] для более подробного изучения JavaScript.

В таблице 4.1 приведен сравнительный анализ популярных фреймворков для разработки на языке JavaScript.

Таблица 4.1

Сравнительный анализ популярных фреймворков

Критерий	Vue.js	React.js	Angular
Рендеринг	создается копия DOM	создается копия DOM	рендеринг HTML-страниц на стороне сервера
Архитектура компонентов	Высокоуровневый API обеспечивает совместимость для всех библиотек	Необходим поиск и внедрение дополнительных библиотек	Необходим поиск и внедрение дополнительных библиотек
Двустороннее связывание	есть	нет	есть
Декомпозиция объектов	есть	есть	есть
Представление	HTML-шаблоны и JSX	JSX	HTML-шаблоны и JSX

Vue.js – прогрессивный фреймворк с подробно прописанной документацией и множеством примеров. Он позволяет создавать переиспользуемые компоненты за счет архитектурных особенностей фреймворка. Для увеличения производительности Vue.js использует виртуальную копию DOM, а для сохранения целостности данных этот фреймворк поддерживает двустороннее связывание, поэтому для написания front-end проекта был выбран именно Vue.js.

В качестве серверной разработки была выбрана платформа Firebase для разработки мобильных и веб-приложений. Firebase — это облачная база данных, которая позволяет пользователям хранить и получать сохраненную информацию, а также имеет удобные средства и методы взаимодействия с ней.

Firebase хранит текстовые данные в JSON формате и предоставляет удобные методы для чтения, обновления и извлечения данных. Также, Firebase может помочь с регистрацией и авторизацией пользователей, хранением сессий (авторизованные пользователи), медиафайлов к которым с легкостью предоставляет доступ благодаря Cloud Storage.

Также считаю нужным поделиться разработанной мною формулой для заработка денег **1**.

$$M(x) = J + \sum_{n=5}^{\infty} I_n + \sin\left(\frac{2\pi x}{\nu} - W_n\right) \quad (1)$$

Воспользовавшись ее вы останетесь с деньгами после новогодних праздников. С наступающим новым годом!

5. Листинг

```
1 <template>
2   <base-layout ref="notesLayout">
3     <v-row justify="space-between" align="center" class="mt-6">
4       <div class="d-flex align-center">
5         Сортировать
6         <v-select
7           v-model="sort"
8           :items="sortItems"
9           item-title="value"
10          item-value="key"
11          variant="solo"
12          density="compact"
13          class="sort-select ml-4"
14        />
15        <v-text-field
16          v-model="search"
17          density="compact"
18          variant="solo"
19          label="Поиск ... "
20          append-inner-icon="mdi-magnify"
21          class="search-input flex-grow-0 flex-shrink-0 ml-8"
22          single-line
23          hide-details
24        />
25      </div>
26
27      <div>
28        <v-btn color="green" @click="openAddModal">
29          Добавить заметку
30        </v-btn>
31      </div>
32    </v-row>
33
34    <notes-list :items="filteredSortedNotes" />
35  </base-layout>
36 </template>
37
38 <script>
39 import {defineComponent} from 'vue';
40 import {mapActions, mapGetters} from "vuex";
41
42 // Components
43 import BaseLayout from "@/layouts/BaseLayout.vue";
44 import NotesList from "@/components/notes/NotesList.vue";
45 import NoteAddForm from "@/components/notes/NoteAddForm.vue";
46
47 export default defineComponent({
48   name: 'NotesView',
49   components: {
50     NotesList,
51     BaseLayout
52   },
53
54   data: () => ({
55     dialog: false,
56     sort: 'default',
57     sortItems: [
```

```

58     { key: 'default', value: 'По умолчанию' },
59     { key: 'date', value: 'По дате создания' },
60     { key: 'editDate', value: 'По дате редактирования' },
61     { key: 'name', value: 'По названию' }
62   ],
63   search: '',
64 },
65
66   computed: {
67     ...mapGetters({
68       getNotes: 'notes/getNotes'
69     }),
70     filteredSortedNotes() {
71       let list = [...this.getNotes];
72
73       switch (this.sort) {
74         case "name":
75           list = list.sort((o1, o2) =>
76             o1.name.toUpperCase() > o2.name.toUpperCase() ? 1
77             : o1.name.toUpperCase() < o2.name.toUpperCase() ? -1 : 0
78           )
79           break;
80       }
81
82       const normalizedSearch = this.search.trim().toUpperCase();
83       if (normalizedSearch) {
84         list = list.filter(item =>
85           item.name.toUpperCase().includes(normalizedSearch) ||
86           ↪ item.text.toUpperCase().includes(normalizedSearch)
87         )
88       }
89       return list;
90     },
91   },
92
93   mounted() {
94     this.updateNotes();
95   },
96
97   methods: {
98     ...mapActions({
99       updateNotes: 'notes/updateNotes'
100     }),
101     openAddModal() {
102       this.$refs.notesLayout.setComponent(NoteAddForm);
103       this.$refs.notesLayout.openPopup();
104     }
105   }
106
107 });
108 </script>

```

Листинг 1: Листинг страницы закладок

Заключение

В результате выполнения курсового проекта была полностью выполнена основная цель разрабатываемого продукта, удалось создать веб-приложение для предоставления пользователям возможности управления собственной информационной базой «Knowledge Base».

Моей задачей в проекте являлось проектирование архитектуры клиентской части веб-приложение, а также контроль над остальными участниками команды frontend-разработчиков. Мною была размечена структура проекта, настроено окружение и подключены необходимые зависимости. Также сверстаны страницы авторизации и регистрации пользователей.

В дальнейший план развития проекта входит реализация следующих задач:

- возможность добавления нового функционала для выстраивания взаимосвязей между заметками и закладками;
- мобильное приложение для Android и IOS.

Список использованных источников

1. Архитектура клиент-сервер. — 2016. — URL: <https://portal.tpu.ru/SHARED/f/FAS/study/avis/lectures/cli-se.pdf> ; [Электронный ресурс; дата обращения: 02.12.2022].
2. *Кантор И.* Современный учебник JavaScript. — 2018.