Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «ВТ»

Тема: “Поиск”

Выполнил:

Студент 3 курса

Группы ПО-7

Комиссаров А.Е.

Проверил:

Михняев А.Л.

2022

**Цель работы:** Получить представление об основных подходах при разработке поисковых функций в веб — приложениях ColdFusion.

**Ход работы.**

1. В администраторе ColdFusion зарегистрировать коллекцию метаданных, проиндексировав содержимое исходного каталога документов веб-сервера Cold Fusion.
2. Написать интерфейс поиска по файлам созданной коллекции.
3. Создать поисковый интерфейс по содержимому статей, собранных в [таблице News базы данных ProgrammingCF](ProgrammingCF.mdb) :
   1. В администраторе ColdFusion зарегистрировать коллекцию, проиндексировав содержимое соответствующих данных базы данных ProgrammingCF.
   2. Написать интерфейс поиска по данным созданной коллекции.
4. Написать интерфейс ''Верхняя десятка'', создав и заполнив предварительно базу данных критериями поиска.

@Injectable()

export class SearchService {

constructor(private readonly prisma: PrismaService) {}

async searchDocuments(query: string, top: boolean): Promise<any> {

let employeeOptions = {}

let awardsOptions = {}

let keywordsOptions = {}

let usersOptions = {}

if (query) {

employeeOptions = {

where: {

OR: [

{ Department: { contains: query, mode: "insensitive" } },

{ Email: { contains: query, mode: "insensitive" } },

{ Name: { contains: query, mode: "insensitive" } },

{ Title: { contains: query, mode: "insensitive" } },

{ Picture: { contains: query, mode: "insensitive" } },

],

},

}

awardsOptions = {

where: {

OR: [{ Category: { contains: query, mode: "insensitive" } }],

},

}

keywordsOptions = {

where: {

SearchString: { contains: query, mode: "insensitive" },

},

}

usersOptions = {

where: {

OR: [

{ FullName: { contains: query, mode: "insensitive" } },

{ Username: { contains: query, mode: "insensitive" } },

],

},

}

}

let topAmount = 10

if (top) {

employeeOptions = { ...employeeOptions, take: topAmount, orderBy: { Salary: "desc" } }

awardsOptions = { ...awardsOptions, take: topAmount, orderBy: { Amount: "desc" } }

keywordsOptions = {

...keywordsOptions,

take: topAmount,

orderBy: { SearchString: "asc" },

}

usersOptions = { ...usersOptions, take: topAmount, orderBy: { FullName: "asc" } }

}

const employeeData = await this.prisma.employeeDirectory.findMany(employeeOptions)

const awardsData = await this.prisma.incentiveAwards.findMany(awardsOptions)

const keywordsData = await this.prisma.keywords.findMany(keywordsOptions)

const usersData = await this.prisma.users.findMany({

...usersOptions,

select: {

Username: true,

FullName: true,

},

})

return [

{

document: "Employees",

data: employeeData,

},

{

document: "Awards",

data: awardsData,

},

{

document: "Keywords",

data: keywordsData,

},

{

document: "Users",

data: usersData,

},

]

}

async searchArticles(query: string): Promise<any> {

let options = {}

if (query) {

options = {

where: {

OR: [

{ Title: { contains: query, mode: "insensitive" } },

{ Article: { contains: query, mode: "insensitive" } },

],

},

}

}

const articlesData = await this.prisma.news.findMany(options)

return articlesData

}

}

Результаты выполнения программы:

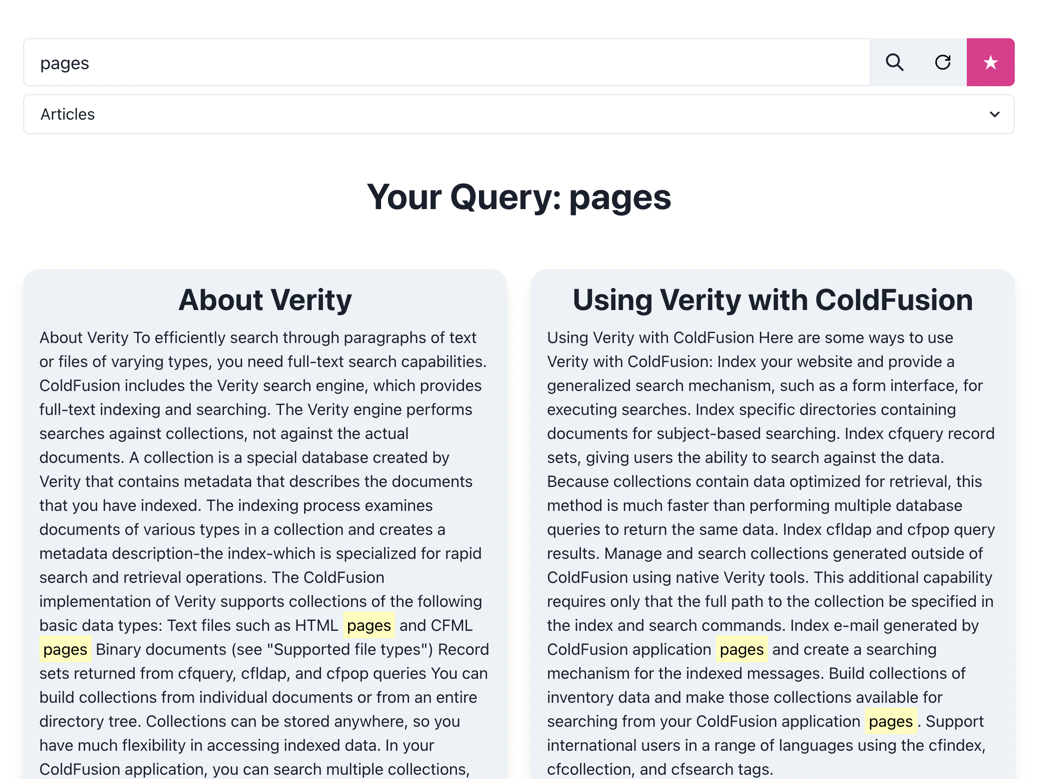
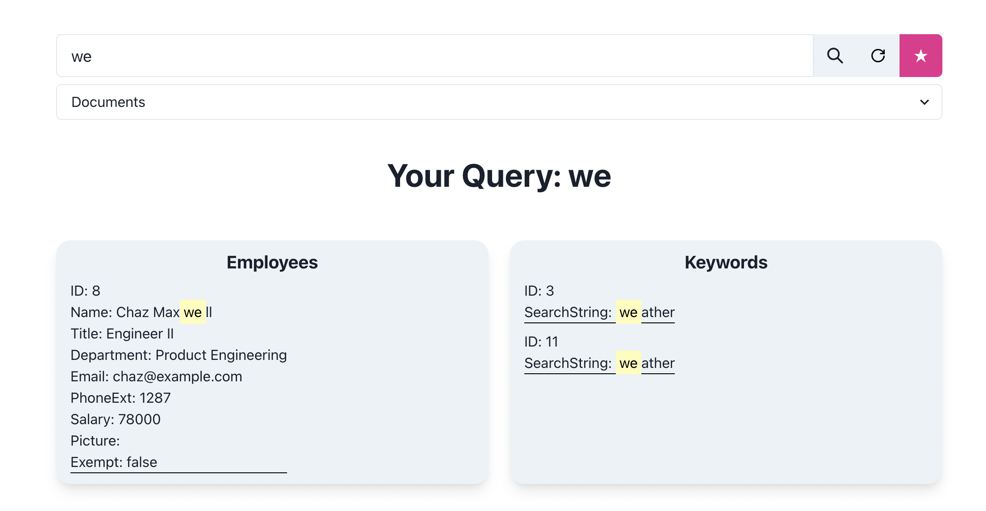


Рис. 1,2 - Результат

**Вывод:** Я получил представление об основных подходах при разработке поисковых функций в веб — приложениях ColdFusion.