

Автоматизация оперативного учета и визуализации персонального бюджета на базе современных веб-технологий

Анишкова Анастасия Сергеевна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью исследования является разработка и программная реализация интерактивной веб-ориентированной аналитической панели для автоматизированного учета, категоризации и динамической визуализации личных финансов в режиме реального времени. Выбор формата графического отображения информации обусловлен спецификой анализируемых данных: круговая диаграмма оптимальна для наглядной демонстрации долевой структуры расходов по категориям, в то время как линейный график тающего лимита позволяет пользователю эффективно оценивать временной тренд и скорость расходования бюджета. Такой метод позволяет автоматизировать процесс сопоставления плановых и фактических показателей, обеспечивая синхронное динамическое изменение табличных данных и графических модулей в режиме реального времени при фиксации любой новой операции.

Ключевые слова: HTML, аналитическая панель, веб-интерфейс, расходы.

Automation of operational accounting and visualization of a personal budget based on modern web technologies

Anishkova Anastasia Sergeevna

Sholom-Aleichem Priamurskiy State University

Student

Abstract

The purpose of the study is to develop and implement an interactive web-based analytical dashboard for automated accounting, categorization, and dynamic visualization of personal finances in real time. The choice of the graphical display format is based on the specific nature of the data being analyzed: a pie chart is optimal for visually representing the proportion of expenses by category, while a line graph of a dwindling limit allows the user to effectively assess the time trend and rate of budget consumption. This method allows you to automate the process of comparing planned and actual indicators, ensuring synchronous dynamic changes in tabular data and graphical modules in real time when any new operation is recorded.

Keywords: HTML, analytics panel, web interface, expenses.

1 Введение

1.1 Актуальность

Актуальность исследования обусловлена стремительным развитием веб-технологий и растущей потребностью общества в кроссплатформенных, интуитивно понятных цифровых инструментах управления личным капиталом. Существующие программные решения зачастую перегружены избыточным функционалом или требуют сложной настройки, что актуализирует разработку легковесных, адаптивных и интерактивных веб-панелей, способных агрегировать и наглядно визуализировать финансовые потоки пользователя в режиме реального времени.

1.2 Обзор исследований

В статье рассматривается организация подготовки педагогических работников к использованию технологий HTML5 для разработки электронных образовательных ресурсов в формате веб-приложений. Отмечена актуальность использования мультимедийных интерактивных электронных образовательных ресурсов в современной школе и обоснована эффективность разработки подобных электронных образовательных ресурсов именно в формате офлайновых или онлайновых веб-приложений [1]. Показала процесс создания и использования мультимедийных интерактивных плакатов по информатике К.А. Цилинская. Представила опыт разработки мультимедийных интерактивных плакатов средствами языка HTML5 и Интернет-сервисов [2]. Автоматизация функционального тестирования web-приложений описали Ю.С. Медведев, В. В. Терехов [3]. В работе рассмотрели возможности языка разметки гипертекста HTML5 для создания макета электронного учебника Д.М. Размологов, А.С. Панина, П.В. Захаров. Привели разработанный макет, а также часть таблицы стилей для дизайна макета [4]. Е. В. Кряжева, Т. А. Васина написали об общие подходах к проектированию веб-приложений [5].

1.3 Цель исследования

Целью исследования является разработка и программная реализация интерактивной веб-ориентированной аналитической панели для автоматизированного учета, категоризации и динамической визуализации личных финансов в режиме реального времени.

2 Материалы и методы

В качестве инструментария исследования использован стандартный стек веб-технологий: HTML5 для создания структуры страницы, CSS3 для настройки адаптивного дизайна и JavaScript для реализации логики расчетов в реальном времени. Для построения интерактивных круговых и линейных графиков была интегрирована специализированная библиотека Chart.js.

3 Результаты

Для создания аналитической панели «Разумный финансовый мониторинг» предусмотрен учет установленного бюджета, расчет общей

суммы потраченных средств и определение оставшегося лимита; интерфейс веб-страницы для мониторинга личных финансов организован в три адаптивные колонки, включает форму настройки бюджета, блок добавления расходов по категориям, компактную таблицу истории операций и два аналитических графика на базе Chart.js, а встроенный JavaScript в реальном времени пересчитывает расходы, обновляет круговую диаграмму и отображает остаток бюджета на линейном графике, который автоматически меняет цвет на красный при перерасходе (рис.1).

Фрагмент кода:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Разумный финансовый мониторинг</title>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</style>
* {
  box-sizing: border-box;
  font-family: 'Segoe UI', system-ui, sans-serif;
  margin: 0;
  padding: 0;
}

body {
  background-color: #f0f2f5;
  color: #1a1a1a;
  padding: 25px;
}

.container {
  max-width: 1400px;
  margin: 0 auto;
}

h1 {
  margin-bottom: 25px;
  color: #2c3e50;
  font-size: 28px;
}

/* Карточки метрик */
.summary-cards {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
  gap: 20px;
  margin-bottom: 30px;
}

.card {
  background: #fff;
  padding: 20px;
  border-radius: 12px;
  box-shadow: 0 4px 12px rgba(0,0,0,0.05);
}

.card h3 {
  font-size: 13px;
  color: #7f8c8d;
  text-transform: uppercase;
  margin-bottom: 8px;
}

.card .amount-val {
  font-size: 24px;
  font-weight: bold;
}

#global-total { color: #e74c3c; }
#budget-left { color: #2ecc71; }

/* Сетка */
```

```
.dashboard-grid {
  display: grid;
  grid-template-columns: 1.2fr 1fr 1fr;
  gap: 25px;
}

@media (max-width: 1100px) {
  .dashboard-grid { grid-template-columns: 1fr 1fr; }
}

@media (max-width: 768px) {
  .dashboard-grid { grid-template-columns: 1fr; }
}

.panel-box {
  background: #fff;
  padding: 25px;
  border-radius: 12px;
  box-shadow: 0 4px 12px rgba(0,0,0,0.05);
  display: flex;
  flex-direction: column;
}

.panel-box h2 {
  font-size: 18px;
  margin-bottom: 15px;
  color: #34495e;
  border-bottom: 2px solid #f0f2f5;
  padding-bottom: 10px;
}

/* Формы и элементы управления */
.budget-setup {
  display: flex;
  gap: 10px;
  margin-bottom: 20px;
  align-items: center;
}

.expense-form {
  display: flex;
  flex-direction: column;
  gap: 12px;
  margin-bottom: 25px;
}

.form-group {
  display: flex;
  gap: 10px;
}

select, input {
  padding: 10px;
  border: 1px solid #ced4da;
  border-radius: 6px;
  font-size: 15px;
  background-color: #fff;
}

select { flex: 1.3; }
input[type="number"] { flex: 1; }

button {
  padding: 10px 15px;
  color: white;
  border: none;
  border-radius: 6px;
  cursor: pointer;
  font-size: 15px;
  font-weight: bold;
  transition: background 0.2s;
}

.btn-add { background-color: #2ecc71; }
.btn-add:hover { background-color: #27ae60; }
.btn-set { background-color: #3498db; }
.btn-set:hover { background-color: #2980b9; }
```

```

/* Таблица */
.table-wrapper {
  max-height: 250px;
  overflow-y: auto;
}

table {
  width: 100%;
  border-collapse: collapse;
}

th, td {
  padding: 10px;
  text-align: left;
  border-bottom: 1px solid #dee2e6;
  font-size: 14px;
}

th {
  background-color: #f8f9fa;
  color: #6c757d;
  position: sticky;
  top: 0;
}

.category-badge {
  display: inline-block;
  padding: 4px 8px;
  border-radius: 4px;
  font-size: 11px;
  font-weight: bold;
  color: white;
}

.chart-container {
  position: relative;
  margin: auto;
  width: 100%;
  height: 280px;
}
</style>
</head>
<body>
<div class="container">
  <h1>Разумный финансовый мониторинг</h1>

  <div class="summary-cards">
    <div class="card">
      <h3>Установленный бюджет</h3>
      <div class="amount-val" style="color: #3498db;"><span id="budget-total">50 000</span> P</div>
    </div>
    <div class="card">
      <h3>Всего потрачено</h3>
      <div class="amount-val"><span id="global-total">0</span> P</div>
    </div>
    <div class="card">
      <h3>Оставшийся лимит</h3>
      <div class="amount-val"><span id="budget-left">50 000</span> P</div>
    </div>
  </div>

  <div class="dashboard-grid">

    <div class="panel-box">
      <h2>Настройка бюджета</h2>
      <div class="budget-setup">
        <input type="number" id="budget-input" value="50000" placeholder="Лимит на месяц">
        <button class="btn-set" onclick="updateBudgetLimit()">Изменить</button>
      </div>

      <h2>Добавить расход</h2>
      <form class="expense-form" onsubmit="addExpense(event)">
        <div class="form-group">
          <select id="category-select" required>
            <option value="" disabled selected>Категория</option>
            <option value="Продукты">🛒 Продукты</option>
            <option value="Жилье и ЖКХ">🏠 Жилье и ЖКХ</option>
            <option value="Транспорт">🚗 Транспорт</option>
          </select>
        </div>
      </form>
    </div>
  </div>
</div>

```

```

<option value="Развлечения">🎮 Развлечения</option>
<option value="Одежда">👕 Одежда</option>
<option value="Другое">📦 Другое</option>
</select>
<input type="number" id="amount-input" min="1" step="any" placeholder="Сумма" required>
</div>
<button type="submit" class="btn-add">Добавить расход</button>
</form>

<h2>История операций</h2>
<div class="table-wrapper">
  <table>
    <thead>
      <tr>
        <th>Категория</th>
        <th>Сумма</th>
      </tr>
    </thead>
    <tbody id="expense-table-body">
      <tr id="empty-row">
        <td colspan="2" style="color: #95a5a6; text-align: center;">Трат пока нет</td>
      </tr>
    </tbody>
  </table>
</div>
</div>

<div class="panel-box">
  <h2>Куда уходят деньги (%)</h2>
  <div class="chart-container">
    <canvas id="doughnutChart"></canvas>
  </div>
</div>

<div class="panel-box">
  <h2>Динамика остатка бюджета</h2>
  <div class="chart-container">
    <canvas id="lineChart"></canvas>
  </div>
</div>
</div>
</div>

```

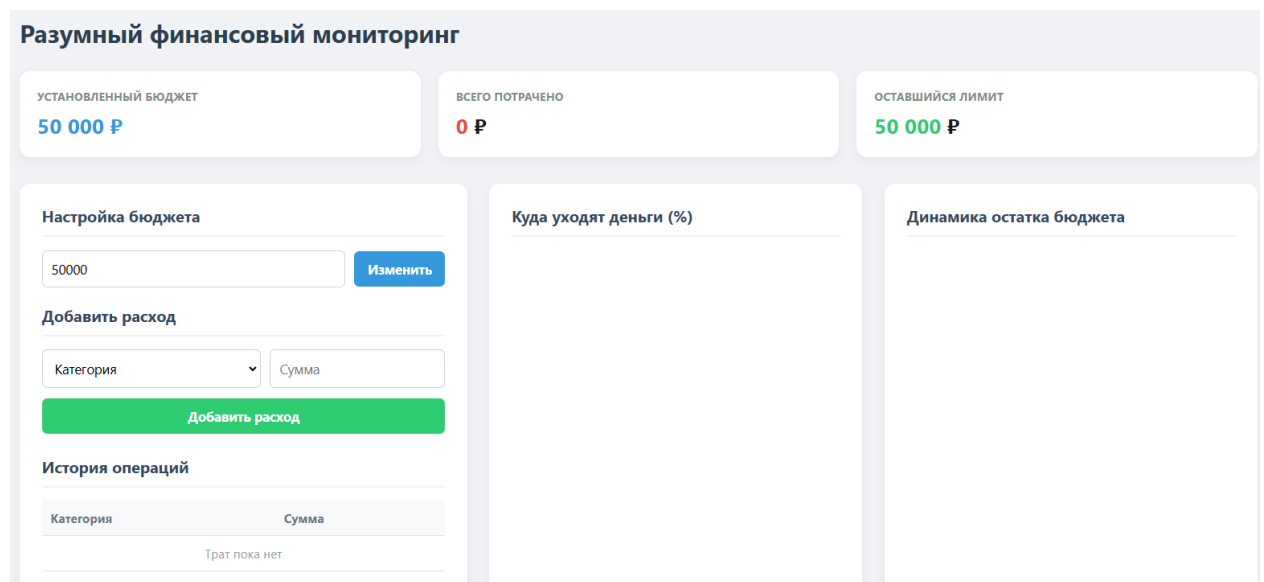


Рисунок 1. Разумная аналитическая модель

Следующий фрагмент кода обеспечивает интерактивную логику финансовой панели, управляя вычислениями, динамическим обновлением HTML-интерфейса и визуализацией данных через библиотеку Chart.js без

перезагрузки страницы. При загрузке он задает цветовые схемы категорий, инициализирует стартовый лимит бюджета и отрисовывает две пустые диаграммы: круговую для долей расходов и линейную для отображения остатка средств (рис.2).

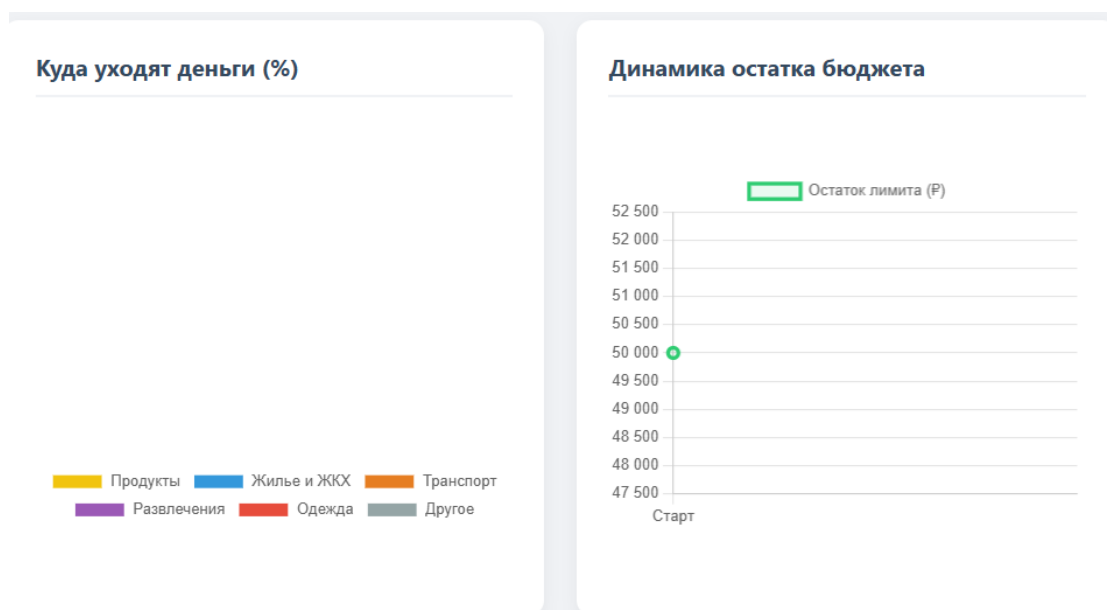


Рисунок 2. Пустые диаграммы

Две основные функции обрабатывают ввод пользователя: одна корректно пересчитывает всю шкалу графиков при изменении общего месячного лимита, а вторая распределяет каждую новую трату по категориям, добавляет запись в интерактивную таблицу истории и фиксирует новый шаг уменьшения бюджета. Финальная функция синхронизирует эти расчеты с экраном, форматирует числа с разделением разрядов и автоматически перекрашивает элементы графиков и карточку баланса из зеленого в красный цвет в случае фиксации перерасхода (рис.3).

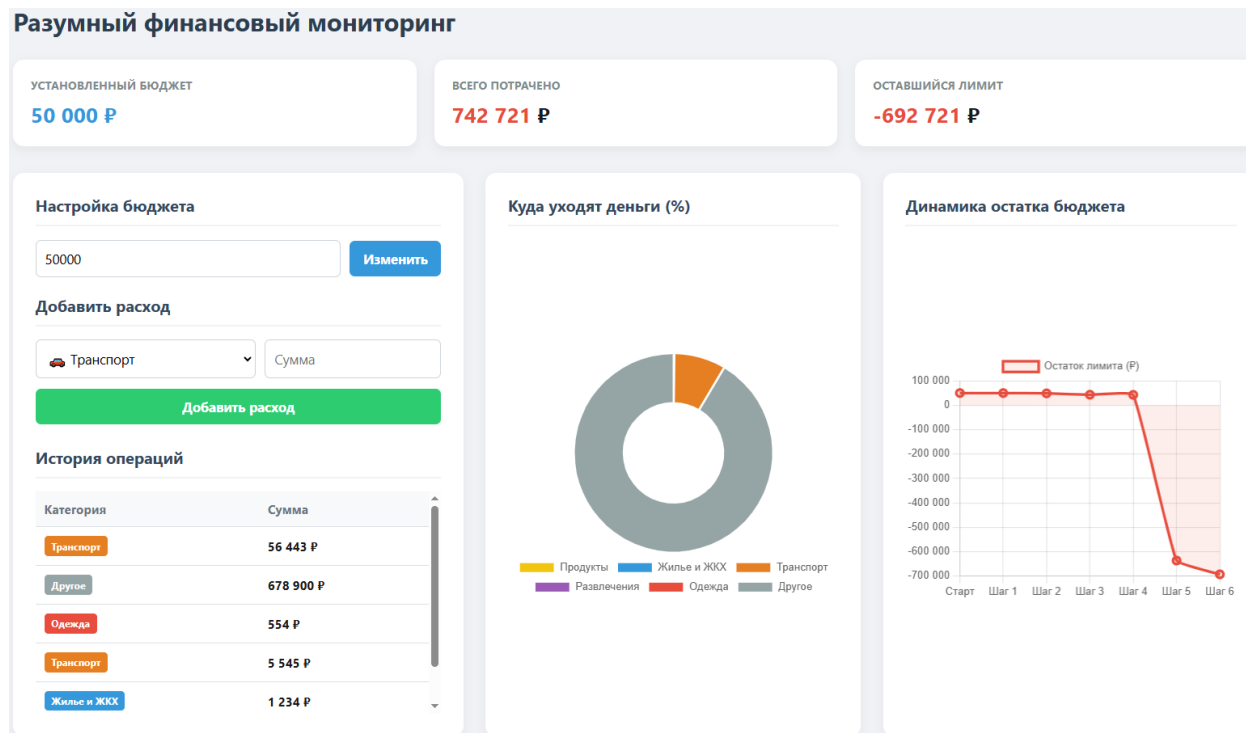


Рисунок 3. Разумный финансовый мониторинг

Фрагмент кода:

```

<script>
const categoryColors = {
  'Продукты': '#f1c40f', 'Жилье и ЖКХ': '#3498db', 'Транспорт': '#e67e22',
  'Развлечения': '#9b59b6', 'Одежда': '#e74c3c', 'Другое': '#95a5a6'
};

const categoryTotals = {
  'Продукты': 0, 'Жилье и ЖКХ': 0, 'Транспорт': 0, 'Развлечения': 0, 'Одежда': 0, 'Другое': 0
};

let monthlyLimit = 50000;
let totalSum = 0;
let doughnutChart = null;
let lineChart = null;

// Массивы для графика остатка бюджета
const timelineLabels = ['Старт'];
const timelineData = [50000]; // Изначально остаток равен лимиту
let transactionCount = 0;

window.onload = function() {
  // Круговой график
  const ctxDoughnut = document.getElementById('doughnutChart').getContext('2d');
  doughnutChart = new Chart(ctxDoughnut, {
    type: 'doughnut',
    data: {
      labels: Object.keys(categoryTotals),
      datasets: [{
        data: Object.values(categoryTotals),
        backgroundColor: Object.values(categoryColors),
        borderWidth: 2
      }]
    },
    options: {
      responsive: true,
      maintainAspectRatio: false,
      plugins: { legend: { position: 'bottom' } }
    }
  });

  // Линейный график остатка бюджета
  const ctxLine = document.getElementById('lineChart').getContext('2d');
  lineChart = new Chart(ctxLine, {

```

```

    type: 'line',
    data: {
      labels: timelineLabels,
      datasets: [{
        label: 'Остаток лимита (P)',
        data: timelineData,
        borderColor: '#2ecc71',
        backgroundColor: 'rgba(46, 204, 113, 0.1)',
        fill: true,
        tension: 0.1,
        borderWidth: 3,
        pointRadius: 4
      }]
    },
    options: {
      responsive: true,
      maintainAspectRatio: false,
      scales: {
        y: {
          // Позволяет графику уходить в минус при перерасходе
          beginAtZero: false
        }
      }
    }
  });
});
};

// Изменение лимита бюджета пользователем
function updateBudgetLimit() {
  const budgetInput = document.getElementById('budget-input');
  const newLimit = parseFloat(budgetInput.value);

  if (!isNaN(newLimit) && newLimit > 0) {
    const diff = newLimit - monthlyLimit;
    monthlyLimit = newLimit;

    // Пересчитываем стартовую точку и всю историю для графика
    timelineData[0] = monthlyLimit;
    for (let i = 1; i < timelineData.length; i++) {
      timelineData[i] += diff;
    }

    document.getElementById('budget-total').textContent = monthlyLimit.toLocaleString('ru-RU');
    updateUI();
  }
}

function addExpense(event) {
  event.preventDefault();

  const categorySelect = document.getElementById('category-select');
  const amountInput = document.getElementById('amount-input');

  const category = categorySelect.value;
  const amount = parseFloat(amountInput.value);

  if (!category || !isNaN(amount)) return;

  // Математика
  totalSum += amount;
  categoryTotals[category] += amount;
  transactionCount++;

  // Таблица
  const tbody = document.getElementById('expense-table-body');
  const emptyRow = document.getElementById('empty-row');
  if (emptyRow) emptyRow.remove();

  const row = document.createElement('tr');
  row.innerHTML = `
    <td><span class="category-badge" style="background-color: ${categoryColors[category]}">${category}</span></td>
    <td><strong>${amount.toLocaleString('ru-RU')} P</strong></td>
  `;
  tbody.insertBefore(row, tbody.firstChild);

  // Расчет остатка для графика
  const currentLeft = monthlyLimit - totalSum;
  timelineLabels.push(`Илr ${transactionCount}`);

```

```
        timelineData.push(currentLeft);

        updateUI();
        amountInput.value = "";
    }

    function updateUI() {
        const currentLeft = monthlyLimit - totalSum;

        // Обновление карточек сверху
        document.getElementById('global-total').textContent = totalSum.toLocaleString('ru-RU');
        const leftBox = document.getElementById('budget-left');
        leftBox.textContent = currentLeft.toLocaleString('ru-RU');

        // Меняем цвет текста остатка и графика, если ушли в перерасход
        if (currentLeft < 0) {
            leftBox.style.color = '#e74c3c'; // Красный при перерасходе
            lineChart.data.datasets[0].borderColor = '#e74c3c';
            lineChart.data.datasets[0].backgroundColor = 'rgba(231, 76, 60, 0.1)';
        } else {
            leftBox.style.color = '#2ecc71'; // Зеленый, если всё ок
            lineChart.data.datasets[0].borderColor = '#2ecc71';
            lineChart.data.datasets[0].backgroundColor = 'rgba(46, 204, 113, 0.1)';
        }

        // Синхронизация графиков
        doughnutChart.data.datasets[0].data = Object.values(categoryTotals);
        doughnutChart.update();
        lineChart.update();
    }
</script>

</body>
</html>
```

4 Выводы

В результате проведенного исследования была успешно спроектирована и программно реализована интерактивная веб-панель «Разумный финансовый мониторинг». Разработанный инструмент на базе HTML5, CSS3, JavaScript и библиотеки Chart.js доказал свою эффективность в решении задач оперативного учета личных финансов. Благодаря переносу вычислений на сторону клиента и интеграции динамической графики, система обеспечивает мгновенный отклик интерфейса, наглядное распределение расходов по категориям и автоматическое предупреждение о перерасходе бюджета в реальном времени.

Библиографический список

1. Бакулевская С. С. Подготовка педагогических работников к использованию технологий html5 для разработки электронных образовательных ресурсов // Информатика и образование. 2019. №. 5. С. 32-40.
2. Цилинская К.А. Создание и использование мультимедийных интерактивных плакатов по информатике //Будущее науки-2019. 2019. С. 340-344.
3. Медведев Ю. С., Терехов В. В. Автоматизация функционального тестирования web-приложений //Современные наукоемкие технологии. 2016. №. 5-1. С. 58-61.
4. Размологов Д. М., Панина А. С., Захаров П. В. Использование html5 для разработки электронного учебника //Современные тенденции развития

науки и технологий. 2016. №. 4-4. С. 92-94.

5. Кряжева Е. В., Васина Т. А. Общие подходы к проектированию ВЕБприложений //Заметки ученого. 2021. №. 9-2. С. 32-36.