

ДОДАТОК А
приклад коду

ЛІСТИНГ модуля Form2.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

namespace Estimate
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DataIn di = new DataIn();
            di.ShowDialog();
        }
        private void Form2_FormClosed(object sender, FormClosedEventArgs e)
        {
            Application.Exit();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Mark newForm = new Mark();
            newForm.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Diagrama df = new Diagrama();
            df.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Perregliad pf = new Perregliad();
            pf.ShowDialog();
        }

        private void button5_Click(object sender, EventArgs e)
        {
        }
    }
}
```

```

        Application.Exit();
    }
}
}

```

ЛІСТИНГ модуля DataIn.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

namespace Estimate
{
    public partial class DataIn : Form
    {
        public DataIn()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Rate temp= new Rate();

            temp.func = (Convert.ToDouble(tbF1.Text) + Convert.ToDouble(tbF2.Text) +
            Convert.ToDouble(tbF3.Text) + Convert.ToDouble(tbF4.Text) + Convert.ToDouble(tbF5.Text))
            / 5;
            temp.usab = (Convert.ToDouble(tbU1.Text) + Convert.ToDouble(tbU2.Text) +
            Convert.ToDouble(tbU3.Text) + Convert.ToDouble(tbU4.Text)) / 4;
            temp.rel = (Convert.ToDouble(tbR1.Text) + Convert.ToDouble(tbR2.Text) +
            Convert.ToDouble(tbR3.Text) + Convert.ToDouble(tbR4.Text)) / 4;
            temp.perf = (Convert.ToDouble(tbP1.Text) + Convert.ToDouble(tbP2.Text) +
            Convert.ToDouble(tbP3.Text)) / 3;
            temp.supr = (Convert.ToDouble(tbS1.Text) + Convert.ToDouble(tbS2.Text) +
            Convert.ToDouble(tbS3.Text)) / 3;
            temp.nazva=tbNazva.Text;

            BinaryFormatter bf = new BinaryFormatter();

            using (FileStream fs = new FileStream("baza.bin", FileMode.Append))
                bf.Serialize(fs, temp);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

ЛІСТИНГ модуля Perregliad.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

namespace Estimate
{
    public partial class Perregliad : Form
    {
        public Perregliad()
        {
            Rate[] p;
            InitializeComponent();
            richTextBox1.SelectionFont = new Font("Arial",10,
System.Drawing.FontStyle.Bold);
            richTextBox1.SelectedText="          Назва          Функціональність          Надійність
Юзабіліті          Продуктивність          Супроводжуваність\n";
            richTextBox1.SelectedText +=
"===== \n";
            BinaryFormatter bf = new BinaryFormatter();

            using (FileStream fs = new FileStream("baza.bin", FileMode.OpenOrCreate))
            {
                while(fs.Position!=fs.Length)
                {
                    Rate t = (Rate)bf.Deserialize(fs);
                    string str = " " + t.nazva.PadLeft(10, ' ');
                    string str1 =
Convert.ToString(Math.Round(t.func,2,MidpointRounding.AwayFromZero));
                    str += str1.PadLeft(20, ' ');
                    str1 = Convert.ToString(Math.Round(t.rel, 2,
MidpointRounding.AwayFromZero));
                    str += str1.PadLeft(30, ' ');
                    str1 = Convert.ToString(Math.Round(t.usab, 2,
MidpointRounding.AwayFromZero)) + " ";
                    str += str1.PadLeft(30, ' ');
                    str1 = Convert.ToString(Math.Round(t.perf, 2,
MidpointRounding.AwayFromZero)) + " ";
                    str += str1.PadLeft(20, ' ');
                    str1 =Convert.ToString(Math.Round(t.supr, 2,
MidpointRounding.AwayFromZero)) + "\n";
                    str += str1.PadLeft(20, ' ');
                    richTextBox1.Text += str;
                    richTextBox1.Text +=
"----- \n";
                }
            }
        }
    }
}

```

```
}
```

ЛІСТИНГ модуля Mark.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

namespace Estimate
{
    public partial class Mark : Form
    {
        public Mark()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            double Mark_p;

            double f_m, u_m, r_m, p_m, s_m;
            double w_f = 0, w_u = 0, w_r = 0, w_p = 0, w_s = 0;
            string str_n;
            int fl_s = 0;

            f_m = Convert.ToDouble(tbFunc.Text);
            u_m = Convert.ToDouble(tbUs.Text);
            r_m = Convert.ToDouble(tbRel.Text);
            p_m = Convert.ToDouble(tbPerf.Text);
            s_m = Convert.ToDouble(tbSupr.Text);
            str_n = tbNazva.Text;

            BinaryFormatter bf = new BinaryFormatter();

            using (FileStream fs = new FileStream("baza.bin", FileMode.OpenOrCreate))
            {
                while (fs.Position != fs.Length)
                {
                    Rate t = (Rate)bf.Deserialize(fs);
                    if (t.nazva == str_n)
                    {
                        w_f = t.func;
                        w_u = t.usab;
                        w_r = t.rel;
                        w_p = t.perf;
                        w_s = t.supr;
                        fl_s = 1;
                    }
                }
            }
        }
    }
}
```

```

    }
}

if (f1_s == 1)
{
    Mark_p = Math.Sqrt(w_f * f_m + w_u * u_m + w_r * r_m + w_p * p_m + w_s *
s_m);
    Form3 Fm3 = new Form3();
    Fm3.label14.Text = str_n;
    Fm3.label16.Text = Convert.ToString(Math.Round(Mark_p, 3,
MidpointRounding.AwayFromZero));
    Fm3.Show();
}
else
{
    Form3 Fm3 = new Form3();
    Fm3.label14.Text = "не знайдено оцінок";
    Fm3.label16.Text = Convert.ToString(-1);
    Fm3.Show();
}
}
}
}
}

```

ЛІСТИНГ модуля Diagram.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

namespace Estimate
{
    public partial class Diagrama : Form
    {
        public Diagrama()
        {
            InitializeComponent();
        }

        private void Diagrama_Load(object sender, EventArgs e)
        {
            List<Rate> result = new List<Rate> { };

            BinaryFormatter bf = new BinaryFormatter();

            using (FileStream fs = new FileStream("baza.bin", FileMode.OpenOrCreate))
            {
                while (fs.Position != fs.Length)
                {
                    Rate t = (Rate)bf.Deserialize(fs);
                    result.Add(t);
                }
            }
            int count = result.Count();

```

```
double[] mas_f = new double[count];
double[] mas_u = new double[count];
double[] mas_r = new double[count];
double[] mas_p = new double[count];
double[] mas_s = new double[count];

foreach(Rate p in result)
{
    this.chart1.Series["Функціональність"].Points.AddXY(p.nazva, p.func);
    this.chart1.Series["Usability"].Points.AddXY(p.nazva, p.usab);
    this.chart1.Series["Надійність"].Points.AddXY(p.nazva, p.rel);
    this.chart1.Series["Продуктивність"].Points.AddXY(p.nazva, p.perf);
    this.chart1.Series["Супроводжуваність"].Points.AddXY(p.nazva, p.supr);
}

}

}
```

ДОДАТОК Б
слайди презентації

ДОСЛІДЖЕННЯ МЕТОДІВ МОНІТОРИНГУ СИСТЕМ FRONT-END



Виконав:
ІПЗм-18-1 Швець В.С.
Керівник:
Валенда Н.А.

МЕТА РОБОТИ

- Метою роботи є Дослідження методів моніторингу систем Front-end



ПОСТАВКА ЗАДАЧІ

Метою дослідження методів моніторингу систем Front-end було:

- Зрівняння ферймворків
- Їх популярність при використанні
- Параметри дослідження системи Front-end
- Програмна реалізація зрівнювання

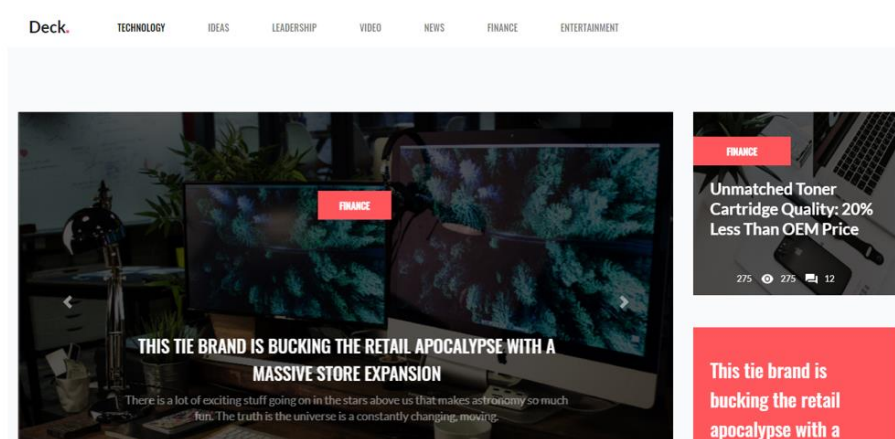
3

ІСТОРІЯ РОЗВИТКУ САЙТІВ



4

ІСТОРІЯ РОЗВИТКУ САЙТІВ



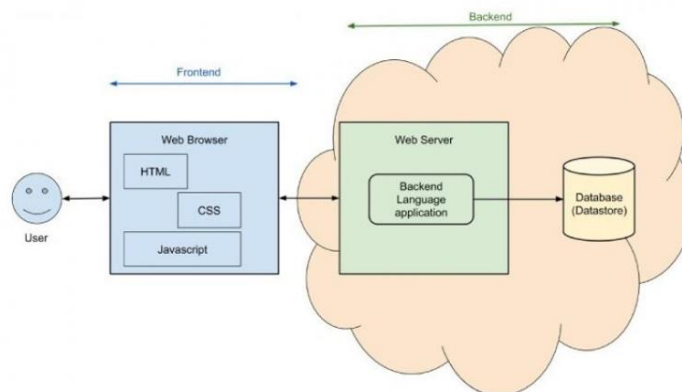
5

ПОПУЛЯРНІ ФРЕЙМВОРКИ

- React;
- Angular;
- Bootstrap;
- Vue.js
- jQuery

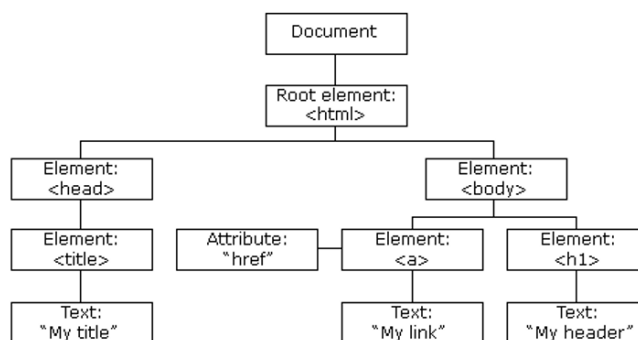
6

УЗАГАЛЬНЕНА СТРУКТУРА ВЕБ-ДОДАТКУ



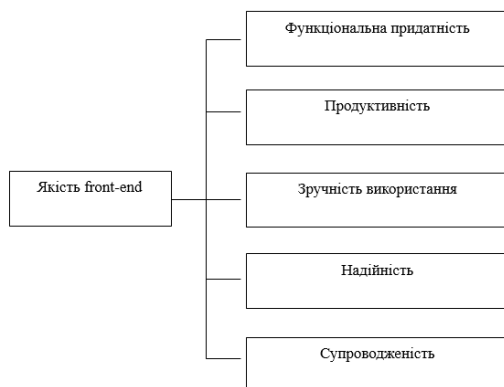
7

ДЕРЕВО ОБ'ЄКТІВ HTML DOM



8

ХАРАКТЕРИСТИКИ ЯКОСТІ FRONT-END



9

РОЗРОБКА МЕТОДИКИ ІНТЕГРАЛЬНОГО ОЦІНЮВАННЯ СИСТЕМ FRONT-END РОЗРОБКИ

Для отримання інтегральної оцінки за групами показників якості використовують фактори якості (1-й рівень): функціональність, надійність, зручність використання, продуктивність, супроводженість.

Для забезпечення об'єктивності процесу оцінювання, шкала ранжується на три рівні прийнятності:

- 0 – 39% – незадовільний рівень;
- 40% – 59% – граничний рівень;
- 60% – 100% – задовільний рівень.

10

Алгоритм реалізації оцінювання якості Front-end додатків містить три фази: вимірювання показників реалізації web-ресурсу, елементарне оцінювання та частинне або загальне оцінювання web-ресурсу

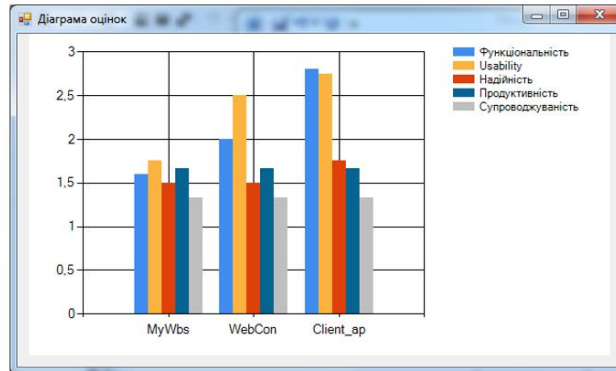


11

ВВЕДЕННЯ ХАРАКТЕРИСТИКИ СИСТЕМИ

12

ПОРІВНЯЛЬНИЙ АНАЛІЗ ХАРАКТЕРИСТИК FRONT-END СИСТЕМ



13

ВИСНОВКИ

- Сучасні розробники при проектуванні веб-сайтів використовують компонентний підхід
- Для розробки **Front-end** систем використовують різні засоби та технології, які допомагають дотримуватися вимог щодо розробки програмного забезпечення.
- Більшість моделей якості представляють собою ієрархічні системи, які складаються з окремих оцінювальних елементів, метрик та критеріїв.
- Запропонована модель якості для **Front-end** систем дозволяє врахувати ці особливості, основною з яких є оцінювання **usability** системи.
- Дана модель може бути врахована в процесі розробки моделі якості веб-додатків. Її можна використати, як джерело даних щодо функціонування **Front-end** системи

14

ДЯКУЮ ЗА УВАГУ



ДОДАТОК В
Електронні матеріали