

Интеллектуальные диалоговые системы

Признаки, назначение,
примеры, реализация

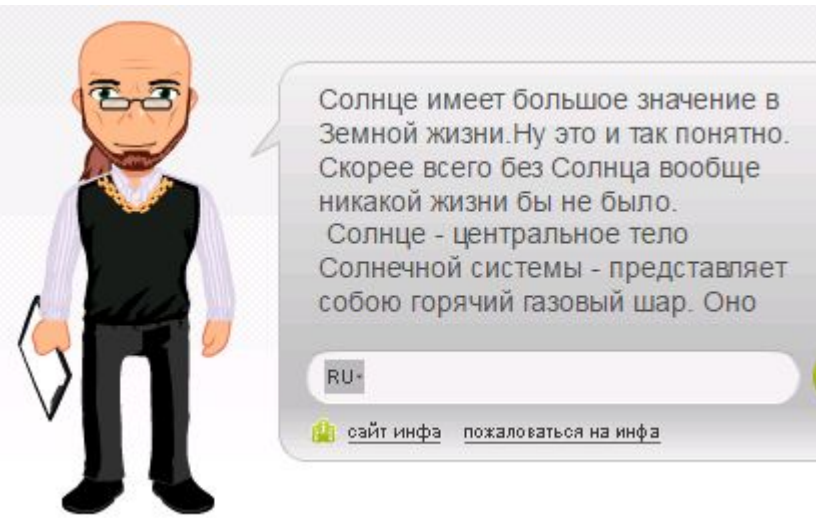
Признаки интеллектуальных диалоговых систем

- Наличие базы знаний
- Обучаемость
- Формулировка реплик на естественном языке из имеющихся знаний
- Исполнение назначения
- Реплики – полноценные предложения, не набор ссылок

Назначение интеллектуальных диалоговых систем

- Поддержка пользователей
- Консультация пациентов (медицина)
- Разъяснение (образование)

Примеры интеллектуальных диалоговых систем



Инфы – интеллектуальные
слуги (iii.ru)



вокруг чего вращается земля

Все

Видео

Картинки

Новости

Карты

Ещё

Результатов: примерно 390 000 (0,55 сек.)

Земля вращается вокруг наклонной оси с запада на восток. Половина земного шара освещается солнцем, там в это время день, вторая половина находится в тени, там ночь. Благодаря **вращению Земли** происходит смена дня и ночи. Один оборот **вокруг** своей оси **Земля** делает за 24 часа — сутки.

Вращение Земли вокруг своей оси и вокруг Солнца
<https://geographyofrussia.com/vrashhenie-zemli-vokrug-svoego-solntsa>

Семантический поиск Google

Инструменты для создания диалоговых систем

- Конструкторы онтологий (Protégé)
- Компоненты полнотекстового поиска в составе СУБД (MySQL, MS SQL)
- Библиотеки для морфологического анализа слов (ruMorphu), и синтаксического анализа предложений (SDK грамматического словаря)

Сложности естественного языка

- Произвольный порядок слов

Буря мглою небо кроет

Буря кроет мглою небо

- Омонимия слов и целых предложений

Эти типы стали есть у нас на складе

- Синонимия

рассматривать / изучать

- Морфология

падежи / числа / лица / роды / ...

Научный стиль речи

- Четкий порядок членов предложения
- Использование терминологии во избежание синонимии

Принципы архитектуры

2 этапа, аналогично поисковым системам:

- Наполнение базы знаний (индексация)
- Поиск подходящей реплики в базе знаний

3 компонента:

- База знаний
- Модуль индексирования
- Модуль диалога с пользователем

Реализация базы знаний

- Размеченный файл (XML, JSON)
- Таблица реляционной базы данных
- Сетевая база данных

Требуется обеспечить удобство
заполнения, анализа при поиске и
получения грамотных реплик

Индексация

- Перевод слов в начальную форму (инфинитив глаголов и именительный падеж единственного число именных частей речи)
- Выделение частей (ключевых объектов и типов связей)

Индексация вручную

~~Из этого следует вывод, что~~

для превращения воды в пар

требуется

приток теплоты, подобно тому как это

имеет место при превращении

~~кристалла~~ (льда) в жидкость

Лед является кристаллом

Индексация вручную

```
string[,] knowledge = {  
    {"для превращения воды в пар",  
        "требуется",  
        "приток теплоты, подобно тому как это имеет место " +  
        "при превращении кристалла в жидкость"},  
    {"лед", "является", "кристаллом"}  
};
```

Поиск в базе знаний

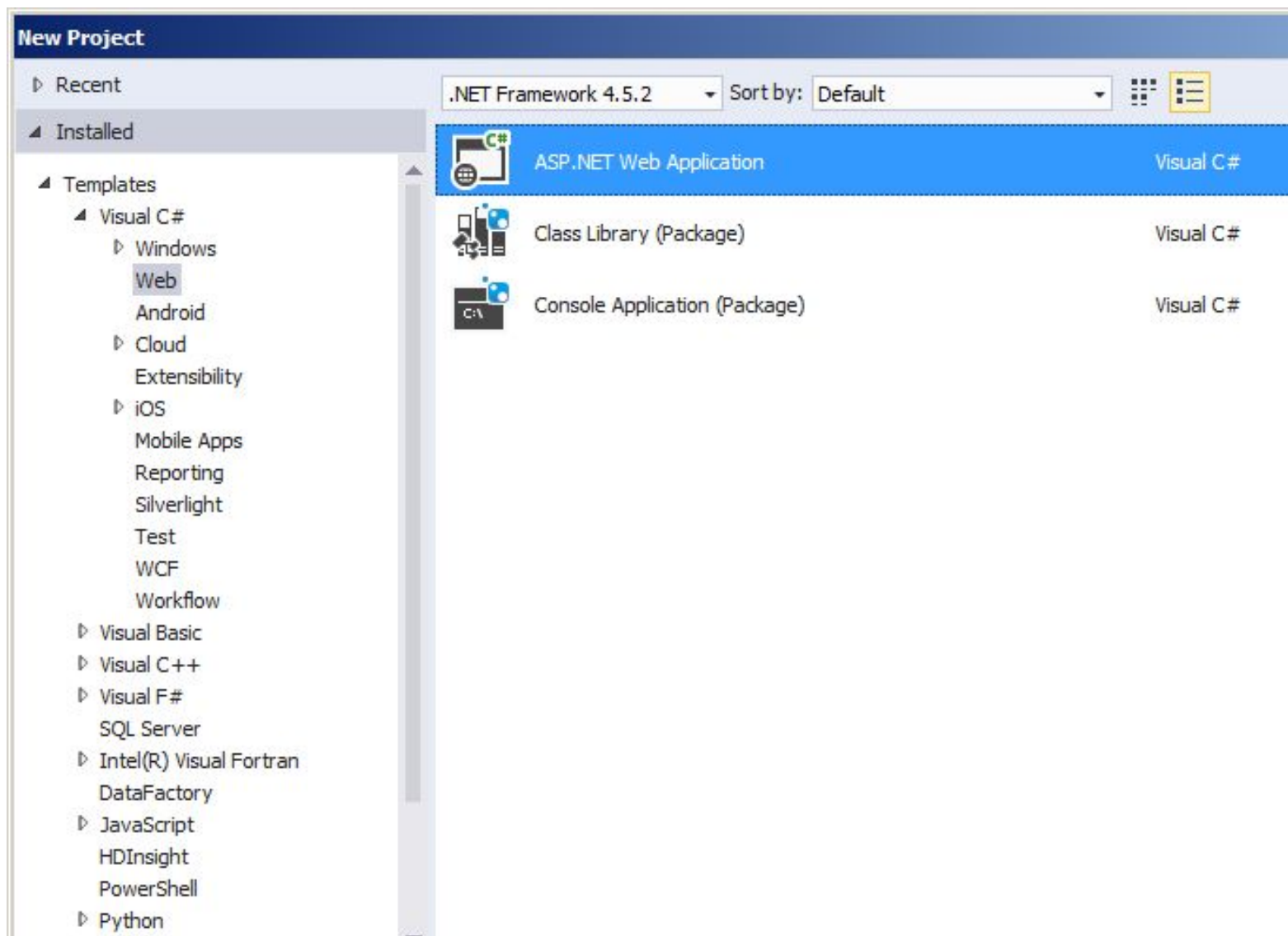
Проблемы:

- Синонимия
- Морфология

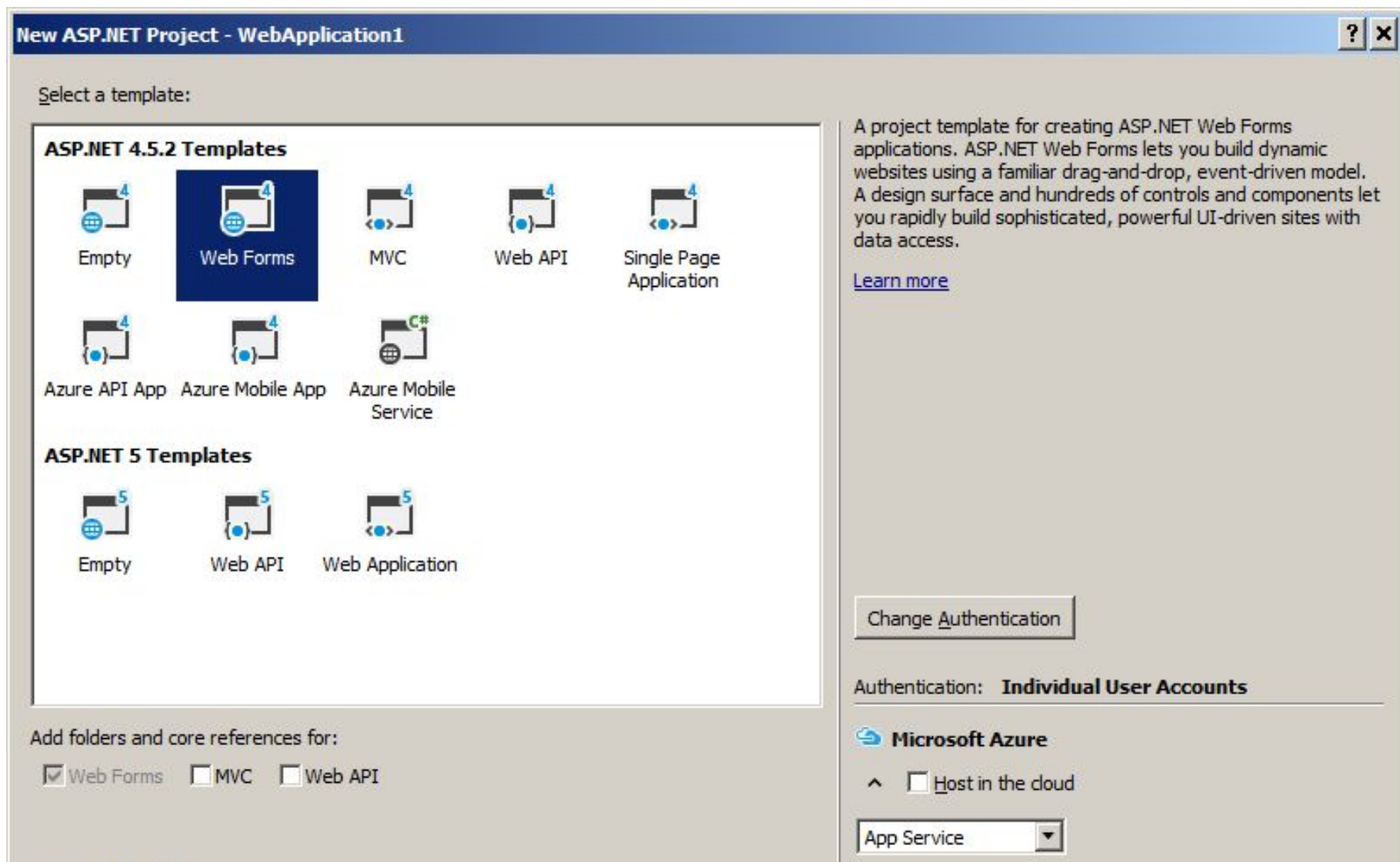
Инструменты:

- Функции работы со строками
- Регулярные выражения
- Платформа ASP.NET

Создание веб-приложения ASP.NET



Создание веб-приложения ASP.NET

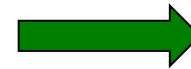


Структура проекта

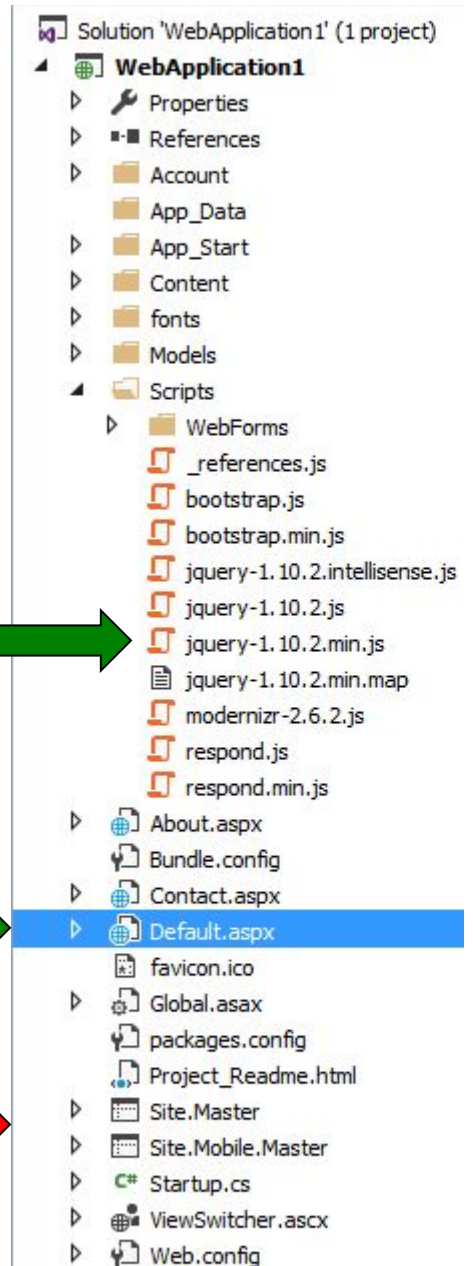
Использовать в Default.aspx
для отправки асинхронных запросов



Сделать самостоятельной страницей



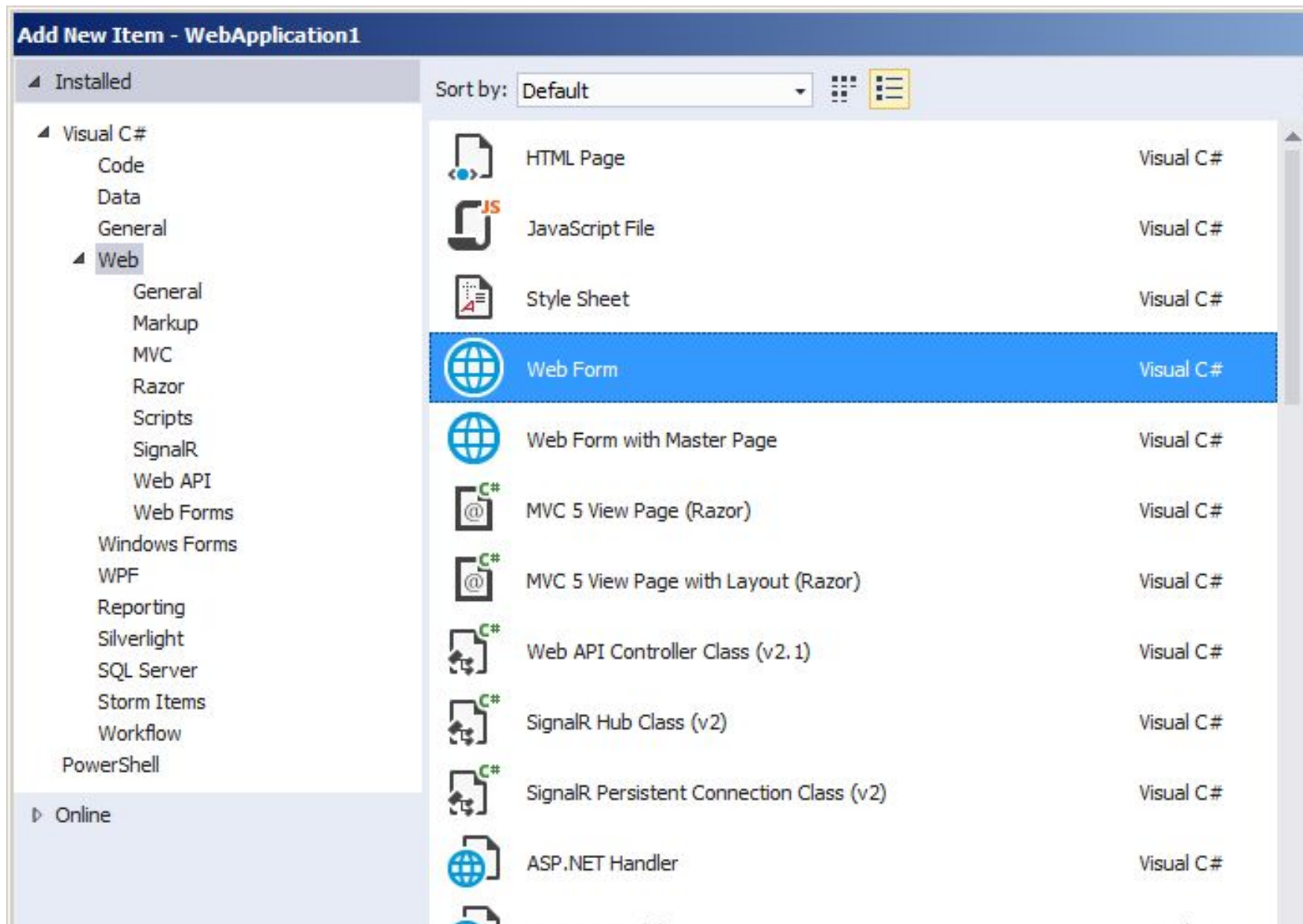
Удалить на ненадобность



Клиентская часть (Default.aspx)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ru">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Диалоговый модуль</title>
  <link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
  <script language="javascript" src="/Scripts/jquery-1.10.2.min.js"></script>
  <script>...</script>
  <style>...</style>
</head>
<body>
  <div class="page">
    <div class="title">
      <h1>
        Диалоговый модуль
      </h1>
    </div>
    <div class="content">
      <p>
        <textarea id="txt">чем являются измерения ЭДС</textarea>
      </p>
      <p>
        <button onclick="send()">
          Поиск ответа</button>
        </p>
      </div>
      <div id="result">
        Здесь будут выводиться результаты.
      </div>
    </div>
  </body>
</html>
```

Серверная часть (Parser.aspx)



Отправка асинхронного запроса

```
Default.aspx*  X
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ru">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Диалоговый модуль</title>
  <link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
  <script language="javascript" src="/Scripts/jquery-1.10.2.min.js"></script>
  <script language="javascript">
    function send() {
      var txt = $("#txt").val();
      //отправка
      $.post("Parser.aspx", { "txt": txt }, show);
    }
    function show(data, success) {
      //обработка ответа на запрос
      document.getElementById("result").innerHTML=data;
    }
  </script>
  <style>...</style>
</head>
<body>
```

Поиск с помощью регулярных выражений

- Разбиение вопроса на слова
- Поиск сказуемого в вопросе
- Формирование регулярного выражения для сказуемого
- Формирование регулярного выражения для подлежащего
- Поиск по двум регулярным выражениям
- Если результата нет, попытка поиска только по подлежащему

Разбиение вопроса на слова

```
//функция, которая делает первую букву маленькой
string small1(string str)
{
return str.Substring(0, 1).ToLower() + str.Substring(1);
}

//главная функция, обрабатывающая запросы клиентов
protected void Page_Load(object sender, EventArgs e)
{
    //знаки препинания
    char[] separators = "'\",.!?()[\\".ToCharArray();
    //получение текста из параметра запроса
    string txt = small1(Request["txt"]);
    //добавление пробелов перед знаками препинания
    for (int i = 0; i < separators.Length; i++)
        txt = txt.Replace("" + separators[i], " " + separators[i]);
    //массив слов и знаков препинания
    List<string> words = new List<string>(txt.Split());
}
```

Псевдоокончания для поиска сказуемых

```
//массив пар нужных псевдоокончаний и их тегированных замен
string[,] endings = {
{"ет", "(ет|ут|ют) "}, {"ут", "(ет|ут|ют) "}, {"ют", "(ет|ут|ют) "},
{"ит", "(ит|ат|ят) "}, {"ат", "(ит|ат|ят) "}, {"ят", "(ит|ат|ят) "},
{"ется", "(ет|ут|ют) ся"}, {"утся", "(ет|ут|ют) ся"}, {"ются", "(ет|ут|ют) ся"},
{"ится", "(ит|ат|ят) ся"}, {"атся", "(ит|ат|ят) ся"}, {"ятся", "(ит|ат|ят) ся"},
{"ен", "ен"}, {"ена", "ена"}, {"ено", "ено"}, {"ены", "ены"},
{"ан", "ан"}, {"ана", "ана"}, {"ано", "ано"}, {"аны", "аны"},
{"жен", "жен"}, {"жна", "жна"}, {"жно", "жно"}, {"жны", "жны"}};
//черный список слов, попадающих по ошибке
string[] blacklist = { "замена", "замены", "атрибут", "маршрут",
                      "член", "нет" };
```

Поиск сказуемого в вопросе

```
//функция определения соответствующих псевдоокончаний
private int getEnding(string word)
{
    //проверка по черному списку
    if (blacklist.Contains(word)) return -1;
    //перебор псевдоокончаний
    for (int j = 0; j < endings.Length / 2; j++)
    {
        //проверка, оканчивается ли i-ое слово на j-ое псевдоокончание
        if (word.EndsWith(endings[j], 0))
        {
            return j;    //возврат номера псевдоокончания
        }
    }
    return -1;    //если совпадений нет - возврат -1
}
```

Формирование регулярных выражений

```
using System.Text.RegularExpressions;

//замена псевдоокончания на набор возможных окончаний
words[i] = words[i].Substring(0, words[i].Length -
    endings[ending, 0].Length) + endings[ending, 1];
//создание регулярного выражения для поиска по сказуемому из вопроса
Regex predicate = new Regex(words[i]);
//для кратких прилагательных захватываем следующее слово
if (endings[ending, 0] == endings[ending, 1])
{
    predicate = new Regex(words[i] + " " + words[i + 1]);
    i++;
}
//создание регулярного выражения для поиска по подлежащему из вопроса
Regex subject = new Regex(".*" +
    string.Join(".*", words.ToArray().Skip(i + 1)) +
    ".*");
```

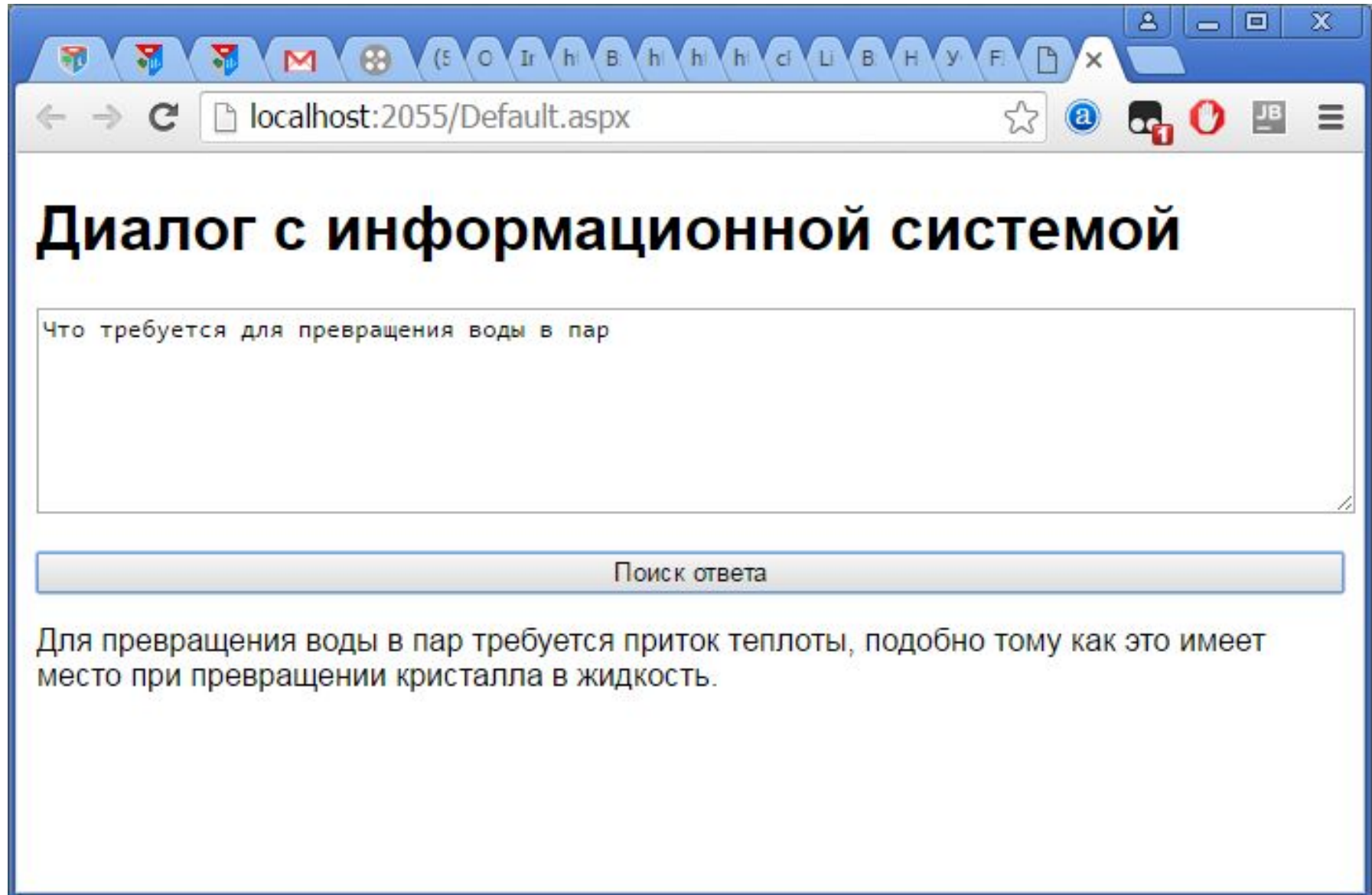
Поиск по двум регулярным выражениям

```
//поиск совпадений с шаблонами среди связей семантической сети
for (int j = 0; j < knowledge.Length / 3; j++)
{
    if (predicate.IsMatch(knowledge[j, 1]) &&
        (subject.IsMatch(knowledge[j, 0]) ||
         subject.IsMatch(knowledge[j, 2])))
    {
        //создание простого предложения из семантической связи
        Response.Write(big1(knowledge[j, 0] + " " +
                             knowledge[j, 1] + " " + knowledge[j, 2] + ". "));
        result = true;
    }
}
```

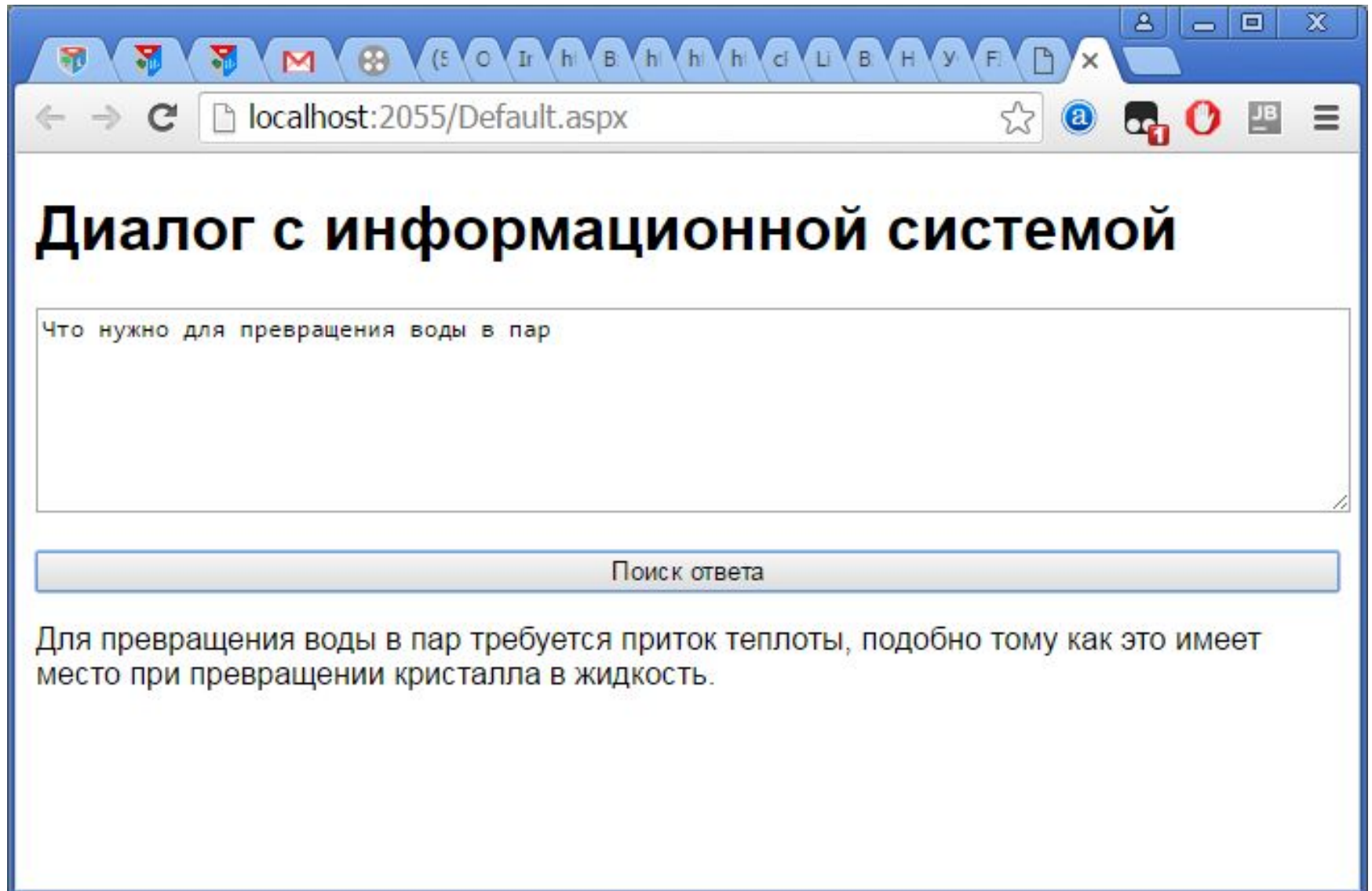
Поиск по подлежащему, если предыдущий поиск не дал результатов

```
//если совпадений с двумя шаблонами нет,  
if (result == false)  
    //поиск совпадений только с шаблоном подлежащего  
    for (int j = 0; j < knowledge.Length / 3; j++)  
    {  
        if ((subject.IsMatch(knowledge[j, 0]) ||  
            subject.IsMatch(knowledge[j, 2])))  
        {  
            //создание простого предложения из семантической связи  
            Response.Write(big1(knowledge[j, 0] + " " +  
                knowledge[j, 1] + " " + knowledge[j, 2] + ". "));  
            result = true;  
        }  
    }  
}
```

Отладка (Default.aspx)



Синонимия



Пути развития

- Добавление в качестве ключевых объектов изображений, звука и видео
- Выделение в ответе ссылок на упоминаемые ключевые объекты
- Озвучивание ответа при помощи веб-сервисов или специальных библиотек
- Реализация ввода вопроса голосом